Advances in Artificial Intelligence

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov, CIC-IPN, Mexico
Gerhard X. Ritter, University of Florida, USA
Jean Serra, Ecole des Mines de Paris, France
Ulises Cortés, UPC, Barcelona, Spain

Associate Editors:

Jesús Angulo, Ecole des Mines de Paris, France
Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel
Alexander Gelbukh, CIC-IPN, Mexico
Ioannis Kakadiaris, University of Houston, USA
Petros Maragos, Nat. Tech. Univ. of Athens, Greece
Julian Padget, University of Bath, UK
Mateo Valero, UPC, Barcelona, Spain
Rafael Guzmán, Univ. of Guanajuato, Mexico
Juan Manuel Torres Moreno, U. of Avignon, France

Editorial Coordination:

Alejandra Ramos Porras

Research in Computing Science, Año 19, Volumen 149, No. 8, agosto de 2020, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 30 de agosto de 2020.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

Research in Computing Science, year 19, Volume 149, No. 8, August 2020, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research of the IPN.

Volume 149(8)

Advances in Artificial Intelligence

Oscar Herrera Alcántara
Noé Alejandro Castro Sánchez (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2020

ISSN: in process

Copyright © Instituto Politécnico Nacional 2020
Formerly ISSNs: 1870-4069, 1665-9899.

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Editorial

Este volumen incluye noventa y tres artículos relacionados con la Inteligencia Artificial, en temas y aplicaciones de procesamiento de lenguaje natural, procesamiento de imágenes y reconocimiento de patrones, aprendizaje automático, y algoritmos bioinspirados.

Los trabajos se seleccionaron por un comité editorial, con el apoyo de revisores externos que evaluaron su calidad y originalidad.

Los artículos de procesamiento del lenguaje natural abordan temas de análisis de datos en redes sociales como incidencia delictiva, propagación de contagios durante la pandemia del COVID-19, críticas de cine, detección de humor en textos, noticias falsas en sitios web y redes sociales, entre otros temas.

Los artículos de procesamiento de imágenes se relacionan con el seguimiento de trayectorias y localización de robots, seguimiento de objetos con redes de sensores, análisis de la postura esquelética de la mano para representar lenguaje de señas, detección y clasificación de señales de tráfico, clasificación de granos de polen, así como clasificación de imágenes de retina, entre otros temas.

En cuanto a temas de aprendizaje automático, se incluye el análisis de datos de detección de fraudes, señales de EEG, diabetes mellitus, detección de intrusos, identificación de biomarcadores de bacterias, así como la aplicación en la corrección en sistemas de reconocimiento del habla, entre otras aplicaciones.

La calibración de hiperparámetros en evolución diferencial para clasificación de requisitos de software, la evolución automática de circuitos analógicos, la implementación de un algoritmo genéticos en hardware, la optimización multiobjetivo de un microalgoritmo genético para la asignación de cargas horarias, y el estudio de redes neuronales evolutivas, entre otros temas se estudian en artículos relacionados con la optimización y el cómputo evolutivo.

También se incluyen artículos de aplicaciones para predecir el desempeño académico, la adopción de medios digitales en la banca de inversión en México, la integración de un brazo robótico en aplicaciones de la industria 4.0, la evaluación del abandono de clientes en una compañía de telecomunicaciones, el uso de un sistema multiagente en el control de tráfico urbano, el uso de una ontología de estilos de aprendizaje para asistir el aprendizaje significativo, el enfoque semántico de políticas para gestionar la conciencia de grupo en groupware, y el estudio del comportamiento de peregrinos en un evento de sismo para planificar rutas de evacuación, entre otras aplicaciones.

Oscar Herrera, UAM Azcapotzalco, México
Noé Castro, CENIDET, México
Guest Editors

Agosto 2020

ISSN 1870-4069

Table of Contents

Page

Skill de Alexa para la enseñanza de la programación estructurada 19
Luis Japeth Trujillo-Mendieta, Oscar Herrera-Alcántara

Aplicación de una red neuronal probabilística para predecir el desempeño académico de estudiantes de educación superior en línea 31
Rosa Leonor Ulloa Cazarez

Clasificadores basados en reglas y selección de atributos para el diagnóstico clínico de subtipos del Síndrome de Guillain-Barré 41
Daniel Alarcón-Narváez, Jerusa Marchi, José Hernández-Torruco, Oscar Chávez-Bosquez, Betania Hernández-Ocaña

Rendimiento de los métodos de sobremuestreo para los modelos predictivos del Síndrome de Guillain-Barré 55
Manuel Torres-Vásquez, José Hernández-Torruco, Betania Hernández-Ocaña, Oscar Chávez-Bosquez

Control difuso para el seguimiento de trayectoria de un robot móvil 71
Camilo Morales Corral, Israel Soto

Localización de un robot móvil mediante el filtro de Kalman extendido y el simulador Gazebo 85
Rolan Bacilio Anota, Eduardo Sánchez-Watanabe, Alberto Petrilli-Barceló, Fermín Ramírez-Leyva

Algoritmo de clasificación basado en la función Softmax 95
Arturo Zúñiga-López, Carlos Avilés-Cruz, Andrés Ferreyra-Ramírez, Eduardo Rodríguez-Martínez

Inteligencia artificial en la prueba de software: una revisión sistemática de la literatura 109
Alfonso Robles-Aguilar, Angel Juan Sanchez-Garcia, Jorge Octavio Ocharán-Hernández

Modelo de clasificación para la adopción de medios digitales en la Banca de Inversión 123
Jessica Guadalupe Vega Ledesma, Cristina Hernández Márquez, Alma Chávez Arrieta

Extracción de eventos en electrocardiograma mediante el procesamiento de señales, utilizando el algoritmo MTEO 135
Leticia Gómez Rivera, María Margarita Labastida Roldán

Generalización de requisitos de software de dominio específico para clasificación de texto 145
J. Manuel Pérez-Verdejo, Ángel Juan Sánchez-García, Jorge Octavio Ocharán-Hernández

Sistema portátil no invasivo para el monitoreo y desplegado de los signos vitales de pacientes en salas de urgencias en un dispositivo móvil 157
Israel Santoyo-Luévano, Carlos Avilés-Cruz, Arturo Zuñiga-López

Clasificación de objetos mediante momentos de Hahn 3D y aprendizaje profundo 173
Saúl Rivera-López, Lucia Gutiérrez-Lazcano, Horlando Vargas-Vargas, Alicia Noriega-Escamilla

Momentos de Bessel-Fourier de orden fraccional para el análisis de imágenes a color 187
Horlando Vargas-Vargas, César Camacho-Bello, José S. Rivera-López, Alicia Noriega-Escamilla

Reconocimiento de imágenes RGB con momentos multicanal de Zernike 201
Alicia Noriega-Escamilla, César Camacho-Bello, José Saúl Rivera-López, Horlando Vargas-Vargas

Detección automática de zonas de alto riesgo de eventos delictivos a través de noticias periodísticas 213
Yadira Laureano de Jésus, Guillermo De Ita Luna, Mireya Tovar Vidal

Postura esquelética de la mano para representar las configuraciones dactilológicas de la lengua de señas mexicana en un entorno virtual 227
Alejandro Gómez-Terán, Belém Priego-Sánchez, David Pinto

Calibración de hiperparámetros para la clasificación de requisitos de software mediante evolución diferencial 241
J. Manuel Pérez-Verdejo, Efrén Mezura-Montes, Ángel Juan Sánchez-García, Jorge Octavio Ocharán-Hernández

Mecanismo de representación para la evolución automática de circuitos analógicos 253
Aurora Torres Soto, María Dolores Torres Soto, Eunice Esther Ponce de León Sentí

Modelos de aprendizaje automático para clasificar patologías oculares en imágenes de fondo de ojo como prueba de tamizaje 267
Miguel Ángel Méndez Lucero, E. Mahuina Campos Castolo, Dania N. Lima Sánchez, Josué Fabricio Urbina González, Orlando Cerón Solís, Alejandro Alayola-Sansores, German Fajardo Dolci

Chalecos amarillos CHA: un análisis a través de diccionarios en Python y teoría de medios 281
María Beatríz Bernabé Loranca, Alberto José Luís Carrillo Canán, Marleni Reyes Monreal

Identificación y localización de células gliales en imágenes de tejido cerebral 297
Karla Mandujano-López, Juan Villegas-Cortez, Graciela Román-Alonso, Arturo Zuñiga-López, Edgar Rangel-López, César Benavides-Alvarez

Implementación de una red neuronal profunda en tres etapas paralelas para el reconocimiento de actividades humanas e imágenes 311
Yafte A. Flores-Morales, Juan Villegas-Cortez, Graciela Román-Alonso, Arturo Zúñiga-López, César Benavides-Álvarez, Salomón Cordero-Sánchez

Red neuronal convolucional de baja latencia para la estimación de profundidad monocular 327
Edgar Rodrigo López Silva, Luis Rogelio Román Rivera, Jesús Carlos Pedraza Ortega, Marco Antonio Aceves Fernández, Juan Manuel Ramos Arreguín

Localización de referencias cefalométricas en radiografías laterales de cráneo: Una revisión sistemática 339
José Luis López-Ramírez, José Ruiz-Pinales, Juan Gabriel Aviña-Cervantes, Pedro López-Rodríguez, José Guadalupe Beltrán-Hernández, Mario Alberto Ibarra-Manzano

Propuesta de asistente virtual basado en inteligencia artificial en la integración de un brazo robótico para aplicaciones en la industria 4.0 353
Miguel Ángel Hernández Figueroa, Rodrigo Gibran Vázquez Báez, José Orlando Castro Rangel

Análisis comparativo de la calidad de agrupamiento del algoritmo Intuitionistic Fuzzy C-Means 365
Celia Ramos-Palencia, Dante Mújica-Vargas, Manuel Mejía-Lavalle

Implementación en hardware de un algoritmo genético para resolver un problema combinatorio 379
Eliseo Díaz-Nacar, Katya Rodríguez-Vázquez

Análisis con aprendizaje automático en el proceso de flotación de burbujas para la obtención de minerales 393
Nelsón Romero-García, Sonia Regina Meneses, Julio Waissman Vilanova, Oscar Mario Rodríguez Elias, César Enrique Rose Gómez, Álvaro Rendón Montoya

Aplicación del algoritmo K-means para el análisis de resultados de la prueba PLANEA 2017 407
Israel Gutiérrez González, Doricela Gutiérrez Cruz, Jenny Elizabeth Juan Ramírez, Liliana Rodríguez Paez, Ricardo Rico Molina, Mauricio Sánchez Medina

Estudio comparativo de algoritmos de súper resolución de una sola imagen basados en aprendizaje profundo 421
Leandro José Rodríguez Hernández, Humberto de Jesús Ochoa Domínguez, Osslan Osiris Vergara Villegas, Juan Humberto Sossa Azuela

Detección y clasificación de señales de tráfico mexicanas mediante aprendizaje profundo 435
Rubén Castruita Rodríguez, Carlos Mendoza Carlos, Osslan Osiris Vergara Villegas, Vianey Guadalupe Cruz Sánchez

Análisis de la incidencia delictiva del fuero común mediante la red social Twitter 451
Rosa G. Hernández Cuacua, Sandra D. Rosas Ortega, M. Beatriz Bernábe Loranca

Identificación y pesado de términos para la detección de depresión en Twitter 465
Monserrat Vázquez-Hernández, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez

Análisis de Sentimiento a nivel de documento en críticas de cine en español 475
Maritza Flores Domínguez, José Luis Tapia Fabela

Detección de fraude interno utilizando metodologías de aprendizaje máquina 489
Ulises Jiménez Cardoso, José M. Sarmiento Chávez

Determinación de factores involucrados en la toma de decisiones conductuales mediante minería de datos 501
Carlos Hiram Moreno Montiel, José Luis Bernal López, Ezequiel Alpuche de la Cruz, Benjamín Moreno Montiel, Marisol Sandoval Ríos

Contrastación de algoritmos de aprendizaje automático para la clasificación de señales EEG 515
Alexis Edmundo Gallegos Acosta, María Dolores Torres Soto, Aurora Torres Soto, Eunice Esther Ponce de León Sentí

Mecanismo de clasificación para diabetes mellitus en la población de Aguascalientes, México 527
Pablo Rodríguez de León, María Dolores Torres Soto, Aurora Torres Soto

Redes neuronales artificiales y árboles de decisión para la clasificación con datos categóricos 541
C. Guadalupe Origel-Rivas, Eréndira Rendón-Lara, Itzel María Abundez-Barrera, Roberto Alejo-Eleuterio

Parametrización de terapia física para falanges de la mano humana 555
Juan Carlos Belen Luna, José Martin Flores Albino, Héctor Rafael Orozco Aguirre, Carlos Omar González Morán

Extracción de respuestas a partir de ontologías utilizando patrones estructurales en SQWRL 571
Gabriela A. García-Robledo, Beatriz A. González-Beltrán, José A. Reyes-Ortiz, Maricela Bravo

Usando representaciones distribuidas para el análisis de técnicas propagandísticas en noticias falsas 587
Jennifer Pérez-Santiago, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez

Selección de características con método wrapper para un sistema de detección de intruso: caso CICIDS-2017 599
Edson Manuel Ortiz Martínez, Pedro Arguijo, Antonio Hiram Vázquez López, Roberto Ángel Meléndez Armenta

Evaluación del abandono de clientes de una compañía de telecomunicaciones por medio de cuatro modelos de aprendizaje máquina 611
Jorge Octavio Castro Rodríguez, Ernesto Pérez Vázquez

Estimación de la frecuencia cardíaca con rPPG: Comparación entre diferentes señales rPPG 625
Oswaldo Vázquez, Pedro Arguijo, Antonio Hiram Vázquez López, Roberto Ángel Meléndez Armenta

Reconocimiento de patrones para la clasificación de componentes argumentales en textos académicos en español 637
Kenia Nieto-Benitez, Noé Alejandro Castro-Sánchez, Héctor Jiménez-Salazar

Red neuronal Backpropagation para clasificación de pinturas cubistas aplicando segmentación y momentos invariantes de Hu (CIBUrnbp) 649
Miriam de la Cruz Teomitzi, Ma. Margarita Labastida Roldán

Reconocimiento de caracteres manuscritos a partir de señales EMG usando redes neuronales convolucionales y recurrentes de Elman 659
José Guadalupe Beltrán-Hernández, José Ruiz-Pinales, Pedro López-Rodriguez, José Luis López-Ramírez

Modelo de agrupamiento de precios en el autoservicio 669
Leopoldo Javier Gómez Álvarez, Marco Antonio Aristeo García, Luis Alberto Álvarez Ayala

Clasificación de plantas de maíz y maleza: Hacia la mejora de la fertilización en México 683
Marcos Yamir Gómez-Ramos, J. Sergio Ruíz-Castilla, Farid García-Lamont

Análisis comparativo de métodos de desdoblamiento de fase en perfilometría por desplazamiento de fase 699
Luis Arturo Alvarado Escoto, Jesús Carlos Pedraza Ortega, Juan Manuel Ramos Arreguin, Efren Gorrostieta Hurtado, Marco Antonio Aceves Fernandez

Sistema multi-agente para el control de tráfico urbano para vehículos de prioridad 713
Guillermo Fernández Estrella, Joel Antonio Trejo Sánchez, Mario Renán Moreno Sabido

Comparación de dos modelos para resolver el problema de implicación textual multilingüe 725
Darnes Vilariño, Saul Leon Silverio, Beatriz Beltrán

Evaluación del modelo neuronal de atención visual en la descripción automática de imágenes en español 739
Rafael Gallardo-García, Beatriz Beltrán-Martínez, Darnes Vilariño Ayala

Clasificación de granos de polen: Efecto de la distancia inter-pixel en GLCM 753
Leyra Martínez, Pedro Arguijo, Antonio Hiram Vázquez López, Roberto Ángel Meléndez Armenta

Predicción del éxito de proyectos de videojuegos en Kickstarter con aprendizaje automático 763
Salvador Nevarez-Castellanos, Vicente García, Gilberto Rivera, Rogelio Florencia, Julia Patricia Sánchez-Solís, Alberto Ochoa-Zezzatti

Implementación de sistema de visión inteligente para reconocimiento de rostros en robot de código abierto BOB 777
Carlos Leal-Beltrán, Juan Villegas-Cortez, Arturo Zúñiga-López, Salomón Cordero-Sánchez

Generación de cargas horarias para estudiantes basadas en un algoritmo micro-genético 791
Brenda Sunuami González López, René Arnulfo García-Hernández

Aprendizaje automático aplicado en física: Una revisión de la literatura científica 803
Iris Iddaly Méndez-Gurrola, Abdiel Ramírez-Reyes, Román Anselmo Mora-Gutiérrez

Nuevo enfoque para la extracción de características en la clasificación de textos para la atribución de autoría 817
Omar González Brito, José Luis Tapia Fabela, Silvia Salas Hernández

Minería de texto para identificar las principales preocupaciones de los usuarios de Twitter durante COVID-19 en la Ciudad de México 827
Anabel Pineda-Briseño, Josimar Edinson Chire Saire

Reconocimiento facial basado en Least-Squares Support Vector Machine y características LNDP 841
Pablo Augusto Campos-Zárate, Efrén Gorrostieta-Hurtado, Jesús Carlos Pedraza-Ortega

Clasificación de A/V en imágenes de retina usando arquitectura inspirada en U-Net con ConvLSTM bidireccionales 853
M.G. Hernández-Ramírez, Saul Tovar-Arriaga, Jesús Carlos Pedraza-Ortega, Efrén Gorrostieta-Hurtado

Representación de información de una institución educativa a través de XML 867
Juan Manuel Cancino-Gordillo, Luis Yael Méndez-Sánchez, Francisco Valdez-Tovar, Mireya Tovar-Vidal

Control difuso mediante la estimación mejorada de la distancia inter-electrodos en un proceso de micro maquinado electroquímico por pulsos 879
Octavio Catarino Aguilar, Iván Gallegos Miranda, Everardo Efrén Granda Gutiérrez, José Luis Tapia Fabela, José Arturo Pérez Martínez

Sobre la automatización del razonamiento y el aprendizaje en lógicas modales 893
Yensen Limón, Everardo Bárcenas, Guillermo Molero-Castillo, Rocío Aldeco-Pérez

Detección de noticias falsas en sitios web y redes sociales: Una investigación del estado del arte 907
Sergio Damián, Alexander Gelbukh, Hiram Calvo

Combinación de rankings como método para la identificación de biomarcadores de vaginosis bacteriana 915
Jesús Francisco Pérez-Gómez, Juana Canul-Reich, Erick De-La-Cruz-Hernandez

Ontología de estilos de aprendizaje para asistir en el aprendizaje significativo 929
Yuridiana Alemán, María-Josefa Somodevilla, Darnes Vilariño

Propuesta para determinar la ruta más corta utilizando la arquitectura M-ANFIS 943
Eduardo Chandomí-Castellanos, Elías N. Escobar-Gómez, Sabino Velázquez-Trujillo, J. Alexander López-Rosas, Carlos Venturino De-Coss-Pérez, Yordi López-Escalante, Daniel Domínguez-Romero, Alejandro Flores-Méndez

Clasificación del género de peatones y conductores basada en su comportamiento en la vía pública 957
Francisco Fernando Torres Rosas, Maricela Quintana López, Héctor Rafael Orozco Aguirre, Saul Lazcano Salas

Procesamiento de Lenguaje Natural: una solución para detectar noticias falsas sobre la 4T en México 971
M. Beatriz Bernábe Loranca, Mayra Vanessa Montiel Estrada, Rogelio González Velázquez, Dylan Salomón Lozada Mendoza

Máquinas de soporte vectorial para pronóstico en la osteosíntesis de la fractura transtrocantérica de cadera 985
María Dolores Torres Soto, Aurora Torres Soto, Julio César Aguilar Gómez, María de la Luz Torres Soto, Jesús Adrián Lozano Martínez, Eunice Esther Ponce de León Sentí

Modelo de inferencia difuso en la recomendación de tallas de prendas de vestir en el comercio electrónico 997
Eliana Cornejo Cortez

Reconocimiento de trayectorias del brazo usando sensores de aceleración y giroscopio 1011
Pedro López-Rodríguez, José Ruiz-Pinales, Juan Gabriel Aviña-Cervantes, José Luis López-Ramírez, José Guadalupe Beltrán-Hernández

Uso de n-gramas sintácticos en la atribución de traductor 1023
Christian Caballero, Hiram Calvo, Ildar Batyrshin

Detección de ciberataques a través del análisis de mensajes de redes sociales: revisión del estado del arte 1031
Jorge Enrique Coyac-Torres, Grigori Sidorov, Eleazar Aguirre-Anaya

Detección de aleta caudal de ballena jorobada por medio de mapas sobresalientes basado en mediana 1043
Rosa I. Ramos-Arredondo, Blanca E. Carvajal-Gámez, Francisco J. Gallegos-Funes, Miguel Ángel Castillo-Martinez

Generación de un corpus en español con expresiones agresivas 1055
César Jesús Núñez-Prado, Liliana Chanona-Hernández, Grigori Sidorov

Evaluación de cuatro clasificadores para el reconocimiento de síndrome de muerte súbita del lactante utilizando ácidos grasos de cadena corta como fuente de información 1061
Karen E. Villagrana-Bañuelos, Laura A. Zanella-Calzada, Carlos E. Galván-Tejada, Hamurabi Gamboa-Rosales, Jorge I. Galván-Tejada

Detección de trayectorias anómalas en otillas de vehículo 1073
Alexis Jair López-Gutiérrez, Perfecto Malaquias Quintero-Flores, Ana Karen Hernandez-Vazquez

Desarrollo e implementación de un simulador para el seguimiento de objetos en redes de sensores 1089
Ana Karen Hernández-Vázquez, Alexis Jair López-Gutiérrez, Perfecto Malaquías Quintero-Flores

Revisión del estado del arte en técnicas de procesamiento de lenguaje natural para análisis de malware 1105
Horacio Rodriguez-Bazan, Grigori Sidorov, Ponciano Jorge Escamilla-Ambrosio

Enfoque semántico de políticas para gestionar la conciencia de grupo en Groupware 1117
Mario Anzures-García, Luz A. Sánchez-Gálvez

Clasificación binaria de neuropatía diabética implementando el operador PS-Merge 1133
América Velasco-Cétera, Eddy Sánchez-DelaCruz, Pilar Pozos-Parra

Obtención de parámetros de la prueba de la estrella utilizando algoritmos genéticos y cómputo de alto rendimiento 1147
Juan Miguel Pérez-García, Moisés Emmanuel Ramírez-Guzmán, Jorge González-García

Modelo híbrido fonético-neural para corrección en sistemas de reconocimiento del habla 1163
Rafael Viana-Cámara, Mario Campos-Soberanis, Diego Campos-Sobrino

Redes neuronales evolutivas con modelos de Lotka-Volterra 1179
Fernando Javier Aguilar Canto

Simulación y evaluación del comportamiento de peregrinos guadalupanos en un evento de sismo para planificar rutas de evacuación 1195
José Abraham Montelongo Campos, Héctor Rafael Orozco Aguirre, Maricela Quintana López

Recocido simulado y el algoritmo de selección clonal aplicados al problema del agente viajero..... 1211
Nayeli Joaquinita Meléndez Acosta

Toward a Natural-Language Interface in Ontologies for Disambiguation..... 1227
Alejandro Solís-Sánchez, Pablo Barraza Cornejo, Juan Acosta-Guadarrama, Juan Ferret

Diseño industrial e inteligencia artificial: Proyección del diseño industrial a 100 años de su creación como disciplina y sus aplicaciones con IA hoy 1243
Sandra Rodríguez-Mondragón, Oscar Herrera-Alcántara, Luis Jorge Soto-Walls, Manuel Martín Clavé-Almeida

ISSN 1870-4069

Skill de Alexa para la enseñanza de la programación estructurada

Luis Japeth Trujillo-Mendieta, Oscar Herrera-Alcántara

Universidad Autónoma Metropolitana,
División de Ciencias Básicas e Ingeniería, Departamento de Sistemas,
México

{al2133004432, oha}@azc.uam.mx

Resumen. Se presenta una skill para Alexa que es la asistente virtual desarrollada por Amazon. La skill es utilizada como una herramienta de enseñanza de la programación estructurada teniendo como idea principal desarrollar un tutor virtual que resuelva dudas en el aprendizaje de este paradigma de programación. Para evaluar la funcionalidad y el nivel de satisfacción de usuarios de la skill, se realizó una encuesta a alumnos voluntarios quienes probaron y calificaron la skill.

Palabras clave: Amazon Alexa, skill, tutor virtual.

Alexa's Skill for Teaching Structured Programming

Abstract. A skill for Alexa, the virtual assistant developed by Amazon, is presented as a tool for teaching structured programming. The main idea is to develop a virtual tutor that solves doubts about the structured programming paradigm. To evaluate the functionality and the user's satisfaction, a survey was carried out with volunteers who tested and rated the skill.

Keywords: Amazon Alexa, skill, virtual tutor.

1. Introducción

En la actualidad existen variedad de asistentes virtuales, los cuales nos ayudan a realizar un sinfín de tareas cotidianas y los encontramos en variedad de dispositivos tanto en el hogar como en dispositivos de uso diario como son teléfonos inteligentes, relojes inteligentes, bocinas inteligentes, entre otros.

Alexa [1] es el asistente virtual desarrollado por Amazon, que es diferente a otros como Siri de Apple [2] y Google Assistant [3], porque provee una API

abierta de desarrollo de skills que son aplicaciones que le agregan funcionalidades para crear una experiencia más personalizada con el usuario.

Algunas categorías de skills de Alexa son [4]: Alimentos y bebidas, noticias, música y audio, estilo de vida, viaje y transporte, casa inteligente, entre otras. En efecto, existen muchas y variadas skills de Alexa. Este trabajo propone incursionar en una categoría de tutores inteligentes o ambientes virtuales de aprendizaje [5] mediante una skill para Alexa.

El resto del artículo está organizado como sigue, en el Capítulo 2 se mencionan algunos trabajos relacionados. En el Capítulo 3, se aborda el marco teórico que sustenta el desarrollo de una skill de Alexa, en el Capítulo 4 se describe el desarrollo de los módulos de la skill, en el Capítulo 5 se presentan resultados de una encuesta a un grupo de alumnos voluntarios y, finalmente, en el Capítulo 6 se presentan conclusiones y trabajos futuros.

2. Trabajos relacionados

Existe evidencia de un gran número de sistemas tutores inteligentes. Para este trabajo se considera suficiente citar sólo algunos enfocados en la enseñanza de lenguajes de programación.

RAPITS, por ejemplo, es un tutor que facilita el aprendizaje del lenguaje Pascal, al proveer más de una estrategia de enseñanza que se adapte de acuerdo con el histórico del estudiante [6].

INTELLITUTOR se describe en [7] como un sistema tutor que toma de entrada el código fuente del estudiante y lista la posición de los bugs en el programa, además de la descripción de cada error al compilar, el tipo de error, así como sugerencias de cómo rectificarlos.

En [8] se presenta CPP-Tutor que es un sistema tutor inteligente para enseñar C++. CPP-Tutor almacena problemas con su solución que involucra definición de variables, uso de operadores, sentencias de asignación y estructuras cíclicas for y while. CPP-Tutor propone un problema, espera la respuesta del usuario en forma de código fuente y luego da retroalimentación mostrando, mediante un sitio web, un código de solución y ayuda sobre algunos casos típicos de error.

Annete, un asistente de tutoría inmerso en el IDE Eclipse es descrito por Day, Rao y González en [9]. Básicamente, mencionan que al estar programando en Eclipse, Annete colecta piezas de datos del código de los estudiantes, mensajes de error del compilador y evalúa la complejidad ciclomática. Luego, Annete usa esos datos para dar retroalimentación sustentada por un modelo de red neuronal.

Existen más trabajos relacionados con tutores inteligentes enfocados a la enseñanza de lenguajes de programación, pero abordarlos en forma amplia quedaría fuera del alcance de este trabajo. Basta mencionar otros tres ejemplos con Lisp Tutor [10] que provee asistencia al estudiante al programar en Lisp, y SIPLeS [11] y MoleHill [12] para Smalltalk.

Cabe mencionar también el enfoque basado en emociones en donde, por ejemplo, Barrón et al. [13] presentan un sistema tutor cognitivo para aprender el lenguaje de programación Java.

El objetivo es disminuir la frustración y mejorar el aprendizaje del estudiante mediante respuestas concretas a sus dudas, y en [14] se describe EmoRemSys, un sistema de recomendación de recursos educativos basado en detección de emociones.

Un acercamiento muy próximo a este trabajo es el elaborado por Yijun Yu [15] en donde se propone una pedagogía novedosa llamada "Asistente de Programación", el cual tiene como objetivo el uso de un tutor personal que responda a comandos de voz del usuario y conteste con conceptos de programación. Además, se apoyaron en la inserción del programa en una Skill de Alexa para la generación de los diálogos.

Excluyendo al trabajo de Yijun Yu, ninguno de los sistemas tutores revisados alude al uso de la asistente virtual Alexa de Amazon, que ha comenzado a ganar popularidad al aceptar comandos de voz, la cual es procesada para brindar una respuesta interactiva con el usuario.

Algunas frases comunes para Alexa son "Alexa, cuéntame un chiste" "Alexa, dime la hora" "Alexa, programa un temporizador de un minuto" y "Alexa, dime un dato curioso". Una característica relevante de Alexa es que es posible parafrasear las instrucciones, y dejar que el motor de lenguaje natural de Amazon determine cuál es la mejor respuesta para ello. Esto es atractivo para atender a los estudiantes, ya que es sabido que no hay una única forma en que un alumno pueda hacer la misma pregunta. Para los propósitos de la skill de Alexa, el tema de estudio es la programación estructurada ya que es un tema que prevalece en los planes de estudio de varias carreras de ingeniería, y durante años se ha detectado la necesidad de aclarar dudas a estudiantes que inician en la programación con este paradigma, y más aún con los desafíos que representa el uso del lenguaje C.

Alexa está disponible como App de dispositivos móviles en donde se puede usar el micrófono, bocina e incluso la pantalla para ofrecer una interfaz de usuario con opciones de manejo de casas inteligentes, reproducción de música, reporte del clima, noticias, y comunicación por texto y voz con otros usuarios que cuenten con Alexa mediante su sistema "Drop in". Alexa también está disponible como bocina inteligente, que en sus versiones más completas cuenta con varios micrófonos que mejoran considerablemente la calidad de detección de la voz en ambientes ruidosos, con eco y a distancia de varios metros.

3. Marco teórico

Las aplicaciones tipo skill de Alexa son una opción para el desarrollo de una interacción personalizada e interactiva, como se mencionó anteriormente. Además, Amazon Alexa ofrece una API de programación de JavaScript o Python para desarrollar skills, lo cual marca una diferencia con otras bocinas inteligentes.

Para crear una skill de Alexa, puede hacerse a través de la consola de desarrollo de Amazon (Alexa Developer Console) y se puede conectar a un servicio alojado en AWS Lambda. Es importante mencionar que el entorno de desarrollo de la consola de Amazon se ajusta a la preferencia del programador, esto es, si el programador prefiere programar en Python, todos los recursos que

se crean al generar una nueva skill serán en el lenguaje de Pyhton, y lo mismo sucede para JavaScript. Para este proyecto se utilizó JavaScript.

Automáticamente, al crear una nueva skill, la consola genera tres archivos: index.js, util.js y package.json. El primer archivo cuenta con los manejadores de los intentos, los interceptores y el flujo principal de la skill, es por ello que aquí van las instrucciones que realizará la skill. El segundo archivo obtiene las dependencias y la exporta a la skill, es por eso que no es recomendable modificar este archivo, ya que se genera automáticamente al crear la skill con los parámetros dados. El tercero es un archivo formato JSON que contiene la información de la skill como es el nombre, versión, descripción, autor, etc. Este archivo se utiliza para la presentación de la skill ante el usuario e indica cual es el archivo principal a ejecutar (por defecto, index.js).

4. Desarrollo del proyecto

El desarrollo de proyecto se realizó con cuatro módulos, como se puede ver en la Figura 1, y los cuales se describen en seguida.

Fig. 1. Módulos del proyecto.

4.1. Módulo de enseñanza

En este módulo se recopiló información relevante del paradigma de programación estructurada y de lenguaje C. Para ello, se realizaron encuestas a varios alumnos voluntarios que estaban cursando programación estructurada, preguntando cuáles eran las principales dudas del momento o a lo largo del curso. Se obtuvieron las dudas más comunes entre los alumnos y con ello se facilitó la búsqueda de información que atendiera esas dudas en forma de preguntas y respuestas.

Dicha información (tanto preguntas como respuestas) se almacenó en una hoja de cálculo. Cabe mencionar que, si la pregunta pudiera ser explicada de mejor manera con un fragmento de código en C, este se agregaba a la hoja de cálculo, aprovechando que Alexa ofrece la opción de interactuar con el usuario a través de su teléfono celular o correo electrónico.

4.2. Módulo de memoria

Para este módulo se utilizó la consola de desarrollo de Amazon [16], la cual está disponible gratuitamente. La composición del desarrollo de la skill se puede dividir en dos partes:

–La configuración de los intentos que puede realizar el usuario a la skill,
–La configuración de las respuestas que se darán a partir de los intentos.

Es importante mencionar que al realizar una skill, por cada pregunta o intento que realice un usuario debe haber una relación uno a uno con la respuesta asociada, esto condiciona a que si se quiere que la skill responda 100 preguntas diferentes debería haber configurado 100 intentos independientes, con sus respectivas 100 respuestas. Si se trabajara de esta manera sería ineficiente. Es por eso que Amazon cuenta con ranuras o slots [17] que es una manera de representar frases o palabras como variables, de esta forma, si la pregunta o el intento están relacionadas con objetos de un mismo tipo, los objetos se guardarán en un slot y en lugar de realizar varias preguntas con cada uno de los objetos, se realiza una. Por ejemplo, supongamos que en la skill queremos preguntar qué son los tipos de datos en C, las preguntas quedarían de la siguiente manera:

"¿Qué es un entero?", "¿Qué es un flotante?", etc...

se puede sustituir por:

"¿Qué es un {tipo_variable}?"donde: tipo_variable = {entero, flotante, etc.}.

Los slots tienen identificadores únicos que permiten relacionarlos con una funcionalidad o respuesta a la que se denomina intento.

4.3. Configuración de intentos

Para este módulo primero se hizo un análisis de todas las preguntas obtenidas en la hoja de cálculo para poder agruparlas, y con ello obtener las preguntas con una misma forma y crear slots con frases o palabras que fueran a cambiar a fin de minimizar la creación de intentos. En total se crearon veinticinco intentos con la información seleccionada.

4.4. Configuración de las respuestas

Para poder manejar lo que la skill debe realizar al momento de empatar una frase o instrucción de parte del usuario, se debe crear un manejador por cada intento que indique las instrucciones a seguir después de que se reconozca la indicación del usuario. Para que el código fuera legible y de mayor entendimiento se hicieron tres archivos: lenguageStrings.js, respuestas.js y respuestas1.js.

Cada uno de estos archivos contienen las respuestas a las preguntas de la hoja de cálculo con el mismo identificador que se colocó en los slots de los intentos. También cuentan con las URL de las imágenes del código fuente si así lo ameritan. Las imágenes de los códigos fuente se pueden almacenar en el servicio de almacenamiento S3 de AWS [18].

Lo siguiente fue obtener el identificador del slot, compararlo con los identificadores de las respuestas y si eran iguales mandar la información por voz e imagen (según fuera el caso). Un punto importante es el envío de las imágenes a los dispositivos, lo cual se logra con el Amazon Presentation Language (APL) [19], ya que regresa directivas que le indican al dispositivo cómo mostrar el contenido descrito en un documento APL que consiste en una estructura JSON con los parámetros e imágenes que se van a mostrar. Al igual que la respuesta por voz, se crean los archivos: apl_Inicio.json, apl_Codigo.json y apl_Ayuda.json. Los cuales, a su vez, obtienen la imagen del código a partir de la búsqueda del identificador de la respuesta, en forma análoga a la respuesta por voz.

4.5. Módulo de respuesta

En este módulo se revisaron problemas de gramática y de respuesta de la skill, así como la incorporación de sinónimos. También se ajustaron las imágenes de los códigos para que fuesen visibles en la mayoría de los dispositivos que cuenten con APL.

Una vez finalizado los ajustes de la skill, el siguiente paso fue publicarlo para que estuviera disponible para realizar el siguiente módulo. Para esto, Amazon requiere cumplir con dos apartados: la distribución y la certificación [20]. En el primer apartado, se ingresan los datos con los que la Skill se verá en la tienda de Amazon. La Certificación es un proceso en donde la skill es revisada por Amazon que comprobará que cumpla con todos sus estándares de calidad, incluidos: respuestas relevantes, sin faltas de ortografía, cierre de sesiones, etc.

4.6. Módulo de evaluación

Por último, se realizó una encuesta [21] a diversos estudiantes voluntarios que probaron la skill. Los estudiantes evaluaron la respuesta de la skill, el entendimiento de la skill a las preguntas y su nivel de satisfacción. Adicionalmente, en la encuesta se les preguntó acerca de posibles preguntas faltantes o respuestas que fueran poco claras.

5. Resultados

Al término del desarrollo, la skill “PT Programación Estructurada” fue certificada y publicada en la tienda virtual de Amazon. Se anexan imágenes con la ejecución de la skill en un dispositivo que soporta APL. En la Figura 2 se muestra la pantalla de bienvenida a la skill en una computadora.

Fig. 2. Pantalla al iniciar la skill.

Fig. 3. Ejecución de la skill a un intento sin código.

La Figura 3 ilustra una pregunta que no involucra imágenes en su respuesta. La Figura 4 ilustra una respuesta con una imagen de código fuente para contestar sobre la sintáxis de la instrucción while en lenguaje C.

Las encuestas realizadas dieron los resultados ilustrados en las Figuras 5, 6 y 7. La Figura 5 corresponde a la pregunta "¿Cómo calificarías tu experiencia con la skill?" en donde el 73,3 % afirmó tener una respuesta excelente.

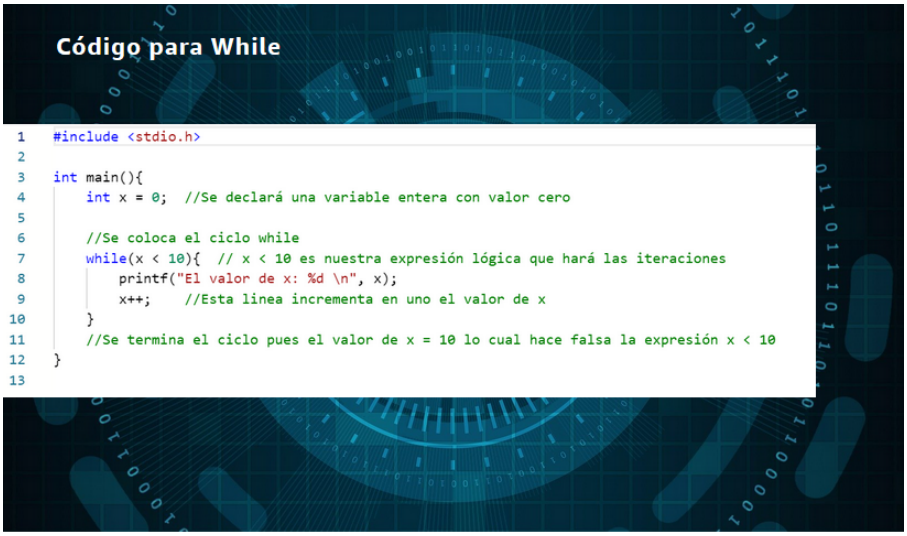


Fig. 4. Ejecución de la Skill cuando el intento contiene código.

Fig. 5. Experiencia con la Skill.

La Figura 6 corresponde la pregunta "¿El entendimiento de las preguntas y solicitudes realizadas a la skill fueron?" en donde el 66,7 % de los participantes evaluaron como excelente, y el restante 33,3 % consideró como bueno.

La Figura 7 corresponde a la pregunta "¿Cómo calificarías la respuesta que te brindó la skill?" en donde el 73,3 % las calificó como excelentes, y 26,7 % como buenas.

Los resultados de la satisfacción de los usuarios, teniendo en cuenta que 1 es totalmente en desacuerdo y 5 totalmente de acuerdo, se ilustran en las Figuras 8 y 9. Las preguntas fueron "¿Usarías esta skill como apoyo a la UEA (unidad de enseñanza aprendizaje) programación estructurada?".

Fig. 6. Entendimiento de las preguntas.

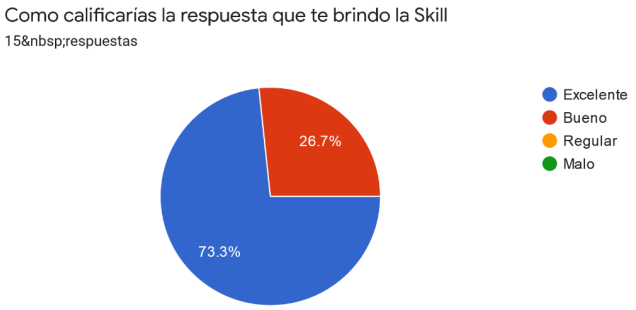


Fig. 7. Calificación de las respuestas.

La mayoría de las respuestas (13 de 15) fueron de la opción "totalmente de acuerdo". Y en la Figura 9 se observa prácticamente el mismo comportamiento para la pregunta "¿Recomendarías esta skill con alguno de tus compañeros?" en donde la respuesta principal fue "totalmente de acuerdo" con un 86,7 %.

La última pregunta de la encuesta fue "¿Qué calificación le darías a esta skill?" con un resultado de 86,7 % para el máximo nivel equivalente a 5, según se ilustra en la Figura 10.

Como se observa las calificaciones fueron positivas, y más aún hay gran posibilidad de que la skill sea recomendada a otros usuarios. Cabe mencionar que los encuestados no identificaron preguntas confusas o tediosas.

En cuanto a qué preguntas convendría agregar a la skill, la recomendaciones fueron: ¿Qué es la programación? y ¿Qué es el lenguaje C?.

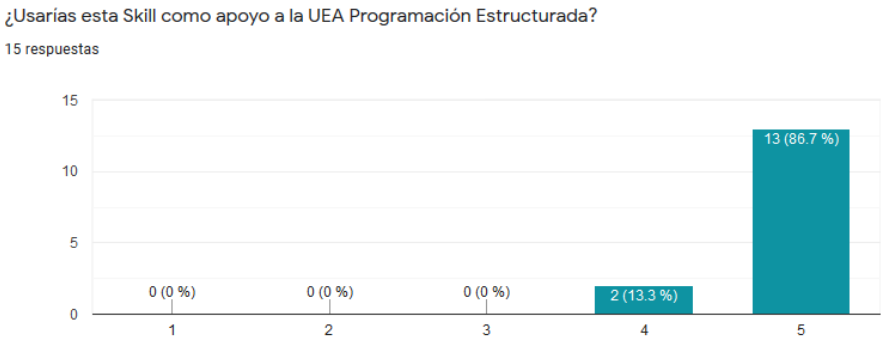


Fig. 8. Uso de la skill.

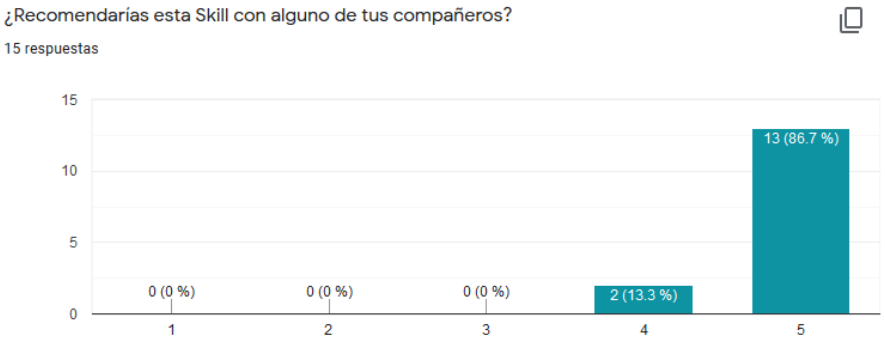


Fig. 9. Posibilidad de recomendación de la skill.

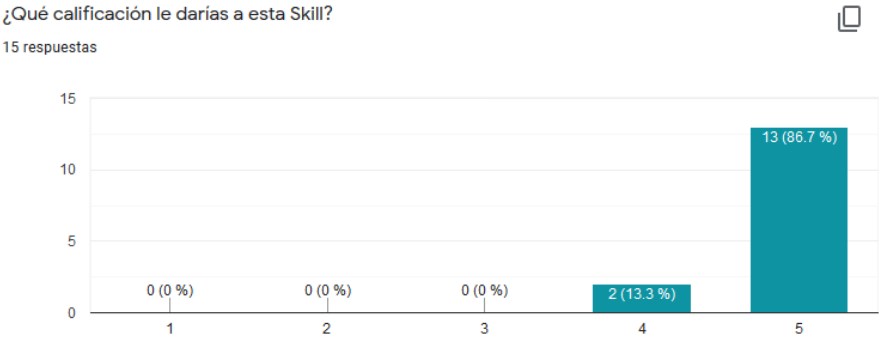


Fig. 10. Calificación de la Skill.

6. Conclusiones y trabajo a futuro

La Skill "PT Programación Estructurada" fue desarrollada satisfactoriamente siguiendo las guías de desarrollo, y prueba de ello es que sí pasó la certificación

de Amazon. En cuanto a la funcionalidad, si nos referimos a la encuesta, podemos observar que puede llegar a ser una gran herramienta tipo tutor inteligente como apoyo interactivo con tecnologías modernas para apoyar la enseñanza de cursos de programación estructurada.

Cabe mencionar que hubo algunos casos en que los estudiantes no sabían, en un inicio, activar la Skill o usar Alexa en sus teléfonos celulares principalmente, pero la motivación por usar estas tecnologías de fácil alcance entre los estudiantes hizo que adquirieran confianza en su uso y terminaron descubriendo las características de Alexa.

No podemos decir que esta versión de Skill sea final, pero marca un precedente en la incursión de Alexa en los sistemas tutores inteligentes para apoyar el aprendizaje de lenguajes de programación, ya que como observamos en las encuestas, seguirán surgiendo nuevas dudas y preguntas para enriquecer la skill. Puesto que ya no es necesario desarrollar la skill desde cero, basta generar una nueva versión con más información y mejorarla como trabajo futuro.

La conclusión principal, es que las skills de Alexa puedan ser un parteaguas como herramienta en entornos virtuales de aprendizaje donde funjan como tutores de apoyo al aprendizaje, resolviendo dudas de una manera interactiva, para pasar de la era de la información a la era de la sabiduría.

Referencias

1. Amazon: ¿Qué es Alexa?. https://developer.amazon.com/es-ES/alexa (2020)
2. Apple: Siri. https://www.apple.com/mx/siri/ (2020)
3. Google: google assistant. https://assistant.google.com/ (2020)
4. Amazon: Amazon Alexa skills store. https://www.amazon.com.mx/b?node=17553254011 (2020)
5. Sánchez M., Bonett, C. M., Moreno, R. C., Venegas, M. A.: Ambientes virtuales de aprendizaje, como apoyo de la educación presencial. Reencuentro, (72)1, pp. 55–70 (2016)
6. Woods, P. J., Warren, J. R.: Rapid prototyping of an intelligente tutorial system. In Proceedings of (ASCILITE), University of Melbourne, pp. 557–563 (1995)
7. Ueno, H.: A generalized knowledge-based approach to comprehend Pascal and C Programs. In IEICE Transactions on Information Systems, E81-D(12), IOS Press, pp. 11323–11329 (2000)
8. Abu, S.S.: Developing an intelligent tutoring system for students learning to program in C++, information technology journal, 7(7), pp. 1055–1060 (2008)
9. Dary, M., Rao, M., Gonzalez, J.: Annete: An intelligent tutoring companion embedded into the eclipse IDE, First International Conference on Cognitive Machine Intelligence (2019)
10. Anderson, J. R., Skwarecki, E.: The automated tutoring of introductory computer programming, communications of the ACM, 29(1), pp. 842–849 (1996)
11. Chee, Y. S., Xu, S. SIPLeS: Supporting intermediate Smalltalk programming through goal-based learning scenarios. In Proceedings of AI-ED 97: 8th World Conference on Artificial Intelligence in Education, Kobe, IOS Press, pp. 95–102 (1997)

12. Alphert S. R., Singley, M. K., Carroll, J. M.: Multiple multimodal mentors, delivering computer-based instruction via specialized anthropomorphic advisors. In Behavior and Information Technology, 4(2), pp. 69–79, Taylor and Francis (1995)
13. Barrón, M., Zatarain, R., González, F., Oramas, R., Reyes, C. A.: An affective and cognitive tutoring system for learning programming, pp. 171–182 (2015)
14. López, M. B., Hernández, A. J., Ramírez, R. V., Hernández, G. A., Cabada, R. Z., Barrón, M.: EmoRemSys: Sistema de recomendación de recursos educativos basado en detección de emociones/EmoRemSys: An educational recommender system by using emotions detection. Revista Ibérica De Sistemas e Tecnologias De Informação, (17)80 (2016)
15. Yu, Y., Wang, X., Rauf, I.: Teaching the art of computer programming at a distance by generating dialogues using deep neural networks (2019)
16. Amazon: Alexa skill kit developer console. https://developer.amazon.com/ alexa/console/ask (2020)
17. Amazon: Create intents, utterances, and slots, Alexa skill kit. https://developer.amazon.com/es-ES/docs/ alexa/custom-skills/create-intents-utterances-and-slots.html (2020)
18. Amazon: AWS Almacenamiento de datos seguro en la nube (S3). https://aws.amazon.com/es/s3/ (2020)
19. Amazon: Understand Alexa presentation language (APL). https://developer.amazon.com/es-MX/docs/alexa/alexa-presentation-language/understand-apl.html (2020)
20. Amazom: Test and submit your skill for certification. https://developer.amazon.com/es-MX/docs/alexa/devconsole/test-and-submit-your-skill.html (2020)
21. Encuesta para evaluar la Skill PT programación estructurada. https://forms.gle/7XqzFqpyw44CGrYr9 (2020)

ISSN 1870-4069

Aplicación de una red neuronal probabilística para predecir el desempeño académico de estudiantes de educación superior en línea

Rosa Leonor Ulloa Cazarez

Universidad de Guadalajara,
Sistema de Universidad Virtual, Guadalajara,
México

rosa.ulloa@udgvirtual.udg.mx

Resumen. La educación superior en línea (ESL) es una modalidad en crecimiento que presenta altos índices de reprobación. La predicción del desempeño del estudiante (DE) se ha utilizado para identificar oportunamente a los estudiantes en riesgo de reprobación, por lo que se propone la aplicación de una red neuronal probabilística (RNP) para clasificar el DE en cursos de ESL. Se utilizaron las calificaciones de los estudiantes de cuatro materias de ESL, que se prepararon para realizar dos experimentos, cada uno utilizando cuatro conjuntos de datos. La precisión de la RNP es comparada con la del Análisis Discriminante (AD). Los resultados mostraron mejor porcentaje de clasificación de la RNP que el AD en cinco de los ocho conjuntos de datos, mientras que en los otros tres resultaron iguales en sus porcentajes. Con base en ello, podemos concluir que una RNP es útil para clasificar el desempeño de un estudiante de ESL.

Palabras clave: Red neuronal probabilística, análisis discriminante, educación superior en línea, predicción del desempeño.

Probabilistic Neural Network to Predict the Academic Performance of Online Higher Education Students

Abstract. Enrolments of Online Higher Education (OHE) keep growing, but also its failure rates. The prediction of student academic performance (SP) is used to timely detect students at risk of failure. Thus, I propose to use a probabilistic neural network (PNN) to classify SP in terms of failure or success as a more understandable form to present prediction results. I use the scores of four subjects of an OHE program to execute two experiments, each one using the four sets of scores. I compared the accuracy of the PNN against the Discriminant Analysis (DA). The results show a better percentage of classification for the PNN than the DA in five of the eight datasets, while the rest of the three datasets obtained the same percentage for both models. On this basis, we can conclude that the PNN is useful to classify the academic performance of OHE students.

Keywords: Probabilistic neural network, discriminant analysis, online higher education, academic performance.

1. Introducción

La Educación Superior en línea es la modalidad instruccional con mayor crecimiento en el mundo [1] gracias a sus características de flexibilidad [2, 3] y a sus capacidades de adaptación y evolución. Alcanza estudiantes y profesores fuera de los contextos geográficos locales de las instituciones [4] y junto a este crecimiento de la matrícula, han surgido una serie de problemáticas relacionadas con el desempeño de los programas en línea en términos de la eficiencia terminal, la deserción y la reprobación de estudiantes [5-7].

En este contexto, la analítica educativa ha tomado prevalencia [8], en particular el uso de algoritmos de clasificación para predecir el desempeño del estudiante [9].

Algunos autores, con este propósito, exploran la aplicación de acciones gerenciales tales como la predicción temprana de estudiantes en riesgo de reprobación [10, 11] o de deserción [3]. También, se identifican trabajos evaluando el desarrollo de avanzadas herramientas tecnológicas para el monitoreo y acompañamiento de los estudiantes [12].

Se afirma que la aplicación de algoritmos de predicción en términos de clasificación guiados por una adecuada fundamentación teórica pedagógica y psicológica, proporcionan más elementos para la toma de decisiones por su facilidad de interpretación, que algoritmos con otro tipo de salidas, por ejemplo, una predicción numérica [13].

Sin embargo, se ha encontrado que la calidad de los datos utilizados es una de las deficiencias más preponderantes en la literatura sobre el tema [8, 14].

En trabajos previos, con el propósito de identificar tempranamente los estudiantes de ESL en riesgo [15, 16], se ha explorado el uso de algoritmos predictivos con salidas numéricas. Sin embargo, para continuar fortaleciendo la generación de conocimiento en el área de la analítica educativa, en este trabajo se propone un modelo predictivo en términos de clasificación, un método más comprensible para los profesionistas que no son del área computacional y que puede utilizarse de una forma más directa para sustentar la toma de decisiones y el despliegue de estrategias de retención y mejoramiento del desempeño de estudiantes [9].

En este trabajo utilicé datos de las calificaciones de los estudiantes en cuatro materias del primer semestre de un programa de Educación Superior en línea para predecir, clasificando, el desempeño de dichos estudiantes en cuanto a la aprobación o reprobación del curso.

El principal objetivo en este trabajo es explorar las ventajas de utilizar una Red Neuronal Probabilística (RNP) para predecir el desempeño de los estudiantes de Educación Superior en Línea, en términos de su precisión, facilidad de uso y de interpretación, y compararlo frente a un modelo estadístico común utilizado, el Análisis Discriminante (AD).

Se realizaron dos experimentos, uno utilizando el total de registros del conjunto de datos y otro con un subconjunto de estos datos. Para ello, se generó una RNP por cada conjunto de datos y se aplicaron en la predicción con un enfoque de clasificación, en lugar de predecir un valor numérico.

La precisión de la RNP (medida en porcentaje de estudiantes correctamente clasificados), es comparada con la del modelo estadístico, el AD.

Se formularon las siguientes hipótesis nula y alternativa a ser comprobadas:

H_o: El porcentaje del clasificador de una RNP es igual al del AD,
H_1: El porcentaje del clasificador de una RNP es diferente al del AD.

El RNP y el AD se generaron teniendo como variable predictora (independiente), a la calificación obtenida por los estudiantes en la primera unidad (U1) de cuatro cursos; la variable dependiente categórica, corresponde a una de las dos siguientes: "aprueba" y "no aprueba" en función de la calificación final de los mismos cursos, considerando que una calificación menor a 60 es igual a reprobar la materia.

El documento se desarrolla en las siguientes 4 secciones. La siguiente sección presenta la revisión de literatura sobre la predicción del desempeño del estudiante y sobre algoritmos de clasificación. La sección 3, describe la metodología para generar los modelos, la descripción de los datos utilizados, así como la teoría que sustenta a los dos modelos propuestos para la comparación, la RNP y el AD. La sección 4 detalla los resultados obtenidos y en la sección 5 se presenta la discusión y la identificación del trabajo futuro.

2. Trabajo relacionado

En la literatura se encuentra suficiente trabajo en cuanto a la predicción del desempeño del estudiante, implementando técnicas estadísticas y computacionales, con diferentes resultados y utilizando una variedad amplia de variables, que son relacionadas con una calificación final [9]. Esta variedad de resultados anima a continuar con esta línea de investigación, contribuyendo a la generación de conocimiento en las dos áreas, la de las ciencias computacionales para la predicción y la de la Educación Superior en línea.

Se identifican tres tipos de predicciones, la clasificación, la regresión y la estimación de la densidad [14]. Cada una aporta a la analítica educativa con una perspectiva rica para ser explorada, siendo la predicción con clasificadores, el método más comúnmente utilizado en educación [5]. También se encuentran trabajos que identifican estilos de aprendizaje y entrenan sistemas inteligentes de tutorías y para la evaluación del desempeño [13], así como la creación de modelos para predecir la deserción o retención de estudiantes [17].

En algunos artículos se analizan diversos factores relacionados con el contexto del estudiante y los vinculan con su desempeño académico [18]. Otros estudios analizan factores psicológicos, de la personalidad, y relacionados con prácticas de estudio y aprendizaje [19]. Y finalmente, hay estudios que utilizan los datos de las plataformas de gestión del aprendizaje [18, 20] como datos de entrada de los modelos.

Los algoritmos de clasificación más reportados en la literatura son C4.5, EM, Naïve Bayes, Support Vector Machines [18], algunas redes neuronales [21] y varios métodos basados en reglas y árboles de decisión [9]. No se encontró la aplicación de RNP para la predicción del DE lo cual justifica la elección de este método en el contexto específico de este trabajo.

Tabla 1. Descripción de los conjuntos de datos.

Materia	Conjunto	Matrícula total	Aprobados	Reprobados	Deserción
Pensamiento lógico-matemático (DPLM)	1	180	51	99	30
Fundamentos de Computación (CFC)	2	169	42	69	60
Planeación de proyectos (DPP)	3	165	45	56	64
Diagnóstico situacional (DS)	4	179	61	65	51
Totales		693	199	289	205

La predicción del desempeño del estudiante ha resultado en beneficios en cuanto a la prevención de la deserción y la emisión de señales de alerta de reprobación y en lo general, el uso de la analítica educativa ha probado ser de utilidad para fortalecer la toma de decisiones [8].

Sin embargo, resaltan dos características mencionadas en la literatura, mismas que se abordan en este trabajo como una contribución del mismo: la ya mencionada falta de claridad y calidad en el manejo, recolección y validación de los datos [8, 14, 18] y, la generación de un modelo predictivo que sea manejable tanto por los administradores educativos, como por los profesores y mentores [13, 19].

3. Metodología

3.1. Descripción de los datos

En referencia a uno de los propósitos en este trabajo sobre facilitar el uso y la aplicación de este modelo para los profesores y administradores educativos, se optó por trabajar con las calificaciones de los estudiantes que se pueden obtener mediante las herramientas de reporte que tienen prácticamente todas las plataformas de gestión del aprendizaje. En el caso que se presenta, se usaron los reportes de dos plataformas distintas, Moodle, y una plataforma de desarrollo propio de la institución de Educación Superior que oferta el programa de análisis, denominada Metacampus [22].

Se descargaron las calificaciones obtenidas en el total de materias cursadas (cuatro) por los estudiantes durante el primer semestre de un programa de ESL, mismas que conforman los cuatro conjuntos de datos descritos en la Tabla 1.

Trabajar con datos del primer ciclo escolar del estudiante, permite tomar decisiones en tiempo oportuno, así como el despliegue de estrategias remediales. En ambos casos de plataformas, las calificaciones finales son calculadas automáticamente, ya sea obteniendo el promedio general o la sumatoria de los valores ponderados de cada actividad en el curso. Los puntajes obtenidos por los estudiantes en las actividades, son otorgadas manualmente por los profesores de cada curso, considerando rúbricas con criterios de evaluación especificados de antemano por una academia [23]. El valor de la calificación de la unidad 1, es calculado manualmente por la autora de este trabajo, como el promedio (Pensamiento lógico matemático-DPLM, Planeación de proyectos-

DPP, y Diagnóstico situacional-DS) o la sumatoria (Fundamentos de Computación-CFC) de las calificaciones de las actividades que tuvieron lugar en el periodo de la unidad en congruencia con el sistema de evaluación establecido para cada curso.

Se realizaron dos experimentos, uno, utilizando el total de registros de la base de datos, siendo esta BD1, con 693 registros. En el segundo, se eliminaron los registros de los estudiantes que desertaron del curso, creando una segunda base de datos, denominada BD2, con un total de 488 registros.

3.2. Red neuronal probabilística

Una RNP es una red neuronal a la que se le reemplaza la función de activación sigmodea y ejecuta un método no paramétrico para clasificar observaciones con bastante confiabilidad en la clasificación y una velocidad de respuesta que puede ser superior al mecanismo de propagación inversa [24]. En nuestro caso, las observaciones corresponden a los estudiantes en línea dentro de g grupos o categorías (para nuestro estudio, g = 2, siendo "aprueba/no aprueba"). La RNP funciona con base en v variables cuantitativas, que para nuestro estudio v=1 es la variable independiente cuyo valor corresponde a la calificación obtenida por cada estudiante al término de la primera unidad del curso [25].

La RNP construye una estimación no paramétrica de cada función de densidad por cada uno de las g categorías a una posición deseada, con base en observaciones cercanas a esa categoría. La estimación es basada en una ventana de Parzen que pondera observaciones a partir de cada categoría de acuerdo a su distancia a partir de la posición especificada.

Una RNP consiste de cuatro capas, que son ilustradas en la Figura 1:

1) Capa de entrada – tiene v neuronas, donde v = 1 (una variable de entrada, que es la calificación de la primera unidad obtenida por cada estudiante);
2) Capa de patrones – con n neuronas, donde n = 180 patrones (que corresponde al número de estudiantes del curso 1)
3) Capa sumatoria – tiene g neuronas, donde g = 2 (relacionados con las dos clasificaciones: "*aprueba*" y "*no aprueba*")
4) Capa de salida – contiene una neurona binaria para cada categoría g, que se activa o desactiva si determinado estudiante es o no asignado a la categoría correspondiente

En la Fig. 1, se diagrama uno de los conjuntos de datos, a fin de ilustrar el flujo de datos a través de la RNP, en el sentido de que la variable predictora v de la capa de entrada, es estandarizada con base en la media de sus valores, divididos entre su desviación estándar. Estos valores sirven de entrada a las neuronas de la capa de patrones, que los turna a través de una función de activación usando los valores de entrada para estimar la función de densidad de probabilidad para cada una de las g categorías a una posición dada. La función de activación usada es la expresada en la Ecuación 1, si el estudiante i pertenece a la categoría j, es decir, la Ecuación 1 sería igual a cero si el estudiante i no pertenece a la categoría j (g_{ij}=0), donde:

— X corresponde al conjunto de valores de la variable de entrada, es decir, al conjunto de calificaciones de la primera unidad obtenida por cada estudiante.

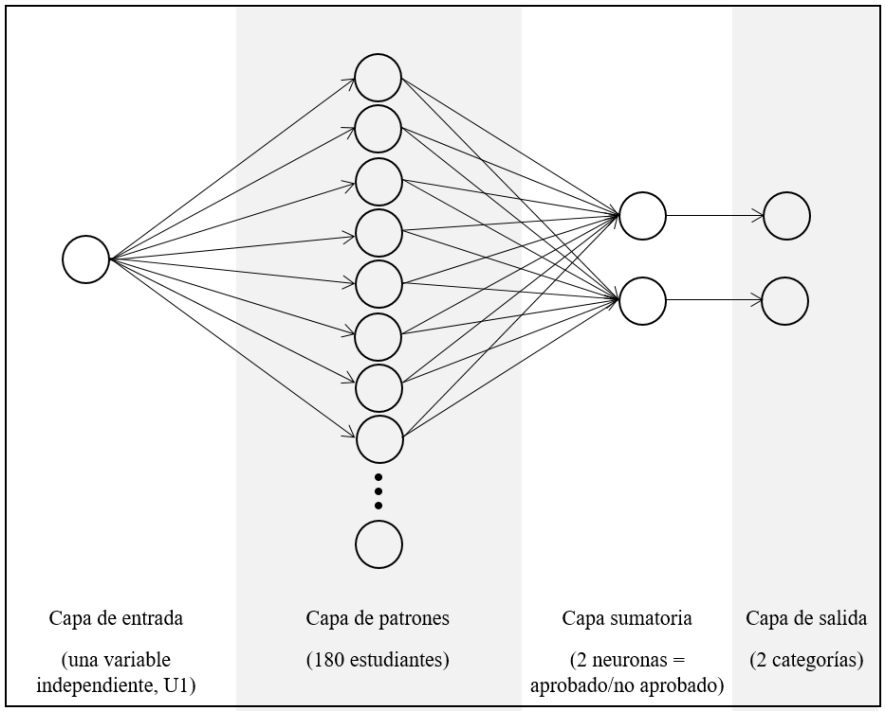


Fig. 1 Descripción del diagrama de una RNP, mostrando el primer conjunto de datos en nuestro estudio.

— x_i corresponde a la *i*-ésima calificación del *i*-ésimo estudiante.

— σ es la desviación estándar de las calificaciones del grupo:

$$g_{ij} = W\left(\frac{X - x_i}{\sigma}\right). \tag{1}$$

La función *W* corresponde a la Gausiana expresada en la Ecuación 2, donde $\|X - x_i\|^2$ es la distancia euclidiana entre *X* y x_i:

$$\exp\left(-\frac{\|X - x_i\|^2}{\sigma^2}\right). \tag{2}$$

Estas estimaciones de densidad son turnadas a la capa sumatoria, misma que combina la información recibida con los datos de los *n* estudiantes con sus correspondientes costos tanto de probabilidad como de clasificación errónea para derivar un puntaje para cada una de las *g* categorías. Estos puntajes son turnados a las neuronas binarias de la capa de salida.

3.3. Análisis discriminante

En un análisis estadístico de regresión, la variable dependiente es numérica, mientras en el AD, la variable dependiente es categórica. Esta variable en nuestro estudio adquiere dos variantes o categorías: "aprueba" o "no aprueba".

Las variables independientes son numéricas, correspondiendo a una sola en nuestro estudio, es decir, la calificación obtenida por cada estudiante al término de la primera unidad del curso, U1. Estas variables son útiles para determinar a qué categoría pertenecen los estudiantes.

El AD encuentra relaciones lineales entre las variables independientes que mejor discriminen a los estudiantes en las categorías. Enseguida, el AD construye una regla de decisión para asignar un estudiante nuevo a una de las categorías preestablecidas, es decir, una regla que responda a la pregunta "¿a qué categoría debería pertenecer cierto estudiante con determinada calificación obtenida en su primea unidad del curso?

El objetivo en un modelo estadístico de AD es construir *n* combinaciones lineales de las *v* variables independientes tal que mejor discriminen entre las *g* categorías. La *j*-ésima función discriminante tiene la forma de la Ecuación 3, donde cada valor de *Z* es estandarizada, al igual que en la RNP, con base en la media de sus valores dividida entre su desviación estándar. Las funciones discriminantes se encuentran a partir de los eigenvalores de $W^{-1}B$, donde *W* corresponde a la suma de cuadrados dentro de las categorías y la matriz de los productos cruz, mientras que B corresponde a la suma de cuadrados entre las categorías y la matriz de los productos cruz. Los coeficientes de las funciones discriminantes son obtenidos a partir de eigenvectores [26]:

$$D_j = d_{j1}Z_i + d_{j2}Z_2 + ... + d_{jv}Z_v . \tag{3}$$

Para clasificar un nuevo estudiante, las funciones de clasificación son derivadas y una puntuación se deriva para cada categoría. Una puntuación para la *j*-ésima categoría se calcula con base en la Ecuación 4; el nuevo estudiante es clasificado en una categoría teniendo el valor más grande para C_j*$previo_j$, donde $previo_j$ corresponde a la probabilidad previa de pertenecer a la *j*-ésima categoría:

$$C_j = c_{j1}X_i + c_{j2}X_2 + ... + X_{jv}Z_v + C_{j0} . \tag{4}$$

4. Resultados

Tanto la RNP, como el AD fueron entrenados y probados mediante el método de validación *leave-one-out (LOO)*, ello por tratarse de conjuntos de datos pequeños y porque el LOO es un método determinístico eliminando cualquier aleatoriedad en la selección de sus datos para ser entrenado [27].

En el caso del primer experimento, se utilizó la BD1 con un total de 693 registros. En el segundo experimento se utilizó la BD2 con un total de 488 registros. Los resultados por conjunto de datos para la RNP y el AD, se muestran en la Tabla 2.

Con base en los resultados, podemos rechazar la hipótesis nula en cinco de los ocho conjuntos de datos y aceptar la hipótesis alternativa en favor de la RNP formulada en la sección de Introducción de nuestro estudio:

Tabla 2. Resultados en la precisión de la RNP y AD por conjunto de datos, utilizando BD1 y BD2. El valor en columnas RNP y AD corresponde al porcentaje de estudiantes clasificados correctamente, el Tamaño corresponde al total de registros por materia utilizados).

Experimento	Materia	Tamaño	RNP	AD
1 – 693 registros totals	DPLM	180	**81.67**	72.22
	DPP	165	**94.55**	93.33
	DS	179	**86.03**	75.98
	CFC	169	85.80	85.80
2 – 488 registros totals	$DPLM_b$	150	**80.67**	73.33
	DPP_b	101	90.10	90.10
	DS_b	128	**81.25**	74.22
	CFC_b	109	79.92	79.92

H_1: La precisión del clasificador de una RNP es diferente a la del AD.

Asimismo, se acepta la hipótesis nula para los conjuntos CFC, DPP_b y CFC_b:

H_0: La precisión del clasificador de una RNP es igual a la del AD.

5. Conclusiones

Con base en los resultados de la Tabla 2 y las hipótesis aceptadas, podemos concluir que una RNP fue mejor o igual, (pero nunca peor) que un AD cuando se comparan en su precisión. Por tanto, una RNP puede ser aplicada para predecir con base en un enfoque de clasificación *aprueba*/*no aprueba*, el desempeño de un estudiante en programas de ESL.

Los dos modelos, RNP y AD generados, producen resultados predictivos que permiten la comprensión de la predicción y pueden ser utilizados para tomar acciones rápidas de intervención por un profesor o un mentor. Sin embargo, la rapidez de respuesta de la RNP así como su mayor precisión, la hacen una alternativa más viable que el AD.

Por otro lado, se usaron únicamente las calificaciones de los estudiantes para generar los modelos, lo que le da mayor simplicidad a la construcción del modelo siendo también más sencillo de replicar pues los datos son de relativo fácil alcance y estandarización.

Se resalta la generación de dos experimentos a partir de crear una segunda base de datos con las calificaciones de los estudiantes, eliminando a aquellos que desertaron (BD2). Sin embargo, se observó que la precisión de los dos modelos disminuyó cuando se usó BD2 respecto a los obtenidos con la BD1.

Estos resultados abonan al conocimiento en la generación y aplicación de algoritmos computacionales para predecir en términos de clasificación aprueba/no aprueba, y se habilita la replicación de este estudio bajo las condiciones estipuladas en la sección 3 de Metodología. Este tipo de predicciones en términos de clasificación, son útiles para

la toma rápida de decisiones por parte de docentes, mentores y administradores educativos, pues el algoritmo directamente interpreta la clasificación y la presenta de manera sencilla, tal que ayuda a identificar a los estudiantes en riesgo.

Para el trabajo futuro se incluye la aplicación de clasificadores basados en máquinas de soporte vectorial, cuyo desempeño será comparado con los obtenidos de la RNP y AD en este estudio.

Referencias

1. Clinefelter, D.L., Aslanian, C.B.: Online college students 2016: Comprehensive Data on Demands and Preferences. Learning House, Louiseville (2016)
2. García Aretio, L.: Historia de la educación a distancia. Revista Iberoamericana de Educación a Distancia, 2(1), pp. 8–27 (1999)
3. Lykourentzou, I., Nikolopoulos, V., Mpardis, G., Loumos, V.: Dropout prediction in e-learning courses through the combination of machine learning techniques. Computers & Education (2009)
4. Kardan, A.A., Sadeghi, H., Ghidary, S.S., Fani-Sani, M.R.: Prediction of students course slecction in online higher education institutes using neural network. Computers & Education, pp. 1–11 (2009)
5. Thakar, P., Mehta, A.: Manisha, Performance Analysis and Prediction in Educational Data Mining: A Research Travelogue. International Journal of Computer Applications, 110(15), pp. 60–68 (2015)
6. Mohammadyari, S., Singh H.: Understanding the effect of e-learning on individual performance: The role of digital literacy. Computers & Education, 82, pp. 11–25 (2015)
7. Bawa, P.: Retention in online courses. Exploring issues and solutions. A literature review. SAGE Open, 6(1), pp. 1–11 (2016)
8. Banihashem, S.K., Aliabadi, K., Ardakani, S.P., Delaver, A., Ahmadabadi, M.N.: Learning Analytics: A Systematic Literature Review. Interdisciplinary Journal of Virtual Learning in Medical Sciences, 9(2) (2018)
9. Helal, S., Li, J., Liu, L., Ebrahimie, E., Dawson, S., Murray, D.J., Long, Q.: Predicting academic performance by considering student heterogeneity. Knowledge-Based Systems, 161(1), pp. 134–146 (2018)
10. Castillo Merino, D., Serradell López, E.: An analysis of the determinants of students' performance in e-learning. Computers in human behavior, 30, pp. 476–484 (2014)
11. Romero, C., López, M.-I., Luna, J.-M., Ventura S.: Predicting students' final performance from participation in on-line discussion forums. Computers & Education, 68(1), pp. 458–472 (2013)
12. Chrysafiadi, K., Virvou, M.: PeRSIVA: An empirical evaluation method of a student model of an intelligent e-learning environment for computer programming. Computers & Education, pp. 322–333 (2013)
13. Gobert, J.D., Kim, Y.J., Sao Pedro, M.A., Kennedy, M., Betts, C.G.: Using educational data mining to assess students' skills at designing and conducting experiments within a complex systems microworld. Thinking Skills and Creativity, 18(1), pp. 81–90 (2015)
14. Avella, J.T., Kebritchi, M., Nunn, S.G.: Learning Analytics Methods, Benefits, and Challenges in Higher Education: A Systematic Literature Review. Online Learning, 20(2), pp. 13–29 (2016)
15. Ulloa-Cazarez, R.L., Lopez-Martin, C.: Neural networks for predicting student performance in online education. IEEE Latin America Transactions, 16(7), pp. 2053–2060 (2018)

16. Ulloa-Cazarez, R., Lopez-Martin, L., Abran, C.A., Yañez-Marquez, C.: Prediction of Online Students Performance by Means of Genetic Programming. Applied Artificial Intelligence, 32(9-10), pp. 858–881 (2018)
17. Mason, C., Twomey, J., Wright, D., Whitman, L.: Predicting Engineering Student Attrition Risk Using a Probabilistic Neural Network and Comparing Results with a Backpropagation Neural Network and Logistic Regression. Research in Higher Education, 59(1), pp. 382–400 (2018)
18. Gašević, D., Dawson, S., Rogers, T., Gasevic, D.: Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. Internet and Higher Education, 28, pp. 68–84 (2016)
19. Yeh, Y.-C., Kwok, O.-M., Chien, H.-Y., Sweany, N.W., Baek, E., McIntosh, W.: How College Students' Achievement Goal Orientations Predict Their Expected Online Learning Outcome: The Mediation Roles of Self-Regulated Learning Strategies and Supportive Online Learning Behaviors. Online Learning, 23(4) (2019)
20. Yang, T.-Y., Brinton, C.G., Joe-Wong, C., Chiang, M.: Behavior-Based Grade Prediction for MOOCs Via Time Series Neural Networks. IEEE Journal of Selected Topics in Signal Processing, 11(5), pp. 716–728 (2017)
21. Yan, N., Tat-Sheung Au, O.: Online learning behavior analysis based on machine learning. Asian Association of Open Universities Journal, 14(2), pp. 97–106 (2019)
22. Rodríguez Armenta, C.E.: El Metacampus virtual. Boletín Informativo CUDI (2006)
23. Sistema de Universidad Virtual, Licenciatura en Tecnologías e Información, s/d 2020. [En línea]. Available: http://www.udgvirtual.udg.mx/lti#. [Último acceso: 30 marzo 2020] (2020)
24. Specht, D.F.: Probabilistic neural networks. Neural Networks, 3(1), pp. 109–118 (1990)
25. Masters, T.: Advanced algorithms for neural networks. Nueva York: Wiley (1995)
26. McLachlan, G. J.: Discriminant Analysis and Statistical Pattern Recognition. New York: Wiley (1992)
27. López-Martín, C.: Predictive accuracy comparison between neural networks and statistical regression for development effort of software projects. Applied Soft Computing, 27(1), pp. 434–449 (2015)

ISSN 1870-4069

Clasificadores basados en reglas y selección de atributos para el diagnóstico clínico de subtipos del Síndrome de Guillain-Barré

Daniel Alarcón-Narváez[1], Jerusa Marchi[2], José Hernández-Torruco[1], Oscar Chávez-Bosquez[1], Betania Hernández-Ocaña[1]

[1] Universidad Juárez Autónoma de Tabasco, División Académica de Ciencias y Tecnologías de la Información, México
[2] Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, Brasil

{jose.hernandezt,oscar.chavez,betania.hernandez}@ujat.mx, 182H13002@alumno.ujat.mx, jerusa.marchi@ufsc.br

Resumen. Los algoritmos de aprendizaje automático se han utilizado para identificar diferentes tipos de enfermedades, a través de modelos diagnósticos o predictivos. El síndrome de Guillain-Barré (SGB) es un trastorno neurológico que generalmente afecta a personas de cualquier edad y sexo. En este trabajo, vamos a investigar un modelo de diagnóstico para los subtipos del SGB utilizando exclusivamente variables clínicas. Para ello aplicamos los clasificadores: (1) JRip, (2) OneR y (3) PART, comparándolos entre sí para encontrar el mejor de ellos. También se realizaron experimentos con los filtros: (1) CFS, (2) Chi.Squared, (3) Consistency y (4) RandomForest. Estos experimentos utilizaron un conjunto de datos reales que incluye 129 registros de pacientes mexicanos y 26 variables clínicas representativas. Se registraron las medidas de rendimiento estándar y se utilizó el *accuracy* balanceado para medir el rendimiento de los modelos y obtener el mejor clasificador en cada experimento. Encontramos con estos experimentos que OneR con filtro Chi.squared, así como OneR con filtro RandomForest, obtienen el mejor *accuracy* balanceado con un valor de 0.6564.

Palabras clave: SGB, métodos de selección de atributos, clasificación multiclase, clasificadores individuales, medidas de desempeño.

Rule Based Classifiers and Feature Selection for Clinical Diagnosis of Guillain-Barré Syndrome Subtipes

Abstract. Machine learning algorithms have been used to identify different types of diseases, through diagnostic or predictive models. The

Guillain-Barré syndrome (GBS) is a neurological disorder that generally affects people of any age and sex. In this work, we aim to investigate if it is possible to create a diagnostic model for GBS subtypes using exclusively clinical variables, since according to the reviewed literature, there is no model with these characteristics. For this, we applied the classifiers: (1) JRip, (2) OneR and (3) PART, comparing them to find the best one. We also made experiments with filters: (1) CFS, (2) Chi.Squared, (3) Consistency and (4) RandomForest. These experiments used a real data set that includes 129 mexican patient records and 26 representative clinical variables. Standard performance measures were recorded and the balanced accuracy was used to measure the performance of the models and obtain the best classifier in each experiment. We found with these experiments than OneR with Chi.squared filter, as well as OneR with RandomForest filter, they get the best balanced *accuracy* with a value of 0.6564.

Keywords: SGB, feature selection methods, multiclass classification, single classifiers, performance measures.

1. Introducción

El Síndrome de Guillain-Barré (SGB) es una enfermedad en la que el sistema inmunitario ataca al propio cuerpo por error, confundiendo la mielina con una célula extraña. Causa parálisis flácida, debilidad de músculos y extremidades [17]. Eso se debe comúnmente a la inflamación multifocal de las raíces espinales y nervios periféricos. Este síndrome difiere en términos de su apariencia, duración, simetría de manifestaciones clínicas y si dañan o no principalmente la mielina, axón o fibras nerviosas periféricas. En casos graves, las prolongaciones de las neuronas que son las responsables de enviar el impulso nervioso también se dañan [13]. La incidencia anual estimada del SGB es 0.61-2 casos por 100,000 personas y aproximadamente el 25 % de los pacientes con SGB requieren cuidados intensivos. A pesar de un adecuado tratamiento, el 3.5 % muere debido a complicaciones relacionadas con la parálisis respiratoria, ataque cardíaco o trombosis [28]. Los subtipos más comunes del SGB se dividen en cuatro: polineuropatía desmielinizante inflamatoria aguda (AIDP), neuropatía axonal motora aguda (AMAN), neuropatía axonal sensorial motora aguda (AMSAN) y síndrome de Miller-Fisher (MF) [23]. Debido a la variación en gravedad y tratamiento entre subtipos, la diferenciación entre ellos es crítica.

Existen investigaciones en la literatura, donde se realizaron experimentos para la creación de modelos para el diagnóstico del SGB. Por mencionar algunos, tenemos el trabajo de Hernández-Torruco et al. [9]. En ese proyecto se utilizaron un conjunto de datos que contenía pruebas clínicas, serológicas y de conducción nerviosa y aplicaron el árbol de decisión C4.5, SVM (*Support Vector Machines*) usando un núcleo gaussiano y kNN (*k Nearest Neighbour*); alcanzando de *accuracy* con C4.5 0.9211, con kNN 0.9179 y SVM 0.9154. Canul-Reich et al. [1] crearon diferentes modelos predictivos basados en clasificadores individuales,

utilizando un conjunto de datos con 4 variables clínicas y 12 obtenidas de una prueba de conducción nerviosa. Aplicaron clasificadores con diferentes enfoques como árboles de decisión (C4.5), basados en instancias (kNN), basados en kernel (SVM), redes neuronales como SLP (*Single-Layer Perceptron*, por sus siglas en inglés), MLP (*Multilayer feed-forward*, por sus siglas en inglés) y RBF-DDA (*Radial Basis Function-Dynamic Decay Adjustment*, por sus siglas en inglés), clasificadores basados en reglas (OneR, JRip), Regresión logística y método OvA (One vs All). El mejor *accuracy* alcanzado fue del 0.9235.

Torres-Vásquez et al. [25] utilizaron la técnica SMOTE (*Synthetic Minority Over-sampling Technique*, por sus siglas en inglés) para balancear las instancias de cada subtipo del sindrome en el dataset, con el objetivo de investigar si era posible mejorar los modelos predictivos encontrados anteriormente. Aplicaron los clasificadores C4.5, SVM, JRip y la técnica OvO (*One vs One*) para crear subproblemas binarios de los subtipos del síndrome. Estos resultados se evaluaron usando la métrica AUC (*Area Under the ROC, Receiver Operating Characteristic, Curve*, por sus siglas en inglés). El mejor resultado lo alcanzaron con JRip para los subtipos AMSAN vs MF con un valor de 0.8720 de AUC. En ese trabajo pudieron demostrar que algunos modelos con datos balanceados, superan a los modelos con datos desbalanceados.

También Torres-Vásquez et al. [24] usaron 4 métodos de submuestreo para balancear las instancias, con diferentes enfoques como: *Random Undersampling* (RUS), *Tomek Link* (TML), *One Side Selection* (OSS) y *Neighborhood Cleaning Rule* (NCR). Un método de sobremuestreo fue probado para balancear las instancias de cada subtipo del síndrome. Aplicaron los clasificadores C4.5, SVM, JRip y las técnicas de binarización OvA y OvO. De los experimentos de muestreo que realizaron, se obtuvieron los mejores resultados combinando SMOTE con OvA. Con respecto a los clasificadores, obtuvieron con JRip el mejor rendimiento con un valor mayor a 0.90 de AUC, ya que encontró más casos con diferencias estadísticamente significativas para todos los experimentos. Los resultados que obtuvieron en esa nvestigación mostraron que balancear el conjunto de datos original mejora los modelos predictivos anteriores.

Todos estos trabajos mencionados anteriormente tienen en común el dataset, donde utilizan solo 4 variables clńicas y 12 variables de pruebas de conducción nerviosa. Estas 16 variables fueron las relevantes encontradas a partir de un dataset de 156 variables, usando QSA-PAM (*Quenching Simulated Annealing-Partitions Around Medoids*, por sus siglas en inglés) [2]. Por lo que el 75 % de estas variables requieren de la visita a un laboratorio para realizarte las pruebas, lo que implica pérdida de tiempo. Tomando en cuenta que las primeras semanas de este síndrome son cruciales para el bienestar del paciente en este trabajo decidimos utilizar solo variables clínicas. Es decir variables que se basan en la experiencia del especialista y lo que siente el paciente al momento de ir a consulta. Reduciendo así los tiempos para la detección de alguno de los subtipos del síndrome y poder dar el tratamiento adecuado lo antes posible.

En este estudio utilizamos un conjunto reducido de 25 atributos clínicos, seleccionados de un conjunto de datos original de 365 atributos.

Estos atributos seleccionados corresponden a variables clínicas y fueron tomadas de 129 pacientes mexicanos que contienen los cuatro subtipos de SGB mencionados anteriormente. Para los experimentos, aplicamos clasificadores individuales a través de los algoritmos JRip, OneR y PART. Estos tres clasificadores son basados en reglas [12].

Un clasificador basado en reglas utiliza un conjunto de reglas *IF-THEN* para la clasificación. Una regla *IF-THEN* es una expresión de la forma `IF condition THEN conclusion`. La parte *IF* de una regla se conoce como el antecedente o precondición de la regla. La parte *THEN* es la regla consecuente [8]. El clasificador basado en reglas tiene varias ventajas, es altamente expresivo, fácil de interpretar, fácil de generar, puede clasificar nuevas instancias rápidamente, y el rendimiento es comparable con árboles de decisiones [5]. Los clasificadores basados en reglas tienen dos criterios importantes: el *accuracy* de la clasificación en un conjunto de prueba independiente y la diversidad de reglas en los mismos, lo que los hace interesantes en el campo de aprendizaje automático [11]. Existen algunos indicios en la literatura de que reglas muy simples pueden lograr una precisión sorprendentemente alta en muchos conjuntos de datos [18].

En conclusión, escogimos estos clasificadores basados en reglas porque son fáciles de implementar de manera efectiva. Por el hecho también que son fáciles de interpretar para el ser humano, debido a las reglas que generan [3].

También aplicamos filtros, con el objetivo de seleccionar las variables más importantes para la clasificación. Los filtros evalúan las características intrínsecas de los atributos y en base a ello identifican las variables relevantes para un proceso de clasificación. Además tienen la ventaja de tener un bajo costo computational. En este estudio seleccionamos cuatro filtros: Chi.squared, CFS (Selección de atributos basada en correlación), Consistency y RandomForest. En cada modelo elaborado, se realizaron 30 ejecuciones con una semilla diferente. Lo cual proporciona variabilidad en las instancias usadas en cada ejecución. Los experimentos se realizaron utilizando el lenguaje R bajo el entorno RStudio 1.1.463. Utilizamos los paquetes *psych* [19], *Fselector* [21], *caret* [16], *pROC* [20], *rJava* [26] y *Rweka* [12].

Este documento se organizó de la siguiente manera: en la sección 2 presentamos una descripción del conjunto de datos, los algoritmos de aprendizaje automático y las medidas de rendimiento utilizadas en el estudio. La sección 3 describe el procedimiento experimental. En la sección 4 presentamos y discutimos los resultados experimentales. Finalmente, en la sección 5 resumimos los resultados y damos conclusiones del estudio.

2. Materiales y métodos

2.1. Conjunto de datos

Los datos utilizados en este trabajo se recopilaron en el Instituto Nacional de Neurología y Neurocirugía desde 1993 hasta 2002. Hay 129 registros de pacientes, cada uno clasificado con un tipo de los subtipos de SGB: 20 pacientes con AIDP, 37 con AMAN, 59 con AMSAN y 13 con Miller-Fisher.

Tabla 1. Conjunto de atributos usados en este trabajo.

Variable	Nombre de la característica
v2	Criterios que cumple para el Diagnóstico
v4	Edad
v7	Días de inicio de disminución de fuerza muscular o compromiso de nervios craneales al evento previo
v9	Días antes de acudir al médico del inicio de los síntomas
v10	Días de inicio de la dificultad respiratoria
v36	Cuantos días requirió de la máquina para respirar
v5	Sexo
v21	Debilidad
v23	Parestesias (cosquilleos, ardor, molestias en piel)
v34	Disnea (dificultad respiratoria)
v35	Requirió VMA (Ventilación Mecánica Asistida)
v6	Patología evento previo
v22	Simetría
v24	Miembros superiores-fuerza muscular (Fuerza en brazos)
v25	Miembros inferiores - fuerza muscular (Fuerza en piernas)
v26	Lugar donde iniciaron los síntomas
v27	Reflejos
v29	Afecta músculos extraoculares
v30	Ptosis (caída de parpado, III nervio craneal)
v31	Afecta cerebelo
v32	Marcha atáxica (falta de control muscular o de coordinación voluntaria "clínico")
v33	Nervios Craneales involucrados
v37	Complicaciones
v38	Involucra esfínter urinario y rectal
vS	Estación (primavera, verano, otoño, invierno)

El conjunto de datos original tiene 365 atributos, de los cuales los primeros 38 se consideran clínicos, los otros 327 atributos corresponden a pruebas de laboratorio, tratamiento y seguimiento del paciente. De los 38 atributos, 13 fueron descartados debido a que pertenecían a datos como el número de caso, el número de archivo, la admisión hospitalaria y la fecha de alta, la duración en días en diversas situaciones, dejando 25 atributos clínicos relevantes que se muestran en la Tabla 1.

2.2. Algoritmos de aprendizaje automático

JRip: Aprendizaje basado en reglas que implementa el algoritmo de poda incremental repetida para la reducción de errores (RIPPER, por las siglas en Inglés de *Repeated Incremental Pruning to Produce Error Reduction*). JRip identifica las clases creando un conjunto de reglas [10]. La regla tiene la forma siguiente:

```
si (atributo1 <operador relacional> valor1 <operador logico>
   atributo2 <operador relacional> valor2 <...>) entonces
      valor-decisión
```

OneR: Es la abreviatura de Una Regla, es un algoritmo individual de clasificación de aprendizaje automático que genera un árbol de decisión de un nivel. OneR puede inferir reglas de clasificación típicamente simples a partir de un conjunto de instancias que son sencillas para interpretar. Este algoritmo crea una regla para cada atributo en los datos de entrenamiento y luego selecciona la mejor regla como su única regla [4].

PART: Usa el enfoque de Frank y Witten [6], la idea clave en este clasificador es construir un árbol de decisión parcial en lugar de explorar uno completo. Una de las mayores ventajas de PART sobre otros métodos es la simplicidad, combinando dos paradigmas de aprendizaje de reglas produciendo un buen conjunto de reglas sin la necesidad de optimización global.

2.3. Selección de atributos

Los filtros son técnicas de selección de atributos que reducen o establecen atributos por nivel de prioridad del conjunto de datos para tratar de aumentar el rendimiento [15]. Se tomaron del paquete *Fselector* cuatro filtros: CFS, Consistency (usando medida de consistencia), Chi.squared (basado en una prueba de chi-cuadrado) y RandomForest (usando el algoritmo RandomForest). Los dos primeros son algoritmos que encuentran un subconjunto de atributos para datos continuos y discretos; los dos restantes encuentran pesos de atributos discretos [21].

2.4. Medidas de rendimiento

Las medidas de rendimiento son técnicas estadísticas utilizadas para evaluar la calidad del modelo creado.

Se obtuvieron y registraron las medidas de rendimiento clásicas de la literatura [14]. Nuestra medida de rendimiento principal de los modelos fue el *accuracy* balanceado, pues como se indica debajo, se utiliza para cuando los datos no están balanceados. Tal es el caso de nuestro dataset.

Accuracy: Mide el porcentaje de respuestas correctas del modelo [22]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

donde: TP = Verdadero positivo, TN = Verdadero negativo, FP = Falso positivo, FN = Falso negativo.

***Accuracy* Balanceado:** Es la media aritmética de sensibilidad y especificidad, o la exactitud promedio obtenida en cualquiera de los subtipos del SGB cuando los datos no están balanceados [22,27]:

$$AccuracyBalanceado = \frac{\left(\frac{TP}{(TP+FN)}\right) + \left(\frac{TN}{(FP+TN)}\right)}{2}. \quad (2)$$

Sensibilidad: Se puede definir como la capacidad de la prueba para clasificar correctamente los subtipos del SGB [22]:

$$Sensibilidad = \frac{TP}{TP+FN}. \quad (3)$$

Especificidad: Es el complemento de la especificidad, se conoce como falsos positivos [22]:

$$Especificidad = \frac{TN}{TN+FP}. \quad (4)$$

Área bajo la curva ROC: La métrica AUC es un gráfico bidimensional que proporciona un resumen del rendimiento de un modelo de clasificación en presencia de conjuntos de datos desequilibrados con error desigual. Se utiliza una medida llamada AUC, que varía de 0 a 1 (que representa el verdadero positivo y el falso positivo) asociada con la curva ROC. [7].

2.5. Validación

Utilizamos la evaluación *train-test* para los experimentos, con dos terceras partes de los datos para el entrenamiento y una tercera parte para pruebas.

3. Diseño experimental

En la sección 2.1 se describe el conjunto de datos utilizado para el diseño de los modelos. Primero se crearon tres modelos usando todas las variables del conjunto de datos con los clasificadores JRip, OneR y PART. Realizamos 30 ejecuciones independientes, en cada ejecución calculamos el *Accuracy* balanceado.

Al final calculamos el promedio de las 30 ejecuciones del *Accuracy* balanceado, la desviación estándar e identificamos el mejor y el peor valor de las ejecuciones efectuadas.

Siguiente a la creación de los modelos anteriores, realizamos la selección de los atributos mediante los filtros seleccionados, obteniendo un total de 12 modelos diferentes, un modelo por filtro aplicado a cada clasificador. El *accuracy* balanceado es la métrica principal en cada modelo.

Los filtros CFS y Consistency proporcionan una lista de los atributos más relevantes. Con ellos creamos un subconjunto a partir del conjunto de datos para

realizar los experimentos con los clasificadores, tal como se describe en el primer párrafo de esta sección.

Los filtros Chi.squared y RandomForest asignan una puntuación a cada atributo, que significa su relevancia. De cada filtro, tomamos los dos mejores atributos, los mejores tres atributos y así sucesivamente hasta tomarlos todos. Con cada uno de estos atributos, creamos conjuntos de datos que contienen las instancias del SGB junto con el atributo clase. La idea fue evaluar los modelos de diagnóstico que contenían los atributos más relevantes, de acuerdo a cada uno de estos filtros.

4. Resultados y discusión

La Tabla 2 muestra los subconjuntos seleccionados por los filtros CFS y Consistency, también muestra los atributos con los que se alcanzaron los mejores resultados: para Chi.squared el mejor resultado se logró con 5 atributos y para RandomForest con 4 atributos. Se puede observar que la variable `v25` (que hace referencia a la falta de fuerza en las extremidades inferiores) y la variable `v29` (referente a la afectación de los musculos que mueven el ojo), fueron seleccionadas por los cuatro filtros, por lo que se infiere la relevancia de las variables para los modelos.

Tabla 2. Atributos seleccionados por cada filtro.

CFS	Consistency	Chi-squared	RandomForest
v2	v6	v21	v25
v21	v25	v2	v30
v22	v26	v30	v29
v24	v27	v25	v2
v25	v29	v29	-
v29	v35	-	-
v30	vS	-	-
v31	-	-	-
v33	-	-	-

La Tabla 3 muestra los resultados de los experimentos con todas las variables del conjunto de datos, ordenados de la forma en como se explica en la sección 3. En negritas se muestra el resultado más alto logrado por uno de los tres clasificadores.

La Tabla 4 muestra la comparación de los mejores resultados obtenidos por los clasificadores JRip, OneR y PART con cada filtro correspondiente, así como el promedio de las 30 ejecuciones, la desviación estándar, el mejor y el peor resultado. Se puede observar por medio de las tablas que de manera general los filtros mejoran el rendimiento de los clasificadores. El resultado promedio más alto con todas las variables del conjunto de datos fue de 0.6265 y los resultados

Tabla 3. Rendimiento de los clasificadores basados en reglas, usando todas las variables en el dataset.

Clasificador	Parámetros	Accuracy	**Accuracy balanceado**	Sensibilidad	Especificidad	AUC
JRip	Promedio	0.4927	**0.6265**	0.4619	0.7912	0.4917
	Desv Est	0.0522	0.0459	0.0750	0.0208	0.1257
	Mejor	0.5854	0.7009	0.5855	0.8325	0.7222
	Peor	0.3902	0.5426	0.3202	0.7410	0.2778
OneR	Promedio	0.4577	0.5982	0.4171	0.7792	0.5894
	Desv Est	0.0692	0.0649	0.1051	0.0271	0.1417
	Mejor	0.5610	0.6939	0.5669	0.8209	0.875
	Peor	0.2195	0.4335	0.1645	0.7026	0.3056
PART	Promedio	0.4122	0.5877	0.4015	0.7739	0.5556
	Desv Est	0.0514	0.0502	0.0817	0.0214	0.1292
	Mejor	0.5366	0.7073	0.5899	0.8248	0.7778
	Peor	0.2927	0.4923	0.2336	0.7351	0.3194

promedios con la aplicación de los filtros superaron en su mayoria (9 de 12 modelos) al resultado anterior. El promedio más alto fue de 0.6564.

En la Figura 1 se observan los mejores resultados que pertenecen a los clasificadores con todas las variables (+) y a la aplicación de los filtros seleccionados. Se obtuvo el valor más alto de 0.6564 con el clasificador OneR y con los filtros Chi.squared (▲) y RandomForest (⊠) que aparece con la abreviatura RanFor en la imagen. La linea punteada intercepta el número 0.6564 en el eje y, que representa al valor más alto alcanzando por los modelos mencionados anteriormente. Se observa en el eje x, el clasificador OneR y sobre él, los puntos más altos alcanzado por los filtros Chi.squared y RandomForest, sobrepuestos uno sobre el otro. Esto debido a que los dos lograron el mismo resultado promedio.

Del mismo modo se aprecia que debajo del mayor resultado alcanzado con todas las variables (+) solo se encuentran los modelos JRip con filtro Consistency (■) y PART con filtros CFS (●) y Consistency (■) por lo que se puede definir que mediante la selección de atriburos por los métodos filtros se incrementa el rendimiento con respecto a los modelos con todas las variables.

5. Conclusión

El objetivo de este trabajo fue observar el comportamiento de los clasificadores basados en reglas para la creación de un modelo de diagnóstico efectivo para el SGB con variables clínicas. Para ello aplicamos tres clasificadores basados en reglas (JRip, OneR, PART) y cuatro métodos de selección de atributos relevantes (Chi.squared, CFS, Consistency, RandomForest). Realizamos experimentos usando todas las variables del conjunto de datos y usando únicamente las variables relevantes identificadas por cada filtro.

Para este estudio comparamos el rendimiento de los modelos elaborados en este trabajo, con el parámetro del *accuracy* balanceado. El mejor rendimiento

Tabla 4. Rendimiento de los modelos utilizando las variables relevantes identificadas por cada filtro.

Chi.squared						
Clasificador	Parámetros	Accuracy	**Accuracy balanceado**	Sensibilidad	Especificidad	AUC
JRip	Promedio	0.5398	**0.6549**	0.5023	0.8075	0.5097
	Desv Est	0.0554	0.0453	0.0717	0.0218	0.1047
	Mejor	0.6585	0.7336	0.6272	0.8501	0.7153
	Peor	0.4146	0.5179	0.2785	0.7573	0.2500
OneR	Promedio	0.5122	**0.6564**	0.5113	0.8014	0.6588
	Desv Est	0.0675	0.0518	0.0795	0.0255	0.1317
	Mejor	0.6098	0.7294	0.6272	0.8402	0.8611
	Peor	0.3659	0.5544	0.3509	0.7503	0.3056
PART	Promedio	0.5065	**0.6561**	0.5143	0.7979	0.5736
	Desv Est	0.0646	0.0549	0.0857	0.0251	0.1187
	Mejor	0.6098	0.7599	0.6711	0.8488	0.8333
	Peor	0.3415	0.5219	0.2982	0.7455	0.3889
CFS						
Clasificador	Parámetros	Accuracy	**Accuracy balanceado**	Sensibilidad	Especificidad	AUC
JRip	Promedio	0.5293	0.6463	0.4899	0.8027	0.5100
	Desv Est	0.0521	0.0523	0.0845	0.0222	0.1138
	Mejor	0.6341	0.7371	0.6272	0.847	0.7083
	Peor	0.439	0.5179	0.2785	0.7525	0.2778
OneR	Promedio	0.4894	0.6319	0.4712	0.7926	0.616
	Desv Est	0.0617	0.0479	0.0756	0.0225	0.1549
	Mejor	0.6098	0.7294	0.6195	0.8402	0.875
	Peor	0.3659	0.544	0.3224	0.7503	0.3056
PART	Promedio	0.4561	0.6241	0.4615	0.7868	0.5475
	Desv Est	0.0651	0.0499	0.0795	0.0227	0.1357
	Mejor	0.561	0.705	0.5844	0.8277	0.8125
	Peor	0.2927	0.526	0.3114	0.7363	0.3472
Consistency						
Clasificador	Parámetros	Accuracy	**Accuracy balanceado**	Sensibilidad	Especificidad	AUC
JRip	Promedio	0.4813	0.6098	0.4394	0.7803	0.5352
	Desv Est	0.0693	0.0436	0.0673	0.0221	0.1096
	Mejor	0.6098	0.7142	0.5987	0.8298	0.7431
	Peor	0.2683	0.4557	0.2018	0.7097	0.2500
OneR	Promedio	0.5041	0.6473	0.4974	0.7972	0.647
	Desv Est	0.0728	0.0625	0.0965	0.0298	0.1334
	Mejor	0.6098	0.7294	0.6272	0.8402	0.8611
	Peor	0.3415	0.4508	0.1919	0.7096	0.3056
PART	Promedio	0.4333	0.5997	0.4264	0.773	0.5435
	Desv Est	0.0743	0.0422	0.0632	0.0249	0.0815
	Mejor	0.5610	0.6932	0.5669	0.8196	0.7083
	Peor	0.2927	0.5146	0.3059	0.7233	0.3611
RandomForest						
Clasificador	Parámetros	Accuracy	**Accuracy balanceado**	Sensibilidad	Especificidad	AUC
JRip	Promedio	0.5260	0.6540	0.5083	0.7996	0.5495
	Desv Est	0.0499	0.0408	0.0643	0.0193	0.0906
	Mejor	0.6829	0.7684	0.6743	0.8624	0.7500
	Peor	0.4390	0.5739	0.3695	0.7660	0.3333
OneR	Promedio	0.5122	**0.6564**	0.5113	0.8014	0.6588
	Desv Est	0.0675	0.0518	0.0795	0.0255	0.1317
	Mejor	0.6098	0.7294	0.6272	0.8402	0.8611
	Peor	0.3659	0.5544	0.3509	0.7503	0.3056
PART	Promedio	0.4992	0.6483	0.501	0.7956	0.5595
	Desv Est	0.0683	0.0445	0.067	0.0239	0.1257
	Mejor	0.5854	0.7299	0.6272	0.8416	0.8611
	Peor	0.3415	0.5317	0.3268	0.7366	0.2778

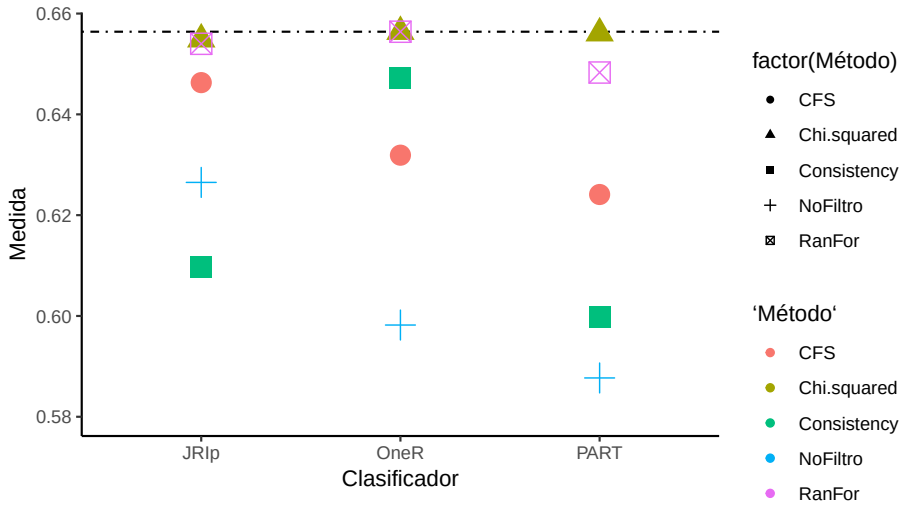


Fig. 1. Comparación de los promedios del *accuracy* balanceado de cada modelo.

obtenido fue de 0.6564 con el clasificador OneR y los filtros Chi.squared y RandomForest.

Es de mucho interés nuestro el poder construir un clasificador con un *accuracy* balanceado con un porcentaje de confiabilidad mejorado, para asegurar el diagnóstico de los 4 subtipos de SGB. Por lo tanto se propone continuar con diferentes experimentos para mejorar este resultado como algoritmos de árbol de decisión, de regresión, selección de atributos mediante *wrappers*, y demás, para apoyar de mejor manera a esta área de la medicina.

Agradecimientos Al CONACYT por el apoyo a la Maestría en Ciencias de la Computación de la Universidad Juárez Autónoma de Tabasco.

Referencias

1. Canul-Reich, J., Frausto-Solís, J., Hernández-Torruco, J.: A predictive model for Guillain-Barré syndrome based on single learning algorithms. Computational and Mathematical Methods in Medicine 2017, 1–9 (2017)
2. Canul-Reich, J., Hernández-Torruco, J., Frausto-Solís, J., Méndez Castillo, J.J.: Finding relevant features for identifying subtypes of Guillain-Barré syndrome using quenching simulated annealing and partitions around medoids. International Journal of Combinatorial Optimization Problems and Informatics 6(2), 11–27 (2015)
3. Cohen, W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning. pp. 115–123. Morgan Kaufmann (1995)
4. Cruz-Monteagudo, M., Cordeiro, M.N.D., Borges, F.: Computational chemistry approach for the early detection of drug-induced idiosyncratic liver toxicity. Journal of computational chemistry 29(4), 533–549 (2008)

5. Farid, D.M., Al-Mamun, M.A., Manderick, B., Nowe, A.: An adaptive rule-based classifier for mining big biological data. Expert Systems with Applications 64, 305–316 (2016)
6. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: Proceedings of the Fifteenth International Conference. pp. 144–151. University of Waikato, Department of Computer Science (1998)
7. Gu, J., Ghosal, S.: Bayesian ROC curve estimation under binormality using a rank likelihood. Journal of statistical planning and inference 139(6), 2076–2083 (2009)
8. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2011)
9. Hernández-Torruco, J., Canul-Reich, J., Frausto-Solís, J., Méndez-Castillo, J.J.: Towards a predictive model for Guillain-Barré syndrome. In: Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 7234–7237. Milano, Italy (2015)
10. Hindle, A., German, D.M., Godfrey, M.W., Holt, R.C.: Automatic classication of large changes into maintenance categories. In: IEEE 17th International Conference on Program Comprehension. pp. 30–39 (2009)
11. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning 11(1), 63–90 (1993)
12. Hornik, K., Buchta, C., Hothorn, T., Karatzoglou, A., Meyer, D., Zeileis, A., Hornik, M.K.: Package *RWeka* (2020)
13. Instituto Mexicano del Seguro Social: Diagnóstico y tratamiento: Síndrome de Guillain-Barré, segundo y tercer nivel de atención. Catálogo Maestro de Guías de Práctica Clínica. IMSS (2016)
14. Jiao, Y., Du, P.: Performance measures in evaluating Machine Learning based bioinformatics predictors for classifications. Quantitative Biology 4(4), 320–330 (2016)
15. Keleş, M.K., Kılıç, Ü.: Artificial Bee Colony algorithm for feature selection on SCA-DI dataset. In: 3rd International Conference on Computer Science and Engineering (UBMK). pp. 463–466. IEEE (2018)
16. Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., Team, R.C., et al.: Package *caret*. The R Journal (2020)
17. Peric, S., Milosevic, V., Berisavac, I., Stojiljkovic, O., Beslac-Bumbasirevic, L., Marjanovic, I., Djuric, V., Djordjevic, G., Rajic, S., Cvijanovic, M.: Clinical and epidemiological features of Guillain-Barré syndrome in the Western Balkans. Journal of the peripheral nervous system 19(4), 317–321 (2014)
18. Rendell, L., Seshu, R.: Learning hard concepts through constructive induction: Framework and rationale. Computational Intelligence 6(4), 247–270 (1990)
19. Revelle, W., Revelle, M.W.: Package psych. The Comprehensive R Archive Network (2015)
20. Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.C., Müller, M., Siegert, S., Doering, M., Robin, M.X.: Package *pROC*. Tech. rep. (2020)
21. Romanski P., Kotthoff L., .K.M.L.: Selecting Attributes Package Fselector. CRAN (2018)
22. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information processing & management 45(4), 427–437 (2009)
23. Steinberg, J.S., Koski, C.L.: Síndrome de Guillain-Barré, polineuropatía desmielinizante inflamatoria aguda (PDIA) y sus variantes. The GBS/CIDP Fundation International (2010)
24. Torres-Vásquez, M., Chávez-Bosquez, O., Hernández-Ocaña, B., Hernández-Torruco, J.: Classification of guillain–barré syndrome subtypes using sampling techniques with binary approach. Symmetry 12(3), 482 (2020)

25. Torres-Vásquez, M., Hernández-Torruco, J., Hernández-Ocana, B., Chávez-Bosquez, O.: Balanceo de datos del sındrome de Guillain-Barré utilizando SMOTE para la clasificación de subtipos. Research in Computing Science 148(7), 113–125 (2019)
26. Urbanek, S., Urbanek, M.S., JDK, S.J.: Package *rJava* (2019)
27. Witten, I., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and techniques. Morgan Kaufmann, third edn. (2011)
28. World Health Organization: Identification and treatment of Guillain-Barré syndrome in the context of the Zika virus outbreak. WHO 1 (2016)

ISSN 1870-4069

Rendimiento de los métodos de sobremuestreo para los modelos predictivos del Síndrome de Guillain-Barré

Manuel Torres-Vásquez[1,2], José Hernández-Torruco[1], Betania Hernández-Ocaña[1], Oscar Chávez-Bosquez[1]

[1] Universidad Juárez Autónoma de Tabasco, División Académica de Ciencias y Tecnologías de la Información, México
[2] Tecnológico Nacional de México/ITS de Centla, División Sistemas Computacionales, México

{jose.hernandezt, betania.hernandez, oscar.chavez}@ujat.mx, mtorresv@itscentla.edu.mx

Resumen. El Síndrome de Guillain-Barré (SGB) es un trastorno neurológico donde el sistema inmune del cuerpo ataca al sistema nervioso periférico. Esta enfermedad es de rápida evolución y puede llevar a una parálisis de todo el cuerpo. En casos graves necesita ventilación asistida. Existen cuatro variantes del SGB: Polineuropatía Desmielinizante Inflamatoria Aguda (AIDP), Neuropatía Axonal Motora Aguda (AMAN), Neuropatía Axonal Sensorial Aguda (AMSAN) y Síndrome de Miller-Fisher (MF). Es necesario identificar el subtipo de SGB que el paciente contrajo debido a que el tratamiento es diferente de acuerdo al subtipo contraído. Esta enfermedad puede afectar la vida y la familia del paciente debido a que puede llevar años la recuperación total. Realizamos este estudio experimental con un *dataset* desbalanceado multiclase. Este *dataset* es un registro de 129 pacientes diagnosticados con alguna variante del SGB. Binarizamos el *dataset* original aplicando la técnica "Uno contra todos". Aplicamos tres métodos de sobremuestreo (MWMOTE, RWO y RACOG), los cuales generan instancias sintéticas usando diferente enfoque. Aplicamos tres clasificadores individuales (Árbol de decisión, SVM y JRip). El rendimiento de los algoritmos se obtuvo mediante la curva ROC y el área bajo la curva AUC. Se obtuvieron modelos predictivos con datos balanceados y modelos predictivos con datos desbalanceados. Aplicamos la prueba Wilcoxon para conocer si existen diferencias estadísticamente significativas entre los modelos balanceados y los modelos desbalanceados. Los resultados muestran que balancear el *dataset* mejora el rendimiento de los modelos predictivos. Balancear la clase minoritaria utilizando el método RWO en combinación con el clasificador SVM obtuvo los mejores resultados.

Palabras clave: Dataset multiclase, desbalanceo, sobremuestreo, aprendizaje automático, clasificación, binarización.

Manuel Torres-Vásquez, José Hernández-Torruco, Betania Hernández-Ocaña, Oscar Chávez-Bosquez

Performance of Oversampling Methods for Guillain-Barré Syndrome Predictive Models

Abstract. Guillain-Barré Syndrome (GBS) is a neurological disorder where the body's immune system attacks the peripheral nervous system. This disease is rapidly evolving and can lead to paralysis of the entire body. In severe cases, it needs assisted ventilation. There are four GBS variants: Acute Inflammatory Demyelinating Polyneuropathy (AIDP), Acute Motor Axonal Neuropathy (AMAN), Acute Sensory Axonal Neuropathy (AMSAN), and Miller-Fisher Syndrome (MF). It is necessary to identify the GBS subtype that the patient has because the treatment is different according to the developed subtype. This disease can affect the life and family of the patient because it can take years for the patient to fully recover. We conducted this experimental study with an imbalanced multiclass dataset. This dataset is a record of 129 patients diagnosed with some variant of GBS. We converted in binary the original dataset applied the technique "One versus all". We applied three oversampling methods (MWMOTE, RWO, and RACOG) which generate synthetic instances with a different approach. We applied three simple classifiers (decision tree, SVM and JRip). The performance of the algorithms was obtained using ROC curve and the area under the curve AUC. Predictive models with balanced data were obtained, along with predictive models with imbalanced data. We applied the Wilcoxon test to find out if there were statistically significant differences between the balanced models and the imbalanced ones. Results show that balancing the dataset improves the performance of predictive models. Balancing the minority class using the RWO method in combination with the SVM classifier obtained the best results.

Keywords: Multiclass dataset, imbalance, oversampling, machine learning, classification, binarization.

1. Introducción

El Síndrome de Guillain-Barré es una rara afección que ocurre cuando el sistema de defensa del cuerpo, comúnmente llamado sistema inmunitario, ataca parte del sistema nervioso periférico. Los primeros síntomas son debilidad muscular y hormigueo en las extremidades. La rápida evolución de este padecimiento puede llevar a paralizar todo el cuerpo. Existen cuatro variantes conocidas: Polineuropatía Desmielinizante Inflamatoria Aguda (AIDP), Neuropatía Axonal Motora Aguda (AMAN), Neuropatía Axonal Sensorial Aguda (AMSAN) y Síndrome de Miller-Fisher (MF). Se desconoce la causa exacta del SGB, pero del 50 % al 70 % de los casos aparecen de una a dos semanas después de una infección respiratoria o gastrointestinal [7]. Actualmente existen estudios en los cuales se relaciona con el Zika [15].

Los casos graves son raros, sin embargo, pueden producir una parálisis casi total y requerir una monitorización de cuidados intensivos prolongada e incluso ventilación mecánica, aunque la mayoría de los casos se recuperan totalmente.

Esta enfermedad puede afectar extremadamente la vida y la familia de un paciente porque puede llevar años la recuperación total del individuo [19]. El ingreso temprano al hospital y el tratamiento temprano son importantes para disminuir la necesidad de asistencia respiratoria y mejorar el resultado. El diagnóstico se basa en hallazgos clínicos, de laboratorio y neurofisiológicos. El tratamiento temprano es crucial para un resultado favorable. El tratamiento varía de acuerdo al subtipo contraído. La identificación del subtipo de SGB que el paciente contrajo en forma temprana, permitirá aplicar el tratamiento adecuado para disminuir la morbilidad y mortalidad a largo plazo [10].

En trabajos anteriores [2,3] se crearon modelos predictivos utilizando los datos originales, es decir, un conjunto de datos conformado por clases desbalanceadas. El desbalanceo de clases es un problema que afecta el resultado de los clasificadores. Este se presenta debido a que los algoritmos de clasificación están basados en la idea que los *datasets* se encuentran balanceados. Cuando un clasificador utiliza datos desbalanceados tiende a ignorar la clase minoritaria y sesgar su resultado en la clase mayoritaria. El desbalanceo de datos se presenta cuando en un *dataset* la cantidad de datos que conforma una de las clases tiene significativamente menos datos respecto a la otra clase [11].

Existen dos tipos de desbalanceo de datos. El primero se conoce como desbalanceo binario. El desbalanceo binario está formado por un *dataset* con dos clases, sin embargo, una de ellas contiene un mayor número de datos con respecto a la otra. A esta clase se le llama clase mayoritaria y a la clase que tiene un menor número de instancias se le conoce como clase minoritaria. El segundo tipo de desbalanceo, es el desbalanceo multiclase. El desbalanceo multiclase se presenta cuando el *dataset* está formado por más de dos clases y su distribución de datos es desigual para cada una de las clases. Para solventar el problema del desbalanceo de datos, han surgido varias técnicas que mejoran el rendimiento de los clasificadores.

Técnica a nivel algoritmo: esta técnica modifica el algoritmo de clasificación para darle mayor peso a la clase minoritaria para ser más preciso. Sin embargo, esta técnica no es variable, debido que depende de un clasificador específico y las modificaciones están destinadas a resolver un problema de desbalanceo para un algoritmo en particular [11].

Técnica a nivel de datos: basándose en el hecho que los datos son los que deben estar equilibrados, este método realiza a través de diferentes técnicas el balanceo de datos antes de pasarlos por el clasificador. A este método se le llama preprocesamiento de datos. La ventaja de esta técnica es que puede ser utilizada prácticamente con cualquier clasificador [8].

Técnica costo-sensitivo: este método asigna pesos o instancias al conjunto de prueba y no al conjunto de entrenamiento como en el modelo a nivel de datos [8].

En la técnica a nivel de datos existen tres diferentes formas de balancear los datos de entrenamiento. El submuestreo de clases, que elimina instancias de la clase mayoritaria hasta equilibrarla con la clase minoritaria. Sin embargo, no es recomendable para un *dataset* con pocos datos debido a la pérdida de datos. El sobremuestreo es una de las técnicas más utilizadas para balancear los datos. Esta técnica genera nuevas instancias para anexarlas a la clase minoritaria y equilibrarla con la clase mayoritaria. Por último, existen métodos híbridos que combinan el sobremuestreo y el submuestreo. En la literatura especializada existen diferentes métodos de sobremuestreo clásicos para balancear la clase minoritaria del conjunto de entrenamiento.

Chawla y colaboradores [4] presentan SMOTE *(Synthetic Minority Oversampling Technique)*, este método es el más usado y con mayor éxito para balancear un *dataset*. SMOTE crea instancias sintéticas a lo largo de los segmentos de la línea que unen a cualquiera de los vecinos más cercanos a la clase k minoritaria. Las nuevas instancias se generan a través de diferencias entre las instancias y su vector de característica considerando sus vecinos más cercanos. Existen métodos basados en SMOTE creados para mejorar la generación de instancias sintéticas, pero SMOTE es el considerado clásico en la literatura.

En un primer estudio exploratorio [17], analizamos el comportamiento de los datos balanceados aplicando el método SMOTE con seis diferentes porcentajes de sobremuestreo sintético para la clase minoritaria, binarizando el dataset aplicando OVO. Los resultados arrojaron que SMOTE al 100 % obtuvieron los mejores resultados. En un estudio más amplio [18], utilizamos dos técnicas diferentes de binarización: OVO (Uno contra Uno) y OVA (Uno contra todos). Los resultados demostraron que eliminar datos baja el rendimiento de los clasificadores, y SMOTE volvió a demostrar su eficacia para mejorar el rendimiento de los clasificadores respecto a los datos desbalanceados.

En este trabajo cumplimos dos objetivos. El primero fue explorar tres diferentes métodos actuales de sobremuestreo para conocer cuál de ellos mejora el rendimiento de los clasificadores, y saber si estos métodos superan el rendimiento de los métodos utilizados en trabajos anteriores. Estos métodos crean instancias sintéticas de diferente manera. El segundo, conocer si crear modelos predictivos con *datasets* balanceados mejoran los modelos predictivos creados con *datasets* desbalanceados realizados en trabajos anteriores.

Para esto, convertimos el *dataset* multiclase desbalanceado en *subsets* binarios utilizando la técnica de binarización "Uno contra todos". Para realizar el balanceo de datos, aplicamos tres métodos de sobremuestreo: MWMOTE [1], RWO [21] y RACOG [6], que generan instancias sintéticas con diferente enfoque cada uno. Las instancias creadas se anexan a la clase minoritaria hasta alcanzar el equilibrio con la clase mayoritaria. Aplicamos tres algoritmos de clasificación que crean modelos predictivos de diferente manera. C4.5 (Árbol de decisión), forma jerárquica; SVM basado en kernel; JRip *(Ripper)* basado en reglas. Utilizamos la curva ROC y el área bajo la curva AUC como medidas de rendimiento. Los modelos creados son el promedio de 60 ejecuciones con diferentes semillas.

Se aplicó la prueba Wilcoxon, para conocer si existe una diferencia estadísticamente significativa entre los modelos predictivos creados con datos balanceados versus los modelos predictivos creados con los datos desbalanceados.

Este artículo está organizado de la siguiente forma: En la sección 2 presentamos una descripción del *dataset*, los algoritmos de clasificación y los métodos de sobremuestreo y la medida de rendimiento utilizada en el estudio. La sección 3 describe el procedimiento experimental. En la sección 4 presentamos los resultados experimentales. Finalmente, en la sección 5 damos conclusiones del estudio y sugerimos trabajos futuros.

2. Materiales y métodos

2.1. Conjunto de datos

El *dataset* usado en este estudio es un registro de pacientes diagnosticados con SGB. Estos datos son una recopilación de información de 129 individuos con algún subtipo de SGB, divididos en: 20 con AIDP, 37 con AMAN, 59 con AMSAN y 13 con MF. Los datos originales están conformados por 4 clases desbalanceadas, es decir, la cantidad de pacientes con algún subtipo de SGB es desigual respecto a las otras. El *dataset* se obtuvo a través del Instituto Nacional de Neurología y Neurocirugía de la Ciudad de México.

El *dataset* original consta de 356 variables. En un artículo anterior seleccionamos las variables más relevantes [9]. Las variables v22, v29, v30 y v31 son de tipo clínico. El resto de variables pertenecen a la prueba de conducción nerviosa. A continuación, se muestran las variables utilizadas en este estudio.
`[v22]`= Simetría (en debilidad), `[v29]`= Afectación de los músculos extraoculares, `[v30]`= Ptosis, `[v31]`= Implicación cerebelosa, `[v63]`= Amplitud del nervio motor mediano izquierdo, `[v106]`= Área bajo la curva del nervio motor cubital izquierdo, `[v120]`= Área bajo la curva del nervio motor cubital derecho, `[v130]`= Amplitud del nervio motor tibial izquierdo, `[v141]`= Amplitud del nervio motor tibial derecho, `[v161]`= Área bajo la curva del nervio motor peroneo derecho, `[v172]`= Amplitud del nervio sensorial mediano izquierdo, `[v177]`= Amplitud del nervio sensorial mediano derecho, `[v178]`= Área bajo la curva del nervio sensorial mediano derecho, `[v186]`= Latencia del nervio sensorial cubital derecho, `[v187]`= Amplitud del nervio sensorial cubital derecho, `[v198]`= Área bajo la curva del nervio sensorial sural derecho.

2.2. Algoritmos de aprendizaje automático

Algoritmos de sobremuestreo

- MWMOTE *(Majority Weighted Minority Oversampling TEchnique)*: Está basado en que no solo el desbalanceo de clases afecta el resultado del clasificador, sino que también la posición de las instancias de la clase minorita repercute en el resultado. Además, en que los métodos de sobremuestreo actuales pueden producir instancias sintéticas incorrectas. El objetivo de

este método es mejorar la selección de las instancias minoritarias y mejorar la forma de generar los nuevos datos sintéticos. MWMOTEestá divido en tres etapas: 1) identifica las muestras de la clase minoritaria más importantes y difíciles de aprender del *dataset* minoritario original, 2) asigna un peso de selección de acuerdo a la importancia en los datos, 3) genera las muestras sintéticas utilizando el enfoque de agrupamiento para asegurarse que se encuentran dentro de un grupo de la clase minoritaria.
- RWO *(Random Walk Over-Sampling approach):* Utiliza un enfoque de sobremuestreo de caminata aleatoria. El objetivo es agregar nuevas instancias sintéticas lo más cerca posible de las originales. Además, respetar la distribución original de la clase minoritaria más expandir los límites de la clase minoritaria lograrán obtener un conjunto de datos balanceado más beneficioso que ayudará en el rendimiento de los clasificadores. RWO intenta mantener la distribución de los datos de la clase minoritaria sin cambios calculando la media y la varianza de los atributos numéricos.
- RACOG *(RApidy COnverging Gibbs):* Utiliza la distribución de probabilidad de la clase minoritaria para seleccionar y crear estratégicamente nuevas muestras de entrenamiento de clase minoritaria. RACOG utiliza un esquema de muestreo de Gibbs para generar una cadena de Márkov con nuevas muestras sintéticas de clase minoritaria utilizando el algoritmo del árbol de dependencia de Chow-Liu. Este método evita la superposición de clases.

Utilizamos estos métodos de sobremuestreo para balancear los datos de entrenamiento debido a que son algoritmos creados recientemente. Además, cada uno de ellos genera de forma diferente instancias sintéticas para la clase minoritaria.

Clasificadores individuales

- C4.5 (Árbol de decisión): Este clasificador basa su resultado en forma jerárquica. Es fácil de interpretar, rápido y robusto al ruido. Es recomendado para diagnóstico médico [8].
- SVM (Máquina de vector soporte): SVM construye un hiperplano en un espacio bidimensional para separar las clases. Es uno de los clasificadores con muy alta precisión respecto a otros clasificadores, como regresión logística y Naive Bayes [8].
- JRip (RIPPER, *repeated incremental pruning to produce error reduction*): Identifica un conjunto de reglas para obtener el resultado. Es fácil de interpretar para los humanos [16].

Seleccionamos estos algoritmos con el objetivo de obtener los resultados usando diferentes enfoques de clasificación.

2.3. Medidas de rendimiento

Evaluamos la eficiencia de los modelos predictivos mediante la curva ROC *(Receiver Operating Characteristic)* y el área bajo la curva AUC *(Area Under*

the Curve). La curva ROC mide qué tan bien se clasifican las predicciones, así como la calidad de las predicciones del modelo. La curva ROC se define como la sensibilidad, que es la tasa de verdadero positivos y 1-especificidad que es la tasa de falsos positivos. El área bajo la curva AUC nos permite identificar una clase. Para este experimento, nos sirve para discriminar entre un subtipo de SGB entre los otros subtipos. En esta medida de rendimiento, los valores $\geq .900$ se consideran modelos excelentes. Los valores $\geq .700$ significa que son buenos modelos. Sin embargo, los valores con $\leq .500$ están considerados malos modelos [14].

2.4. Técnica de binarización

En problemas de clasificación con *datasets* multiclase es común convertir el *dataset* original en subproblemas binarios. Una de las formas generalmente utilizadas es la técnica "Uno contra todos" [13]. Esta técnica toma una clase que se convertirá en la clase minoritaria. La suma de las clases restantes serán la clase mayoritaria. Este proceso se repetirá dependiendo el número de clases que tenga el *dataset* original.

Utilizamos la técnica de binarización "Uno contra todos" para reconstruir el *dataset* original en varios *datasets* binarios:

El primer *subset* está formado por las instancias de la clase 1 (AIDP), llamada clase minoritaria. La clase mayoritaria se forma a través de la suma de las instancias de las clases restantes 2 (AMAN), 3 (AMSAN) y 4 (MF).

El segundo *subset* está formado por las instancias de la clase 2 (AMAN), llamada clase minoritaria. La clase mayoritaria se forma a través de la suma de las instancias de las clases restantes 1 (AIDP), 3 (AMSAN) y 4 (MF).

El tercer *subset* está formado por las instancias de la clase 3 (AMSAN), llamada clase minoritaria. La clase mayoritaria se forma a través de la suma de las instancias de las clases restantes 1 (AIDP), 2 (AMAN) y 4 (MF).

El cuarto *subset* está formado por las instancias de la clase 4 (MF), llamada clase minoritaria. La clase mayoritaria se forma a través de la suma de las instancias de las clases restantes 1 (AIDP), 2 (AMAN) y 3 (AMSAN).

Utilizar esta técnica nos ayuda a identificar uno de los cuatro subtipos de SGB, discriminando de los restantes.

2.5. Validación

Utilizamos la evaluación *train-test* para cada clasificador. Dividimos el *dataset* en dos subconjuntos de datos. El primero son los datos de entrenamiento, estos se usaron para construir el modelo. El segundo conjunto de datos llamado prueba, se mantienen aparte y a través de ellos se evaluó el modelo. El rendimiento de este modelo se realizó con los datos de prueba. Usamos dos tercios de los datos para el entrenamiento y un tercio para las pruebas del modelo.

3. Procedimiento experimental

Utilizamos la técnica de binarización "Uno contra todos" para crear cuatro *subsets* binarios desbalanceados. En la Tabla 1 se muestra el resultado de la binarización del *dataset* original multiclase desbalanceado SGB. La primera columna, muestra los cuatro *subsets* obtenidos con la técnica "Uno contra todos". La segunda columna, muestra el número de instancias que forman la clase minoritaria para cada *subset*. La tercera columna, muestra el subtipo que pertenece la clase minoritaria. La cuarta columna, muestra el número de instancias que forman la clase mayoritaria para cada *subset*. La quinta columna, muestra los subtipos que forman la clase mayoritaria.

Tabla 1. *Subsets* binarios obtenidos con el método "Uno contra todos".

Subset	**Clase minoritaria**	**Subtipo**	**Clase mayoritaria**	**Suma subtipos**
$SGB1$	20	AIDP	109	AMAN, AMSAN y MF
$SGB2$	37	AMAN	92	AIDP, AMSAN y MF
$SGB3$	59	AMSAN	70	AIDP, AMAN y MF
$SGB4$	13	MF	116	AIDP, AMAN y MF

Tabla 2. Resultados de los *subsets* balanceados aplicando los métodos de sobremuestreo.

Subset	**Datos de entrenamiento desbalanceados**	**Datos generados con MWMOTE, RWO y RACOG**	**Clase minoritaria balanceada**	**Clase mayoritaria original**
$SGB1$	15	56	71	71
$SGB2$	25	37	62	62
$SGB3$	40	7	47	47
$SGB4$	9	69	78	78

Particionamos cada *subset* en entrenamiento (66 %) y prueba (34 %). A los cuatro *subsets* de entrenamiento de la clase minoritaria, se le aplicaron tres diferentes métodos de sobremuestreo (MWMOTE, RWO y RACOG) con el objetivo de equilibrarlos con la clase mayoritaria. Los datos de prueba se utilizaron para medir el rendimiento de los modelos obtenidos. En la Tabla 2 se muestran los cuatro *subsets* balanceados con las tres diferentes técnicas de sobremuestreo. En la primera columna se muestran los cuatro *subsets*. La segunda columna, muestran los datos de entrenamiento de la clase minoritaria. La tercera columna, muestra las instancias generadas a través de cada método de sobremuestreo.

La cuarta columna, muestra la suma de las instancias originales con las instancias generadas por los métodos de sobremuestreo. La quinta columna, muestra los datos de entrenamiento de la clase mayoritaria.

Una vez balanceados los cuatro *subsets* se les aplicaron los tres clasificadores individuales (C4.5, SVM y JRip). Cada uno de los modelos es el promedio de 60 ejecuciones con diferente semilla. La medida de rendimiento fue el área bajo la curva AUC. Por otro lado, se utilizaron los *subsets* desbalanceados para aplicar los tres clasificadores individuales. Se obtuvieron modelos predictivos con los datos desbalanceados.

Realizamos una comparación de los modelos obtenidos con los datos balanceados contra los modelos conseguidos con los datos desbalanceados. Se aplicó la prueba Wilcoxon para conocer si existe diferencia estadísticamente significativa, siempre y cuando los modelos balanceados superaran los modelos desbalanceados.

Los experimentos en R se realizaron sobre RStudio 1.2.1335. Utilizamos el paquete `imbalance` [5] para los tres métodos de sobremuestreo MWMOTE, RACOG y RWO. Respecto a los clasificadores, utilizamos el paquete `RWeka 0.4-39` [20] para C4.5 y JRip. El clasificador SVM usamos el paquete `e1071 1.7-0` [12].

SVM se optimizó a través de la función *tune*, asignando los valores 0.001, 0.01, .1, 1, 10, 50, 80, 100 para el parámetro C.

4. Resultados y discusión

La Tabla 3 muestra los modelos predictivos obtenidos mediante la aplicación de tres diferentes métodos de sobremuestreo (MWMOTE, RWO y RACOG). Utilizamos la curva ROC y el área bajo lacurva AUC como métrica de los modelos. Cada modelo es el promedio de los resultados obtenidos a través de 60 ejecuciones. Se realizó la prueba estadística Wilcoxon de los modelos balanceados contra los modelos desbalanceados, siempre y cuando el modelo balanceado haya obtenido un mejor rendimiento respecto al modelo desbalanceado.

El objetivo fue conocer si existe una diferencia estadísticamente significativa entre dichos modelos. En la columna 1 se muestran los cuatro *subsets* obtenidos mediante la binarización del *dataset*. La columna 2 muestra los tres clasificadores utilizados en cada *subset* para obtener los modelos predictivos. En la columna 3 se muestran los modelos obtenidos con los clasificadores utilizando datos desbalanceados. En las columnas 4, 6 y 8 se observan los modelos obtenidos usando datos balanceados aplicando tres diferentes técnicas de sobremuestreo y tres clasificadores. Las columnas 5, 7 y 9 muestra el resultado de la prueba Wilcoxon. NPW representa que no fue necesario realizar la prueba Wilcoxon; $\mathcal{NS}$ significa que no se encontró diferencia significativa entre los modelos balanceados y los modelos desbalanceados; $\boldsymbol{\mathcal{S}}$ muestra que existe diferencia estadísticamente significativa entre los modelos balanceados y los modelos desbalanceados. Se obtuvieron 36 modelos balanceados creados con los tres diferentes métodos de sobremuestreo (MWMOTE, RWO y RACOG) y tres clasificadores (C4.5, SVM y

JRip). Por otro lado, se crearon 12 modelos utilizando los datos desbalanceados. Se realizó una comparación entre los modelos desbalanceados y los modelos balanceados. La comparación se realizó a través de la prueba estadística Wilcoxon para conocer si existen diferencias estadísticamente significativas entre dichos modelos. Con cada *subset* se crearon 9 modelos balanceados y 3 modelos desbalanceados

Usando el *subset* SGB1 los resultados muestran que el método MWMOTE con el clasificador JRip obtuvieron un modelo con diferencia estadísticamente significativa. El clasificador SVM con los datos balanceados utilizando los métodos MWMOTE y RWO, obtuvieron dos modelos que superaron el rendimiento de los datos desbalanceados, sin embargo, no se encontraron diferencias estadísticamente significativas. En los 6 modelos restantes no se realizaron prueba Wilcoxon debido que los datos desbalanceados tuvieron un mejor rendimiento respecto a los datos balanceados.

Con el *subset* SGB2 los resultados muestran que el método RWO utilizando el clasificador SVM obtuvo una diferencia estadísticamente significativa. El clasificador SVM en combinación con RACOG mejoró los datos desbalanceados; el clasificador JRip aplicando MWMOTE y RWO obtuvieron dos modelos que mejoraron el rendimiento de los modelos desbalanceados; El clasificador SVM usando RACOG obtuvo un modelo que mejoró los modelos desbalanceados, sin embargo, en ninguno de los casos se encontraron diferencias estadísticamente significativas. En otros cinco modelos balanceados no se realizó la prueba Wilcoxon debido que los modelos con datos desbalanceados obtuvieron un mejor rendimiento respecto a los modelos balanceados.

Aplicando el *subset* SGB3 los resultaron muestran que 6 modelos balanceados superaron a los modelos desbalanceados. El método de sobremuestreo RACOG con los tres clasificadores, los tres modelos balanceados mejoraron a los modelos desbalanceados. El método RWO con los clasificadores SVM y JRip, dos modelos balanceados superaron a los modelos desbalanceados. El método MWMOTE utilizando el clasificador SVM, un modelo balanceado mejoró los modelos desbalanceados. Sin embargo, en los seis casos no se encontraron diferencias estadísticamente significativas. En tres modelos balanceados no se realizó la prueba Wilcoxon debido que los modelos con datos desbalanceados obtuvieron un mejor rendimiento respecto a los modelos balanceados. Este *subset* obtuvo el peor rendimiento. En ninguno de los 9 modelos creados con datos balanceados se encontró diferencia estadísticamente significativa.

Finalmente, con el *subset* SGB4, en todos los casos los modelos balanceados superaron el rendimiento de los modelos desbalanceados. El método RWO con los tres clasificadores,tres modelos obtuvieron diferencias estadísticamente significativas. MWMOTE en combinación con C4.5 y SVM, dos modelos obtuvieron diferencias estadísticamente significativas. RACOG usando SVM obtuvo un modelo con diferencia estadísticamente significativa. Los tres modelos balanceados restantes mejoraron el rendimiento de los datos desbalanceados, sin embargo, no se obtuvieron diferencias estadísticamente significativas. Este *subset* obtuvo el mejor rendimiento respecto a los demás.

Tabla 3. Resultados de los modelos predictivos utilizando MWMOTE, RWO y RACOG para sobremuestrar la clase minoritaria. Los valores son el promedio del área bajo la curvas AUC obtenidas (NPW = No fue necesaria la prueba de Wilcoxon; $\mathcal{NS}$ = No significantivo; $\boldsymbol{\mathcal{S}}$ = Significantivo).

Sub-conjuntos	**Clasificador**	**Datos desbalan-ceados**	**Balanceo aplicando MWMOTE**	**Resultado Prueba Wilcoxon**	**Balanceo aplicando RWO**	**Resultado Prueba Wilcoxon**	**Balanceo aplicando RACOG**	**Resultado Prueba Wilcoxon**
SGB1	C4.5	0.8130	0.8042	NPW	0.7826	NPW	0.7838	NPW
	SVM	0.7477	0.7542	$\mathcal{NS}$	0.7681	$\mathcal{NS}$	0.7380	NPW
	JRip	0.7826	**0.8275**	$\boldsymbol{\mathcal{S}}$	0.7639	NPW	0.7745	NPW
SGB2	C4.5	0.9003	0.8957	NPW	0.8954	NPW	0.8749	NPW
	SVM	0.8594	0.8593	NPW	**0.9082**	$\boldsymbol{\mathcal{S}}$	0.8625	$\mathcal{NS}$
	JRip	0.8608	0.8667	$\mathcal{NS}$	0.8826	$\mathcal{NS}$	0.8444	NPW
SGB3	C4.5	0.8632	0.8629	NPW	0.8495	NPW	0.8683	$\mathcal{NS}$
	SVM	0.7898	0.7901	$\mathcal{NS}$	0.7960	$\mathcal{NS}$	0.7903	$\mathcal{NS}$
	JRip	0.8470	0.8461	NPW	0.8494	$\mathcal{NS}$	0.8479	$\mathcal{NS}$
SGB4	C4.5	0.7662	**0.8271**	$\boldsymbol{\mathcal{S}}$	**0.8639**	$\boldsymbol{\mathcal{S}}$	0.7746	$\mathcal{NS}$
	SVM	0.6846	**0.7917**	$\boldsymbol{\mathcal{S}}$	**0.8787**	$\boldsymbol{\mathcal{S}}$	**0.7729**	$\boldsymbol{\mathcal{S}}$
	JRip	0.8319	0.8509	$\mathcal{NS}$	**0.8845**	$\boldsymbol{\mathcal{S}}$	0.8537	$\mathcal{NS}$

Tabla 4. Resumen de los modelos con diferencia estadísticamente significativa, por método y por clasificador.

Subsets	**MWMOTE**	**RWO**	**RACOG**
$SGB1$	JRip	-	-
$SGB2$	-	SVM	-
$SGB3$	-	-	-
$SGB4$	C4.5, SVM	C4.5, SVM, JRip	SVM

En la Tabla 4 se muestran los resultados de los modelos que obtuvieron diferencias estadísticamente significativas. El método RWO obtuvo el mejor rendimiento respecto a los otros dos métodos de sobremuestreo. RWO obtuvo cuatro modelos con diferencia estadísticamente significativa.

MWMOTE obtuvo 3 modelos con diferencia estadísticamente significativa. RACOG fue el método con el peor rendimiento, solo se encontró un modelo con diferencia estadísticamente significativa. En cuanto a los clasificadores, empleando SVM se hallaron 4 modelos con diferencia estadísticamente significativa. Aplicando JRip, se encontraron 2 modelos con diferencia estadísticamente significativa. El clasificador C4.5, encontró 2 modelos con diferencia estadísticamente significativa.

De los 36 modelos obtenidos con datos balanceados, 14 modelos superaron los modelos desbalanceados, sin embargo, no se encontró diferencia estadísticamente significativa. Por otro lado, 14 modelos fueron superados por los modelos desbalanceados, por lo tanto, no se aplicó la prueba Wilcoxon. Por último 8 modelos tuvieron diferencia estadísticamente significativa.

El método de sobremuestreo RWO en combinación con el clasificador SVM obtuvo los mejores resultados.

5. Conclusiones

En este estudio, exploramos tres diferentes técnicas de sobremuestreo con la finalidad de balancear el *dataset* multiclase desbalanceado SGB y obtener mediante tres clasificadores, modelos predictivos utilizando datos balanceados. Buscamos comprobar si balancear los datos mejoran el rendimiento de los modelos.

Se obtuvieron cuatro *subsets* a través del método de binarización "uno contra todos". A cada *subset* de entrenamiento aplicamos los métodos MWMOTE, RWO, y RACOG para sobremuestrear la clase minoritaria y equilibrarla con la clase mayoritaria. Por otro lado, se crearon modelos predictivos utilizando datos desbalanceados. Utilizamos 3 clasificadores que determinan sus resultados de diferente manera: C4.5 (jerárquico), SVM lineal (kernel) y JRip (reglas). La medida de rendimiento utilizada fue la Curva ROC y el área bajo la curva AUC. Aplicamos la prueba Wilcoxon para conocer si existe diferencia estadísticamente significativa entre los modelos balanceados y los modelos desbalanceados, siempre y cuando los datos balanceados sean superiores a los datos desbalanceados.

El método de sobremuestreo RWO obtuvo los mejores resultados. Aplicando RWO obtuvimos 12 modelos, en ocho ocasiones superaron los datos desbalanceados, de los cuales, 4 de ellos obtuvieron diferencia estadísticamente significativa; en 4 modelos tuvieron bajo rendimiento respecto a los datos desbalanceados. Usando MWMOTE, 7 modelos superaron los datos desbalanceados, en 3 ocasiones se encontró diferencia estadísticamente significativa; en 5 modelos obtuvieron mejor rendimiento los datos desbalanceados. El método RACOG obtuvo el peor rendimiento. En 7 modelos que superaron los datos desbalanceados, solo uno de ellos obtuvo una diferencia estadísticamente significativa; en 5 modelos obtuvieron mejor rendimiento los datos desbalanceados.

Respecto a los clasificadores, SVM obtuvo el mejor rendimiento. Utilizando SVM se crearon 12 modelos balanceados, 10 modelos superaron a los modelos desbalanceados, en 4 de ellos se encontró diferencia estadísticamente significativa; 2 modelos fueron superados por los datos desbalanceados. Utilizando el clasificador JRip, ocho modelos superaron los modelos desbalanceados, sin embargo, solo dos de ellos se encontró diferencia estadísticamente significativa; cuatro modelos fueron superados por los modelos desbalanceados. El clasificador C4.5 obtuvo el peor rendimiento. Aplicando C4.5 se obtuvieron 12 modelos balanceados, cuatro de ellos superaron los modelos desbalanceados, sin embargo, solo dos tuvieron diferencia estadísticamente significativa; ocho modelos balanceados fueron superados por los modelos desbalanceados.

Los resultados revelan que balancear el conjunto de datos optimiza el rendimiento de los modelos predictivos. Sobremuestrear la clase minoritaria utilizando el método RWO en combinación con el clasificador SVM obtuvo los mejores resultados.

Este estudio es una serie de trabajos que nos permitirán encontrar el mejor método de balanceo a nivel de datos, utilizando sobremuestreo y submuestreo, que incremente el rendimiento de los modelos predictivos para identificar el subtipo de SGB. Esto podrá ser una herramienta complementaria para los especialistas y les permita ayudar en la identificación de alguna de las variantes de SGB que padezca el paciente.

Como trabajos futuros, experimentaremos con variantes clásicas de SMOTE, hibridación de diferentes técnicas de muestreo. Aplicaremos diferentes clasificadores individuales y clasificadores combinados para mejorar el rendimiento de los modelos actuales.

Referencias

1. Barua, S., Islam, M.M., Yao, X., Murase, K.: MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on Knowledge and Data Engineering 26(2), 405–425 (2014)
2. Canul-Reich, J., Frausto-Solís, J., Hernández-Torruco, J.: A predictive model for guillain-barré syndrome based on single learning algorithms. Computational and Mathematical Methods in Medicine pp. 1–9 (2017)

3. Canul-Reich, J., Hernández-Torruco, J., Chávez-Bosquez, O., Hernández-Ocaña, B.: A predictive model for guillain–barré syndrome based on ensemble methods. Computational Intelligence and Neuroscience pp. 1–10 (2018)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
5. Cordón, I., García, S., Fernández, A., Herrera, F.: imbalance: Preprocessing Algorithms for Imbalanced Datasets (2018)
6. Das, B., Krishnan, N.C., Cook, D.J.: RACOG and wRACOG: Two probabilistic oversampling techniques. IEEE Transactions on Knowledge and Data Engineering 27(1), 222–234 (2015)
7. Esposito, S., Longo, M.R.: Guillain–barré syndrome. Autoimmunity Reviews 16(1), 96–101 (2017)
8. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from Imbalanced Data Sets. Springer International Publishing (2018)
9. Hernández-Torruco, J., Canul-Reich, J., Frausto-Solís, J., Méndez-Castillo, J.J.: Feature selection for better identification of subtypes of guillain-barré syndrome. Computational and Mathematical Methods in Medicine 2014, 1–9 (2014)
10. Karalok, Z.S., Taskin, B.D., Yanginlar, Z.B., Gurkas, E., Guven, A., Degerliyurt, A., Unlu, E., Kose, G.: Guillain-barré syndrome in children: subtypes and outcome. Child's Nervous System 34(11), 2291–2297 (2018)
11. Loyola-González, O., Medina-Pérez, M.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Monroy, R., García-Borroto, M.: PBC4cip: A new contrast pattern-based classifier for class imbalance problems. Knowledge-Based Systems 115, 100–109 (2017)
12. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F.: e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien (2018)
13. Ng, S., Tse, P., Tsui, K.: A one-versus-all class binarization strategy for bearing diagnostics of concurrent defects. Sensors 14(1), 1295–1321 (2014)
14. Shatnawi, R., Li, W., Swain, J., Newman, T.: Finding software metrics threshold values using ROC curves. Journal of Software Maintenance and Evolution: Research and Practice 22(1), 1–16 (2009)
15. Soto-Hernández, J.L., de León Rosales, S.P., Cañas, E.S.V., Cárdenas, G., Loza, K.C., Díaz-Quiñonez, J.A., López-Martínez, I., Jiménez-Corona, M.E., Ruiz-Matus, C., Morales, P.K.: Guillain-Barré syndrome associated with Zika virus infection: A prospective case series from Mexico. Frontiers in Neurology 10 (2019)
16. Thakur, S., Meenakshi, E., Priya, A.: Detection of malicious URLs in big data using RIPPER algorithm. In: IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology RTEICT (2017)
17. Torres-Vásquez, M., Chávez-Bosquez, O., Hernández-Ocaña, B., Hernández-Torruco, J.: Balanceo de datos del sındrome de guillain-barré utilizando smote para la clasificación de subtipos. Research in Computing Science 148(7), 113–125 (2019)
18. Torres-Vásquez, M., Chávez-Bosquez, O., Hernández-Ocaña, B., Hernández-Torruco, J.: Classification of guillain–barré syndrome subtypes using sampling techniques with binary approach. Symmetry 12(3), 482 (2020)
19. Wijdicks, E.F., Klein, C.J.: Guillain-barré syndrome. Mayo Clinic Proceedings 92(3), 467–479 (2017)
20. Witten, I.H., Frank, E., Hall, M.A., Pañ, C.: Data Mining, Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2017)

21. Zhang, H., Li, M.: RWO-sampling: A random walk over-sampling approach to imbalanced data classification. Information Fusion 20, 99–116 (2014)

ISSN 1870-4069

Control difuso para el seguimiento de trayectoria de un robot móvil

Camilo Morales Corral, Israel Soto

Universidad Autónoma de Ciudad Juárez,
Instituto de Ingeniería y Tecnología,
México

{camilo.morales,angel.soto}@uacj.mx

Resumen. En el presente trabajo se diseña un sistema de control con la capacidad de manejar los motores motrices de un robot móvil de tracción diferencial. Se parte del modelo dinámico del robot, para lograr el ajuste de un controlador difuso normalizado con estructura PID que permita un correcto seguimiento de los perfiles de velocidad deseados. Para comprobar el buen desempeño del controlador se realizaron simulaciones de tres maniobras básicas que pueden ser llevadas a cabo por el robot.

Palabras clave: Robot móvil, control difuso, seguimiento de trayectoria.

Fuzzy Speed Controller for Trajectory Tracking on a Mobile Robot

Abstract. A controller with the capability to drive the motors of a mobile robot with differential traction was designed. Taking based of the dynamic model of the robot to adjust a fuzzy controller that allow a correct tracking of the desired speed profiles. To verify the good performance of the controller, simulations of three of the basic movement were performed.

Keywords: Mobile robot, fuzzy controller, trajectory tracking.

1. Introducción

En [1] se mencionan los principales desarrollos reportados en la robótica móvil para concluir que a partir de la década del setenta del siglo pasado la investigación y el diseño en esta rama de la tecnología ha ido creciendo de manera exponencial. En los últimos años, los desarrollos en robótica móvil no solo se aplican en la industria, hoy en día se observan dispositivos dedicados a tareas vinculadas al área de servicio. Estos no son más que robots que realizan tareas útiles para el bienestar de los humanos: labores de rescate, guías turísticos,

trasporte de materiales, cuidado y ayuda de personas, limpieza, entre otras [2, 3]. Debido a este gran número de aplicaciones que se han desarrollado en la actualidad, es necesario desarrollar investigaciones en las áreas de: posicionamiento, seguimiento de trayectorias, evasión de obstáculos, y la interacción robot-humano.

Un robot móvil es un sistema electromecánico con la capacidad de desplazarse, interpretar, planear y ejecutar tareas; además de navegar de manera autónoma en un ambiente desconocido. Las parte que lo componen son una estructura rígida (arreglo cinemático), un sistema de actuadores, que dotan de movimiento a la estructura, y un sistema de control [1, 2].

Los robots móviles se pueden clasificar de acuerdo con el tipo de locomoción que utilicen para desplazarse dentro de su espacio de trabajo en tres categorías: robots de ruedas, robots de patas y robots de orugas. Los vehículos con ruedas son la solución más simple para conseguir la movilidad, por lo cual son los más empleados. Aunque presentan la desventaja de que sus prestaciones solo se aprovechan al máximo en superficies duras y libres de obstáculos. Existen diferentes configuraciones en este tipo de locomoción, las cuales confieren diferentes características y propiedades dependiendo de cual se use. Las más usuales son: Ackerman o tipo coche, triciclo, *skid steer*, esteras, omnidireccionales, síncronas, o tracción diferencial.

Un robot móvil con tracción diferencial posee dos ruedas montadas en un único eje, pero propulsadas y controladas independientemente. La tracción se consigue mediante estas dos ruedas y el direccionamiento viene dado por la diferencia de velocidades de cada una. Este sistema es muy útil si consideramos la posibilidad de cambiar su orientación sin movimiento de traslación. Adicionalmente existen una o más ruedas locas o puntos de sustentación para lograr estabilidad en la plataforma móvil.

Según [1], el sistema de actuadores más empleado en robótica móvil son los motores de corriente directa (DC, por sus siglas en ingles), y en especial los de imán permanente. Esto se debe a la sencillez de su control, ya que su modelo es lineal y que el voltaje de control se aplica a la armadura independientemente a la excitación del circuito de campo. De esta forma, en [4] se propone una estrategia de control jerarquizada en dos niveles para el seguimiento de trayectoria. La cual esta conformada por un controlador PID interno asociado a los motores, y un lazo externo asociado al modelo cinemático del robot que se encarga de generar los perfiles de velocidad de cada motor. Según [3, 5], las características no lineales (saturación y fricción) presentes en los motores de DC hacen que se degrade el desempeño de los controladores PID tradicionales. Como vía para mejorar la respuesta del lazo, estos autores emplean controladores difusos.

En el presente trabajo se propone el uso de controladores difusos para el control de movimiento de un robot móvil de tracción diferencial. La estructura establecida incluye el ajuste del controlador y el modelo dinámico del robot, el cual es empleado en la sección de resultados para comprobar la trayectoria del robot en diferentes maniobras.

2. Metodología

2.1. Modelo dinámico del robot móvil

La dinámica considera la evolución de la posición, velocidad y aceleración del robot en respuesta a los pares de actuación de las ruedas. En consecuencia, a la hora de realizar el modelo dinámico de un robot móvil se distinguen los siguientes efectos: pares inerciales y de coriolis, pares de actuación, fricciones viscosas, interacción del contacto de las ruedas con el suelo, entre otros. Para el modelo presentado se consideran las siguientes limitaciones en su construcción:

1. El robot se mueve sobre una superficie plana.
2. No existen elementos flexibles en la estructura del robot.
3. Las ruedas poseen uno o ningún eje de direccionamiento, de manera que este último siempre es perpendicular al suelo.
4. No se consideran ningún tipo de fricciones en elementos móviles del vehículo, o contra el suelo.
5. No se tienen en cuenta los pares gravitacionales, porque se supone que el vehículo viaja por una superficie horizontal.

Modelo dinámico del robot. La configuración geométrica del robot de tracción diferencial empleado en el presente trabajo se presenta en la figura 1.

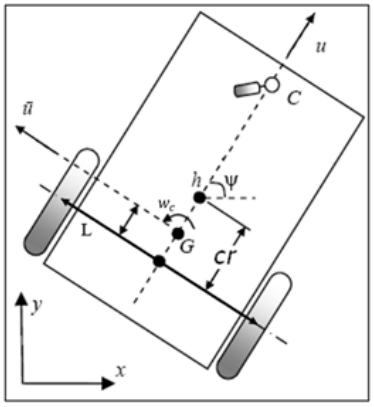


Fig. 1. Diagrama del robot.

En la figura 1 el punto que define la posición del robot es $h = [x, y]^T$, el cual está ubicado a una distancia cr delante del centro del eje que une las ruedas traseras del robot, L representa la distancia entre los centros de las ruedas, u y $\overline{u}$ son las velocidades longitudinal y lateral del centro de masa, w_c es la velocidad angular del robot y ψ es el ángulo de orientación del robot, G es el centro de gravedad y C es el punto de ubicación de la rueda loca.

A partir del diagrama de la figura 1 [6], citado en [7], obtiene el modelo dinámico del robot móvil presentado en la ecuación 1:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \\ \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} ucos\psi - crwsen\psi \\ usen\psi + crwcos\psi \\ w \\ \frac{c}{a}w^2 - \frac{d}{a}u \\ -\frac{e}{b}uw - \frac{f}{b}w \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{a} & 0 \\ 0 & \frac{1}{b} \end{bmatrix} \begin{bmatrix} u_c \\ w_c \end{bmatrix}, \tag{1}$$

donde, $x(t) = \begin{bmatrix} X\ Y\ \psi\ u\ w \end{bmatrix}^T$ es el vector de estado del sistema, $v(t) = \begin{bmatrix} u_c\ w_c \end{bmatrix}^T$ es el vector de control y $p = \begin{bmatrix} a\ b\ c\ d\ e\ f \end{bmatrix}$ el vector de los parámetros dinámicos del robot. Estos parámetros solo dependen de las características físicas del robot (masa, radio de las ruedas, etc.). Los parámetros dinámicos que se emplearon ya fueron validados en [8], donde se presentan las ecuaciones para su cálculo, sus valores numéricos son:

a=0.24089	d=0.99629	r=0.2
b=0.2424	e = -0.00373	cr=0.2
c = -0.00094	f = 1.0915	L = 0.39

Las variables de entrada son la velocidad longitudinal y angular del robot. Estas velocidades son resultantes de las velocidades de cada una de las ruedas independientes, que es sobre las cuales se puede actuar variando la tensión de alimentación a los motores. Según [9] la velocidad lineal del centro del chasis es:

$$\sum_{k=1}^{2} \dot{C}_k = rw_k + (C - O_k)\, w_c, \tag{2}$$

$$\dot{C} = \frac{r}{2}\left(w_i + w_d\right) + crw_c, \tag{3}$$

donde $\dot{C}$ es la velocidad lineal del centro del chasis, r es el radio de las ruedas, w es la velocidad rotacional de las ruedas, C es el centro del chasis, O es velocidad lineal de las ruedas energizadas.

Para obtener el valor de w_c, en lugar de aplicar la sumatoria en la ecuación 2 se restan ambos términos, obteniéndose:

$$w_c = \frac{r}{L}\left(w_i - w_d\right), \tag{4}$$

Con base en las ecuaciones 3 y 4, se despeja de forma conveniente quedando las siguientes relaciones:

$$u_c = \frac{\pi r}{L}\left[\left(w_i + w_d\right) + 2cr\left(w_i - w_d\right)\right], \tag{5}$$

$$w_c = \frac{2\pi r}{L}\left(w_i - w_d\right). \tag{6}$$

Modelado del sistema motriz: reductor de velocidad. Un reductor de velocidad es un acople de ruedas dentadas. En él, debido a la diferencia de diámetros de los engranes, se logra cambiar el par del motor y la velocidad. Está regido por la siguiente ecuación:

$$w_o = \frac{N_1}{N_2} w_i, \tag{7}$$

donde N_1 y N_2 representan el número de dientes de los engranajes (su cociente es el factor de reducción), w_i la velocidad de entrada y w_o la velocidad de salida.

Modelado del sistema motriz: motor de corriente directa. En este caso se emplearán motores de corriente directa de imanes permanentes de grafito sin escobillas los cuales pueden ser modelados siguiendo el procedimiento descrito en [10]. El modelo resultante para describir la dinámica de la velocidad angular del motor en función de la tensión de alimentación queda descrito en la ecuación 8:

$$M(s) = \frac{w(s)}{V(s)} = \frac{k_t r_2}{(L_a s + R_a)(Js + b) + k_t k_v r_2}, \tag{8}$$

donde k_t representa la constante de par del motor, r_2 el rendimiento del reductor, L_a y R_a la inductancia y la resistencia del bobinado del motor, J el momento de inercia, b coeficiente de fricción viscosa y k_v la constante electromotriz. Los datos de los motores empleados son mostrados en la Tabla 1.

Sustituyendo datos de la Tabla 1 en la ecuación 8, y agrupando convenientemente:

$$M(s) = \frac{5{,}9 \times 10^8}{S^2 + 47452S + 2{,}9 \times 10^6}. \tag{9}$$

Teniendo en cuenta que la respuesta del sistema tiene un comportamiento marcadamente de primer orden, es conveniente reducir el modelo a uno de este tipo. Para lo cual se elimina el polo de más alta frecuencia, el cual se encuentra ubicado a una frecuencia mucho mayor que el polo dominante.

Tabla 1. Datos de los motores de corriente directa.

Parámetros	Valor
Constante de par del motor	$k_t = 4{,}95 mNm/A$
Constante electromotriz	$k_v = 1930 rpm/V$
Rendimiento del reductor	$r_2 = 0{,}83$
Inductancia	$L_a = 0{,}24 mH$
Resistencia	$R_a = 11{,}4 \Omega$
Momento de inercia	$J = 2{,}9 \times 10_{-8}$
Coeficiente de fricción viscosa	$b = 3{,}77 \times 10_{-10}$
Número de dientes	$N_1 = 3249$ $N_2 = 196$

El modelo de primer orden obtenido es presentado en la ecuación 10:

$$M(s) = \frac{12420}{S + 61{,}5}. \tag{10}$$

Para comprobar que la reducción realizada fue correcta se les aplica un paso escalón a ambas funciones de transferencia. Empleando las dos respuestas obtenidas se calcula el índice de error medio cuadrático normalizado, el cual cuantifica el error entre ambas. El valor calculado es igual a 0.98 lo que corresponde con un buen ajuste del modelo de primer orden.

2.2. Control difuso

Según [14], el control difuso es un intento de formalizar la experiencia y el conocimiento empírico de los expertos en la operación de un proceso, de manera que puedan reproducirse con semejantes resultados. La estructura de este tipo de controlador, figura 2 está compuesta por cuatro elementos:

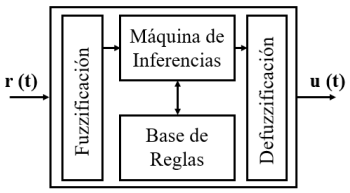


Fig. 2. Estructura del controlador difuso.

1. Base de reglas. Contiene el conocimiento experto resumido en un conjunto de reglas.
2. Máquina de inferencias. Simula el proceso de toma de decisiones del experto, teniendo en cuenta la base de reglas y el conocimiento fuzzificado del proceso.
3. Interfase de fuzzificación. Transforma la información de entrada al controlador en información lingüística que puede ser interpretada por la base de reglas y la máquina de inferencias.
4. Interfase de defuzzificación. Convierte las conclusiones de la máquina de inferencia en acciones de control.

Diseño del controlador difuso. El controlador difuso diseñado para esta aplicación usa dos variables de entrada: el error de la velocidad de los motores ($e(t)$) y su velocidad de cambio($\partial e(t)$). La salida del controlador es la tensión ($u(t)$) que se debe aplicar al devanado del motor para minimizar el error.

Para poder simplificar el diseño del controlador difuso se realizan una serie de consideración presentadas en [11], haciendo uso de estas se llega al controlador difuso normalizado.

El universo del discurso de todas las variables de entrada se define en el rango (-1, 1) y se distribuye en un numero único de funciones de pertenencia (Npart), generalmente impar. Las funciones de pertenencia tienen diferentes formas, las más empleadas son las triangulares, trapezoidales y campanas Gaussianas [12]. En la literatura no existe reportado un método específico para seleccionar la forma de la curva a utilizar. Por lo cual, habitualmente esta selección se realiza mediante el criterio de expertos, empleando algunos experimentos previos. Las funciones de pertenencia seleccionadas para las entradas y salidas del controlador son triangulares, simétricas y distribuidas uniformemente. Esta selección se realiza considerando la futura implementación de un prototipo de pruebas, y la sencillez y bajo costo computacional de estas funciones [13]. Los valores lingüísticos posibles para las variables son: negativo grande (NG), negativo pequeño (NP), cero (CC), positivo pequeño (PP) y positivo grande (PG). La estructura de las funciones de pertenencia es mostrada en la figura 3.

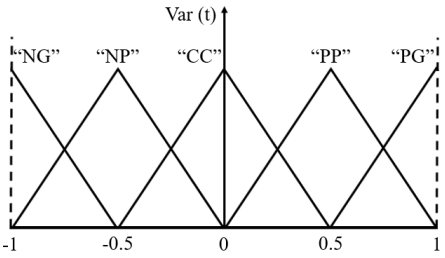


Fig. 3. Funciones de pertenencia para Npart =5.

Después de definir las funciones de pertenencia se establece la base de reglas. Son definidas 25 reglas difusas de tipo IF-THEN, las cuales son presentadas en la Tabla 2. Estas reglas se generan automáticamente partiendo de los rangos de las variables y del signo de la ganancia del proceso (Gan), haciendo uso del algoritmo presentado en [14].

Tabla 2. Base de reglas para el controlador difuso.

u(t)	$\partial e(t)$				
	NG	NP	CC	PP	PG
e(t) NG	NG	NG	NG	NP	CC
e(t) NP	NG	NG	NP	CC	PP
e(t) CC	NG	NP	CC	PP	PG
e(t) PP	NP	CC	PP	PG	PG
e(t) PG	CC	PP	PG	PG	PG

En la máquina de inferencias para la evaluación de las premisas y la implicación de la variable de control se usa el operador mínimo. El método de defuzzificación empleado por este controlador es el de centro de gravedad. Una vez obtenida la salida normalizada hay que calcular la salida real.

Para lo cual se emplea una transformación lineal de la salida normalizada con la siguiente forma:

$$u(t) = u_{min} + \frac{(u+1)(u_{máx} - u_{min})}{2}, \tag{11}$$

donde u_{max} y u_{min} representan los valores máximo y mínimo de la variable de control y $u(t)$ la salida del regulador normalizado.

La estructura empleada para el control de la velocidad de los motores motrices del robot móvil es un controlador difuso normalizado con estructura PID. En este, las dos entradas estarán afectadas respectivamente por los coeficientes de ganancia K_p y K_d, que asumen generalmente valores positivos, y la salida es ahora el incremento de la variable de control $\triangle u(t)$ que es afectado por la ganancia K_i y posteriormente integrado. El significado de estas ganancias es semejante al del regulador PID convencional.

Ajuste del controlador difuso normalizado. Con el fin de ajustar los parámetros del controlador y comprobar su desempeño se realizaron varias simulaciones del sistema. El *software* empleado en las simulaciones es el paquete Simulink®, perteneciente a MATLAB 2018a. El diagrama en bloques para el control de los motores se muestra en la figura 4.

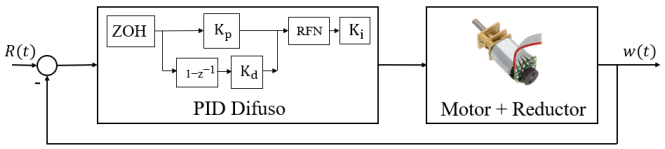


Fig. 4. Diagrama de bloques del sistema de control de un motor.

Para poder acoplar el controlador discreto a un proceso continuo es necesario incluir un retenedor de orden cero (ZOH, por sus siglas en inglés).

El tiempo de muestreo (T_m) a emplear en el ZOH se define con base en el Teorema de muestreo. Si la frecuencia del polo es de $9{,}79 rev/s$, lo que representa $102{,}15 ms$, entonces: $T_m \leq 102{,}15/2$, por lo que se decide usar un $T_m = 50 ms$.

Tras un proceso de ajuste empírico de los parámetros del controlador, se obtuvieron los mostrados en la Tabla 3, con los cuales se logra un correcto seguimiento de la referencia. La respuesta del sistema obtenida a lazo cerrado se muestra en la figura 5 y los indicadores que la describen su respuesta temporal se presentan en la Tabla 4. Los valores obtenidos ratifican los resultados presentados en [15],

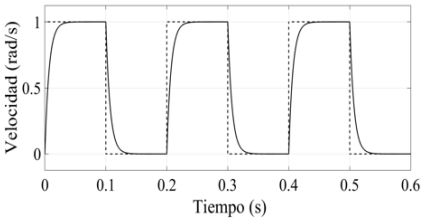


Fig. 5. Respuesta del sistema a lazo cerrado.

donde se compara el controlador difuso con el PID tradicional, demostrando el mejor desempeño del primero en este tipo de sistemas.

Tabla 3. Parámetros de ajuste del controlador.

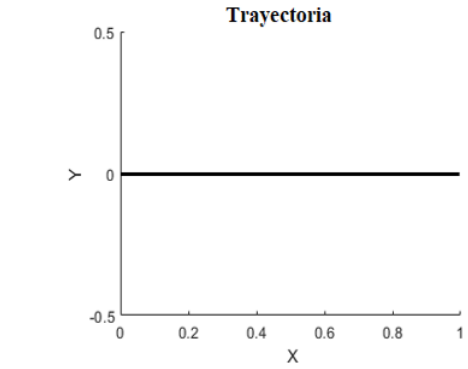
Parámetros	Valor
Ganancia Proporcional	$K_p = 0{,}07$
Ganancia Derivativa	$K_d = 6$
Ganancia Integral	$K_i = 0{,}005$
Número de funciones de pertenencia	$N_{part} = 5$
Signo de la ganancia del proceso	$Gan = 1$
Mínimo de la señal de control	$U_{min} = -6$
Máximo de la señal de control	$U_{macuteax} = 6$

Tabla 4. Indicadores de la respuesta del sistema a lazo cerrado.

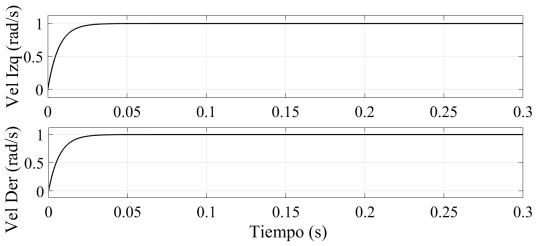
Parámetros	Valor
Error de estado estacionario	$E_{ee} = 0$
Pico máximo	$\%PM = 0\,\%$
Tiempo de establecimiento	$T_e = 27ms$
Tiempo de subida	$T_s = 6{,}92ms$

3. Resultados y discusión

Para simular el comportamiento del robot móvil de tracción diferencial empleado en el presente trabajo, se agrego el modelo dinámico obtenido en la sub sección 2.1. El diagrama en bloques resultante se muestra en la figura 7.



(a) Trayectoria rectilínea.



(b) Velocidades para trayectoria rectilínea.

Fig. 6. Traslación con trayectoria rectilínea.

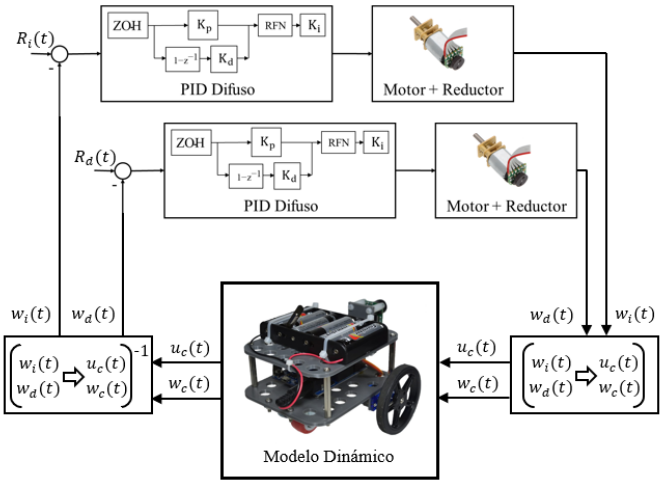


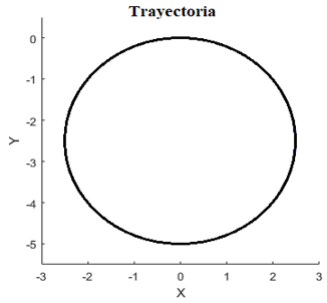
Fig. 7. Diagrama en bloques del sistema de control del robot.

Si se considera el robot móvil como un solido rígido (la separación entre dos puntos cualesquiera es fija e independiente del tiempo), existen cinco tipos de

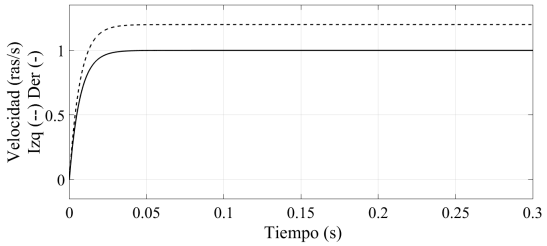
movimiento que puede ejecutar: traslación, rotación alrededor de un eje fijo, movimiento plano general, rotación en torno a un punto fijo y movimiento general. En el análisis cinemático realizado se abordó los casos más simples: movimiento de traslación y movimiento de rotación con eje fijo. Para la simulación se han definido las siguientes maniobras elementales:

Traslación con trayectoria rectilínea: Cuando la orientación y magnitud del vector de velocidad es constante en el tiempo, la trayectoria obtenida es lineal (Figura 6(a)). Para el robot móvil esto se consigue con una velocidad igual en ambas ruedas (Figura 6(b)), para especificar la dirección y sentido del movimiento; y con velocidad angular nula.

Traslación con trayectoria circular: En el caso que la magnitud del vector de velocidad sea lo único constante en el tiempo, el movimiento tiene trayectoria curvilínea. Si la trayectoria corresponde a una curva conocida, como lo es un circulo, la denominación es traslación circular (Figura 8(a)). En el robot móvil esto implica que la velocidad de una de las ruedas sea mayor que la otra (Figura 8(b)).



(a) Trayectoria circular.

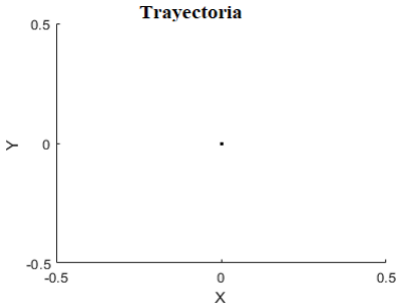


(b) Velocidades para trayectoria circular.

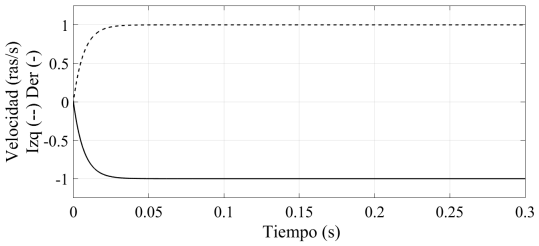
Fig. 8. Traslación con trayectoria circular.

Movimiento de rotación con eje fijo: Este movimiento esta presente cuando existen dos puntos del solido rígido que tienen velocidad nula. Denominándose a la recta que los contiene como eje instantáneo de rotación. En el

caso del robot móvil una rotación sobre el eje que pasa por su centro de masa (Figura 9(a)), se consigue cuando las velocidades de ambas ruedas son iguales en magnitud pero contrarias en sentido (Figura 9(b)).



(a) Trayectoria de rotación.



(b) Velocidades para trayectoria de rotación.

Fig. 9. Movimiento de rotación con eje fijo.

4. Conclusiones

Se logró representar el motor de CD con un modelo tan simple como lo es uno de primer orden. Es conocido que la respuesta del sistema real es algo más compleja ya que presenta no linealidades debido a la saturación, a la rotación y otras motivadas por las transiciones de rozamiento estático y cinético.

Para dar solución al problema real se diseñó un controlador difuso, ya que no necesita un conocimiento preciso del proceso a controlar y su salida de control está limitada a un rango. La estructura se seleccionó del tipo PID, porque el sistema no tiene comportamiento integrador intrínsecamente y se necesita que el error de estado estacionario sea cero. Sobre los resultados obtenidos mediante simulación se describieron tres tipos de maniobras para analizar la movilidad del vehículo, demostrando un correcto desempeño del controlador.

Referencias

1. Sotelo, V. R. B., Sánchez, J. R. G., Ortigoza, R. S.: Robots Móviles: Evolución y Estado del Arte. Polibits, (35), 12-17 (2007)
2. Serrano, A. R.: Navegación autónoma de robots móviles mediante lógica difusa. Instituto Tecnológico y de Estudios Superiores de Monterrey (1996)
3. Luna, K. L., Palacios, E. R., Marín,A.: A fuzzy speed controller for a guide robot using an HRI approach, IEEE Latin America Transactions, vol. 16, no. 8, p. 6 (2018)
4. Jiménez T. S.: Diseño, construcción y control en tiempo real de un robot móvil, Departamento de Ingeniera Eléctrica, CINVESTAV (1999)
5. Ismail, N. L., Zakaria, K. A., Nazar, N. S. M., Syaripuddin, M., Mokhtar, A. S. N., Thanakodif, S.: DC Motor Speed Control using Fuzzy Logic Controller, International Conference on Engineering and Technology (IntCET), Putrajaya, Malaysia (2017)
6. Cruz, C. d. l., Carelli, R.: Linealización con realimentación del modelo dinámico de un robot móvil y control de seguimiento de trayectoria, XX Congreso Argentino de Control Automático, Buenos Aires, Argentina (2006)
7. Rosales, A., Peña, M., Scaglia, G., Mut, V., Sciascio, F. d.: Dynamic model based predictive control for mobile robots, XII Reunión de Trabajo en Procesamiento de la Información y Control, Santa Cruz, Argentina (2007)
8. Rossomando, F. G., Soria, C., Carelli, R.: Algoritmo compensador neuronal discreto de dinámica en robots móviles usando filtro de Kalman extendido, Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería, vol. 29, no. 1, p. 9 (2013)
9. Quintero, A. d. P.: Modelo cinemático dinámico del mini robot móvil RICIMAF, Revista de Ingeniería Electrónica, Automática y Comunicaciones (RIELAC), vol. XXXIII, no. 3, pp. 49-62 (2012)
10. Azcon, A. B.: Análisis y diseño del control de posición de un robot móvil con tracción diferencial., Departamento de Ingeniería Electrónica, Eléctrica y Automática, Universidad Rovira (2003)
11. Martínez, R. F. T., Behar, A. A.: Laboratorio 4 Reguladores Difusos Normalizados. Estructuras PD, PI y PID., Diplomado "Control Avanzado", Instituto de Cibernética, Matemática y Física (2010)
12. Mitaim, S., Kosko, B.: What is the Best Shape for a Fuzzy Set in Function Approximation?, IEEE 5th International Fuzzy Systems, New Orleans, USA, (1996)
13. Farooq, u., Hasan, K. M., Hanif, A., Amar, M., Asad, M. U.: Fuzzy Logic Based Path Tracking Controller for Wheeled Mobile Robots, International Journal of Computer and Electrical Engineering, vol. 6, no. 2, p. 6, (2014)
14. Behar, A. A.: Control Difuso, Diplomado "Control Avanzado", Instituto de Cibernética, Matemática y Física (2010)
15. Almatheel,Y. A., Abdelrahman, A.: Speed Control of DC Motor Using Fuzzy Logic Controller, International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), Khartoum, Sudan (2017)

ISSN 1870-4069

Localización de un robot móvil mediante el filtro de Kalman extendido y el simulador Gazebo

Rolan Bacilio Anota, Eduardo Sánchez-Watanabe,
Alberto Petrilli-Barceló, Fermín Ramírez-Leyva

Universidad Tecnológica de la Mixteca,
México

{adairbacilio,e.akio.watanabe}@gmail.com,
{petrilli,hugo}@mixteco.utm.mx

Resumen. La localización de un robot móvil permite realizar tareas de planeación de trayectorias. Aunque existen modelos dinámicos para distintos tipos de robots móviles, los efectos dinámicos no lineales en el ambiente, modelo, sensores y mecánica impiden un conocimiento exacto de la posición del robot. Por lo anterior es necesario estimar la posición considerando efectos probabilísticos. El Filtro de Kalman Extendido (EKF) permite la estimación de la pose de un robot móvil considerando desviaciones respecto al modelo dinámico no lineal. Se presenta la implementación del EKF para la localización de un robot móvil en la plataforma de simulación Gazebo. La posición del robot es obtenida mediante segmentación por color, la imagen se adquiere a través de una cámara colocada en el entorno de simulación. Se compara la estimación de la posición con respecto a la posición adquirida mediante segmentación por color.

Palabras clave: Filtro de Kalman extendido, localización, ROS, Gazebo.

Mobile Robot Localization Using the Extended Kalman Filter and Gazebo Simulator

Abstract. Locating a mobile robot allows performing path planning tasks. Although there are dynamic models for different types of mobile robots, the non-linear dynamic effects on the environment, model, sensors, and mechanics prevent an exact knowledge of the robot's position. Therefore, it is necessary to estimate the position considering probabilistic effects. The Extended Kalman Filter (EKF) allows the estimation of the pose of a mobile robot considering deviations from the non-linear dynamic model. The implementation of the EKF for the location of a mobile robot on the Gazebo simulation platform is presented. The position of the robot is obtained through color segmentation, the image is acquired through a camera placed in the simulation environment. The position estimate is compared to the position acquired by color segmentation.

Keywords: Extended Kalman filter, localization, ROS, Gazebo.

1. Introducción

La evolución actual de la robótica se enfoca a la aplicación en ambientes no estructurados. Este tipo de ambientes se caracteriza por la presencia de incertidumbre que puede provenir de los siguientes factores: ambiente, sensores, robots, modelos y cómputo [10]. El manejo de la incertidumbre en robótica se realiza a través de la consideración de procesos estocásticos, utilizando los principios de la robótica probabilística.

La robótica probabilística se enfoca en la cuantificación de una distribución de probabilidad condicional conocida como creencia $Bel(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{z}_{1:k}, \mathbf{u}_{1:k})$, que mide la probabilidad de obtener el estado $\mathbf{x_k}$ en el tiempo k cuando se tiene el conocimiento de las variables medidas ($\mathbf{z}_{1:k}$) y las entradas aplicadas ($\mathbf{u}_{1:k}$), la notacion $1 : k$ toma encuenta todas las variables desde la muestra 1 hasta k. La estimación del estado se realiza al obtener la esperanza de $Bel(\mathbf{x}_k)$. La forma general para el cálculo de la creencia se realiza mediante el filtro recursivo de Bayes o filtro de Bayes [2]. El procedimiento para el cálculo de la creencia se basa en dos etapas: predicción y corrección. La etapa de predicción obtiene una estimación del estado a partir del comando de control utilizado y los datos de medición anteriores, la predicción de la estimación se denota como $\overline{Bel}(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{z}_{1:k-1}, \mathbf{u}_{1:k})$. Una vez realizada la predicción de estado, se realiza la corrección a partir de la medida del sensor en el tiempo t y la predicción anteriormente realizada.

La implementación del filtro de Bayes depende de la forma matemática que tengan las distribuciones de probabilidad utilizadas. Si se utiliza la distribución normal en su forma unimodal, entonces se obtiene el filtro de Kalman [10]. El filtro de Kalman toma al modelo en espacio de estados para estimar el estado y la salida, realizando un filtrado óptimo a la señal de salida [9], fue propuesto por Rudolf E. Kalman en 1960 [5] para sistemas discretos. Debido a que el filtro de Kalman tiene la restricción de ser utilizado únicamente para sistemas lineales, se ha propuesto el EKF como una extensión a sistemas no lineales.

Para el problema de localización de un robot móvil, el estado del sistema corresponde a la pose del robot. La entrada depende de la configuración del robot móvil, en este caso se tiene como comandos de entrada a la velocidad lineal y a la velocidad angular del robot. El modelo de movimiento propuesto es el modelo de movimiento de velocidad [10]. Como variable sensada se utiliza una cámara virtual, que mediante segmentación por color estimará la posición del robot. Se elige como simulador a Gazebo [7] debido a su compatibilidad directa con ROS, el cual es un framework popular para controlar robots.

2. Modelo de movimiento de velocidad

El modelo dinámico de un sistema describe el comportamiento de su salida con respecto a un tiempo t utilizando modelos matemáticos descritos como ecuaciones diferenciales.

Los sistemas dinámicos se caracterizan porque las entradas anteriores al tiempo t afectan a su salida, cuando la salida de un sistema sólo depende de

la entrada en el momento t se le conocen como sistemas estáticos. En ocasiones el tiempo t es sustituido por una variable de instante k ($t = k\Delta t$), esto sucede cuando el sistema se actualiza en periodos no continuos de instantes de tiempo Δt, a estos sistemas se les conoce como sistemas discretos. Un sistema dinámico no lineal en tiempo discreto se representa por la siguiente ecuación [10]:

$$\mathbf{x}_k = g_k(\mathbf{x}_{k-1}, \mathbf{u}_k) + \epsilon_k, \tag{1}$$

donde $\mathbf{x}_k$ representa el estado del sistema, $g_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ es el vector de transición del estado. Para la ecuación de medición del sistema se tiene:

$$z_k = h_k(\mathbf{x}_k) + \delta_k. \tag{2}$$

La variable $h_k(\mathbf{x}_k)$ representa la relación entre el vector de estado y las variables a medir, cuando las variables de medición coinciden con el estado del sistema se tiene $h_k(\mathbf{x}_k) = \mathbf{x}_k$. Los vectores ϵ_k, δ_k representan el ruido del sistema que depende de variables aleatorias. La matriz de covarianza de ϵ_k es R y la matriz de covarianza de δ_k es Q.

El modelo dinámico para el movimiento de un robot móvil asume como entrada de control al vector $\mathbf{u}_k = \begin{bmatrix} v_k & \omega_k \end{bmatrix}^T$ donde v_k representa la velocidad lineal y ω_k la velocidad angular en el instante k, así mismo, $g_k(\mathbf{x}_{k-1}, \mathbf{u}_k)$ es una ecuación de movimiento no lineal dependiente de la entrada $\mathbf{u}_k$ y del estado $\mathbf{x}_{k-1}$ que está conformado por la posición (x, y) y la orientación θ en el instante $k-1$.

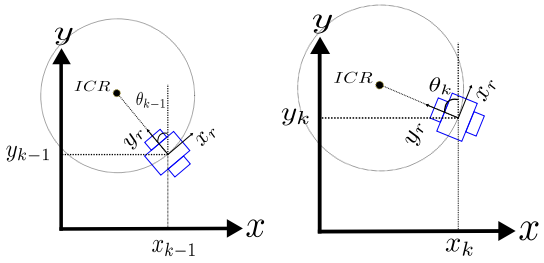
Se considera el movimiento del robot en el plano, por ello se utiliza el centro instantáneo de rotación (ICR) para obtener las ecuaciones de movimiento [8]. Se asume que la entrada $\mathbf{u}_k$ es ejecutada en el estado $\mathbf{x}_{k-1}$. Antes de ejecutar el movimiento (Figura 1(a)) el robot tiene como velocidad lineal y angular $\mathbf{u}_k = \begin{bmatrix} v_k & \omega_k \end{bmatrix}^T$, con los valores de velocidad se puede obtener el radio de curvatura mediante:

$$r = \frac{v_k}{\omega_k}. \tag{3}$$

Entonces se obtienen las coordenadas ICR como:

$$x_{icr} = x_{k-1} - \frac{v_k}{\omega_k} \sin(\theta_{k-1}), \tag{4}$$

$$y_{icr} = y_{k-1} + \frac{v_k}{\omega_k} \cos(\theta_{k-1}). \tag{5}$$



(a) Antes de moverse (b) Después de moverse

Fig. 1. Deducción de las ecuaciones de movimiento mediante el centro instantáneo de rotación.

Después de realizar el movimiento con velocidad $\mathbf{u}_k$ y un tiempo constante Δ_t la orientación del robot habrá cambiado a:

$$\theta_k = \theta_{k-1} + \omega_k \Delta_t. \tag{6}$$

Entonces el robot cambiará a su nueva posición espacial como muestra la Figura 1(b). Se puede determinar la nueva posición a partir de la posición del ICR debido a que este se conserva si $\mathbf{u}_k$ es constante. Bajo la consideración anterior se tiene:

$$x_k = x_{icr} + \frac{v_k}{\omega_k} \sin(\theta_k), \tag{7}$$

$$y_k = y_{icr} - \frac{v_k}{\omega_k} \cos(\theta_k). \tag{8}$$

Combinando las ecuaciónes (4) y (5) con (7),(8) y (6), se obtienen las ecuaciones de movimiento para un robot móvil con entradas (v_k, ω_k) mediante la ecuación (9) [10]:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \frac{v_k}{w_k} \left(\sin(\theta_{k-1} + \omega_k \Delta_t) - \sin(\theta_{k-1})\right) \\ y_{k-1} - \frac{v_k}{w_k} \left(\cos(\theta_{k-1} + \omega_k \Delta_t) - \cos(\theta_{k-1})\right) \\ \theta_{k-1} + \omega_k \Delta_t. \end{bmatrix} \tag{9}$$

La ecuación (9) representa las ecuaciones cinemáticas del robot móvil en tiempo discreto. Las variables sensadas del robot móvil. corresponden directamente a la pose del robot, esto es:

$$\mathbf{z}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}. \tag{10}$$

3. Filtro de kalman extendido

El EKF puede verse como una extensión al filtro de Kalman clásico que abarca sistemas dinámicos de tipo no lineal. La única diferencia entre el Filtro

de Kalman clásico y el filtro de Kalman extendido se basa en la linealización del sistema dinámico utilizando la matriz Jacobiana para obtener la matriz de transición de estado, entonces se puede utilizar el algoritmo del filtro de Kalman [6]. Para ello se calcula el Jacobiano de las funciones g_k y z_k utilizando las ecuaciones (9) y (10) respectivamente:

$$G_k = \frac{\partial g_k(\mathbf{x}_{k-1}, \mathbf{u}_k)}{\partial \mathbf{x}_{k-1}} = \begin{bmatrix} 1 & 0 & \frac{v_k}{w_k}\left(\cos(\theta_{k-1} + \omega_k \Delta_t) - \cos(\theta_{k-1})\right) \\ 0 & 1 & \frac{v_k}{w_k}\left(\sin(\theta_{k-1} + \omega_k \Delta_t) - \sin(\theta_{k-1})\right) \\ 0 & 0 & 1 \end{bmatrix}, \tag{11}$$

$$H_k = \frac{\partial h_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

Por simplicidad, se proponen a las matrices de covarianza como:

$$R = Q = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}. \tag{13}$$

Con las ecuaciones (9), (10), (11), (12) y (13) se implementa el algoritmo del filtro de Kalman extendido. Para la orientación, se extraen los datos de la unidad de medición inercial (IMU) del robot. Los algoritmos 1 y 2 muestran el procedimiento para el cálculo de la corrección y predicción respectivamente. En la fase de predicción las variables $\hat{x}_k^-$ y P_k^- corresponden a los valores de la media y matriz de covarianza de $Bel(\mathbf{x}_k)$. La etapa de predicción toma a consideración al sistema no lineal y actualiza la covarianza a través de el valor de covarianza del ruido inducido.

Algoritmo 1 Filtro de Kalman Extendido: Fase de Predicción.

Entrada: $\hat{x}_{k-1}$,u_k, P_{k-1}, R
Salida: $\hat{x}_k^-$, P_k^-
1: $\hat{x}_k^- = g(x_{k-1}, u_k)$
2: $P_k^- = G_k P_{k-1} G_k^T + R$

La fase de corrección toma las salidas de la fase de predicción, en conjunto con la medición y covarianza del ruido proveniente del sensor para calcular una ganancia de corrección K_k. Como elementos de salida se tiene a la media $\hat{x}_k^-$ y a la covarianza del sistema P_k.

Algoritmo 2 Filtro de Kalman Extendido: Fase de Corrección.

Entrada: $\hat{x}_k^-$, P_k^-, z_k, Q
Salida: $\hat{x}_k$, P_k
1: $K_k = P_k^- H_k^T (H_k P_k^- H_k^T + Q)^{-1}$
2: $\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-))$
3: $P_k = (I - K_k H_k) P_k^-$

El Algoritmo 2 muestra el procedimiento de cálculo para la obtención de la corrección, es necesario realizar la operación de matriz inversa. La implementación de los algoritmos se realiza en C++ utilizando la biblioteca de matrices Eigen [4].

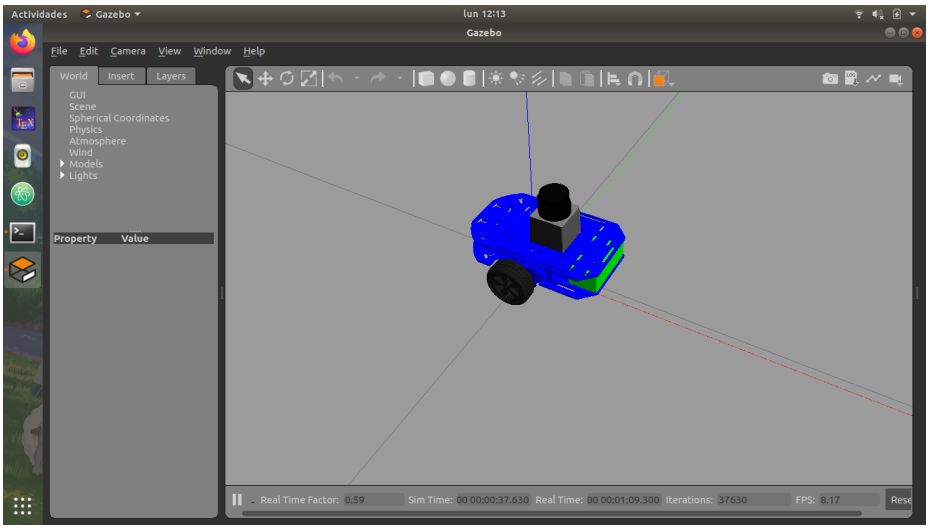


Fig. 2. Simulación de robot móvil.

4. Gazebo

Gazebo es un simulador multirobot para ambientes externos diseñado por Open-Source Robotics Foundation. En [3] se citan las principales características del simulador Gazebo obtenidas de una encuesta a diferentes personas que trabajan en el área de la robótica, de las cuales se citan algunas a continuación:

- Soporta múltiples motores de física. (*Open Dynamics Engine*, Bullet).
- Sistema operativo: 100 % GNU/Linux.
- Interfaz de programación de aplicaciones (API) Principal: 80 % C++.

- Razones de uso: mejor herramienta de evaluación, software listo para usar en el laboratorio y es de código abierto.
- Uso principal: en robótica móvil, en robótica de servicios, en robots humanoides.
- Middleware utilizado: 93 % ROS.

Las razones anteriores influyeron en la decisión de utilizar el simulador Gazebo. Para realizar la simulación se optó por un modelo de robot móvil de tipo diferencial. Este robot se puede controlar mediante los comandos de velocidad lineal y angular, además cuenta con una IMU y con cámaras para poder visualizar [1]. Para identificar la posición del robot se utiliza segmentación por color y se extrae la posición a través del centroide de la imagen. La compatibilidad de Gazebo con ROS permite implementar algoritmos que pueden usarse en un robot con alteraciones mínimas en el código.

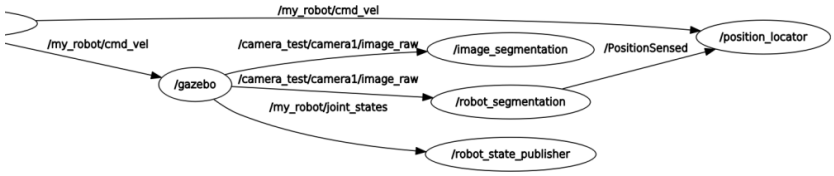


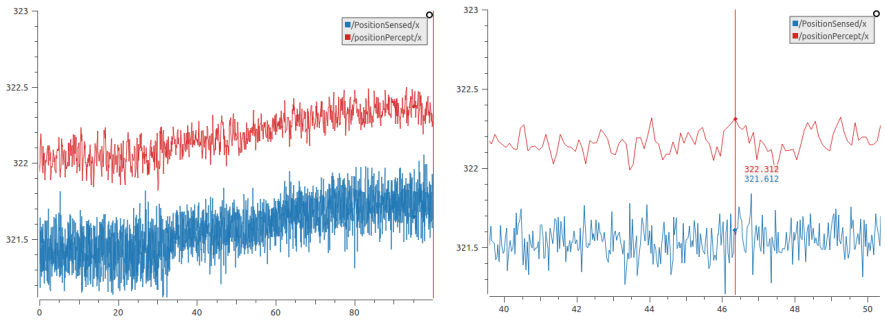
Fig. 3. Nodos y tópicos utilizados en la simulación.

Para realizar el experimento, el robot móvil recibe una velocidad lineal y angular constante, esto para generar una trayectoria de tipo circular. Entonces, mediante el comando *rqt_plot* se grafican los valores obtenidos del simulador y del filtro de Kalman extendido. La Figura 3 muestra los nodos y tópicos implementados para implementar el EKF, el comando de velocidad es enviado a la simulación de Gazebo (mediante el tópico *my_robot/cmd_vel*), donde se utiliza la cámara ubicada arriba del escenario y se publican el tópico de la imagen *camera_test/camera1/image_raw*. El nodo */my_robot/segmentation* obtiene la posición del robot que es enviada a */position_locator* donde se utiliza el algoritmo de EKF.

5. Resultados

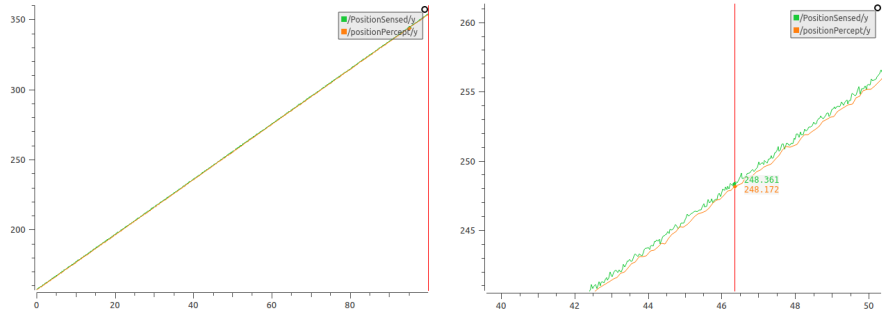
Se realizaron dos pruebas en el simulador Gazebo: movimiento en linea recta, movimiento alrededor de un círculo. Para el movimiento en línea recta, el robot esta orientado al eje y. Las líneas azul y verde corresponden al valor obtenido a partir de la segmentación. La Figura 4(a) muestra los valores adquiridos, mientras que 4(b) es un acercamiento de la imagen, se observa un desfase máximo de un milímetro, y menores oscilaciones de los valores. Se obtiene un resultado similar para la coordenada y (Figura 5), donde se observa un comportamiento

lineal. Los valores introducidos para el movimiento lineal son $\mathbf{u}_k = \begin{bmatrix} 10 & 0 \end{bmatrix}^T$, con v_k en mm/s.



(a) x_k (b) x_k

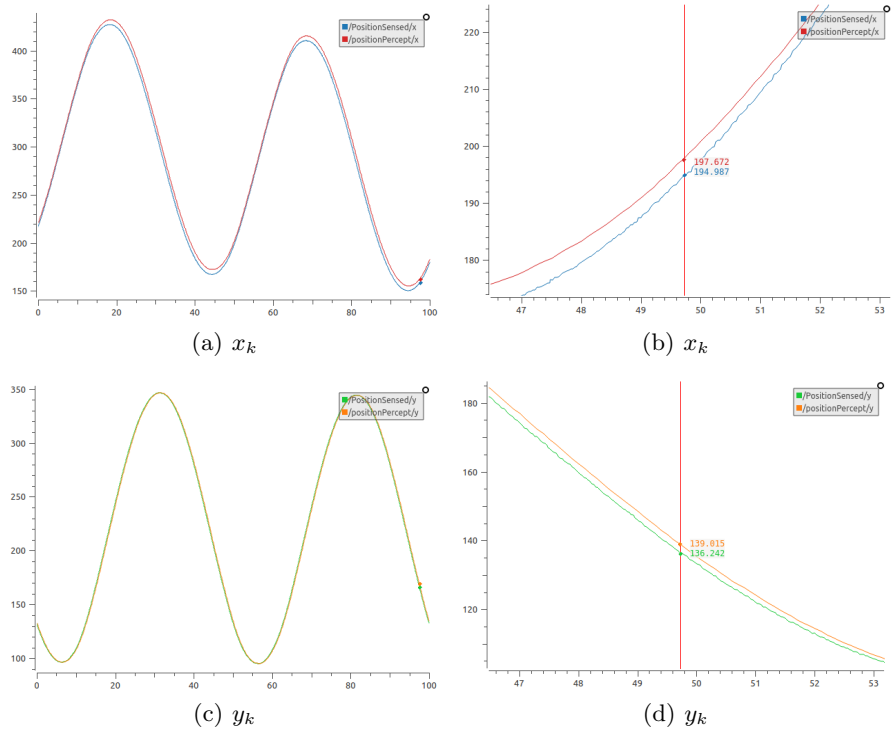
Fig. 4. Resultados del movimiento lineal, posición en el eje x.



(a) y_k (b) y_k

Fig. 5. Resultados del movimiento lineal, posición en el eje y.

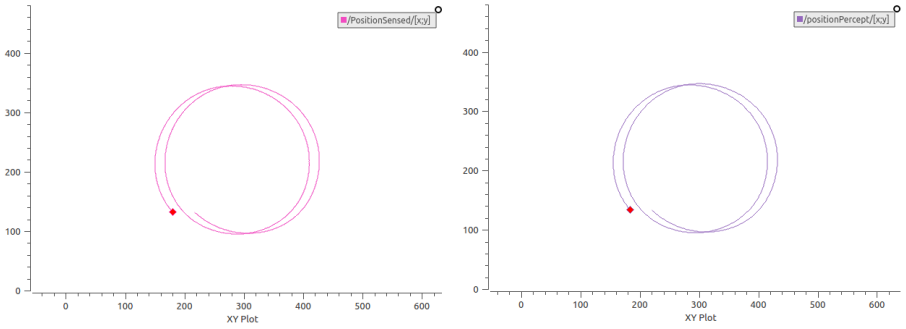
Los resultados del movimiento en una trayectoria circular se muestran en la Figura 6. Las gráficas de color azul y verde corresponden a la medida adquirida mediante los sensores, mientras que las líneas roja y naranja representan el valor al aplicar el filtro. Similar al resultado anterior, se observa una desviación no mayor a 5 milímetros y disminución del ruido al utilizar el filtro. En la Figura 7(a) se puede observar el movimiento realizado en el plano xy mientras que 7(b) se observa el resultado del filtro.



(a) x_k (b) x_k

(c) y_k (d) y_k

Fig. 6. Resultados de la aplicación del filtro de Kalman Extendido, movimiento en forma de circular.



(a) Movimiento sensado (b) Movimiento predecido

Fig. 7. Gráfica XY del movimiento en forma circular.

6. Conclusiones

La implementación del EKF mejora significativamente el sistema de localización del robot, al proveer una medida estable de la posición del robot con la

que se puede en un futuro trabajar en conjunto con algoritmos de navegación más complejos. El filtro de Kalman extendido programado en C++ y utilizando el framework de ROS obtuvo una respuesta en el tiempo de $1ms$, por lo que lo convierte en una opción eficiente para la implementación del filtro de Bayes. Utilizar Gazebo permite la migración del mismo código a una aplicación real.

Referencias

1. Coleman, D.: Gazebo ROS demos (2019), https://github.com/ros-simulation/gazebo_ros_demos
2. Dardari, D., Falletti, E., Luise, M.: Satellite and Terrestrial Radio Positioning Techniques: A Signal Processing Perspective (2012)
3. Ivaldi, S., Padois, V., Nori, F.: Tools for dynamics simulation of robots: a survey based on user feedback pp. 1–15 (2014), http://arxiv.org/abs/1402.7050
4. Jacob, B., Guennebaud, G.: Eigen (aug 2019), http://eigen.tuxfamily.org/index.php?title=Main_Page
5. Kalman, R.E.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82(1), 35–45 (1960)
6. Keatmanee, C., Baber, J., Bakhtyar, M.: Simple Example of Applying Extended Kalman Filter Simple Example of Applying Extended Kalman Filter. 1st International Electrical Engineering Congress (March) (2014)
7. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 3, 2149–2154 (2004)
8. Peacock, T.: Kinematics of Rigid Bodies Instant Centers or Instantaneous Centers. Dynamics and Control pp. 1–6 (2007)
9. Superiores, E., Graduados, P.: Aplicación de los Filtros de Kalman a Sistemas de Control. Ingenieria UC (January 2001), 2403 (2001)
10. Thrun, S.: Probabilistic robotics, vol. 45. Association for Computing Machinery, New Yor, NY, USA, 1 edn. (2002)

ISSN 1870-4069

Algoritmo de clasificación basado en la función Softmax

Arturo Zúñiga-López, Carlos Avilés-Cruz,
Andrés Ferreyra-Ramírez, Eduardo Rodríguez-Martínez

Universidad Autónoma Metropolitana,
Departamento de Electrónica,
División de Ciencias Básicas e Ingeniería,
México

{azl,caviles,fra,erm}@azc.uam.mx

Resumen. El análisis de datos en los campos de *ciencia de datos* y la *minería de datos* involucra la agrupación y clasificación de información que posee propiedades comunes. Existe gran cantidad de algoritmos para clasificación, sin embargo, uno solo no es capaz de clasificar todo tipo de distribuciones y densidades. En el presente trabajo se introduce un algoritmo de clasificación basado en la función softmax implementada en redes neuronales artificiales. El algoritmo propuesto concatena un número $k-$funciones *softmax* para llevar a cabo clasificaciones de datos linealmente no separable. Se presentan resultados usando bases de datos tipo benchmark, en donde se muestra casos "simples" de clasificación, i.e., linealmente separables, así como casos "complejos", i.e., casos no linealmente separables. Se presenta el cambio del espacio de partición de las funciones *softmax* en función del número de iteraciones.

Palabras clave: Función Softmax, agrupamiento, clasificación, atributos, bases de datos sintéticas.

Classification Algorithm Based on the Softmax Function

Abstract. Data analysis in *data science* and *data mining* involves the clustering and classification of information with common properties. There are many algorithms for classification; however, no one of them is capable of classifying all types of distributions and densities. In the present work, a classification algorithm based on the *softmax* function implemented in artificial neural networks is introduced. The proposed algorithm concatenates many $k-$softmax functions to carry out linearly non-separable data classifications. Results are presented using benchmark databases, where simple classification cases are shown, i.e., linearly separable, as well as more complex cases, i.e., non-linearly separable cases. The change of the partition space of the *softmax* functions is presented according to the number of iterations.

Keywords: Softmax function, clustering, classification, features, synthetic databases.

1. Introducción

El creciente interés por agrupar y clasificar información es debido a la gran cantidad de información que se maneja en las redes sociales y en Internet en general. La ardua tarea de agrupar y clasificar información -atributos- que poseen o comparten características comunes, es una tarea angular en las áreas de, *ciencia de datos*, *minería de datos* y *reconocimiento de patrones*. A pesar de la gran cantidad de algoritmos tanto de agrupamiento como de clasificación, aún no se tienen un solo algoritmo que funcionen para agrupar y clasificar todo tipo de datos.

Los algoritmos de agrupamiento-clasificación se clasifican de acuerdo con su criterio de agrupamiento, y también de acuerdo con la cantidad de datos que pueden manejar (dimensiones y extensión). Existen diferentes clasificaciones sobre los algoritmos de agrupamiento-clasificación, por ejemplo, Dong et al. [8] los clasifican en *supervisados* y *no supervisados*. Una clasificación más extensa es la hecha por Zhao et al. [23] en donde los autores presentan las clasificaciones siguientes: *Particional* [10,9,13], *Jerárquica* [22,17,4], *Basada en densidad* [12,3,14], *Basada en rejilla* [6,5,21], *Espectral* [20,15,16], *Gravitacional* [19,7,2], y *basada en redes neuronales* [1,18,11] . La presente propuesta pertenece al grupo de los *Particionales*, es decir, que particiona o divide es espacio de los atributos.

El algoritmo propuesto particiona el espacio de los atributos de acuerdo con funciones *Softmax* (funciones de tipo exponencial normalizadas, se explica en capítulo 2). Se utilizó redes neuronales artificiales para su diseño y se implementó en lenguaje de programación Python 3.7 con todos los módulos necesarios. La metodología propuesta fue evaluada usando las bases bi-dimensionales *Shape* ampliamente usadas en la bibliografía de agrupamiento-clasificación. La base de datos *Shape* se tomó del grupo *Machine Learning* de la Universidad del Este de Finlandia [1]. Esta base se tomó porque contiene distribuciones simples (linealmente separables, R15, D31). Distribuciones más complejas en donde clases están embebidas en otras (Jain, flame, and compound). Por último, distribuciones muy complejas en forma de espiral con clases embebidas (spiral and pathbased).

El resto de este documento esta organizado de la siguiente manera. La sección 2 introduce la metodología propuesta. La sección 3 introduce los conjuntos de datos utilizados y el protocolo de evaluación. La sección 4 muestra los resultados experimentales. Finalmente, la sección 5 discute las limitaciones y conclusiones de este trabajo.

[1] http://cs.joensuu.fi/sipu/datasets/

2. Propuesta

La presente propuesta está basada en el clasificador *Softmax*, cuya función de probabilidad está en términos de funciones exponenciales normalizadas. Sea $X \in \Re^d$ un punto muestra en el espacio d y contando con N puntos muestra.

Sea c la etiqueta de la clase correspondiente, con $c = 1, c = 2, c = 3..., c = K$ y K es el número total de clases. La ecuación 1 representa el cálculo de las probabilidades para $K-$clases:

$$\Psi(X, W) = \begin{bmatrix} p(c=1|X) \\ p(c=2|X) \\ p(c=3|X) \\ \vdots \\ p(c=K|X) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} e^{W_j^T \cdot X}} \begin{bmatrix} e^{(W_1^T \cdot X)} \\ e^{(W_2^T \cdot X)} \\ e^{(W_3^T \cdot X)} \\ \vdots \\ e^{(W_K^T \cdot X)} \end{bmatrix}. \tag{1}$$

Siendo W el vector de pesos, con $W \in \Re^d$.

Por otro lado, Si consideramos a la entropía cruzada H como métrica para medir la precisión entre *modelos pronósticos probabilísticos* y los valores reales, la ecuación 2 representa la entropía cruzada:

$$H(p, q) = -\sum_{X} p(X) log[q(X)]. \tag{2}$$

Siendo p la probabilidad verdadera y q la probabilidad obtenida (o predicha) a partir de un modelo, en éste caso será $\Psi(X, W)$ (ver ecuación 3):

$$H(p, q) = -\sum_{X} p(X) log[\Psi(X, W)]. \tag{3}$$

Cabe remarcar que, en realidad H es la función que observa el error, de ahí que vamos a definir a la función de costo por medio de la ecuación 4. Observamos que en la función de costo está en término de los vectores de peso W. Es decir, la optimización versará en encontrar los vectores de peso W que mejor separen las clases:

$$J(W) = \frac{1}{N} \sum_{i=1}^{N} H(p, q), \tag{4}$$

donde N es el número de observaciones.

Ahora la función $J(W)$ se puede reescribir de la siguiente manera:

$$J(W) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{K} p(c) log(q(c)). \tag{5}$$

Si consideramos que las clases c (o etiquetas) las podemos expresar como 'one hot' por ejemplo $c_1 = [1, 0, 0, ...]$ para la clase 1, $c_2 = [0, 1, 0, ...]$ para la clase 2 y así para K clases. La ecuación 5 se puede escribir como:

$$J(W) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{K} \delta(y_i) \cdot log \left[\frac{e^{(W_c^T \cdot X_i)}}{\sum_{j=1}^{K} e^{(W_j^T \cdot X_i)}} \right], \tag{6}$$

donde $\delta(y_i)$ es una delta de Dirac y vale 1 para una c clase dada.

Para el cálculo de actualización de pesos W, se determinará de la siguiente forma (ver ecuación 7):

$$W_j = W_j - \alpha \nabla_{W_j} J(W), \tag{7}$$

donde W_j es un vector de pesos de la matriz de pesos $W \in \Re^{d \times K}$, α es tasa de aprendizaje y $\nabla_{W_j} J(W)$ es el gradiente del costo para esa vector j.

Por último, el gradiente descendente se calcula de la expresión 8 siguiente:

$$\nabla_{W_j} J(W) = -\frac{1}{N} \sum_{i=1}^{N} [\delta(c_i) - \frac{e^{(W_c^T \cdot X_i)}}{\sum_{j=1}^{K} e^{(W_j^T \cdot X_i)}}] X_i. \tag{8}$$

Finalmente, remarcamos que los espacios de partición delimitado por el clasificador propuesto serán linealmente separables, pudiendo ser linealmente separable a trozos.

3. Bases de datos

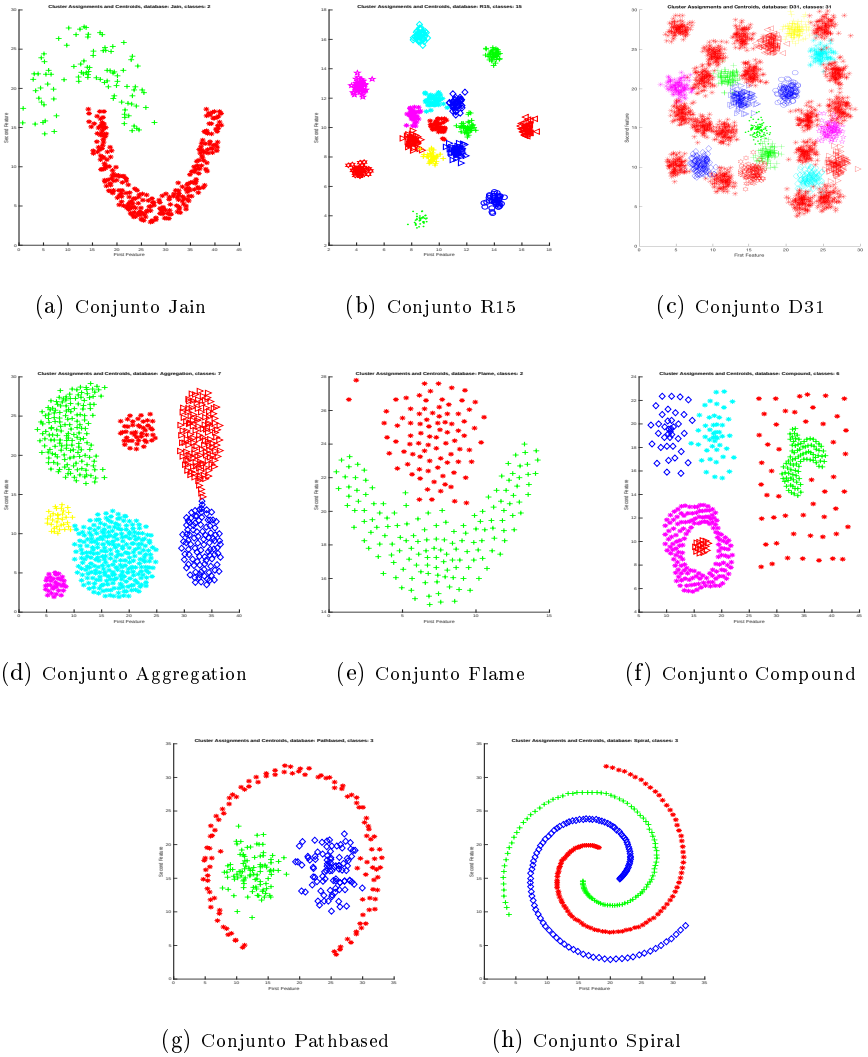
Se utilizaron bases de atributos bi-dimensionales tomados del grupo de aprendizaje de máquina de la Universidad del Este de Finlandia [2]. Se eligió la base de atributos *Shape* por sus características desde las simples, distribuciones esféricas separadas, distribuciones embebidas y aún más complejas, en forma de espiral.

La Fig. 1 muestra los 8 conjuntos de datos, así mismo, la Tabla 1 muestra las características de cada conjunto. Los conjuntos de datos son : *Aggregation* (Fig. 1(a)), *Compound* (Fig. 1(b)), *Pathbase* (Fig. 1(c)), *Spiral* (Fig. 1(d)), *D31* (Fig. 1(e)), *R15* (Fig. 1(f)), *Jain* (Fig. 1(g)), y *Flame* (Fig. 1(h)). El conjunto de bases de atributos contiene desde 2 clases hasta 31. El mínimo número de atributos es de 240 y el máximo de 3,100.

Tabla 1. Bases de datos sintéticas *Shape* a clasificar.

Conjunto	No. clases	No. muestras
Aggregation	7	788
Compound	6	399
Pathbase	3	300
Spiral	3	312
D31	31	3100
R15	15	600
Jain	2	373
Flame	2	240

[2] http://cs.joensuu.fi/sipu/datasets/



(a) Conjunto Jain (b) Conjunto R15 (c) Conjunto D31

(d) Conjunto Aggregation (e) Conjunto Flame (f) Conjunto Compound

(g) Conjunto Pathbased (h) Conjunto Spiral

Fig. 1. Base de datos *Shape* original.

4. Metodología

El algoritmo de clasificación propuesto fue implementado en una computadora Apple-MAC i27" [3], se utilizó Python 3.7 como lenguaje de simulación con los módulos requeridos tanto de redes neuronales como de graficación.

[3] 3.1GHz quad-core Intel Core i5 con 6MB on-chip shared L3 cache, 16GB de memoria, procesador gráfico AMD Radeon HD 6970M con 1GB de memoria.

Función Principal(x,y, iteraciones)
Comentario: x atributos y y etiquetas de archivo txt
inicio
  X ← matrix 'x' + una columna de $1's$
  Y ← transformación de las etiquetas 'y' a la representación "one-hot"
  k ← número de clases
  n ← número de atributos
  W ← valores aleatorios de W de tamaño $[n + 1 \times k]$
  Z ← hipotesis(X,W)
  mientras $m <$ *iteraciones* **hacer**
    W=gradiente(X,Y,Z,n,k)
    L ← costo(Z,Y)
    Z ← hipotesis(X,W)
  fin
  ypred ← max_por_fila_de$(W * X)$
fin

Algoritmo 1: Función Principal.

Función hipotesis(X,W)
Comentario: X es la matriz de atributos y W es la matriz de pesos
z ← X*W
s ← matriz de ceros de tamaño z
mientras $j <$ *números de observaciones* **hacer**
  Z ← z[j fila]
  s[j fila] ← exp(Z)/suma(exp(Z))
fin
Regresar s

Algoritmo 2: Función hipótesis.

Función costo(Z,Y)
Comentario: Z es la matriz de probabilidades y Y es la matriz de etiquetas
suma ← 0
mientras $j <$ *números de observaciones* **hacer**
  Z ← z[j fila]
  J ← producto_punto(Z[j fila],Y[j fila])
  J ← log(J)
  suma ← suma+J
fin
Regresar suma$*(-1/$número de observaciones$)$

Algoritmo 3: Función costo.

El seudocódigo del programa principal se muestra en el Algoritmo 1 y las funciones invocadas son tres: *Función de hipótesis* (ver Algoritmo 2), *Función de costo* (ver Algoritmo 3) y la *Función de gradiente* (ver Algoritmo 4). Se usó la base de datos *Shape* descrita en el capítulo anterior, la cual cuanta con 8 tipo de distribuciones-densidades.

Función gradiente(X,Y,Z,n,k)
Comentario: Z es la matriz de probabilidades y Y es la matriz de etiquetas
mientras $j <$ *números de observaciones* **hacer**
  suma ← matriz de 1's de tamaño 1x(n+1)
  alfa ← 0.01
  mientras $i <$ *números de observaciones* **hacer**
    suma ← suma + (Y[i,j]-Z[i,j])*X[i fila]
  fin
  W[j columna] ← W[j columna] - (-alfa/número de observaciones)∗suma
fin
Regresar W

Algoritmo 4: Función gradiente.

Dado que el algoritmo es iterativo, se tomó como criterio de *paro* el error, es decir, cuando el error encuentra su valor mínimo, para cada una de las bases de datos el número de iteraciones es variable.

5. Resultados experimentales

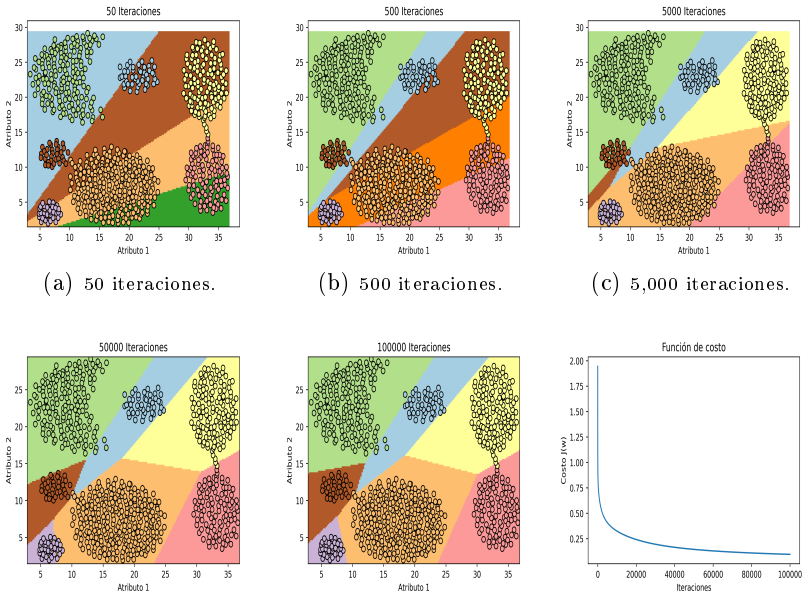
Se ha llevado a cabo la segmentación-clasificación con el algoritmo propuesto usando la base de atributos bi-dimensional *shape*. Los ocho conjuntos sintéticos fueron clasificados dando una medida del error (ϵ) para cada grupo. La ecuación 9 permite expresar un cociente entre el número de puntos clasificados incorrectamente, con respecto al número total de puntos de cada clase:

$$\epsilon = \frac{\text{Clasificación incorrecta}}{\text{Total elementos por clase}}. \tag{9}$$

5.1. Evolución de clasificación

A manera de ejemplo, presentamos la evolución del espacio de partición en dos dimensiones, para el conjunto de datos *Aggregation*. La base está constituida de 7 clases con 788 puntos muestra por clase. En la Fig. 2 podemos apreciar la evolución para 50, 500, 5,000, 50,000 y 100,000 iteraciones, respectivamente. Así mismo, la Fig. 2(f) presenta la evolución del error en función del número de iteraciones.

Como se puede apreciar en la Fig. 2(e), el espacio bidimensional pudo ser dividido en 7 zonas independientes, zonas: verde, azul, amarilla, naranja, café, magenta y rosa). Cada zona logró abarcada los puntos correspondientes a cada una de las clases de forma exitosa, obteniendo una precisión del 98.98 % (ver primera línea de la Tabla 2). Cabe recordar que el sistema de clasificación está basado en la función *Softmax*, la cual trabaja eficientemente en espacios linealmente separables.



(a) 50 iteraciones. (b) 500 iteraciones. (c) 5,000 iteraciones.

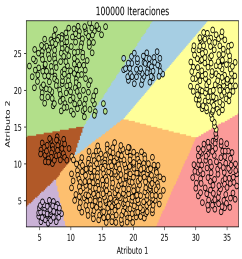
(d) 50,000 iteraciones. (e) 1000,000 iteraciones. (f) Error en aprendizaje

Fig. 2. Evolución del espacio de división del clasificador Softmax sobre la base de datos *Aggregation*.

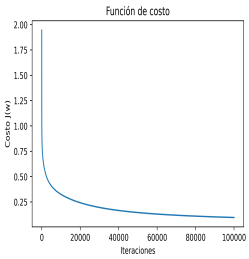
5.2. Evolución de desempeño sobre la base "Shape"

Las ocho bases de atributos de la base *Shape*, fueron clasificadas por el sistema de clasificación propuesto. Se analizaron tres componentes: el error de aprendizaje, el espacio bi-dimensional de partición y la matriz de confusión. Las Figs. 3 y 4 muestran los resultados para las 8 bases de atributos. Los resultados cuantificables del desempeño se muestran en la Tabla 2. El performance se obtuvo como el promedio de la diagonal principal de la matriz de confusión. Las bases de atributos se agruparon en tres categorías, de acuerdo con su forma de distribución y su complejidad en: *"Fáciles"*, *"Medios"* y *"Complejos"*.

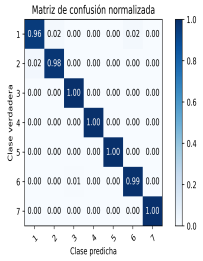
- **Fáciles:** El sistema propuesto funciona bien sobre los conjuntos *Aggregation*, *R15* y *Jain*, con desempeños de 98.98 %, 94.50 % y 90.08 %, respectivamente (ver Tabla 2).
- **Medios:** Sobre las bases *Flame*, *Compound* y *R31*, los resultados son buenos con desempeños promedio de 88.36 %, 77.94 % y 76.87 % y 65.66 %, respectivamente.
- **Complejos:** Para éste tercer grupo, el más complejo, los resultados con la metodología propuesta, son los siguiente: Para *Pathbased* y *Spiral* los resultados de desempeño son 65.66 % y 32.69 %, respectivamente. Los valores de desempeño son muy bajos debido a que son distribuciones linealmente no separables.



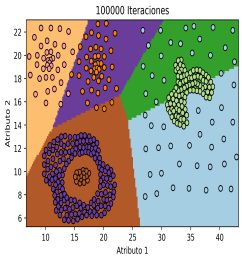
(a) Conjunto Aggregation: resultado de clasificación



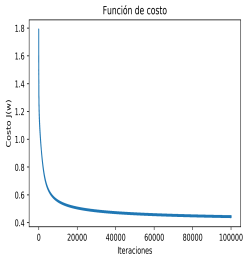
(b) Conjunto Aggregation: error en aprendizaje



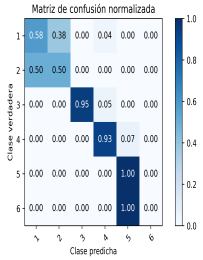
(c) Conjunto Aggregation: matriz de confusión



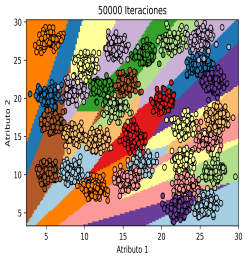
(d) Conjunto Compound: resultado de clasificación



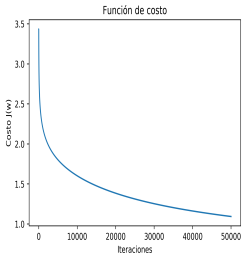
(e) Conjunto Compound: error en aprendizaje



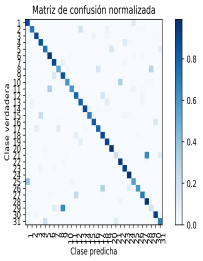
(f) Conjunto Compound: matriz de confusión



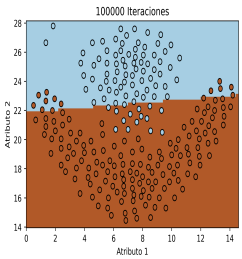
(g) Conjunto D31: resultado de clasificación



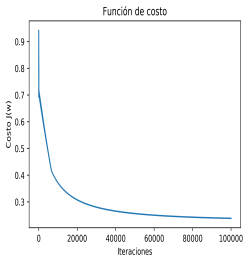
(h) Conjunto D31: error en aprendizaje



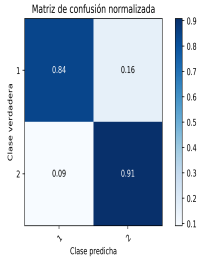
(i) Conjunto D31: matriz de confusión



(j) Conjunto Flame: resultado de clasificación

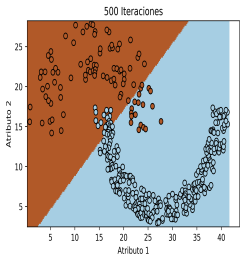


(k) Conjunto Flame: error en aprendizaje

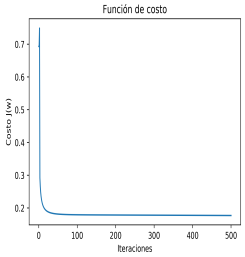


(l) Conjunto Flame: matriz de confusión

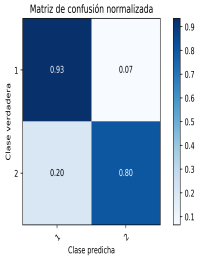
Fig. 3. Clasificación del sub conjunto "simple" de la base de datos *Shape*.



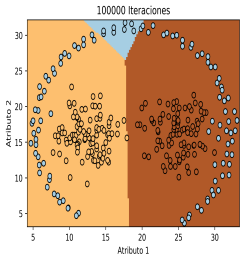
(a) Conjunto Jain: resultado de clasificación



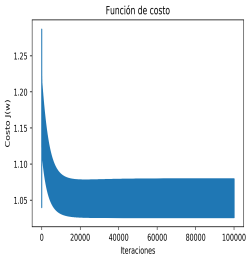
(b) Conjunto Jain: error en aprendizaje



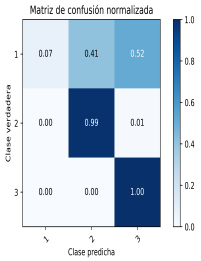
(c) Conjunto Jain: matriz de confusión



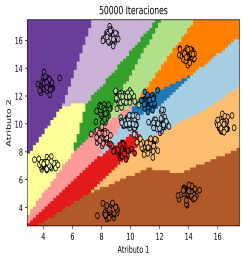
(d) Conjunto Pathbased: resultado de clasificación



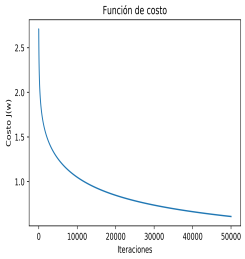
(e) Conjunto Pathbased: error en aprendizaje



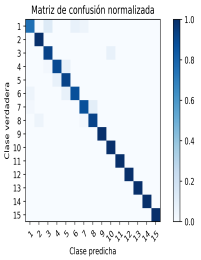
(f) Conjunto Pathbased: matriz de confusión



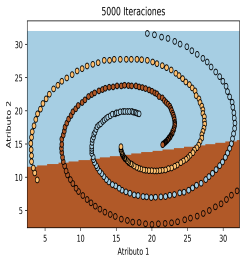
(g) Conjunto R15: resultado de clasificación



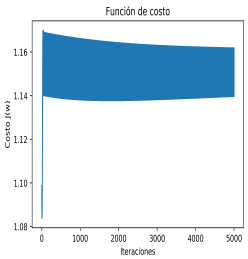
(h) Conjunto R15: error en aprendizaje



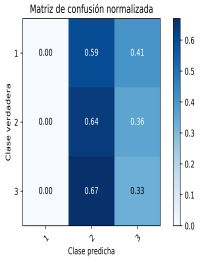
(i) Conjunto R15: matriz de confusión



(j) Conjunto Spiral: resultado de clasificación



(k) Conjunto Spiral: error en aprendizaje



(l) Conjunto Spiral: matriz de confusión

Fig. 4. Clasificación del sub conjunto "complejo" de la base de datos *Shape*.

Tabla 2. Clasificación de las 8 bases de datos sintéticas *Shape*.

Conjunto	Precisión promedio
Aggregation	98.98 %
Compound	77.94 %
Pathbase	65.66 %
Spiral	32.69 %
D31	76.87 %
R15	94.50 %
Jain	90.08 %
Flame	88.36 %

6. Conclusiones

Se ha podido diseñar e implementar un clasificador basado en la función *Softmax* de manera exitosa. El sistema propuesto es capaz de desplegar n clasificadores *Softmax*, que como es bien sabido, es un clasificador lineal. A partir de subdivisiones del espacio de clasificación se logró, linealmente clasificar la base de datos *Shape*.

Las bases de atributos se agruparon en tres categorías, de acuerdo con su forma de distribución y su complejidad en: *Fáciles*, *Medios* y *Complejos*.

- **Fáciles: *Aggregation*, *R15* y *Jain***, para los cuales el sistema propuesto funciona excelentemente (desempeños entre el 90 % y 99 %)
- **Medios: *Flame*, *Compound* y *R31***, para este tipo de distribuciones los resultados son buenos, con desempeños entre 66 % y 88 %.
- **Complejos: *Pathbased* y *Spiral*** Para este tercer grupo, los desempeños son muy bajos (en los $30s$ %) debido a que son distribuciones linealmente no separables.

Finalmente, podemos afirmar que el sistema propuesta funciona muy bien cuando se trata de distribuciones linealmente separables.

Como trabajo futuro se probará el algoritmo propuesto a n dimensiones de los atributos (más de 2 dimensiones). Adicionalmente se trabajará con más bases de datos utilizadas (*benchmark*) en el campo de clasificación de atributos.

Referencias

1. Abavisani, M., Patel, V.: Deep multimodal subspace clustering networks. IEEE Journal on Selected Topics in Signal Processing 12(6), 1601–1614 (2018)
2. Alswaitti, M., Ishak, M., Isa, N.: Optimized gravitational-based data clustering algorithm. Engineering Applications of Artificial Intelligence 73, 126–148 (2018)
3. Behzadi, S., Ibrahim, M.A., Plant, C.: Parameter free mixed-type density-based clustering. In: Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., Wagner, R.R. (eds.) Database and Expert Systems Applications. pp. 19–34. Springer International Publishing, Cham (2018)

4. Cao, X., Su, T., Wang, P., Wang, G., Lv, Z., Li, X.: An optimized chameleon algorithm based on local features. In: Proceedings of the 2018 10th International Conference on Machine Learning and Computing. pp. 184–192. ICMLC 2018, ACM, New York, NY, USA (2018)
5. Chen, J., Lin, X., Xuan, Q., Xiang, Y.: Fgch: a fast and grid based clustering algorithm for hybrid data stream. Applied Intelligence 49(4), 1228–1244 (Apr 2019)
6. Deng, C., Song, J., Sun, R., Cai, S., Shi, Y.: Gridwave: a grid-based clustering algorithm for market transaction data based on spatial-temporal density-waves and synchronization. Multimedia Tools and Applications 77(22), 29623–29637 (Nov 2018)
7. Deng, Z., Qian, G., Chen, Z., Su, H.: Identifying tor anonymous traffic based on gravitational clustering analysis. In: 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). vol. 2, pp. 79–83 (Aug 2017)
8. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Frontiers of Computer Science 14(2), 241–258 (2020)
9. Gupta, T., Panda, S.P.: A comparison of k-means clustering algorithm and clara clustering algorithm on iris dataset. International Journal of Engineering & Technology 7(4), 4766–4768 (2019)
10. Kanika, Rani, K., Sangeeta, Preeti: Visual analytics for comparing the impact of outliers in k-means and k-medoids algorithm. In: 2019 Amity International Conference on Artificial Intelligence (AICAI). pp. 93–97 (Feb 2019)
11. Kimura, M.: Autoclustering: A feed-forward neural network based clustering algorithm. In: Proceedings of the IEEE International Conference on Data Mining Workshops, ICDMW. vol. 2018-November, pp. 659–666 (2019)
12. Kumar, K.M., Reddy, A.R.M.: A fast dbscan clustering algorithm by accelerating neighbor searching using groups method. Pattern Recognition 58, 39 – 48 (2016)
13. Li, Y., Cai, J., Yang, H., Zhang, J., Zhao, X.: A novel algorithm for initial cluster center selection. IEEE Access 7, 74683–74693 (2019)
14. Matioli, L.C., Santos, S., Kleina, M., Leite, E.A.: A new algorithm for clustering based on kernel density estimation. Journal of Applied Statistics 45(2), 347–366 (2018)
15. Nemade, V., Shastri, A., Ahuja, K., Tiwari, A.: Scaled and projected spectral clustering with vector quantization for handling big data. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 2174–2179 (Nov 2018)
16. Nemade, V., Shastri, A., Ahuja, K., Tiwari, A.: Scaled and projected spectral clustering with vector quantization for handling big data. pp. 2174–2179 (2019)
17. Pitolli, G., Aniello, L., Laurenza, G., Querzoni, L., Baldoni, R.: Malware family identification with birch clustering. In: 2017 International Carnahan Conference on Security Technology (ICCST). pp. 1–6 (Oct 2017)
18. Ren, Z., Chen, J., Ye, L., Wang, C., Liu, Y., Zhou, W.: Application of rbf neural network optimized based on k-means cluster algorithm in fault diagnosis. In: 2018 21st International Conference on Electrical Machines and Systems (ICEMS). pp. 2492–2496 (Oct 2018)
19. Sheshasaayee, A., Sridevi, D.: Fuzzy c-means algorithm with gravitational search algorithm in spatial data mining. In: 2016 International Conference on Inventive Computation Technologies (ICICT). vol. 1, pp. 1–5 (Aug 2016)
20. Tian, L., Du, Q., Kopriva, I., Younan, N.: Spatial-spectral based multi-view low-rank sparse sbuspace clustering for hyperspectral imagery. In: IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium. pp. 8488–8491 (July 2018)

21. Yang, Y., Zhu, Z.: A fast and efficient grid-based k-means++ clustering algorithm for large-scale datasets. In: Krömer, P., Zhang, H., Liang, Y., Pan, J.S. (eds.) Proceedings of the Fifth Euro-China Conference on Intelligent Data Analysis and Applications. pp. 508–515. Springer International Publishing, Cham (2019)
22. Yao, W., Dumitru, C., Loffeld, O., Datcu, M.: Semi-supervised hierarchical clustering for semantic sar image annotation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9(5), 1993–2008 (2016)
23. Zhao, Y., Tarus, S., Yang, L., Sun, J., Ge, Y., Wang, J.: Privacy-preserving clustering for big data in cyber-physical-social systems: Survey and perspectives. Information Sciences 515, 132–155 (2020)

ISSN 1870-4069

Inteligencia artificial en la prueba de software: una revisión sistemática de la literatura

Alfonso Robles-Aguilar[1], Angel Juan Sánchez-Garcia, Jorge Octavio Ocharán-Hernández

Facultad de Estadística e Informática,
Universidad Veracruzana,
México

alfonsoroag8@gmail.com, {angesanchez, jocharan}@uv.mx

Resumen. Este estudio presenta el método y el resultado de una revisión sistemática de la literatura para identificar las técnicas de Inteligencia Artificial (IA) usadas en la prueba de software (PS). Cincuenta y un estudios fueron seleccionados y analizados para encontrar las técnicas que han sido utilizadas para asistir actividades de PS e identificar nichos de oportunidad para investigaciones futuras. En estos estudios 18 actividades de la PS fueron abordados con 20 técnicas de IA. Como resultado se obtuvo que las técnicas de IA más utilizadas en la PS son la Inteligencia de Enjambre, los Algoritmos Genéticos y las Redes Neuronales Artificiales. Por otro lado, las actividades de PS más abordadas con IA fueron la generación de casos o datos de prueba y localización de fallas. Por último, los resultados muestran que la colaboración de estas dos disciplinas ha ido en aumento, lo que confirma que la relación de estas dos áreas es activa y prometedora.

Palabras clave: Inteligencia artificial, prueba de software, revisión sistemática de la literatura, calidad de software.

Artificial Intelligence in Software Testing: A Systematic Literature Review

Abstract. This paper presents the method and the result of a systematic literature review to identify the Artificial Intelligence (AI) techniques used in the Software Testing (ST). Fifty-one studies were selected and analyzed to find the techniques that have been used to assist ST activities and identify niches of opportunity for future research in this area. In these papers, 18 ST activities were approached with 20 AI techniques. As a result, it was obtained that the AI techniques most used in the ST are Swarm Intelligence, Genetic Algorithms, and Artificial Neural Networks. On the other hand, the activities of ST most approached with AI were test cases or test data generation and fault location. Finally, the results show that

the collaboration of these two disciplines has been increasing, confirming that the relationship between these two areas is active and promising.

Keywords: Artificial intelligence, software testing, systematic literature review, software quality.

1. Introducción

En años recientes, la Inteligencia Artificial (IA) ha ayudado a encontrar soluciones en problemas de diferentes disciplinas, desde ciencias como la arqueología [1] o la biología [2], hasta ingenierías como la eléctrica [3] o la ambiental [4]. El apoyo que ha representado la IA a muchas disciplinas se ha visto reflejado con mayor investigación y búsqueda de soluciones, llegando a campos como la Ingeniería de Software.

La Ingeniería de Software (IS) es una disciplina que comprende todos los aspectos de producción de software útil y de calidad [5]. Para lograr esto, se debe seguir un conjunto de actividades entre las que se encuentra la elicitación, el diseño, la construcción y la prueba de software (PS). En ocasiones, la prueba de software toma hasta el 50% de los recursos totales asignados para un proyecto [6]. La PS hace la diferencia entre un producto de calidad o uno sin garantías, pues el aseguramiento de la calidad se define en esta fase [7]. Esto se logra aplicando casos de prueba para la comprobación de la correcta funcionalidad del sistema y detectando fallas cuando existan casos de prueba exitosos.

La importancia de esta fase y la necesidad de optimizarla se ha reflejado en la búsqueda de soluciones a problemas de actividades en la PS. En esta búsqueda, algunas de esas soluciones se han encontrado en la IA [8]. El objetivo de esta Revisión Sistemática de la Literatura es tener un panorama general de las técnicas de la IA que se han aplicado en actividades de la PS y los resultados de su aplicación.

El trabajo está organizado de la siguiente manera: en la Sección 2 se muestra el trabajo relacionado. En la Sección 3 se presenta el método de investigación. En la Sección 4 se exponen los resultados obtenidos. Por último, en la Sección 5, se extraen conclusiones y se describe el trabajo futuro.

2. Trabajos relacionados

Para la realización de este trabajo no se encontraron revisiones se encontraron tres trabajos relacionados con el tema de la IA y la PS.

Sorte *et al.* [8] divulgaron una revisión del estado del arte del uso de IA en todo el ciclo de vida del desarrollo de software, en el cual abordaron a la PS pero con alcance muy acotado, ya que solamente cuenta con dos secciones que exponen el uso que se le ha dado a la IA en dos actividades de la PS.

Bunke *et al.* [9] publicaron un libro donde exponen algunos usos que se le ha dado a la IA en actividades de la PS, como por ejemplo: generación y reducción de sets de prueba, pruebas de caja negra y pruebas automatizadas de interfaces de usuario. Esta publicación muestra un panorama del uso de varias técnicas de IA en diferentes

actividades de PS, así como las tendencias que existían en este campo en los primeros años del siglo XXI. Al ser una publicación de 2004, es viable una actualización de la información con los resultados más recientes en el área.

Por útlimo, Hourani *et al.* [10] publicaron un artículo en el cual explican el impacto de la IA en la PS, donde llevaron a cabo una revisión de la literatura. En esta revisión encontraron el uso de técnicas de IA en PS como Algoritmos Genéticos, Redes Neuronales Artificiales, Árboles Aleatorios, entre otros. Una oportunidad de mejora de este trabajo es exponer la metodología que se usó para realizar la revisión de la literatura, y conocer el cómo se llegó a estos resultados.

3. Método de investigación

Esta investigación fue basada en la guía provista por Kitchenham y Charters [11] y se condujo en tres fases: la planeación, la ejecución de la RSL y el reporte. En estas tres fases se cubrió lo siguiente: definir términos de búsqueda, construir una cadena de búsqueda utilizando estos términos, establecer las fuentes de búsqueda y determinar los criterios para la selección de los estudios primarios. También, se definió el proceso para la extracción de datos y su análisis.

3.1. Planeación

Preguntas de investigación. En esta sección se formularon las preguntas de investigación (PI) que fueron de interés en esta RSL. A través de ellas se pretende exponer trabajo previo sobre técnicas de la IA en la PS, así como los resultados que se han obtenido al usarlas y encontrar nichos de oportunidad para futuras investigaciones en este campo. Las PI formuladas y su motivación se muestran en la Tabla 1.

Tabla 1. Preguntas de investigación.

Pregunta	Motivación
PI1.- ¿Cuáles son las técnicas de la IA que se han aplicado en la PS?	Conocer las técnicas de la IA que se han aplicado en la prueba de software.
PI2.- ¿Qué actividades de la PS han sido abordados con estas técnicas?	Identificar las actividades de la PS que se han intentado solucionar o mejorar con IA.
PI3.- ¿Cuáles han sido los resultados de las técnicas en su aplicación?	Conocer las ventajas y desventajas que representa el uso de estas técnicas.
PI4.- ¿Qué áreas de oportunidad o trabajo futuro se mencionan en las publicaciones?	Identificar áreas de investigación prometedoras en este ámbito.

Selección de fuentes y estrategia de búsqueda. Para responder las PI se identificaron los términos de búsqueda ubicados en la Tabla 2. Estos términos surgieron de las mismas PI. De la PI1 se ubicaron las palabras “técnicas” e “Inteligencia Artificial” como los conceptos más importantes de esta pregunta. En cuanto a la PI2, “Pruebas de Software” es la palabra que se consideró necesaria para responder esta pregunta. De los tres conceptos, se establecieron cuatro palabras clave de esta revisión.

Tabla 2. Términos de búsqueda.

Concepto	Palabras clave
Inteligencia Artificial	Artificial intelligence
Técnica	Technique, techniques
Prueba de software	Software testing

Tabla 3. Base de datos electrónicas seleccionadas.

Base de datos electrónica	Enlace
ScienceDirect	https://www.sciencedirect.com/
SpringerLink	https://link.springer.com/
IEEE Xplore	https://ieeexplore.ieee.org/
ACM Digital Library	https://dl.acm.org/

Tabla 4. Criterios de selección de estudios primarios

Criterios de inclusión	Criterios de exclusión
CI1.- Publicados entre el año 2015 y 2019.	CE1.- Publicado en un idioma diferente a inglés.
CI2.- Es un artículo de un journal o memoria de congreso.	CE2.- Es reporte técnico, poster, diapositivas, opinión, resumen de workshop o Keynote.
CI3.- En el resumen o palabras clave dan indicios de que el artículo contesta por lo menos una pregunta de investigación.	CE3.- Habla de pruebas a sistemas que usan Inteligencia Artificial pero no aplican técnicas de IA en el proceso de prueba
CI4.- Contesta al menos una pregunta de investigación.	CE4.- Artículos duplicados.

Tomando las palabras clave, se construyó la siguiente cadena de búsqueda:

("software testing") AND ("technique" OR "techniques") AND ("artificial intelligence").

A partir de los términos de búsqueda que se encuentran en la cadena, debido a que las dos más importantes (Artificial intelligence y Software testing) son del área de la Informática y la Computación, se optó como fuentes las bases de datos de la Tabla 3. La elección de estas fue debido a que tienen una gran colección de artículos de esta área, y para ser publicados en estas bases de datos, los artículos pasan por un proceso de arbitraje por expertos en el área.

Criterios de selección de estudios primarios. Para seleccionar los estudios a revisar, se eligieron las publicaciones que contaron con los cuatro criterios de inclusión (CI) y que no cumplieron con ningún criterio de exclusión (CE) de la Tabla 4.

Proceso de selección de estudios primarios. Para incluir a los estudios como primarios se llevó a cabo un proceso de selección en diferentes fases, en las cuales se aplicaron los CI y CE en el orden de la Fig. 1.

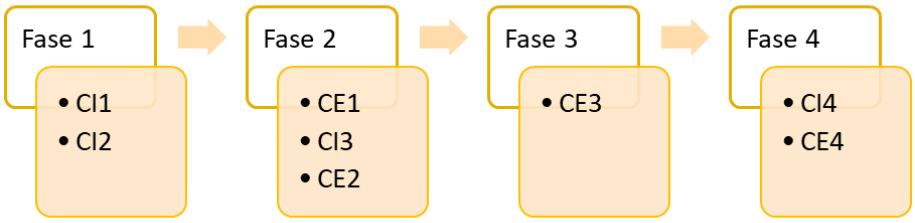


Fig. 1. Criterios de selección por fase.

Tabla 5. Formato para la extracción de datos.

Título	Autores	Año	Fuente	Palabras clave	Técnica de la IA usada	Actividad de PS	Resultados	Trabajo futuro

Estrategia de extracción de datos. Se definieron nueve datos de importancia para responder las PI de este estudio. Estos datos son mostrados en la Tabla 5.

Las primeras cinco características son para conocer las generalidades del artículo. Las siguientes cuatro son las secciones del estudio que contestan directamente cada una de las PI. Estos datos se extrajeron llenando una plantilla en una hoja de cálculo, donde cada columna correspondía a un dato de la Tabla 5 y cada fila era un artículo, y los espacios se llenaron con los fragmentos de texto que respondían a cada dato.

Método de síntesis de datos. Se siguió la Síntesis Temática propuesta por Cruzes y Dyba [12] para el análisis de datos cualitativos. Se optó por este método ya que se ha demostrado que es útil en revisiones con datos cualitativos del área.

3.2. Ejecución de la búsqueda

Para la selección de los estudios primarios, en la Fase 1 se aplicó la cadena de búsqueda en cada una de las bibliotecas digitales y se utilizó el filtro automático con la finalidad de que se mostraran artículos de *journals* y congresos publicados entre los años 2015 y 2019. En la Fase 2 se verificó que estuvieran escritos en idioma inglés, que no fueran diferentes a artículo de *journal* o congreso y se leyeron los resúmenes y palabras clave del artículo. Si estos dieron algún indicio de que pudieran contestar al menos una pregunta de investigación (no tenía que incluir un término de búsqueda obligatoriamente, ya que en algunos estudios nunca se menciona IA, pero si se menciona alguna técnica de IA) fueron incluidos para la siguiente fase. La Fase 3 consistió en eliminar a todos los estudios que expusieran sistemas que usaran IA pero que no se usó IA en la PS. La última fase fue la lectura de cada uno de los estudios, donde fueron incluidos como estudios primarios si contestaron al menos una pregunta de investigación y se excluyeron los artículos repetidos. La cantidad de estudios seleccionados por fase de cada una de las bibliotecas se muestra en la Tabla 6.

Tabla 6. Número de estudios por fase.

Biblioteca digital	Fase 1	Fase 2	Fase 3	Fase 4
SpringerLink	149	44	15	10
ScienceDirect	336	43	17	12
ACM Digital Library	146	43	14	10
IEEE Xplore	83	30	22	19
Totales	714	160	68	51

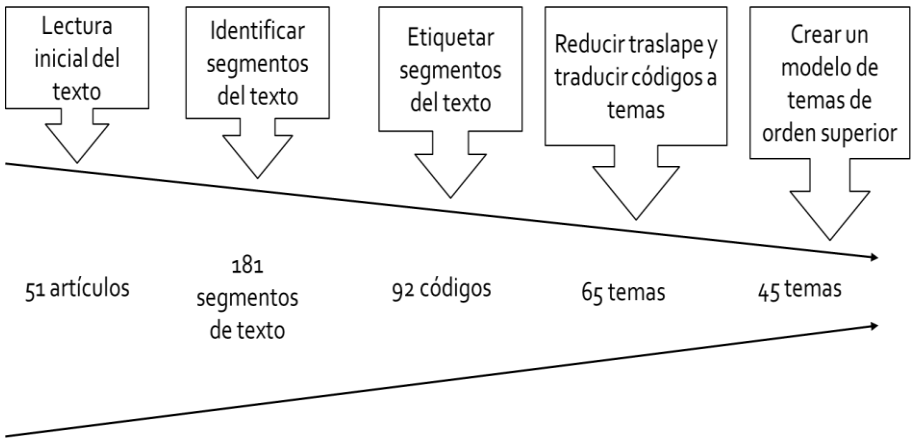


Fig. 2. Proceso de síntesis temática de datos.

4. Resultados

4.1. Resultados del análisis de datos

El análisis de datos cualitativos, se siguió el proceso que se muestra en la Fig. 2. La lectura inicial fue de 51 artículos considerados como Estudios Primarios. De ellos se extrajo 181 segmentos de texto, donde se encontraban respuestas a las PI. Después de codificar y tematizar los segmentos de texto, se obtuvieron 45 temas de orden superior repartidos de la siguiente manera: 20 temas en las técnicas de IA (las técnicas de IA se agruparon según su taxonomía), 18 temas en las actividades de PS (se determinó el tema de orden superior según a qué actividad exactamente atacaba la técnica de IA), 2 temas en resultados y 5 temas para trabajo futuro.

4.2. Respuestas de las preguntas de investigación

Para responder las PI, en los 51 artículos se encontraron 65 casos del uso de técnicas de IA en actividades de PS, ya que algunos artículos reportaban más de un caso.

Tabla 7. Técnicas de IA reportadas y frecuencia de aparición.

Técnica (Identificador)	Frecuencia	Referencias
Inteligencia de Enjambre (8)	18	[6, 13–23]
Algoritmos Genéticos (1)	15	[17, 22–34]
Redes Neuronales (4)	11	[17, 35–44]
Redes Bayesianas (2)	2	[28, 45]
Búsqueda de Monte Carlo (3)	2	[28, 46]
Bosques Aleatorios (7)	2	[47, 48]
K Vecinos Más Cercanos (12)	2	[49, 50]
Búsqueda Hiperheurística (5)	1	[51]
PRINCE (6)	1	[52]
Lógica Difusa (9)	1	[23]
Regresión de Huber (10)	1	[38]
Regresión de Vectores de Soporte (11)	1	[38]
LOUD (13)	1	[53]
Exploración Basada en *Q-Learining* (14)	1	[54]
Prueba por Comité (15)	1	[55]
Algoritmo de Agrupamiento (16)	1	[56]
Clasificador Bayesiano Ingenuo (17)	1	[57]
Sistema de Inferencia Difusa Mamdani (18)	1	[58]
Algoritmo K-medias (19)	1	[59]
Agentes (20)	1	[60]

En respuesta a la **PI1**, las técnicas de IA que se usan en la PS reportadas en los Estudios Primarios se muestran en la Tabla 7. Con respecto a la frecuencia de aparición, se encontró que la técnica de IA que más se ha utilizado en la PS es la Inteligencia de Enjambre, abarcando todos los algoritmos que pertenecen a ella.

Mientras que la segunda técnica más utilizada son los Algoritmos Genéticos, y la tercera más utilizada son las Redes Neuronales. El hecho de que estas tres técnicas se hayan utilizado en el 68% de los 65 casos registrados, indica que hay un relativo abandono generalizado en las otras 17 técnicas. Algoritmos clásicos como KNN y enfoques como los Agentes o las Redes Bayesianas no tienen mucha presencia en esta área, por lo que hacer investigaciones aplicándolos a la PS puede resultar en un estudio novedoso.

En respuesta a la **PI2**, las actividades de PS en donde se han aplicado técnicas se muestran la Tabla 8, en donde se expone a la generación de casos de prueba y la localización de fallas como las dos actividades más abordadas con IA. Estas dos son las únicas que representan verdaderos repuntes donde se busca mejoras para sus procesos con IA en comparación con las otras.

Tabla 8. Actividades de la PS y frecuencia de aparición.

Actividades (Identificador)	Frecuencia	Referencias
Generación de casos/datos de prueba (6)	21	[6, 14, 20–24, 26, 27, 29–31, 33, 34, 55, 58, 60]
Localización de fallas (5)	11	[17, 32, 40, 41, 47, 48, 52, 53]
Priorización de casos de prueba (11)	6	[15, 23, 57, 59]
Cobertura de ramas (7)	5	[13, 37, 38]
Pruebas de Software basada en búsqueda (1)	3	[28]
Pruebas de interfaz de usuario (3)	3	[42, 43, 54]
Exploración de caminos (10)	2	[19, 46]
Clasificación de resultados de prueba (13)	2	[49, 56]
Covering Arrays (15)	2	[16, 25]
Optimización de casos de prueba (18)	2	[14]
Pruebas automáticas (2)	1	[35]
Prueba de interacción combinatoria (4)	1	[51]
Identificación de casos correctos de prueba (8)	1	[50]
Prueba de mutación (9)	1	[18]
Pruebas a Sistemas Orientados a Servicios (12)	1	[44]
Detección de código maligno (14)	1	[45]
Entrenamiento de datos (16)	1	[39]
Oráculos (17)	1	[36]

Entre ambas representan el 34% de los casos, un porcentaje muy alto tomando en cuenta que la tercera actividad de PS más frecuente (priorización de casos de prueba) representa al 7% de los casos. Considerando la existencia de otras 16 actividades registradas y su poca exploración, muchos estudios de investigación pueden surgir a partir de ellas. Además, es muy útil para los Ingenieros de Software y *Testers* conocer la existencia de una amplia gama de estrategias para realizar generación de casos de prueba y localización de fallas de una forma óptima utilizando IA.

Para identificar cuáles fueron las técnicas más aplicadas en las actividades de la PS, se usó el gráfico de burbuja de la Fig. 3, en el cual se representa a cada una de las técnicas de IA y actividades de PS con el identificador que les fue asignado en la Tabla 7 y la Tabla 8, respectivamente. En la Fig. 3 el tamaño de las burbujas representa la frecuencia con la que una actividad de PS es abordada con una técnica de IA. El número que se encuentra en el centro de la burbuja es la magnitud que representa.

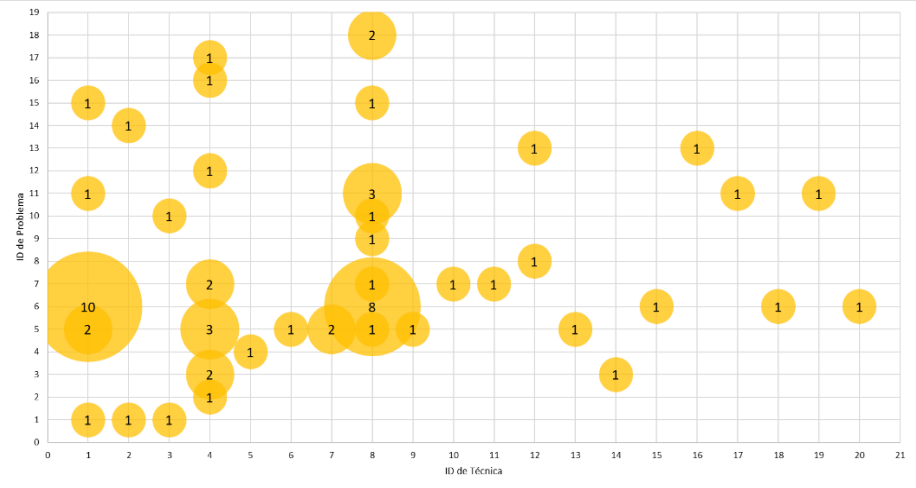


Fig. 3. Técnicas de IA utilizadas (horizontal) en actividades de la PS (vertical).

En la Fig. 3 se puede distinguir que los Algoritmos Genéticos, representados por el número 1 horizontal, se han utilizado para diversas actividades, pero principalmente para la generación de casos de prueba, representado con el número 6 vertical, sumando en total 10 reportes de esta aplicación. Otro punto para destacar es el amplio uso de la Inteligencia de Enjambres, representada por el 8 horizontal, siendo esta técnica la más utilizada en diferentes actividades de software, estando presente en ocho diferentes de las reportadas en este trabajo. El número 4 horizontal, que son las Redes Neuronales, también se distribuye en la aplicación de diferentes actividades de PS ya que, de sus 11 apariciones, se han aplicado a siete actividades.

Por el lado de las actividades de la PS, como se muestra en las frecuencias de la Tabla 8, la generación de casos de prueba, representada por el identificador 6 vertical, ha sido la más estudiada, principalmente con Algoritmos Genéticos (1 horizontal) e Inteligencia de Enjambre (8 horizontal). En cambio, la localización de fallas, representada por el número 5 vertical, ha sido tratada con 7 diferentes técnicas de la IA.

La Fig. 3 es muy útil para la identificación de áreas muy estudiadas, en las que un Ingeniero de Software o *Tester* pudiera buscar información para aplicarlas en sus actividades, así como para los investigadores, en donde fácilmente se pueden distinguir nichos de oportunidad.

En respuesta a la **PI3**, se encontró que de los 65 reportes de uso de técnicas de IA en actividades de PS todos lograron realizar la tarea para la que fueron diseñadas, y no sólo eso, ya que en 26 (40%) de los 65 casos lograron mejorar el proceso de la actividad de PS usando la técnica de IA. Como se puede observar, existe un gran potencial en el uso de IA en PS debido a los buenos resultados obtenidos por los diferentes estudios.

Para responder la **PI4** se muestra en la Fig. 4, que, de los 31 trabajos, 29 continuarán con algún tipo de investigación, mientras que los dos restantes tienen planeado implementar una herramienta que utilice las técnicas de la IA que aplicaron, para que los

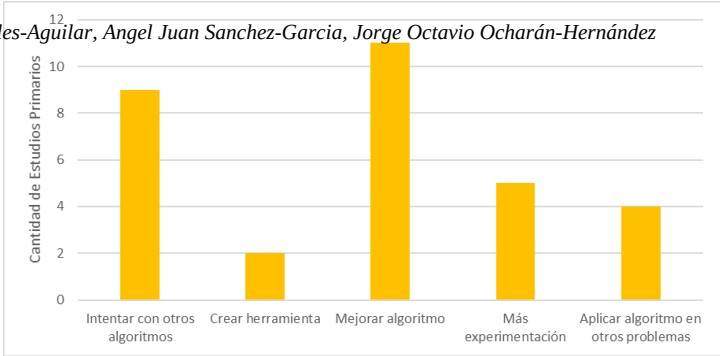


Fig. 4. Trabajo futuro reportado.

Ingenieros puedan consumir el conocimiento a través de una herramienta, mientras que una gran parte sigue en investigaciones.

5. Conclusiones y trabajo a futuro

Este documento presenta la planeación, ejecución y reporte de una RSL, enfocada en conocer la aplicación de la IA en la fase de PS. Reconocer las áreas con poca cobertura de pruebas nos deja nichos de oportunidad para hacer nuevas investigaciones, como por ejemplo la aplicación Lógica Difusa, Agentes o Algoritmos de Agrupamiento, o bien buscar soluciones con IA en entrenamiento de datos, prueba de mutación o en la identificación de casos correctos de prueba, por citar algunos ejemplos. Igualmente, al ver los resultados que se han obtenido al usar IA, donde todos los estudios reportan que tienen la misma o mejor eficiencia comparando con las técnicas actuales y sin IA, nos explica por qué la tendencia de investigaciones en esta área ha aumentado durante los últimos cinco años. Como trabajo futuro de este estudio, al ser parte de un trabajo más grande, se busca conocer la aplicación de la IA en otras áreas y fases de la IS, como Requisitos, Diseño y Construcción de Software, con la finalidad de encontrar más nichos de oportunidad para la colaboración entre la Inteligencia Artificial y la Ingeniería de Software.

Referencias

1. Bewes, J., Low, A., Morphett, A., Pate, F.D., Henneberg, M.: Artificial intelligence for sex determination of skeletal remains: Application of a deep learning artificial neural network to human skulls. J. Forensic Leg. Med. 62, pp. 40–43 (2019)
2. Turan, E., Çetin, G.: Using artificial intelligence for modeling of the realistic animal behaviors in a virtual island. Comput. Stand. Interfaces. 66, 103361 (2019)
3. Mayr, A., Weigelt, M., Masuch, M., Meiners, M., Hüttel, F., Franke, J.: Application Scenarios of Artificial Intelligence in Electric Drives Production. Procedia Manuf. 24, pp. 40–47 (2018)
4. Ye, Z., Yang, J., Zhong, N., Tu, X., Jia, J., Wang, J.: Tackle environmental challenges in pollution controls using artificial intelligence: A review. Sci. Total Environ. 134279 (2019)
5. Sommerville, I., Galipienso, M.I.A.: Ingeniería del software. Pearson Educación (2005)

6. Rabbi, K., Mamun, Q., Islam, M.R.: An efficient particle swarm intelligence based strategy to generate optimum test data in t-way testing. In: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 123–128 (2015)
7. Bourque, P., Fairley, R.E.: SWEBOK: guide to the software engineering body of knowledge. IEEE Computer Society (2014)
8. Sorte, B., Joshi, P., Jagtap, V.: Use of Artificial Intelligence in Software Development Life Cycle: A state of the Art Review. Int. J. Technol. Manag. 03, pp. 2309–4893 (2015)
9. Bunke, H., Last, M., Kandel, A.: Artificial Intelligence Methods in Software Testing. World Scientific (2004)
10. Hourani, H., Hammad, A., Lafi, M.: The Impact of Artificial Intelligence on Software Testing. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 565–570 (2019)
11. Kitchenham, B., S.C.: Guidelines for performing systematic literature reviews in software engineering (2007)
12. Cruzes, D.S., Dyba, T.: Recommended Steps for Thematic Synthesis in Software Engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement, pp. 275–284 (2011)
13. Boopathi, M., Sujatha, R., Senthil Kumar, C., Narasimman, S.: Quantification of Software Code Coverage Using Artificial Bee Colony Optimization Based on Markov Approach. Arab. J. Sci. Eng. 42, pp. 3503–3519 (2017)
14. Khari, M., Kumar, P., Burgos, D., Crespo, R.G.: Optimized test suites for automated testing using different optimization techniques. Soft Comput. 22, pp. 8341–8352 (2018)
15. Khatibsyarbini, M., Isa, M.A., Jawawi, D.N.A., Hamed, H.N.A., Mohamed Suffian, M.D.: Test Case Prioritization Using Firefly Algorithm for Software Testing. IEEE Access. 7, pp. 132360–132373 (2019)
16. Mahmoud, T., Ahmed, B.S.: An efficient strategy for covering array construction with fuzzy logic-based adaptive swarm optimization for software testing use. Expert Syst. Appl. 42, pp. 8753–8765 (2015)
17. Singh, P.K., Garg, S., Kaur, M., Bajwa, M.S., Kumar, Y.: Fault localization in software testing using soft computing approaches. In: 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), pp. 627–631 (2017)
18. Sadath, L., Nair, R.: Oncological Inspired Techniques for Intelligent Software Testing. In: 2019 Amity International Conference on Artificial Intelligence (AICAI) pp. 327–333 (2019)
19. Popentiu-Vladicescu, F., Albeanu, G.: Nature-inspired Approaches in Software Faults Identification and Debugging. Procedia Comput. Sci. 92, pp. 6–12 (2016)
20. Cai, L., Zhang, Y., Ji, W.: Variable Strength Combinatorial Test Data Generation Using Enhanced Bird Swarm Algorithm. In: 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp. 391–398 (2018)
21. Sharifipour, H., Shakeri, M., Haghighi, H.: Structural test data generation using a memetic ant colony optimization based on evolution strategies. Swarm Evol. Comput. 40, pp. 76–91 (2018)
22. Chawla, P., Chana, I., Rana, A.: A novel strategy for automatic test data generation using soft computing technique. Front. Comput. Sci. 9, pp. 346–363 (2015)
23. Mann, M., Tomar, P., Sangwan, O.P.: Bio-inspired metaheuristics: evolving and prioritizing software test data. Appl. Intell. 48, pp. 687–702 (2018)
24. Sabharwal, S., Bansal, P., Mittal, N., Malik, S.: Construction of Mixed Covering Arrays for Pair-wise Testing Using Probabilistic Approach in Genetic Algorithm. Arab. J. Sci. Eng. 41, pp. 2821–2835 (2016)
25. Sabharwal, S., Bansal, P., Mittal, N.: Construction of t-way covering arrays using genetic algorithm. Int. J. Syst. Assur. Eng. Manag. 8, pp. 264–274 (2017)

26. Li, X., Wong, W.E., Gao, R., Hu, L., Hosono, S.: Genetic Algorithm-based Test Generation for Software Product Line with the Integration of Fault Localization Techniques. Empir. Softw. Eng. pp. 23, 1–51 (2018)
27. Manikumar, T., John Sanjeev Kumar, A., Maruthamuthu, R.: Automated test data generation for branch testing using incremental genetic algorithm. S{\={a}}dhan{\={a}}. 41, pp. 959–976 (2016)
28. Feldt, R., Poulding, S.: Broadening the Search in Search-based Software Testing: It Need Not Be Evolutionary. In: Proceedings of the Eighth International Workshop on Search-Based Software Testing. pp. 1–7. IEEE Press, Piscataway (2015)
29. Shamshiri, S., Rojas, J.M., Fraser, G., McMinn, P.: Random or Genetic Algorithm Search for Object-Oriented Test Suite Generation? In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 1367–1374. ACM (2015)
30. Rodrigues, D.S., Delamaro, M.E., Corrêa, C.G., Nunes, F.L.S.: Using Genetic Algorithms in Test Data Generation: A Critical Systematic Mapping. ACM Comput. Surv. 51(41), pp. 1–41 (2018)
31. Esfandyari, S., Rafe, V.: A tuned version of genetic algorithm for efficient test suite generation in interactive t-way testing strategy. Inf. Softw. Technol. 94, pp. 165–185 (2018)
32. Arasteh, B., Bouyer, A., Pirahesh, S.: An efficient vulnerability-driven method for hardening a program against soft-error using genetic algorithm. Comput. Electr. Eng. 48, pp. 25–43 (2015)
33. Aleti, A., Grunske, L.: Test data generation with a Kalman filter-based adaptive genetic algorithm. J. Syst. Softw. 103, pp. 343–352 (2015)
34. Qi, R.-Z., Wang, Z.-J., Li, S.-Y.: A Parallel Genetic Algorithm Based on Spark for Pairwise Test Suite Generation. J. Comput. Sci. Technol. 31, pp. 417–427 (2016)
35. Zhang, M., Zhang, Y., Zhang, L., Liu, C., Khurshid, S.: DeepRoad: GAN-based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. pp. 132–142. ACM (2018)
36. Singhal, A., Bansal, A., Kumar, A.: An approach to design test oracle for aspect oriented software systems using soft computing approach. Int. J. Syst. Assur. Eng. Manag. 7, pp. 1–5 (2016)
37. Jitsunari, Y., Arahori, Y.: Coverage-Guided Learning-Assisted Grammar-Based Fuzzing. In: IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 275–280 (2019)
38. Grano, G., Titov, T. V, Panichella, S., Gall, H.C.: How high will it be? Using machine learning models to predict branch coverage in automated testing. In: 2018 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE) pp. 19–24 (2018)
39. Karmore, S.P., Mahajan, A.R.: New Approach for Testing and providing Security Mechanism for Embedded Systems. Procedia Comput. Sci. 78, pp. 851–858 (2016)
40. Kahles, J., Törrönen, J., Huuhtanen, T., Jung, A.: Automating Root Cause Analysis via Machine Learning in Agile Software Testing Environments. In: 2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST), pp. 379–390 (2019)
41. Heris, S.R., Keyvanpour, M.: Effectiveness of Weighted Neural Network on Accuracy of Software Fault Localization. In 5th International Conference on Web Research (ICWR), pp. 100–104 (2019)
42. Kamal, M.M., Darwish, S.M., Elfatatry, A.: Enhancing the Automation of GUI Testing. In: Proceedings of the 8th International Conference on Software and Information Engineering. 66–70. ACM (2019)

43. White, T.D., Fraser, G., Brown, G.J.: Improving Random GUI Testing with Image-based Widget Detection. In: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 307–317, ACM (2019)
44. Eniser, H.F., Sen, A.: Testing Service Oriented Architectures Using Stateful Service Virtualization via Machine Learning. In: IEEE/ACM 13th International Workshop on Automation of Software Test (AST), pp. 9–15 (2018)
45. Zein, S., Salleh, N., Grundy, J.: A systematic mapping study of mobile application testing techniques. J. Syst. Softw. 117, pp 334–356 (2016)
46. Yeh, C., Lu, H., Yeh, J., Huang, S.: Path Exploration Based on Monte Carlo Tree Search for Symbolic Execution. In: 2017 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 33–37 (2017)
47. Raman, M., Abdallah, N., Dunoyer, J.: An Artificial Intelligence Approach to EDA Software Testing: Application to Net Delay Algorithms in FPGAs. In: 20th International Symposium on Quality Electronic Design (ISQED), pp. 311–316 (2019)
48. Gupta, L., Salman, T., Zolanvari, M., Erbad, A., Jain, R.: Fault and performance management in multi-cloud virtual network services using AI: A tutorial and a case study. Comput. Networks. 165, 106950 (2019)
49. Liu, Y., Li, M., Wu, Y., Li, Z.: A weighted fuzzy classification approach to identify and manipulate coincidental correct test cases for fault localization. J. Syst. Softw. 151, pp. 20–37 (2019)
50. Li, Z., Li, M., Liu, Y., Geng, J.: Identify Coincidental Correct Test Cases Based on Fuzzy Classification. In: 2016 International Conference on Software Analysis, Testing and Evolution (SATE), pp. 72–77 (2016)
51. Jia, Y., Cohen, M.B., Harman, M., Petke, J.: Learning Combinatorial Interaction Test Generation Strategies Using Hyperheuristic Search. In: Proceedings of the 37th International Conference on Software Engineering - Volume 1. 540–550. IEEE Press, Piscataway (2015)
52. Kim, Y., Mun, S., Yoo, S., Kim, M.: Precise Learn-to-Rank Fault Localization Using Dynamic and Static Features of Target Programs. ACM Trans. Softw. Eng. Methodol. 28(23), pp.1–23:34 (2019)
53. Mariani, L., Monni, C., Pezzé, M., Riganelli, O., Xin, R.: Localizing Faults in Cloud Systems. In: IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), pp. 262–273 (2018)
54. Koroglu, Y., Sen, A., Muslu, O., Mete, Y., Ulker, C., Tanriverdi, T., Donmez, Y.: QBE: QLearning-Based Exploration of Android Applications. In: IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), pp. 105–115 (2018)
55. Walkinshaw, N., Fraser, G.: Uncertainty-Driven Black-Box Test Data Generation. In: 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST), pp. 253–263 (2017)
56. Roper, M.: Using Machine Learning to Classify Test Outcomes. In: 2019 IEEE International Conference On Artificial Intelligence Testing (AITest),pp. 99–100 (2019)
57. Butgereit, L.: Using Machine Learning to Prioritize Automated Testing in an Agile Environment. In: 2019 Conference on Information Communications Technology and Society (ICTAS), pp. 1–6 (2019)
58. Zamli, K.Z., Din, F., Baharom, S., Ahmed, B.S.: Fuzzy adaptive teaching learning-based optimization strategy for the problem of generating mixed strength t-way test suites. Eng. Appl. Artif. Intell. 59, pp. 35–50 (2017)
59. Wang, S., Huang, K.: Improving the efficiency of functional verification based on test prioritization. Microprocess. Microsyst. 41, pp. 1–11 (2016)
60. Arora, P.K., Bhatia, R.: A Systematic Review of Agent-Based Test Case Generation for Regression Testing. Arab. J. Sci. Eng. 43, pp. 447–470 (2018)

ISSN 1870-4069

Modelo de clasificación para la adopción de medios digitales en la Banca de Inversión

Jessica Guadalupe Vega Ledesma, Cristina Hernández Márquez, Alma Chávez Arrieta

Universidad Panamericana,
Facultad de Ingeniería,
México

{0011638, 0136183, 0234447}@up.edu.mx

Resumen. El presente estudio analiza la digitalización del área de Servicio al Cliente de un Banco de Inversión que desea aprovechar medios electrónicos desarrollados (Banca en Línea, Casa de Bolsa Digital o un asistente virtual) para mejorar la atención a sus clientes, reduciendo el número de llamadas al área de Servicio al Cliente y por tanto el costo asociado. Se utilizará aprendizaje supervisado aplicado al registro de llamadas realizadas al área de Servicio a Clientes en donde se podrá obtener la relación directa con la variable que controla la suscripción a la Banca en Línea o Casa de Bolsa Digital. El modelo resultante se utilizará para evaluar el total de llamadas recibidas por el área de Servicio a Cliente logrando predecir si el cliente con base en su perfil es un buen candidato para ser inducido a la utilización de medios electrónicos automatizados o bien, dirigirlo a la atención de un ejecutivo del área de Servicio al Cliente. Se balancearon las muestras y se utilizaron cinco modelos para evaluar donde el Clasificador de Árboles Aleatorios resultó con mejor precisión.

Palabras clave: Adopción de medios digitales, árboles aleatorios, servicios digitales financieros, aprendizaje supervisado.

Classification Model for Digital Media in an Investment Bank

Abstract. The following research analyzes the digitalization of the Customer Service of an Investment Bank that aims to take advantage of mobile devices (Online Banking, Digital Stock Exchange or a virtual assistant) to improve customer service, reducing the number of calls and therefore the associated cost. Supervised learning will be used through the logging of received calls at Customer Service; in where we can obtain the relationship between the variable that controls the subscription to digital banking and the customer´s profile. The resulting model will be used to evaluate the total number of calls received in the Call Center, predicting whether the client, based on his profile, is a good candidate to be induced to use automated sources or to direct them to the attention of an executive in the Customer Service area. The samples were balanced, and

five models were used to assess where the Random Tree Classifier was with the best precision.

Keywords: Adoption of digital media, tree conditional random fields, digital financial services, supervised machine learning methods.

1. Introducción

Actualmente uno de los más grandes retos de Sistema Financiero es aprovechar todos los medios electrónicos, la digitalización está cambiando el modo en que los clientes se relacionan con sus servicios y toman control de sus cuentas de inversión. Los grandes bancos alrededor del mundo utilizan modelos de ciencia de datos para el perfilamiento de sus clientes y selección de servicios de atención, mejorando la experiencia del cliente, logrando disminuir costos y bancarizando al mayor número de personas posibles obteniendo a su vez datos del comportamiento de sus clientes para ofrecer productos y servicios personalizados como lo demuestra Arguedas [1] y Cabrera [2].

En este trabajo se presenta un modelo clasificador para la digitalización bancaria que va directamente ligado al desarrollo de la telefonía celular, es decir no basta con que la telefonía sea móvil si no con accesos a internet; siendo ésta su única limitante como lo demuestra el estudio realizado en sus inicios Prior [3] para países en vías de desarrollo como lo son los africanos, en donde su penetración es más difícil.

En este trabajo se presenta un modelo clasificador basado en el algoritmo de Bosque Aleatorio, el cual se pretende implementar como filtro inicial en la recepción de llamadas del área de Servicio al Cliente cuyo objetivo principal sea determinar desde el inicio de la llamada el poder remitir al cliente al uso del servicio digital o de lo contrario al ejecutivo de servicio. Con esta medida, el Banco de Inversión generará un ahorro en número de llamadas personalizadas de al menos un 15% en un periodo de un año y, por otro lado, se atenderá de manera más eficiente al segmento de clientes de nuevas generaciones que prefieren un servicio más rápido y estandarizado.

El análisis siguió un flujo de trabajo estándar: recolección de datos, preprocesamiento de datos, selección de características, selección y evaluación de modelos y evaluación y presentación de resultados.

La organización del presente documento se divide en cuatro secciones. En la sección 2 presenta trabajo relacionados a la adopción de medios digitales, el análisis comparativo de métodos supervisados se propone en la sección 3, así como también se describe la metodología utilizada y descripción del conjunto de datos. En la sección 4 se comparan los resultados de cada método de aprendizaje comparado y finalmente en la sección 5 se resumen las conclusiones del ejercicio y recomendaciones al negocio.

2. Trabajos relacionados

El Sistema Financiero alrededor del mundo es particularmente sensible a la transformación digital, al tratamiento de los datos personales, a desarrollar nuevos canales y personalizar los servicios para inducir a los clientes a la bancarización digital; estudios alrededor del mundo han presentado que el uso de Inteligencia Artificial ha

mejorado sustancialmente dicha adopción como lo presenta Tater [4], en donde los principales factores en India fueron la comodidad, protección, seguridad, accesibilidad y un registro exacto de las transacciones. De igual manera se presentaron factores estadísticos como edad, sexo, sueldo y capacidad de ahorro que lograban que la innovación bancaria fuera recibida de una manera positiva en los clientes.

Según la interpretación de Angelov [5] acerca del informe de Deutsche Bank (2015), la digitalización del sector financiero es una tendencia que está siendo impulsada especialmente por tres motores: la experiencia del consumidor, el empuje tecnológico y los beneficios económicos.

Los Bancos de Inversión siguen esta misma tendencia, buscan desarrollar medios digitales especializados como bancas en línea personalizadas, casas de bolsa digitales y aplicaciones de fácil adopción ya que los clientes de este perfil con cierta resistencia a la atención tradicional, por pertenecer a un nivel social medio y medio-alto en el que han sido atendidos de manera persona a persona, adicional a que el rango de edad normal es conocido por tener mayor resistencia a la adopción de sistemas automatizados.

Segmentar a los clientes de acuerdo con sus elecciones anteriores es uno de los primeros pasos que han seguido la mayoría de las propuestas de este tipo de análisis. En el trabajo de Ladyzynski [6], los sistemas de recomendación o filtrado son utilizados cada vez más en el sector bancario: se aplican en la generación de calificación bancaria, análisis de riesgos, perfilamiento del cliente, predicción de tipos cambiarios y detección de fraudes.

Sin duda, la cuestión que nos hacemos en este trabajo ha sido analizada años atrás por instituciones financieras como lo estudia Moro [7] en donde el estudio se ha dirigido a la predicción del éxito del marketing de productos bancarios con base en las llamadas recibidas (o de salida) del área de Servicio al cliente pues la centralización de las interacciones con los clientes facilita la gestión de la información y permite enfocarse en maximizar el valor de la vida del cliente. En el trabajo de Moro han sido considerados varios clasificadores como regresión logística clásica, árboles de decisión y redes neuronales. Una de las contribuciones del trabajo de Moro, es haber comprobado que una de las grandes ventajas de los árboles de decisión es que son modelos de fácil entendimiento y que proporcionan buenas predicciones en las tareas de clasificación.

En 2019, Ambulamar [8] inicia con el establecimiento de las principales características de los clientes, aquellas que proporcionan información valiosa sobre los hábitos de vida de los clientes y sus preferencias, permitiendo definir la orientación de productos a clientes realmente interesados.

Adicionalmente es necesario realizar un esfuerzo mayor para lograr adaptar sistemas como asistentes virtuales para aquellos clientes cuya necesidad aún no es cubierta por herramientas automatizadas (Banca en línea) y que logren una adopción de una manera más natural, pues como dice Angelov [9], a raíz de la creciente complejidad tecnológica, los bancos necesitan de la ayuda de colaboradores externos y entran en juego los nuevos agentes con tecnologías que permiten reducir costes y automatizar procesos, de este modo la bancarización digital los clientes podrán mejorar su experiencia y al mismo tiempo reducir costos operativos a la institución de acuerdo con Raja [10].

La atención al cliente automatizada es una funcionalidad que puede brindar beneficios tangibles tanto al cliente por su disponibilidad 24 horas, aumento de la capacidad de atención de llamadas sin aumentar la plantilla de ejecutivos en el área de servicio a clientes, eliminación de tiempos de espera y disminución de porcentajes de abandono. Y de manera muy importante, hay que mencionar que la adopción de procesos automatizados disminuye considerablemente el riesgo de fraude como lo analiza Kartman [11].

En el trabajo de Medvedev [12] donde se referencia un caso de estudio de clasificación para identificar clientes que serán más receptivos en una campaña de telemarketing utilizando un perfilamiento previo de los clientes y aplicando el método de aprendizaje de máquina de árboles de decisión para una mejor penetración en su campaña.

En síntesis, la incorporación de tecnologías de avanzada en la producción, la velocidad de los cambios y la convergencia y combinación de múltiples tecnologías de avanzada está permitiendo la transformación digital produciendo eficiencias operativas (reducción de costos), mejora de productividad y nuevas oportunidades y modelos de negocio. Entre las tecnologías que comprenden la cuarta revolución industrial o la tercera ola se puede identificar: Análisis de Grandes Bases de Datos, Internet de las Cosas, Robótica e Impresión 3D e Inteligencia Artificial y el aprendizaje automático como estandarte de esta última que aporta un componente crítico que es la predicción según Agrawal [13], que constituye un valor fundamental en la toma de decisiones. Estas máquinas que predicen son tan valiosas porque pueden realizar mejores predicciones, más rápidas y más baratas que las que realizan los humanos como menciona Brynjolfsson [14]. La inteligencia artificial ofrece, por un lado, decisiones, es decir, conclusiones alcanzadas a partir de la deliberación algorítmica basada en los datos disponibles y por otro, soluciones, es decir cursos alternativos de acción para resolver un problema como dimensiona Von Krogh [15]. Las empresas generan una enorme cantidad de datos que, gracias a estos nuevos sistemas computacionales y algoritmos avanzados, pueden ser procesados y analizados minimizando el esfuerzo humano. Esto permite descentralizar la toma de decisiones, y como se menciona lo menciona Nocceti [16] pasar de modelos preventivos a modelos predictivos que pueden aplicarse en todas las áreas de la organización como lo demuestra Basco [17]. La inteligencia Artificial es el único campo que persigue la construcción de máquinas que funcionen automáticamente en medios complejos y cambiantes analizado por Russell [17]. Este campo no sólo intenta comprender, sino que también se esfuerza en construir entidades inteligentes.

3. Clasificación de clientes basado en aprendizaje supervisado

3.1. Descripción general

Se desea encontrar el modelo que permita la clasificación automática de clientes de acuerdo con sus características básicas. Este modelo se implementará en el receptor de llamadas de atención al cliente: los clientes clasificados con mayor posibilidad de estar registrados en banca electrónica serán motivados a utilizar la Banca digital mediante un mensaje de voz y de aceptar dicha invitación posteriormente podrán ser atendidos

Fig. 1. Flujo de trabajo para el perfilamiento de clientes y aplicación de métodos de aprendizaje automático.

por un agente automático. En la fase de creación de modelos se seleccionaron varios modelos de aprendizaje con el fin de encontrar el que mejor precisión de predicción resulte para el caso de estudio. Se decidió utilizar la característica que nos indica si el cliente está enrolado en Banca electrónica. Esta nos servirá como etiqueta clasificadora, ya que se considera por el negocio, que una persona que ya tiene posibilidades de entrar a su banca electrónica busca la velocidad de la atención sobre el contacto personal y por lo tanto recibirá el autoservicio como una funcionalidad deseable.

En seguida se describen la metodología y materiales utilizados para seleccionar un método de aprendizaje de máquina para clasificar en dos categorías a los clientes que realizan llamadas al área de Servicio al Cliente. El flujo para la evaluación del modelo se visualiza se realizó de acuerdo con la Fig. 1.

1. **Recolección de datos.** Del conjunto total de llamadas recibidas durante un trimestre en el área de Servicio a clientes se escogieron como las variables de estudio las características particulares del cliente dejando a un lado el registro, propios de la llamada como horario, duración, etc.

2. **Perfilamiento del cliente.** El conjunto de datos obtenido nos provee de una serie de características de los clientes que han realizado llamadas. Dentro de este conjunto de características, existe una que nos servirá como etiqueta clasificadora, es aquella que define si el cliente se encuentra enrolado en alguna de las plataformas digitales actualmente. Esta característica toma un valor 0 cuando no se encuentra enrolado en la banca en línea, o 1 cuando ya se encuentra enrolado. Ya que contamos con esta característica etiqueta, nuestro conjunto podrá ajustarse a un modelo de aprendizaje supervisado.

3. **Pre procesamiento.** Se requirió una limpieza de datos básica que consiste en evitar datos vacíos intercambiándose con el promedio de la población, esta acción se realizó para evitar sesgar los datos. De esta manera, para la característica "Tipo de Persona (Física/ Moral)"se encontraba vacía, se modificó como Persona Física. De la misma manera analizando la variable saldo contrato (dato representativo que muestra la cantidad a favor invertida en la cuenta de donde el cliente realizó la consulta u operación en Servicio a Clientes)y su correlación con la variable saldo cliente (dato representativo que muestra la suma de las cantidad a favor invertidas en todas y cada una de las cuentas del cliente que llamó a Servicio a Clientes) encontramos prácticamente es directa 1 a 1 ; es decir que los clientes seleccionados tienen solo una cuenta registrada en el banco al ser saldo contrato=saldo cliente.

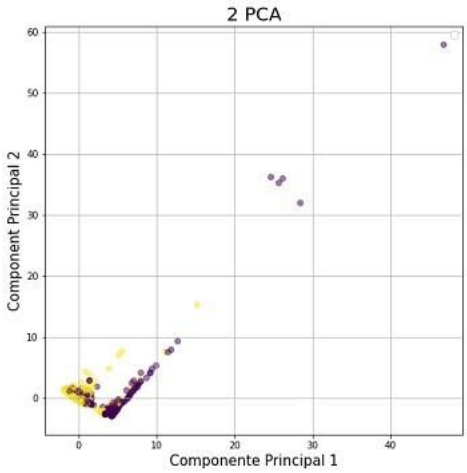


Fig. 2. Aplicación de PCA a las características.

Por lo que se toma la decisión de descartar una de ella sin que haya afectación en los resultados. Por último, usabilidad de internet

4. **Extracción y selección de características.** Se tienen ocho diferentes características a modelar que aportan valor al perfil del cliente e indispensables para el presente trabajo como son: la unidad de negocio es decir si el cliente pertenece al Banco o a la Casa de Bolsa en donde la experiencia nos ha demostrado que los clientes de casa de bolsa cuentan con saldos mayores y mayor resistencia a la adopción digital; la variable Centro Financiero nos otorgará la ubicación geográfica del cliente. Saldo cliente y Tipo de persona previamente explicado; el género (femenino /masculino) que de acuerdo con la Asociación de Internet Mx de su edición 2019 expone que el 66% de los usuarios de internet que usan servicios financieros son hombres usando este parámetro de referencia; edad y por último el estatus de suscripción a la Banca móvil o a la casa de bolsa digital. El análisis de componentes Principales PCA con 90% como varianza total deseada, nos ayudará a mejorar la aproximación de los modelos, así como a lograr visualizar las ocho dimensiones en un plano bidimensional como se muestra en la Fig. 2.

5. **Creación de modelos.** Se tiene la noción básica que el aprendizaje supervisado tendrá mejores resultados en nuestro caso de estudio, sin embargo, se seleccionan cinco diferentes modelos de aprendizaje, uno de ellos no supervisado y el resto supervisado: K-Medias, Árboles de decisión aleatorios, Máquinas de Soporte Vectorial, Clasificador Naive Bayes, Regresión Logística.

6. **Evaluación de modelos**. Para evaluar los modelos seleccionados, se utilizará la ecuación Error (1), donde N es el número de ejemplos en el conjunto de datos, y_i es el valor real y $f(x_i)$ es la estimación de la entrada x_i en el modelo f. Como auxiliar de decisión se utilizará la matriz de confusión, para revelar la distribución de los errores y ubicar el mejor método a aplicar en el centro de atención telefónica:

$$E = \frac{1}{N} \sum_{i=1}^{N} \|y_i - f(x_i)\|^2 \quad (1)$$

3.2. Conjunto de datos utilizado

El conjunto de datos utilizado describe las características de los clientes que han realizado llamadas al área de Servicio a Clientes de un Banco de Inversión durante un trimestre. Contiene 8,158 registros telefónicos correspondientes al perfil del cliente, así como de las características de la llamada como lo era el horario, duración, tipo de llamada, etc; siendo estas últimas variables eliminadas por no aportar información relevante para el presente estudio.

Las variables a considerar son las siguientes:

- Unidad Negocio. Fuente del contrato principal del cliente: (Banco/ Casa de Bolsa).
- Centro Financiero. Es la sucursal en la que la cuenta del cliente está adscrita, se refiere a la localización geográfica. Existen 20 diferentes posibles centros financieros repartidos en México.
- Saldo Cliente. Suma del saldo del total de cuentas del cliente.
- Tipo de Persona. Física o Moral.
- Género. Masculino o Femenino.
- Edad del Cliente.
- Usa Internet. Se refiere al porcentaje de penetración de uso de internet en compras por internet en la entidad federativa de la sucursal donde se encuentra el contrato del cliente.
- Status BE: Etiqueta que indica si el cliente está registrado en banca electrónica o no, puede tomar valores (0,1).
- Se utilizará la característica 'Status BE', como etiqueta predefinida, para lograr aplicar métodos de aprendizaje supervisado, ya que el negocio considera que los clientes que actualmente cuentan con acceso a la Banca en línea son clientes que buscan atención rápida y precisa y que por lo tanto tendrán mayor éxito de satisfacción en un medio automatizado.

3.3. Métodos de aprendizaje

Los métodos de aprendizaje máquina utilizados en este trabajo son los siguientes. Para entrenarlos se toma del conjunto total de datos, un 70% para entrenamiento, dejando un 30% para validación del modelo.

1. **K-medias** Para aplicar el modelo de K-Medias, ya que se requieren dos categorías a clasificar, se definen dos centroides. Comparando con la etiqueta predictora 'Status BE', es lograda una precisión de únicamente un 0.2904411. Ejemplo Fig 3.
2. **Árboles de decisión aleatoria** son algoritmos de aprendizaje supervisado tanto para clasificación como para regresión. Se basa en árboles, se dice que entre más árboles tenga, más robusto será el resultado. Como parámetros se

Tabla 1. Precisión obtenida en cada modelo.

Modelo	Precisión Obtenida	Análisis
KMeans	0.2904411	Precisión más baja
Árbol de Decisión	0.866013	Mejor Precisión
Soporte Vectorial	0.822712	
Naive Bayes	0.705065	
Regresión Logística	0.763071	

utilizan 8 estimadores y una profundidad máxima de 8, se definen dos categorías de acuerdo al modelo de negocio a implementar. Con esta modelo se logra una precisión del 0.866013.

3. **Máquina de soporte vectorial** Este clasificador separa puntos usando hiperplanos con el mejor margen posible. Se conocen como un clasificador discriminativo. Se aplicó un kernel de tipo rbf, con 16 grados como parámetros de generación de modelo. Se logra una precisión del 0.822712.

4. **Clasificador Naive Bayes** Este clasificador estadístico está basado en el Teorema de Bayes. Es uno de los más simples algoritmos de aprendizaje supervisado. Se recomienda en segmentos de datos muy grandes. Al aplicarse al conjunto de datos de negocio, logramos obtener únicamente una precisión del 0.705065.

5. **Regresión logística** es uno de los más simples y comúnmente usado para clasificar entre dos clases. Es fácil de implementar y puede ser usado como clasificador binario. Describe y estima la relación entre una variable binaria dependiente y una variable independiente. Por lo que nuestros datos pueden ajustarse bastante bien a este algoritmo. Al tomar un máximo de 500 iteraciones, utilizando la misma proporción 70-30 clásica, se logra una precisión del 0.763071.

4. Resultados y discusión

De los cinco modelos aplicados se obtuvieron las siguientes precisiones habiendo encontrado el modelo de árboles de decisión aleatoria con mayor precisión. Lo elegimos ganador y procedemos a afinar los parámetros estimadores, cuyo principal parámetro es el número de árboles en el bosque a definir. Para tener una mejor aproximación, se realiza un barrido de 40 interacciones.

A continuación, se presentan los resultados de acuerdo con la Tabla 1.

De acuerdo Fig. 2 podemos visualizar que el mejor resultado se obtiene cuando se utilizan 69 árboles y una profundidad máxima de 8 finalmente logrando una precisión de 87.6225%. Después de dichos 69 estimadores, la precisión no incrementa, únicamente varía y se mantiene cercana a la misma precisión.

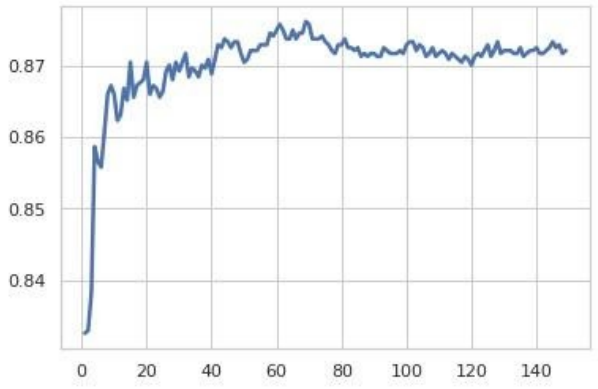


Fig. 2. Relación n estimadores - precisión resultante.

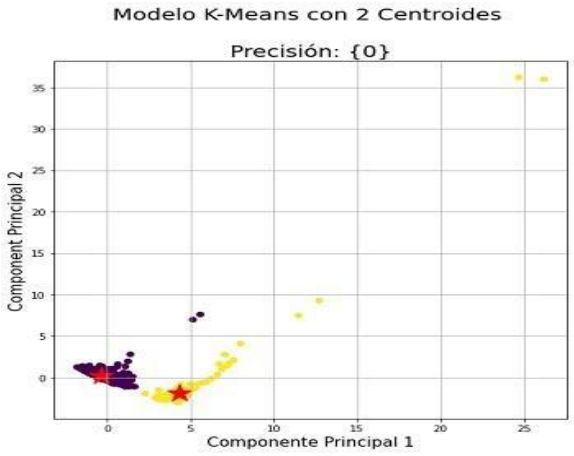


Fig. 3. Modelo K-Medias.

Tabla 2. Predicción de porcentaje de redireccionamiento de llamadas.

Etiqueta	Número de casos	Porcentaje
A Redireccionar	1425	91%
Atención Personal	141	9%

La discusión siguiente, es definir si la precisión resultante es suficiente para el modelo de negocio analizado ya que se requiere obtener al menos un 15% de las llamadas se direccionen a la atención automática de Banca en línea. Teniendo esta precisión, podemos obtener un 91% de direccionamiento de Banca en Línea, con un 87.62% de precisión con lo cual se da por válido el modelo para el negocio y se procederá a su implementación productiva. Tabla 2.

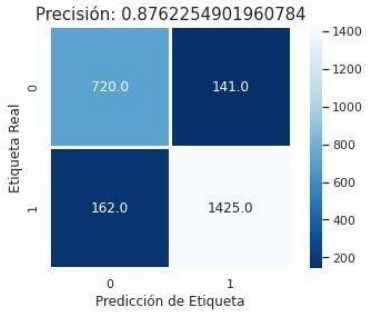


Fig. 3. Matriz de confusión resultado del método de Árbol de decisión.

Adicionalmente al modelo, se recomendará al negocio repetir el ejercicio después de un trimestre de ejecución para revalidar los resultados y reevaluar las características de entrada.

Por último, se obtuvo la matriz de confusión en donde se puede observar que existen 720 verdaderos positivos y 1425 falsos negativos siendo estas cifras las más altas de nuestra matriz apoyando a los resultados generados por el modelo.

5. Conclusiones

El aprendizaje supervisado a través del modelo de árboles de decisión es muy útil en el caso de la clasificación de datos cuando se tiene una característica que funcione como etiqueta clasificadora.

Se propone una aproximación distinta para nuestro caso de estudio, obteniendo una relación adecuada a la población mexicana, utilizando diferentes fuentes de datos que den características con mayor valor a cada cliente, por ejemplo, explorar la calificación dada a la experiencia anterior y no únicamente basado en el perfil del cliente.

Adicionalmente al modelo, se realizarán las siguientes recomendaciones al negocio en donde se deduce que la adopción de la tecnología puede estar relacionado con la edad de los clientes y posiblemente el monto de inversión Los clientes pertenecen a la generación de los sesentas aproximadamente y será necesario acoplar tanto los scripts de ventas como las aplicaciones a las características propias de esta generación, por lo que creemos que la utilización de las herramientas de Banca en Línea o Casa de Bolsa Digital para operaciones sencillas puede ser adoptada con mayor facilidad.

Otro dato importante analizado fue la co-relación que obtuvimos entre saldo cliente y saldo contrato, lo que quiere decir que la mayoría de estos usuarios únicamente cuentan con un solo contrato, es decir son clientes buscan simplificación y utilizan una sola cuenta para diversificar su dinero y no varias cuentas destinadas a fines distintos, lo que hace que la aplicación se entienda de mejor manera. El género del cliente que predomina es masculino, aunque no parece tener relevancia en el análisis realizado.

El presente análisis está basado en datos reales y en el reto que representa ahora para los bancos poder llegar a la digitalización de los servicios.

Referencias

1. Arguedas-Sanz, R.: La Transformación Digital en el Sector Financiero. Universidad Nacional de Educación a Distancia, pp. 1–3 (2019)
2. Cabrera-Vergara, C.: Análisis de los principales factores que permitieron desarrollar una eficiente transformación digital en las principales empresas del sector banca múltiple peruano. Universidad Peruana de Ciencias Aplicadas (UPC) https://repositorioacademico.upc.edu.pe/handle/10757/650437 (2019)
3. Prior, F.: La banca móvil como catalizadora de la bancarización de los pobres: modelos de negocio y desafíos regulatorios. (2008)
4. Tater, B., Tanwar, M., Murari, K.: Customer Adoption of Banking Technology in Private Banks of India. International Journal of Banking and Finance, 8(3) (2011)
5. Angelov, A.S.: El sistema financiero digital: los nuevos agentes. Universidad de Alcalá, http: //www3.uah.es/iaes/publicaciones/DT_04_19.pdf (2019)
6. Ładyżyński, P., Żbikowski, K., Gawrysiak, P.: Direct marketing campaigns in retail banking with the use of deep learning and random forests. Expert Systems with Applications, 134, pp. 28—35 (2019)
7. Moro, S., C.P.R.P.: A data-driven approach to predict the success of bank telemarketing. https://www.sciencedirect.com/science/article/pii/ (2014)
8. Anbumalar S., C.J.H.: A study on perception of customer toewards digital bankinging present scenario. Our Heritage (2), pp. 47–54 (2020)
9. Angelov, A. S.: El sistema financiero digital: los nuevos agentes Universidad de Alcalá. http:// www3.uah.es/iaes/publicaciones/DT_04_19.pdf (2019)
10. Raja, K.: Virtual banking Paramakudi, India: Studies in Indian Place Names (UGC Care Journal) (2020)
11. Squadex: The top 9 machine learning use cases in business. https://squadex.com/insights/top-machine-learning-use-cases-business/amp/ (2019)
12. Towards Data Science: Machine learning classification with python for direct marketing. https://towardsdatascience.com/machine-learning-classification-with-python-for-direct-marketing-2da27906ddac (2019)
13. Agrawal, A., Gans, J., Goldfarb, A.: Prediction machines: the simple economics of artificial intelligence. Harvard Business Press (2018)
14. Brynjolfsson, E., McAfee, A.: Machine, platform, crowd: Harnessing our digital future. WW Norton & Company (2017)
15. Von-Krogh, G.: Artificial Intelligence in Organizations: New Opportunities for Phenomenon-Based Theorizing. Academy of Management Discoveries, 4(4), pp. 404–409 (2018)
16. Nocetti, N.: Promesas de la Inteligencia Artificial y el Aprendizaje Automático para la Banca Tradicional Privada. Un análisis de oportunidades, aplicaciones, barreras y riesgos. Universidad de San Andrés, pp. 63–69 (2019)
17. Basco, A., Beliz, G., Coatz, D., Garnero, P.: Industria 4.0. Fabricando el Futuro.
18. Russell, S., & Norvig, P. Artificial intelligence: A modern approach. Prentice Hall, Englewood cliffs, NJ (2018)

ISSN 1870-4069

Extracción de eventos en electrocardiograma mediante el procesamiento de señales, utilizando el algoritmo MTEO

Leticia Gómez Rivera, María Margarita Labastida Roldán

Universidad Autónoma de Tlaxcala,
Facultad de Ciencias Básicas, Ingeniería y Tecnología,
México

lety.gr0108@gmail.com,
mariamargarita.labastida.r@uatx.mx

Resumen. Los sistemas enfocados en el área de la salud son importantes para mejorar los resultados obtenidos por los médicos especializados, debido a que estos sistemas son capaces de procesar y analizar grandes cantidades de información. En esta investigación, se implementa un sistema que representa una alternativa a los electrocardiógrafos multicanal con interpretación, ya que en México, el número de electrocardiógrafos por Entidad Federativa es completamente bajo. En promedio, Querétaro tiene el mayor número de electrocardiógrafos y solo cuenta con siete dispositivos por hospital [1]. El sistema desarrollado en esta investigación realiza el filtrado de la señal de electrocardiograma, obtiene la ubicación de las ondas P, Q, R, S y T, la duración de los intervalos RR, PR, QRS y PP, además de la Frecuencia Cardíaca. Estos datos son importantes porque el especialista puede consultarlos con un grado de precisión de milisegundos, lo cual ayuda a realizar un diagnóstico más preciso, ya que diversos factores como la falta de experiencia del médico o las dificultades para detectar irregularidades en grandes cantidades de información pueden desencadenar en un diagnóstico erróneo y afectar la calidad de vida del paciente, y en casos más graves, su propia salud.

Palabras clave: Electrocardiograma, filtro, señal, intervalos.

Extraction of Electrocardiogram Events through Signal Processing Using the MTEO Algorithm

Abstract. Systems focused on the health area are important to improve the results obtained by medical specialists, because these systems are capable of processing and analyzing large amounts of information. In this research, a system is implemented that represents an alternative to multichannel electrocardiographs with interpretation, since in Mexico, the number of electrocardiographs by Federal Entity is completely low. On average, Querétaro has the highest number of electrocardiographs and only seven devices per hospital [1]. The system developed in this research performs the filtering of the electrocardiogram signal, obtaining the location of the P, Q, R, S and T waves, the duration of the RR, PR, QRS and PP intervals, as well as the Heart Rate. This data is important because

the specialist can consult it with a degree of precision of milliseconds, which helps to make a more accurate diagnosis, since various factors such as the lack of experience of the doctor or the difficulties in detecting irregularities in large amounts of information can trigger a wrong diagnosis and affect the quality of life of the patient, and in more serious cases, their own health.

Keywords: Electrocardiogram, filter, signal, intervals.

1. Introducción

Las enfermedades cardiovasculares constituyen la primera causa de muerte en nuestro país a nivel mundial. En 2015, la Organización Mundial de la Salud (OMS), estimó que el 31% de todas las muertes registradas en el mundo se debieron a estas condiciones. En México se estima que las enfermedades cardiovasculares representan el 20% del total de muertes en los adultos. De acuerdo con el Instituto Nacional de Estadística y Geografía (INEGI), en 2016 se reportaron 136,342 fallecimientos a causa de enfermedades del corazón, un aumento de 7,611 muertes respecto al 2015. Las causas de muerte incluyen enfermedad isquémica del corazón, enfermedades cerebrovasculares, hipertensivas, entre otras [2].

Observando las estadísticas de las muertes ocasionadas por enfermedades cardiovasculares resulta alarmante la cifra de personas afectadas, por lo cual, es necesario hacer un análisis de este tema en el área de la computación debido a que se puede realizar un aporte importante al campo de la cardiología, ayudando a los médicos a obtener diagnósticos precisos para dar un tratamiento oportuno a la enfermedad.

El electrocardiograma es un gráfico en el que se estudian las variaciones de voltaje en relación con el tiempo y en el que se registra la actividad de la corriente eléctrica que se está desarrollando en el corazón durante un tiempo determinado [3], por este motivo, obtener las ondas que componen al electrocardiograma, sus intervalos y la frecuencia cardíaca es un elemento clave para realizar una evaluación del estado del corazón. En esta investigación se propone una herramienta implementada en MatLab, para filtrar la señal de electrocardiograma, obtener las ondas P, Q, R, S y T presentes en ella, realizar el cálculo de intervalos RR, PP, RR y QRS, los cuales se miden como el tiempo transcurrido en segundos de una onda a la siguiente y calcular la Frecuencia Cardíaca. El motivo principal para realizar esta investigación, es implementar una alternativa diferente de los sistemas convencionales que realizan la extracción de ondas e intervalos (electrocardiógrafos) y así obtener esta información solo haciendo uso de una computadora.

2. Conceptos básicos

En esta sección se definen brevemente los conceptos básicos que están presentes en esta investigación.

Filtrado: Se emplea de formas muy variadas en el Procesamiento Digital de Señales, por ejemplo, para eliminar el ruido indeseado que pueda existir en las señales, para la conformación espectral en la ecualización de canales de comunicación, en la detección de señales de radar, sonar, etc. [4].

Electrocardiógrafo: Detecta las señales eléctricas asociadas con la actividad cardíaca y produce un electrocardiograma.

Electrocardiógrafo multicanal con interpretación: Identifica la señal completa y determina todas las señales de las ondas P, complejos QRS y ondas T. Imprime una lista de señales como son: frecuencia cardíaca, la amplitud de la señal, el tamaño de las ondas y los intervalos entre los componentes de las ondas [5].

Onda P: Es la primera flexión hacia arriba que aparece en el Electrocardiograma. Esta onda representa el momento en el que las aurículas se están contrayendo y enviando sangre hacia los ventrículos.

Complejo QRS: Corresponde con el momento en que los ventrículos se contraen y expulsan su contenido sanguíneo, como su nombre lo indica, consta de las ondas Q, R y S.

Onda T: La onda T representa el momento en el que el corazón se encuentra en un periodo de relajación, una vez que expulsó la sangre de los ventrículos. Consiste en una deflexión normalmente positiva

Intervalo PR: Es la distancia desde el inicio de la onda P hasta el inicio del complejo QRS. La duración normal en adultos del intervalo PR oscila entre 120 y 200 ms.

Segmento PR: Es la distancia que se encuentra entre el final de la onda P y la siguiente deflexión positiva o negativa del ECG [6].

3. Trabajos relacionados

Los trabajos que se mencionan en esta sección se relacionan con este trabajo porque en cada uno de ellos se extraen eventos del electrocardiograma, los cuales contemplan ondas P, Q, R, S, T, complejos QRS, frecuencia cardíaca, intervalos y segmentos, además de los porcentajes de exactitud obtenidos después de la extracción, lo cual es un marco de referencia para comparar los resultados y verificar si el sistema de extracción de eventos tiene un mejor desempeño o resultados similares.

En [7] se plantea un sistema de procesamiento para señales electrocardiográficas, basado en el empleo de una herramienta de software diseñada en *LabVIEW 2011*. Esta herramienta consta de tres etapas que son: ajuste de nivel, filtrado digital de señal y detección automática de los complejos QRS.

También se menciona un nuevo método para obtener la frecuencia cardíaca de un electrocardiograma, en el que primero se detectan los complejos QRS presentes en la señal y a partir de ellos se determina la frecuencia cardíaca. Para lograr la detección de los complejos QRS se describe que es necesario obtener las ondas R, discriminando las ondas P y T.

En [7] se encuentra una clara relación con esta investigación debido a que uno de los alcances contemplados es realizar el cálculo de la frecuencia cardíaca a partir del conteo de complejos QRS. Se menciona también que se obtuvo un valor promedio de 94.38% de exactitud al detectar la frecuencia cardíaca, lo que constituye un marco de referencia para poder observar el desempeño del Sistema de Extracción de Eventos.

En [8] se propone implementar técnicas de análisis de bajo costo computacional para medir con precisión los principales intervalos que componen el latido cardíaco, por tal motivo, se describe que la detección del latido cardíaco consiste en ubicar los intervalos donde ocurren los complejos QRS y se menciona que la detección de los puntos Q y S se realiza para medir algunos intervalos como lo son el QS y el ST. Esta detección se basa en la estimación del punto más pronunciado en la curva inicial y final del pulso.

En [8] se describe un algoritmo de detección de la Onda T, que consiste en generar una señal donde se captan las ondas T y se elimina el resto de los componentes de la señal, después se genera un pulso único que se define sobre el espacio de tiempo de cada onda T y finalmente se ubica con precisión el final del pulso obtenido correspondiente al final de la onda T. Se menciona que como resultado de la detección de Complejos QRS se obtuvo un error promedio de 0.095%, y el error porcentual promedio en la medición de los intervalos QT, en los pacientes de control fue de 1.89% y en pacientes enfermos fue de 5.09%. El aporte de este trabajo a la investigación es que propone un algoritmo para la detección de la onda T y también sirve como punto de referencia para comparar el porcentaje de error promedio en la detección de complejos QRS y observar si el sistema de extracción de eventos obtiene resultados similares a los obtenidos en [8].

En [9] se menciona la necesidad de realizar la prueba de Electrocardiograma de forma inalámbrica para facilitar la aplicación de dicha prueba en situaciones cotidianas sin perder eficacia, por tal motivo, se propone el desarrollo de un dispositivo inalámbrico, con una interfaz de pc-usuario visualizada en MatLab. También se menciona que para realizar las pruebas se tomaron muestras de voluntarios con una duración de un minuto cada una. En [9] se menciona que se obtienen resultados de precisión de 95.54% en la detección de ondas P, de 98.11% en detección de los complejos QRS y de 95.54% en la detección de la onda T, los cuales pueden ser comparados con los resultados del sistema de extracción de eventos para su evaluación.

4. Descripción de la aplicación

En este trabajo, se implementa un sistema que obtiene las ondas, intervalos y frecuencia cardíaca presentes en un electrocardiograma, mostrando así una alternativa a los electrocardiógrafos convencionales. Como primer paso, el sistema toma como dato de entrada una prueba de Electrocardiograma en formato de archivo .mat, para ser procesado por Matlab. El Electrocardiograma se carga y se muestra en la interfaz gráfica. Después, haciendo uso del algoritmo MTEO se realiza el filtrado de la señal y la eliminación del ruido para obtener las ondas P, Q, R, S y T.

El posicionamiento de las ondas se muestra de manera gráfica en el electrocardiograma y en forma numérica en un contenedor rotulado con la letra de la onda, donde se muestra el segundo en el que ocurrió. El siguiente paso por realizar es el cálculo de los intervalos, los cuales son: Intervalo RR, PP, PR y QRS, sus valores también se muestran de segundos en la interfaz gráfica del sistema. Por último, se calcula la frecuencia cardíaca del electrocardiograma y se muestran los latidos por minuto en un apartado de la interfaz gráfica. El proceso se muestra de manera general en la Fig. 1.

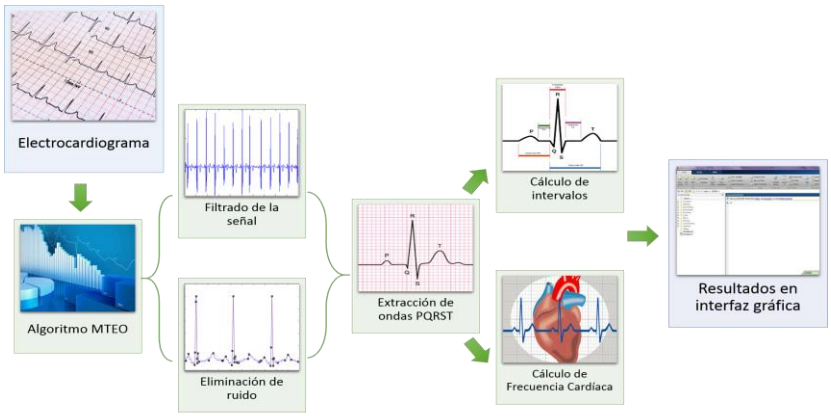


Fig. 1. Diagrama General del Sistema.

Fig. 2. Opción Cargar ECG.

5. Interfaz de usuario

En este apartado se muestra la interfaz gráfica del sistema, la cual fue implementada en MatLab y está conformada por un componente en el que se muestra la señal del electrocardiograma y por contenedores donde se muestran los resultados después del filtrado. Se presentan cuatro opciones que pueden ser realizadas por el usuario del sistema:

Cargar ECG: En esta opción, el usuario puede elegir el electrocardiograma que quiere procesar y después de la elección, la señal se muestra en pantalla (ver Fig. 2.)

Extraer eventos: Cuando el usuario selecciona esta opción, se inicia con el filtrado de la señal y se muestra la ubicación de las ondas P, Q, R, S y T, además de los segundos en los que ocurrieron. Esto se puede observar en Fig. 3.

Obtener intervalos: En esta opción se realiza el cálculo de los intervalos RR, PP, PR y QRS y se muestra su duración en segundos. También se realiza el cálculo de la Frecuencia Cardíaca y se muestran los latidos por minuto (ver Fig. 4.)

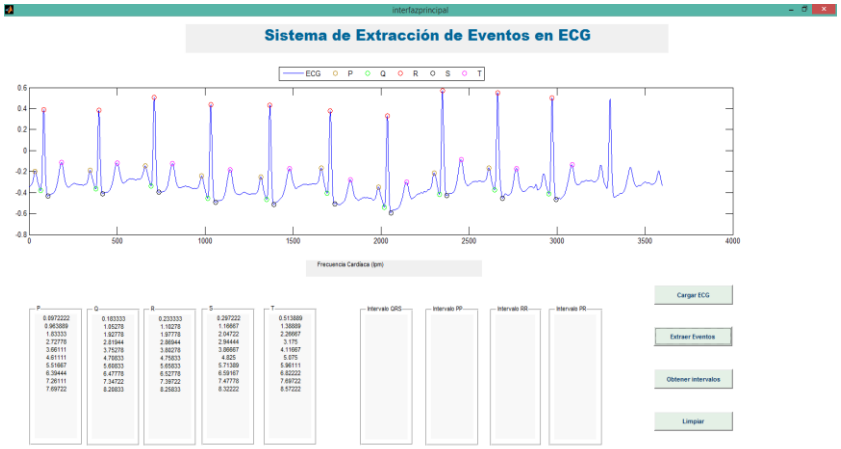


Fig. 3. Opción Extraer Eventos.

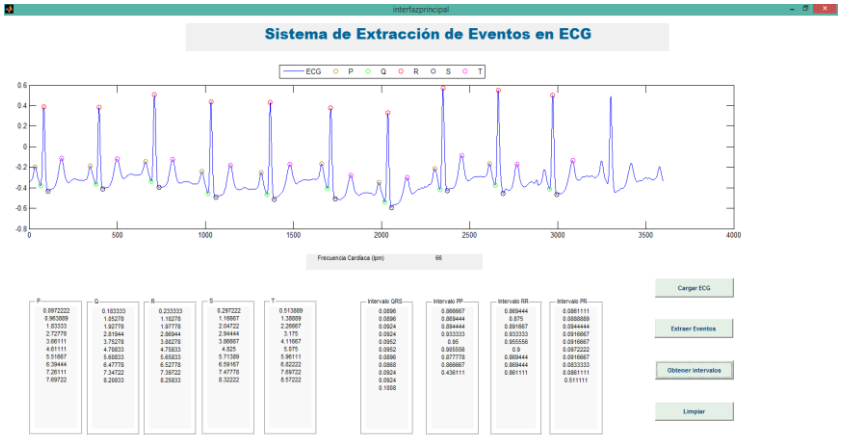


Fig. 4. Opción Obtener intervalos.

- **Limpiar:** Esta opción limpia los datos presentes en los contenedores para que el usuario pueda cargar una nueva señal de Electrocardiograma**.**

6. Implementación

Como primer paso a seguir para obtener las ondas P, Q, R, S y T, se realiza el filtrado de la señal de electrocardiograma haciendo uso de las funciones de Matlab *butter()* y *filfilt()*. La función butter*()* diseña un filtro Butterworth de paso bajo, paso alto, paso de banda o bandstop.

La función *filfilt()* realiza el filtrado digital de fase cero procesando los datos de entrada, tanto en las direcciones hacia delante como en las direcciones inversa *x*.

Para realizar la extracción de las ondas P, Q, R, S y T, se hace uso del algoritmo MTEO [10] implementado en Matlab. El Operador de Energía Teager Multirresolución (MTEO) en conjunto con un análisis estadístico se emplea en la etapa de detección de ondas en electrocardiograma y prueba simultáneamente tanto la frecuencia como la amplitud de potenciales de acción de unidades motoras (MUAP), para reducir las falsas detecciones y mejorar la precisión de detección. La salida MTEO se utiliza para la alineación de MUAP, la reducción de plantillas y la extracción de características para reducir el ruido de fondo y los picos no relacionados [10].

El cálculo de intervalos se realiza a partir del posicionamiento de las ondas. El intervalo RR es el tiempo transcurrido en segundos de una onda R a la siguiente y para calcularlo se toma como dato de entrada el vector de picos R y se realiza una sustracción. Para calcular el intervalo PP se sigue el mismo procedimiento que en el cálculo del intervalo RR solo que en este caso el dato de entrada es la onda P. El intervalo PR se mide desde el inicio de la onda P, hasta el inicio de la onda R o de la onda Q y para realizar su cálculo se toman como datos de entrada el vector de ondas Q y el de ondas P y se realiza una sustracción. El intervalo QRS se mide desde el comienzo de la onda Q hasta el final de la onda S y como datos de entrada para realizar su cálculo se utilizan los vectores de ondas Q y S.

En este trabajo, el cálculo de la Frecuencia Cardíaca se realiza utilizando el procedimiento médico de conteo de número de complejos QRS que ocurren en 10 s. y después multiplicarlos por seis para obtener los latidos por minuto. Para esto, se obtiene el número de elementos contenidos en el vector de intervalos QRS y después se multiplica por seis.

7. Experimentación y análisis de resultados

Para realizar las pruebas del Sistema de Extracción de Eventos propuesto en este trabajo, se reunieron 20 muestras de Electrocardiograma de la Base de Datos MIT-BIH Arrhythmia [11], las cuales cuentan con un identificador de tipo numérico que inicia en 100 y termina en 124, se tomó un tiempo de duración de 10 s. de cada muestra y fueron descargadas en formato .mat para ser procesadas en MatLab.

7.1. Pruebas cuantitativas

En este trabajo se realizaron pruebas cuantitativas al sistema para poder determinar la exactitud que tiene al extraer los eventos del electrocardiograma, los cuales son las ondas, intervalos y frecuencia cardíaca. Las pruebas fueron realizadas por el Médico Cirujano Julio César Ruiz Rivera y en este apartado se muestra la evaluación de la señal 119.mat, debido a que es una de las señales en las que es más difícil poder detectar las ondas a simple vista y el objetivo es identificar si el sistema realiza la extracción de eventos de manera eficiente. El procedimiento de evaluación se detalla a continuación:

1. El especialista observa la señal electrocardiográfica obtenida en el sitio de Physionet LightWave [12]. La señal se muestra en la Fig. 5.

Fig. 5. Señal 119.mat. Obtenida de LightWave [12].

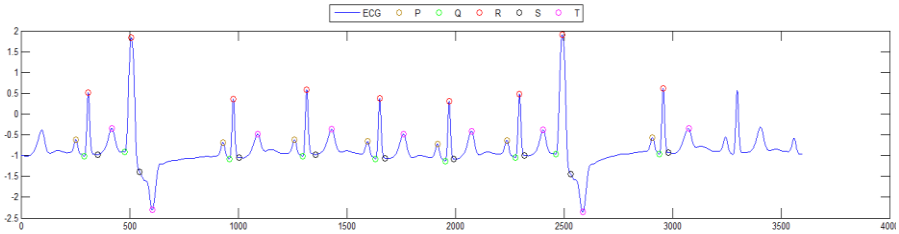


Fig. 6. Señal 119.mat. Procesada por el Sistema.

2. Compara la señal de LightWave con la obtenida por el Sistema después del filtrado y la extracción de ondas. La señal filtrada por el Sistema se muestra en la Fig. 6.
3. El especialista revisa la duración en segundos de los intervalos, además de la frecuencia cardíaca y el segundo en el que ocurre cada onda (ver Fig. 7.).
4. El especialista evalúa los resultados haciendo uso de un artefacto de evaluación donde se incluye el identificador de cada uno de los electrocardiogramas de prueba, un apartado para evaluar si las ondas P, Q, R, S y T fueron ubicadas correctamente en el electrocardiograma, una sección para evaluar si la frecuencia cardíaca fue calculada correctamente y por último, una sección donde se evalúa la cantidad de intervalos QRS, PP, PR y RR calculados correctamente.

7.2. Análisis de resultados

Después de que el especialista realizó la evaluación del Sistema con el artefacto de evaluación en las 20 pruebas de electrocardiograma se obtienen los resultados que se visualizan en la Tabla 1 “Porcentajes de extracción de Eventos en Electrocardiograma”.

8. Conclusiones y trabajo a futuro

Se propone que el sistema de Extracción de Eventos mostrado en este trabajo sea evaluado por lo menos por cinco especialistas más, esto con el objetivo de verificar que los resultados de la etapa de pruebas fueron correctos y así garantizar su buen funcionamiento.

Se concluye que, si los especialistas no cuentan con un electrocardiógrafo multicanal con interpretación, este sistema es útil para ayudarlos a realizar un diagnóstico con

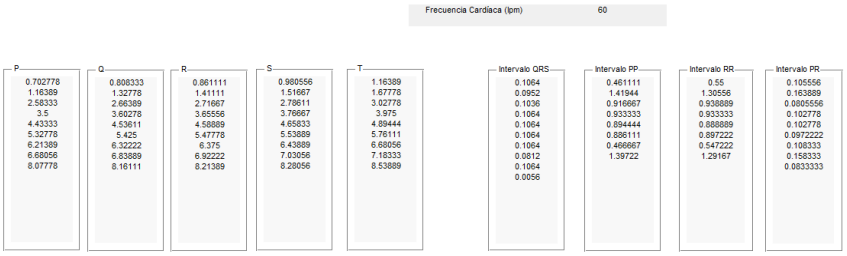


Fig. 7. Resultados obtenidos de la Señal 119.mat.

Tabla 1. Porcentajes de extracción de Eventos en Electrocardiograma.

Elemento Extraído	Porcentaje de Exactitud
Onda P	100%
Onda Q	100%
Onda R	100%
Onda S	100%
Onda T	100%
Frecuencia Cardíaca	100%
Intervalo QRS	100%
Intervalo PP	100%
Intervalo RR	100%
Intervalo PR	100%

mayor grado de precisión, ya que les proporciona los datos necesarios para detectar el tipo de enfermedad que sufre el paciente.

Las contribuciones de este trabajo se pueden observar principalmente en la recolección de muestras para conformar la base de conocimiento, ya que puede apoyar a otras personas interesadas en realizar aportes al área de cardiología porque se detalla el tiempo que fue tomado de cada muestra y el formato en el que se descarga el electrocardiograma para ser procesado en MatLab, además de que esta investigación también puede ser utilizada para realizar nuevos trabajos, como la detección de enfermedades en el corazón, y por último, la rúbrica de evaluación propuesta en esta investigación puede utilizarse y modificarse para que otros trabajos que estén relacionados sean evaluados por un especialista.

Como trabajo futuro, se propone que, utilizando los datos obtenidos por el sistema de Extracción de Eventos, se implemente un clasificador para detectar y clasificar anomalías en el ritmo cardíaco. Los datos de entrada serían las mismas pruebas de electrocardiograma que se utilizaron en esta investigación y las que pertenecen a ocho tipos de anomalías en el ritmo cardíaco: Fibrilación auricular, bradicardia sinusal, taquiarritmia supraventricular, aleteo auricular, bloqueo AV de primer grado, bloqueo cardíaco completo, bloqueo Mobitz II y taquicardia ventricular.

Las variables predictoras contempladas a ser utilizadas por el clasificador son: Intervalo QRS, Intervalo PP, Intervalo PR, Intervalo RR, Frecuencia Cardíaca, Onda P y Onda Q.

Referencias

1. Solis, C.L.: Electrocardiografía y su impacto en México el Hospital, 74(1), pp.13 (2018)
2. OMENT: Un panorama de las enfermedades cardiovasculares, Observatorio Mexicano de Enfermedades No Transmisibles/ Universidad Autónoma de Nuevo León, http://oment.uanl.mx/un-panorama-de-las-enfermedadescardiovasculares (2019)
3. López, A., Macaya, C.: Libro de la salud cardiovascular del Hospital Clínico San Carlos y de la Fundación BBVA. España: Editorial Nerea, S.A. (2009)
4. Proakis, J., Manolakis, D.: Tratamiento Digital de Señales. Pearson Educación S.A. (2007)
5. Centro Nacional de Excelencia Tecnológica en Salud. Guía Tecnológica No. 17: Electrocardiógrafo. Secretaría de Salud (2006)
6. Bayes, A.: Manual de Electrocardiografía básica. Barcelona: Caduceo Multimedia, S.L. (2014)
7. Bistel-Esquivel, R.A., Fajardo-Márquez, A.: Diseño de un Sistema de Adquisición y Procesamiento de la Señal de ECG basado en Instrumentación Virtual. Ingeniería Electrónica, Automática y Comunicaciones, 36(1) (2015)
8. Dugarte, J.N, Alvarez, A., Dugarte, E., Álvarez, G.: Técnicas de procesamiento de la señal ECGAR aplicadas en el prototipo DIGICARDIAC. Revista del Instituto Nacional de Higiene Rafael Rangel, 46(1–2), pp. 84106 (2020)
9. Quintero, J., Tabares, S., Perdomo, J.: Prototipo Electrocardiógrafo Inalámbrico para la detección de enfermedades que desencadenen la muerte súbita, con software de diagnóstico médico aproximado. https://revistas.utp.ac.pa/index.php/memoutp/article/view/1267 (2018)
10. Sedghamiz, H., Santonocito, D.: Unsupervised Detection and Classification of Motor Unit Action Potentials in Intramuscular Electromyography Signals (2015)
11. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH Arrhythmia Database. IEEE Eng in Med and Biol, (PMID: 11446209) , 20(3), pp. 45–50 (2001)
12. Goldberger, A.L., Amaral, L.A.N., Glass, L,, Hausdorff, J.M,, Ivanov, P,Ch,, Mark, R,G,, Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals Circulation, 101(23), pp. e215–e220 (2003)

ISSN 1870-4069

Generalización de requisitos de software de dominio específico para clasificación de texto

J. Manuel Pérez-Verdejo, Ángel Juan Sánchez-García,
Jorge Octavio Ocharán-Hernández

Universidad Veracruzana,
Facultad de Estadística e Informática,
México

manuel.vrdjo@gmail.com, {angesanchez, jocharan}@uv.mx

Resumen. Los requisitos de software representan un conjunto de características que un sistema debe satisfacer y son críticos para el desarrollo de proyectos de software. El conocimiento que representan tiene una estructura sintáctica definida y un dominio propio de cada proyecto. Sin embargo, este vocabulario específico dificulta su reutilización en otros ámbitos, incluyendo su uso para el entrenamiento de modelos de clasificación en aprendizaje automático. Motivado por estos factores, este estudio experimental propone identificar si la generalización del conocimiento propio de un dominio permite incrementar la precisión de modelos para clasificar requisitos de software en categorías de atributos de calidad. Para comparar los resultados del texto procesado contra el original, se llevó a cabo el entrenamiento de una red neuronal convolucional. Los resultados preliminares muestran que la aplicación de este procedimiento incrementa la precisión del clasificador en un 10 % a comparación del texto original.

Palabras clave: Procesamiento de lenguaje natural, requisitos de software, clasificación, redes neuronales.

Generalization of software requirements in specific domains for text classification

Abstract. Software requirements represent a set of characteristics that a system must satisfy and are critical to the development of software projects. The knowledge they represent has a defined syntactic structure and its own domain of each project. However, this specific vocabulary makes it difficult to reuse it in other areas, including its use for training classification models in machine learning. Motivated by these factors, this experimental study proposes to identify whether the generalization of a specific domain knowledge allows increasing the precision of models to classify software requirements into categories of quality attributes.

To compare the results of the processed text against the original, the training of a convolutional neural network was carried out. Preliminary results show that applying this procedure increases the accuracy of the classifier by 10% compared to the original text.

Keywords: Natural language processing, software requirements, classification, neural networks.

1. Introducción

En el proceso de desarrollo de software, los requisitos representan el eje central de las actividades a realizar en todo el ciclo de vida de un sistema. Es a partir del conocimiento plasmado en estos enunciados que se conoce de forma previa las funcionalidades y especificaciones del software. Sin embargo, el lenguaje natural es ambiguo y frecuentemente interpretable, por lo que se considera una buena práctica el uso plantillas para la redacción de requisitos de calidad [18]. La estructura sintáctica con la que se propone se redacten los requisitos se define en un enunciado del tipo "El [nombre del sistema] deberá ser capaz de [objetivo y respuesta del sistema]" [17].

Adicional a esta estructura definida, los requisitos son redactados en términos del dominio al que pertenecen. Este dominio constituye esencialmente el glosario del proyecto [13]. Los términos específicos de un dominio del conocimiento representan el vocabulario con el que se expresan estos requisitos. De esta forma, se establecen niveles de abstracción que corresponden a qué tan específicos son los requisitos, de entre los que destacan dos niveles: los requisitos específicos de un dominio, y los requisitos genéricos [15].

Este criterio al nivel de detalle también se presenta en una categoría de los requisitos de software conocida como requisitos de calidad. Este tipo de requisitos no describe una función, sino la estructura y el comportamiento con el que se realiza la misma [2]. Estos cuentan con una taxonomía definida [5], y se dividen en categorías como disponibilidad, tolerancia a fallos, mantenibilidad, rendimiento, escalabilidad, seguridad, usabilidad entre otras. Estas características deseables en el software, son conocidas como atributos de calidad.

Sin embargo, el nivel de abstracción con el que se redactan este tipo de enunciados afecta directamente su reusabilidad. Esto es, entre más específico sea el dominio de un requisito, es menos probable que se pueda reutilizar junto con otros de distintos dominios [18]. Esta variabilidad de vocabulario y contexto afecta también en tareas de automatización, principalmente en la clasificación de texto.

Es así como el presente estudio parte del supuesto que al transformar requisitos específicos de un dominio en requisitos genéricos, se puede incrementar la precisión de modelos de clasificación de requisitos, específicamente de atributos de calidad.

El resto del documento se divide de la siguiente forma: en la sección 2 se describen trabajos relacionados en el área de la clasificación de requisitos y diferentes enfoques de preprocesamiento para los mismos.

En la sección 3 se explica la propuesta de generalización empleada en la realización de este trabajo. En la sección 4 se muestran los resultados obtenidos y la comparación de desempeño a través del entrenamiento de un modelo de clasificación con el texto preprocesado y el texto sin ningún tratamiento. Finalmente, la sección 5 resume los hallazgos identificados y el trabajo futuro.

2. Trabajo relacionado

El tópico de la clasificación de requisitos de software ha sido abordado por la literatura con anterioridad dentro de área de Ingeniería de Requisitos. En [6] se prueban diferentes combinaciones de representaciones de texto y clasificadores para identificar de forma automática *app reviews* asociadas con requisitos funcionales o alguna categoría de atributo de calidad (usabilidad, confiabilidad, portabilidad y rendimiento). En este caso se exploran *Bag Of Words*, CHI^2, TF-IDF y *Word2Vec* para la representación del texto y Naïve Bayes, J48 y Bagging para su clasificación.

De la misma forma, el trabajo de Abad et. al. [1] se investiga la clasificación de requisitos utilizando enfoques como modelado de tópicos, *clustering* y clasificación bayesiana para identificar categorías de atributos de calidad. En el caso de este estudio, se realizó la redacción de reglas temporales que limitan el vocabulario complejo propio de este tipo de requisitos. Dado que en este estudio se parte de una base de datos con requisitos provenientes de distintos proyectos de software, se realiza un preprocesado con la intención de reducir la inconsistencia entre los mismos. Adicional a un proceso de etiquetado temporal, se componen reglas de expresiones específicas de un dominio que harían referencia a alguna categoría de atributos de calidad. De esta forma expresiones como *authorization* o *encrypt* se pueden asociar a la categoría de *Security*.

Otros estudios, centran su investigación en la identificación de un sólo tipo de atributo de calidad. En [11] se propone el uso de redes neuronales convolucionales para identificar reportes de fallas asociados con el atributo de calidad de Seguridad. En este estudio, se probaron diferentes configuraciones de redes neuronales y se realizó el entrenamiento con texto recolectado de repositorios de GitHub, artículos de Wikipedia y documentación propia de una empresa. Para el caso del preprocesamiento de su texto, únicamente se realizó la vectorización de sus documentos utilizando la biblioteca de *Word2Vec*.

Por otra parte, el estudio llevado a cabo por Rago, Marcos y Diaz-Pace [12] realiza una comparación de diferentes combinaciones semánticas, que permiten enriquecer la información provista en los requisitos de software. De esta forma, se usa la información dentro del dominio de los requisitos, para asociar conceptos que podrían no tener el mismo significado en otro dominio. Tal es el caso de palabras como *register* y *save*, que en el contexto de área de la Ingeniería de Software podrían referirse al mismo concepto. Su método de enriquecimiento semántico demostró una mejora de hasta un 18% en la precisión y exhaustividad de la clasificación de requisitos.

3. Propuesta

El procedimiento de generalización de dominio de los requisitos consiste en la aplicación de tres fases de etiquetado y sustitución de expresiones específicas, las cuales se muestran en la Figura 1. Para la ejecución de la propuesta se identificaron dos fuentes de ejemplos de requisitos de software. Ambos conjuntos fueron preprocesados y comparados con sus respectivas versiones originales.

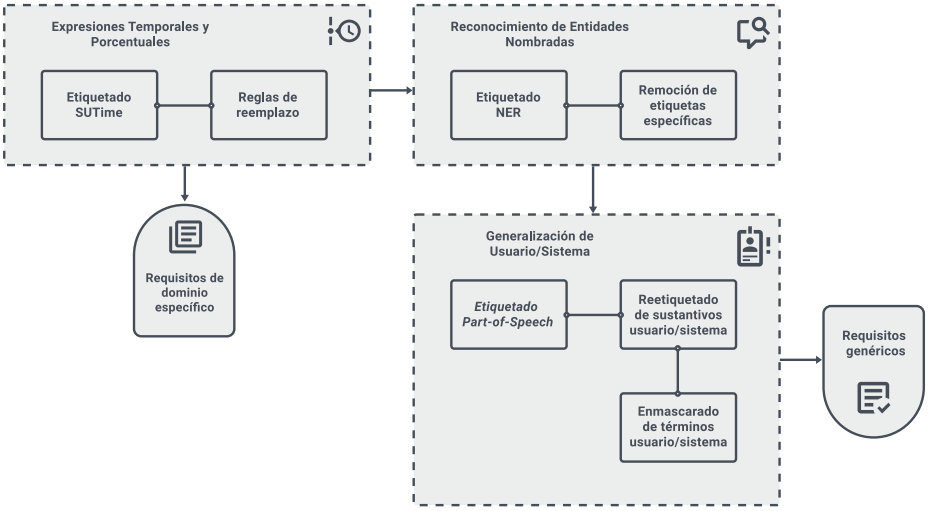


Fig. 1. Procedimiento de generalización del dominio.

3.1. Conjuntos de datos

El entrenamiento se realizó con una muestra de la base de datos de requisitos de software PROMISE [14]. Este conjunto incluye un total de 625 diferentes ejemplos de requisitos etiquetados y provenientes de 15 diferentes proyectos. Se recolectaron todos aquellos requisitos asociados a atributos de calidad. De esta forma, se identificaron 256 ejemplos de las siguientes categorías:

- Disponibilidad,
- Tolerancia a fallos,
- Mantenibilidad,
- Rendimiento,
- Escalabilidad,
- Seguridad,
- Usabilidad.

Por otra parte, para probar la precisión del modelo, se utilizaron 40 ejemplos de atributos de calidad del libro *The Quest for Software Requirements* [8]. Este conjunto de enunciados no son de proyectos en común y provienen de dominios diferentes. Únicamente se seleccionaron de las siguientes categorías que coinciden con enfoque del estudio:

- Disponibilidad,
- Mantenibilidad,
- Escalabilidad,
- Seguridad,
- Usabilidad.

3.2. Expresiones temporales y porcentuales

Con la versión en texto plano de los requisitos, se realizó un etiquetado automático en tres fases utilizando la biblioteca Stanford CoreNLP [7]. Como fase inicial, se identificaron todas las expresiones temporales (intervalos de tiempo, fechas, duraciones, etc.) utilizando la biblioteca de SUTime. [3].

Con la etiquetas generadas, se aplica una versión extendida de las reglas temporales propuestas por Abad et. al. [1] y se agregan expresiones porcentuales:

- Primera Regla: Aquellas expresiones etiquetadas como *"DURATION"* y que incluyan alguna de las siguientes expresiones serán sustituidas por *within*: *no longer than, in under, no more than, not be more than, no later, in, for less than, at a maximun.*
- Segunda Regla: Las expresiones *24 hours per day 365 days per year, 24 hour* se sustituyen por *alltimes.*
- Tercera Regla: Las expresiones de tipo *"DURATION"* y que incluyen *within* y un lapso de tiempo en minutos o segundos, se sustituyen por *fast.*
- Cuarta Regla: Las expresiones que incluyen *timely* o *quick* se sustituyen por *fast.*
- Quinta Regla: Las expresiones que contienen un porcentaje mayor a 80% y la frase *of the time* se sustituyen por *alltimes.* Aquellas que contengan un porcentaje mayor a 80% y la frase *up time* se sustituyen por *uninterrupted uptime.* Las expresiones restantes, si son de tipo *"IN"* o *"SET"* se sustituyen por *alltimes.*
- Sexta Regla: Los porcentajes iguales a 100% se sustituyen por *all*, los mayores a 50% y menores a 100% se sustituyen por *most.* Los porcentajes restantes se transforman en la expresión *little.*

Durante la ejecución de estas reglas, se pudo identificar que las categorías de atributos de calidad que más presentan expresiones de este tipo, son las de rendimiento y de disponibilidad. De esta forma, el siguiente requisito referente al atributo de calidad de rendimiento:

"The CMA report shall be returned no later 60 seconds after the user has entered the CMA report criteria"

Es transformado en la siguiente expresión, que conserva el interés en la velocidad del sistema, pero límita el detalle del tiempo en específico:

"The CMA report shall be returned fast after the user has entered the CMA report criteria"

3.3. Reconocimiento de entidades nombradas

El corpus resultante de la transformación de expresiones temporales se etiqueta utilizando el modelo NER de Stanford CoreNLP [4]. Es de esta forma que se identifican entidades que son dependientes del dominio de los requisitos. Para reducir la complejidad de estos del vocabulario, se eliminan aquellas palabras asociada con alguna de las siguientes etiquetas: *NATIONALITY, NUMBER, ORGANIZATION, COUNTRY, TIME, STATE_OR_PROVINCE.* Así, retomando el ejemplo anterior, se retirarían las siglas *CMA*, detectadas como una organización, resultando en la siguiente expresión genérica:

"The report shall be returned fast after the user has entered the report criteria"

3.4. Generalización de usuario/sistema

Finalmente se realizó un etiquetado utilizando la biblioteca de *Part-of-Speech* [16]. Este procedimiento identifica la categoría léxica que ocupa una palabra dentro de una oración (verbos, sustantivos, adjetivos, por mencionar algunos). Con las etiquetas generadas, se llevó a cabo una revisión utilizando la herramienta para anotación de texto doccano [10]. Es así como, de manera manual, se realizó el reetiquetado en el que se identificó el nombre del sistema y el nombre del usuario únicamente considerando aquellas palabras o frases que ya habían sido etiquetadas como sustantivos previamente. Con las etiquetas usuario/sistema se procedió a enmascarar esta información. Por ejemplo:

"The clinical site shall be intuitive to the Program Administrators / Nursing Staff Members"

Se reduce al enunciado:

"The system shall be intuitive to the user"

4. Experimentos y resultados

Para la evaluación de la propuesta, se realizó el entrenamiento de dos modelos de clasificación de texto utilizando la biblioteca de *deep learnig* de Ludwig [9]. Un modelo se entrenó y probó con los requisitos en su forma original, y otro con los requisitos en nivel de abstracción genérico. La características para le generación del modelo incluyen un proceso de remoción de *stopwords* y de *lemmatization.* Se usa la arquitectura de una red neuronal convolucional con el *encoder* de *parallel_cnn* incluido en la biblioteca. El entrenamiento se realiza utilizando los parámetros por defecto de 100 *epoch* y un *batch_size* de 128. La Figura 2 resume este proceso.

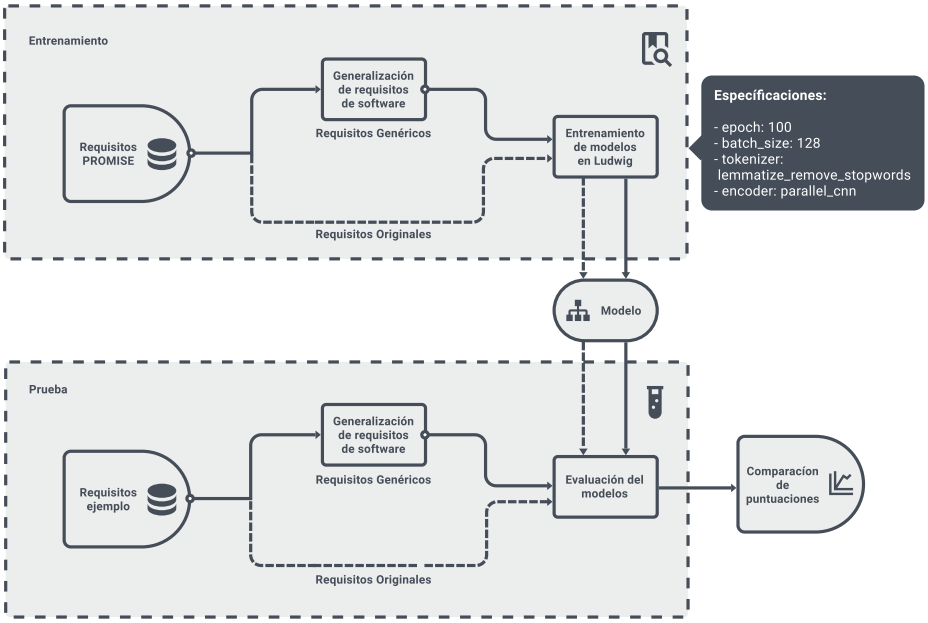


Fig. 2. Procedimiento experimental para la evaluación de la propuesta.

La comparación del desempeño de ambos modelos se muestran en la Figura 3. Se puede observar una mayor tendencia incremental por parte del modelo entrenado con requisitos genéricos, en términos de la precisión con el conjunto de prueba.

El modelo entrenado usando requisitos de dominio específico muestra evidencia de incompatibilidad al momento de predecir requisitos que están fuera de sus dominios conocidos, consiguiendo una precisión de 42.5%. De esta forma, el procedimiento de generalización logró incrementar la precisión de la clasificación en un 10%. La información detallada de ambas ejecuciones se puede consultar en el repositorio del experimento[1], así como el código y las bases de datos[2].

Adicionalmente, se realizó la matriz de confusión de ambos modelos. Se puede observar cómo el modelo entrenado con requisitos genéricos reduce la clasificación errónea de las categorías de Seguridad y de Rendimiento, sin embargo esta última no formaba parte del conjunto de prueba. A pesar de ello, ambos modelos fallaron en clasificar correctamente los atributos de calidad de Mantenibilidad, categoría que sí se encontraba incluida en el conjunto de prueba.

La reducción del error se puede observar en mayor medida en la Figura 5. Mientras que los valores de precisión y valor-F de las categorías de Disponibilidad y Seguridad incrementan considerablemente, la categoría de Usabilidad experimenta un decremento. Asḿismo, es posible observar que el procedimiento de

[1] https://comet.ml/manolomon/domain-masking
[2] http://github.com/quality-attributes/domain-masking

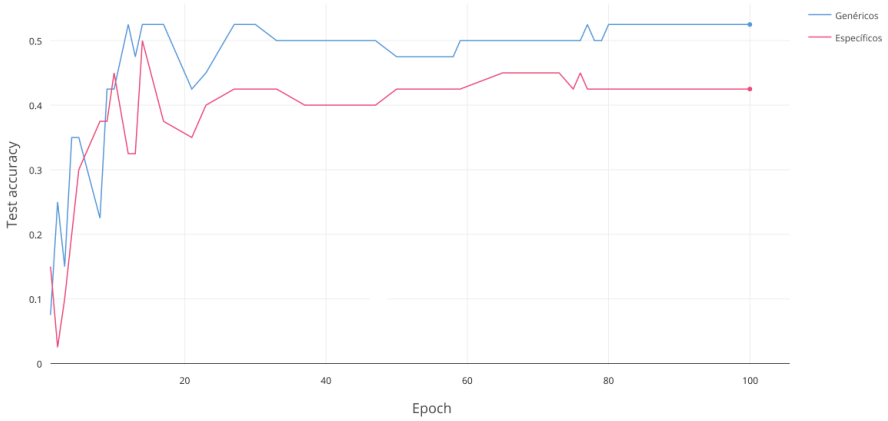


Fig. 3. Evolución a través de los ciclos (*epoch*) de la precisión del modelo en términos del conjunto de prueba.

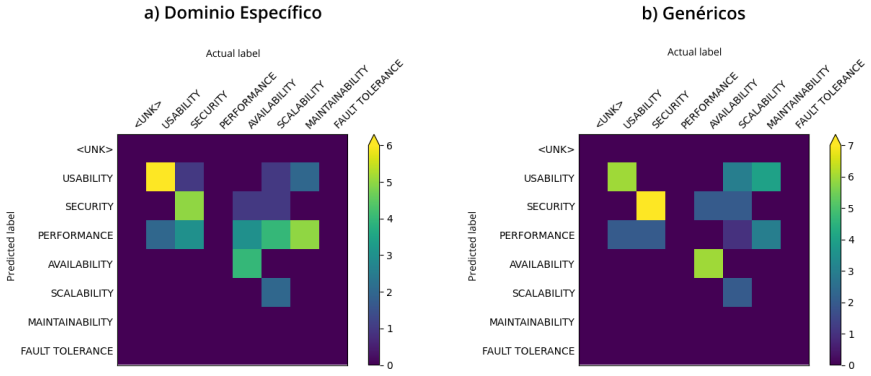


Fig. 4. Matrices de confusión de los modelos sin el procesamiento y una vez aplicado el tratamiento propuesto.

generalización no representó ningún cambio en las puntuaciones de la categoría de Escalabilidad.

5. Conclusiones y trabajo futuro

La clasificación automática de requisitos de software representa un campo de oportunidad para la colaboración de la Ingeniería de Requisitos con la Inteligencia Artificial.

Estos artefactos son críticos para el desarrollo de sistemas, y especialmente en proyectos de gran escala, representan inversiones importantes de dinero y

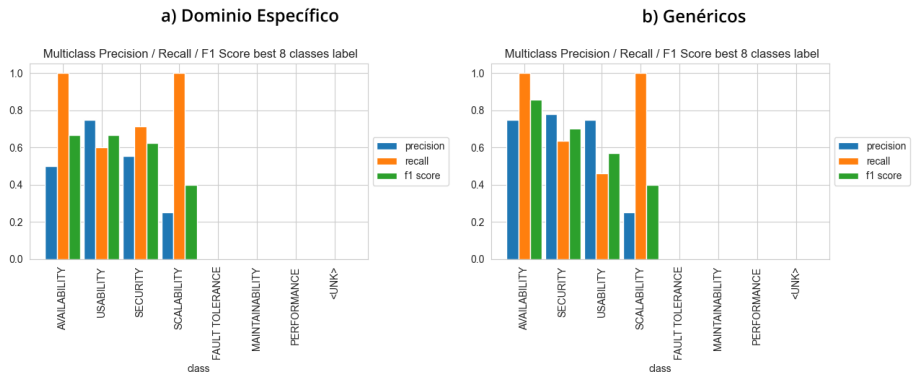


Fig. 5. Precisión, exhaustividad y valor-F de cada categoría evaluada por los modelos.

esfuerzo. Identificar una forma de reutilizar el conocimiento plasmado en los requisitos de forma automática significa una oportunidad para dar paso a algoritmos de clasificación de requisitos con menor margen de error.

Así, en este artículo se presentó una evaluación experimental con el objetivo de investigar un posible enfoque para la transformación de requisitos específicos de un dominio en requisitos genéricos. Los resultados iniciales muestran un incremento en la precisión en un 10% a comparación con el texto sin ningún tratamiento. Sin embargo, esta precisión permanece siendo baja (52.5%), por lo que se contempla explorar campos adicionales del área de estudio del Procesamiento de Lenguaje Natural, así como otras alternativas de clasificadores.

Por otra parte, se identificó que este tipo de tratamiento beneficia la clasificación acertada de ciertas categorías como el Rendimiento y la Seguridad, pero reduce la exhaustividad de otras clases como la Usabilidad.

Una vez analizados los resultados, se identifica cabida para la aplicación de técnicas adicionales, como el uso de palabras clave para agregar pesos modificados, o el uso de mejoras sintácticas. Por otro lado, tanto el entrenamiento y las pruebas de los modelos se vieron limitados por la cantidad de ejemplos disponibles, por lo que se considera la adquisición de más ejemplos documentados.

Finalmente, la fase de enmascaramiento de usuario/sistema se realizó de forma manual, convirtiendo el procedimiento propuesto en semiautomático. No obstante, se identificó como una posibilidad viable para futuros estudios, el entrenamiento de modelos de NER para ocultar esta información de forma automática.

References

1. Abad, Z.S.H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., Schneider, K.: What Works Better? A Study of Classifying Requirements. In: 2017 IEEE 25th International Requirements Engineering Conference (RE). pp. 496–501. IEEE (sep 2017)

2. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley Professional, 3rd edn. (2012)
3. Chang, A.X., Manning, C.: SUTime: A library for recognizing and normalizing time expressions. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 12). pp. 3735–3740. European Language Resources Association (ELRA), Istanbul, Turkey (2012)
4. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05. pp. 363–370. ACL '05, Association for Computational Linguistics, Morristown, NJ, USA (2005)
5. ISO/IEC/IEEE: ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering. ISO/IEC/IEEE 29148:2018(E) pp. 1–104 (2018)
6. Lu, M., Liang, P.: Automatic Classification of Non-Functional Requirements from Augmented App User Reviews. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. pp. 344–353. EASE'17, ACM, New York, NY, USA (2017), http://doi.acm.org/10.1145/3084226.3084241
7. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60. Association for Computational Linguistics, Stroudsburg, PA, USA (2014)
8. Miller, R.E.: The Quest for Software Requirements. MavenMark Books, Oconomowoc, WI, USA (2009)
9. Molino, P., Dudin, Y., Miryala, S.S.: Ludwig: a type-based declarative deep learning toolbox (2019)
10. Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., Liang, X.: doccano: Text Annotation Tool for Human (2018), https://github.com/doccano/doccano
11. Palacio, D.N., McCrystal, D., Moran, K., Bernal-Cardenas, C., Poshyvanyk, D., Shenefiel, C.: Learning to Identify Security-Related Issues Using Convolutional Neural Networks. In: Proceedings - 2019 IEEE International Conference on Software Maintenance and Evolution, ICSME 2019. pp. 140–144 (2019)
12. Rago, A., Marcos, C., Diaz-Pace, J.A.: Using semantic roles to improve text classification in the requirements domain. Language Resources and Evaluation 52(3), 801–837 (2018)
13. Rosenberg, D., Stephens, M.: Use Case Driven Object Modeling with UML. Apress, Berkeley, CA (2007)
14. Sayyad Shirabad, J., Menzies, T.J.: The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada (2005), http://promise.site.uottawa.ca/SERepository
15. Shehata, M.S., Eberlei, A., Hoover, H.J.: Requirements Reuse and Feature Interaction Management. In: Proceedings of the 15th International Conference on Software & Systems Engineering and their Applications. Paris (2002)
16. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics. EMNLP '00, vol. 13, pp. 63–70. Association for Computational Linguistics, Morristown, NJ, USA (2000)

17. Wiegers, K.E.: More About Software Requirements: Thorny Issues and Practical Advice. Microsoft Press, USA (2005)
18. Wiegers, K.E., Beatty, J.: Software Requirements. Microsoft Press, USA, 3rd edn. (2013)

ISSN 1870-4069

Sistema portátil no invasivo para el monitoreo y desplegado de los signos vitales de pacientes en salas de urgencias en un dispositivo móvil

Israel Santoyo-Luévano, Carlos Avilés-Cruz, Arturo Zuñiga-López

Universidad Autónoma Metropolitana,
Departamento de Electrónica,
División de Ciencias Básicas e Ingeniería,
México

{issl, caviles, azl}@azc.uam.mx

Resumen. En este trabajo se presenta un sistema computacional no invasivo para monitorear y desplegar signos vitales, como lo son la temperatura, saturación de oxígeno en sangre, pulsos cardíacos y presión arterial. Se utiliza una tarjeta de adquisición de datos Arduino para recibir los datos obtenidos por los sensores y enviarlos a través de la señal Bluetooth a un dispositivo móvil que los despliegue en una aplicación (App). El sistema propuesto tiene la finalidad de apoyar en las salas de urgencias, reduciendo el tiempo de valoración de los pacientes en función de sus signos vitales.

Palabras clave: Signos vitales, Arduino, teléfono celular, oxigenación en la sangre, temperatura, pulso cardíaco.

Portable Non-invasive System for Monitoring and Displaying Vital Signs of Patients in Emergency Rooms on a Mobile Device

Abstract. This work presents a non-invasive computer system to monitor and display vital signs, such as temperature, blood oxygen saturation, cardiac pulses, and blood pressure. An Arduino data acquisition card is used to receive the data obtained by the sensors and send them through the Bluetooth signal to a mobile device that displays them in an application (App). The proposed system is intended to support emergency rooms, reducing the assessment time of patients based on their vital signs.

Keywords: Vital signs, Arduino, mobile device, blood oxygen saturation, temperature, cardiac pulses.

1. Introducción

En la actualidad en las salas de emergencia en los hospitales de México se presenta una saturación en la demanda del servicio, que es producto de la falta de personal médico. En consecuencia los pacientes deben esperar más de 30 minutos desde que realizan su registro en la ventanilla de la recepción hasta su valoración médica ó toma de signos vitales (SV). En hospitales del sector público como el Instituto Mexicano de Seguro Social (IMSS), el tiempo de espera es de **51 minutos**, y en la mayoría de los hospitales estatales es dé **56.7 minutos** [1].

Esto puede propiciar que se dé el caso de que algún paciente pueda sufrir una complicación alno ser valorado desde su llegada.

La Organización Panamericana de la Salud (OPS) define a los sistemas de salud basados en la Atención Primaria de Salud (APS) como:
"un enfoque amplio de la organización de los sistemas de salud, que hace del derecho a alcanzar el mayor nivel de salud posible su principal objetivo, al tiempo que maximiza la equidad y la solidaridad del sistema".

Un sistema de *Triage* estructurado debe ser entendido como un sistema integral de atención de calidad para los cuartos de urgencias, ya que garantiza su categorización, según el grado de urgencia de los síntomas y signos vitales que presentan. El Triage tiene 4 indicadores de calidad (IC), que se muestran a continuación [2]:

1. *Índice de pacientes perdidos sin ser visitados por el médico*: El estándar establecido se sitúa en $\leq$ al 2 % del total de pacientes que acuden a urgencias.
2. *Tiempo llegada/registro*: El tiempo desde la llegada del paciente al cuarto de urgencias hasta el momento en que se inicia la clasificación. El IC de tiempo *llegada/registro*-triage se define como el porcentaje de pacientes con este tiempo $\leq$ 10 **minutos** sobre el tiempo total de pacientes clasificados. Se establece un estándar $>$ al 85 %. Complementariamente se establece un IC de tiempo de *registro-triage* $\leq$ **15 minutos**, con un estándar $>$ al 95 %.
3. *Tiempo de duración del triage*: Se recomienda que el *tiempo de duración* de la clasificación sea $\leq$ a 5 minutos en $>$ 95 %
4. *Tiempo de espera para ser visitado*: Se establece que al menos un 90 % de los pacientes tienen que ser visitados por el equipo médico en $\leq$ a 2 horas desde su clasificación y el 100 % en $\leq$ a 4 horas.

En este trabajo se propone un sistema de hardware y software que sea capaz de reducir tiempo de la valoración del paciente al realizar la toma de SV. Después de que se registren sus datos en recepción, se le colocará el sistema y los datos serán enviados mediante Bluetooth a un dispositivo móvil. Con los datos obtenidos se puede identificar el número del triage que le corresponde al paciente y así poder canalizarlo para que sea atendido de manera más eficiente y sin poner en riesgo su salud.

El sistema propuesto se puede utilizar no sólo en unidades de urgencias de hospitales, sino también en lugares como el servicio médico que se tiene dentro de las instalaciones de la universidad.

En la Fig. 1 se muestra el diagrama a bloques de funcionamiento del sistema propuesto, donde se puede ver la conexión de los sensores con la tarjeta de adquisición de datos y de transmisor bluetooth para enviar los datos al dispositivo móvil con una App que permita desplegar los SV.

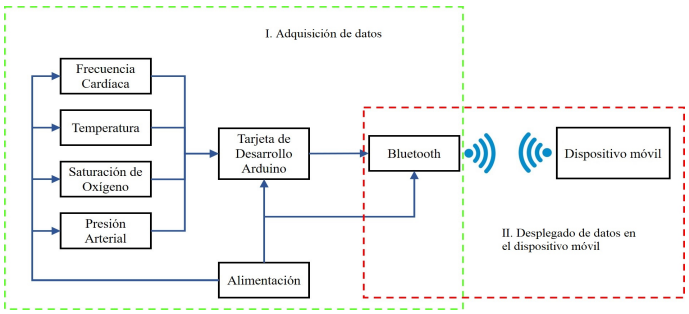


Fig. 1. Diagrama a bloques del funcionamiento del sistema.

El objetivo que se pretende lograr es que, con este dispositivo es que se pueda reducir el tiempo de espera antes mencionado a 10 minutos como lo recomienda la *OPS* [2] y así hacer más eficiente la clasificación de los pacientes según su grado de urgencia.

2. Estado del arte

Los SV son indicadores que reflejan el estado fisiológico de los órganos vitales (cerebro, corazón y pulmones), estos muestran de manera inmediata los cambios funcionales que suceden en el organismo, los cuatro SV son: la temperatura corporal, pulso, respiración y tensión arterial [3,4].

Zeid y otros muestran una aplicación en un dispositivo móvil con sistema operativo Android para desplegar los SV que son monitoreados por los médicos, ya que estos son capaces de conectar las señales recibidas de los dispositivos de varios pacientes y así recibir las lecturas directamente a su dispositivo móvil, es parecido al sistema propuesto en esté trabajo al enviar los datos a un dispositivo móvil [5].

En la Fig. 2 Yong-Jun An y otros, presentan el desarrollo de un sensor de pulso de muñeca no intrusivo para detectar señales de pulso muy débiles de las arterias principales, diseñaron una técnica que mejora de la sensibilidad utilizando un resonador de matriz de radiofrecuencia (RF), es igual que el sistema que se propone en esté trabajo al no ser invasivos y de puede usar en la muñeca [6].

En la Fig. 3 Bibb y otros muestran en desarrollaron un sistema de monitoreo de SV utilizando antenas inalámbricas basadas en microondas. Su sistema utiliza

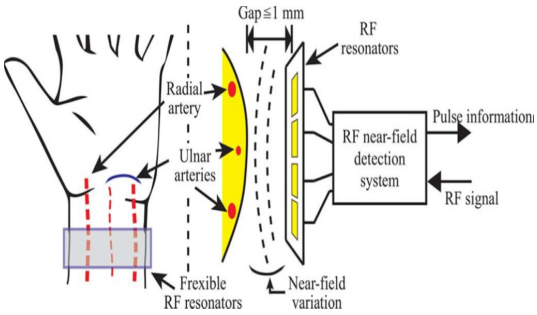


Fig. 2. Sitios del cuerpo para obtener la medición del pulso [5].

un sensor de antena de acoplamiento de energía electromagnética, equipado con una aplicación móvil para mostrar los valores procesados en tiempo real [7].

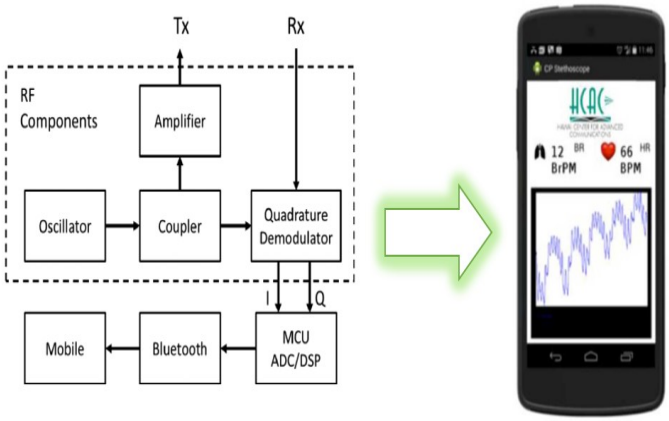


Fig. 3. Diagrama a bloques del sistema y datos desplegados en un dispositivo móvil [7].

Coultier y otros muestra un monitor de SV que utiliza los sensores de temperatura para detectar las señales bio-electricas, a las cuales se les aplican filtros pasa bajas con el OPAA277 para poder obtener la señal del sensor sin ruido [8].

Moreno y otros muestran en su trabajo un sistema para monitoreo de SV que sólo mide el ritmo cardíaco, la frecuencia respiratoria y la temperatura; con la ayuda de un algoritmo analizan las señales obtenidas para determinar los SV que están tratando de medir y utilizan un sistema basado en la tarjeta de desarrollo, su sistema es similar al propuesto, pero esté también medirá saturación de oxígeno en la sangre [9].

Heydari y otros proponen un método no invasivo basado en técnicas de tiempo de llegada del pulso que le permiten monitorear la presión arterial, investigan el cálculo de presión de sangre, basado en cinco lecturas de tiempo de llegada del pulso (PAT) diferentes usando Bio-impedancia (BImp) en el hombro como alternativa a fotopletismografía (PPG) [10].

May y otros propusieron un sistema que permite medir los SV desde el interior de la traquea, presentando una idea diferente del sitio para realizar la medición, es referente para esté trabajo propuesto, ya que se desea realizar las mediciones de los SV desde la muñeca o del antebrazo de los pacientes [11].

Hung y otros proponen un sistema que permita realizar un monitoreo post-operatorio de los SV a pacientes desde su domicilio y guardarlos en la nube, esté sistema al igual que el propuesto tiene la ventaja de poder utilizarlo desde casa [12].

J. M. May et al proponen el método de muestreo de SV en la tráquea, utilizando un sensor capaz de estimar la oxigenación en la sangre y la frecuencia del pulso, como lo muestran una opción diferente de la ubicación de lo sensores llegando a valores cercanos obtenidos con instrumental de monitores clínico [13].

3. Marco teórico

3.1. Signos vitales

Los *signos vitales SV* son los valores que permiten estimar la efectividad de la circulación, respiración y funciones neurológicas basales. Son la cuantificación de acciones fisiológicas, como **frecuencia cardíaca (FC)**, **frecuencia respiratoria (FR)**, **temperatura corporal (TC)**, **presión arterial (PA)** y **oximetría(OXM)**, que indican si un individuo está vivo. Cambian de persona a persona y en el mismo individuo en diferentes momentos del día [3,16].

Las principales variables que afectan los son:

- Edad,
- Género,
- Ejercicio físico,
- El embarazo,
- Estado emocional,
- Hormonas,
- Medicamentos.

Pulso arterial El **pulso arterial** es la onda pulsátil de la sangre, que se origina en la contracción del ventrículo izquierdo del corazón y contracción regular del calibre de las arterias, representa el rendimiento del latido cardíaco y la adaptación de las arterias. La velocidad del pulso, es decir los latidos del corazón por minuto, corresponde a la **frecuencia cardíaca**.

Las cartacerísticas que se estudian al medir el pulso arterial son [3]:

- Frecuencia: número de ondas percibidas por minuto.

- Ritmo: el ritmo es normal regular.
- Volumen o amplitud: Normal cuando el pulso se puede sentir con los dedos con facilidad.
- Elasticidad: capacidad de expansión o deformación de la pared arterial bajo la onda pulsátil. Una arteria normal, es lisa, suave y recta.

Los sitios más usados para medir el pulso son, el *carotídeo*, *radial*, *braquial* y *femoral*.

El ritmo cardíaco habitualmente es regular y con frecuencia cardíaca dentro de la normalidad es entre 60 y 100 latidos por minuto (lpm), aunque determinadas enfermedades pueden provocar que el ritmo cardíaco sea irregular, rápido o lento. En la Tabla 1 se muestran la frecuencia cardíaca típica en reposo, clasificándolas por edades [3].

Tabla 1. Frecuencia cardíaca típica en reposo.

Latidos por minuto	**Población**
100 a 160	Recién nacidos
70 a 120	Niños de 1 a 10 años
60 a 100	Niños mayores de 10 años, adultos y ancianos
40 a 60	Atletas de alto rendimiento

Presión arterial La presión arterial (**PA**) resulta de la fuerza ejercida por la columna de sangre impulsada por el corazón hacía los vasos sanguíneos, la fuerza de la sangre contra la pared arterial en la *presión sanguínea* y la resistencia opuesta por las paredes de las mismas es la *tensión arterial*. La *presión sistólica* es la presión de la sangre debida a la contracción de los ventrículos y la *presión diastólica* es la presión cuando los ventrículos se relajan. Sus valores se registran en milímetros de mercurio (**mm/Hg**). En la Tabla 2 se muestran los valores normales de la presión arterial.

Tabla 2. Valores normales de la presión sistólica y diastólica, dependiendo de las edades.

Edad	**Presión sistólica (mm/Hg)**	**Presión diastólica (mm/Hg)**
Lactante	60 - 90	30 - 62
2 años	78 - 112	48 - 78
8 años	85 - 114	52 - 85
12 años	95 - 135	58 - 88
Adulto	100 - 140	60 - 90

Algunas alteraciones de la PA para clasificar a un individuo en una categoría, se debe promediar al menos dos mediciones de PA tomadas en dos o más controles sucesivos, distintos al control inicial.

1. Hipertensión arterial (HTA): elevación de la presión vascular sanguínea, es la PA anormalmente por encima de 140 mmHg para la PAS o de 100 mmHg para la PAD en varias tomas.
2. Hipotensión arterial: Es la tensión o presión baja en la sangre, es una PA anormal baja, por debajo de 100 mmHg de la PAD. Esta condición puede no ser tan molesta como la hipertensión, a pesar de producir mareos, debilidad, lipotimia, somnolencia y malestar inespecífico asociado a la disminución del gasto cardíaco.

Saturación de oxígeno en la sangre (S_pO_2) El oxígeno es un elemento crucial para el funcionamiento de las células que componen nuestro organismo y, por lo tanto, para el funcionamiento general del mismo. La detección de los niveles reducidos de oxígeno en sangre es un *indicador clave* para identificar a los pacientes que requieren una intervención inmediata. La entrega del oxígeno a las células se realiza gracias a los sistemas respiratorio y circulatorio de nuestro cuerpo [17]. Se lleva acabo a través de la ventilación, que se compone por:

1. Inspiración o inhalación: El diafragma se contrae y desciende mientras los músculos intercostales y los pectorales amplían el volumen de los pulmones.
2. Espiración o exhalación: Las neuronas respiratorias inhiben cíclicamente a las motoras encargadas de la contracción, debido a esto los músculos se relajan, disminuyendo el volumen de los pulmones y elevando el diafragma.

En la Tabla 3 se muestra que el valor ideal de la oxigenación en la sangre es del *95 %* y las acciones a seguir en caso de que el % de oxígeno en la sangre sea menor [15].

Para calcular el grado de oxihemoglobina se hace mediante la ecuación 1 donde H_bO_2 es la hemoglobina oxigenada y H_b es la desoxigenada:

$$S_aO_2 = \frac{H_bO_2}{(H_bO_2 + H_b)}. \tag{1}$$

Algunas alteraciones de la oximetría se detectan en pacientes normosaturados (mayor de 95 %) o con leve saturación (entre 93 y 95 %), moderada (saturación entre 88 y 92 %) y grave (saturación menor de 88 %). Algunos sitios para tomar la oximetría pueden ser cualquier parte de cuerpo donde la piel sea *translucida* y con buen flujo de sangre como los **dedos de la mano** o del pie y el lóbulo de la oreja [18].

Temperatura corporal Se define como el grado de calor conservado por el equilibrio entre calor generado (*Termogénesis*) y el calor perdido (*Termólisis*) por el organismo, la temperatura corporal promedio normal de los adultos sanos,

Tabla 3. Acciones a seguir según sea el nivel del % de saturación de oxígeno en la sangre.

% Saturación	Acciones
>95 %	Ninguna acción
95 - 90 %	Tratamiento inmediato y monitoreo de la respuesta al mismo, según ésta, hacer una valoración. Los pacientes con enfermedades respiratorias crónicas toleran saturaciones en torno a estos valores.
<90 %	Enfermo grave, Hipoxia severa, Oxigenoterapia + tratamiento y traslado a hospital
<80 %	Valorar intubación y ventilación mecánica.
En niños con <92 %: remitida hospital aunque presenten mejoría con maniobras iniciales, por ser más incierta su respuesta al tratamiento.	

media medida en la *cavidad bucal*, es de **36.8 ± 0.4 °C**. Existen diferencias en los valores, de acuerdo con la zona del cuerpo en que se realice la medición, como se muestra en las Tablas 4 y 5 los valores normales de la temperatura de las personas[16].

Tabla 4. Variaciones de mediciones de la temperatura corporal en diferentes partes del cuerpo.

Zona de muestra	Medición
Bucal	36.8°C (± 0. 4°C)
Rectal	+0.3 °C
Axila	-0.6 °C

Tabla 5. Variaciones normales de temperatura.

Edad	Grados centígrados(°C)
Recién nacido	36.1- 37.7
Lactante	37.2
2-8 años	37.0
8-15 años	36.5 - 37.0
Adulto	36.4 - 37.2
Vejez	≤ 36.0

4. Metodología propuesta

Los sensores se escogieron por ser pequeños, de bajo consumo de corriente y por que comparten los mismos requerimientos de voltaje que la tarjeta de desarrollo. En el caso del sensor de temperatura MXL90614 viene calibrado de fabrica, el sensor de pulsos ADA-1093 yl sensor de oxigenación en la sangre MAX30102 tienen circuitos amplificadores de la señal y cancelación de ruido. y el modulo bluetooth se escogió por ser una señal de baja potencia y no interfiere con los dispositivos que se encuentran detrás de la sala de emergencias.

4.1. Medición de signos vitales

En la Fig. 4 se muestra el diagrama que ilustra las etapas fundamentales del sistema, la linea roja es la conexión de la batería a los sensores, la tarjeta de adquisición de datos y el módulo bluetooth, las linea amarilla y verde son la conexión para envió de datos y la transmisión inalámbrica bluetooth.

– **1er. Etapa:** Se programan los sensores (Sensor de ritmo cardíaco, de presión arterial, temperatura y el oxímetro) con la tarjeta de adquisición de datos Arduino.

– **2da. Etapa:** Se programa la tarjeta de desarrollo, Arduino con el dispositivo Bluetooth para realizar la comunicación con el dispositivo móvil (SmartPhone) o PC.

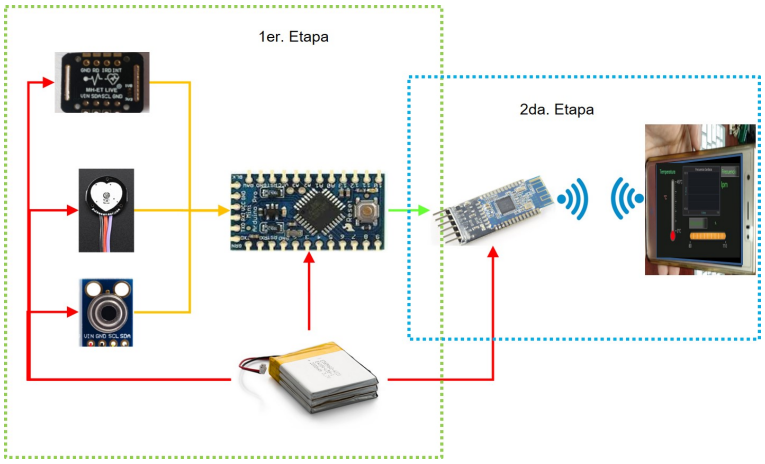


Fig. 4. Metodología utilizada para la construcción del sistema.

Como se vio en la introducción, en la Fig. 4 ahora se va a describir por partes la configuración de cada uno de los sensores, el módulo Bluetooth con la tarjeta de desarrollo Arduino.

Frecuencia cardíaca y presión arterial Para la frecuencia cardíaca se usa el **sensor de pulsos ADA-1093**, que se muestra en la Fig. 5 añade circuitos de amplificación y cancelación de ruido a su hardware y sus lecturas de pulso son confiables.

Fig. 5. Sensor de pulsos ADA-1093.

Saturación de oxígeno en la sangre (S_pO_2) El sensor **MAX30102** mide oximetría de pulso integrada que incluye LED internos, fotodetectores, elementos ópticos y electrónica de bajo ruido con rechazo de luz ambiental. El MAX30102 sus dimensiones pequeñas proporcionan una solución completa del sistema al facilitar el diseño del dispositivo, en la Fig. 6 se muestra al circuito integrado.

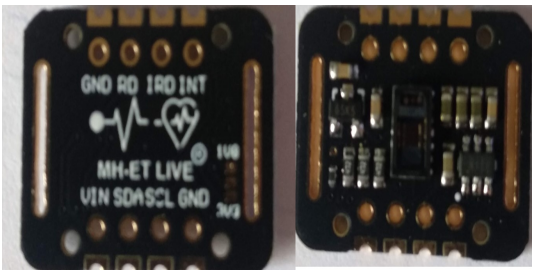


Fig. 6. Sensor MAX30102.

4.2. Temperatura

La temperatura se mide con el sensor **MLX90614**, ya que es un termómetro infrarrojo para mediciones de temperatura de contacto. Gracias a su amplificador de bajo ruido, ADC de 17 bits y potente unidad DSP, el termómetro alcanza una alta precisión y resolución.

El termómetro viene calibrado de fábrica con una salida digital PWM y SMBus (Bus de gestión del sistema). Como estándar, el PWM de 10 bits está configurado para transmitir continuamente la temperatura medida en rango de -20 $\cdots$ 120 °C, con una resolución de **salida de 0.14 °C**. En la Fig. 7 se muestra el sensor.

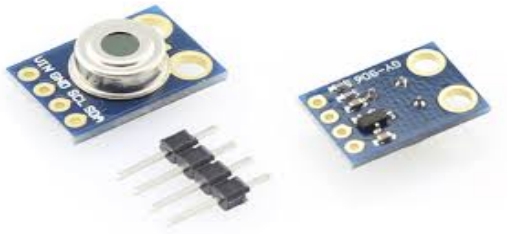


Fig. 7. Sensor MLX90614.

4.3. Módulo bluetooth

En la Fig. 8 se muestra el módulo Bluetooth **HC-05** que se va a utilizar para realizar la comunicación inalámbrica al ser un dispositivo económico y puede ser configurado como *maestro-esclavo*, ya que puede conectarse a una tablet o PC, y con otros dispositivos Bluetooth.

Fig. 8. Módulo Bluetooth HC-05.

4.4. Alimentación del sistema

En la Fig. 9 se muestra la batería de litio DTP5O3048 que se va utilizar por ser pequeña y es de 3.7 volts a 750 mAh. Se escogió esta batería ya que los sensores y la tarjeta de adquisición funcionan con 3.3 volts.

Fig. 9. Batería par alimentar al sistema.

5. Experimentación y generación de muestras

5.1. Medición de signos vitales

En la Fig. 10 se muestra el diagrama eléctrico del dispositivo propuesto, donde se puede observar los pines a los que se conecta cada sensor a la tarjeta de desarrollo.

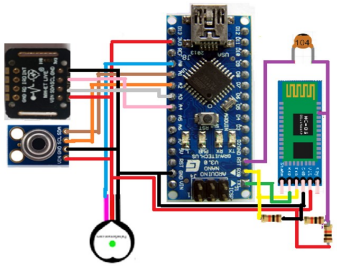
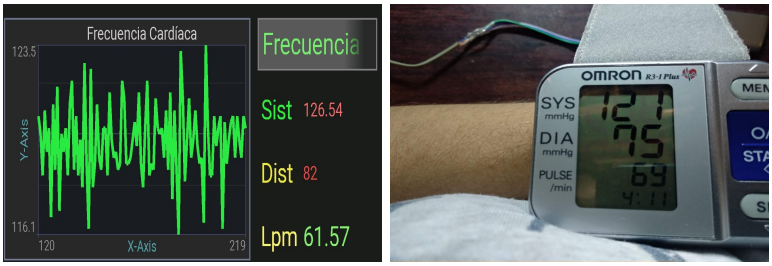


Fig. 10. Diagrama eléctrico de conexión.

Después de programar los sensores con la tarjeta de desarrollo se procedió a realizar algunas pruebas para comparar las lecturas de los sensores con sensores comerciales. En el caso del sensor de pulsos ADA-1093 se obtiene la representación de la señal como se muestra en la Fig. 11(a).

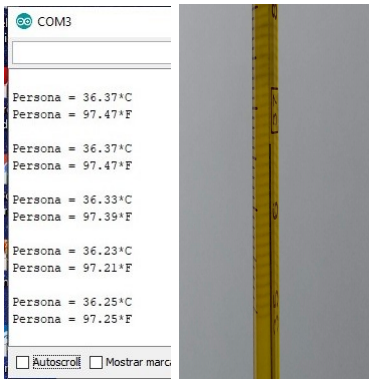
En la Fig. 11 se muestran los datos obtenidos con el sensor de pulsos ADA-1093 y un Monitor de Presión Arterial OMRON modelo R3-IPlus. En la Fig. 11(a) se muestran las lecturas del sensor de pulsos de presión arterial y en la Fig. 11(b) los obtenidos con el Monitor.

En la Fig. 12 se muestra la comparación de las lecturas obtenidas con el sensor MLX90614 y con un termómetro de mercurio y en la Fig. 12(a) se muestran las del sensor MLX90614 y en la Fig. 12(b) se muestra la lectura de un termómetro de mercurio. Como se pudo observar, el sensor MLX90614 es muy eficiente ya que el tiempo necesario para obtener una lectura estable es de 30 segundos.



(a) Lecturas del sensor ADA-1093 (b) Lectura del Monitor de Presión OMRON R3-IPlus

Fig. 11. Comparación del sensor ADA-1093 con un Monitor de Presión OMRON R3-IPlus.



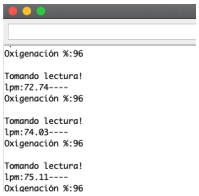
(a) Sensor MXL90614 (b) Termómetro de mercurio

Fig. 12. Comparación del sensor MLX90614 con un termómetro de mercurio.

En la Fig. 13 se muestra el comparativo de la medición de pulsos cardíacos entre el sensor MAX30102 y un Monitor de Presión Arterial OMRON modelo HEM-6131. Con el sensor MAX30102 se midió la saturación de oxígeno en la sangre y la frecuencia cardíaca.

En la Fig. 13(a), se muestran los datos obtenidos con dicho sensor, al igual que en el caso del sensor de temperatura, este es muy preciso y en 20 segundos ya se tenia una lectura estable. En la Fig. 13(b) se muestra la lectura de un Monitor de Presión Arterial, con la que podemos comparar los datos de pulsos por minuto.

Después de realizar las pruebas de control con sensores y Monitores de Presión, se procedió a colocar los sensores en una tablilla para conectarlo con el dispositivo móvil mediante bluetooth, en la Fig. 14 se muestra el dispositivo colocado en una persona.



(a) sensor MAX30102

(b) Monitor de Presión OMRON HEM-6131

Fig. 13. Comparación del sensor MAX30102 con un Monitor de Presión OMRON HEM-6131.

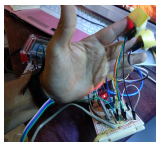


Fig. 14. Sensores en mano del paciente.

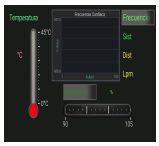


Fig. 15. Aplicación para desplegar los SV.

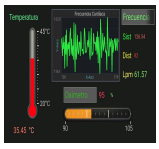


Fig. 16. App desplegando los datos de los sensores.

Con el sistema armado y colocado en la mano del paciente se procede con el desplegado de los datos en el dispositivo móvil.

5.2. Desplegado de signos vitales en dispositivo móvil

Después de comprobar el funcionamiento del sistema al ser capaz de obtener los SV se procedió a configurar la tarjeta de adquisición de datos Arduino con el módulo Bluetooth para enviar los datos al dispositivo móvil, y con la App ya configurada se utiliza para desplegar los resultados. La Fig. 15 muestra como quedó la App configurada en el dispositivo móvil.

Tras seleccionar el dispositivo bluetooth, se selecciona la pantalla donde se tienes las ventanas para desplegar las señales recibidas y en la Fig. 16 se muestra la App funcionando y desplegando los datos del sensor de temperatura, los lpm y los valores de la oxigenación en la sangre.

6. Conclusiones

Se logró exitosamente el diseño, construcción e implementación del hardware necesario para la captura de señales de signos vitales. Los sensores de temperatura, saturación de oxígeno en sangre, pulsos cardíacos y presión arterial funcionaron de acuerdo a como se esperaba. La conexión Bluetooth entre la tarjeta de adquisición de datos Arduino y el teléfono celular funcionó bien y sin problema. El despliegue de las señales en el celular fueron de acuerdo a lo que esperaba en tiempo real y sin retrasos.

Como trabajo futuro se pretende realizar el diseño de una App de Realidad Aumentada que permita reconocer al paciente mediante un algoritmo de reconocimiento facial en un dispositivo móvil. Que sea capaz de comunicarse con el sistema de medición de signos vitales y desplegar los mediante objetos virtuales sobrepuestos en la imagen del paciente.

Referencias

1. Consejo Nacional para la Evaluación de la Política del Desarrollo Social (CONEVAL): Estudio Diagnóstico del Derecho de la Salud (2018)
2. Gómez-Jiménez, J., Ramón-Pardo, P., Rua-Moncada, C.: Manual para la implementación de un sistema de triaje para los cuartos de urgencias. Organización Panamericana de la Salud (2010)
3. Corral-Quiroz, R.J., Corral-Mendoza, M.G.R., Juárez-Barrón, M.E., Ochoa-Chávez, M.L.: Signos vitales: Conocimiento y cumplimiento de técnicas de medición, Revista de Enfermeria IMSS (2006)
4. Berman, A., Snyder, S.J., Kozier, B., Erb, G.: Fundamentos de enfermería, conceptos, proceso y prácticas. Pearson, Prentice Hall (2008)
5. Zeid-Daou, R.A., Aad, E., Nakhle, F., Hayek, A., Börcsök, J.: Patient vital signs monitoring via android application. In: International Conference on Advances in Biomedical Engineering (ICABME) (2015)
6. Yong-Jun An, Byung-Hyun Kim, Gi-Ho Yun, Sung-Woo Kim, Seung-Bum Hong, Joung-Gwan Yook: Flexible non-constrained RF wrist pulse detection sensor based on array resonators. IEEE Transactions on Biomedical Circuits and Systems, 10(2) (2016)
7. Bibb, D., Perron, R.R.G., Chao-huang, G., Iskander, M.F.: Development of a wireless monitoring system for microwave-based comprehensive vital sign measurement. IEEE Antennas and Wireless Propagation Letters, 15 (2016)
8. Coultier, S., Mostes, M., Lightbody, G., Popovici, E., Fennell, W.: Low power IoT platform for vital signs monitoring. In: 28th Irish Signals and Systems Conference (ISSC)(2017)

9. Moreno, S., Quintero, A., Ochoa, C., Bonafante, M., Villareal, R., Pestana, J.: Remote monitoring system vital signs for triage and detection of anomalous patient state in the emergency room. In: 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA) (2016)
10. Heydari, F., Ebrahim, M.P., Redoute, J.M., Joe, K., Walker, K., Yuce, M.R.: A chest-based continuous cuffless blood pressure method: Estimation and evaluation using multiple body sensors. Information Fusion, 54, pp. 119–127 (2019)
11. May, J.M., Tracey-Flitchat, P., Ramaswamy, S., Snidvongs, S., Kyriacou, P.A.: A Novel Photoplethysmography Sensor for Vital Signs Monitoring from the Human Trachea. Biosensors 9(4) (2019)
12. Lun-Ping Hung, Chung-Cheng Lin: A multiple warning and smart monitoring system using wearable devices for home care. International Journal of Human-Computer Studies, 136 (2019)
13. Corral-Quiroz, R.J., Corral-Mendoza, M.G.R., Juárez-Barrón, M.E., Ochoa-Chávez, M.L.: Signos vitales: conocimiento y cumplimiento de técnicas de medición. Revista Enfermeria IMSS (2006)
14. López-Herranz, G.P.: Oximetría de pulso: A la vanguardia en la monitorización no invasiva de la oxigenación. Revista Médica del Hospital General de México, 66(3) (2003)
15. Mejía-Salas, H., Mejía-Suárez, M.: Oximetría de pulso. Revista de la Sociedad Boliviana de Pediatría, 2012, 51(2), pp. 149 (2012)
16. Flores, N.: Facultad Ciencias de la Salud Kinesiología Chillán: Guia Clinica de Control de Signos Vitales. Revista Enfermería y Primeros Auxilios (2013)
17. Berman, A., Snyder, S.J., Kozier, B., Erb, G.: Fundamentos de Enfermería, Conceptos, Proceso y Prácticas. Pearson, Prentice Hall (2008)
18. Villegas-González, J., Villegas-Arenas, O.A., Villegas-González, V.: Semiología de los Signos Vitales: Una Mirada Novedosa a un Problema Vigente. Archivos de Medicina, 12(2) pp. 221–240 (2012)

ISSN 1870-4069

Clasificación de objetos mediante momentos de Hahn 3D y aprendizaje profundo

Saúl Rivera-López, Lucia Gutiérrez-Lazcano, Horlando Vargas-Vargas, Alicia Noriega-Escamilla

Universidad Politécnica de Tulancingo,
México

{1915001,lucia.gutierrez,horlando.vargas.1631027,
alicia.noriega.173102}@upt.edu.mx

Resumen. En este trabajo se presenta un análisis con momentos ortogonales discretos clásicos de Hahn para la clasificación de objetos 3D, utilizando diferentes clasificadores y la implementación de una red neuronal eficiente como propuesta para mejorar la precisión en la clasificación de objetos. Se utiliza la base de datos SHREC11 para comparar con otros métodos existentes. Finalmente, se presenta un análisis de los resultados.

Palabras clave: Clasificación de objetos 3D, polinomios ortogonales discretos 3D, modelos de clasificación, redes neuronales.

Object Classification Using 3D Hahn Moments and Deep Learning

Abstract. This paper presents an analysis with classic Hahn discrete orthogonal moments for the classification of the 3D object using different classifiers and the implementation of a neural network as the proposal to improve precision in the object classification. SHREC11 database is used to compare with other methods. Finally, an analysis of the results is presented.

Keywords: Classification of 3D objects, discrete orthogonal polynomials, classification models, neural networks.

1. Introducción

Actualmente existe un enorme avance en la tecnología de adquisición de modelos 3D para diferentes campos científicos. Además, el desarrollo de modelos para la manipulación y procesamiento de objetos 3D están aumentando rápidamente. Por otra parte, el proceso de clasificación y recuperación de objetos 3D se ha vuelto un tema importante en diversas áreas de investigación, lo que ha llevado a desarrollar herramientas de clasificación eficientes y rápidas[2].

La representación y descripción de imágenes 3D es muy importante para la clasificación de objetos 3D debido a su capacidad de descripción.

Los momentos 2D y 3D han sido considerados herramientas para la descripción de características y se han aplicado en diferentes tareas como: reconocimiento de imágenes [10,22], análisis de imágenes [38,34,1,28], reconstrucción de imágenes [8,24], marca de agua [9,35,23], las cuales pueden ser aplicadas en áreas médicas, sistemas de seguridad, control de calidad, modelado, diseño, entre otros.

Dentro de la literatura han sido propuestos diferentes conjuntos de momentos para el análisis de imágenes, tales como momentos ortogonales discretos de Tchebichef [20], Krawtchouk [33], Hahn [36], Meixnier [18], Charlier [19] y Racah [31]. También ha surgido el interés para analizar objetos 3D mediante momentos ortogonales.

Novotni y Klein [21], introducen los invariantes de Zernike 3D para la recuperación de objetos 3D. Sin embargo, estos métodos basados en momentos radiales no pueden describir los objetos. De igual forma, Hosny y Khalid [14], muestran un algoritmo para el cálculo de los momentos de Legendre 3D, el cual presenta algunos problemas como el alto costo computacional, específicamente en ordenes altos.

Por otra parte, Yang et al.[32] proponen un algoritmo para el cómputo de momentos Gaussiano-Hermit invariantes de rotación, traslación y escala para el análisis de objetos 3D. Benouini et al. [2], presentan un nuevo conjunto de invariantes llamados momentos de Krawtchouk basados en las propiedades algebraicas de los polinomios de Krawtchouk para el reconocimiento de objetos 3D. Karmoni et al. [15], proponen el cómputo rápido de momentos de Charlier 3D mediante cubos para la reducción del tiempo computacional.

Batioua et al. [1], introducen conjuntos de momentos discretos tridimensionales separables TKKM (Momentos de Tchebichef-Krawtchouk-Krawtchouk) y TTKM (Momentos de Tchebichef-Tchebichef-Krawtchouk) en el campo del análisis de imágenes en 3D. Recientemente, Lakhili et al.[16], proponen un modelo de clasificación utilizando una red neuronal en conjunto con los momentos de Racah 3D para la clasificación de objetos.

En este trabajo se propone un análisis para la clasificación de objetos 3D utilizando los momentos ortogonales discretos de Hahn 3D como descriptores de entrada con diferentes clasificadores. Se utilizó la base de datos SHREC2011 [17] para realizar una serie de experimentos con diferentes ordenes.

El trabajo esta organizado de la siguiente manera; en la Sección 2, se presenta una visión breve del cómputo de los polinomios ortogonales de Hahn y el cálculo de los momentos para imágenes volumétricas. En la Sección 3, se muestra una breve descripción de diferentes clasificadores utilizados. En la Sección 4, se describe la propuesta de implementación del modelo de una red neuronal para la clasificación de objetos 3d. Finalmente en las Secciones 5 y 6 se presentan los resultados obtenidos de la clasificación de objetos 3D y conclusiones.

2. Momentos de Hahn 3D

Los momentos ortogonales de Hahn 3D de orden n, m, k de un objeto 3D $f(x, y, z)$ están definidos como [6]:

$$\Phi_{n,m,k} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} \tilde{h}_{\ n}^{(a,b)}(x; N) \tilde{h}_{\ m}^{(a,b)}(y; M), \\ *\tilde{h}_{\ k}^{(a,b)}(z; K) f(x, y, z), \tag{1}$$

donde los parámetros a y b están restringidos para $a > 0, b > 0$, $x = 0, 1, 2, ..., N$, y los polinomios normalizados de Hahn $\tilde{h}_{\ n}^{(a,b)}$, están definidos como:

$$\tilde{h}_{\ n}^{(a,b)}(x; N) = h_{\ n}^{(a,b)}(x; N) \sqrt{\frac{W(x)}{d_n^2}}, \tag{2}$$

donde $W(x)$ es la función de peso, d_n^2 denota el cuadrado de la norma, y $h_n^{(a,b)}(x; N)$ son los polinomios de Hahn, los cuales están definidos de la siguiente manera:

$$h_n^{(a,b)}(x; N) = \frac{(-1)^n (b+1)_n (N-n)_n}{n!} * \tag{3}$$
$${}_3F_2(-n, -x, n+1+a+b, b+1, 1-N; 1),$$
$$W(x) = \frac{\Gamma(N+a-x)\Gamma(n+b+1+x)}{\Gamma(N-n-x)\Gamma(x+1)}, \tag{4}$$
$$d_n^2 = \frac{\Gamma(a+n+1)\Gamma(b+n+1)\Gamma(a+b+n+1)_N}{(a+b+2n+1)n!(N-n-1)}, \tag{5}$$

donde $(a)_k$ es el símbolo de Pochhammer, ${}_3F_2$ es la función hipergeométrica y $\Gamma()$ es la función gamma. Desafortunadamente, la Ec. 3 causa inestabilidad numérica para ordenes mayores, para solucionar este problema se ha propuesto el uso de relaciones de recurrencia. Una forma para obtener los polinomios ortogonales discretos normalizados es mediante la siguiente relación de recurrencia [37]:

$$A\tilde{p}_{\ n}(x) = B \cdot D\tilde{p}_{\ n-1}(x) + C \cdot E\tilde{p}_{\ n-2}(x), \tag{6}$$

donde

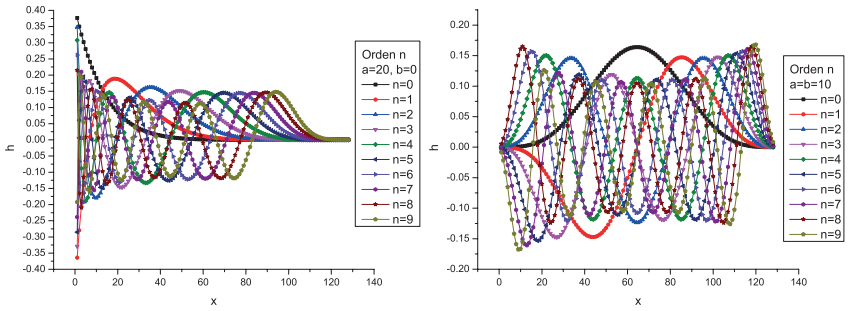
$$A = \frac{n}{(a+b+2n-1)} * \frac{(a+b+n)}{(a+b+2n)}, \tag{7}$$
$$B = x - \frac{a-b+2N-2}{4} - \frac{(b^2-a^2)(a+b+2N)}{4(a+b+2n-2)(a+b+2n)}, \tag{8}$$
$$C = -\frac{(a+n-1)(N-n+2)}{(a+b+2n-1)} \frac{(a+b+N+n-1)(N-n+1)}{(a+b+2n-1)}, \tag{9}$$
$$D = \sqrt{\frac{n(a+b+n)}{(a+n)(b+n)(a+b+n+N)}} \sqrt{\frac{(a+b+2n+1)}{(N-n)(a+b+2n-1)}}, \tag{10}$$

$$E = \sqrt{\frac{n(n-1)(a+b+n)}{(a+n)(a+n-1)(b+n)(b+n-1)(N-n+1)(N-n)}} \qquad (11)$$
$$*\sqrt{\frac{(a+b+n-1)(a+b+2n+1)}{(a+b+2n-3)(a+b+n+N)(a+b+n+N-1)}},$$

Para el cálculo inicial, los polinomios de Hahn normalizados de orden cero y de primer orden están dados por:

$$\tilde{h}_{\ 0}^{(a,b)}(x;N) = \sqrt{\frac{w(x)}{d_0^2}}, \qquad (12)$$
$$\tilde{h}_{\ 1}^{(a,b)}(x;N) = ((a+b+2)x - (b+1)(N-1))\sqrt{\frac{w(x)}{d_1^2}}.$$

La Fig. 1, muestra la gráfica de los polinomios de Hahn con diferentes valores de a y b. La concentración de la distribución de ceros de los polinomios de Hahn es donde los descriptores concentran la capacidad descriptiva.



(a) **(b)**

Fig. 1. Primeros 10 polinomios de Hahn con diferentes valores: a) $a = 20$ y $b = 0$, b) $a = b = 10$.

Una de las características más importante de los momentos ortogonales discretos 3D es de conocer la capacidad descriptiva a partir de la reconstrucción del objeto mediante un número finito de momentos.

La recuperación de un objeto 3D $f(x, y, z)$ mediante la transformación inversa de los momentos esta definida de la siguiente forma:

$$\tilde{f}\ (x,y,z) = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\sum_{z=0}^{K-1}\tilde{h}_{\ n}^{(a,b)}(x;N)\tilde{h}_{\ m}^{(a,b)}(y;M),$$
$$\times\tilde{h}_{\ k}^{(a,b)}(z;K)\Phi_{n,m,k}. \qquad (13)$$

Para evaluar la calidad de la reconstrucción de un objeto 3D mediante momentos de Hahn, es necesario utilizar la métrica del error cuadrático medio (MSE), que se emplea para medir la diferencia entre los objetos originales y reconstruidos. El MSE se define de la siguiente manera:

$$MSE = \frac{1}{NMK} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{K-1} \left(f(x,y,z) - \tilde{f}\ (x,y,z) \right)^2, \tag{14}$$

donde $f(x,y,z)$ y $\tilde{f}\ (x,y,z)$ denotan la imagen original y reconstruida respectivamente para cada voxel(x,y,z). La Fig. 2 muestra la reconstrucción de tres objetos utilizando diferentes ordenes de $n \times m \times k = 10, 15, 20$, así como el error de reconstrucción (MSE) obtenido de cada objeto.

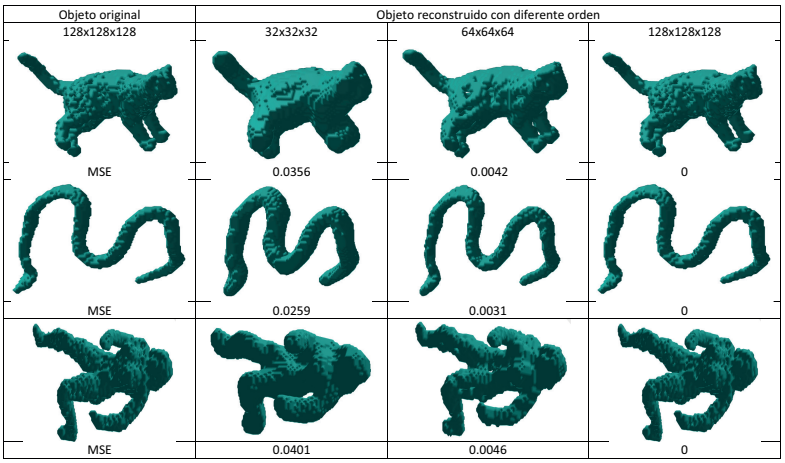


Fig. 2. Resultados de la reconstrucción de objetos 3D de diferentes clases de la base de datos SHREC11 [17].

3. Clasificación de objetos 3D

La clasificación es un área del aprendizaje supervisado en el cual un algoritmo aprende a clasificar las nuevas observaciones a partir de ejemplos de datos etiquetados y descriptores que permiten realizar entrenamiento automatizado para buscar el mejor tipo de modelo de clasificación. En la literatura existen diferentes algoritmos de clasificación que pueden ser utilizados con los momentos discretos ortogonales de Hahn 3D. Los más representativos son los siguientes algoritmos.

Análisis discriminante [**30**] El análisis discriminante es un método de clasificación supervisada que se utiliza para la clasificación de distintos datos en grupos o clases. Este método equivale a un análisis de regresión en donde la variable categórica es la clase y las variables independientes determinan a que clase pertenece un objeto, también se conoce como el discriminante Fisher, suponiendo que tiene g poblaciones conocidas $\Omega_1, ..., \Omega_g$ y en cada una de ellas se observa una muestra de cierto vector de interés $X = (X_1, ...X_p)'$. El análisis discriminante se ocupa de describir, mediante las variables X_i, los rasgos diferenciales entre las poblaciones, encontrando funciones discriminantes o reglas de decisión $h = h(X_1, ...X_p)$ cuyos valores en los distintos grupos o poblaciones estén lo mas separado posible que permitan asignar a cada individuo a una población concreta $\Omega_\alpha, \alpha = 1, ..., g$, minimizando la tasa de error en dicha asignación.

Árboles de decisión [**3**] Los árboles de clasificación y regresión son métodos de clasificación supervisada y de aprendizaje automático, los cuales son utilizados para construir modelos de predicción a partir de los datos de entrada, se basan en una estructura en forma de árbol, donde cada hoja del árbol se marca con una clase o una distribución de probabilidad sobre las clases, las ramificaciones se generan de forma recursiva hasta que se cumplan ciertos criterios, de acuerdo a la cantidad de ramificaciones se termina la profundidad y flexibilidad en árbol grueso para pocas hojas para hacer distinciones baja entre clases con un máximo de 4 máximo de divisiones, árbol mediano con un numero medio de hojas para distinciones más finas entre clases con 20 divisiones como máximo y árbol fino utiliza un numero mayor de hojas para la distinción entre clases mayor a 20 divisiones.

K-NN (vecinos más cercanos) [**13**] El método de vecinos más cercanos es un método de clasificación supervisada, el cual almacena los datos disponibles, y clasifica un objeto según una medida de similitud dada por una función de distancia. La clasificación KNN se realiza midiendo la distancia entre el objeto con los objetos conocidos. Es un método, que puede utilizarse para problemas de predicción tanto de clasificación como de regresión. Este algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro, es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que debe pertenecer. Este grupo será el de mayor frecuencia con menores distancias. Para variables continuas las funciones distancia a utilizar son, euclidiana, manhat, Minkowski, en el caso de variables categóricas se debe usar la distancia de Hamming. Un clasificador KNN presenta diferentes configuraciones de acuerdo al número de vecinos: grueso para un número k=100, mediano para k=10 o fino con k=1 y la métrica de distancia puede utilizar métricas como coseno, cúbica y ponderado.

Máquinas de soporte vectorial [**29**] Una máquina de soporte vectorial es un método de aprendizaje supervisado para la clasificación que genera funciones

de mapeo de entrada-salida a partir de un conjunto de datos de entrenamiento etiquetados, los cuales utilizan un hiper plano de separación lineal para crear un clasificador con un margen máximo. Dado un conjunto de entrenamiento de tamaño N compuesto de pares atributo-etiqueta $(x_i, y_1), 1 \leq i \leq N$, siendo $x_i \in \mathbb{R}^n$ Y $y_i \in -1, 1$, se desea obtener una ecuación para un hiperplano que divida dicho conjunto de entrenamiento, de manera que aquellos puntos con igual etiqueta queden del mismo lado del hiperplano. Si existe un hiperplano que aísle los puntos positivos de los negativos, se dice que los datos son linealmente separables. El hiperplano que separa los datos es: $\pi : w \cdot x + b = 0$, donde $w \in \mathbb{R}^d$ es un vector normal (vector perpendicular) al hiperplano π,$\mid b \mid / \mid\mid w \mid\mid$ la distancia perpendicular del hiperplano π al origen, y $\mid\mid w \mid\mid$ es la norma de w . Las máquinas de vectores de soporte pertenecen a una clase de algoritmos de Machine Learning denominados métodos kernel: lineal, cuadrático, cúbico y Gausiano.

Ensamble [5] El método ensamble o de conjunto son algoritmos de aprendizaje que construyen un conjunto de clasificadores y luego clasifican nuevos datos de entrada tomando un voto ponderado de sus predicciones. Se han desarrollado muchos métodos para construir conjuntos mediante arboles, vecinos cercanos y discriminantes.

Para analizar el rendimiento de clasificación de cada uno de los modelos mencionados, se utiliza el conjunto de imágenes 3D de la base de datos SHREC 2011, la cual consta de 600 objetos 3D clasificados en 30 clases con 20 modelos STL [17]. Las distintas clases de la base da datos SHREC11 se muestran en la Fig. 3.

Es necesario utilizar el formato voxelizado de todos los objetos de la base de datos para el cómputo de los descriptores tipo momento. Para obtener la forma voxelizada de cada objeto 3D del conjunto de datos está encerrado en el cubo delimitador más pequeño, que luego se divide en un conjunto de vóxeles con forma de cubo iguales. En este trabajo, el tamaño del voxel que se utiliza para todos los objetos de la base de datos cuenta con una resolución de $128 \times 128 \times 128$ voxeles.

El proceso de clasificación para cada modelo, se realiza mediante la técnica de validación cruzada K-folds con k=10 tomando como métrica el valor de precisión o accuracy y el cálculo de momentos de Hahn de la base de datos con parámetros de $a = b = 10$ como descriptores de entrada. Los resultados de la clasificación se muestran en la Tabla 1.

4. Implementación de modelo de una red neuronal

Las redes neuronales son un modelo computacional inspirado en el comportamiento biológico observado de las neuronas y en cómo se organizan formando la estructura del cerebro. Recientemente, las redes neuronales profundas han demostrado un gran éxito en muchas tareas de visión por computadora, específicamente en la aplicación de reconocimiento de objetos[16]. El diseño

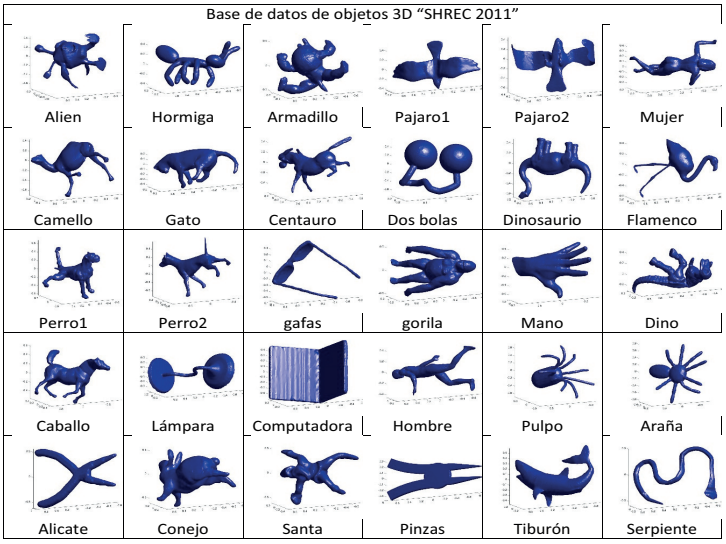


Fig. 3. Imágenes 3D de las diferentes clases de la base de datos SHREC11.

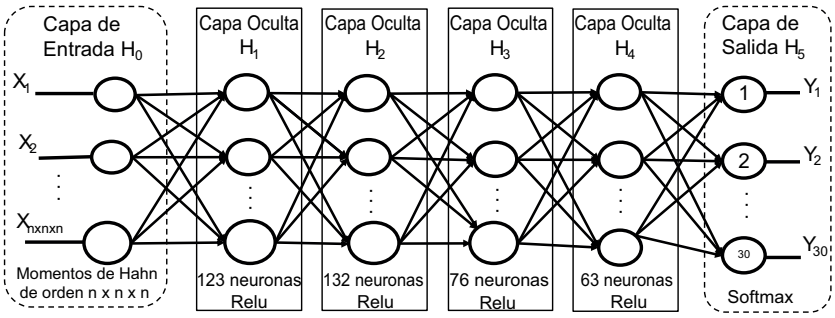


Fig. 4. Representación gráfica de la red neuronal utilizada para la clasificación de objetos.

de arquitecturas de redes neuronales puede ser un problema; si se utiliza una cantidad insuficiente de neuronas puede proporcionar tasas de clasificación insatisfactorias y por lo contrario, si se elige demasiadas neuronas aumenta el tiempo de entrenamiento y provoca un sobre ajuste, el cual disminuye la capacidad de generalización[11]. Un modelo de red neuronal está compuesta por tres capas principales; la capa de entrada, las capas ocultas y la capa de salida.

El proceso de aprendizaje consiste en encontrar los valores óptimos de los pesos y sesgos de los nodos de cada capa de tal forma que la salida corresponda al patrón de entrada con la precisión deseada, mediante el uso algoritmos optimizadores y funciones de activación. La función Softmax es utilizada para modelos

Tabla 1. Comparación de diferentes clasificadores utilizando como vector de entrada distintos ordenes de los momentos de Hahn con valores de parámetros a=10, b=10.

Modelos de Clasificación	**Orden n x m x k**				
	25x25x25	**20x20x20**	**15x15x15**	**10x10x10**	**5x5x5**
Arboles de decisión					
Árbol fino (100 divisiones)	24.3 %	28.2 %	25.2 %	28.0 %	37.7 %
Árbol mediano (20 divisiones)	18.7 %	22.2 %	19.0 %	20.3 %	27.2 %
Árbol grueso (4 divisiones)	12.0 %	12.3 %	12.5 %	12.8 %	12.3 %
Análisis discriminante					
Discriminante Lineal	22.5 %	38.0 %	40.3 %	31.2 %	45.3 %
Máquina de soporte vectorial (SVM)					
SVM Lineal	23.4 %	31.5 %	39.0 %	43.3 %	48.2 %
SVM Cuadrático	20.8 %	32.7 %	38.2 %	46.0 %	56.2 %
SVM Cúbico	21.1 %	25.7 %	32.5 %	43.2 %	55.5 %
SVM Gaussiano fino **(escala=1.9)**	5.2 %	7.3 %	9.2 %	10.5 %	26.3 %
SVM gaussiano medio **(escala=7.7)**	30.7 %	39.7 %	46.2 %	49.7 %	57.8 %
SVM gaussiano grueso **(escala=31)**	23.9 %	31.8 %	38.0 %	43.0 %	38.8 %
Clasificador vecinos cercanos (KNN)					
KNN Fino (k=1)	23. 8 %	29.0 %	34.0 %	38.5 %	55.8 %
KNN Mediano (k=10)	8.3 %	10.2 %	11.2 %	16.7 %	43.8 %
KNN Grueso (k=100)	3.7 %	4.2 %	3.2 %	4.3 %	22.7 %
KNN coseno	34.0 %	35.8 %	37.5 %	41.5 %	41.3 %
KNN cubico	6.7 %	9.3 %	10.5 %	16.3 %	42.0 %
KNN Ponderado	11.5 %	13.3 %	15.2 %	22.5 %	52.8 %
Clasificador ensamble					
Árboles potenciados	23.2 %	25.4 %	27.9 %	31.3 %	39.0 %
Subespacio Discriminante	19.2 %	23.3 %	25. 8 %	30.0 %	53.0 %
Subespacio KNN	40.5 %	44.8 %	48.7 %	57.8 %	65.3 %

de clasificación múltiple, normaliza el vector de entrada en una distribución de probabilidad donde la suma total es uno. En general, esta función devuelve las probabilidades de cada clase sobre todas las clases.

En este trabajo se propone utilizar como vector de entrada los momentos ortogonales discretos 3D a una red neuronal compuesta por seis capas distintas: una capa de entrada, cuatro capas ocultas y una capa de salida.

La Fig. 4, muestra el diagrama de las características utilizadas en la estructura de la red neuronal implementada en el modelo de clasificación propuesto.

La Ec. (15) muestra los valores de entrada en la primer capa con los valores obtenidos del cálculo de momentos 3D de Hahn de orden $n \times m \times k$:

$$V = [\Phi_{n,m,k} | n \times m \times k]. \tag{15}$$

En las capas ocultas se utiliza una estructura compuestas por 125, 135, 76, 63 neuronas respectivamente, se representan como:

$$Y_{Hi} = \phi_{Hi}(b^{Hi} + W^{Hi} \times Y_{Hi-1}), \tag{16}$$

donde $Y_{H0} = V$ es un vector de entrada para $\neq 0$,Y_{Hi} es el vector de salida de la i-esima capa oculta Hi, W^{Hi} es la matriz de pesos que conecta las capas $Hi-1$ y Hi, $b^(Hi)$ es en vector de bias y $\phi_H i$ es la función de activación utilizada en cada capa Hi. La función de activación utilizada para las capas ocultas es la función RELU definida como:

$$f(x) = max(0, x). \tag{17}$$

Finalmente la capa de salida está compuesta por el número de neuronas equivalente al número de clases contenida por la base de datos y emplea la función de activación Softmax representada matemáticamente como:

$$Softmax(y^i) = \frac{e^{y^i}}{\sum_{i=1}^{s} e^{y^i}}. \tag{18}$$

Tabla 2. Porcentajes de clasificación obtenidos utilizando diferentes tamaño de la base de datos SHREC11 para entrenamiento y distintos ordenes de momentos de Hahn con parámetros a =10, b=10.

Momentos de orden n x m x k	**80 % de la base de datos**	**50 % de la base de datos**
4 x 4 x 4	99.76 %	96.64 %
5 x 5 x 5	99.74 %	96.62 %
10 x 10 x 10	99.58 %	96.51 %
15 x 15 x 15	99.36 %	95.88 %
20 x 20 x 20	99.26 %	95.43 %
25 x 25 x 25	99.06 %	95.39 %

5. Experimentos y resultados

En esta sección se realiza un conjunto de experimentos para evaluar el desempeño en la clasificación del modelo de red neuronal propuesto cambiando el

número de descriptores. También, se comprueba la superioridad del método propuesto comparando los porcentajes de clasificación obtenidos al variar la cantidad de datos utilizados durante el proceso de entrenamiento en comparación con los métodos existentes basados en algoritmos de parametrización, obtención de características como puntos de interés, regiones significativas y redes residuales, que han sido propuestos en literatura para la clasificación en [4,21,27,12,25,26,7].

Tabla 3. Comparación de resultados de clasificación de la base de datos con otros métodos.

Método	**50 % de la base de datos**	**80 % de la base de datos**
SG [4]	62.6 %	70.8 %
Zer [21]	43.3 %	50.8 %
LFD [7]	56.7 %	65.8 %
SN [27]	52.7 %	48.4 %
Conf [12]	60.6 %	85.0 %
Sph [25]	59.0 %	82.5 %
GI [26]	88.6 %	96.6 %
DNN propuesta	99.64 %	99.76 %

Durante el primer experimento, se realiza la clasificación de la base de datos tomando dos casos de pruebas mediante la separación de la base de datos en dos grupos de forma aleatoria utilizados en métodos existentes; en el primer caso, se utiliza un total de 480 muestras para el entrenamiento de la base de datos equivalente al 80 % del total de los objetos, mientras que las 120 muestras restantes igual al 20 % se utilizan como objetos de prueba; en el segundo caso, se utiliza para el entrenamiento de la base de datos el 50 % para el entrenamiento y el 50 % restante se utiliza como objetos de prueba.

Para el segundo experimento, se calcularon los momentos de Hahn 3D con diferentes ordenes de $n = m = k = 5, 10, 15, 20, 25$, utilizando como parámetros $a = b = 10$ para el cómputo de los polinomios de Hahn, los cuales fueron utilizados como valores de entrada para la red neuronal implementada.

Los resultados obtenidos del porcentaje la clasificación de la base de datos SHREC11 utilizando como métrica el valor de precisión (accuracy) se observan en la Tabla 2.

De acuerdo con los resultados obtenidos de la Tabla 2, la capacidad de descripción de los momentos de Hahn para ordenes bajos, mejora la capacidad de obtener las características más importantes de cada objeto con un porcentaje de clasificación alto. Además, se realiza un número menor de operaciones durante el entrenamiento.

Por otra parte, la Tabla 3 muestra que la clasificación mediante redes neuronales utiliza la misma cantidad datos que otros métodos y obtiene una mejor precisión durante la clasificación.

6. Conclusiones

En este trabajo, se presentó un análisis del desempeño de clasificadores como árboles de decisión, análisis discriminante, vecinos cercanos, máquina de soporte vectorial y ensamble para los momentos ortogonales de Hahn 3D. Los resultados obtenidos a partir de la clasificación de la base de datos SHREC11 demuestran que la clasificación basada en redes neuronales y momentos de Hahn 3D tiene un mejor porcentaje de clasificación en comparación con diferentes modelos clasificación. Además, como se muestra en la Tabla 3, el método propuesto tiene un buen desempeño en comparación con los métodos existentes. El trabajo presentado ayudará a realizar nuevas arquitecturas de redes neuronales que aprovechen la capacidad descriptiva de los momentos de Hahn 3D de bajo orden.

Referencias

1. Batioua, I., Benouini, R., Zenkouar, K., Zahi, A., et al.: 3d image analysis by separable discrete orthogonal moments based on krawtchouk and tchebichef polynomials. Pattern Recognition 71, 264–277 (2017)
2. Benouini, R., Batioua, I., Zenkouar, K., Najah, S., Qjidaa, H.: Efficient 3d object classification by using direct krawtchouk moment invariants. Multimedia Tools and Applications 77(20), 27517–27542 (2018)
3. Breiman, L., Friedman, J., Stone, C., Olshen, R.: Classification and regression trees; 2017
4. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: Geometric words and expressions for invariant shape retrieval. ACM Transactions on Graphics (TOG) 30(1), 1–20 (2011)
5. Cadenas, J.M., Garrido, M.C., Daz-Valladares, R.: Mejorando el comportamiento de ensambles basados en boosting, bagging y random forest mediante soft computing. In: Proc. Novena Conf. Iberoam. en Sistemas, Ciberntica e Informtica. pp. 1–10 (2010)
6. Canterakis, N.: 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. In: In 11th Scandinavian Conf. on Image Analysis (1999)
7. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3d model retrieval. In: Computer Graphics Forum. vol. 22(3), pp. 223–232. Wiley Online Library (2003)
8. Dai, X., Shu, H., Luo, L., Han, G.N., Coatrieux, J.L.: Reconstruction of tomographic images from limited range projections using discrete radon transform and tchebichef moments. Pattern Recognition 43(3), 1152–1164 (2010)
9. Deng, C., Gao, X., Li, X., Tao, D.: A local tchebichef moments-based robust image watermarking. Signal Processing 89(8), 1531–1539 (2009)
10. Flusser, J., Zitova, B., Suk, T.: Moments and moment invariants in pattern recognition. John Wiley & Sons (2009)
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
12. Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., Yau, S.T.: Genus zero surface conformal mapping and its application to brain surface mapping. IEEE Transactions on Medical Imaging 23(8), 949–958 (2004)

13. He, Q.P., Wang, J.: Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. IEEE Transactions on Semiconductor Manufacturing 20(4), 345–354 (2007)
14. Hosny, K.M.: Fast and low-complexity method for exact computation of 3d legendre moments. Pattern Recognition Letters 32(9), 1305–1314 (2011)
15. Karmouni, H., Jahid, T., Sayyouri, M., Hmimid, A., Qjidaa, H.: Fast reconstruction of 3d images using charlier discrete orthogonal moments. Circuits, Systems, and Signal Processing 38(8), 3715–3742 (2019)
16. Lakhili, Z., El Alami, A., Mesbah, A., Berrahou, A., Qjidaa, H.: Deformable 3d shape classification using 3d racah moments and deep neural networks. Proceeding Computer Science 148, 12–20 (2019)
17. Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavou, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D.: SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes. In: Eurographics Workshop on 3D Object Retrieval. pp. 79–88 (2011)
18. Lorente, A.S., Moreno-Roque, E., Martí-Zamora, J.: Classical orthogonal polynomials in a discrete variable, their history, extensions and applications. Lect. Mat 35, 5–24 (2014)
19. Mikhaĭlov, G.A.: Optimization of weighted Monte Carlo methods. Springer (1992)
20. Mukundan, R., Ong, S., Lee, P.A.: Image analysis by tchebichef moments. IEEE Transactions on image Processing 10(9), 1357–1364 (2001)
21. Novotni, M., Klein, R.: Shape retrieval using 3d zernike descriptors. Computer-Aided Design 36(11), 1047–1062 (2004)
22. Papakostas, G.: Moments and moment invariants: theory and applications. Science Gate 1, 3–32 (2014)
23. Rivera-López, J., Bello, C.C.: Efficient encryption of image data in video sequences using discrete orthogonal moments. In: Applications of Digital Image Processing XL. vol. 10396, p. 103962Z. International Society for Optics and Photonics (2017)
24. Rivera-López, J., Camacho-Bello, C., et al.: Color image reconstruction by discrete orthogonal moment. Journal of Data Analysis and Information Processing 5(4), 156–166 (2017)
25. Shen, L., Makedon, F.: Spherical mapping for processing of 3d closed surfaces. Image and Vision Computing 24(7), 743–761 (2006)
26. Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. In: European Conference on Computer Vision. pp. 223–240. Springer (2016)
27. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 567–576 (2015)
28. Uchaev, D.V., Uchaev, D.V., Malinnikov, V.A.: Chebyshev-based technique for automated restoration of digital copies of faded photographic prints. Journal of electronic imaging 26(1), 011024 (2017)
29. Wang, L.: Support vector machines: theory and applications, vol. 177. Springer Science & Business Media (2005)
30. Wang, M., Ruan, Q.: Palmprint recognition based on two-dimensional methods. In: 2006 8th International Conference on Signal Processing. vol. 4 (2006)
31. Wu, Y., Liao, S.: Chinese characters recognition via racah moments. In: International Conference on Audio, Language and Image Processing. pp. 691–694 (2014)
32. Yang, B., Flusser, J., Suk, T.: 3d rotation invariants of gaussian–hermite moments. Pattern Recognition Letters 54, 18–26 (2015)

33. Yap, P.T., Paramesran, R., Ong, S.H.: Image analysis by krawtchouk moments. IEEE Transactions on image processing 12(11), 1367–1377 (2003)
34. Yap, P.T., Paramesran, R., Ong, S.H.: Image analysis using hahn moments. IEEE transactions on pattern analysis and machine intelligence 29(11), 2057–2062 (2007)
35. Zhang, L., Qian, G.b., Xiao, W.w., Ji, Z.: Geometric invariant blind image watermarking by invariant tchebichef moments. Optics Express 15(5), 2251–2261 (2007)
36. Zhou, J., Shu, H., Zhu, H., Toumoulin, C., Luo, L.: Image analysis by discrete orthogonal hahn moments. In: International Conference Image Analysis and Recognition. pp. 524–531. Springer (2005)
37. Zhu, H., Liu, M., Shu, H., Zhang, H., Luo, L.: General form for obtaining discrete orthogonal moments. IET Image Processing 4(5), 335–352 (2010)
38. Zhu, H., Shu, H., Liang, J., Luo, L., Coatrieux, J.L.: Image analysis by discrete orthogonal racah moments. Signal Processing 87(4), 687–708 (2007)

ISSN 1870-4069

Momentos de Bessel-Fourier de orden fraccional para el análisis de imágenes a color

Horlando Vargas-Vargas, César Camacho-Bello, José S. Rivera-López, Alicia Noriega-Escamilla

Universidad Politécnica de Tulancingo, México

{horlando.vargas.1631027,cesar.camacho,1915001, alicia.noriega.1731021}@upt.edu.mx

Resumen. En este trabajo se presenta un nuevo conjunto de momentos ortogonales de orden fraccional definidos en coordenadas polares para analizar imágenes en color y en escala de grises. Se propone como kernel una generalización mediante el orden fraccional de las funciones Bessel de primer orden. También, se analiza las propiedades y características de la nueva familia de momentos propuesta. Finalmente, se presentan los resultados experimentales para demostrar el rendimiento de los descriptores propuestos en comparación a los existentes.

Palabras clave: Función Bessel, momentos ortogonales, momentos de quaternion.

Bessel-Fourier Moments of Fractional Order for Color Image Analysis

Abstract. This work presents a new set of fractional order orthogonal moments defined in polar coordinates to analyze color and grayscale images. We propose a new kernel through the fractional order of the first-order Bessel functions. Also, the properties and characteristics of the proposed new family of moments are analyzed. Finally, the experimental results are presented to demonstrate the performance of the proposed descriptors compared to the existing ones.

Keywords: Bessel function, orthogonal moments, quaternion moments.

1. Introducción

Los momentos ortogonales son usados como descriptores en muchas aplicaciones de visión por computadora [4]. Teague en 1980 [20], introdujo los momentos a partir de una función de base ortogonal, los cuales tienen la capacidad de caracterizar información con una mínima redundancia. Los momentos ortogonales se pueden definir en coordenadas cartesianas o polares.

Por otra parte, los momentos en coordenadas polares o momentos ortogonales circulares tienen la característica de ser invariantes a la rotación, escalado y traslación; propiedades esenciales en aplicaciones de reconocimiento de patrones. Además, de su capacidad de reconstrucción de una imagen con un mínimo orden. Dentro de los momentos ortogonales circulares se encuentran: los momentos de Zernike [18], momentos pseudo-Zernike [3], los momentos ortogonales de Mellin-Fourier [16], momentos de Chebychev-Fourier [12], momentos de Jacobi-Fourier [11], momentos desplazados de Legendre-Fourier [23], los momentos exponenciales de Fourier [7], momentos armónicos de Fourier armónico [13] entre otros.

En años recientes, Bin Xiao et al. en el 2010 [22] propone un nuevo conjunto de momentos basados en la función Bessel de primer tipo, llamados momentos ortogonales de Bessel-Fourier (MOBF), los cuales, tienen la capacidad de reconstruir imágenes en condiciones de distorsión ruidosas, libres de ruido en comparación con los momentos de Mellin-Fourier y los momentos de Zernike. Años más tarde, Z Shao et al[15]. introducen los momentos de Quaternion Bessel-Fourier para el análisis de imágenes a color y para el reconocimiento de patrones mediante el manejo simultaneo de los tres canales de color. De igual forma, estos descriptores tienen la capacidad de estimar el ángulo de rotación de una imagen digital y son más robustos contra las transformaciones geométricas y fotométricas [15].

Los polinomios ortogonales de orden fraccional han sido utilizados como kernels de los momentos ortogonales de orden fraccional para el análisis de imágenes. Bin Xiao [21] propone los momentos desplazados de Legendre de orden fraccional para la extracción de características de una región de interés, por otra, J. Yang et al. [24] proponen los momentos de Zernike de orden fraccional invariantes a la rotación y robustos al ruido, al mismo tiempo H, Zhang et al. [25] definen a los momentos Mellin-Fourier de orden fraccional para el reconocimiento de patrones.

En el 2020, R Benouini et al. [1] presenta los momentos ortogonales de Chebychev de orden fraccional para el análisis de imágenes en escala de grises. Recientemente, B Chen et al. [2] introdujeron los momentos ortogonales de Zernike de orden fraccional para el procesamiento de imgénes a color para la detección de falsificación de copia-movimiento de imagen de color robusta. Finalmente, KM Hosny et al. [6] propone un nuevo conjunto de transformaciones armónicas polares fraccionales para el análisis de imágenes en color y escala de grises invariantes a la rotación, a la escala, a la traslación y robustos al ruido.

En este trabajo se propone la generalización de los momentos ortogonales de Bessel-Fourier para el análisis de imágenes en escala de grises y color. Esta nueva definición implica la introducción de un parámetro fraccional, con el fin de generalizar la noción de orden entero n. Se realizan experimentos tales como; reconstrucción de imágenes a color y la clasificación de la base datos de COIL-100 utilizando el clasificador kNN ensamble, con la finalidad de analizar la capacidad descriptiva de los MOBFOF. Finalmente se compara el rendimiento de MOBFOF con los momentos que utilizan como kernel las funciones Bessel.

2. Momentos de Bessel-Fourier

Bin Xiao [22] propone los momentos de Bessel-Fourier, los cuales utilizan como kernel radial la función de Bessel de primera clase definida de la siguiente manera:

$$J_v(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{v+2k}}{k!(v+k)!}. \tag{1}$$

Se puede representar en términos de funciones hipergeométricas como:

$$J_v(x) = \frac{(x/2)^v}{\Gamma(v+1)} {}_0F_1(v+1, -(x/2)^2), \tag{2}$$

donde v es una constante real, $\Gamma(a)$ es la función gamma, ${}_0F_1$ es la función hipergeométrica generalizada.

Los momentos de Bessel-Fourier de orden n con repetición m para una imagen en coordenadas polares $f(r, \theta)$, se definen como:

$$B_{n,m} = \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r,\theta) J_v(\lambda_n, r) \exp(-jm\theta) r dr d\theta, \tag{3}$$

donde $n = 0, 1, 2, ...$,$m = 0, \pm 1, \pm 2, ...$ es el orden del momento, $a_n = [J_{v+1}(\lambda_n)]^2/2$ es la constante de normalización, $J_v(\lambda_n, r)$ es el polinomio de Bessel en r de grado n, y λ_n es el n-ésimo cero de la función de Bessel $J_v(r)$. La Fig. 1 muestra la gráfica de los 5 primeros ordenes de los polinomios de Bessel $J_1(\lambda_n, r)$. La función $J_v(\lambda_n, r)$ es ortogonal dentro de un círculo unitario y cumplen la siguiente condición de ortogonalidad:

$$\int_0^1 J_v(\lambda_n, r) J_v(\lambda_k, r) r dr = a_n \delta_{nk}, \tag{4}$$

aquí δ_{nk} es el símbolo de Kronecker definido de la siguiente forma:

$$\delta_{n,k} = \begin{cases} 0, k \neq n \\ 1, k = n. \end{cases} \tag{5}$$

3. Momentos Bessel-Fourier de orden fraccional

En este trabajo realizamos la generalización de los momentos de Bessel-Fourier mediante la modificación del kernel. La funsión base propuesta está dada de la siguiente manera:

$$\widetilde{J}_{n,m}(\alpha, r, \theta) = \sqrt{\alpha} r^{\alpha - 1} J_{n,m}(r^\alpha, \theta), \tag{6}$$

donde α es el parámetro de generalización. La Fig. 2. muestra los polinomios generalizado con distintos valores de α. El nuevo kernel cumple con la condición de ortogonalidad. La demostración está dado por el Teorema 1.

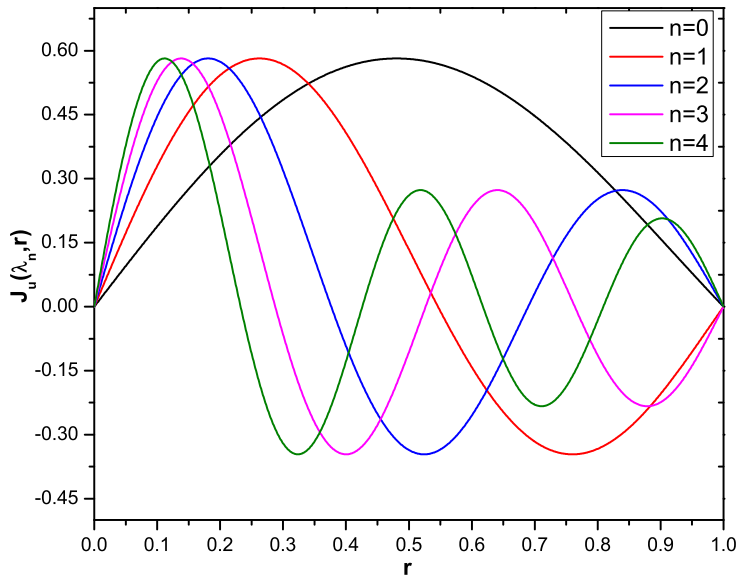


Fig. 1. Polinomio radial de Bessel $J_1(\lambda_n, r)$ de $n = 0, 1, 2, 3, 4$.

Teorema 1 *El kernel de la generalización de los momentos Bessel-Fourier es ortogonal en le intervalo* $[0, 1]$*:*

$$\int_0^{2\pi} \int_0^1 \widetilde{J}_{n,m}(\alpha, r, \theta)\, \widetilde{J}_{k,l}(\alpha, r, \theta)\, r dr d\theta = \delta_{nk}\delta_{ml}. \tag{7}$$

Demostración. Sustituyendo la Ec. 6 a la Ec. 7, tenemos:

$$\begin{aligned}
\int_0^{2\pi}\int_0^1 \widetilde{J}_{n,m}(\alpha,r,\theta)\widetilde{J}_{k,l}(\alpha,r,\theta) r dr d\theta &= \int_0^{2\pi}\int_0^1 \sqrt{\alpha} r^{\alpha-1} J_{n,m}(r^\alpha,\theta)\sqrt{\alpha} r^{\alpha-1} J_{k,l}(r^\alpha,\theta) r dr d\theta, \\
&= \int_0^{2\pi}\int_0^1 \alpha r^{2\alpha-1} J_{n,m}(r^\alpha,\theta) J_{k,l}(r^\alpha,\theta) dr d\theta, \\
&= \int_0^{2\pi}\int_0^1 \alpha r^{\alpha-1} J_{n,m}(r^\alpha,\theta) J_{k,l}(r^\alpha,\theta) r^\alpha dr d\theta, \\
&= \int_0^{2\pi}\int_0^1 J_{n,m}(r^\alpha,\theta) J_{k,l}(r^\alpha,\theta) r^\alpha dr^\alpha d\theta, \\
&= \int_0^{2\pi}\int_0^1 J_{n,m}(r,\theta) J_{k,l}(r,\theta) r dr d\theta, \\
&= \delta_{nk}\delta_{ml}.
\end{aligned} \tag{8}$$

Al igual que el kernel de los momentos de Bessel-Fouriel, el kernel de los momentos de Bessel-Fourier de orden fraccional consta de dos funciones: en su parte radial $\widetilde{J}_{n,m}(\alpha, r, \theta)$ y el factor exponencial de Fourier $\exp(-jm\theta)$.

La expresión general para los Momentos de Bessel-Fourier de orden fraccional de orden n y repetición m para una función imagen $f(r, \theta)$ en coordenadas

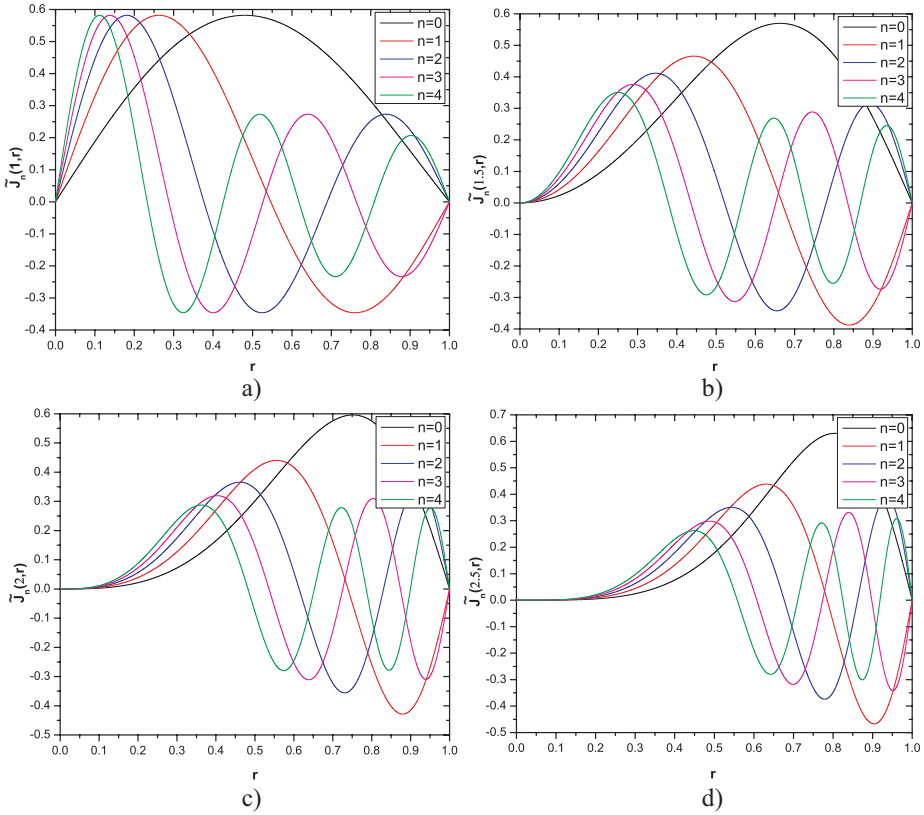


Fig. 2. Polinomio radial de Bessel de orden fraccional de $n = 0, 1, 2, 3, 4$ con diferentes valores de α: a) 1, b) 1.5, c) 2 y d) 2.5.

polares está dada por:

$$\phi_{n,m} = \int_0^{2\pi} \int_0^1 f\,(r,\theta)\,\widetilde{J}_{n,m}\,(\alpha, r, \theta)\exp(-jm\theta) r dr d\theta. \tag{9}$$

4. Momentos de multicanal de Bessel-Fourier de orden fraccional

Recientemente, Singh y Singh [17] definieron los momentos de Zernike multicanal como un conjunto de momentos para cada canal de una imagen a color. Para una imagen $f(r, \theta)$ en RGB, se definen tres canales de color primario, rojo (R), verde (G) y azul (B) donde la función de imagen podría representarse como $f_p(r, \theta) = (f_R(r, \theta), f_G(r, \theta), f_B(r, \theta))$.

Los momentos multicanal pueden representarse como:

$$M_{n,m}^{p} = \int_{0}^{2\pi} \int_{0}^{1} f_p\left(r,\theta\right) V_{n,m}\left(r,\theta\right) r dr d\theta, \quad p \in R,G,B, \tag{10}$$

donde $V_{n,m}(r,\theta)$ es la función kernel y $f_p(\ r,\theta)$ es la función imagen a color. Los momentos multicanal son por lo tanto un conjunto de momentos de cada canal de una imagen a color. Teniendo en cuenta lo anterior los Momentos de Multicanal de Bessel-Fourier de Orden Fraccional (MMBFOF) están definidos de la siguiente manera:

$$\phi_{n,m}^{p} = \int_{0}^{2\pi} \int_{0}^{1} f_p\left(r,\theta\right) \widetilde{J}_{n,m}\left(\alpha,r,\theta\right) \exp(-jm\theta) r dr d\theta, \quad p \in R,G,B, \tag{11}$$

La distribución discreta reconstruida de una imagen RGB está dada por:

$$\widetilde{f}_p\left(r,\theta\right) = \sum_{n=0}^{L} \sum_{m=-n}^{n} \phi_{n,m}^{p} \widetilde{J}_{n,m}\left(\alpha,r\right) \exp\left(-jm\theta\right), \tag{12}$$

donde $\widetilde{f}_p\left(r,\theta\right)$ es la imagen reconstruida de $f_p\left(r,\theta\right)$ y L es el orden máximo que se utiliza para el cálculo de los momentos ó la reconstrucción de una imagen.

5. Experimentos y análisis

En este apartado, se realizan varios experimentos para analizar el comportamiento de alfa en términos del error de reconstrucción y clasificación de imágenes a color. Para evaluar el desempeño los descriptores propuestos se utiliza la base de datos Columbia Object Image Library (COIL-100) [8], la cual es una base de datos pública utilizada por muchos investigadores para evaluar el reconocimiento de objetos en la actualidad. La base de datos COIL-100 contiene 7200 imágenes a color de 100 objetos diferentes. Los objetos fueron colocados en una plataforma giratoria motorizada con fondo negro, esta plataforma se giró a 360 grados con pasos de 5 grados para variar el ángulo de visión del objeto con respecto a una cámara de color fija. Lo que corresponde a 72 orientaciones diferentes de cada objeto. Todas las imágenes tienen un tamaño de 128x128 píxeles con formato PNG. La Fig. 3 muestra las imágenes de los objetos utilizados en la base de datos COIL-100. También, analizamos el comportamiento de la capacidad descriptiva mediante la reconstrucción de imágenes del método propuesto con distintos valores de α y lo comparamos con la reconstrucción con kernels de la misma familia.

5.1. Reconstrucción de imagen en color

La reconstrucción de una imagen puede ayudar a determinar que tan bien una imagen puede ser caracterizada por sus momentos.

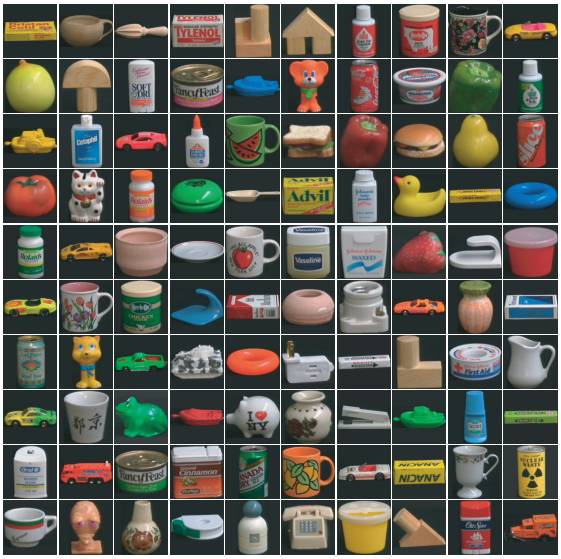


Fig. 3. Algunas imágenes de la base de datos Columbia Object Library (COIL 100).

De acuerdo con la teoría de momentos ortogonales una imagen digital $f(x, y)$ puede ser reconstruida por un número infinito de momentos ortogonales. Para medir el desempeño de los MMBFOF se utiliza el error de reconstrucción normalizado conocido como NIRE por sus siglas en inglés, el cual es calculado mediante el error cuadrático medio normalizado de la imagen original $f_p(r, \theta)$ y la imagen reconstruida $\widetilde{f}_p(r, \theta)$:

$$NIRE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left[\widetilde{f}_p\left(r, \theta\right) - f_p\left(r, \theta\right) \right]^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f_p^2\left(r, \theta\right)}. \tag{13}$$

La Fig. 4 muestra el $NIRE$ con respecto al parámetro α de los momentos multicanal de Bessel-Fourier de orden fraccional para distintos valores de L. En la teoría de momentos ortogonales, el error de construcción que mayor converja a cero tiene una mejor capacidad de caracterizar con un número menor de momentos. Por tal motivo es importante conocer el error de reconstrucción con respecto al número máximo de momentos que se utiliza para reconstruir la imagen. La Fig. 5 muestra el error de reconstrucción de los momentos que utilizan como kernel base las funciones Bessel.

Lo interesante de la reconstrucción es que podemos visualizar como al aumentar el número de momentos podemos ver como aumentan los detalles de la imagen.

Sin embargo, no podemos cuantificar visualmente que conjunto tiene un mejor desempeño. En la Fig. 6 muestra la reconstrucción de la imagen Lena con su respectivo NIRE para verificar el desempeño del método propuesto.

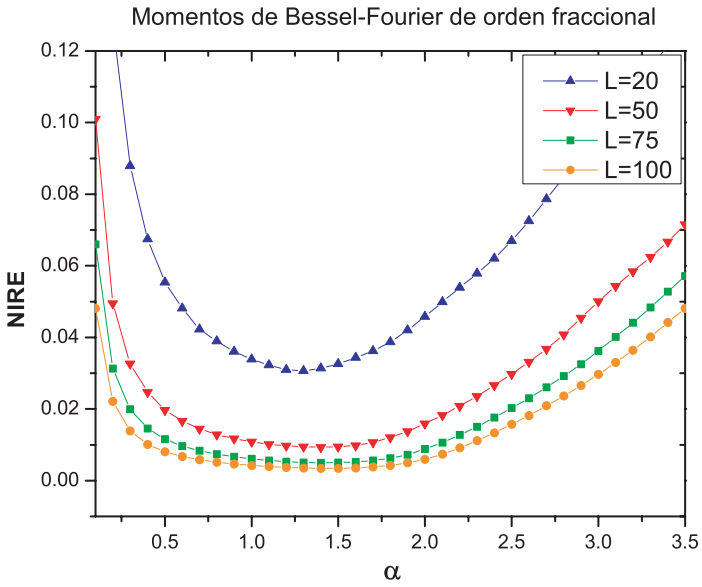


Fig. 4. *NIRE* de la imagen a color de "Lena" de tamaño 128 x 128 pixeles con un orden máximo de $L = 20, 50, 75$ y 100 para diferentes valores de α.

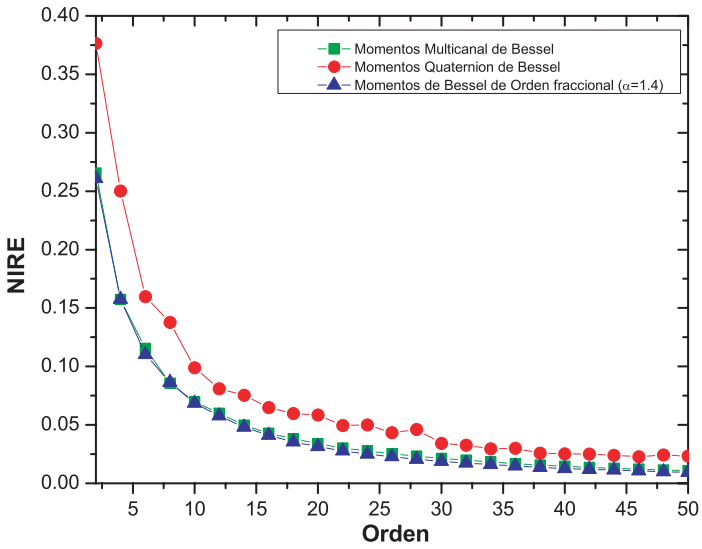


Fig. 5. NIRE de la imagen reconstruida de "Lena".

5.2. Clasificación de imágenes a color

En tareas de clasificación supervisadas, el objetivo es construir un predictor que asigne una etiqueta de clase a las nuevas observaciones en donde las clases

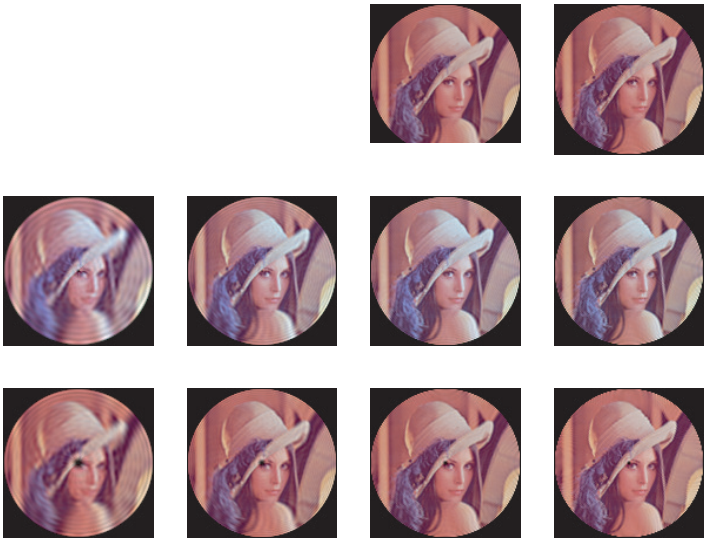


Fig. 6. Reconstrucción de imágenes utilizando diferentes tipos de momentos con diferentes órdenes.

se describen mediante un vector de características. La combinación de múltiples clasificadores conocidos como kNN ensamble proporciona un mejor rendimiento en los algoritmos de aprendizaje, especialmente en presencia de características no informativas en las imágenes como es el caso de los momentos ortogonales. Por otra parte, la generalización de los momentos de Bessel-Fourier con el parámetro α proporciona un espectro para la extracción de características de una imagen a color. En la Tabla 1 muestra los porcentajes de clasificación de la base de datos COIL-100 con KNN ensamble para diferentes valores de α y con cantidades de descriptores distintos. Además, la Tabla 2 muestra una comparación de los porcentajes de clasificación con KNN ensamble de las diferentes familias de momentos de Bessel-Fourier que existen en la literatura y con el método propuesto. Por otra parte, en la Tabla 3 se muestra la comparación de los porcentajes del error de clasificación del método propuesto con los métodos de clasificación que hay en la literatura.

5.3. Análisis de resultados

La generalización de los momentos de Bessel-Furier nos brinda la capacidad de poderse adaptar a una aplicación determinada conociendo el valor de α que mejor se adapte a ella. Sin embargo, la relación entre la capacidad de reconstrucción y la clasificación no es la misma. Se observa en la Fig.4, que la reconstrucción tienen un mejor desempeño en el intervalo de 1 a 1,5 esto se

Tabla 1. Clasificación de la base de datos COIL-100 con MMBFOF.

	α								
Orden	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9	2.1
L=2	86.1	86.5	84.7	87.2	88.6	89.5	90.2	90.1	90.3
L=4	98.9	99.3	99.2	99.1	99.4	99.1	99.1	99.3	99.1
L=6	99.6	99.6	99.5	99.5	99.6	99.6	99.5	99.5	99.5
L=8	99.7	99.6	99.5	99.7	99.6	99.6	99.5	99.5	99.5
L=10	99.8	99.6	99.6	99.5	99.5	99.7	99.6	99.6	99.4
L=12	99.7	99.7	99.5	99.5	99.6	99.6	99.6	99.6	99.6
L=14	99.6	99.6	99.6	99.5	99.6	99.6	99.5	99.6	99.5
L=16	99.5	99.6	99.6	99.6	99.5	99.5	99.6	99.6	99.7
L=18	99.6	99.6	99.7	99.5	99.5	99.6	99.6	99.6	99.6
L=20	99.6	99.6	99.6	99.5	99.5	99.6	99.6	99.6	99.6

Tabla 2. Clasificación de la base de datos COIL-100 con diferentes familias de momentos Bessel-Fourier.

Familia de momentos	**% de clasificación**
Momentos de cuaternión Bessel-Fourier [15]	99.3 %
Momentos multicanal de Bessel-Fourier [9]	99.6 %
Método propuesto	99.8 %

Tabla 3. Clasificación de la base de datos COIL-100 con diferentes métodos de clasificar.

Métodos	**Error de clasificación**
DCWN [5]	0.8 %
LinearBOF [10]	3.19 %
KernelSPM [10]	1.71 %
IRF-NN [19]	0.6 %
DWN [14]	0.8 %
Método propuesto	0.2 %

debe a que la capacidad descriptiva de una imagen depende de la distribución de los ceros de la familia de polinomios ortogonales, para el caso de los momentos de Bessel-Fourier de orden fraccional, el parámetro α influye en la posición de los ceros de los polinomios de Bessel de orden fraccional como se observa en la Fig.2. También, en la Fig.4 se puede apreciar que existe un valor mínimo del error de reconstrucción (NIRE) para un determinado α independientemente del número de momentos que se utilizan para reconstruir la imagen. Por otra parte, la comparación de los valores NIRE obtenidos con diferentes órdenes de los momentos Quaternion de Bessel-Fourier, los momentos Multicanal de Bessel-Fourier con los momentos de Bessel Fourier de orden fraccional se muestran en la Fig. 5. Asimismo, en la Fig. 6 se muestra las imágenes reconstruidas y el valor NIRE para cada conjunto de momentos para un orden de 20, 50, 75 y 100. En ambas figuras se puede ver claramente que: el valor de NIRE disminuye al aumentar el

orden de momento, y los valores de error para los momentos de Bessel-Fourier de orden fraccional son siempre los más pequeños; la inspección de las imágenes reconstruidas muestra que los momentos de Bessel-Fourier de orden fraccional con $L = 100$ conducen a una mejor restitución visual del objeto.

El análisis cualitativo presentado es totalmente coherente con el índice cuantitativo NIRE donde el rendimiento de los momentos Multicanal de Bessel-Fourier es el más cercano a los momentos de Bessel-Fourier de orden fraccional. En el caso de la clasificación de la base de datos COIL-100 con el método propuesto, la Tabla 1 presenta el comportamiento de los porcentajes de clasificación dependiendo del número de descriptores y el parámetro α en donde se puede observar existe un comportamiento que disminuye los números de descriptores y aumenta el porcentaje de clasificación para dos intervalos de α; el primero va de 0,5 a 0,9 en donde el mejor valor es de 99,8 % con $L = 10$ y el segundo, que va de 1,1 a 2,1 en donde el mejor valor es de 99,7 % con $L = 8$. También, se muestra en la Tabla 2 como la propuesta presentada tiene un mejor desempeño que otras familias de momentos que utilizan las funciones de Bessel como kernel. Además, en la Tabla 3 muestra que la generalización de los momentos de Melllin a través de los momentos de orden fraccional obtiene el error mínimo de clasificación de todos los métodos de clasificación comparados.

6. Conclusiones

Se presento la generalización de los momentos multicanal de Bessel-Fourier de orden fraccional para el análisis de imágenes en color. El rendimiento de los momentos propuestos superó las familias de momentos Bessel-Fourier existentes en términos del error de reconstrucción y la clasificación de imágenes en color. La generalización propuesta tiene mejores capacidades para describir y extraer las características finas de las imágenes en color que las transformaciones con kernels de orden entero. Se tiene que realizar más experimentos tanto de reconstrucción como de clasificación para poder caracterizar el parámetro α de los Momentos Multicanal de Bessel Fourier de Orden Fraccional propuestos con la finalidad de conocer como la distribución de ceros del kernel influye en mejora de aplicaciones de clasificación.

Referencias

1. Benouini, R., Batioua, I., Zenkouar, K., Zahi, A., Najah, S., Qjidaa, H.: Fractional-order orthogonal chebyshev moments and moment invariants for image representation and pattern recognition. Pattern Recognition 86, 332–343 (2019)
2. Chen, B., Yu, M., Su, Q., Shim, H.J., Shi, Y.Q.: Fractional quaternion zernike moments for robust color image copy-move forgery detection. IEEE Access 6, 56637–56646 (2018)
3. Chong, C.W., Raveendran, P., Mukundan, R.: The scale invariants of pseudo-zernike moments. Pattern Analysis & Applications 6(3), 176–184 (2003)
4. Flusser, J., Suk, T., Zitová, B.: 2D and 3D Image Analysis by Moments. John Wiley & Sons (2016)

5. Hassairi, S., Ejbali, R., Zaied, M.: Supervised image classification using deep convolutional wavelets network. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 265–271. IEEE (2015)
6. Hosny, K.M., Darwish, M.M., Aboelenen, T.: Novel fractional-order polar harmonic transforms for gray-scale and color image analysis. Journal of the Franklin Institute (2020)
7. Hu, H.t., Zhang, Y.d., Shao, C., Ju, Q.: Orthogonal moments based on exponent functions: Exponent-fourier moments. Pattern Recognition 47(8), 2596–2606 (2014)
8. Nene, S.A., Nayar, S.K., Murase, H., et al.: Columbia object image library (coil-20) (1996)
9. Pang, B., Liao, S.: Color image reconstructions from bessel-fourier moments. In: 2016 International Conference on Multimedia Systems and Signal Processing (ICMSSP). pp. 16–21. IEEE (2016)
10. Peng, X., Yan, R., Zhao, B., Tang, H., Yi, Z.: Fast low rank representation based spatial pyramid matching for image classification. Knowledge-Based Systems 90, 14–22 (2015)
11. Ping, Z., Ren, H., Zou, J., Sheng, Y., Bo, W.: Generic orthogonal moments: Jacobi–fourier moments for invariant image description. Pattern recognition 40(4), 1245–1254 (2007)
12. Ping, Z., Wu, R., Sheng, Y.: Image description with chebyshev–fourier moments. JOSA A 19(9), 1748–1754 (2002)
13. Ren, H., Ping, Z., Bo, W., Wu, W., Sheng, Y.: Multidistortion-invariant image recognition with radial harmonic fourier moments. JOSA A 20(4), 631–637 (2003)
14. Said, S., Jemai, O., Hassairi, S., Ejbali, R., Zaied, M., Amar, C.B.: Deep wavelet network for image classification. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 000922–000927. IEEE (2016)
15. Shao, Z., Shu, H., Wu, J., Chen, B., Coatrieux, J.L.: Quaternion bessel–fourier moments and their invariant descriptors for object reconstruction and recognition. Pattern Recognition 47(2), 603–611 (2014)
16. Sheng, Y., Shen, L.: Orthogonal fourier–mellin moments for invariant pattern recognition. JOSA A 11(6), 1748–1757 (1994)
17. Singh, C., Singh, J.: Multi-channel versus quaternion orthogonal rotation invariant moments for color image representation. Digital Signal Processing 78, 376–392 (2018)
18. Singh, C., Walia, E., Upneja, R.: Accurate calculation of zernike moments. Information Sciences 233, 255–275 (2013)
19. Smagghe, P., Buessler, J.L., Urban, J.P.: Novelty detection in image recognition using irf neural networks properties. In: ESANN (2013)
20. Teague, M.R.: Image analysis via the general theory of moments∗. J. Opt. Soc. Am. 70(8), 920–930 (Aug 1980)
21. Xiao, B., Li, L., Li, Y., Li, W., Wang, G.: Image analysis by fractional-order orthogonal moments. Information Sciences 382, 135–149 (2017)
22. Xiao, B., Ma, J.F., Wang, X.: Image analysis by bessel–fourier moments. Pattern Recognition 43(8), 2620–2629 (2010)
23. Xiao, B., yin Wang, G., sheng Li, W.: Radial shifted legendre moments for image analysis and invariant image recognition. Image and Vision Computing 32(12), 994 – 1006 (2014)
24. Yang, J., Jin, D., Lu, Z.: Fractional order zernike moment [j]. Journal of Computer-Aided Design & Computer Graphics 29(3), 480–484 (2017)

25. Zhang, H., Li, Z., Liu, Y.: Fractional orthogonal fourier-mellin moments for pattern recognition. In: Chinese Conference on Pattern Recognition. pp. 766–778. Springer (2016)

ISSN 1870-4069

Reconocimiento de imágenes RGB con momentos multicanal de Zernike

Alicia Noriega-Escamilla, César Camacho-Bello,
José Saúl Rivera-López, Horlando Vargas-Vargas

Universidad Politécnica de Tulancingo,
México

aliciane7@gmail.com, {cesar.camacho, 1915001,
horlando.vargas.1631027}@upt.edu.mx

Resumen. En este trabajo se presenta un análisis de los descriptores tipo momentos de Zernike y sus invariantes para la clasificación de imágenes en color. Se compara los momentos multicanal de Zernike con los métodos existentes. Los resultados de los experimentos realizados presentan una exactitud en la clasificación superior al 99 %.

Palabras clave: Polinomios de Zernike, momentos de Zernike, momentos multicanal de Zernike.

Recognition of RGB Images with Zernike Multichannel Moments

Resumen. This paper presents an analysis of the typical descriptors of Zernike and its invariants for the classification of color images. Zernike's multichannel moments are compared to existing methods. The results of the experiments carried out show an accuracy in the classification greater than 99 %.

Palabras clave: Zernike Polynomials, Zernike Moments, Zernike Multichannel Moments.

1. Introducción

Los descriptores tipo momentos de una imagen y sus invariantes a transformaciones geométricas han sido utilizados ampliamente en aplicaciones de Visión por Computadora [1] y Reconocimiento de Patrones [5], debido a que los objetos exhiben una complejidad natural a cambios de escala, traslación, ruido y diversos factores involucrados en la adquisición de imágenes. Entre los descriptores más representativos se encuentran los momentos de Zernike, los cuales son capaces de obtener suficiente información para separar objetos en clases y son robustos ante transformaciones de tipo geométrico y ruido [2].

La mayor parte del estudio para el desarrollo de descriptores invariantes se ha realizado en imágenes en escala de grises; sin embargo, hoy en día la mayor parte de la información para el reconocimiento de objetos proviene de imágenes en color.

Por lo general, se utilizan dos enfoques en el procesamiento del color de la imagen; el primero, es transformar la imagen en color en escala de grises promediando los tres canales RGB, con lo que existe pérdida de información de color y el segundo, descompone la imagen en sus tres canales para analizar la información de cada canal por separado.

Para el análisis de imágenes a color con momentos radiales se ha utilizado el álgebra de cuaternión para representar las imágenes mediante los momentos Cuaternión y sus invariantes a escala, rotación y traslación. Recientemente, el enfoque multicanal surge como una propuesta alternativa a los momentos Cuaternión para la representación y reconocimiento de imágenes en color.

Sing y Sing [9] presentan los momentos ortogonales multicanal basados en los momentos Zernike (MM). De igual modo, Hosny y Darwish [3] presentan un nuevo conjunto de momentos multicanal de Chebyshev-Fourier (MMCF) y sus invariantes de rotación, escala y traslación para la representación y el reconocimiento de imágenes en color.

Hosny y Darwish [3] realizan una serie de experimentos comparando el rendimiento de los momentos multicanal de Chebyshev-Fourier propuestos con los momentos de cuaternión existentes y los momentos multicanal de Zernike propuestos por Singh y Singh [9]. Ambas publicaciones confirman la superioridad de los momentos multicanal frente a los momentos cuaternión.

Por otro lado, para evaluar la capacidad descriptiva de los momentos ortogonales multicanal de Zernike se realizan experimentos de clasificación que permiten reconocer objetos de color similares en presencia de distorsiones geométricas y diferentes tipos de ruido. Los artículos citados utilizan distancias entre vectores de características como métrica para medir la similitud entre dos imágenes. Actualmente el uso de diferentes modelos de clasificadores tales como, arboles de decisión, análisis discriminante, máquinas de vectores de soporte y vecinos cercanos permiten realizar clasificación supervisada y generan modelos con base en relaciones, patrones o reglas, para poder clasificar nuevos elementos.

Estos modelos de clasificación presentan precisión en la predicción, eficiencia en costos computacionales, robustez para funcionar con ruido y la habilidad de trabajar con una gran cantidad de datos. Sin embargo, la calidad de los modelos depende en gran medida de la calidad de los datos usados en su entrenamiento. En este trabajo las imágenes son descritas mediante los momentos multicanal de Zernike, y los experimentos realizados muestran el comportamiento de estos descriptores para el reconocimiento de imágenes RGB con la finalidad de determinar el modelo de clasificación adecuado para los momentos multicanal de Zernike.

2. Momentos multicanal de Zernike

Los momentos multicanal quedan definidos como un conjunto de momentos para cada canal de una imagen de color $f_C(r,\theta)$, donde $f_R(r,\theta)$, $f_G(r,\theta)$, $f_B(r,\theta)$ corresponden a las funciones de intensidad del canal rojo, verde y azul respectivamente. Los momentos de cada canal se calculan como [9]:

$$M_{nl}^{C}(f_C(r,\theta)) = \frac{n+1}{\pi}\int_0^{2\pi}\int_0^1 Z_{nl}(r,\theta)f_C(r,\theta)rdrd\theta, \tag{1}$$

donde C se refiere a los canales R, G y B, y $Z_{nl}(r,\theta)$ son los polinomios de Zernike de enésimo grado n y repetición l, definidos de la siguiente manera [10]:

$$Z_{nl}(r,\theta) = R_{nl}(r)e^{il\theta}, \tag{2}$$

donde la parte radial esta definida como:

$$R_{nl}(r) = \sum_{s=0}^{(n-|l|)/2}(-1)^s\frac{(n-s)!}{s!(\frac{n+|l|}{2}-s)!(\frac{n-|l|}{2}-s)!}. \tag{3}$$

Una característica importante de la parte radial de los polinomios de Zernike es su simetría con respecto al orden de repetición l, es decir:

$$R_{n-l}(r) = R_{nl}(r). \tag{4}$$

Además, los polinomios de Zernike satisfacen la condición de ortogonalidad:

$$\int_0^{2\pi}\int_0^{1\pi} Z_{nl}(r,\theta)Z_{mk}(r,\theta)rdrd\theta = \frac{\pi}{n+1}\delta_{mn}\delta_{kl}. \tag{5}$$

La complejidad del cálculo de la componente radial de los polinomios de Zernike definida en la Ec. 3. Se utilizan relaciones de recurrencia, debido a que su implementación numérica es estable y rápida [1]. Existen diferentes propuestas de relaciones de recurrencia [6],[4]; sin embargo, la propuesta de Honarvar y coautores [8] para el cálculo rápido y estable es la más utilizada. La relación de recurrencia está dada por:

$$R_n^m(r) = r[R_{n-l}^{m-l}(r) + R_{n-l}^{m+l}(r)] - R_{n-2}^m(r), \tag{6}$$

con el polinomio radial de orden cero $R_0^0(r) = 1$, se pueden calcular todos los polinomios radiales $R_n^m(r)$ con enteros $\succeq 0$ tal que $n-m$ es par y no negativo, y $R_n^m(r) \equiv 0$ cuando $n \prec m$.

Una de las propiedades más importantes que los momentos ortogonales poseen es su invariancia a la rotación, traslación y escala. En el caso de la invariante de rotación, una imagen puede rotar un ángulo α con respecto a la imagen original:

$$M_{n,l}^c(f) = M_{n,l}^c(f)e^{il\alpha}, \tag{7}$$

por lo tanto, los invariantes de rotación se pueden construir tomando la magnitud de los momentos $|M^C_{n,l}|$. La invariancia a la escala, se obtiene realizando un mapeo de la imagen en un círculo de radio unitario de forma independiente a la escala de la imagen, este mapeo proporciona la invariancia al escalamiento. La invariante de traslación es obtenida realizando una normalización de la imagen donde el objeto se desplaza de tal manera que su centroide coincide con el origen del sistema de coordenadas, de acuerdo con Suk y Flusser [2] el centroide (x_c, y_c) de una imagen RGB está definido como:

$$x_c = m_{10}/m_{00}$$
$$y_c = m_{01}/m_{00}, \tag{8}$$

donde m_{01},m_{10} y m_{00} son los momentos geométricos de primer orden y orden cero respectivamente. Al ubicar el origen de coordenadas en el centroide (x_c, y_c) y mapear la imagen en el círculo unitario, se logra la invariancia a la traslación.

Por otro lado, el número de descriptores obtenidos dependen del orden máximo radial y angular en el cálculo de los momentos. El número total de momentos de Zernike está dado por:

$$(\frac{2 + Z_{max}}{2})^2 Z_{max} par, \tag{9}$$

$$(\frac{1 + Z_{max}}{2})^2 (\frac{3 + Z_{max}}{2}) Z_{max} impar, \tag{10}$$

donde Z_{max} se refiere al orden máximo.

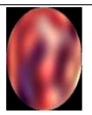
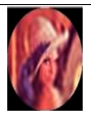
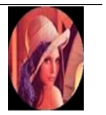
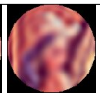
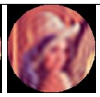
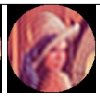
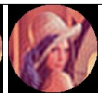
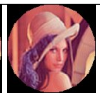
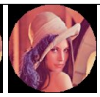
3. Experimentos y resultados

Los experimentos realizados se llevan a cabo en el programa Matlab versión R2016b. En el primer experimento se realiza la reconstrucción de la imagen "Lena" de tamaño 128×128, el proceso de reconstrucción de imágenes en color se utiliza para evaluar la precisión de descripción y la estabilidad numérica que los momentos poseen. Se realiza una comparación de los diferentes modelos de clasificación para determinar el clasificador que mejor se adapta a los descriptores tipo momentos de Zernike. Para el análisis del desempeño de los clasificadores se utiliza la base de datos COIL-100 (Columbia Object Image Library), conformada por 7200 imágenes de tamaño 128×128 que se clasifica en 100 clases de objetos diferentes.Cada clase consiste en una imagen del objeto tomada desde un punto de vista diferente que varía desde el ángulo 0° a 360° con un incremento de ángulo de 5°. Se calculan las distintas medidas de distancia para evaluar la similitud entre imágenes ($L_1 - norm$, $L_2 - norm$, X_2 y $Canberra$) utilizadas en las publicaciones citadas [9,3] para extraer las características de las imágenes.

3.1. Recostrucción de imágenes

La reconstrucción de imágenes ayuda a identificar el número de descriptores tipo momentos necesarios para representar una imagen en escala de grises o en color.

Tabla 1. Imagen de Lena recostruida con ordenes de 0 a 150.

Max. Order	10	20	30	40	50	100	140	150
MMZ [3]								
	0.0524	0.0313	0.0228	0.0207	1.26e+05			
MMCF [3]								
	0.0356	0.0181	0.0116	0.0079	0.0056	0.0020	0.0017	0.0016
MMZ								
	0.0410	0.0233	0.0156	0.0109	0.0075	0.0015	7.564e-04	7.1287e-04

Mientras un orden menor tiene la capacidad de representar características generales de la imagen como área o forma, los órdenes superiores poseen la capacidad de representar detalles de esta. La distribución discreta reconstruida de una imagen RGB está dada por:

$$\widetilde{f}_C(r,\theta) = \sum_{n=0}^{L} \sum_{m=-n}^{n} \phi_{nm} R_n^m(r) \exp(-jm\theta), \tag{11}$$

donde $\widetilde{f}_C(r,\theta)$ es la imagen reconstruida y L es el orden máximo que se utiliza para la reconstrucción de una imagen.

La precisión en la reconstrucción se evalúa por medio del NIRE [7], definido como:

$$NIRE = \frac{\sum_{n-1}^{N} \sum_{m-1}^{M} |f(i,j) - f'(i,j)|^2}{\sum_{n-1}^{N} \sum_{m-1}^{M} |f(i,j)|^2}. \tag{12}$$

Para medir el error de reconstrucción de una imagen a color se toma en cuenta el promedio de los tres canales:

$$MeanNIRE = \frac{NIRE_R + NIRE_G + NIRE_B}{3}. \tag{13}$$

Actualmente Darwish y Hosny presentan buenos resultados en la reconstrucción de imágenes con los momentos multicanal de Chebyshev-Fourier en comparación con los momentos multicanal de Zernike. Sin embargo, calculan los momentos con aproximaciones de primer orden y no con las aproximaciones más exactas que existen en la literatura. El cálculo de los momentos de Zernike se realiza mediante un esquema de pixeles polares propuesto por Xin y Liao [11]. Los resultados obtenidos en la reconstrucción de imágenes con ambos enfoques se muestran en la Tabla 1.

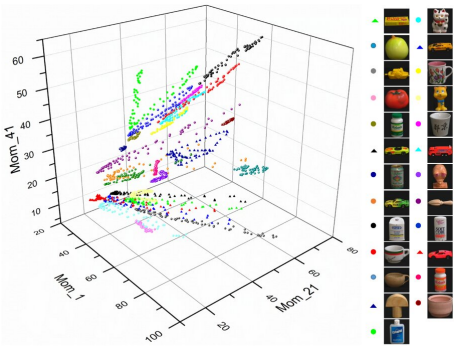


Fig. 1. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 en condiciones normales con MMZ.

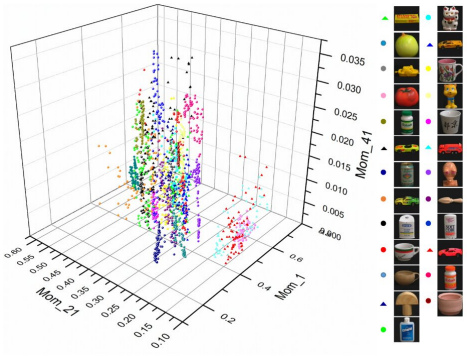


Fig. 2. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 en condiciones normales con MMCF.

3.2. Reconocimiento de imágenes utilizando clasificadores

Los modelos de clasificación son usados en minería de datos como una herramienta estratégica que permite ordenar, clasificar, filtrar y resumir todas las relaciones que un dato puede tener dentro de la información.

Para los modelos de clasificación se calculan los MMZ y los MMCF a la base de datos COIL-100 bajo condiciones normales, rotación aleatoria, escala y ruido, igualando a 60 la cantidad de descriptores obtenidos y se utiliza validación cruzada en el entrenamiento *KFold =10*. Para evaluar los clasificadores se considera la exactitud en la clasificación que determina la proporción de ejemplos correctamente clasificados, se muestra el diagrama de dispersión para visualizar el desempeño del algoritmo empleado. Las medidas de distancia utilizadas en [9] y [3], se realizan con la base de datos COIL-100 dividida en dos conjuntos; en el primero, se toma la imagen frontal de cada clase y las 71 imágenes restantes se colocan en la base de datos de prueba. Por lo tanto, hay 100 imágenes en

el primer conjunto y 7100 imágenes en la base de datos de prueba. La tasa de reconocimiento $R(\%)$ está dada por $P \times 100/Q$ donde P es el número de imágenes reconocidas correctamente y Q es el número de imágenes utilizadas en la prueba [9].

Tabla 2. Parámetros utilizados para los modelos de clasificación.

Clasificador	Parámetros
Complex Tree	Número de Splits = 100
Medium Tree	Número de Splits = 20
Simple Tree	Número de Splits = 4
Linear Discriminant	
Quadratic Discriminant	
Linear SVM	
Quadratic SVM	
Cubic SVM	
Fine Gaussian SVM	Escala del kernel = 1.9
Medium Gaussian SVM	Escala del kernel = 7.7
Coarse Gaussian SVM	Escala del kernel = 31
Fine KNN	Número de vecinos = 1
Medium KNN	Número de vecinos = 10
Coarse KNN	Número de vecinos = 100
Cosine KNN	Número de vecinos = 10
Cubic KNN	Número de vecinos = 10
Weighted KNN	Número de vecinos = 10
Ensemble/Boosted Trees	
Ensemble/Bagged Trees	
Ensemble/Sub Discriminant	
Ensemble/Subspace KNN	
Ensemble/RUSBoosted Trees	
$L_1 - norm$ KNN	Número de vecinos = 1
$L_2 - norm$ KNN	Número de vecinos = 1
X^2 KNN	Número de vecinos = 1
$Canberra$ KNN	Número de vecinos = 1

En la Tabla 2 se muestran los parámetros utilizados en los modelos de clasificación. En la Tabla 3 se muestran los resultados obtenidos para la clasificación de las imágenes en condiciones normales con MMZ y MMCF. En el siguiente experimento las imágenes de la base de datos COIL-100 son rotadas de forma aleatoria en ángulos entre 0° a 180° con un incremento fijo de 30°. Los resultados para la clasificación con rotación aleatoria se muestran en la Tabla 3. La base de datos consta de 100 clases distintas, sin embargo, para visualizar el comportamiento de los descriptores se incluyeron solamente 25 clases en los diagramas de dispersión. Las Figuras 1 y 2 muestran respectivamente el diagrama de dispersión para los MMZ y MMCF en condiciones normales.

En las Figuras 3 y 4 se observa el diagrama de dispersión para los MMZ y MMCF con rotación aleatoria. Para ejemplificar la invariancia a la a escala, la

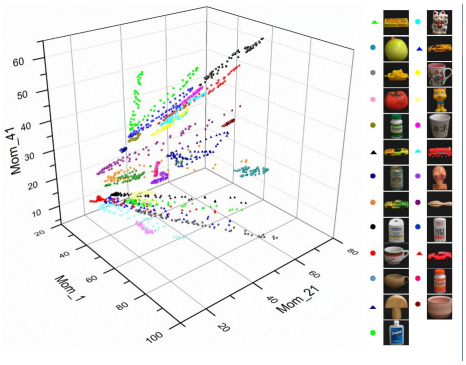


Fig. 3. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 en rotación aleatoria con MMZ.

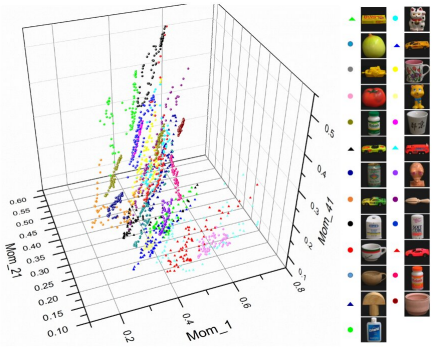


Fig. 4. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 en rotación aleatoria con MMCF.

base de datos COIL-100 es escalada con un factor λ= 1.5. Se evalúa la robustez al ruido añadiendo ruido Gaussiano con $\mu = 0$ y $\sigma^2 = 0{,}20$ a la base de datos. Los resultados de la clasificación con la base de datos a escala λ= 1.5 y con ruido Gaussiano añadido se muestran en la Tabla 4. Los diagramas de dispersión para los MMZ y MMCF con la base de datos a escala λ= 1.5 se muestran en las Figuras 5 y 6 respectivamente y en las Figuras 7 y 8 se muestran los diagramas de dispersión para MMZ y MMCF respectivamente para la base de datos con ruido Gaussiano.

4. Conclusiones

Los experimentos realizados con los momentos multicanal de Zernike presentan un desempeño superior para la reconstrucción de imágenes.

Tabla 3. Exactitud para los diferentes clasificadores con momentos multicanal de Zernike y Chebyshev-Fourier, Modelo de color RGB en condiciones normales y con rotación aleatoria

Clasificador	MMZ	MMCF	MMZ	MMCF
	Normal		Rotación aleatoria	
Complex Tree	46.7 %	48.4 %	46.8 %	48.4 %
Medium Tree	15.4 %	16.3 %	15.4 %	16.3 %
Simple Tree	4.6 %	4.9 %	4.6 %	5.0 %
Linear Discriminant	94.9 %	86.8 %	89.0 %	86.5 %
Quadratic Discriminant	98.9 %	92.4 %	95.2 %	92.2 %
Linear SVM	99.0 %	98.2 %	99.0 %	98.5 %
Quadratic SVM	99.7 %	99.2 %	99.7 %	99.3 %
Cubic SVM	99.7 %	99.3 %	99.7 %	99.3 %
Fine Gaussian SVM	78.9 %	58.1 %	78.9 %	57.9 %
Medium Gaussian SVM	99.7 %	99.1 %	99.7 %	99.1 %
Coarse Gaussian SVM	96.3 %	93.9 %	96.3 %	93.9 %
Fine KNN	99.9 %	93.9 %	99.9 %	99.4 %
Medium KNN	96.2 %	94.8 %	96.2 %	94.6 %
Coarse KNN	71.8 %	72.4 %	72.0 %	72.3 %
Cosine KNN	96.2 %	94.1 %	96.2 %	94.1 %
Cubic KNN	95.5 %	93.8 %	95.5 %	93.6 %
Weighted KNN	98.9 %	97.8 %	98.9 %	97.8 %
Ensemble/Boosted Trees	39.2 %	39.2 %	39.2 %	38 %
Ensemble/Bagged Trees	99.2 %	98.4 %	99.2 %	98.3 %
Ensemble/Sub Discriminant	93.7 %	91.4 %	93.8 %	91.3 %
Ensemble/Subspace KNN	99.9 %	99.2 %	99.9 %	99.3 %
Ensemble/RUSBoosted Trees	17.6 %	16.4 %	16.5 %	16.8 %
$L_1 - norm$	68.69 %	59.56 %	60.70 %	49.63 %
$L_2 - norm$	68.39 %	54.14 %	68.39 %	25.74 %
X^2	63.14 %	49.14 %	63.16 %	23.22 %
$Canberra$	61.83 %	49.05 %	61.83 %	34.66 %

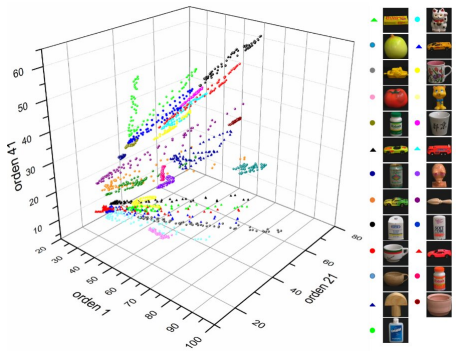


Fig. 5. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 escala con MMZ.

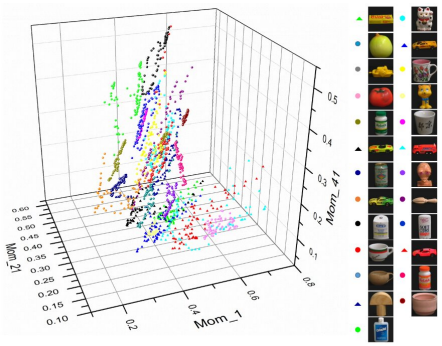


Fig. 6. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 a escala con MMCF.

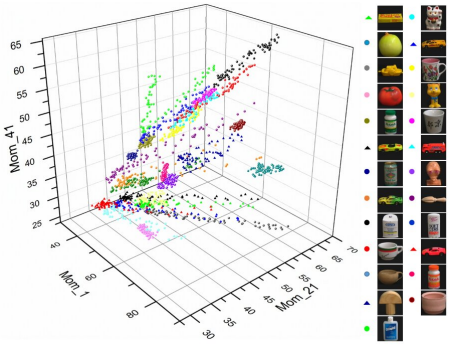


Fig. 7. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 con ruido Gaussiano con MMZ.

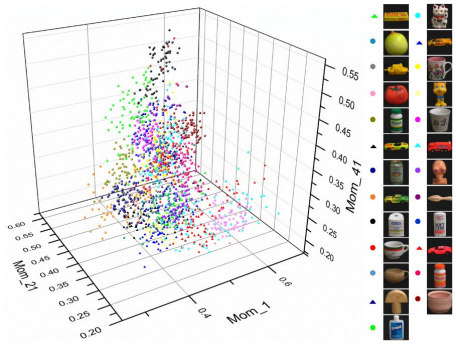


Fig. 8. Diagrama de dispersión para las primeras 25 clases de la base de datos COIL-100 con ruido Gaussiano con MMCF.

Tabla 4. Exactitud para los diferentes clasificadores con Momentos Multicanal de Zernike y Chebyshev-Fourier, Modelo de color RGb con ruido Gaussiano $\mu = 0$ y $\sigma^2 = 0{,}20$

Clasificador	MMZ Escala	MMCF $\lambda = 1{,}5$	MMZ Ruido	MMCF Gaussiano
Complex Tree	46.8 %	47.8 %	49.0 %	34.9 %
Medium Tree	16.5 %	16.4 %	14.7 %	14.1 %
Simple Tree	4.6 %	4.9 %	4.6 %	4.2 %
Linear Discriminant	88.9 %	86.6 %	87.9 %	75.7 %
Quadratic Discriminant	95.1 %	92.3 %	94.3 %	78.2 %
Linear SVM	99.1 %	98.3 %	98.2 %	81.3 %
Quadratic SVM	99.8 %	99.1 %	99.1 %	83.2 %
Cubic SVM	99.7 %	99.2 %	99.1 %	81.9 %
Fine Gaussian SVM	78.5 %	58.0 %	5.9 %	3.5 %
Medium Gaussian SVM	99.8 %	99.0 %	98.9 %	79.0 %
Coarse Gaussian SVM	96.4 %	93.9 %	94.3 %	71.9 %
Fine KNN	99.9 %	99.4 %	99.5 %	71.3 %
Medium KNN	96.0 %	94.5 %	96.0 %	72.7 %
Coarse KNN	72.2 %	72.1 %	72.3 %	63.8 %
Cosine KNN	96.2 %	94.1 %	95.7 %	71.1 %
Cubic KNN	95.7 %	93.5 %	95.3 %	68.2 %
Weighted KNN	99.0 %	97.8 %	97.9 %	74.9 %
Ensemble/Boosted Trees	38.2 %	37.9 %	35.6 %	22.1 %
Ensemble/Bagged Trees	99.1 %	98.4 %	98.4 %	78.7 %
Ensemble/Sub Discriminant	93.7 %	91.2 %	92.6 %	80.5 %
Ensemble/Subspace KNN	99.9 %	99.3 %	99.8 %	83.8 %
Ensemble/RUSBoosted Trees	98.7 %	15.9 %	99.6 %	16.6 %
$L_1 - norm$	68.64 %	49.61 %	68.64 %	49.61 %
$L_2 - norm$	68.35 %	25.73 %	68.35 %	25.73 %
X^2	63.17 %	23.28 %	63.17 %	23.28 %
$Canberra$	61.71 %	34.67 %	61.71 %	34.67 %

Los MMZ calculados poseen una mayor estabilidad numérica en órdenes superiores. Se determina como mejor opción para los descriptores de tipo momento el modelo de clasificación Ensemble/Subspace KNN el cual logra una exactitud de clasificación del 99.9 % en condiciones normales, este porcentaje de exactitud se mantiene con la base de datos rotada aleatoriamente, con escala $\lambda = 1{,}50$ e incluso con ruido Gaussiano $\mu = 0$ y $\sigma^2 = 2{,}0$. Los diagramas de dispersión se realizaron con los descriptores 1, 21 y 41 lo que equivale al primer descriptor de cada canal de color, lo cual demuestra que es posible realizar la clasificación con momentos multicanal de bajo orden. Los modelos de clasificación logran una mayor exactitud en comparación con las medidas de distancia utilizadas en [9] y [3]. Las pruebas realizadas con momentos multicanal de Chebyshev-Fourier, realizadas igualmente con 60 descriptores obtienen d eigual forma el mayor porcentaje de exactitud con el modelo Ensemble/Subspace KNN; 99,3 % de exactitud en condiciones normales, rotada aleatoriamente y con escala $\lambda = 1{,}50$.

La prueba con ruido Gaussiano $\mu = 0$ y $\sigma^2 = 2{,}0$ obtiene una exactitud de 83,8 % en el mismo modelo de clasificación confirmando el buen rendimiento del modelo.

Referencias

1. Camacho-Bello, C., Toxqui-Quitl, C., Padilla-Vivanco, A., Báez-Rojas, J.: High-precision and fast computation of jacobi–fourier moments for image description. JOSA A (2014)
2. Flusser, J., Zitova, B., Suk, T.: Moments and moment invariants in pattern recognition. John Wiley & Sons (2009)
3. Hosny, K.M., Darwish, M.M.: New set of multi-channel orthogonal moments for color image representation and recognition. Pattern Recognition (2019)
4. Janssen, A.J., Dirksen, P.: Computing zernike polynomials of arbitrary degree using the discrete fourier transform. Journal of the European Optical Society-Rapid publications (2007)
5. Khotanzad, A., Hong, Y.H.: Invariant image recognition by zernike moments. IEEE Transactions on pattern analysis and machine intelligence (1990)
6. Prata, A., Rusch, W.: Algorithm for computation of zernike polynomials expansion coefficients. Applied Optics (1989)
7. Rivera-López, J., Camacho-Bello, C., et al.: Color image reconstruction by discrete orthogonal moment. Journal of Data Analysis and Information Processing (2017)
8. Shakibaei, B.H., Paramesran, R.: Recursive formula to compute zernike radial polynomials. Optics letters (2013)
9. Singh, C., Singh, J.: Multi-channel versus quaternion orthogonal rotation invariant moments for color image representation. Digital Signal Processing (2018)
10. Teague, M.R.: Image analysis via the general theory of moments. JOSA (1980)
11. Xin, Y., Pawlak, M., Liao, S.: Accurate computation of zernike moments in polar coordinates. IEEE Transactions on Image Processing (2007)

ISSN 1870-4069

Detección automática de zonas de alto riesgo de eventos delictivos a través de noticias periodísticas

Yadira Laureano de Jésus, Guillermo De Ita Luna,
Mireya Tovar Vidal

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

star95ariday@hotmail.com,{deita,mtovar}@cs.buap.mx

Resumen. Actualmente en las noticias, los eventos criminales están más presentes y cada año aumentan significativamente. Debido a esto, se genera inseguridad y miedo en las personas que día a día se informan mediante este medio de información. Por ello es importante estar al tanto de estos eventos, principalmente para las autoridades para que tomen medidas de seguridad y apliquen las políticas de prevención de estos eventos criminales. En este trabajo de investigación se desarrolla un algoritmo basado en aprendizaje profundo, para la clasificación automática sobre noticias de eventos criminales (homicidio, secuestro, asalto, suicidio y violación) y para la detección automática de zonas de alto riesgo. Para la clasificación de noticias, el entrenamiento se realiza con un corpus de encabezados de noticias de Twitter y para las pruebas se realizan con un corpus formado por noticias obtenidas de un periódico local (Milenio). Finalmente, en el proceso de clasificación se obtuvo una exactitud del 98 % y también se distinguió el estado de México por ser el estado donde suceden la mayoría de estos eventos criminales.

Palabras clave: Aprendizaje profundo, clasificación automática, eventos criminales.

Automatic Detection of High-risk Areas for Criminal Events through Journalistic News

Abstract. Currently in the news, criminal events are more present and each year they increase significantly. Due to this, insecurity and fear are generated in people who are informed daily through this information medium. Therefore, it is important to be aware of these events, mainly for the authorities so that they take security measures and apply the prevention policies for these criminal events. In this research work, an algorithm based on deep learning is developed, for the automatic classification of criminal events news (homicide, kidnapping, assault, suicide

and rape) and for the automatic detection of high risk areas. For the news classification, the training is done with a corpus of Twitter news headlines, and for the tests they are carried out with a corpus formed by news obtained from a local newspaper (Milenium). Finally, in the classification process an accuracy of 98 % was obtained and the state of Mexico was also distinguished for being the state where most of these criminal events occur.

Keywords: Deep learning, automatic classification, criminal events.

1. Introducción

Las noticias sobre crimen o violencia [3], giran en entorno a investigaciones de un delito convirtiéndose en el foco de atención de los lectores dejando a un lado otros temas. Las noticas con el tema del delito genera miedo y preocupación al lector puesto que ponen en peligro su seguridad y la seguridad de sus familiares. Este tipo de noticias que presentan un grado de criminalidad perjudica a la sociedad, debido a que su temor es más grande que su confianza y por ello en vez de ayudar a combatir el crimen, favorece a las organizaciones criminales. Por lo tanto, es de suma importancia analizar las noticias, dado que al hacer un análisis de las noticias se pueden obtener los eventos delictivos con más incidencia de ocurrencia por país, estado o entidad, al igual que el mes, año, etc. Además de que se pueden obtener las zonas con mayor índice de peligro y sobre todo el delito con más ocurrencia. Al obtener un análisis reforzado se puede alertar a las comunidades más vulnerables y así tomar estrictas medidas de seguridad por parte de las autoridades.

En la revisión de la literatura científica de trabajos relacionados a los objetivos de este tema de investigación, hasta este momento no se han encontrado investigaciones que permitan realizar la clasificación de eventos delictivos en noticias periodísticas a través de redes neuronales, con la clasificación de los eventos sobre: homicidio, secuestro, asalto, suicidio y violación, en noticias periodísticas mexicanas. En la revisión realizada se pudo observar que hay una variedad de estudios realizados en los temas de aprendizaje profundo en diferentes áreas, todas ellas abordadas desde el idioma inglés. Se ha abordado el tema de clasificación de crímenes a través de patrones [11], minería de textos [13], aprendizaje automático [10], todos estos estudios reportan buenas soluciones, todos ellos con un grado de dificultad. Con aprendizaje profundo se obtiene una solución más efectiva, pues gracias a todos los beneficios que nos ofrece con el tratamiento de grandes cantidades de datos por medio de varios niveles. Es por ello, que en este trabajo de investigación se pretende desarrollar un algoritmo basado en redes neuronales para la clasificación automática de noticias de eventos delictivos en el idioma español y la detección automática de zonas de alto riesgo en los estados y municipios de México, además de obtener datos que puedan ser de gran ayuda para la seguridad de las personas y sobre todo para la prevención del delito.

El presente documento se organiza de la siguiente manera: en la sección 2 se presentan los trabajos relacionados con el tema de investigación propuesto.

La sección 3 presenta la propuesta de solución donde se mencionan las técnicas que se aplicaron. La sección 4 presenta los resultados experimentales que se realizaron en la clasificación, así mismo se mencionan los datos que se requirieron para la prueba. Finalmente, la Sección 5 muestra las conclusiones y el trabajo futuro.

2. Trabajo relacionado

En esta sección, se presenta un estudio detallado de los trabajos relacionados con el objetivo de conocer investigaciones anteriores en las tareas de clasificación de noticias periodísticas, los cuales se describen a continuación.

En [11] presentan un enfoque para mejorar los patrones con información morfológica y categorías POS para reconocer y extraer eventos criminales de noticias publicadas en periódicos digitales mexicanos. Se consideran seis eventos criminales, de los grupos principales del CED (Clasificación Estadística de los Delitos) los cuales son: asesinato, violación, asalto, suicidio, secuestro y explotación sexual. El corpus de capacitación está compuesto por 1,600 noticias en español para cada evento criminal. Otra forma de clasificación automática de textos es la dependencia entre variables la cual queda plasmada en forma de enlaces en grafos de palabras co-ocurrentes como se muestra en [7], clasificaron el sentido positivo, negativo o neutral de más de 1,000 mensajes de Twitter escritos en español.

En [13] se profundiza sobre el nivel de criminalidad presente en las noticias policiales las cuales son recopiladas desde julio a diciembre del año 2011, que contiene 23,726 noticias. Se distinguieron siete temáticas, las cuales son: delitos sexuales, incendios, drogas, disturbios, homicidios, tránsito y robos. Se estudiaron dos modelos de clasificación distintos: Naïve Bayes y K-NN, seleccionando el método de evaluación cruzada k-fold (con k=10). También, en [15] extrajeron información sobre desastres naturales a partir de noticias en español con Naïve Bayes, C4.5, k-vecinos más cercanos y máquinas de soporte vectorial. El dominio de extracción se basó en la clasificación de sólo cinco clases de eventos: Forestal, huracán, inundación, sequía y sismo.

Otro enfoque que se aborda en la clasificación, son las aplicaciones de técnicas de aprendizaje automático. En [10] se presenta un proyecto donde aplican estas técnicas a la seguridad, utilizando algoritmos de aprendizaje supervisado, concretamente usan algoritmos de clasificación (*Random Tree*, *Random Forest*, J48, regresión logística, SVM y Naïve Bayes). El conjunto de datos utilizado es "KDD Cup 1999" que contiene un conjunto de conexiones clasificadas como normales o como un ataque determinado.

Además de estos modelos de clasificación en [1] abordaron arboles de decisión, *Random Forest*, Máxima Entropía, SVC Kernel Lineal, NB Multinomial, para el análisis de sentimientos. Se realizaron las pruebas con dos bases de datos alojadas en la web, la primera se llama *Sentiment Labelled Sentences Data Set*, *Movie Review Data Set* y la base de datos de incidencias, proporcionada por la empresa *Cognodata Consulting*.

Posteriormente se aborda el tema de clasificación mediante aprendizaje profundo como se aborda en [9], donde se analizó el uso de modelos de redes convolucionales (CNN), Long short Term Memory (LSTM), LSTM bidireccionales (BI-LSTM) y una aproximación híbrida entre CNN y LSTM para el análisis de sentimiento en Twitter, utilizando el conjunto de datos: InterTASS ES, que permitirá predecir la polaridad de los tweets en base a cuatro niveles: P (Positiva), N (Negativa), NEU (Neutra), NONE (sin opinión).

También en [5] se estudia la aplicación del aprendizaje profundo a la tarea del resumen automático y abstractivo de textos. El conjunto de datos con el que se trabajó se llama CNN DailyMail, que consiste en aproximadamente 300,000 pares de artículos periodísticos y su resumen se analizaron las técnicas aprendizaje profundo enfocadas al procesamiento de lenguaje natural: las redes neuronales recurrentes y los modelos encoder-decoder, entre otras.

En otra investigación que abarca este mismo tema, se presenta en [4] donde se realiza una investigación, la cual tiene como objetivo clasificar los sílabos mediante la técnica de redes neuronales conectadas de aprendizaje profundo. El conjunto de datos con el que se realizó la experimentación consta de 2,316 sílabos. El lenguaje de programación utilizado fue Python.

Además, el modelo fue comparado con respecto a algoritmos basados máquinas de soporte vectorial (SVM), Naïve Bayes y árboles de decisión. Los resultados demostraron que el modelo de aprendizaje profundo propuesto fue superior en 1.4 % con respecto a Naïve Bayes, 6.2 % con respecto a SVM y 7.2 % con respecto a árboles de decisión.

Además, en [2] propone consolidar y preparar un cuerpo con expresiones de texto extraídos de Twitter el cual permite analizar la información a través de la ciencia de datos, con el fin de utilizarlo como insumo esencial para entrenar una red neuronal convolucional (CNN), mediante técnicas de aprendizaje profundo. Como resultado de este entrenamiento, se genera un modelo de predicción de textos que puedan o no presentar signos de cyber acoso (*cyber bullying*), en los cuales se pueda manifestar agresión verbal grave, como insultos, ataques racistas, ataques homofóbicos, etc. Finalmente, los temas abordados en esta sección utilizan las medidas de evaluación tales como: validación cruzada *Macro-*F_1, *precisión*, *recall*, *Exactitud*, *ROUGE-1*, *ROUGE-2* y *ROUGE-L*

En este trabajo se propone el uso de redes neuronales profundas para la clasificación de noticias de eventos criminales. Con los trabajos revisados anteriormente se optó por utilizar este método y se llegó a la conclusión de que el aprendizaje profundo genera buenos resultados, además de que está teniendo mucho auge actualmente. También se propone realizar un análisis de las noticias clasificadas para la obtención de zonas de alto riesgo. Considerando cinco eventos criminales, los cuales son: homicidio, secuestro, asalto, suicidio y violación. El reconocimiento automático de estos cinco eventos criminales y su respectivo análisis, es crucial para que el gobierno tome decisiones sobre la implementación de políticas y estrategias de prevención del delito para prevenir eventos violentos en un futuro cercano.

3. Propuesta de solución

Para llevar acabo el trabajo propuesto se definen dos fases. La primera fase consiste en la clasificación automática de las noticias sobre eventos criminales, pues es necesario que estas noticias estén previamente clasificadas y así llevar a cabo la segunda fase, la cual consiste en la detección de zonas de alto riesgo. Teniendo la clase a la que pertenece cada noticia se procede a obtener las características más importantes, por ejemplo: ubicación, personas, organizaciones y entidades diversas, en este caso los estados y municipios de México y así mostrar estas zonas, en donde suceden con mayor frecuencia estos eventos. Posteriormente en esta sección se describen ambas fases y en las pruebas se profundizan las implementaciones.

A continuación se describe la fase de clasificación de eventos criminales en noticias periodísticas. En la cual se incluye el desarrollo de una red neuronal convolucional basada en aprendizaje profundo. El algoritmo propuesto consiste en los siguientes pasos:

1. Entrada: corpus de entrenamiento y corpus de prueba.
2. Aplicación del algoritmo de clasificación basado en redes neuronales.
3. Evaluación del modelo obtenido con los datos de prueba.
4. Salida: noticias clasificadas.

En la Fig. 1 se pueden observar las fases que se deben seguir para llevar a cabo el algoritmo propuesto.

Fig. 1. Diseño propuesto para la primera fase de clasificación automática de eventos periodísticos.

Para llevar a cabo la primera fase, que es la aplicación del algoritmo de clasificación basado en redes neuronales de los eventos criminales (clases), se realizan los pasos del Algoritmo 1.

Algoritmo 1.1: Algoritmo de clasificación basado en redes neuronales.

Require: Corpus para entrenar, corpus para realizar las pruebas y corpus de palabras preentrenadas [14].

Ensure: Corpus de noticias clasificadas en un csv.

1: Se genera la matriz de incrustaciones de palabras, donde se realiza una representación de cada palabra, donde a cada palabra que sea similar, tendrá una representación similar.
2: Se aplican las redes neuronales CNN y LSTM.
3: En la aplicación de la red neuronal convolucional (Conv1D) se utilizó la función de activación *relu*, un Dropout (0.2, 0.25, 0.275) entre otras.
4: En la aplicación de LSTM se utilizó la función de activación *relu* y *sofmax*, un Dropout(0.25) entre otras.
5: Se aplica el optimizador *Adam*.
6: Se aplica la métrica *Accuracy*.
7: Se realiza la prueba con 10 épocas.

Después de la primera fase, ya que las noticias se encuentran clasificadas en homicidio, secuestro, asalto, suicidio y violación, se procede a la segunda fase que es el análisis de las noticias. Está segunda fase consta de los siguientes pasos (ver Fig. 2):

1. Datos de entrada: corpus de noticias etiquetadas.
2. Aplicación del algoritmo de reconocimiento de entidades con nombre.
3. Datos de salida: Zonas de riesgo.

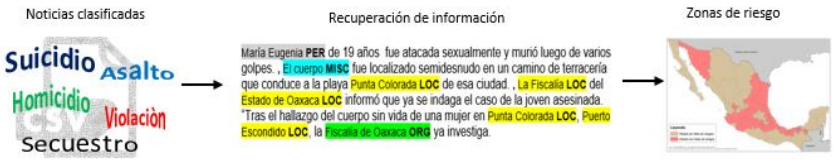


Fig. 2. Diseño propuesto para la segunda fase del análisis de las noticias etiquetadas.

4. Resultados experimentales

En esta sección se presentan los datos utilizados para este experimento. Posteriormente se muestran los resultados de clasificación y el análisis de las noticias.

4.1. Conjunto de datos

Para el conjunto de datos se cuenta con dos corpus, uno para entrenamiento y otro para prueba. El corpus para entrenamiento se recuperó de [11] el cual cuenta con encabezados de noticias obtenido de la red social de Twitter, en este artículo se mencionan seis clases de delitos los cuales son: asesinato, violación, asalto, suicidio, secuestro y explotación sexual. En la Fig. 3 se muestran los datos con lo que se cuenta para el entrenamiento. Estos datos son número del tweet, fecha de la publicación del tweet, medio en el que se transmitió, el tweet que contiene el encabezado de la noticia y la etiqueta a la que pertenece esa noticia.

numero	fecha	medio	tweet	etiqueta
9.32E+17	Fri Nov 17 22:38:04 CST 2017	NTelevisa_com	#EnPunto Peligra atención médica en zona serrana de #Chihuahua tras el secuestro del médico Blas Godínez& https://t.co/j4	Secuestro
9.32E+17	Fri Nov 17 23:07:04 CST 2017	NTelevisa_com	Se metieron a robar un cajero automático con todo y camionetas a un centro comercial de Ecatepec en el #Edomex& https://	Asalto
9.32E+17	Sat Nov 18 00:27:23 CST 2017	NTelevisa_com	Con camionetas y cuerdas intentaron robar un cajero del centro comercial Plaza Aragón, en Ecatepec, #Edomex.& https://t.c	Asalto
9.32E+17	Sat Nov 18 12:01:01 CST 2017	SSP_CDMX	Detuvimos a tres sujetos acusados de robar 24 mil pesos a un hombre de 28 años que ingresaba a su casa en la col. I& https://	Asalto
9.32E+17	Sat Nov 18 14:07:34 CST 2017	elsolde_mexico	Comando ejecuta a dos personas en un bar en Nuevo León https://t.co/Cw9b7mQWJk https://t.co/k6QWyYurBg	Homicidio
9.32E+17	Sat Nov 18 17:00:01 CST 2017	lajornadaonline	#Morrissey cuestiona ola de denuncias de acoso sexual en #Hollywood https://t.co/vXG0zNgnS1 https://t.co/O88iogIbNK	Violación
9.32E+17	Sat Nov 18 18:16:00 CST 2017	Milenio	El #Vaticano investigará abusos sexuales en escuela de #Roma después de la publicación del libro 'El pecado origina& https:/	Violación
9.32E+17	Sat Nov 18 18:40:00 CST 2017	Pajaropolitico	La violencia aumentó en el país aún con la gendarmería: entre 2015 y 2017 los homicidios dolosos se han incrementad& https	Homicidio
9.32E+17	Sat Nov 18 19:36:00 CST 2017	Milenio	'Naturista' envenena a 14 y mata a otra en #Oaxaca https://t.co/GW8l3R9xbG https://t.co/JO7DfLe2pb	Homicidio
9.32E+17	Sat Nov 18 20:10:00 CST 2017	El_Universal_Mx	Los homicidios ocurrieron en los municipios de Naucalpan, Ecatepec y Chalco https://t.co/oPaZSNf8uy	Homicidio
9.32E+17	Sat Nov 18 20:34:01 CST 2017	Pajaropolitico	A un año de que @EPN termine, @anairamzepol y @ccastan3da, reflexionan sobre el aumento de homicidios en México.& h	Homicidio
9.32E+17	Sun Nov 19 00:17:00 CST 2017	Pajaropolitico	Los niños indígenas en México no tienen muchas opciones: son reclutados por el narco o víctimas de homicidio o desa& http:	Homicidio
9.32E+17	Sun Nov 19 04:30:00 CST 2017	NoticiasMVS	.@Uber encara demanda colectiva en #EEUU por agresiones sexuales de conductores https://t.co/zIIb5Yk21p	Violación
9.32E+17	Sun Nov 19 11:55:25 CST 2017	Notimex	15 personas murieron y 5 resultaron heridas durante una avalancha humana en #Marruecos, mientras se repartía ayuda& htt	Homicidio
9.32E+17	Sun Nov 19 13:25:00 CST 2017	El_Universal_Mx	La PGJ cumplimentó las órdenes de aprehensión en su contra por los delitos de robo a casa habitación y robo de vehí& https:	Asalto
9.32E+17	Sun Nov 19 13:45:00 CST 2017	El_Universal_Mx	El asesinato fue perpetrado por hombres armados en el llamado Triángulo Rojo de robo de hidrocarburo https://t.co/aEkdYb	Asalto
9.32E+17	Sun Nov 19 15:55:02 CST 2017	GoogleNewsMX	Silencioso aumento: precio del gas se disparó en más de 30% durante el año https://t.co/hkT3pdT2uL	Suicidio
9.32E+17	Sun Nov 19 16:49:08 CST 2017	Reforma	El director de Izzi, Adolfo Lagos, fue baleado por asaltantes que le robaron una bici en Teotihuacán https://t.co/GnR44RRByD	Homicidio
9.32E+17	Sun Nov 19 18:26:38 CST 2017	NTelevisa_com	Los homicidios contra mujeres y los #feminicidios se duplican en #Zacatecas; en lo que va del año se han registrado& https:/,	Homicidio
9.32E+17	Sun Nov 19 19:12:30 CST 2017	Milenio	#PGR colaborará en caso de asesinato de directivo de #Izzi: @EPN https://t.co/wIdLZ2wo28 https://t.co/SNFuQoEMfE	Homicidio
9.32E+17	Sun Nov 19 19:30:01 CST 2017	elsolde_mexico	#Almomento | @EPN condena asesinato del vicepresidente de @Televisa; @PGR_mx cooperará en investigación& https://t.	Homicidio
9.32E+17	Sun Nov 19 19:51:32 CST 2017	Siete24Noticias	La @PGR_mx participará en la invetigación del homicidio del vicepresidente de @Televisa https://t.co/SVcwHTHZ8f	Homicidio
9.32E+17	Sun Nov 19 20:45:29 CST 2017	El_Universal_Mx	Después de intentar robar a una familia, los delincuentes se escondieron en en una vecindad, lo que provocó un inte& https	Asalto
9.32E+17	Sun Nov 19 22:12:00 CST 2017	Milenio	#PGR colaborará en caso de asesinato de directivo de #Izzi: @EPN https://t.co/wIdLZ2eMDy https://t.co/qogt5Xhve3	Homicidio
9.32E+17	Sun Nov 19 22:40:01 CST 2017	NTelevisa_com	Los homicidios contra mujeres y los #feminicidios se duplican en #Zacatecas; en lo que va del año se han registrado& https:/,	Homicidio
9.32E+17	Mon Nov 20 00:03:56 CST 2017	Reforma	Charles Manson, quien ordenó una serie de asesinatos en 1969, murió hoy a los 83 años https://t.co/oEBdHNuDF5	Homicidio

Fig. 3. Encabezado de noticias de twitter [11].

En este caso para llevar a cabo la clasificación de nuestro interés se tomaron cinco etiquetas, las cuales son: homicidio, secuestro, asalto, suicidio y violación. Esta información fue utilizada para entrenar el algoritmo, como se muestra en la Tabla 1, la cual contiene: los temas de encabezados de noticias de Twitter con las etiquetas respectivas de los eventos delictivos que se clasificarán, la cantidad de noticias y su respectivo vocabulario por tema, además se realiza un recuento total de noticias y su vocabulario

En el caso del corpus para probar, la fuente de información por la que se optó es Milenio una página de noticias local, como se mencionó anteriormente, debido a que esta página nos proporciona un apartado de noticias Policíacas en la cual contiene las noticias de eventos criminales, además de que cuenta con noticias de toda la república mexicana. También se revisaron noticias de otros periódicos digitales, tales como: el Sol de Puebla, el Sol de México y la Prensa, pero solo se proporciona información de temas en general, por lo que no fueron considerados.

Con ayuda del lenguaje de programación Python, se crea un raspado web con las bibliotecas scrapy [8] y BeautifulSoup [12].

Tabla 1. Datos para entrenar.

Tipo de noticias	Cantidad	Total de vocabulario
Homicidio	4,352	95,734
Suicidio	334	5,844
Asalto	2,997	59,568
Secuestro	365	7,183
Violación	666	13,704
Total	8,714	182,033

En el Algoritmo 2 se muestran los pasos que se siguieron para la obtención de las noticias periodísticas por medio de la página local: Milenio. El algoritmo consiste en la identificación de los elementos más importantes de la página de milenio los cuales son: título, contenido, fecha y lugar de la noticia, posteriormente recupera la información de estos elementos en modo texto, se estructura y se genera un archivo csv con la información recuperada de la noticia. Cada noticia está estructurada de la siguiente forma: contienen un título, encabezado, autor, fecha, lugar y el cuerpo de la noticia. En la Fig. 4 se muestran los elementos principales extraídos, con ayuda de la función *add_xpath* se recupera el texto de las etiquetas título, fecha y lugar y para la etiqueta de contenido se recupera el texto con la función *soup.find* de la página html.

Algoritmo 1.2: Algoritmo de raspado web.

Require: Página local Milenio
Ensure: Archivo estructurado csv que contiene título, contenido, fecha y lugar de la noticia
1: Definir las etiquetas a extraer de la página HTML:
2: Título
3: Contenido
4: Fecha
5: Lugar
6: Obtener la URL de la sección "Policía" de la página local (Milenio)
7: Obtener el xpath de la etiqueta título, fecha y lugar de la página HTML
8: Obtener el identificador de la etiqueta contenido de la página HTML
9: Ejecución del código para la recuperación del texto con las funciones *add_xpath* y *soup.find*, de las etiquetas título, contenido, fecha y lugar
10: Eliminar noticias duplicadas.

Después de aplicar el algoritmo de extracción de noticias de la página local Milenio se lograron recuperar 179 noticias del 06 de agosto del 2019 al 27 de noviembre del 2019. En la Tabla 2 se puede observar el tipo de delito (clase), así como el total de vocabulario y la cantidad de noticias en cada delito.

Fig. 4. Variables extraídas de la página local (Milenio).

4.2. Resultados

A través del uso de la red neuronal convolucional (CNN) y una red neuronal llamada memoria a corto y largo plazo (LSTM), se realizó el entrenamiento con 8,714 encabezados de noticias y la prueba se realizó con 179 noticias. Los resultados de la clasificación se pueden ver en la Tabla 3, la medida de evaluación que presentó mejores resultados fue la de exactitud, donde se logró obtener un 98 % de clasificación.

En la segunda fase, sobre el análisis de las noticias, se aplicó un algoritmo de reconocimiento de entidades con nombre, a través de la librería de Python en Spacy [6], donde se utilizó el modelo *es_core_news_sm* que proporciona Spacy para el idioma español, el cual asigna vectores de token específicos de contexto, etiquetas POS, análisis de dependencia y entidades con nombre. Admite la identificación de entidades como nombres de personas (PER), organizaciones (ORG), la ubicación definida política o geográficamente (LOC) y entidades diversas (MISC). En la Fig. 5 se presenta un ejemplo del resultado de la aplicación del algoritmo de reconocimiento de entidades con nombre a una noticia de la página local Milenio.

Tabla 2. Datos para la prueba.

Tipo de noticias (clase)	Total de vocabulario	Cantidad de noticias
Homicidio	40,240	116
Suicidio	0	0
Asalto	7,799	23
Secuestro	12,211	30
Violación	4,046	10
Total	64,296	179

Tabla 3. Resultado de la prueba.

Tipo de noticias	Cantidad de noticias	Clasificadas por el sistema
Homicidio	116	117
Suicidio	0	5
Asalto	23	32
Secuestro	30	17
Violación	10	8
Total	179	179

Como puede observarse se identificaron estados tales como el de Oaxaca (LOC), organizaciones como la fiscalía de Oaxaca (ORG), personas como María Eugenia (PER) y entidades diversas tales como: El cuerpo (MISC).

El cuerpo de una mujer, que había sido reportada como desaparecida por sus familiares fue hallado en las inmediaciones de Puerto Escondido **LOC**, Oaxaca **LOC**. De acuerdo con el reporte de la Comandancia Regional de la Policía Estatal **MISC**, la víctima identificada como María Eugenia **PER** de 19 años fue atacada sexualmente y murió luego de varios golpes. , El cuerpo **MISC** fue localizado semidesnudo en un camino de terracería que conduce a la playa Punta Colorada **LOC** de esa ciudad. , La Fiscalía **LOC** del Estado de Oaxaca **LOC** informó que ya se indaga el caso de la joven asesinada. "Tras el hallazgo del cuerpo sin vida de una mujer en Punta Colorada **LOC**, Puerto Escondido **LOC**, la Fiscalía de Oaxaca **ORG** ya investiga bajo los parámetros establecidos en el protocolo de feminicidios" **MISC**. Tras el hallazgo del cuerpo sin vida de una mujer en Punta Colorada **LOC**, Puerto Escondido **LOC**, la FISCALIA_GobOax **LOC** ya investiga

Fig. 5. Ejemplo de reconocimiento de entidades con nombre.

Después de la aplicación del algoritmo de entidades con nombre, procedemos a la detección de características particulares en el texto tales como el lugar en el que ocurren estos eventos delictivos. Enfocándonos solo en ubicaciones, en la Fig. 6 se presentan las ubicaciones que pueden ser estados y municipios de la república mexicana.

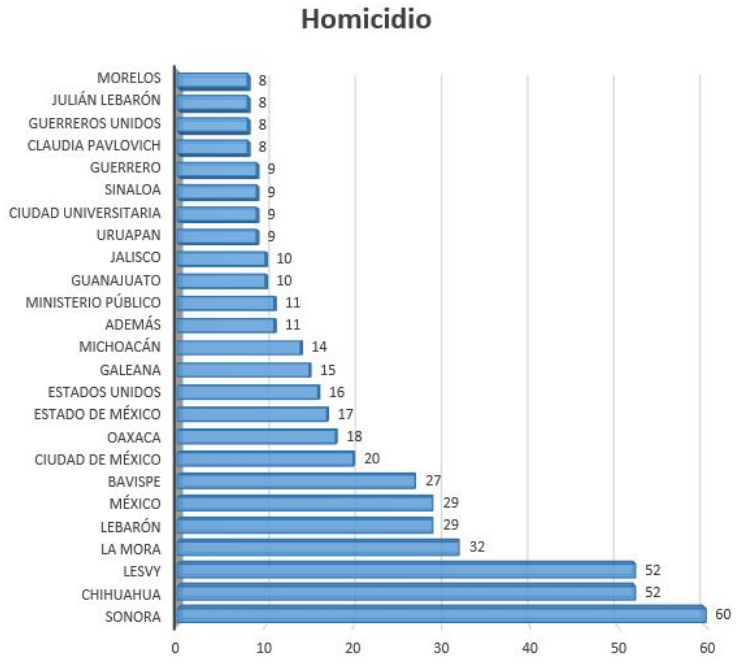


Fig. 6. Gráfica de homicidio.

En la Fig. 6 se observa que el estado de Sonora cuenta con el mayor número de homicidios, seguido del estado de Chihuahua, además se ubica el nombre Lesvy, entre otros. El algoritmo de etiquetado de entidades con nombre muestra algunas ubicaciones las cuales no son necesariamente estados, comunidades o municipios, en este caso el etiquetador cuenta con algunos errores de la extracción de la ubicación.

Las noticias clasificadas en asalto, secuestro, suicidio y violación sexual, presentan un comportamiento muy similar con respecto a los estados obtenidos en la clase de homicidio. Las zonas de alto riesgo identificadas fueron Sonora, Chihuahua, Ciudad de México, Oaxaca, Estado de México, entre otros. Estos estados fueron identificados con el mayor índice de delitos tanto en homicidio como en asalto, secuestro, suicidio y violación sexual. Con los resultados obtenidos en el etiquetador de entidades con nombre se observaron algunos errores, por ejemplo: "La Mora" corresponde con un apellido, el nombre de un rancho y una localidad.

5. Conclusiones

En este trabajo de investigación se desarrolló un algoritmo que consta de dos fases. En la primera fase se desarrolló un algoritmo para la clasificación automática

de eventos delictivos en noticias periodísticas basado en redes neuronales. En la segunda fase se desarrolló un algoritmo para el análisis de las noticas clasificadas, este análisis consiste en la identificación de entidades con nombre, por ejemplo: el estado en la que ocurre el delito. En base a los resultados experimentales en la fase de clasificación se obtuvo una exactitud del 98 %, con esta métrica se obtuvieron los mejores resultados de evaluación. En la fase del análisis de las noticias clasificadas se obtuvieron los estados con mayor frecuencia, resaltando el estado de Sonora, Chihuahua, Ciudad de México, Oaxaca, Estado de México, donde existe un mayor índice de delincuencia con respecto a homicidio, asalto, secuestro, suicidio y violación sexual. Como trabajo futuro se observó que es necesario hacer una mejora en los resultados obtenidos por el algoritmo de entidades con nombre, además de incorporar otras técnicas utilizadas en la literatura como TF-IDF o añadir un recurso para identificar los estados y sus municipios con la finalidad de identificar la frecuencia de los estados presentes en cada noticia, todo ello para mejorar los resultados obtenidos en el etiquetador de entidades con nombre y obtener resultados precisos en cuanto a zonas de alto riesgo.

Agradecimientos. Esta investigación es parcialmente apoyada por el Fondo Sectorial de Investigación para la Educación, con el proyecto CONACyT CB/257357 y por el proyecto VIEP-BUAP 100409344-VIEP2019.

Referencias

1. Álvaro., G.G.: Machine Learning en Bases de Datos de Lenguaje Natural. Master's thesis, Universidad Autónoma de Madrid. Departamento de Ingeniería Informática (2016)
2. Armijos, P.D.C.: Predicción de ataques de cyber bullying mediante técnicas de aprendizaje profundo apoyandose en un corpus de entrenamiento para la clasificación de texto en español. Master's thesis, Universidad Internacional SEK (2018)
3. Beatriz Magaloni, A.D.C.y.V.R.: La raíz del miedo: ¿por qué es la percepción de riesgo mucho más grande que las tasas de victimización? las bases sociales del crimen organizado y la violencia en méxico, josé antonio aguilar ed. mexico city: Secretaría de seguridad pública (2012)
4. Carrera, J.M.A.: Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo (2018)
5. Hernández, A.J.A.: Deep Learning aplicado al resumen de textos. Master's thesis, Universidad Complutense de Madrid (2018)
6. Honnibal, M., Montani, I.: Spacy 2.0: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear
7. Juan Pablo Cárdenas, G.O., Alfaro, R.: Clasificación automática de textos usando redes de palabras. Revista Signos 4(86), 346–364 (2013)
8. Kouzis-Loukas, D.: Learning Scrapy. Packt Publishing Ltd (2016)
9. R. Montañés, R. Aznar, y.R.D.H.: Aplicación de un modelo hibrido de aprendizaje profundo para el análisis de sentimiento en twitter. CEUR Workshop Proceedings 2172, 3027–3036 (2018)

10. Rama, J.M.R.: Aplicación de técnicas de machine learning a la detección de ataques. Master's thesis, Universitat Oberta de Catalunya (2017)
11. Reyes-Ortiz, J.A., Bravo, M.: Enhancing patterns with linguistic information for criminal event recognition. Journal of Intelligent and Fuzzy Systems 34(5), 3027–3036 (2018)
12. Richardson, L.: Beautiful soup documentation. April (2007)
13. Silva, T., Alejandro, D.: Diseño y aplicación de una metodología para análisis de noticias policiales utilizando minería de textos. Master's thesis, Universidad de Chile (2013)
14. Tatman, R.: Pre-trained word vectors for spanish (2019), https://www.kaggle.com/rtatman/pretrained-word-vectors-for-spanish
15. Valero, A.T.: Extracción de Información con Algoritmos de Clasificación. Master's thesis, Instituto Nacional de Astrofísica, óptica y Electrónica (2005)

ISSN 1870-4069

Postura esquelética de la mano para representar las configuraciones dactilológicas de la lengua de señas mexicana en un entorno virtual

Alejandro Gómez-Terán[1], Belém Priego-Sánchez[1], David Pinto[2]

[1] Universidad Autónoma Metropolitana,
Departamento de Sistemas,
México
[2] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{alejandrog.teran, belemps}@gmail.com, dpinto@cs.buap.mx

Resumen. Entre otras formas de comunicación, existen las de modalidad visogestual; tal es el caso de las lenguas de señas desarrolladas por las personas sordas. En este artículo se presentan distintas descripciones de la postura esquelética de la mano con la finalidad de generar una representación de las 29 configuraciones dactilológicas de la lengua de señas mexicana en un entorno virtual. La descripción de la postura esquelética de la mano depende de sus parámetros antropométricos y de los valores de sus ángulos articulares: sus variables de configuración. Así, en este artículo, las configuraciones dactilológicas de la LSM son descritas por medio de 34 variables de configuración de la mano.

Palabras clave: Lengua de señas, configuraciones dactilololológicas, planeación de movimiento.

Skeletal Hand Posture for Signing the dactylological Configurations of Mexican Sign Language in a Virtual Environment

Abstract. Among other forms of communication there are the visual-manual modalities; such is the case of sign languages created by deaf people. In this article some skeletal hand posture descriptions are shown in order to develop a suitable representation of the 29 dactylology configurations of the Mexican sign language within a virtual environment. Skeletal hand posture description depends on its anthropometric parameters and its articular angular values i.e. the hand's configuration values. On the whole, in this article we describe the dactylology configurations of the Mexican sign language by means of 34 configuration parameters of the hand.

Keywords: Sign language, fingerspelling, motion planning.

1. Introducción

Las lenguas de señas son lenguajes de modalidad visogestual que utilizan movimientos de las manos, rostro y cuerpo como forma de comunicación [2]. En México, en el Diario Oficial de la Federación publicado el 10 de junio del 2015, se reconoció a la lengua de señas mexicana (LSM) como la lengua de la comunidad de sordos signantes [5]. La LSM se compone de ideogramas y de dactilología; mientras que los ideogramas representan una palabra con una o varias configuraciones de mano, la dactilología se ocupa de lo que en lengua oral se denomina deletreo y se representa por medio del alfabeto dactilológico de la lengua de señas [15].

El alfabeto dactilológico de la LSM consta de 29 configuraciones; 27 corresponden a las 27 letras del idioma español y dos más se utilizan para representar la 'R' doble y la 'L' doble. Estas configuraciones se pueden caracterizar por medio de la descripción de la postura esquelética de la mano.

La descripción de la postura esquelética de la mano se puede realizar especificando sus parámetros antropométricos y los valores de sus ángulos articulares: las variables de configuración de la mano. Así, en este trabajo realizamos una descripción de las 29 configuraciones dactilológicas de la LSM a través de los valores de las variables de configuración de la mano. Con esta descripción, posteriormente, se desarrollará un entorno virtual para deletrear texto utilizando tales configuraciones.

1.1. Anatomía esquelética descriptiva de la mano

La mano está formada por la muñeca y cinco dedos; en ella hay 27 huesos: ocho carpianos (*ca*) situados en la muñeca, cinco metacarpianos (*mc*) situados en la palma y 14 falanges. Los dedos se identifican enumerándolos del I al V —en alusión al pulgar, índice, medio, anular y al meñique, respectivamente—. El dedo I consta de un *mc* y dos falanges: la proximal (*pp*) y la distal (*dp*). Los dedos II a V poseen, cada uno, un *mc* y tres falanges: una *pp*, una media (*mp*) y una *dp*.

Los huesos de la mano se unen entre sí por medio de articulaciones; 20 en total. La radiocarpiana (RC) une la mano con el antebrazo. Las del dedo I son tres: la carpometacarpiana (CMC), la metacarpofalangiana (MCP) y la interfalangiana (IP). Los dedos II a V poseen cada uno cuatro: la carpometacarpiana (CMC), la metacarpofalangiana (MCP), la interfalangiana proximal (PIP) y la interfalangiana distal (DIP). La ubicación de estas articulaciones se puede apreciar en la Figura 1 que muestra una representación esquelética de la mano derecha y los identificadores que se utilizarán en adelante.

Las articulaciones la mano tienen distinto número de grados de libertad (DOF, por sus siglas en inglés: *degrees of freedom*). El número de DOF de cada articulación corresponde al número de ejes de rotación que posee [16]. Entre las instancias más representativas de modelos de la mano humana el número de DOF considerados varía de 12 a 25 [14].

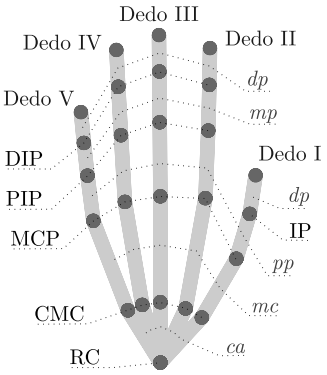


Fig. 1. Identificadores de las articulaciones y los huesos de la mano (vista palmar).

1.2. Movimientos de los elementos de la mano

En este artículo, como en [16], consideramos que los dedos de la mano pueden ejecutar cuatro tipos de movimientos: el de flexión, el cual realizan al acercarse a la palma; el de aducción, el cual ejecutan al acercarse al plano que divide la mano entre los dedos II y III (plano sagital de la mano) y, finalmente, el de abducción que es en aquel que realizan al separarse de dicho plano —lo anterior no proscribe que algunas articulaciones de los dedos puedan rotar ligeramente alrededor del eje longitudinal de MC o PP—.

2. Modelos cinemáticos de la mano humana

Diseñar un sistema para reproducir la LSM en un entorno virtual no es tarea trivial [18]; entre otras, es necesario resolver la de construir un modelo cinemático de la mano que permita caracterizar adecuadamente los movimientos involucrados en la ejecución de las configuraciones dactilológicas.

Un modelo cinemático describe la postura de un sistema de cuerpos y sus relaciones mecánicas. Tales relaciones las definen las dimensiones de sus elementos y cualquier restricción de movimiento involucrada. En el caso de la mano humana, su esqueleto forma un sistema de cuerpos articulados donde las restricciones de movimiento las imponen las articulaciones y las colisiones que pueden ocurrir entre sus elementos. Es decir, un modelo cinemático de la mano debe dar cuenta tanto de sus parámetros antropométricos como de los articulares.

En lo que refiere a los modelos para representar los movimientos de los elementos de la mano en un entorno virtual, estos varían desde los simplificados como estructuras de alambre [19,20], hasta los volumétricos visualmente ricos [1,6,12,11,17]. A continuación, se describen seis modelos cinemáticos de la mano y las características que resultan aquí relevantes para diseñar un modelo adecuado para reproducir las posturas dactilológicas de la LSM.

El modelo [9] (Figura 2a) toma en cuenta 27 DOF; 6 son de la muñeca (tres para ubicación y tres para orientación) y 21 son de los dedos. Considera que el dedo I tiene 5 DOF: dos en CMC, dos en MCP y uno en IP. Y considera que los dedos II a IV poseen cada uno 4 DOF: cero en CMC, dos de MCP, uno en PIP y otro en DIP.

El modelo [16] (Figura 2b) se basa en la formulación Denavit-Hartenberg y utiliza, inicialmente, 26 DOF; todos pertenecientes a los dedos (no toma en cuenta la articulación RC). Considera que el dedo I posee 6 DOF: tres en CMC, dos en MCP y uno en IP. Asume que los dedos II a V poseen, cada uno, 5 DOF: uno en CMC, dos en MCP, uno en PIP y uno en DIP. Al final, después de aplicar restricciones cinemáticas debidas a interdependencias entre las articulaciones MCP, PIP, DIP en los dedos II a V, los autores del modelo lo redijeron 26 a 18 DOF.

El modelo [7] (Figura 2c) se planteó como objetivo modelar movimientos naturales de la mano. Considera 22 DOF; 3 en la muñeca y 19 en los dedos. En el dedo I hay 3 DOF: cero en CMC, dos en MCP y uno en IP. Los dedos II a V poseen 4 DOF: cero en CMC, dos en MCP, uno en PIP y otro en DIP. Al final sus autores concluyen que este modelo no es suficiente para reproducir gesticulaciones arbitrarias.

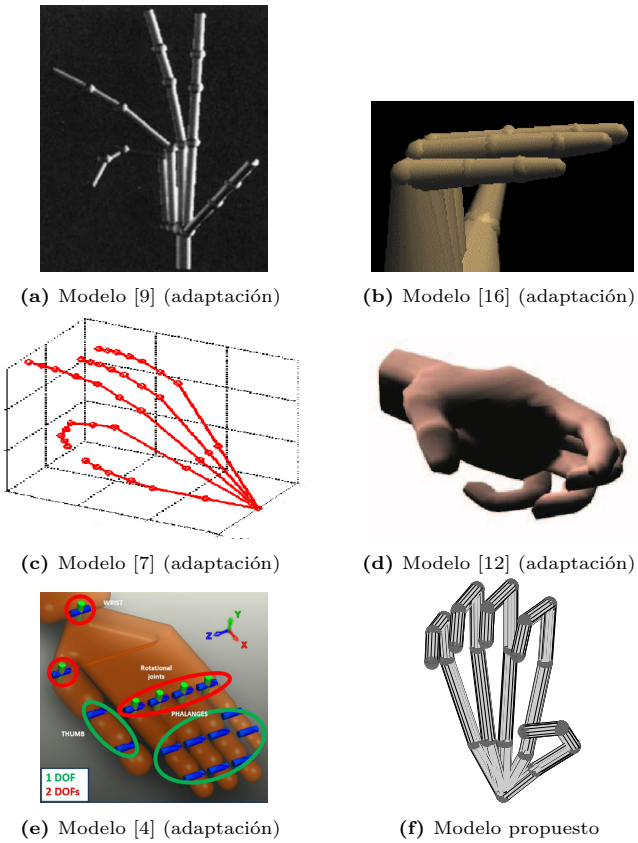
El modelo [13] fue diseñado para imitar lo mejor posible la forma y funciones de la mano. Toma en cuenta 25 DOF; todos pertenecientes a los dedos. El dedo I consta de 5 DOF: dos en CMC, dos en MCP y uno en IP. Los dedos II y III tienen, cada uno, 4 DOF: cero en CMC, dos en MCP, uno en PIP y uno en DIP. Finalmente, el modelo considera que los dedos IV y V constan, cada uno, de 6 DOF: dos en CMC, dos en MCP, uno en PIP y otro más en DIP.

El modelo [12] (Figura 2d) —uno de los que visualmente son más ricos— posee 20 DOF; todos en los dedos. En el dedo I considera que hay 4 DOF: dos en MCP, uno en MCP y otro en IP. En los dedos II a V hay, en cada uno, 4 DOF: cero en CMC, dos en MCP, uno en PIP y otro en DOP. Al final, el modelo utiliza menos variables que el número DOF; lo anterior al que CMC del dedo III carece de movimiento de aducción-abducción y al considerar restricciones cinemáticas de producto de las interdependencias entre las articulaciones PIP y DIP en los dedos II a V.

El último modelo aquí descrito es [4] (Figura 2e); tuvo como propósito de diseño la construcción de un exoesqueleto de mano como ayuda para personas con problemas de apertura de los dedos II a V. Considera 12 DOF; todos en los dedos II a V: cero en CMC y uno en cada articulación MCP. PIP y DIP.

Con lo anterior, tomando como base las descripciones de [4,7,9,12,13,16], el modelo que proponemos en este artículo consta de 34 DOF; 3 en la muñeca y 31 en los dedos. Los tres de la muñeca pertenecen a la articulación RC. El dedo I —el más complejo desde el punto de vista cinemático— posee 7 DOF: tres en CMC, tres en MCP y uno en IP. Los dedos II a V tienen, cada uno, 6 DOF: dos en CMC, dos en MCP, uno en PIP y otro más en DIP. En lo referente a los DOF que asignamos a las articulaciones, adadoptamos la estructura que [13] utiliza para los dedos IV y V, pero la usamos también en II y III.

Y respecto al dedo I, el modelo más parecido al que nosotros proponemos es [16] que le adjudica 6 DOF; en el modelo que proponemos, la articulación MCP posee 3 DOF en vez de 2: añadimos un DOF de rotación alrededor del eje longitudinal de MC. Así, la representación gráfica del modelo propuesto es la que muestra la (Figura 2f).



(a) Modelo [9] (adaptación) **(b)** Modelo [16] (adaptación)

(c) Modelo [7] (adaptación) **(d)** Modelo [12] (adaptación)

(e) Modelo [4] (adaptación) **(f)** Modelo propuesto

Fig. 2. Representación gráfica de los modelos [9,16,7,12,4] y el modelo aquí propuesto.

3. Representación de las configuraciones dactilológicas de la LSM

Obtener aquí un modelo cinemático de la mano tiene como propósito servir como base para realizar una descripción de las configuraciones dactilológicas de LSM en el contexto del diseño e implementación de un sistema computacional

que sea capaz de, deletrear texto mediante la representación de tales configuraciones en un entorno virtual. Dicha representación será gobernada por un subsistema de planeación de movimiento basado en un modelo cinemático de la mano.

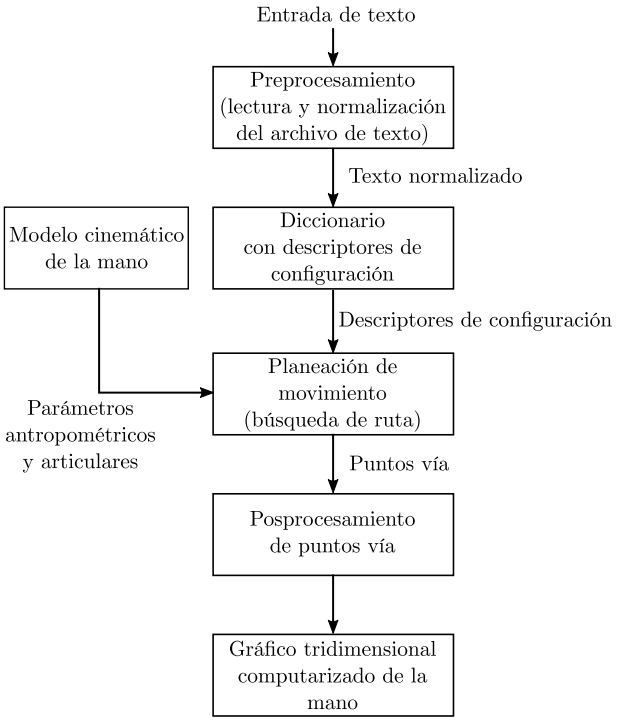


Fig. 3. Diagrama de bloques del sistema computacional de deletreo de texto a través de las configuraciones dactilológicas de la LSM.

Así, el sistema —esquematizado en la Figura 3 y cuyos módulos se describen en la tabla 1— recibirá como entrada un archivo de texto sin formato, con codificación UTF-8, escrito en español y libre de imágenes. A la salida producirá un gráfico tridimensional por computadora de la mano capaz de reproducir las configuraciones dactilológicas de LSM.

3.1. Descripción cinemática de la mano

Como fue mencionado al final de la sección 2, el modelo que proponemos en este artículo para representar las 29 configuraciones dactilológicas de la LSM considera 20 articulaciones de la mano con un total de 34 DOF; 3 en la muñeca y 31 en los dedos. Los tres de la muñeca pertenecen a la articulación RC. El dedo I posee 7 DOF: tres en CMC, tres en MCP y uno en IP. Los dedos II a V tienen,

Tabla 1. Descripción funcional de los módulos del sistema de deletreo de texto a través de las configuraciones dactilológicas de la LSM.

Módulo	Función
Preprocesamiento	Lectura del archivo de entrada y normalización el texto dentro del conjunto de las 27 letras del alfabeto en español y tres marcadores más: los caracteres *espacio*, *punto* y *salto de línea*
Diccionario	Entregar, para cada letra del alfabeto en español, el conjunto de descriptores de la postura esquelética de la mano de la configuración dactilológica correspondiente.
Modelo cinemático	Contiene la descripción de los parámetros antropométricos de una mano y la especificación de las relaciones articulares entre sus elementos.
Planeación de movimiento	Resuelve dos problemas de movimiento: el de búsqueda de espacio para la configuración a ejecutar y el de búsqueda de ruta para generar los movimientos de transición entre la configuración dactilológica inicial y la siguiente. A la salida debe entregar —a modo de ruta entre ambas configuraciones— una secuencia de puntos vía en el espacio generado por los parámetros de configuración de la mano.
Posprocesamiento	Interpretación y adecuación de los puntos vía de la ruta generada en un formato *ad hoc* para el módulo del gráfico tridimensional computarizado de la mano.
Gáfico computarizado tridimensional	Construcción del gráfico que reproduzca inteligiblemente las gesticulaciones correspondientes a las configuraciones dactilológicas de la LSM acorde con el texto de entrada.

cada uno, 6 DOF: dos en CMC, dos en MCP, uno en PIP y otro más en DIP. Lo anterior y las coordenadas de las articulaciones consideradas se muestran en la tabla 2; tales coordenadas se encuentran definidas en relación a un sistema de coordenadas inercial que no forma parte del modelo y se ubica en la apófisis estiloides radial (en el sitio donde se unen el antebrazo y la mano).

Respecto a los ejes de rotación de las articulaciones, estos corresponden a líneas en $\mathbb{R}^3$ de las que optamos por describir su orientación a través de un sistema de coordenadas donde, como en [3], la orientación es descrita por medio de una terna de *ángulos de Euler*. La terna seleccionada corresponde a la secuencia de rotaciones denominada *Roll-Pitch-Jaw*; donde *Roll* es al ángulo de rotación (α) alrededor del eje x, *Pitch* el ángulo (β) alrededor del eje x y *Yaw* el ángulo (γ) alrededor del eje Z — es decir, secuencia 321 o Zyx— como en [8]. Así, la tabla 3 muestra las ternas de orientación de los ejes de movimiento de los 34 DOF de las articulaciones de la mano cuando se encuentra en reposo. En dicha tabla, los subíndices r, p, y indican, respectivamente, rotación, cabeceo y guiñada de la mano; y los subíndices r, a, f indican, respectivamente eje de rotación, eje de aducción-abducción y eje de flexión-extensión del resto de las articulaciones.

Las últimas características de la mano por describir son las ubicaciones de la punta de los dedos. Dichas ubicaciones, cuando la mano se encuentra en postura de reposo, se muestran en la tabla 4.

Tabla 2. Cantidad de DOF y coordenadas de las articulaciones del modelo propuesto para describir la postura de la mano.

			Ubicación [mm]		
Grupo	Articulación	DOF	x	y	z
Muñeca					
	RC	3	0	0	0
Dedo I					
	CMC	3	23	-4	25
	MCP	3	40.03	-6.993	58.986
	IP	1	47.204	-13.778	83.828
Dedo II					
	CMC	2	14	1	30
	MCP	2	25.1	5.9	91
	PIP	1	26.606	4.872	128.464
	DIP	1	27.086	1.236	153.564
Dedo III					
	CMC	2	0	2	33.5
	MCP	2	0.01	9	90
	PIP	1	-0.332	6.284	132.957
	DIP	1	-0.579	2.296	160.891
Dedo IV					
	CMC	2	-10	1	32
	MCP	2	-19.619	5.082	89.139
	PIP	1	-21.679	3.582	126.079
	DIP	1	-23.058	0.752	152.823
Dedo V					
	CMC	2	-18	-0.5	29
	MCP	2	-37.143	0.928	78.6
	PIP	1	-40.319	0.133	103.615
	DIP	1	-42.072	-2.581	121.611

Con lo anterior, en el modelo cinemático propuesto, la posición en $\mathbb{R}^3$ (ubicación y orientación) de cada DOF es aquí expresada mediante un sistema de coordenadas cuyo origen define la ubicación de la articulación y donde el eje de rotación de la articulación es colineal con el eje Z de tal sistema. Este sistema es codificado utilizando la matriz ${}^{h}\boldsymbol{F}$ de transformación homogénea (1); donde (x, y, z) corresponden las coordenadas de la articulación y que se muestran en la tabla 2; y donde α, β, γ son los ángulos RPY mostrados en la tabla 3. Es decir, por cada uno de los DOF del modelo, existe una matriz (1) que establece su posición en términos del sistema de referencia inercial cuando la mano se encuentra en la postura de reposo (h).

Así, dado que cada dedo puede ser considerado como una cadena cinemática abierta que inicia en RC y termina en la punta del dedo, la posición, en términos del sistema de referencia inercial (sistema 0), de su i-ésimo elemento en función de los ángulos de articulación $(\theta_1, \theta_2, ..., \theta_i)$ está dada por la transformación (2);

Tabla 3. Orientación de los 34 ejes de movimiento del modelo propuesto para describir la postura de la mano.

		Orientación en ángulos *RPY* [grados sexagesimales]		
Grupo	Eje de movimiento	α	β	γ
Muñeca				
	RC_r	0	0	90
	RC_p	0	-90	0
	RC_y	45	-90	45
Dedo I				
	CMC_r	25.431734	17.148859	52.139412
	CMC_a	54.300595	-59.649168	-6.058481
	CMC_f	-35.699405	-59.649168	-6.058481
	MCP_r	19.532247	-0.618973	34.98159
	MCP_a	91.85077	-70.458331	-56.762641
	MCP_f	1.85077	-70.458331	-56.762641
	IP_f	30.416905	-68.271632	-87.81352
Dedo II				
	CMC_a	81.199843	-64.927192	4.085617
	CMC_f	-8.800157	-64.927192	4.085617
	MCP_a	27.704791	-83.78587	57.740348
	MCP_f	-62.295209	-83.78587	57.740348
	PIP_f	54.551232	-85.869683	-59.568265
	DIP_f	78.596457	-80.713926	-83.970312
Dedo III				
	CMC_a	-0.163644	-86.583412	90
	CMC_f	-90.163644	-86.583412	90
	MCP_a	-4.064264	-83.159258	94.119652
	MCP_f	-94.064264	-83.159258	94.119652
	PIP_f	96.796548	-86.17686	-97.173502
	DIP_f	92.959173	-80.860215	-93.189486
Dedo IV				
	CMC_a	-81.620963	-73.031315	177.280555
	CMC_f	-171.620963	-73.031315	177.280555
	MCP_a	-44.76741	-83.874907	137.772348
	MCP_f	-134.76741	-83.874907	137.772348
	PIP_f	149.342867	-83.798346	-146.530651
	DIP_f	121.831765	-81.900256	-118.976747
Dedo V				
	CMC_a	-80.719227	-67.509726	-175.841279
	CMC_f	-170.719227	-67.509726	-175.841279
	MCP_a	-66.901464	-75.482075	167.098877
	MCP_f	-156.901464	-75.482075	167.098877
	PIP_f	151.999798	-78.333119	-146.248576
	DIP_f	134.714313	-76.020383	-130.322515

donde $\boldsymbol{R}_{\mathrm{Z}}(\theta)$ es la matriz de transformación homogénea para rotar alrededor del eje Z, ${}^{0}\boldsymbol{F}_{0}$ es la matriz identidad y cφ, sφ representan respectivamente, en notación compacta, cos(φ) y sin(φ).

$$ {}^{h}\boldsymbol{F} = \begin{bmatrix} c\beta c\gamma & -c\alpha s\gamma + s\alpha s\beta c\gamma & s\alpha s\gamma + c\alpha s\beta c\gamma & x \\ c\beta s\gamma & c\alpha c\gamma + s\alpha s\beta s\gamma & -s\alpha c\gamma + c\alpha s\beta s\gamma & y \\ -s\beta & s\alpha c\beta & c\alpha c\beta & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1} $$

$$ {}^{0}\boldsymbol{F}_{i}(\theta_{1,2,\cdots,i}) = {}^{0}\boldsymbol{F}_{i-1}(\theta_{1,2,\cdots,i-1}) \, ({}^{h}\boldsymbol{F}_{i-1})^{-1} \, {}^{h}\boldsymbol{F}_{i} \, \boldsymbol{R}_{\mathrm{Z}}(\theta_{i}), \tag{2} $$

Tabla 4. Coordenadas de las puntas de los dedos del modelo propuesto para describir la postura de la mano.

	Ubicación [mm]		
Puntas de los dedos	x	y	z
Punta dedo I	47.022	-20.967	104.798
Punta dedo II	27.102	-3.506	173.436
Punta dedo III	-0.76	-1.4	180.773
Punta dedo IV	-23.963	-3.225	172.88
Punta dedo V	-43.378	-6.701	138.255

$$\boldsymbol{R}_{\mathrm{Z}}(\theta) = \begin{bmatrix} c\theta & -s\theta & 0 & 0 \\ s\theta & c\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

3.2. Descriptores de las configuraciones dactilológicas de la LSM

Con el modelo cinemático implementado, las configuraciones dactilológicas de la LSM pueden ser descritas por medio de los ángulos articulares descritos; es decir, por una variable de configuración para cada uno de los 34 DOF del modelo.

En particular, cuando el valor de las 34 variables de articulación es cero, el modelo de la mano luce como en la Figura 1; esta postura será utilizada como postura de reposo y será la postura a mostrar siempre que el modelo virtual no se encuentre ejecutando alguna configuración dactilológica de la LSM.

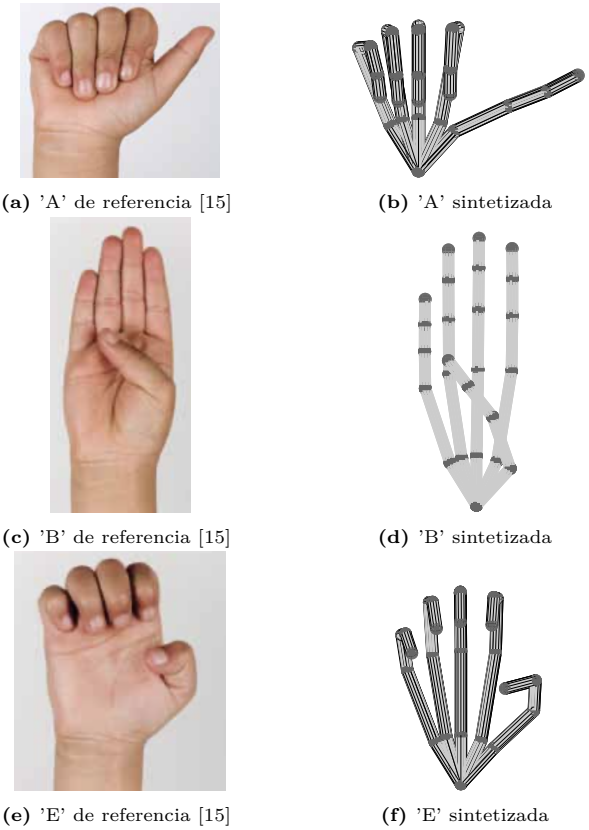
Cuando se analizan las posturas individuales de los dedos en las configuraciones dactilológicas de la LSM, se tiene indicio de que algunas pueden ser utilizadas como base para describir a muchas de las restantes; tal es el caso de las configuraciones 'A', 'B', 'C' y 'E'. Así, para el modelo que implementamos, los ángulos de las configuraciones 'A', 'B', y 'E' se muestran en la tabla 5; y las representaciones preliminares de tales configuraciones sintetizadas por el modelo —junto con las imágenes de referencia obtenidas de [15]— se muestran en las Figuras 4a, 4b, 4c, 4d, 4e y 4f.

4. Conclusiones y perspectivas

Las lenguas de señas son lenguajes utilizados por personas sordas signantes como medio de comunicación. En México la LSM es la lengua reconocida como la de la comunidad de sordos signantes. En la LSM la dactilología se ocupa de lo que en lengua oral se denomina deletreo y consta de 29 configuraciones. En este artículo utilizamos un modelo cinemático para describir la postura esquelética

Tabla 5. Ángulos articulares para representar tres de las configuraciones dactilológicas de la LSM con el modelo cinemático propuesto.

		Ángulos de configuración [grados sexagesimales]		
Grupo	Variables articulares (θ)	'A'	'B'	'E'
Muñeca				
	RC_r	0	0	0
	RC_p	0	0	0
	RC_y	0	0	0
Dedo I				
	CMC_r	-55	25	0
	CMC_a	-25	-35	-18
	CMC_f	-42	15	10
	MCP_r	-30	15	0
	MCP_a	-3	-8	0
	MCP_f	-4	8	-35
	IP_f	5	11	-80
Dedo II				
	CMC_a	1	0	1
	CMC_f	-2	0	0
	MCP_a	0.5	0	0
	MCP_f	3	-90	6
	PIP_f	4	-90	-86
	DIP_f	5	-24	-68
Dedo III				
	CMC_a	-1	0	0
	CMC_f	-2	0	0
	MCP_a	0	0	0
	MCP_f	4	-90	5
	PIP_f	2	-90	-88
	DIP_f	3	-24	-80
Dedo IV				
	CMC_a	-1	0	0
	CMC_f	0.5	0	0
	MCP_a	-2	0	0
	MCP_f	2	-90	8
	PIP_f	2	-90	-92
	DIP_f	3	-24	-60
Dedo V				
	CMC_a	-3	1	0
	CMC_f	1	0	0
	MCP_a	-4	3	0
	MCP_f	2	-90	10
	PIP_f	5	-90	-90
	DIP_f	4	-18	-50



(a) 'A' de referencia [15] **(b)** 'A' sintetizada

(c) 'B' de referencia [15] **(d)** 'B' sintetizada

(e) 'E' de referencia [15] **(f)** 'E' sintetizada

Fig. 4. Configuracióones dactilológicas 'A', 'B' y 'E' de la LSM.

de la mano y, con él, caracterizar las configuraciones dactilológicas de la LSM por medio de los ángulos articulares de la mano.

Luego de analizar distintos modelos con diferente número de DOF, decidimos describir la postura esquelética de la mano y sus 20 articulaciones mediante un modelo cinemático con 34 DOF; 3 en la muñeca y 31 en los dedos. Con este modelo fue posible representar inteligiblemente las 29 configuraciones dactilológicas de la LSM.

Como trabajo futuro se plantea probar los valores de las variables de configuración obtenidos en la síntesis de un sistema computacional de deletreo de palabras (como el mostrado en la Figura 3). En dicho sistema, las letras en una palabra serán agrupadas en pares —configuración inicial y configuración objetivo— para que el módulo de planeación de movimiento genere los movimientos de la mano para ir de una configuración a otra sin que los dedos colisionen entre sí: una ruta libre de colisiones para mover un sistema de objetos en el espacio.

Referencias

1. Albrecht, I., Jörg, H., Hans-Peter, S.: Construction and animation of anatomically based human hand models. In: Proceedings of the ACM SIGGRAPH, pp. 98–109 (2003)
2. Caballero-Morales, S.O., Trujillo-Romero, F.: 3D Modeling of the Mexican sign language for a speech-to-sign language system. Computación y Sistemas, 17(4), pp. 593–608 (2013)
3. Caccavale, F., Natale, C., Siciliano, B., Villani, L.: Six-DOF impedance control based on angle/axis representations. IEEE Transactions on Robotics and Automation, 15(2), pp. 289–300 (1999)
4. Conti, R., Meli, E., Ridolfi, A.: A novel kinematic architecture for portable hand exoskeletons. Mechatronics, 35, pp. 192–207 (2016)
5. Cruz-Aldrete, M.: Manos a la obra: Lengua de señas, comunidad sorda y educación. Bonilla Artigas Editores, Universidad Autónoma del Estado de México (2014)
6. Efthimiou, E., Fotinea, S.E., Vogler, C., Hanke, T., Glauert, J., Bowden, R., Braffort, A., Collet, C., Maragos, P., Segouat, J.: Sign language recognition, generation and modelling: A research Effort with applications in deaf communication. In: Stephanidis, C., (ed.), Universal access in human-computer interaction. Lecture Notes in Computer Science, Addressing Diversity, 5614, pp. 21–30 (2009)
7. Fahime, B.E., Haddad, M.Y.E., Radouani, M., Saka, A.: Modeling the natural movements of the human hand. In: International Conference on Bioengineering & Biomaterials: IC2B'09, pp. 10, Meknes, Morocco (2009)
8. Jazar, R.N.: Theory of applied robotics. Springer (2010)
9. Lee, J., Kunii, T.: Model-based analysis of hand posture. IEEE Computer Graphics and Applications, 15(5), pp. 77–86 (1995)
10. Lee, J., Kunii, T.L.: Constraint-based hand animation. In: Thalmann, N.M., Thalmann, D. (eds.), Models and Techniques in Computer Animation, pp 110–127, Springer Japan, Tokyo (1993)
11. López-Colino, F., Colás, J.: Hybrid paradigm for spanish sign language synthesis. Universal Access in the Information Society, 11(2), pp. 151–168 (2012)
12. Mulatto, S., Formaglio, A., Malvezzi, M., Prattichizzo, D.: Using postural synergies to animate a low-dimensional hand avatar in haptic simulation. IEEE Transactions on Haptics, 6(1), pp. 106–116 (2013)
13. Parida, P.K., Biswal, B.B.: Design and analysis of a multifingered robot hand. IAES International Journal of Robotics and Automation IJRA, 1(2), pp. 69–77 (2012)
14. Rosell, J., Suarez, R., Rosales, C., Garcia, J.A., Perez, A.: Motion planning for high DOF anthropomorphic hands. In: IEEE International Conference on Robotics and Automation, pp. 4025–4030 (2009)
15. Serafín, M.E., González, R.: Manos con voz: Diccionario de lengua de señas mexicana. Consejo Nacional para Prevenir la Discriminación. (2011)
16. Vardy, A., Brown, E.: Computer graphics project report: Articulated human hand model with inter-joint dependency constraints. (1998)
17. Verschoor, M., Lobo, D., Otaduy, M.A.: Soft hand simulation for smooth and robust natural interaction. In: IEEE Conference on Virtual Reality and 3D User Interfaces VR, pp. 183–190 (2018)
18. Wheatland, N., Wang, Y., Song, H., Neff, M., Zordan, V., Jörg, S.: State of the art in hand and finger modeling and animation. Computer Graphics Forum, 34(2), pp. 735–760 (2015)

19. You, W.S., Lee, Y.H., Kang, G., Oh, H.S., Seo, J.K., Choi, H.R.: Kinematic design optimization of anthropomorphic robot hand using a new performance index. In: 14th International Conference on Ubiquitous Robots and Ambient Intelligence URAI, pp. 20–25 (2017)
20. Zelinka, J., Kanis, J., Salajka, P.: NN-based Czech sign language synthesis. In: Salah, A.A., Karpov, A., Potapova, R. (eds). Speech and computer, 11658, pp. 559–568, Cham, Springer International Publishing (2019)

ISSN 1870-4069

Calibración de hiperparámetros para la clasificación de requisitos de software mediante evolución diferencial

J. Manuel Pérez-Verdejo[1], Efrén Mezura-Montes[2],
Ángel Juan Sánchez-García[1], Jorge Octavio Ocharán-Hernández[1]

[1] Universidad Veracruzana,
Facultad de Estadística e Informática,
México
[2] Universidad Veracruzana,
Centro de Investigacion en Inteligencía Artificial,
México

manuel.vrdjo@gmail.com, {emezura,jocharan}@uv.mx,
angelsg89@hotmail.com

Resumen. La clasificación automática de requisitos representa un área de interés dentro de la Ingeniería de Software. Es a partir del tipo de requisito, que se toman estrategias para todo el ciclo de vida de un sistema. Trabajos previos han presentado enfoques de clasificación por aprendizaje máquina sobre este tópico, sin embargo, la selección de los hiperparámetros y su efecto sobre la precisión de estos modelos permanecen como problemas abiertos. En este estudio, se presenta una propuesta para la calibración de estos parámetros a través del algoritmo de evolución diferencial DE/rand/1/bin. Este procedimiento se probó con un modelo de Naïve Bayes y se comparó contra la herramienta para la optimización TPOT. Los resultados preliminares muestran un incremento de la precisión del modelo hasta por un 7 %, evaluado por validación cruzada, y una precisión máxima de 69.7 % con el modelo generado por TPOT.

Palabras clave: Evolución diferencial, aprendizaje máquina automático, ingeniería de requisitos, algoritmos evolutivos, clasificación de requisitos.

Calibration of hyperparameters for the classification of software requirements using differential evolution

Abstract. Automatic classification of requirements represents an area of interest within Software Engineering. It is from the type of requirement that strategies are taken for the entire life cycle of a system. Previous

work has presented machine learning classification approaches on this topic, however, the selection of hyperparameters and their effect on the precision of these models remain open problems. In this study, a proposal is presented for the calibration of these parameters through the differential evolution algorithm DE/rand/1/bin. This procedure was tested with a Naïve Bayes model and compared against the TPOT optimization tool. Preliminary results show an increase in model precision of up to 7%, evaluated by cross-validation, and a maximum precision of 69.7% with the model generated by TPOT.

Keywords: Differential evolution, machine learning, requirements engineering, evolutionary algorithms, classification of requirements.

1. Introducción

Uno de los objetivos fundamentales de la Ingeniería de Requisitos (IR) se enfoca en entender y documentar las necesidades y deseos de los interesados en un producto de software [7]. Dentro de las actividades de la IR, se encuentra la especificación, en la que un analista plasma los intereses y necesidades de los usuarios en enunciados estructurados conocidos como requisitos de software [21].

Para la especificación y posterior administración de los requisitos identificados, es necesario que sean separados a partir de la categoría a la que pertenecen, pues esta determina la estrategia que se sigue para que se pueda satisfacer [25]. De estas categorías, los Atributos de Calidad resultan especialmente críticos, pues su correcta identificación afecta de forma directa la arquitectura del sistema [5].

Este análisis suele consumir mucho tiempo y ser propenso a errores, además de repetitivo, pues se efectúa de nuevo cuando se agregan o modifican los requisitos [2]. Es por estos motivos que se han desarrollado diferentes enfoques para la clasificación automática de requisitos de software [8,10,11]. De la misma forma, la implementación de estos modelos de aprendizaje máquina implican una combinación de parámetros que determinan su composición. Estos, denominados hiperparámetros, influyen en el rendimiento y precisión de los clasificadores [12]. Es por eso que la calibración de estos datos se representa un paso importante en el entrenamiento de modelos de aprendizaje máquina.

Este tipo de problemas combinatorios son viables para ser abordados por técnicas de optimización como los algoritmos evolutivos. Estos algoritmos ya ha sido explorados con anterioridad en el área de la Ingeniería de Software, específicamente para la configuración de sistemas auto-adaptativos [6].

Los algoritmos evolutivos realizan la búsqueda de soluciones competitivas mediante conjuntos de soluciones potenciales llamados poblaciones. Estos algoritmos están inspirados en el proceso de la evolución natural [1]. Especialmente, los algoritmos de evolución diferencial son fáciles de implementar para la optimización, pues utilizan pocos parámetros que suelen mantenerse fijos a lo largo del proceso evolutivo [22].

Por lo anterior, este estudio plantea la aplicación de algoritmos evolutivos, específicamente de evolución diferencial para la calibración de los hiperparámetros de un clasificador de requisitos de software. Lo anterior con el objetivo de maximizar el rendimiento y precisión del mismo.

El resto de este documento está organizado de la siguiente forma: la Sección 2 presenta el problema de optimización de hiperpárametros y su aplicación en la clasificación de requisitos. La Sección 3 expone la contribución, una propuesta de la aplicación del algoritmo DE/rand/1/bin para la optimización de hiperparámetros. En la Sección 4 se describen los experimentos y los resultados obtenidos. Finalmente, la Sección 5 resume los hallazgos identificados y el trabajo futuro.

2. Definición del problema

La configuración de modelos de aprendizaje máquina afecta en gran medida su desempeño [12]. La selección de estas características se complica en la medida del clasificador que se selecciona. La exploración de las posibles combinaciones de hiperparámetros, representan una oportunidad para mejorar el rendimiento de un modelo, cuando este es comparado con una comparación con valores por defecto.

Calibrar este tipo de valores para maximizar el rendimiento de modelos de aprendizaje máquina a través de métodos evolutivos ha sido explorado anteriormente. En [4] se exploran diferentes algoritmos evolutivos para incrementar el rendimiento de modelos de Árboles de Decisión; a partir de este estudio, se identifican ventajas de utilizar este tipo de optimizaciones para adaptar algoritmos a dominios específicos. De la misma forma, en [12] se exploran estrategias de calibración de hiperparámetros para clasificadores en el contexto de información biomédica.

Sin embargo, y hasta donde los autores han podido verificar, no se han identificado estrategias de calibración de hiperparámetros para clasificadores en el dominio de la Ingeniería de Requisitos. Dado que estás estrategias se han explorado para otros ámbitos de la Ingeniería de Software [6], se realiza la propuesta de aplicar este conocimiento para clasificar requisitos de software.

De esta forma, el objetivo principal de este estudio es aplicar el algoritmo de evolución diferencial DE/rand/1/bin [22] para explorar combinaciones de hiperparámetros que permitan maximizar la precisión de un clasificador de requisitos de software.

3. Contribución

En este estudio, se presenta una propuesta de optimización de hiperparámetros para un modelo de clasificación Naïve Bayes. Para este fin se estableció un procedimiento de extracción de características para el entrenamiento del modelo de aprendizaje máquina, y posteriormente se realizó la calibración del mismo utilizando Evolución diferencial. La Figura 1 muestra el proceso llevado a cabo.

3.1. Clasificación de requisitos

La compatibilidad de texto con el clasificador por aprendizaje máquina se realizó a través de un proceso de vectorización y codificado de etiquetas. Este procedimiento de extracción de características se realizó de la siguiente manera:

1. Remoción de caracteres especiales. De esta forma se quitan caracteres de salto de línea o tabulaciones.
2. Conversión de texto a minúsculas.
3. Remoción de signos de puntuación. Tales como puntos, comas y signos de admiración o interrogación.
4. Lematización. Así, se reemplazan los sufijos de las palabras para obtener una palabra raíz más básica. De esta forma se reduce la variabilidad de palabras [9].
5. Remoción de palabras vacías (*stop words*). Se retiran palabras predefinidas que no enriquecen semánticamente el enunciado.
6. Vectorización por TF-IDF. *Term Frequency - Inverse Document Frecuency* por sus siglas en inglés es una técnica de representación de términos a través de su peso en una determinada categoría [24]. Esta representación de términos por pesos está definida por la Ecuación 1:

$$W(d,t) = TF(d,t) \times log(\frac{N}{df(t)}). \tag{1}$$

A partir de este modelo es posible calcular el peso de un término con respecto a la frecuencia con la que aparece en un conjunto de texto $W(d,t)$. Donde t corresponde al término y d al documento. Como primer factor en esta ecuación se calcula la frecuencia del término (*Term Frequency*), al mismo tiempo que la

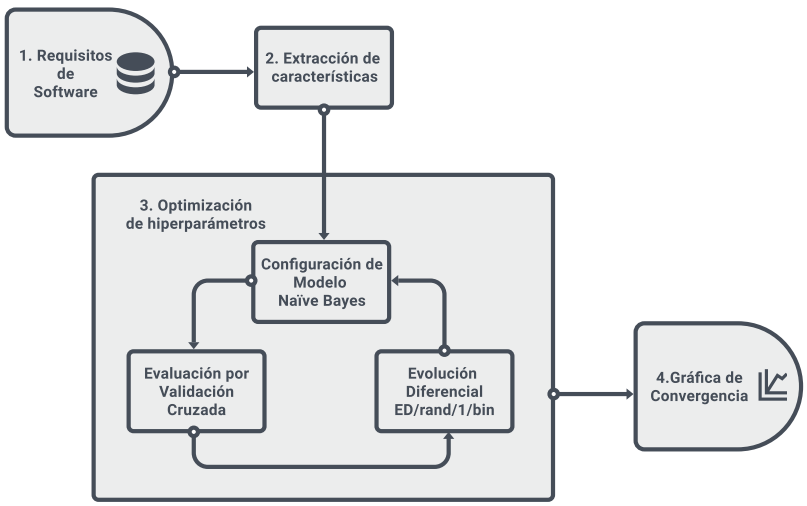


Fig. 1. Procedimiento para la optimización del modelo de aprendizaje máquina.

frecuencia inversa en relación a los documentos (*Inverse Document Frequency*) se representa en el segundo. En este factor se calcula qué tan específico es un término, donde N es el número total de textos y $df(t)$ es el número de textos que contienen al término t [19].

Una vez aplicada la transformación de los conjuntos de datos, se realiza la selección y entrenamiento con valores por defecto del clasificador. Se selecciona para este estudio el modelo MultinomialNB incluido en la biblioteca sklearn [17]. Para este fin, el clasificador seleccionado únicamente cuenta con dos hiperparámetros, denominados `alpha` y `fit_prior`, siendo valores de tipo real y booleano respectivamente. Para corroborar el proceso de optimización, se utilizó el modelo de Naïve Bayes con su configuración por defecto, y se comparó con un modelo con el mismo entrenamiento, pero con los parámetros resultantes del proceso de evolución diferencial.

3.2. Optimización por evolución diferencial

Con el objetivo de maximizar la precisión del modelo, se implementó el algoritmo de evolución diferencial de DE/rand/1/bin [22]. La representación para el conjunto de soluciones se realiza de forma vectorial, y se inicializa de forma aleatoria a lo largo de una población inicial.

Para la ejecución de este algoritmo, únicamente se definen cuatro valores. El número de individuos presentes en cada generación, la cantidad de generaciones que se ejecutarán; un factor de escala que representa la diferencia para calcular el vector de mutación, y finalmente el porcentaje de cruza, que determina el parecido entre individuos padres e hijos.

Este proceso evolutivo se realiza a lo largo de cada generación, en la que prevalecen aquellas combinaciones de hiperparámetros que generan una precisión mayor. La evaluación se realiza través de la configuración de un modelo MultinomialNB con los parámetros de los individuos. Para este proceso se define como función de aptitud a maximizar, la validación cruzada. Lo anterior con el fin de minimizar el sobre ajuste al conjunto de datos de entrenamiento.

Debido a que el proceso de mutación y cruza resulta en individuos con parámetros aleatorios que pueden exceder los límites deseados, se utilizó la reflexión como técnica de manejo de restricciones de límite. Esta estrategia se expresa por la Ecuación 2, así se reingresa cualquier valor fuera de los límites [18]:

$$x_i^c = \begin{cases} x_i & \text{si } l_i \leq x_i \leq u_i, \\ 2 \times l_i - x_i & \text{si } x_i < l_i, \\ 2 \times u_i - x_i & \text{si } x_i > u_i. \end{cases} \tag{2}$$

3.3. TPOT

Adicionalmente, se comparó el proceso de optimización aplicado con la biblioteca TPOT. Esta herramienta realiza la configuración automática de una cadena de modelos de *machine learning* denominada *pipeline* [16]. A través de

un proceso de optimización por programación genética [3], esta herramienta es capaz de diseñar y configurar *pipelines* de transformaciones de datos y modelos de aprendizaje máquina con el fin de maximizar la precisión de un modelo de clasificación [15]. La Figura 2 muestra el procedimiento que involucra esta herramienta y su uso para la clasificación de requisitos de software.

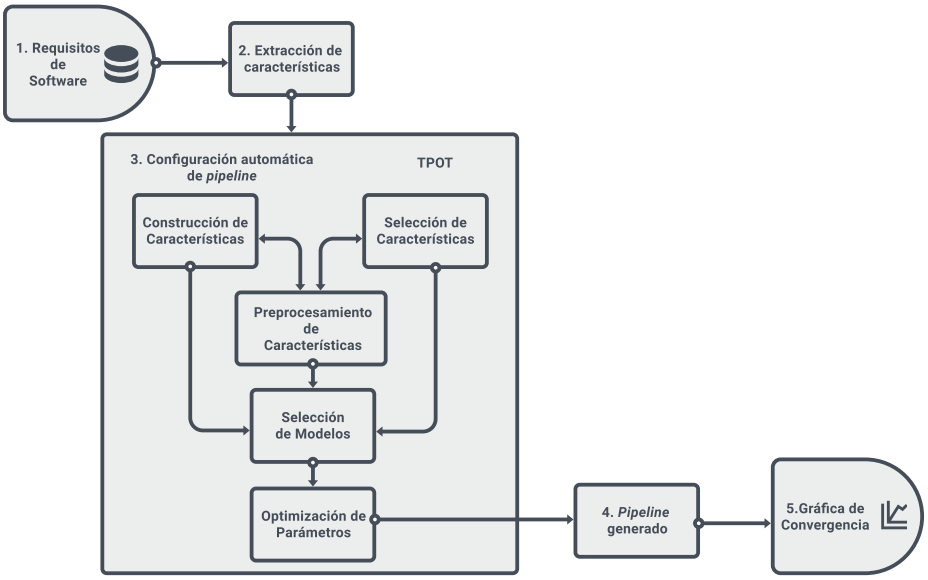


Fig. 2. Proceso de configuración de modelo generado por la herramienta TPOT.

4. Experimentos y resultados

Para evaluar la precisión del clasificador con los requisitos de software, se utilizaron dos bases de datos con enunciados de este tipo, la base de PROMISE [20] y 40 ejemplos recolectados de *The Quest for Software Requirements* [14]. Este proceso de extracción y optimización propuesto se puede observar en la Figura 1. De estas dos bases de datos, se recolectaron aquellos requisitos que estuvieran asociados con los Atributos de Calidad siguientes: Disponibilidad, Tolerancia a Fallos, Mantenibilidad, Rendimiento, Escalabilidad, Seguridad y Usabilidad.

A partir de los requisitos en formato de texto, se realizó el proceso de extracción de características antes descrito, con el cual se representó el formato de lenguaje en espacios de vectores para su subsecuente uso en el entrenamiento del modelo Naïve Bayes configurado con parámetros por defecto.

Table 1. Distribución de las clases del conjunto de entrenamiento.

Atributo de Calidad	Frecuencia
Tolerancia a Fallos	10
Mantenibilidad	25
Disponibilidad	29
Escalabilidad	29
Rendimiento	54
Seguridad	74
Usabilidad	75

Estos valores, de acuerdo a la documentación de la biblioteca sklearn [17] son $alpha = 1.0$ y $fit_prior = True$. Evaluado a través de validación cruzada por 5 dobleces, este modelo obtuvo una precisión ($accuracy$) promedio de 0.546.

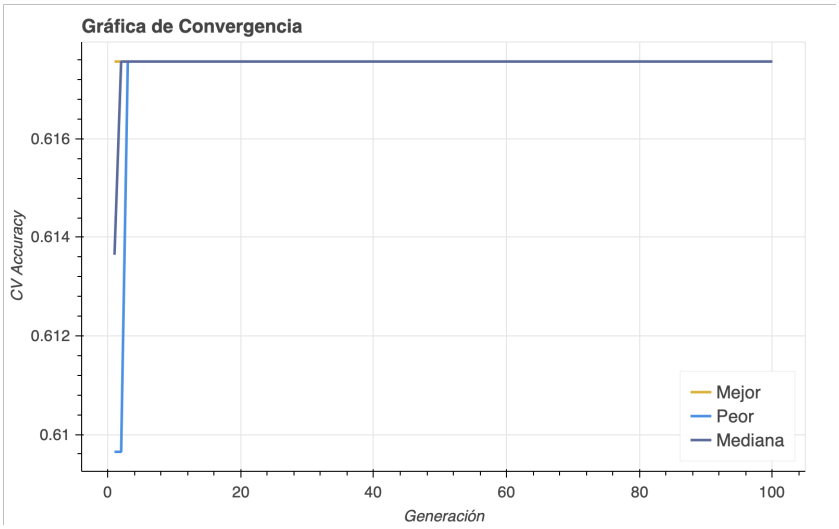


Fig. 3. Gráfica de convergencia del algoritmo DE/rand/1/bin.

Para la aplicación del algoritmo de evolución diferencial, se realizó una configuración basada en el estudio [13]. De esta forma, los parámetros con los que se realizaría el proceso de evolución son un tamaño de población de 100 individuos, a través de 100 generaciones; usando un factor de escala de 0.9 y un porcentaje de cruza de 1.0. Como función de aptitud, los individuos fueron evaluados en los mismos términos que el modelo por defecto. De esta forma, el proceso evolutivo se llevó a cabo con el objetivo de maximizar la precisión promedio del modelo, evaluada por validación cruzada.

Así mismo, y dado que la aplicación de Evolución Diferencial se enfoca en exploración de espacios continuos [23], se limitaron espacios de búsqueda de los parámetros, siendo el espacio $[0, 100]$ para el valor *alpha* y $[0, 1]$ para fit_prior. Para este último dado que es de tipo booleano, se determinó que aquellas soluciones con un valor de 0.5 o superior se consideraría como "Verdadero", mientras que las soluciones menores a 0.5 serían interpretadas como un valor "Falso".

Finalmente, y dada la naturaleza no determinista de este tipo de algoritmos [23], se realizaron 30 ejecuciones del proceso de Evolución Diferencial. De estas, se identificaron y compararon aquellas con la mejor, peor y mediana aptitud obtenida. La Figura 3 muestra la gráfica de convergencia de estas ejecuciones. En esta representación se muestra el proceso de optimización con respecto de la precisión del mejor individuo de una generación, a lo largo del proceso evolutivo. En esta gráfica, se observa que el algoritmo alcanza una convergencia durante las primeras generaciones. Notablemente, todas las ejecuciones alcanzaron este estado con una precisión de 0.617, aún con combinaciones de parámetros diferentes. Esto se puede deber a muchos factores, tales como el espacio de búsqueda de los parámetros, la cantidad de los mismos (pues en el caso de este modelo son dos), el método de representación de lenguaje seleccionado o el dataset mismo.

De la misma forma, se realizaron 30 ejecuciones de la herramienta TPOT con el mismo dataset preprocesado. El objetivo de optimización del algoritmo se configuró para ser el mismo que el aplicado para el DE/rand/1/bin, siendo maximizar el valor de precisión por validación cruzada, en 5 dobleces. La gráfica de convergencia en la Figura 4 muestra la comparación de las ejecuciones mejor, peor y mediana. En esta gráfica se puede apreciar la forma en la que gradualmente se incrementa el valor de precisión. En el caso de la mejor ejecución, se logró obtener una precisión de 0.697, mientras que en la peor se logra 0.661, ambas puntuaciones superando al máximo alcanzado por el clasificador Naïve Bayes optimizado. Adicionalmente, la información de las 30 ejecuciones de ambos algoritmos, así como su comparación son accesibles en el repositorio del proyecto[3].

5. Conclusiones y trabajo futuro

Las aplicaciones de aprendizaje automático (*AutoML*) así como de optimización son de especial interés en el área de Ingeniería de Software, pues su uso ha mostrado resultados favorables en diferentes etapas del desarrollo de sistemas, dada la cantidad de procesos aptos para ser optimizados [6,26]. Con el interés de clasificar de forma automática requisitos de software, se realizó el entrenamiento de un modelo Naïve Bayes. De la misma forma, se aplicó el algoritmo de Evolución Diferencial DE/rand/1/bin, que como objetivo tuvo maximizar la precisión para clasificar este tipo de información. Adicionalmente, se probó el uso de una herramienta de generación de secuencias de modelos, conocida como TPOT. Esta última implementación mostró una estrategia de

[3] https://github.com/quality-attributes/optimization

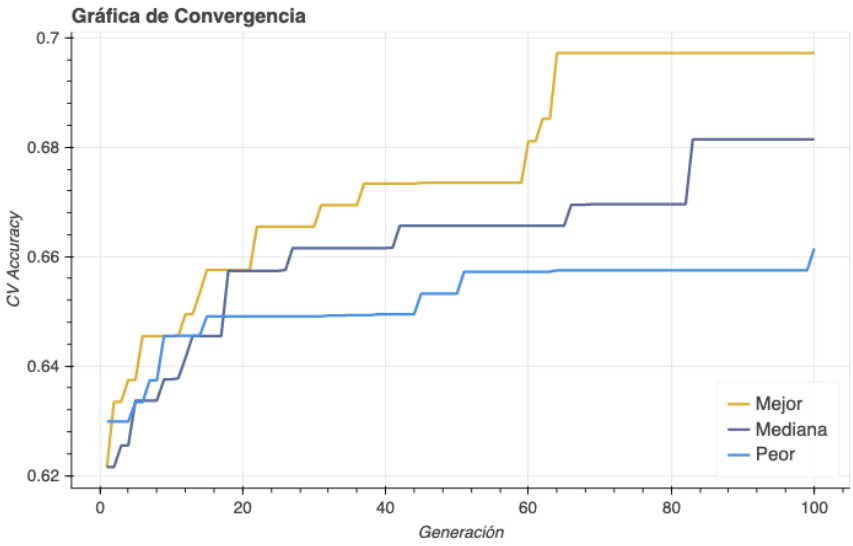


Fig. 4. Gráfica de convergencia de TPOT.

diseño generación de modelos más elaborados, calibrados a partir de programación genética. Ambos acercamientos mostraron resultados favorables para la calibración de parámetros de modelos de aprendizaje máquina, pues demostraron capacidad para incrementar su precisión. De está forma, fue posible mejorar la precisión promedio del clasificador con los parámetros por defecto, pasando de 54.6% a 61.7% una vez ejecutado el proceso de calibración Naïve Bayes mediante Evolución Diferencial. No obstante, el *pipelane* generado por la herramienta de TPOT mostró una puntuación superior al modelo optimizado, obteniendo 69.7% en términos de precisión.

Durante la conducción del presente estudio, se identificaron áreas de oportunidad y trabajo futuro para la aplicación de algoritmos de optimización en la clasificación de requisitos de software. Particularmente, se identifica la implementación de modelos de aprendizaje máquina adicionales, con el objetivo de complementar los resultados obtenidos con el modelo Naïve Bayes. De esta forma, se pueden examinar modelos con más hiperparámetros, permitiendo la exploración en más espacios de búsqueda, y posiblemente de más combinaciones que incrementen la precisión de los mismos.

Finalmente, se plantea la aplicación de las técnicas de optimización en más etapas del proceso de clasificación de requisitos propuesto, especialmente en el procedimiento de representación de texto. Así, se propone la exploración de más procedimientos de procesamiento de lenguaje [9], así como sus respectivas configuraciones. Lo anterior el objetivo de incrementar la compatibilidad de los modelos con el conjunto de datos de requisitos de software.

References

1. Alcalá, R., Gacto, M.J., Alcalá-Fdez, J.: Evolutionary data mining and applications: A revision on the most cited papers from the last 10 years (2007–2017). WIREs Data Mining and Knowledge Discovery 8(2) (mar 2018), https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1239
2. Ambriola, V., Gervasi, V.: On the Systematic Analysis of Natural Language Requirements with CIRCE. Automated Software Engineering 13(1), 107–167 (jan 2006), http://link.springer.com/10.1007/s10515-006-5468-2
3. Banzhaf, W., Francone, F.D., Keller, R.E., Nordin, P.: Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)
4. Barros, R.C., Basgalupp, M.P., de Carvalho, A.C.P.L.F., Freitas, A.A.: A Survey of Evolutionary Algorithms for Decision-Tree Induction. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(3), 291–312 (may 2012)
5. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley Professional, 3rd edn. (2012)
6. Chen, T., Li, M., Li, K., Deb, K.: Search-Based Software Engineering for Self-Adaptive Systems: One Survey, Five Disappointments and Six Opportunities (jan 2020), http://arxiv.org/abs/2001.08236
7. Glinz, M.: A Glossary of Requirements Engineering Terminology. Tech. rep., International Requirements Engineering Board (2017)
8. Jindal, R., Malhotra, R., Jain, A.: Automated classification of security requirements. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp. 2027–2033 (2016)
9. Kowsari, K., Meimandi, K.J., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: A survey (2019)
10. Kurtanović, Z., Maalej, W.: Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning. In: 2017 IEEE 25th International Requirements Engineering Conference (RE). pp. 490–495 (2017)
11. Lu, M., Liang, P.: Automatic Classification of Non-Functional Requirements from Augmented App User Reviews. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. pp. 344–353. EASE'17, ACM, New York, NY, USA (2017)
12. Luo, G.: A review of automatic selection methods for machine learning algorithms and hyper-parameter values. Network Modeling Analysis in Health Informatics and Bioinformatics 5(1), 18 (dec 2016)
13. Mezura-Montes, E., Miranda-Varela, M.E., del Carmen Gómez-Ramón, R.: Differential evolution in constrained numerical optimization: An empirical study. Inf. Sci. 180(22), 4223–4262 (Nov 2010)
14. Miller, R.E.: The Quest for Software Requirements. MavenMark Books, Oconomowoc, WI, USA (2009)
15. Olson, R.S., Bartley, N., Urbanowicz, R.J., Moore, J.H.: Evaluation of a tree-based pipeline optimization tool for automating data science. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 485–492. GECCO '16, ACM, New York, NY, USA (2016)
16. Olson, R.S., Urbanowicz, R.J., Andrews, P.C., Lavender, N.A., Kidd, L.C., Moore, J.H.: Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings,

Part I, chap. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization, pp. 123–137. Springer International Publishing (2016)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
18. Ronkkonen, J., Kukkonen, S., Price, K.: Real-Parameter Optimization with Differential Evolution. In: 2005 IEEE Congress on Evolutionary Computation. vol. 1, pp. 506–513. IEEE (2005)
19. Salton, G.: Automatic text processing: The transformation, analysis, and retrieval of information by computer (1989)
20. Sayyad Shirabad, J., Menzies, T.J.: The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada (2005), http://promise.site.uottawa.ca/SERepository
21. Sommervile, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. John Wiley & Sons, Inc., New York, United States (1997)
22. Storn, R.: On the usage of differential evolution for function optimization. In: Proceedings of North American Fuzzy Information Processing. pp. 519–523 (jun 1996)
23. Storn, R., Price, K.: Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. Journal of Global Optimization 23 (01 1995)
24. Tokunaga, T., Iwayama, M.: Text categorization based on weighted inverse document frequency (1994)
25. Wiegers, K.E., Beatty, J.: Software Requirements. Microsoft Press, USA, 3rd edn. (2013)
26. Zhang, D., Tsai, J.J.P.: Machine Learning and Software Engineering. Software Quality Journal 11(2), 87–119 (2003)

ISSN 1870-4069

Mecanismo de representación para la evolución automática de circuitos analógicos

Aurora Torres Soto[1], María Dolores Torres Soto[2],
Eunice Esther Ponce de León Sentí[1]

[1] Benemérita Universidad Autónoma de Aguascalientes,
Departamento de Ciencias de la Computación, México
[2] Benemérita Universidad Autónoma de Aguascalientes,
Departamento de Sistemas de Información,
México

{mdtorres, atorres, epoce}@correo.uaa.mx.com

Resumen. Este trabajo expone el problema de diseño de circuitos analógicos y presenta un mecanismo de representación para su diseño automático mediante herramientas de la computación evolutiva. Para mostrar la flexibilidad y potencialidad de nuestra propuesta, se incluyen ejemplos de la codificación de tres circuitos, entre los que se encuentran un filtro pasa-bajas de Butterworth de cuarto orden y uno de Chebyshev de tercer orden. Mediante estos ejemplos, se muestra que el mecanismo es muy compacto, que codifica simultáneamente topología y tamaño de los elementos de circuito y que es capaz de manejar valores de dispositivos disponibles comercialmente, lo que reduce las discrepancias entre los circuitos diseñados evolutivamente y los implementados físicamente. Finalmente se hace acopio de las bondades que presenta nuestro mecanismo de representación de circuitos, entre las que sobresale la posibilidad de construir circuitos con toda clase de topología, abonando al uso de la inteligencia artificial en este campo.

Palabras clave: Circuitos analógicos, computación evolutiva, diseño automático, representación de redes eléctricas, filtros pasa-bajas.

Representation Mechanism for Automatic Evolution of Analog Circuits

Abstract. This work exposes the analog circuit design problem and presents a representation mechanism for its automatic design using evolutionary computing tools. To show the flexibility and potentiality of our proposal, examples of the coding of three circuits are included, among them are a fourth order Butterworth low pass filter and a third order Chebyshev low pass filter. Using these examples, it is shown that the mechanism is very compact, that it simultaneously encodes topology and size of circuit elements, and that it is capable of handling commercially available device values, reducing discrepancies between evolvable designed and physically implemented circuits. Finally, the benefits presented by our circuit representation mechanism are gathered, among which the possibility

of building circuits with all kinds of topology stands out, contributing to the use of artificial intelligence in this field.

Keywords: Analog circuits, evolutionary computing, automatic design, representation of electrical networks, low-pass filters.

1. Introducción

El diseño de circuitos es considerado como una tarea artesanal debido a la gran cantidad de conocimientos, habilidades y experiencia que implica de los diseñadores. La complejidad de su naturaleza, le ha hecho objeto para el desarrollo de diferentes herramientas que pretenden asistir a los ingenieros en sus diferentes etapas. Actualmente las herramientas de diseño de circuitos digitales han alcanzado una enorme versatilidad; sin embargo, el escenario en relación al diseño de circuitos analógicos es muy diferente.

Aunque desde finales de los años 70, los circuitos analógicos han sido reemplazados por los digitales, muchas funciones permanecen siendo analógicas debido a la naturaleza analógica de los transductores [1]. Agregando a esto, el hecho de que el diseño analógico es una tarea intensiva, muy consumidora de tiempo y que requiere de habilidades y conocimientos especializados [2], no debe sorprender que se realicen diferentes esfuerzos para asistirlo.

El diseño electrónico tiene como objetivo la obtención de un circuito funcionalmente correcto, lo más sencillo y eficiente posible y en el menor plazo de tiempo. El uso de herramientas de simulación ha modificado sustancialmente el proceso de diseño, pues ahora es posible realizar numerosos análisis de forma automática; antes de llevar el diseño a su implementación.

Por otro lado, el enorme éxito que los algoritmos evolutivos han tenido en el diseño de circuitos digitales [3-6] entre otros, ha motivado el uso de estas herramientas en el diseño de circuitos analógicos. Este trabajo propone abordar el diseño analógico mediante el uso de la computación evolutiva; lo que implica, que se establezcan al menos, un mecanismo de representación y una función objetivo como lo menciona [7].

El área conocida como electrónica evolutiva aparece en 1998 [8]; a partir de entonces, su objetivo ha sido el diseño de circuitos electrónicos mediante la creación de algoritmos evolutivos. Uno de los principales retos a resolver cuando se usa electrónica evolutiva, es la forma como se codificará el circuito [1], así es que en este trabajo se propone un mecanismo de representación de circuitos analógicos que sea suficientemente flexible para la construcción de diferentes tipos de circuitos y que permita la evolución simultánea de la topología del circuito y del establecimiento de los tipos de elementos de circuito y sus valores. Este mecanismo es descrito detalladamente y sus posibilidades son discutidas mediante la descripción de dos filtros pasa-bajas convencionales, reservando su implementación en herramientas evolutivas para trabajos futuros.

En la siguiente sección se habla sobre el problema de diseño de circuitos analógicos, en la sección 3 se establecen los conceptos básicos de la computación evolutiva y en la sección 4 se expone el mecanismo de representación propuesto; así como algunos conceptos relacionados con los filtros analógicos. Finalmente, en la sección 5 se discuten las conclusiones del trabajo.

2. Diseño de circuitos

El diseño o síntesis de circuitos, es el proceso de creación de un dispositivo electrónico que cumple un grupo de especificaciones previamente establecidas. Este proceso comprende dos operaciones básicas: el establecimiento de la disposición de los elementos de circuito "topología" y la determinación del valor y tipo de cada elemento del circuito "tamaño"[9]; sin embargo, visualizando el proceso de diseño desde la definición del dispositivo a construir, podemos identificar tres fases fundamentales [2]:

1) El establecimiento de las especificaciones del circuito,

2) La determinación de la topología del circuito,

3) La optimización y establecimiento de los valores de los parámetros.

Respecto al establecimiento de las especificaciones de un circuito, es el paso en el que se define el problema de diseño; la optimización y establecimiento de los valores de los parámetros, suponiendo una topología fija, es una parte que ha sido abordada por múltiples investigadores [10, 11, 12], mediante el uso de métodos deterministas, estocásticos e híbridos; sin embargo, aún no se ha resuelto la generación automática de la topología.

En la mayor parte de los trabajos relacionados con el diseño automático, los algoritmos evolutivos se han empleado exclusivamente para la tercera fase (optimización y establecimiento de los valores de los parámetros), habiendo establecido previamente la topología por diseñadores expertos o habiendo partido de un grupo de posibles topologías [13], sin embargo, estas técnicas dependen en gran medida de la experiencia y conocimiento de los diseñadores.

Aun cuando se trabaja con topologías convenientes y fijas, con frecuencia la optimización y el establecimiento de los valores de los parámetros es insuficiente para alcanzar el desempeño esperado, de manera que se debe modificar la topología [14]; lo que implica que el proceso de optimización de parámetros deba comenzar desde el principio [2]. Este escenario confirma que la síntesis automática de la topología de los circuitos analógicos y de radio frecuencia, es actualmente un problema abierto como lo establecen Sorkhabi y Zhang en [15].

Por otro lado, el enorme costo que implica la construcción de nuevas bibliotecas para cada nuevo circuito; ha ocasionado que el diseño mediante la selección de topologías fijas estén siendo desplazadas por su generación automática [16].

La generación automática de la topología se refiere a la construcción de la estructura que un circuito va a presentar, es decir, el establecimiento de la forma como se van a conectar los elementos que lo forman partiendo de una "hoja en blanco"[13]; enfoque adoptado en este trabajo.

3. Computación evolutiva

Desde un punto de vista simple, la evolución natural es un proceso de optimización en el que el objetivo es mejorar la habilidad de los organismos o sistemas para sobrevivir en un medio ambiente dinámico y competitivo [17].

Algoritmo Evolutivo General

1. Sea G=0 el contador de generaciones
2. Crea e inicializa una población P(0) n-dimensional
3. Repite
   - Evalúa la aptitud f(x), de cada individuo x, en la población P(G)
   - Produce la descendencia
   - Selecciona la población P(G+1) de la nueva generación
   - Avanza a la nueva generación. G=G+1
4. Hasta que la condición de paro sea cumplida

Fig. 1. Algoritmo evolutivo genérico.

El hecho de que la evolución se comporte como un proceso de depuración y mejora de las especies, ha motivado a la comunidad científica para adoptar algunos de sus mecanismos en la solución de problemas. La computación evolutiva es el área de las ciencias computacionales que utiliza modelos computacionales del proceso evolutivo, tales como la selección natural y la supervivencia y reproducción de los individuos más aptos, para la solución de problemas basados en la computadora [17]. Aunque existe una amplia variedad de algoritmos evolutivos, todos tienen en común que implementan un proceso evolutivo para formar un algoritmo de optimización. La figura 1, resume los aspectos más relevantes de un algoritmo evolutivo; sin embargo, se debe mencionar que algunos paradigmas pueden variar de este patrón.

Algunos de los principales paradigmas de la computación evolutiva, así como los trabajos clásicos sobre ella o literatura que se recomienda, se listan a continuación:

1. Algoritmos Genéticos [18, 19].
2. Programación Genética [20].
3. Programación Evolutiva [21].
4. Algoritmos de Estimación de la Distribución[22].
5. Estrategias Evolutivas [23].
6. Evolución Diferencial [24].

Los primeros algoritmos que emularon la evolución genética con resultados prácticos fueron los algoritmos genéticos, razón a la que deben su enorme popularidad; sin embargo, el mecanismo de representación que se propone en este trabajo puede ser adoptado por cualquier técnica de la computación evolutiva.

4. Descripción de la propuesta

Todas las herramientas de generación de la topología, usan como punto de partida una plantilla a la que se conectan los elementos [13]. En el caso de las redes eléctricas de dos puertos (categoría a la que pertenecen los filtros analógicos pasivos), es

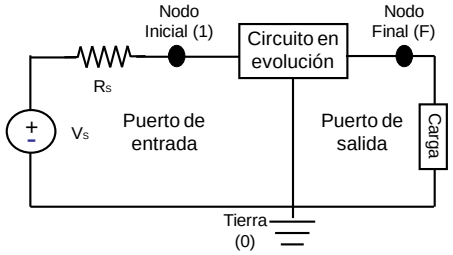


Fig. 2. Plantilla para una red eléctrica de dos puertos.

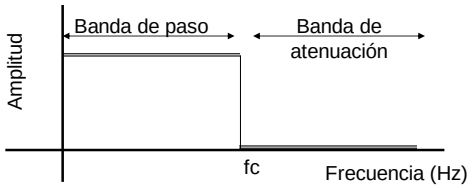


Fig. 3. Respuesta en frecuencia de un filtro ideal.

frecuente que la plantilla en la que se evoluciona el circuito, esté constituida de los nodos que representan al puerto de entrada y al puerto de salida del circuito. La plantilla empleada en este trabajo se muestra en la figura 2. Este patrón ha sido utilizado en la evolución de diferentes tipos de filtros [25-28 y 16].

De esta figura se puede apreciar que los nodos de entrada de la red eléctrica (puerto de entrada) son el nodo 1 "Nodo inicial" y el nodo 0 "Tierra", mientras que la salida (puerto de salida) se encuentra entre el nodo F "Final" y el nodo 0 "Tierra". Tanto el proceso de generación de la topología, como el de determinación de los valores de los elementos (establecimiento del tamaño), tienen lugar entre los tres nodos mencionados. Naturalmente, esta plantilla puede ser cambiada de acuerdo al objetivo del diseño.

Para este trabajo se analiza el mecanismo de representación suponiendo que el objetivo de diseño es un filtro analógico pasivo, es decir, un circuito que se constituye exclusivamente de elementos de circuito pasivos. Esta decisión obedece al hecho de que este tipo de red eléctrica es la que ha sido mas frecuentemente utilizada por los investigadores del área, pues el filtrado es una disciplina bien comprendida [27].

Un filtro electrónico puede considerarse como un dispositivo selectivo a la frecuencia, que permite dar forma de una manera prescrita a la respuesta en magnitud o en fase [29]. En otras palabras, es un circuito que bloquea las señales de cierto rango de frecuencias, mientras permite el paso de aquellas con frecuencias en distinto rango [30].

La figura 3 muestra lo que se conoce como la respuesta en frecuencia de un filtro pasa-bajas ideal. Este filtro, de acuerdo con su nombre, permite el paso de las señales eléctricas cuya frecuencia se encuentre por debajo de la frecuencia de corte (fc), atenuando completamente las de frecuencia superior a fc.

Cabe mencionar que la respuesta descrita en la figura anterior corresponde a un modelo teórico ideal, pues con dispositivos reales, no es factible obtener este comportamiento; sin embargo, en la medida en la que la respuesta real se aproxime a la ideal, el filtro es de mejor calidad.

Para medir la funcionalidad de los filtros, éstos son sometidos a diferentes condiciones de frecuencia de la fuente de poder, mientras que se evalúa el nivel de voltaje a la salida.

Este procedimiento es conocido como barrido de frecuencias, produciendo la respuesta en frecuencia del circuito.

Si se ajusta la amplitud de un generador de voltaje a "Vin" volts y su frecuencia a "F" Hz, mientras se mide la salida de voltaje de un filtro "Vout" volts, se puede obtener la ganancia de voltaje como la relación de Vout/Vin. Este valor se mide en decibeles [dB] y está dado por la ecuación 1:

$$Hv = 20log\frac{V_{out}}{V_{in}}. \tag{1}$$

De manera que la ganancia de voltaje de un filtro en su banda de paso puede ser de 0dB, mientras que, en la banda de atenuación, se espera que exhiba ganancias negativas.

4.1. Representación de Circuitos

A fin de emplear una herramienta evolutiva para la solución de un problema complejo, el primer paso es el establecimiento de la representación de una posible solución al problema. Al mecanismo empleado para representar un individuo (solución), en términos que la computación evolutiva ha heredado de la biología, se le conoce como cromosoma.

Una representación adecuada es fundamental para que el proceso de optimización sea exitoso y es conveniente que exhiba las siguientes características [13]:

- Flexibilidad: Codificación suficientemente adaptable para representar una amplia variedad de individuos.
- Sencillez: Decodificación tan fácil como sea posible.
- Robustez: Capacidad para soportar las modificaciones ejecutadas por los operadores genéticos.

Teniendo en consideración que la evolución en nuestro problema debe producirse tanto a la topología como a la determinación de los valores de los elementos (tamaño), es decir, que los operadores genéticos deben ser capaces de introducir diversidad no solamente en la forma como se conectan los elementos de circuito, sino en el número de elementos que lo constituyen y en los tipos y valores de estos, se optó por crear una técnica de representación de desarrollo. De acuerdo a Mattiussi y Floreano [31], aunque éstas son más complicadas que las técnicas de representación directas; son más compactas aunque requieren de un mecanismo que dirija el proceso de desarrollo (construcción de la red).

En este trabajo, el mecanismo de construcción de la red es el algoritmo de generación de circuitos cuya discusión esta fuera del alcance de este trabajo, pero que esta descrito en [13].

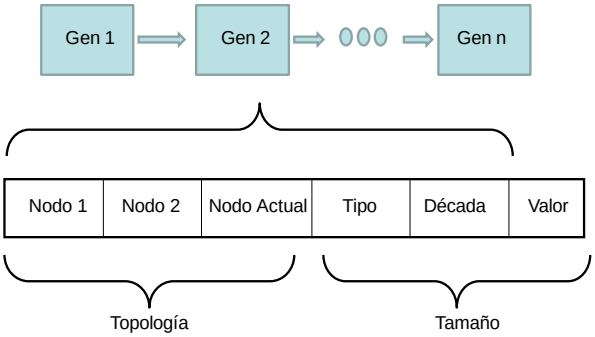


Fig. 4. Cromosoma y descripción de un gen.

Un circuito es representado por cromosomas de longitud variable, pues de esta manera no se requiere de conocimiento experto del problema y no se limita la potencialidad de exploración de los algoritmos evolutivos [32]. Un cromosoma (solución candidata o circuito), es representado por una lista enlazada de nodos (genes); donde cada nodo (gen) representa a un elemento de circuito con información de su topología y de su dimensión. La figura 4 muestra el cromosoma diseñado para la evolución de circuitos analógicos pasivos. En ésta se presenta la manera como un cromosoma se constituye de genes y el tipo de información que almacena cada gen.

De la figura anterior se aprecia que cada elemento de circuito posee información sobre el lugar que ocupa en la red eléctrica (primeros dos campos) y sobre su naturaleza y tamaño (últimos tres campos).

4.2. Codificación de Elementos del Circuito

Para facilitar la implementación del circuito resultante y reducir las discrepancias entre el circuito evolucionado y el implementado; el mecanismo de representación de circuitos incluye el concepto de valores comerciales propuesto por Horrocks y Khalifa [33]. Adicionalmente se emplean las series E6 y E12 de valores comerciales.

El tipo de elemento de circuito se codifica con un valor entero, por lo que este mecanismo puede integrar 232 diferentes tipos de dispositivos electrónicos (cantidad más que sobrada para este propósito). Y el valor se configura en el formato de notación científica siguiente:

$$cvalor * 10^{cdécada}, \tag{2}$$

donde *cvalor* es el valor codificado en el campo valor del gen de acuerdo con el tipo de elemento de circuito y *cdédaca* es el exponente codificado en el campo década del gen de acuerdo con el tipo de elemento de circuito.

Debido al hecho de que tanto valor como década están codificados por un entero, también éstos disponen de 232 posibles valores; que comparados con los 6 o 12 de las series E6 y E12 respectivamente (vea la tabla 1), resultan fácilmente codificables; incluso si se desea usar una serie de valores comerciales más extensa, como es la E24 o la E48, esto no implica ninguna modificación a la representación.

Tabla 1. Series de valores comerciales.

Valor	E12	E6	E3
0	10	10	10
1	12	15	22
2	15	22	47
3	18	33	
4	22	47	
5	27	68	
6	33		
7	39		
8	47		
9	56		
10	68		
11	82		

Tabla 2. Codificación propuesta para redes eléctricas pasivas.

Tipo	Década	Valor
C (0)	$10^{-5} – 10^{-9}$ (0 – 4)	E6 (0 – 5)
R (1)	$10^{1} – 10^{6}$ (0 – 5)	E12 (0 – 11)
L (2)	$10^{-1} – 10^{-6}$ (0 – 5)	E12 (0 – 11)

En relación con la década, este es un multiplicador que representa los prefijos del sistema internacional de unidades y que tomará valores convenientes según el tipo de elemento de circuito que se está codificando.

La tabla 1 muestra los valores normalizados de componentes electrónicos de acuerdo con las series E12, E6 y E3. De acuerdo con la serie de valores comerciales que se esté utilizando, el valor del dispositivo corresponderá al valor de la serie; por ejemplo, si el campo valor tiene un 2 y se está utilizando la serie E12, significa que el valor codificado que se debe tomar es 15; mientras que, si se trata de la serie E6, un valor de 2 corresponde a 22.

El rango de valores que un elemento puede tomar depende del tipo de elemento. Por ejemplo, los valores típicos prácticos de las resistencias oscilan entre los KΩ y los MΩ; mientras que los valores típicos de los capacitores están por debajo de los µF y las inductancias, normalmente exhiben valores en el orden de los mH. Teniendo estas limitaciones prácticas presentes; un rango de valores que pueden tomar los últimos tres campos del cromosoma se muestra en la Tabla 2. Aunque cabe mencionar que no existe ningún impedimento para expandir o restringir este rango de acuerdo con el tipo de circuito que se pretende diseñar.

Así, para representar una resistencia de 100KΩ mediante la serie E12, deberemos usar Tipo =1, Década =3 y Valor =0; es decir 10x104 que equivale a 100000 ohms.

En relación con la codificación de la topología de la red eléctrica, ésta se encuentra en los primeros campos del cromosoma: Nodo 1 corresponde al nodo al que se conecta la terminal 1 del elemento de circuito y Nodo 2 corresponde al nodo de la red al que se conecta la terminal 2 del dispositivo. Nodo Actual es un apuntador hacia el nodo que deberá ser conectado el siguiente elemento de la red. El propósito de este apuntador es cuidar que la red eléctrica no quede abierta en alguna parte, impidiendo el flujo de corriente y produciendo como consecuencia, un circuito no factible.

4.3. Ejemplos de codificación

A fin de integrar todos los elementos discutidos en la sección anterior, se utilizará el mecanismo de representación propuesto en la codificación de diferentes redes eléctricas; entre las que se encuentra una red diseñada usando la aproximación de Butterworth y otra usando la aproximación de Chebyshev.

La figura 5 muestra un circuito pasivo arbitrario construido en la plantilla discutida en la sección 4 y la figura 6, su correspondiente codificación mediante el mecanismo de representación propuesto. En esta figura se aprecia que el circuito construido entre los nodos inicial (1), final (F) y tierra (0), se compone de 5 elementos de circuito (tres inductancias, un capacitor y una resistencia). Siguiendo el cromosoma que lo representa, entre los nodos 1 y 0, se encuentra un elemento tipo = 0 (capacitor), cuyo valor = 3 (E6:33) y década = 3 (-8). Es decir, es un capacitor de 33x10-8 F o 330nF. Enseguida, entre los nodos 1 y 2, se encuentra un elemento tipo =2 (inductancia), con valor = 5 (E12:27) y década = 2 (-3); lo que codifica a una inductancia de 27mH y así sucesivamente. De manera que independientemente de lo complicado de la topología de una red, la codificación depende exclusivamente del número de elementos que constituyen al circuito.

El mecanismo de representación presentado es capaz de construir desde topologías clásicas hasta aquellas que retan la intuición del diseñador humano.

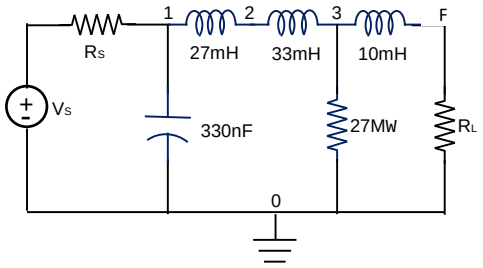


Fig. 5. Circuito eléctrico.

1,0,1,0,3,3	1,2,2,2,2,5	2,3,3,2,2,6	3,0,3,1,5,5	3,F,F,2,2,0

Fig. 6. Representación del circuito de la figura 5.

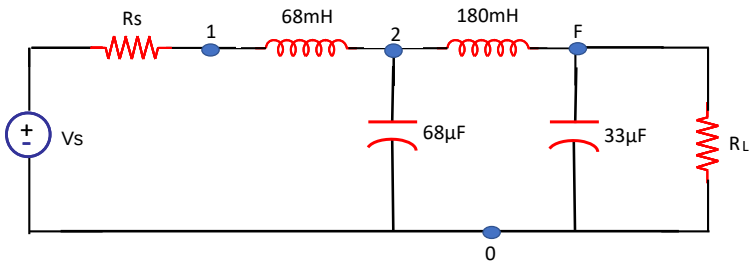


Fig. 7. Filtro pasa-bajas Butterworth de cuarto orden.

1,2,2,2,2,10	2,0,2,0,1,5	2,F,F,2,1,3	F,0,F,0,1,3

Fig. 8. Cromosoma del filtro de Butterworth de cuarto orden.

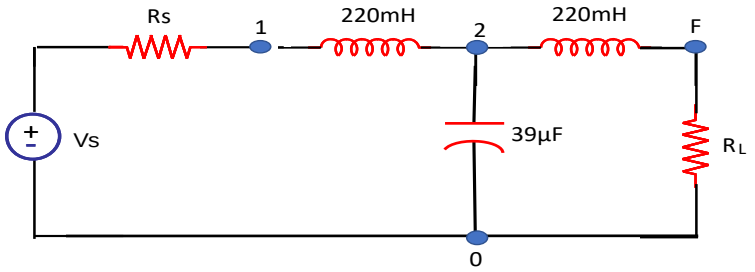


Fig. 9. Filtro pasa-bajas Chebyshev de tercer orden

1,2,2,2,1,4	2,0,2,0,1,4	2,F,F,2,1,4

Fig. 10. Cromosoma del filtro Chebyshev de tercer orden

La figura 7, muestra un filtro pasabajas diseñado mediante la aproximación de Butterworth. Las especificaciones de este filtro son que atenúe 60dB las señales por encima de los 750Hz, que tenga un rizado en la banda de paso de 1dB y que tanto la resistencia de carga como la del generador sean de 50Ω.

El cromosoma que representa a este circuito se muestra en la figura 8. Note que una vez que se ha establecido la plantilla en la que el circuito evolucionará, basta con codificar cada elemento de circuito.

Finalmente, en la figura 9 se presenta el filtro descrito anteriormente, diseñado mediante el uso de la aproximación de Chebyshev.

El cromosoma que representa este circuito se muestra en la figura 10.

Como se aprecia de los ejemplos anteriores, el mecanismo de representación propuesto es compacto y versátil, pues permite una representación compacta independientemente de la topología del circuito.

5. Conclusiones

Debido a que uno de los principales retos en el diseño automático de circuitos analógicos es su codificación, en este trabajo se propone un mecanismo de representación que fue creado teniendo en cuenta que debe ser capaz de evolucionar tanto la topología como el tamaño de los elementos de circuito. El mecanismo además es muy compacto, pues independientemente de la complejidad de la topología del circuito, codifica un solo gen por cada elemento de circuito. Esta situación produce una aportación interesante en relación con otras propuestas, pues, aunque no se discute en este documento, los operadores de transformación explorados producen topologías válidas, lo que reduce enormemente la carga computacional que otros esquemas como el de Koza [20], deben afrontar una vez que el mecanismo se encuentra en operación dentro de una herramienta de computación evolutiva.

El cromosoma es de tamaño variable y se implementa computacionalmente, usando listas de nodos (genes) enlazados, lo que de acuerdo con diferentes investigadores incluidos los autores de este trabajo, explota la potencialidad de búsqueda de los algoritmos evolutivos. En este trabajo se muestra que el mecanismo de representación codifica fácilmente topologías clásicas (como las que se producen siguiendo aproximaciones clásicas de diseño de circuitos). Además, se adapta a la plantilla de circuito que resulte conveniente, según el tipo de dispositivo electrónico que se desee diseñar.

Un aspecto que aún no se discute en este trabajo, pero que ya se ha explorado, es la facilidad de conversión de circuitos codificados a sus equivalentes en lenguaje de descripción de circuitos, para su simulación.

Otra ventaja de este mecanismo es que es adaptable para la manipulación de un numeroso grupo de familias de dispositivos electrónicos, lo que reduce las discrepancias entre un circuito evolucionado y uno implementado. En relación con el manejo de diferentes escalas de dispositivos, nuestra propuesta es muy flexible.

Finalmente, en este trabajo se muestra la versión usada para la representación de elementos analógicos pasivos, pero ya se está analizando la versión que integra elementos activos.

En este trabajo aún no se discuten aspectos de ejecución de algoritmos evolutivos, pues su propósito es la presentación del mecanismo de representación y prueba con diferentes circuitos; sin embargo, el paso siguiente es su aplicación en el entorno de un algoritmo genético.

Referencias

1. Castejón, F., Carmona, E.J.: Automatic design of analog electronic circuits using grammatical evolution. Appl. Soft Comput., 62, pp. 1003–1018 (2018)
2. Rojec, Ž., Bűrmen, Á., Fajfar, I.: Analog circuit topology synthesis by means of evolutionary computation. Eng. Appl. Artif. Intell., 80, pp. 48–65 (2019)

3. Coello-Coello, C.A., Aguirre, A.H.: Design of combinational logic circuits through an evolutionary multiobjective optimization approach. Artif. Intell. Eng. Des. Anal. Manuf., 16(1), pp. 39–53 (2002)
4. Karpuzcu, U. R.: Automatic verilog code generation through grammatical evolution. In: Proceedings of the 7th Annual Workshop on Genetic and Evolutionary Computation, pp. 394–397 (2005)
5. Yan, X., Li, W., Zhang, Y., Zhang, H., Wu, J. : Electronic circuit automatic design based on genetic algorithms. Procedia Eng., 15, pp. 2948–2954 (2011)
6. Anjomshoa, M., Mahani, A., Sadeghifard, S.: A new automated design and optimization method of CMOS logic circuits based on Modified Imperialistic Competitive Algorithm. Appl. Soft Comput., 21, pp. 423–432 (2014)
7. Talbi, E.G.: Metaheuristics: from design to implementation. El-Ghazali Talbi (2009)
8. Zebulum, R.S., Pacheco, M.A., Vellasco, M.: Comparison of different evolutionary methodologies applied to electronic filter design. In: IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), pp. 434–439 (1998)
9. Das, A., Vemuri, R.: A graph grammar based approach to automated multi-objective analog circuit design. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 700–705 (2009)
10. El Dor, A., Fakhfakh, M., Siarry, P.: Multiobjective differential evolution algorithm using crowding distance for the optimal design of analog circuits (2016)
11. Sanabria-Borbon, A.C., Tlelo-Cuautle, E.: Sizing analog integrated circuits by combining gm/ID technique and evolutionary algorithms. In: IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS), Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on, pp. 234–237 (2014)
12. Olenšek, J., Tuma, T., Puhan, J., Brmen, Á.: A new asynchronous parallel global optimization method based on simulated annealing and differential evolution. Appl. Soft Comput., 11(1), pp. 1481–1489 (2011)
13. Torres-Soto, A., Ponce de León-Sentí, E.E.: Metaheurísticas evolutivas en el diseño de circuitos analógicos (2010)
14. Shi, G.: Toward automated reasoning for analog IC design by symbolic computation–A survey. Integration, 60, pp. 117–131 (2018)
15. Sorkhabi, S.E., Zhang, L.: Automated topology synthesis of analog and RF integrated circuits: A survey. Integr. VLSI J., 56, pp. 128–138 (2017)
16. Das, A.: Algorithms for topology synthesis of analog circuits. University of Cincinnati, USA (2008)
17. Engelbrecht, A.P.: Fundamentals of computational swarm intelligence. Wiley (2005)
18. Holland, J.H., Holland, P.P.E.E.C.S.J.H., Holland, S.L.H.R.M.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology. Control, and Artificial Intelligence. M.I.T.P. (1992)
19. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company (1989)
20. Koza, J.R., Rice, J.P.: Genetic programming: on the programming of computers by means of natural selection. Bradford (1992)
21. Fogel, L.J.: On the organization of intellect. University of California, Engineering (1964)
22. Larrañaga, P., Lozano, J.A.: Estimation of distribution algorithms: a new tool for evolutionary computation. Springer US (2012)
23. Schwefel, H.P.: Evolutionsstrategie und numerische optimierung. Technische Universität Berlin (1975)
24. Feoktistov, V.: Differential evolution: in search of solutions. Springer US (2007)
25. Lohn, J., Colombano, S.: Automated analog circuit sythesis using a linear representation (1998)

26. Lohn. J.D., Colombano, S.P.: A circuit representation technique for automated circuit design. IEEE Trans. Evol. Comput., vol. 3, no. 3, pp. 205–219 (1999)
27. Goh, C., Li, Y.: GA automated design and synthesis of analog circuits with practical constraints. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), vol. 1, pp. 170–177 (2001)
28. Hu, J., Zhong, X., Goodman, E.D.: Open-ended robust design of analog filters using genetic programming. In: in Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 1619–1626 (2005)
29. Ambardar, A.: Procesamiento de señales analógicas y digitales. Thomson (2002)
30. Johnson, C. D.: Process Control Instrumentation Technology. Prentice Hall (1997)
31. Mattiussi, C., Floreano, D.: Analog Genetic Encoding for the Evolution of Circuits and Networks. IEEE Trans. Evol. Comput., vol. 11, no. 5, pp. 596–607 (2007)
32. Zebulum, R., Pacheco, M., Vellasco, M.: Variable Length Representation in Evolutionary Electronics. Evol. Comput., vol. 8, pp. 93–120 (2000)
33. Horrocks, D.H., Khalifa, Y.M.A.: Genetically derived filter circuits using preferred value components. In: IEE Colloquium on Analogue Signal Processing, pp. 1–4 (1994)

ISSN 1870-4069

Modelos de aprendizaje automático para clasificar patologías oculares en imágenes de fondo de ojo como prueba de tamizaje

Miguel Ángel Méndez Lucero[1], E. Mahuina Campos Castolo[2], Dania N. Lima Sánchez[2], Josué Fabricio Urbina González[3], Orlando Cerón Solís[2], Alejandro Alayola-Sansores[2], German Fajardo Dolci[4]

[1] University of Edinburgh, School of Informatics, Artificial Intelligence and its Applications Institute, Reino Unido
[2] Universidad Nacional Autónoma de México, Facultad de Medicina, Departamento de Informática Biomédica, México
[3] Oracle México Development Center, México
[4] Universidad Nacional Autónoma de México, Director de la Facultad de Medicina, México

{Mendezluceromiguelangel, orlandoceronsolis, ale.alayola, urbgon}@gmail.com, {dibfm, ,direccionfm}@unam.mx, danianimbe@hotmail,com

Resumen. Ante la posibilidad de múltiples patologías en cada fotografía del fondo de ojo, escasez de datos para enfermedades poco comunes; y la falta de bases de datos públicas bien clasificadas para enfermedades oculares, son algunos de los principales problemas que obstaculizan la creación de clasificadores. Estos modelos podrían ayudar a los oftalmólogos en el diagnóstico de enfermedades oculares. Utilizando técnicas de transferencia de aprendizaje y aumento de datos, analizamos la eficiencia de diversos algoritmos de varias clases y métodos de conjunto. Después de comparar el rendimiento de aproximadamente 96 modelos multicategoríales, propusimos un modelo probado con 1350 imágenes; con un conjunto de datos no balanceado de 10 clases.

Palabras clave: Aprendizaje de transferencia, clasificador de enfermedades oculares, métodos de conjunto, aprendizaje profundo.

Machine Learning Models to Classify Eye Pathologies in Fundus Images as Screening Test

Abstract. The possibility of multiple pathologies in each fundus photography, scarcity of data for uncommon diseases and lack of properly classified public databases for ocular diseases are some of the main problems that hinder the

creation of classifiers. These models could assist ophthalmologists in diagnostics of eye diseases. Using transfer learning and data augmentation techniques, we analyze the efficiency of diverse multi-class algorithms and ensemble methods. After comparing the performance of approximately 96 different multi-categorical models, we proposed a model tested with 1350 images; which is an unbalanced data set of 10 classes.

Keywords: Transfer learning, eye disease classifier, ensemble methods, deep learning.

1. Introducción

En México, las discapacidades de todo tipo afectaron al 6% de la población en 2014, de las cuales el 58.7% se asociaron con discapacidad visual. Las primeras cuatro causas de discapacidad visual en México son: 1. Cataratas, 2. Retinopatía diabética, 3. Glaucoma y 4. Degeneración macular relacionada con la edad. Estas cuatro enfermedades son responsables de hasta 575,954 casos de ceguera por año y 4, 917,340 casos de otras patologías, con un costo estimado en 2013 de $ 539, 000,000 dólares estadounidenses por año [1]. Se considera que la mayor parte de las complicaciones que puede llevar a la perdida irremediable de la visión son prevenibles con diagnóstico precoz, por lo que es imprescindible tener herramientas de apoyo el diagnóstico para una detección temprana. La detección de patología por fotografía digital es ampliamente aceptada para el diagnóstico de la retina. Sin embargo, un diagnóstico definitivo requiere una evaluación completa por parte de un oftalmólogo.

En los últimos tiempos, las técnicas de aprendizaje automático se han utilizado para resolver problemas de reconocimiento y clasificación de patrones en diferentes áreas. El desarrollo de algoritmos de inteligencia artificial (IA) para resolver problemas médicos ha sido una de las áreas más prometedoras, especialmente con el fin de realizar tamizaje masivo de los pacientes, el aprendizaje automático es una rama importante de la IA, que puede dividirse en supervisado o no supervisado, cuando están las imágenes etiquetadas previamente [2]. Las técnicas de clasificación que se utilizaron son las siguientes: Clasificación mediante modelos probabilísticos: Naïve Bayes, Maquinas de Soporte vectorial (SVM por sus siglas en ingles), Bosque Aleatorio (Random Forest Regression), Regresión Logística, Naïve Bayes Multinomial.

Estos modelos son ideales cuando el número de imágenes son pequeños, para la otra alternativa, el aprendizaje profundo, se requiere regularmente cientos de miles de datos [2]. La eficacia del aprendizaje automático toma en cuenta el rendimiento del modelo, para ello se utiliza una matriz de confusión, donde la imagen propuesta será clasificada y comparada con la clasificación real, obteniendo los valores de sensibilidad y especificidad. La exactitud evalúa el entrenamiento del modelo cuando existe un amplio número de muestras, cuando existe un desbalance en la muestra de categorías es más confiable utilizar la precisión, que evalúa la calidad del modelo en la tarea de clasificación, y la métrica de exhaustividad sobre la cantidad que el modelo es capaz de identificar, es decir que porcentaje es capaz de seleccionar adecuadamente.

Para evaluar de una manera más integral este proceso, se utiliza el valor F, un valor igual a 1 considera que el modelo es perfecto y 0 implica que el modelo no se ajusta a la realidad.

Además se utiliza la curva característica operativa del receptor (COR o ROC en sus siglas en ingles). Los métodos de ensamblaje implica la combinación de múltiples modelos clasificadores, produciendo un modelo clasificador final, mientras que el aprendizaje por transferencia es un método para adaptar un modelo entrenado en un dominio a otro dominio [3].

2. Trabajos relacionados

En el campo de la medicina, es necesario tener precauciones en relación a la privacidad del paciente, lo que ocasiona que en ocasiones el acceso a grandes bases de datos públicas de patologías médicas se encuentre limitadas, esto dificulta la aplicación de aprendizaje profundo, por lo cual diversos autores han trabajado con diferentes técnicas para aumentar los datos.

En un estudio realizado por Asperti y Mastronardo, utilizaron 4000 imágenes endoscópicas de enfermedades gastrointestinales etiquetadas con médicos endoscopistas, con diferentes patologías, mostrando que la aplicación de diferentes técnicas, permitía una mejora en la clasificación [4].

En relación a las imágenes obtenidas de fondo de ojo, se han hecho múltiples investigaciones, debido a que las imagines digitales proporcionan valiosa información, no sólo con respecto a patologías, sino también para datos morfológicos que pueden analizarse de manera rápida y no invasiva utilizando inteligencia artificial, en una amplia revisión realizada por Schmidt-Erfurth y cols, con respecto a la aplicación de la IA en patologías de retina; encontraron que la mayor parte de los estudios automatizados han sido aplicados a retinopatía diabética, utilizando aprendizaje profundo para la detección, clasificación diagnóstica y orientación terapéutica en esta patología, que podría ofrecer nuevos tratamientos, y una medicina personalizada, sin embargo la mayor parte de estos avances se basan en una sola patología [5]. Desafortunadamente, la mayor parte de las patologías de fondo de ojo presentar comorbilidades, y las causas de discapacidad implican múltiples diagnósticos.

Abordando este problema, Choi y cols, utiliza el aprendizaje profundo para analizar imágenes de retina, utilizando una red neuronal convolucional mediante el uso de MatConvNet, para la detección de múltiples enfermedades de la retina, utilizando una base de datos de retina (STARE). Utilizaron 10 categorías, incluyendo retina normal y nueve enfermedades con 25 imágenes por cada grupo, obteniendo el mejor rendimiento mediante el uso de aprendizaje basado en bosques aleatorios, utilizando el modelo VGG-19. Encontraron que al aumentar el número de categorías, disminuía el rendimiento del modelo, obteniendo una precisión de 30.5% cuando incluyeron 10 categorías, una kappa de Cohen de 0.244, al contrario, cuando sólo compararon tres categorías obtuvieron una precisión de 72.8 y un valor kappa de 0.577.

El bajo rendimiento lo atribuyeron al pequeño tamaño de los conjuntos de datos, pero encontraron que al utilizar Transfer Learning pudieron mejorar el rendimiento, por lo que sugieren realizar otros estudios para confirmar la efectividad de los algoritmos [6]. Tomando estos antecedentes, decidimos evaluar un modelo con la precisión y sensibilidad suficientes para los estándares medicos, a fin de utilizarlo como prueba de detección y clasificación de enfermedades oculares.

3. Material y métodos

3.1. Obtención de datos

El conjunto de datos utilizado en este artículo fue obtenido de bases abiertas de imágenes, fotografías tomadas de un Centro de salud, y del Instituto Mexicano de Oftalmología I.A.P, aprobado por el comité de ética e investigación de esta misma institución. Se trabajó con un conjunto de 1352 imágenes, clasificados en 10 categorías verificado por un médico especialista en retina, dentro de paréntesis se menciona el número de imágenes: adelgazamiento de la retina (n=36), cataratas (n=10), drusas (n=25) ,excavación del nervio óptico/glaucoma (n=103), degeneración macular (n=15), retinopatía diabética leve (n=380), retinopatía diabética moderada (n=45), retinopatía diabética severa (n=20) , retinopatía hipertensiva (n=68) y normal (n=650). Con este banco de imágenes se hicieron cinco propuestas de clases:

1. Adelgazamiento de retina, cataratas, drusas, excavación del nervio óptico/glaucoma, degeneración macular, retinopatía diabética general, retinopatía hipertensiva y normal. (Total 8 clases).
2. Adelgazamiento de la retina, cataratas, drusas, excavación del nervio óptico/glaucoma, degeneración macular, retinopatía diabética severa, retinopatía hipertensiva y normal. (Total 8 clases).
3. Adelgazamiento de la retina, cataratas, drusas, excavación del nervio óptico/glaucoma, degeneración macular, retinopatía, retinopatía diabética moderada, retinopatía diabética severa, retinopatía hipertensiva y normal (total 10 clases).
4. Adelgazamiento de la retina, excavación del nervio óptico/glaucoma, degeneración macular, retinopatía diabética, retinopatía hipertensiva y normal (total de 6 clases).
5. Imagen de ojo normal y enfermo, conformado con las otras nueve patologías. (Total 2 clases).

En la primera subdivisión consideramos todos los diferentes grados de retinopatía diabética (retinopatía diabética leve, moderada y severa) en una clase, esta clasificación se consideró ya que formaban diferentes grados de severidad de una misma patología; por otro lado, para la segunda subdivisión consideramos sólo la retinopatía diabética severa, dado que tenía diferencias importantes en sus alteraciones, con el resto de las imágenes. Para la cuarta subdivisión incluimos la categoría de Drusas como parte de degeneración macular, ya que comparten la misma presentación clínica y excluimos cataratas debido a la falta de datos.

3.2. Aumento de datos

Se ha demostrado que el uso de técnicas de aumento de datos en conjuntos que tienen una muestra pequeña con clases desequilibradas, mejora significativamente los puntajes de precisión, exhaustividad y el valor Fde los modelos entrenados [4].

Para el aumento de datos utilizamos los siguientes parámetros: rotation_range=180°, width_shift_range=12%, height_shift_range=12%, zoom_range=12%, shear_range= 20%, horizontal_flip=True, fill_mode='nearest'.

Además utilizamos dos enfoques. El primero fue aplicar el aumento de datos para igualar las demás clases a la clase con el mayor número de datos (la clase de ojo normal, que contenía 650 imágenes), con lo cual al final del proceso teníamos 650 imágenes para cada clase. El segundo enfoque fue reducir el número de imágenes para cada clase, de modo que estuvieran equilibradas. En este caso, redujimos el número de imágenes por clase a 32. Luego, el proceso de aumento de datos genero nueve veces más imágenes, dándonos un total de 320 imágenes por clase. Finalmente, analizamos el rendimiento, entrenando varios modelos usando estas subdivisiones de clases.

3.3. Transfer learning con aprendizaje profundo

Las técnicas de aprendizaje profundo evolucionan constantemente debido a su gran uso en las tareas de clasificación [7], además las redes neuronales convolucionales han demostrado sobresalir en tareas que involucran la clasificación de imágenes [8]. En el caso de nuestro artículo, las técnicas de transferencia de aprendizaje se usan como un extractor de funciones [9]. Debido a la falta de datos en nuestra población problema y la gran diferencia entre las imágenes utilizadas para entrenar la CNN contra las contenidas en nuestro conjunto de datos. Utilizamos la CNN pre entrenada para extraer características relevantes de cada imagen [10] y luego alimentarlas a un algoritmo más simple de múltiples clases.

Si bien hay muchos CNN pre entrenado que se pueden usar para la extracción de características, en este enfoque utilizamos el modelo VGG16, debido a su estructura relativamente simple (16 capas) y al buen rendimiento. Este modelo ganó el ImageNet ILSVRC-2014, también vale la pena mencionar que debido a la simplicidad de su arquitectura, el proceso de la función de extracción tarda 55 segundos para aproximadamente 7000 imágenes.

3.4. Algoritmos de clases múltiples y métodos de ensamble

Para encontrar el modelo que mejor se ajustara y clasificara nuestros datos pre procesados, usamos e implementamos varios algoritmos de clases múltiples. Primero, sin ningún método de ensamblaje, implementamos algoritmos simples de varias clases, tales como: Máquina de Soporte Vectorial (SVM), Bosques aleatorios (con una profundidad de 8 niveles), Regresión logística, Bernoulli Naïve Bayes y Multinomial Naïve Bayes. Además, implementamos varios métodos de ensamble [11], como Bagging Classifier utilizando los algoritmos mencionados anteriormente como estimadores base y bosque aleatorio. Éste clasificador es un meta estimador de ensamblaje que combina varios algoritmos de aprendizaje automático y los ajusta con subconjuntos aleatorios del conjunto de datos de entrenamiento. Al igual que Bosques aleatorios, los clasificadores de ensamblaje funcionan como métodos de promedio. Esto significa que usamos cada uno de los modelos ajustados y promediamos sus predicciones que resultan en un estimador combinado que generalmente es mejor que cualquiera de los estimadores de base debido a que su varianza se reduce.

3.5. Prueba de conjunto de datos y métricas para evaluar el rendimiento

Después de entrenar nuestros modelos, el departamento de informática biomédica recopiló un conjunto de datos de prueba que consta de un conjunto de 121 dividido en las 10 clases iniciales para evaluar el rendimiento de cada modelo entrenado. En los datos utilizados para la prueba, se detectó un pequeño sesgo debido a un diseño de muestra no óptimo causado por la disponibilidad de datos y tiempo; que se ajustó para poder reproducir de manera más confiable el rendimiento real de cada modelo.

Para evaluar el rendimiento de cada modelo entrenado, utilizamos diferentes medidas para la clasificación [12].

Puntuación de precisión: esta función calcula la precisión de las predicciones correctas:

$$\frac{\sum_{i=1}^{c} TP_i}{n_{muestra}}. \tag{1}$$

Simbología: la variable 'c' representa el número de clases, tasa de verdaderos positivos (TP), tasa de verdaderos negativos (TN), tasa de falsos positivos (FP), tasa de falsos negativos (FN); El hiperparámetro ß es igual a uno, ya que estamos calculando la puntuación de F1.

Exhaustividad: efectividad promedio por clase de un clasificador para identificar etiquetas de clase:

$$\frac{\sum_{i=1}^{c} \frac{TP_i}{TP_i + FP_i}}{c}. \tag{2}$$

Precisión: un acuerdo promedio por clase de las etiquetas de clase de datos con las de un clasificador:

$$\frac{\sum_{i=1}^{c} \frac{TP_i}{TP_i + FN_i}}{c}. \tag{3}$$

Puntuación F1: relación entre las etiquetas positivas de los datos y las dadas por un clasificador basado en un promedio por clase:

$$\frac{(\beta^2 + 1)Sensibilidad \times Precisión}{\beta^2 Sensibilidad + Precisión}. \tag{4}$$

ß es igual a uno para calcular la puntuación de F1.

4. Resultados

4.1. Enfoque de aumento de datos

Se utilizaron varios algoritmos de clases múltiples, utilizando dos enfoques diferentes de aumento de datos. Para medir el rendimiento de estos enfoques, utilizamos las primeras tres subdivisiones de clase; usando la puntuación de precisión para medir ambos enfoques.

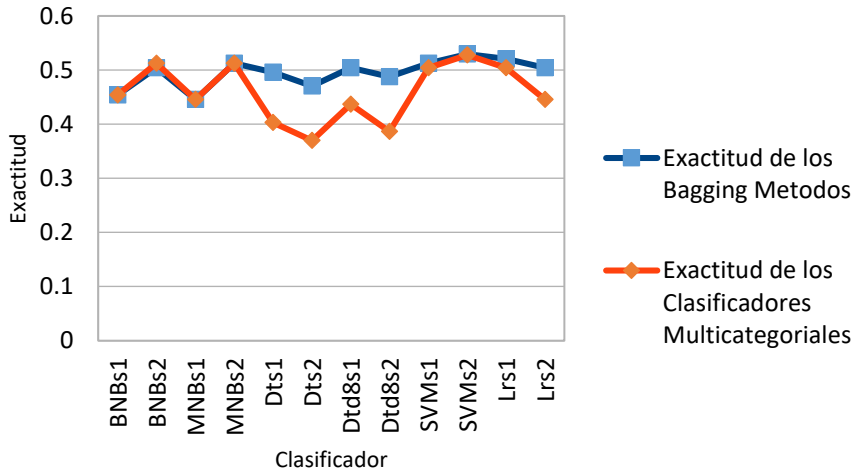


Fig. 1. Método Bagging Vs. Clasificadores Multicategoríales.

Observamos que el primer enfoque sobresalió sobre el segundo en casi todos los clasificadores en un rango entre 5% y 25%, con la excepción de Multinomial y Bernoulli Naïve Bayes, en el que el segundo enfoque sobresalió sobre el primero con 5% más de precisión.

4.2. Algoritmos de clases múltiples frente a métodos Bagging

Usando las dos primeras subdivisiones, medimos la eficiencia que los métodos de Ensemble realizaron en algoritmos normales de clases múltiples Figura 1. Los métodos de embolsado mejoran la puntuación de precisión en un rango entre 2% -8%, dependiendo del algoritmo de clasificación.

Vale la pena mencionar que aunque el número de estimadores utilizados para el entrenamiento fue bajo (10 estimadores de línea de base), un análisis posterior mostró que con 100 estimadores de base en algunos modelos, la precisión mejoró en un 2% (el caso de regresores lineales y decisión árboles), mientras que los demás mantuvieron su precisión sin cambios.

Este gráfico representa el rendimiento de cada modelo de múltiples categorías en comparación con su implementación como un estimador de referencia en el método de ensacado. Los modelos de bagging en este cuadro se construyeron utilizando estimadores de 10 bases y permitiendo un arranque.

Los clasificadores multicategoría utilizados en este cuadro son: Máquina de vectores de soporte (SVM), Árbol de decisiones (Dt y Dtd8, donde d8 representa la profundidad máxima utilizada en el árbol de decisión), Regresión logística (LR), Bayes ingenuos de Bernoulli (BNB), Bayes ingenuos multinomiales (MNB). El subfix s1 y s2 representan la subdivisión de la categoría utilizada para el entrenamiento.

Tabla 1. Resultados de rendimiento de modelos de aprendizaje profundo de múltiples categorías para cada subdivisión de clase.

Numero de Categorías	Exactitud	Sensibilidad	Precisión	F
10 Clases	0.4308	0.4332	0.2012	0.1773
8 Clases	0.5294	0.5629	0.2503	0.2383
6 Clases	0.5855	0.6003	0.3380	0.3303
2 Clases	0.8884	0.8973	0.8915	0.8943

4.3. Conjuntos de subdivisión de clase y análisis de estimador base

Como se esperaba, el rendimiento de cada subdivisión aumentó en proporción al número de clases, la Tabla 1 muestra el modelo con el mejor rendimiento en cada subdivisión. Encontramos un aumento general del rendimiento en la subdivisión 2 sobre la subdivisión 1 en un rango entre 4% y 16% (ambos midieron las mismas clases 6 y 8). Por lo tanto, la Tabla 1 solo se muestra los mejores resultados obtenidos de la subdivisión 2 en la clasificación de seis y ocho categorías.

Además, encontramos que los resultados sobresalieron en desempeño en comparación con investigaciones similares [6]; utilizando un conjunto de datos de entrenamiento más pequeño (después de aplicar técnicas de aumento de datos) y para probar, un conjunto de datos diferente para eliminar el sesgo y evaluar de manera más confiable el rendimiento de cada modelo.

En la Tabla 1 podemos observar los resultados de mejor rendimiento de modelos de aprendizaje profundo de múltiples categorías para cada subdivisión de clase. En todas las subdivisiones, la máquina de vectores de soporte de ensacado fue el modelo con los mejores puntajes de rendimiento. Los modelos con 8 clases fueron entrenados por las subdivisiones 1 y 2; Los resultados que se muestran aquí se obtuvieron del modelo entrenado por la segunda subdivisión. En la clasificación múltiple, el mejor rendimiento se obtuvo en la subdivisión de 6 clases, en la que dos modelos superaron a los demás.

El primero fue entrenado con Bayes Naïve multinomial como estimador base, obteniendo una sensibilidad del 61.38%, una precisión del 56.75% y una puntuación F1 del 35.85%, mientras que la segunda máquina de vectores de soporte obtuvo una sensibilidad o 61.38%, una precisión de 56.75% y un F1 - Puntaje de 35.85%. De estos resultados podemos concluir que los modelos probabilísticos son tan buenos estimadores como los modelos de regresión que normalmente se destacan cuando se combinan con bosques aleatorios en este tipo de tareas.

Para el problema de la clasificación múltiple como se muestra en la Figura 2, el mejor rendimiento se obtuvo en la subdivisión de 6 clases, en las que dos modelos superaron a los demás. El primero fue entre sensibilidad del 61,38%, una precisión del 56,75% y una puntuación F1 del 35,85%, mientras que la segunda máquina de vectores de soporte obtuvo una sensibilidad del 61,38%, una precisión del 56,75% y una puntuación F1 del 35,85%.

En la Figura 2 observamos los resultados de las métricas utilizadas para evaluar el rendimiento de varios estimadores básicos en un problema de clasificación de 6 clases.

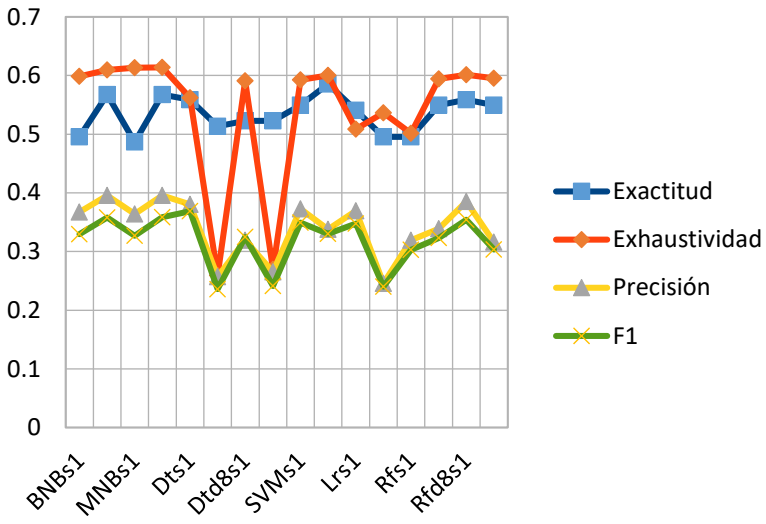


Fig. 1. Medidas de Rendimiento de los métodos Bagging de seis clases.

Los estimadores básicos utilizados en este gráfico son la máquina de vectores de soporte (SVM), el árbol de decisión (Dt y Dtd8 donde d8 representa la profundidad máxima utilizada en el árbol de decisión), la regresión logística (LR), las bayas ingenuas de Bernoulli (BNB), Bayes ingenuos multinomiales (MNB) y bosque aleatorio (RF). El subfijo s1 representa los modelos que usaron todos los niveles de retinopatía diabética para entrenar a la clase de retinopatía diabética, mientras que s2 representa los modelos que usaron solo la retinopatía diabética severa para entrenar a la clase de retinopatía diabética.

Podemos concluir con estos resultados que los modelos probabilísticos son tan buenos estimadores como los modelos de regresión, que normalmente se destacan junto con bosques aleatorios en este tipo de tareas. Para el algoritmo de prueba de detección se realizó una ligera modificación de los datos para medir el rendimiento. Para el conjunto de datos inicial tuvimos 650 que pertenecían a la clase normal y 702 en total para las patologías. Dividimos el conjunto de datos en 90% para capacitación y 10% para pruebas, luego aplicamos el aumento de datos para duplicar la cantidad de datos para capacitación.

Se utilizaron varios algoritmos para entrenar los modelos: SVM, Bayes Naïve de Bernoulli, Bayes Naives multinomiales (MBN), árboles de decisión y regresión logística. El algoritmo SVM fue mejor sobre el segundo en casi todas las subclasificaciones (entre 5% y 25%), con la excepción de los basados en Bayes, que pudieron clasificar un 5% más preciso la segunda propuesta de subclasificación de clases.

Usando las dos primeras subdivisiones, medimos la eficiencia de los métodos de conjunto en algoritmos normales de clases múltiples (Figura 1). Los métodos de ensamble mejoran el puntaje de precisión en un rango de 2% -8%, dependiendo del algoritmo de clasificación. Vale la pena mencionar que, a pesar de que el número de

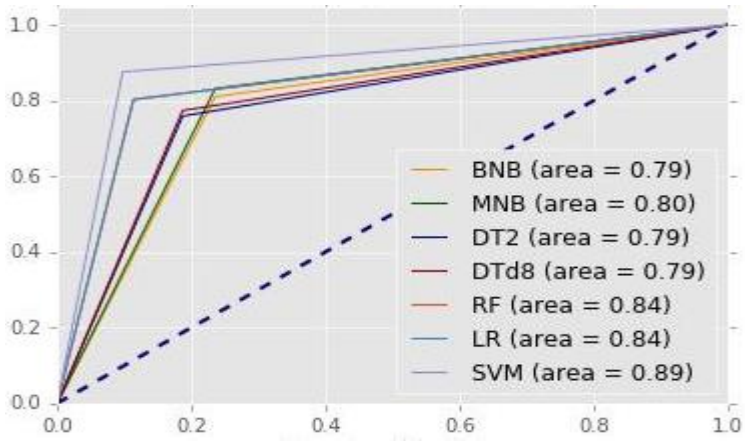


Fig. 3. Curvas características de funcionamiento del receptor de varios modelos entrenados.

estimadores utilizados para el entrenamiento es bajo (10 estimadores de referencia), un análisis posterior mostró que, con 100 estimadores básicos en algunos modelos, la precisión mejoró en un 2% (este fue el caso de las regresiones lineales y el árboles de decisión), mientras que los demás mantuvieron su precisión sin cambios.

Los modelos fueron entrenados utilizando las características extraídas de un conjunto de datos de imágenes de fondo con el VGG-16. El conjunto de datos se dividió en 90% para el entrenamiento y 10% para las pruebas. Las técnicas de aumento de datos se aplicaron al conjunto de datos de entrenamiento para mejorar la sensibilidad de los modelos. Para el algoritmo de prueba de detección, se realizó una ligera modificación de los datos para medir el rendimiento. Del conjunto de datos inicial, tuvimos 650 que pertenecían a la clase normal y 702 en total para las patologías. Dividimos el conjunto de datos en 90% para entrenamiento y 10% para pruebas, luego aplicamos el aumento de datos para duplicar la cantidad de datos para entrenamiento.

Los resultados del rendimiento se describen en la curva ROC que se muestra en la Figura 3. Una vez más se revela que SVM es el algoritmo que destaca sobre el resto de las técnicas de aprendizaje automático, mostrando una sensibilidad del 89,7%, seguido de RF y LR. Los resultados del rendimiento se describen en la curva ROC que se muestra en la Figura 3. Una vez más se revela que SVM es el algoritmo que destaca sobre el resto de las técnicas de aprendizaje automático, mostrando una sensibilidad del 89,7%, seguido de RF y LR.

En la Figura 3 se puede observar las Curvas ROC como prueba de detección para las enfermedades en imágenes de fondo de ojo.

Los modelos fueron entrenados utilizando las características extraídas de un conjunto de datos de imágenes de fondo con el VGG-16. El conjunto de datos se dividió el 90% para el entrenamiento y el 10% para las pruebas, se aplicaron técnicas de aumento de datos al conjunto de datos del entrenamiento para mejorar la sensibilidad de los modelos.

5. Conclusiones y trabajo a futuro

Se realizó un análisis en varias subdivisiones de clase para aumentar la puntuación de precisión de los modelos; cada subdivisión fue creada cuidadosamente para evaluar el desempeño de varias clases. Para la primera y segunda subdivisiones, analizamos el efecto de usar solo imágenes de retinopatía severa durante el entrenamiento contra la retinopatía general (todos los grados de retinopatía). Observamos que, con la excepción de los modelos entrenados con MNB, los resultados mostraron que cuando se utilizó únicamente retinopatía severa, tuvo mejor rendimiento. Confirmando que la razón principal es probablemente las características bien definidas mostradas en la retinopatía severa contra los otros niveles de retinopatía

El segundo análisis mostró que las cataratas, drusas y retinopatía hipertensiva son las patologías con menos precisión en el rendimiento general del clasificador. Las cataratas se excluyeron en la subdivisión 4 ya que los resultados mostraron que la falta significativa de datos está altamente relacionada con el bajo rendimiento de los modelos que predecían esta clase. Es probable que para la evaluación de cataratas sea necesario un modelado propio.

Otra de las dificultades en la clasificación fue en degeneración macular relacionada con la edad (DMAE), ya que se menciona que puede contener drusas [13]. Si bien las drusas no se considera una causa directa de degeneración macular, es uno de los síntomas visibles observados en las imágenes de fondo de ojo que contienen etapas iniciales e intermedias de DMAE. Debido a esto, la subdivisión cuatro incluyó datos de drusas como parte de la clase DMAE. Los resultados de la prueba mostraron una disminución en el rendimiento de los modelos que incluyen Drusas como parte de DMAE, en comparación con aquellos entrenados con solo DMAE y un probable factor confusor es que se incluyó todos los tipos de DMAE (es decir, degeneración macular seca y húmeda en todos sus grados). Con el fin de eliminar estos sesgos, será deseable que en estudios futuros se tome en cuenta estas subdivisiones de la enfermedad.

La investigación actual en inteligencia artificial aplicados a imágenes de retina, generalmente involucra grandes conjuntos de datos, con cientos de miles de imágenes para capacitación y validación, lo que facilita la obtención de mejores resultados debido a la cantidad de datos discriminatorios disponibles para la toma de decisiones [14], sin embargo, en nuestra población no resulta tan accesible tener este número de imágenes, por lo cual, aunque el rendimiento general del algoritmo de varias clases (es decir, el algoritmo clasificador de 6 clases) no es adecuado para ser considerado para un diagnóstico confiable de la enfermedad, tiene un mejor rendimiento comparado con un trabajo similar [6].

Como se mencionó anteriormente, la investigación se ha llevado a cabo en el campo de los algoritmos de aprendizaje automático utilizados en la tarea de clasificar patologías oculares, pero solo en problemas específicos, como la identificación de una patología específica [14]. La investigación generalmente involucra grandes conjuntos de datos, con cientos de miles de imágenes para capacitación y validación, lo que facilita la obtención de mejores resultados debido a la cantidad de datos discriminatorios disponibles para la toma de decisiones. Por esta razón, si bien el rendimiento general del algoritmo de varias clases (es decir, el algoritmo clasificador de 6 clases) no es adecuado para ser considerado para un diagnóstico confiable de la enfermedad, pero tenían un mejor rendimiento, en comparación con un trabajo similar

[6], a pesar de usar un conjunto de datos de entrenamiento más pequeño (después de aplicar técnicas de aumento de datos).

Esperábamos un valor de más del 95% de exactitud para considerarlo óptimo para uso médico, sin embargo el valor obtenido sigue siendo significativo en comparación con los resultados en otras publicaciones, donde los algoritmos tienen una sensibilidad del 80% para varias patologías [6] o, en el caso de la prueba de detección, 97% con el uso de cientos de miles de imágenes [14].

En el caso de los resultados del algoritmo de detección, obtuvimos valores prometedores al utilizar la detección de 10 patologías, ya que existen reportes para siete patologías. Se encontraron varias limitaciones y dificultades durante la realización de este trabajo; el principal fue la compilación del conjunto de datos, así como que las diferentes categorías de enfermedad, tendrían diferentes valores de muestra, además de que varios pacientes presentaban múltiples patologías, que es bastante común en el área médica. Este problema generó un sesgo en nuestro conjunto de datos que puede explicar los resultados al clasificar múltiples patologías.

Los resultados que se muestran tanto en el algoritmo de prueba de detección, como en el algoritmo de clasificación múltiple nos permiten desarrollar nuevos modelos con un conjunto de datos más amplio que podría ser utilizado por el personal de atención sanitaria, enseñanza y para realizar clasificación de otro tipo de imágenes.

Referencias

1. López-Star, E.M., Allison-Eckert, K., Limburg, H., Brea-Rodríguez, I., Lansingh, V. C.: Evaluación rápida de la ceguera evitable, incluida la retinopatía diabética, en Querétaro, México. Revista Mexicana de Oftalmología, 92, pp. 84–93 (2018)
2. Tong, T., Lu, W., Yu, Y., Shen, Y.: Application of machine learning in ophthalmic imaging modalities. Eye Vis (Lond) 16(22) (2020)
3. Lim, G., Bellemo, V., Xie, Y., Lee, X.Q., Yip, M.Y.T., Ting, D.S.W.: Different fundus imaging modalities and technical factors in AI screening for diabetic retinopathy: a review. Eye Vis (Lond), pp.14–21 (2020)
4. Asperti, A., Mastronardo, C.: The effectiveness of data augmentation for detection of gastrointestinal diseases from endoscopical images. In: S. Wiebe, H. Gamboa, A. Fred, & S. Bermúdez i Badia (Eds.), Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies, 2: KALSIMIS. Madeira Portugal, pp. 199–205 (2018)
5. Schmidt-Erfurth, U., Sadeghipour, A., Gerendas, B.S., Waldstein, S.M., Bogunović, H.: Artificial intelligence in retina. Progress in Retinal and Eye Research, 67, pp. 1–29 (2018)
6. Choi, J.Y., Yoo, T.K., Seo, J.G., Kwak, J., Um, T.T., Rim, T.H.: Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database. PloS ONE, 12, pp. 1–16 (2017)
7. Torrey, L., Shavlik, J.: Transfer learning. In: E. Soria, J. Martin, R. Magdalena, M. Martinez & A. Serrano, editor, Handbook of Research on Machine Learning Applications. IGI Global. pp 1–22 (2009)
8. Waseem, R.: Deep convolutional neural networks for image classification: a comprehensive review. 2449, pp. 2352–2449 (2017)
9. Garcia-Gasulla, D., Vilalta, A., Ayguad, E., Cort, U.: On the behaviour of convolutional nets for feature extraction, pp. 1–29 (2018)
10. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In Advances in neural information processing systems, pp. 3320–3328 (2014)

11. Dietterich, T.: Ensemble methods in machine learning. in international workshop on multiple classifier systems, pp. 1–15 (2000)
12. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing and Management, 45, pp. 427–437 (2009)
13. Bowes-Rickman, C., Farsiu, S., Toth, C.A., Klingeborn, M.: Dry age-related macular degeneration: mechanisms, therapeutic targets, and imaging. Investigative Ophthalmology and Visual Science, 54, pp. 68–80 (2013)
14. Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Webster, D. R.: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22) pp. 2402–2410 (2016)

ISSN 1870-4069

Chalecos amarillos CHA: un análisis a través de diccionarios en Python y teoría de medios

María Beatríz Bernabé Loranca[1], Alberto José Luís Carrillo Canán[2], Marleni Reyes Monreal[3]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
[2] Benemérita Universidad Autónoma de Puebla, Facultad de Filosofía y Letras,
[3] Benemérita Universidad Autónoma de Puebla, Escuela de Artes Plásticas y Audiovisuales, México

{maria.bernabe,alberto.carrillo, marleni.reyes}@correo.buap.mx

Resumen. En este trabajo se presenta un análisis del movimiento social francés de los chalecos amarillos (CHA) en tres partes. La primera parte se centra en observar e identificar la importancia de los contenidos *web*, los *influencers* y los usuarios. A partir de la observación del descontento (*colère*), se identificaron palabras representativas del problema y se almacenaron en un arreglo (*arreglo 1*). La segunda parte, toma la información generada en la primera para realizar un análisis de los contenidos de la red social *Twitter* a través del *procesamiento de lenguaje natural* (PLN) con el fin de identificar nuevos adjetivos o subrayar conceptos ya observados en *arreglo 1* para la creación de un diccionario. La tercera y última parte construye un *arreglo 2*, el cual contiene palabras resultantes de una búsqueda enumerativa de los adjetivos en el sitio *web* de noticias *FranceInfo* [1]. Se produjo un mapeo gráfico sobre el número de veces y correlaciones que una palabra se relaciona con otras palabras que describen a CHA. Para este propósito, el contenido propuesto en el *arreglo 1* y el contenido del *arreglo 2* se consideraron en su totalidad. Tanto los diccionarios PLN como los dos arreglos, coinciden en un alto porcentaje con las palabras iniciales, lo cual implica que estos tres procedimientos independientes permiten comparar los resultados e interpretar el enojo del público en el movimiento de los chalecos amarillos y sus posibles implicaciones.

Palabras clave: Diccionarios, PLN, twitter, colère, chalecos amarillos.

Yellow Vests (YV): An Analysis through Dictionaries in Python and Media Theory

Abstract. This paper presents an analysis of the French social movement of yellow vests (YV) in three parts. The first part focuses on observing and

identifying the importance of web content, influencers and users. From the observation of discontent (colère), words representative of the problem were identified and stored in an array (array 1). The second part takes the information generated in the first to carry out an analysis of the contents of the social network Twitter through natural language processing (PLN) in order to identify new adjectives or underline concepts already observed in array 1 for creating a dictionary. The third and final part builds array 2, which contains words resulting from an enumerative search for adjectives on the FranceInfo news website [1]. A graphical mapping was produced on the number of times and correlations that a word is related to other words that describe the YV movement. For this purpose, the content proposed in array 1 and the content of array 2 were considered in their entirety. Both the PLN dictionaries and the two arrangements coincide in a high percentage with the initial words, which implies that these three independent procedures allow comparing the results and interpreting the anger of the public in the movement of the yellow vests and its possible implications.

Keywords: Dictionaries, PLN, twitter, colère, yellow vests.

1. Introducción

En este trabajo se ha desarrollado un estudio semántico comparativo y de procesamiento de lenguaje natural (PLN) sobre el movimiento social de los chalecos amarillos (CHA) que se gestó contra el gobierno del presidente francés Macron.

Un antecedente de este trabajo es el análisis de Bernábe en 2018 respecto de los candidatos de la última elección presidencial mexicana [2]. Se han utilizado las técnicas de creación de diccionarios de esta referencia para el presente artículo, las cuales han sido de apoyo en el proceso de obtener información para el caso del análisis del movimiento CHA. La información extraída se centra en la red social Twitter y del sitio *FranceInfo* considerando el periodo del primero octubre de 2018 al 30 de octubre de 2019 [1]. De este periodo, se observaron imágenes fotográficas, palabras utilizadas que definen conceptualmente al movimiento y adjetivos que expresan las emociones asociadas con CHA. El precedente para esto es el concepto de e-comunicología desarrollado por Carrillo en 2018 [3]. Examinando los contenidos de las distintas fuentes en el periodo indicado, se construyó una lista de conceptos sobre actitudes de rechazo y enojo expresadas durante el movimiento. Esta lista denominada *arreglo 1*, ha dado lugar a todo el desarrollo del trabajo, y consiste en un conjunto de substantivos y adjetivos que describen las temáticas y algunas emociones de CHA. Considerando el *arreglo 1* y bajo herramientas para la creación de diccionarios en PLN, se desarrolló un análisis para conocer opiniones de usuarios de Twitter acerca del movimiento CHA y sus líderes.

Al mismo tiempo, se obtuvo otro conjunto de palabras consistentes y similar al *arreglo 1*. Esto se consiguió aplicando un análisis tradicional de mapeo visual y búsqueda de textos. Tales palabras se incorporaron a un *arreglo 2*, es decir, se define *arreglo 2* como el resultado de revisar contenidos noticiosos de *FranceInfo* dedicados al movimiento. Consecuentemente, y con el fin de comparar las palabras resultantes, se toma los diccionarios generados por PLN y los *arreglos 1 y 2*.

El movimiento de los CHA tiene la particularidad respecto de otros de que se preparó y se sostuvo durante 18 meses sobre la base de las redes sociales, careciendo y evitando

toda institucionalización con la palabra que centraba sus 42 demandas y sus movilizaciones sabatinas. La palabra base fue *cólere*, ira, enojo, en castellano.

El presente trabajo se encuentra organizado como sigue: Esta introducción es la sección 1. La sección 2 se encarga de los aspectos generales del movimiento y la manera en que las redes sociales intervinieron en él. Para describir el problema desde el enfoque computacional, en la sección 3 se expone el mecanismo para la creación de diccionarios [4, 5]. En la sección 4 se utiliza un enfoque enumerativo para conseguir un conjunto de palabras con relación a temáticas denominado *arreglo 2* a través de mapeo y lectura manual. Finalmente, en la sección 5 se discute los resultados.

2. Preliminares

En las primeras semanas del movimiento de los CHA en las calles, a partir del 17 de noviembre de 2018, hasta cuatro de cada cinco franceses apoyaba el movimiento de los CHA por la subida al impuesto a los carburantes donde las protestas se convirtieron en un eco amplio contra la política tributaria del gobierno, que muchos pensaron que favorecía a los más ricos: "Estos territorios han sufrido una disminución de los servicios públicos en los últimos años. Sus habitantes se sienten abandonados por los poderes públicos e ignorados por los políticos" [6].

El movimiento se caracterizó por A) un gran tráfico cotidiano de contenidos en las redes sociales, especialmente en Facebook, que fungió como el instrumento de organización y movilización a partir de la integración de varios "grupos", que es una de las opciones de tráfico de dicha red. B) Manifestaciones y acciones de ocupación de cada sábado, convocadas a través de Facebook. El movimiento, entonces, tuvo un carácter dual, virtual real, de manera tal que entre sábado y sábado había una gran actividad – movilización digital – en la red social Facebook y, en menor medida, en Twitter. A pesar de ello nos concentramos en Twitter dado que Facebook no permite el acceso de las librerías de Python.

Es importante analizar la inconformidad de los CHA no solo a través de la recolección manual de documentos, es necesario también el uso de herramientas computacionales para estudiar la información con el fin de presentar y clasificar las palabras que expresaron dicha inconformidad para que las asocien y organicen por medio de algoritmos en PLN.

2.1. Los medios virtuales

En general la dinámica de la información digital a llevado a la obligada transferencia de datos para propósitos mercantiles, económicos, didácticos, de entretenimiento, etc. a la web. A ello hay que añadir el fenómeno adicional de las redes sociales, a través de las cuales, de manera espontánea, independiente, descentralizada y también amateur, millones de usuarios generan o reciclan contenidos por iniciativa propia, según su gusto, sus intereses y su humor. Para las instituciones – gobiernos, secretarías de estado, hospitales, alcaldías, partidos políticos, sindicatos, empresas de todo tipo, universidades y escuelas, equipos deportivos, etc. – la presencia en la web se ha convertido en una necesidad ineludible. Lo mismo sucede para los medios de masas y

para los individuos en términos de una muy fuerte tendencia a la interconexión entre ellos en las redes sociales de todo tipo.

En este escenario, la investigación acerca de los intereses, deseos, gustos y/o preocupaciones de los individuos, tanto en situaciones normales como en situaciones extraordinarias que incluyen un descontento amplio (*colère*) – como en el caso de los CHA –, no es posible de ninguna manera, prescindir el examen de los contenidos que los usuarios circulan en las redes sociales. Por ello los contenidos que el usuario de una red social genera o hace circular tienen la característica de la espontaneidad del uso del teléfono inteligente delocalizadamente y en tiempo real. En este sentido, si nos interesan particularmente los deseos y los estados de ánimo de la gente tanto en la vida cotidiana como en la situación excepcional de movimientos de masas, podemos decir que su expresión más espontánea/directa, auténtica o menos filtrada por la reflexión, es hallada en los contenidos de las redes sociales, lo que justifica que este trabajo haya elegido Twitter como almacén de datos para descubrir información de CHA.

En cierto sentido podemos decir que existe el matrimonio entre el teléfono inteligente y el *software* personalizado en forma de redes sociales que nos presenta en tiempo real la mejor y más auténtica manifestación del humor y las pasiones sociales [7]. Esta es la ventaja que nos ofrece el análisis de una red social como Twitter, que está diseñada no para la expresión de la vida cotidiana del usuario sino para sus tomas de posición respecto del acontecer social.

La alternativa doble para desarrollar este trabajo consistió en recurrir al sitio público *FranceInfo*, que presentó cotidianamente notas informativas sobre el movimiento de los CHA mientras este tuvo una presencia real notable antes de disolverse en manifestaciones sabatinas cada vez menos numerosas, menos relevantes. Por otro lado, se encuentra la opción de aproximarse a dicho movimiento a través del tráfico en *Twitter*. Se observaron dos tipos de personajes relativos a los CHA: 1) políticos profesionales y 2) los *influencers* (nacidos en y al calor del movimiento); por ello se eligieron a algunos de estos personajes para nuestro análisis en usando *Twitter*.

3. Creación de diccionarios en Twitter para el movimiento de chalecos amarillos

Este trabajo se enfoca en identificar para el movimiento de los CHA la presencia de palabras y asociaciones entre ellas en la red social Twitter mediante el uso de un algoritmo de etiquetado para creación de diccionarios sobre temas y emociones [2, 8].

La red social Twitter tiene la particularidad de que el tráfico de los contenidos en ella, se encuentran en *clusters* dado que los *tweets* se organizan muy fuertemente alrededor de los mensajes de los *influencers*. Esto lleva a que la mejor posibilidad de localizar los intereses y emociones de los usuarios sea concentrarse en el examen del tráfico alrededor de mensajes de dichos *influencers*. En el caso de los CHA, los mensajes están muy bien definidos respecto a los líderes del movimiento. Ciertamente los políticos y las personalidades oficiales también se expresaron ampliamente respecto del movimiento de los CHA y son *influencers* en general, no solo respecto de dicho movimiento. Pero junto a ellos están en primer lugar y muy destacadamente personas totalmente comunes, que no tienen ninguna presencia institucional ni mediática previa al movimiento pero que devinieron sus líderes, estos son los *influencers*

Table 1. Palabras representativas del movimiento de los CHA y sus líderes relacionadas con la palabra ciudadano.

Usuario	habitante *(habitant)*	persona *(personne)*	hombre *(homme)*	nación *(national)*	personaje *(personnage)*	individuo *(Individu)*	particular *(particulier)*	mortal *(mortel)*
@alancelin	1	70	2	6	0	0	0	0
@anatolium	0	1092	15	0	3	0	0	2
@BouloGiletJaune	9	48	7	4	0	0	0	0
@CCastaner	50	188	54	1529	3	60	3	0
@drouet_eric	0	8	0	0	0	0	6	0
@FaouziLellouche	0	2	1	0	1	1	0	0
@FlyRiderGjGRAL	3	9	4	0	0	2	3	0
@GiletsJaunesFr_	0	7	0	14	0	0	0	0
@J_Rodrigues_Off	1	41	48	1	3	3	3	0
@PLudosky	0	30	17	19	2	1	3	0

específicamente asociados con el fenómeno de los CHA y que fuera de él prácticamente no tienen presencia alguna.

En este punto y para el análisis de este trabajo, se han escogido a 10 líderes, procediendo a examinar el tráfico por sus cuentas de Twitter.

El algoritmo utilizado para el análisis del tráfico consiste en separar de los comentarios de los *tweets* las palabras relacionadas con *colère* (descontento, enojo, ira, cólera). Los títulos de las diferentes tablas del artículo, son las que corresponden a la lista que se ha denominado *arreglo 1*, la cual es el origen del desarrollo de este trabajo.

En las tablas 1-14 puede verse la columna llamada usuario. Esta columna indica los nombres de los líderes que corresponden a los *hashtags*.

Primero se procedió a la descarga de tweets y con ello a la construcción de diccionarios. En este caso, los tweets de interés se descargan con toda la información (ID, autor, mensaje, fecha, idioma, etc.). El algoritmo desarrollado utiliza una biblioteca de clips de Python que permite la extracción de comentarios de forma personalizada, de este modo es posible marcar desde una palabra hasta una oración con el contenido semántico relevante. Los comentarios que se toman de Twitter tienen una variable especial llamada *unicode*, intratable como una variable de tipo cadena. En este punto, la información se almacena en un archivo csv (valores separados por comas) y después se codifica como UTF8 para admitir caracteres especiales que incluyen marcas de acento. Esto es, una vez que se han almacenado las descargas, se separan las palabras relacionadas con los líderes y se contabilizan por búsqueda lineal para crear los diccionarios. La información procesada comprende 2000 tweets aleatorios por día y se seleccionaron 500 comentarios para cada candidato. Para el análisis de los tweets, se usaron técnicas de *crawling* para rastreo y extracción de información de palabras bien identificadas por el problema [8].

3.1. Procedimiento de selección de palabras relativas a CHA

El primer acercamiento para descubrir la información sobre el tema de CHA, se centró en obtener a través de descargas en *Twitter* la información que se relaciona entre los líderes del movimiento social CHA y las palabras representativas identificadas en la observación inicial. Este proceso se efectúa mediante un algoritmo programado en

Python que devuelve *tweets* con la información que el usuario escribió en su *hashtag,* las palabras que se indican en la tabla 1.

Tabla 2. Frecuencia de aparición de palabras relacionadas al enojo (*colère*).

Usuario	**enojo** *(colère)*	**odio** *(haine)*
@alancelin	4	15
@anatolium	0	19
@BouloGiletJaune	0	20
@CCastaner	0	114
@drouet_eric	0	4
@FaouziLellouche	0	1
@FlyRiderGjGRAL	0	1
@GiletsJaunesFr_	0	2
@J_Rodrigues_Off	0	13
@PLudosky	6	4

Tabla 3. Justicia (*justice*).

	ley *(loi)*	**correcto** *(droit)*	***acto*** *(acto)*	**código** *(code)*
@alancelin	97	43	5	1
@anatolium	40	0	6	12
@BouloGiletJaune	51	24	10	0
@CCastaner	151	353	143	120
@drouet_eric	2	4	12	2
@FaouziLellouche	0	2	0	0
@FlyRiderGjGRAL	18	19	4	1
@GiletsJaunesFr_	3	2	11	0
@J_Rodrigues_Off	9	11	34	21
@PLudosky	2	11	2	23

Tabla 4. Desconfianza (*méfiance*).

	sombra *(ombre)*	**duda** *(dout)*	**prudencia** *(prudence)*	**vigilancia** *(vigilance)*
@alancelin	2	0	0	0
@anatolium	13	295	0	0
@BouloGiletJaune	13	2	0	0
@CCastaner	32	18	31	6
@drouet_eric	2	0	0	0
@FaouziLellouche	0	2	0	0
@FlyRiderGjGRAL	12	2	0	0
@GiletsJaunesFr_	6	0	0	0
@J_Rodrigues_Off	8	4	0	0
@PLudosky	5	5	0	0

Los periodos de descargas fueron 5 de 10 días cada uno con interrupción de dos días entre cada periodo, proceso iterativo durante dos meses. Para este problema, se ajustó un algoritmo PLN con fines electorales [2, 5].

3.1.1. Tablas de sinónimos

El segundo paso fue generar un diccionario cuyo peso de probabilidad era casi nulo ya que fue realizado con palabras propuestas (descontento, enojo, ira, cólera)., de tal modo que este *arreglo 1*, se extendió para formar el primer diccionario. Se repitió el proceso para encontrar sinónimos de las palabras de este primer diccionario y formar

los siguientes diccionarios. La búsqueda fue sintáctica contando el número de veces que cada una de las palabras eran repetidas.

Tabla 5. Pueblo (*peuple*).

usuario	publico (*public*)	prisa (*presse*)	mundo (*monde*)	masa (*masse*)	nombre (*nombre*)	fluir (*flot*)	multitud (*foule*)	asistencia (*assistance*)
@alancelin	3	17	13	4	1	0	0	0
@anatolium	297	8	51	1	11	1	0	0
@BouloGiletJaune	6	4	14	0	10	0	11	0
@CCastaner	101	88	52	5	29	0	3	1
@drouet_eric	0	0	2	4	2	0	0	0
@FaouziLellouche	0	85	0	0	0	0	0	0
@FlyRiderGjGRAL	3	1	512	4	11	0	0	0
@GiletsJaunesFr_	5	0	1	0	6	0	0	0
@J_Rodrigues_Off	2	86	18	3	7	0	0	0
@PLudosky	3	83	13	0	5	0	0	0

Tabla 6. Fatiga (*Fatigue*).

usuario	malestar (*malaise*)	tristeza (*tristesse*)	abadono (*abandon*)	dolor (*souffrance*)	aterrador (*effrayant*)	débiles (*faibles*)
@alancelin	0	0	1	0	0	0
@anatolium	0	3	0	0	0	0
@BouloGiletJaune	1	3	649	0	0	1
@CCastaner	0	1	0	3	2	40
@drouet_eric	0	0	0	0	0	0
@FaouziLellouche	0	0	0	0	0	0
@FlyRiderGjGRAL	0	0	0	0	0	0
@GiletsJaunesFr_	0	0	0	0	0	1
@J_Rodrigues_Off	0	0	0	0	1	2
@PLudosky	0	0	0	0	0	33

Tabla 7. Esperanza 1 (*spoir*).

usuario	entender (*comprendre*)	apoyo (*soutenir*)	pacifista (*pacifiste*)	comprender (*comprends*)	cambio (*change*)	honor (*honneur*)
@alancelin	0	0	0	1	1	0
@anatolium	6	0	0	7	45	2
@BouloGiletJaune	3	5	2	2	13	1
@CCastaner	16	11	0	13	105	3
@drouet_eric	4	0	0	0	0	0
@FaouziLellouche	0	0	0	1	0	0
@FlyRiderGjGRAL	2	0	0	3	6	0
@GiletsJaunesFr_	0	4	0	2	0	1
@J_Rodrigues_Off	3	5	0	2	2	2
@PLudosky	0	0	0	3	5	1

Los resultados se muestran en las siguientes tablas. Las tablas 2-14 muestran la frecuencia de aparición de palabras relacionadas al enojo, la justicia, la desconfianza, el pueblo, la fatiga, la esperanza, la humillación y la indignación. Las palabras encontradas se presentan entre paréntesis en su idioma original.

Tabla 8. Esperanza 2 (*spoir*).

usuario	milagro *(miracle)*	fe *(foi)*	oferta *(tendre)*	felicidad *(bonheur)*	tranquilizar *(rassure)*	agarrar *(saisir)*
@alancelin	0	2	1	0	0	0
@anatolium	0	1076	4	0	0	0
@BouloGiletJaune	0	13	6	0	0	0
@CCastaner	3	128	147	0	8	70
@drouet_eric	0	4	0	0	0	0
@FaouziLellouche	1	2	0	0	0	0
@FlyRiderGjGRAL	0	9	86	2	1	0
@GiletsJaunesFr_	0	4	0	0	0	0
@J_Rodrigues_Off	1	19	84	1	0	0
@PLudosky	1	8	85	0	0	0

Tabla 9. Humillación (*humilliation*).

usuario	vergüenza *(honte)*	miedo *(peur)*	tontos *(cons)*	criminales *(criminels)*
@alancelin	3	0	104	0
@anatolium	15	5	981	0
@BouloGiletJaune	0	8	66	0
@CCastaner	523	200	301	1
@drouet_eric	4	0	28	0
@FaouziLellouche	1	0	1	0
@FlyRiderGjGRAL	7	0	28	0
@GiletsJaunesFr_	0	1	4	0
@J_Rodrigues_Off	3	1	25	0
@PLudosky	4	1	20	0

Tabla 10. Indiganación 1 (*indignation*).

usuario	harto *(ras-le-bol)*	injusticia *(injustice)*	victimas *(victimes)*	cargar *(charger)*	*hipocresía* *(hypocrisie)*
@alancelin	0	0	0	0	0
@anatolium	0	0	4	0	0
@BouloGiletJaune	0	1	2	0	0
@CCastaner	1	6	22	0	4
@drouet_eric	0	0	0	8	0
@FaouziLellouche	0	0	0	0	0
@FlyRiderGjGRAL	0	0	2	0	0
@GiletsJaunesFr_	0	1	0	0	0
@J_Rodrigues_Off	0	1	1	0	0
@PLudosky	0	0	1	0	0

Para finalizar el proceso se realizó una búsqueda manual mediante lectura de artículos recientes en el sitio FranceInfo, donde se encontraron más palabras significativas para el análisis.

Así se construyó otro diccionario denominado arreglo 2 que contiene las palabras

encontradas en los artículos y se comparó con las palabras de las descargas de Twitter y los diccionarios PLN.

Tabla 11. Indiganación 2 (*indignation*).

usuario	**manifestantes** *(manifestants)*	**traición** *(trahis)*	**movilización** *(mobilisation)*	**movimiento** *(mouvement)*	**oligarquía** *(oligarchie)*
@alancelin	0	0	0	66	0
@anatolium	2	40	0	5	1
@BouloGiletJaune	1	1	1	65	0
@CCastaner	8	0	4	12	1
@drouet_eric	0	0	0	0	0
@FaouziLellouche	0	0	0	0	0
@FlyRiderGjGRAL	2	0	149	6	0
@GiletsJaunesFr_	0	0	1	31	0
@J_Rodrigues_Off	1	0	0	7	0
@PLudosky	0	1	0	12	1

Tabla 12. Justicia 2 (*justice*).

usuario	**demandas** *(demandes)*	**Imparcial** *(impartial)*	**rigor** *(rigueur)*	**justicia** *(justice)*	**crisis** *(crise)*
@alancelin	0	0	0	7	0
@anatolium	0	1	1	1	0
@BouloGiletJaune	0	0	0	79	0
@CCastaner	6	0	0	222	24
@drouet_eric	0	0	0	2	0
@FaouziLellouche	0	0	1	0	0
@FlyRiderGjGRAL	0	1	0	75	0
@GiletsJaunesFr_	0	0	0	6	3
@J_Rodrigues_Off	0	0	1	80	0
@PLudosky	0	0	1	127	0

Tabla 13. Justicia 2 continuación (*justice*).

usuario	**reivindicación** *(revendication)*	**Inaceptable** *(inaceptable)*	**En contra** *(contre)*	**manifestar** *(manifester)*
@alancelin	0	0	9	0
@anatolium	0	8	56	0
@BouloGiletJaune	7	0	378	1
@CCastaner	5	4	1702	1
@drouet_eric	0	0	8	0
@FaouziLellouche	0	0	87	0
@FlyRiderGjGRAL	4	0	17	3
@GiletsJaunesFr_	3	0	21	1
@J_Rodrigues_Off	7	0	118	6
@PLudosky	42	8	117	1

La tabla 8 presenta la continuación de la tabla 7, ambas relacionadas con la esperanza.

Tabla 14. Ira (*colère*).

usuario	tensión *(tension)*	querer *(veulent)*	mierda *(chier)*	fuerza *(force)*	peligro *(danger)*	alboroto *(saccage)*	violencia *(violence)*	roto *(casse)*	clamar *(clamer)*
@alancelin	0	4	0	2	0	0	6	2	0
@anatolium	0	1	1	17	2	0	1	0	0
@BouloGiletJaune	0	10	1	29	3	0	22	4	5
@CCastaner	7	56	19	207	165	3	67	11	0
@drouet_eric	0	0	0	0	0	0	0	0	0
@FaouziLellouche	0	0	0	0	0	1	0	0	0
@FlyRiderGjGRAL	0	2	0	1	0	0	11	6	3
@GiletsJaunesFr_	0	2	0	3	0	0	14	4	0
@J_Rodrigues_Off	1	10	0	2	0	2	22	5	0
@PLudosky	0	34	0	2	2	2	13	2	3

Tabla 15. Sentimientos.

haine	*excédés*	*répressive*	*ressenti*	*mortes*	*impartial*	*tendre*	*dépassés*
tension	*dèsespoir*	*infâmes*	*énerve*	*choqués*	*impartial*	*bonheur*	*Emmerdez*
colere	*ras-le-bol*	*hypocrisie*	*fait mal*	*force*	*l'indignation*	*solidaritè*	*faibles*
émotion	*inefficace*	*peur*	*réclamer*	*danger*	*traître*	*fierté*	
sincère	*insensée*	*dégoût*	*profonde*	*saccage*	*inacceptable*	*tristesse*	
calmer	*manipulabl*	*dégâts*	*combatif*	*violence*	*contre*	*désespère*	
veulent	*fierté*	*sacrifice*	*chier*	*périlleuse*	*beautè*	*désespérant*	
exprimée	*honneur*	*déçus*	*vandalisé*	*clamer*	*foi*	*souffrance*	

Tabla 16. Acciones.

acte	*abandon*
vigilance	*criè*
presse	*profané*
assistance	*décolèrent*
apaisera	*saisir*
rassure	

3.2. Relaciones de sinónimos

Una relación y agrupación de palabras en los diccionarios creados por los algoritmos PLN, generó grupos nuevos de palabras. Para este objetivo, se tomó la información de los diccionarios creados y se estableció el vínculo de sinonimia que tenían las palabras. En consecuencia, se produjeron nuevos grupos de palabras para poder describir la relación que existe entre ellas como se muestra en las siguientes tablas.

Tabla 17. Personas.

public	*homme*	*criminels*	*victimes*	*justicier*
monde	*national*	*inoffensive*	*manifestants*	
masse	*personnage*	*pacifiste*	*trahis*	
nombre	*individu*	*fatigué*	*cons*	
flot	*particulier*	*malaise*	*déplorables*	
foule	*effrayant*	*douloureux*	*prédatrice*	
habitant	*d'exaspération*	*déception*	*salauds*	
personne	*intenable*	*résignés*	*détruire*	

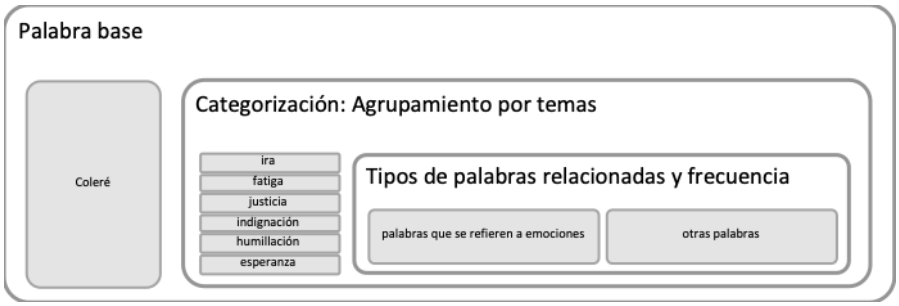


Fig. 1. Esquema de construcción del arreglo 2.

Tabla 18. Gobierno.

loi	*pilles*
droit	*intérêts*
code	*oligarchie*

Tabla 19. Muerte.

mortel	*abattus*
suicidez-vouz	

4. Noticias de FanceInfo y Twitter

Además del tráfico en Twitter, la actividad de los chalecos amarillos en la web se ha podido seguir a partir del estudio de las noticias francesas en el canal FranceInfo, dedicado a la publicación de noticias ininterrumpidamente y accesibles mundialmente en su sitio web francetvinfo.fr [9]. Las publicaciones del sitio muestran la postura de articulistas, actores políticos, y con ello posibilitan una visión general del contenido de las noticias sobre los CHA en Francia.

Las notas periodísticas como registro diario y cambiante son expresión de la sociedad actual, su importancia en el entendimiento y reacción emotiva del público sobre los temas es crucial; de ahí la iniciativa de diseñar un primer diccionario que represente el uso de las palabras expresivas en las notas, aunque hay que tener en cuenta que están definidas por el contexto de su uso, en este caso el periodístico, pero que pueden ser base para un estudio profundo de contenidos en redes sociales.

El trabajo consistió en encontrar conceptos a partir de la palabra *colère* y determinar la relación de dichos conceptos con lo que se dice sobre el movimiento de los CHA entre 16 de marzo al 20 de octubre de 2019. El lugar donde se ubican estas noticias es un tipo específico de contenido en la *web* que son las crónicas del sitio *FranceInfo*. La finalidad es reconocer los patrones de uso de conceptos que expresan temas y emociones del movimiento de los CHA. Después de la revisión detallada de 1200 notas publicadas por *FanceInfo* dentro del periodo establecido, se obtuvieron 213 menciones de la palabra *colère* [10].

Las asociaciones cercanas que acompañan a la palabra *colère* que se ubican dentro del párrafo se extrajeron para construir una tabla con agrupamiento en temáticas.

Tabla 16. Palabras representativas del problema CHA en las categorías ira y justicia.

IRA				**JUSTICIA**			
Emoción	**R**	**Otro**	**R**	**Emoción**	**R**	**Otro**	R
emoción	4	*violence*	33	*intérêts*	2	(*colère*) *contre*	20
calma calmer	4	*force*	21	*l'indignation*	1	*crise*	8
veulent	4	*casse*	9			*revendication*	6
exprimée	2	*problèmes*	6			*justice*	5
tension	1	*pilles*	4			*manifester*	5
sincère	1	*clamer*	3			*l'incompréhension*	3
fait mal	1	*réclamer*	2			*légitime*	2
ressenti	1	*profonde*	2			*révolution*	2
		chier	2			*justicier*	1
		saccage	2			*dénoncer*	1
		combatif	1			*demandes*	1
		vandalisé	1			*impartial*	1
		mortes	1			*rigueur*	1
		choqués	1			*traître*	1
		danger	1			*inacceptable*	1
		périlleuse	1				

Tabla 17. Palabras representativas del problema CHA en las categorías indignación y humillación.

INDIGNACION				**HUMILLACIÓN**			
Emoción	**R**	**Otro**	**R**	**Emoción**	**R**	**Otro**	**R**
ras-le-bol	2	*mouvement*	22	*honte*	6	*déçus*	2
excédés	1	*manifestants*	17	*peur*	1	*criminels*	2
dèsespoir	1	*mobilisation*	11	*dégoût*	1	*sacrifice*	1
inefficace	1	*victimes*	3	*dégâts*	1	*cons*	1
		hypocrisie	2			*déplorables*	1
		débordements	2			*prédatrice*	1
		insensée	1			*salauds*	1
		manipulabl	1			*détruire*	1
		injustice	1				
		charger	1				
		trahis	1				
		répressive	1				
		infâmes	1				
		confrontation	1				
		oligarchie	1				
		périlleuse	1				

Las palabras que aparecen con mayor frecuencia fueron colocadas en el diccionario. Para fines del estudio, antes de presentar el mapa se presenta una tabla fraccionada por una segunda relación que las palabras encontradas tienen con las seis temáticas predominantes: ira, justicia, indignación, humillación, esperanza y fatiga; como se muestra en la figura 1.

Tabla 18. Palabras representativas del problema CHA en las categorías esperanza y fatiga.

ESPERANZA				FATIGA			
Emoción	**R**	**Otro**	**R**	**Emoción**	**R**	**Otro**	**R**
inoffensive	1	*change*	5	*fatigué*	4	*intenable*	1
comprendre	1	*pacifiste*	3	*malaise*	3	*faibles*	1
fierté	1	*comprends*	3	*tristesse*	3	*criè*	1
		honneur	2	*douloureux*	2	*profané*	1
		rêve	2	*déception*	2	*suicidez-vouz*	1
		tendre	2	*abandon*	2	*abattus*	1
		apaisera	2	*souffrance*	2	*dépassés*	1
		saisir	2	*décolèrent*	1	*emmerdez*	1

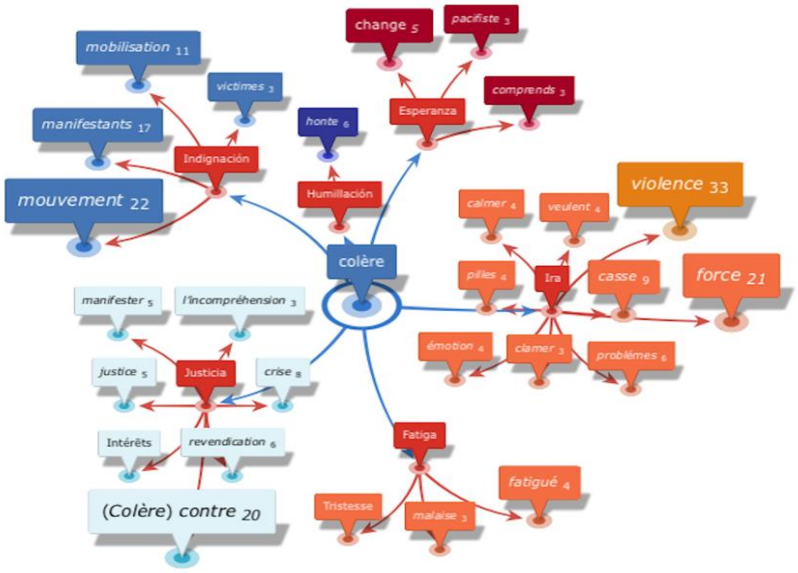


Fig. 2. Mapa de palabras utilizadas con mayor frecuencia en FranceInfo.

Las palabras que aparecen con mayor frecuencia relacionadas con cólera en las noticias de FranceInfo son violencia, fuerza, (furia en) contra, manifestantes y movimiento. Mientras que de las emociones de las que más se habla son vergüenza, emoción y fatiga, seguidas por la incomprensión, malestar y tristeza (arreglo 2).

Para la mejor visualización de las palabras y el esquema antes propuesto se creo un mapa (Figura 2) donde la palabra central es *colére* y las relaciones se agrupan en las seis temáticas principales. Se pueden apreciar las palabras con mayor frecuencia de aparición en tamaño mayor a las que tienen una frecuencia menor pero que aún así sobresalen.

5. Conclusiones

A lo largo del texto se ha buscado entender los deseos, estados de ánimo y tomas de posición desde y sobre el movimiento de los chalecos amarillos con base en el análisis de contenidos en dos medios específicos: Twitter y FranceInfo. Los autores de los contenidos estudiados son periodistas, líderes de opinión e *influensers*.

El desarrollo de la investigación se llevó a cabo a partir de un primer arreglo de palabras obtenido de la observación empírica, posteriormente, esas expresiones fueron evaluadas con un algoritmo de PLN desarrollado concretamente para ese propósito y que analiza las relaciones entre las palabras del arreglo 1, contenidos de Twitter y sus autores. Adicionalmente, se realizó un estudio de los contenidos de noticias publicadas por FranceInfo que resultó en un segundo arreglo de palabras y temáticas.

Como resultado de la intersección y estudio comparativo de los arreglos y diccionarios desarrollados se encontró:

1. Que las palabras (*colère*) *contre* (cólera en contra), *justice* (justicia) y *honte* (vergüenza) muestran los temas primordiales del movimiento debido a que se encuentran con mayor frecuencia en todos los contenidos estudiados. Estás expresiones son compatibles con la palabra guía inicial *colère*.

2. Se encontró que los conceptos que se repiten en los estudios están relacionados con estados de ánimo, emociones y temáticas complejas. La mayoría negativas, de inconformidad y con un alto contendido de malestar, describen al movimiento en cuanto a una agitación social en la que ciertos conceptos abstractos como justicia o movilización son compatibles con palabras emotivas como *honte* o *peur* y negativamente como *bonheur* en tanto algo faltante o a lo que se aspira.

3. Las palabras que se repiten independientemente de la frecuencia son: *malaise, tristesse, faibles, comprendre, soutenir, pacifiste, comprends, change, honneur, miracle, foi, tendre, bonheur, rassure, saisir, peur, cons. criminels, injustice, victimes, charger, hypocrisie, manifestants, trahis, mobilisation, mouvement, oligarchie, demandes, impartial, rigueur, justice, crise, revendication, inaceptable, manifester, tensión, veulent, chier, force, danger, violence, casse, clamer.* (*Inquietud, tristeza, debilidad, comprensión, apoyo, pacifismo, comprensión, cambio, honor, milagro, fe, ternura, felicidad, tranquilidad, comprensión, miedo, idiotas. criminales, injusticia, víctimas, acusación, hipocresía, manifestantes, traicionados, movilización, movimiento, oligarquía, demandas, imparcial, minuciosidad, justicia, crisis, demanda, inaceptable, demostrar, tensión, querer, mierda, fuerza, peligro, violencia, rotura, gritar*).

Las herramientas computacionales son de utilidad al analizar los contenidos web para exponer las tomas de posición y el entendimiento de fenómenos sociales como el movimiento de los chalecos amarillos.

Los resultados del análisis semántico del movimiento de los chalecos amarillos arrojan una estructura como conjunto de relaciones entre conceptos abstractos (como justicia o injusticia, abandono, desigualdad, etc.) y conceptos para emociones (cólera,

sufrimiento, tristeza, etc.). Estas relaciones son independientes de los conceptos que definen los temas específicos del movimiento (carburantes, impuestos, salarios, etc.), es decir, tal independencia es la hipótesis que comprobar comparando la estructura semántica de otros movimientos de descontento social; tales relaciones deben arrojar lo que es el patrón o forma de los movimientos de descontento. Si esto resulta cierto, entonces, conociendo la forma o estructura del descontento social, es posible utilizar los estudios de tráfico de una manera ya no formal sino cuantitativa para determinar ya no la estructura formal del descontento sino la magnitud o grado de este. En otras palabras, debemos suponer que existe el patrón o la forma del descontento social, y que dado este, existe también el grado de descontento propio de cada movimiento, el cual puede ser estudiado cuantitativamente. Lo anterior abre un programa de investigación formal y cuantitativo de los movimientos sociales.

Referencias

1. Franceinfo: https://www.francetvinfo.fr/ (2019)
2. Bernábe, M., Espinoza, E., Estrada, J., Sánchez, A.: Algoritmo para recopilar y ordenar datos de Twitter mediante el uso de diccionarios en Python, Innovaciones tecnológicas en las ciencias computacionales: Montiel & Soriano Editores S.A. de C.V., pp.103–109 (2019)
3. Carrillo, A.: Métodos visuales. Mediología de la existencia digital en su imagen icónica. La e-comunicología mediológica. En Comunicación y música: mensajes, manifestaciones y negocios (2018)
4. Ortiz, Á., Pérez, O., Vargas, E.: Estudio en tendencia diarias en TWITTER", pp. 1–80 (2015)
5. Bernábe, M., Espinoza, E., González, R., Cerón C.: Algorithm for collecting and sorting data from twitter through the use of dictionaries in Python. In: LKE, 7th International Symposium on Language & Knowledge Engineering LKE (2019)
6. COLPISA / AFP: ¿Quiénes son los 'chalecos amarillos'? HOY, https://www.hoy.es/internacional/union-europea/quienes-chalecos-amarillos-20181205014100-ntrc.html?ref=https:%2F%2Fwww.google.com%2F (2018)
7. McLuhan, M.: Understanding Media. The Extensions of Man. MIT Press (1994)
8. Waykar, P., Wadhwani, K., More, P.: Sentiment analysis in twitter using Natural Language Processing (NLP) and classification algorithm. In International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), pp. 79–81 (2016)
9. Franceinfo: https://www.francetvinfo.fr/ (2019)
10. Noticias sobre Chalecos Amarillos en Franceinfo, https://www.francetvinfo.fr/recherche/?request=col%C3%A8re (2019)

ISSN 1870-4069

Identificación y localización de células gliales en imágenes de tejido cerebral

Karla Mandujano-López[1], Juan Villegas-Cortez[2], Graciela Román-Alonso[1], Arturo Zuñiga-López[3], Edgar Rangel-López[4], César Benavides-Alvarez[1]

[1] Universidad Autónoma Metropolitana,
Departamento de Ingeniería Eléctrica,
Ciudad de México,
México
[2] Universidad Autónoma Metropolitana,
Departamento de Sistemas,
Ciudad de México,
México
[3] Universidad Autónoma Metropolitana,
Departamento de Electrónica,
Ciudad de México,
México
[4]Instituto Nacional de Neurología y Neurocirugia
Manuel Velasco Suárez,
Ciudad de México,
México

{karlamanlop, cesarbenavides32}@gmail.com, juanvc@azc.uam.mx,
grac@xanum.uam.mx, raledg@hotmail.com

Resumen. En este artículo se propone un sistema de procesamiento digital de imágenes para la identificación automática de células gliales de tejido cerebral. El propósito es apoyar al especialista, que al seleccionar una región de interés (ROI) en una imagen de tejido cerebral, se procede a buscar esa ROI en toda la imagen en cuestión. La metodología propuesta trabaja a nivel textura por lo que primero aplica un preprocesamiento a toda la imagen: se extraen atributos CBIR probados en la literatura, dando como resultado la posibilidad de analizar de forma local a una imagen específica para localizar regiones similares en la misma imagen. Nuestra propuesta hace uso de una discriminación de los patrones conformados usando métrica euclidiana, tras implementar el método K-Nearest Neighbors para una auto-organización de los patrones de textura conformados. Se han realizado pruebas con un total de 312 imágenes de laboratorio, puramente experimentales obtenidas del microscopio en formato TIFF, y se han obtenido resultados prometedores bajo la supervisión del especialista en biología celular.

Palabras clave: Clasificación de células, procesamiento digital de imágenes, CBIR, K-NN, reconocimiento de patrones.

Identification and Localization of Glial Cells of the Brain Tissue

Abstract. This paper proposes an image digital processing system for the automatic identification of glial cells of the brain tissue. The purpose is to support the specialist to find within an image all the areas where exist the region of interest (ROI) previously defined by the specialist. The proposed methodology is developed at the texture level, therefore, it is first applied a preprocessing of the whole image by extracting CBIR attributes found in references. This preprocessing stage results in the possibility of analyzing on a local basis, a specific image to localize similar regions within the same image. Our proposal discriminates the conformed patterns by using an Euclidean metric after implementing a K-Nearest Neighbors method in order to self organize the conformed texture patterns. 312 images were analyzed. These images are completely experimental and were obtained from a microscope in TIFF format. The results are promising and were discussed with the aid of a specialist in cell research.

Keywords: Classifying cells, digital image processing, CBIR, K-NN, pattern recognition.

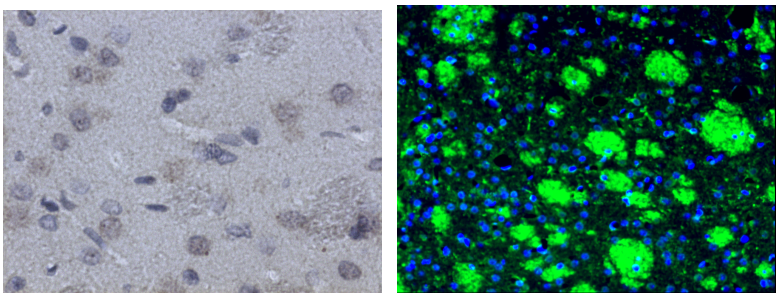
1. Introducción

Para nosotros como seres humanos, seres inteligentes y dinámicos, el sentido de la vista es el más común que usamos, prácticamente pasa desapercibido que de los cinco sentidos que tenemos éste sea actualmente el más explotado. Un ejemplo de lo refinado que es nuestro sentido de la vista es el "reconocimiento automático" que hacemos de imágenes, ya sea en color y/o forma, logrando la clasificación o identificación de objetos.

Un problema actual es la identificación de objetos, o puntos, en imágenes de tejidos celulares; estas imágenes son adquiridas y analizadas para poder diagnosticar enfermedades en pacientes acorde con un propósito médico, siendo así que actualmente, el diagnóstico para cada paciente requiere del análisis visual de muestras de tejidos por parte de un médico especializado, o personal debidamente capacitado. Siendo ésta una tarea repetitiva y dependiente de la experiencia y propósitos particulares del médico, es que no siempre es posible garantizar un correcto análisis visual de dichas imágenes.

La *glía* es un grupo heterogéneo de células neurales que cumplen funciones diversas en la fisiología del cerebro. Con base en su morfología, fisiología y localización en el cerebro podemos identificar los siguientes tipos principales de glía: microglía, oligodendroglía y astroglía [8]. En particular, el estudio de las células gliales es muy importante porque puede otorgar información sobre si existe o no una enfermad neurodegenerativa en el paciente.

Para esto, se cuenta con dos tipos de imágenes de campo claro y de fluorescencia, que aunque a primera vista son muy diferentes por los colores, la información que encontramos dentro de ellas es la misma.



(a) (b)

Fig. 1. Ejemplo de imágenes de tejido cerebral a) Campo Claro y b) Fluorescencia.

Hasta ahora lo que se realiza es que la persona especialista observa tejidos a través de un microscopio ver Figura 1, ve su forma y hace un conteo de los objetos que se encuentran dentro de la imagen. Esta tarea suele llevar mucho tiempo por cada imagen y también depende de la experiencia del especialista y de la capacidad visual de la persona.

En este trabajo mostramos el desarrollo e implementación de un sistema de identificación y localización de células gliales, aplicado en una base de datos de un grupo de imágenes digitales de tejido cerebral con muestras experimentales del Instituto Nacional de Neurología en México[1]. Presentamos un avance basado en el estudio de textura de la imagen, con la finalidad de caracterizar el patrón de la textura de superficie a partir de rasgos estadísticos, basándonos en rasgos ya probados en otros estudios de reconocimiento y clasificación de imágenes por medio de la técnica CBIR (Content Based Image Retrieval), a saber: la media, la varianza y la homogeneidad en el espacio de color HSI.

La metodología de "reconocimiento de imágenes basada en su contenido propio", o Content Based Image Retrieval (CBIR), se ha utilizado con éxito para el reconocimiento de imágenes partiendo del contenido de las imágenes mismas, sin tener que apoyarse en el uso de datos externos, e.g. etiquetas de texto que indiquen qué contiene la imagen, tan sólo estudiando y analizando la imagen propia desde diferentes enfoques, como lo es por medio de la textura [1]. Debido a las características de las imágenes del tejido cerebral, consideramos prudente su utilización en este trabajo.

En nuestra propuesta presentamos un alcance de clasificación de regiones de imágenes de tejido cerebral a partir de un análisis local de la imagen.

Para esto se hace una extracción de atributos de textura en un espacio de color HSI buscando tener rasgos independientes del color, para tener un patrón robusto que pueda caracterizar una región específica de la imagen. El procedimiento requiere que previamente el especialista de la salud indique la región

[1] Página oficial: https://www.gob.mx/salud/innn, Fecha de consulta: 19 de mayo, 2020.

que sea de su interés localizar dentro de la imagen. Posteriormente hacemos un entrenamiento de ese tipo de regiones a buscar dentro de la imagen, a partir de un algoritmo K-Nearest Neighbors de auto organización de atributos, para poder hacer comparaciones posteriores y así poder determinar el tipo de regiones más parecidas dentro de la misma.

El artículo se organiza de la siguiente forma: en la Sección 2 presentamos el estado del arte de nuestro problema de estudio, en la Sección 3 la metodología se presenta con detalle. La sección 4 muestra la implementación y resultados obtenidos, así como una discusión de los mismos. Finalmente, en la Sección 5 compartimos nuestras conclusiones.

2. Estado del arte

En la última década, con el aumento de la capacidad de almacenamiento en la red o en la nube y la masificación de las redes sociales, la disponibilidad por Internet de un gran volumen de imágenes digitales con diversas características es posible; y el hecho de que tales recursos no estén clasificados ha propiciado el desarrollo de herramientas que traten de paliar este problema. Hablando de cantidades reportadas en el año 2017, se estimó en un billón la cantidad de imágenes digitales acorde a una investigación reportada por el servicio MYLIO[2], tal que el 85 % de ellas fueron tomadas por un teléfono móvil. Ante esta realidad se tiene la posibilidad de crear nuevos repositorios de bases de datos de imágenes, pero el reto es clasificarlas acorde a su contenido, de forma no-supervisada, y con una alta eficiencia de forma simple y rápida, todo ello orientado al trabajo fundamental de organizar, proteger y compartir.

En particular, en el campo de la ciencia los grandes avances en la tecnología digital, así como en el campo de la microscopía de luz en los últimos años, determinaron que la imagen celular digital había adquirido una importancia creciente en la biología celular [7]. En el diagnóstico médico, el recuento de células sanguíneas juega un papel muy importante, ya que su aumento o disminución hace que se produzcan muchas enfermedades. Existen diferentes técnicas de recuento de células sanguíneas, el método convencional de conteo manual bajo microscopio requiere mucho tiempo y produce resultados inexactos [3], puede tener algunas limitaciones inherentes de los observadores humanos [7].

Por estas limitaciones existen ya nuevos métodos de adquisición y más sofisticados, como los empleados en la detección de alto contenido, generalmente producen una gran cantidad de datos que exige el poder de las computadoras para analizarlos [7], si bien existen soluciones de hardware como el Contador de hematología automatizado, los países en desarrollo no son capaces de disponer de máquinas costosas e inasequibles en todos los laboratorios de los hospitales [3].

Por lo cual al no tener el hardware necesario, podemos hacer uso de el análisis de imágenes por computadora en microscopía celular ya que puede abordar diversas tareas como clasificación y conteo celular, estudios sobre estructuras

[2] Página oficial: http://tinyurl.com/y2jo5wal, Fecha de consulta: 19 de mayo, 2020.

subcelulares o estudios sobre células vivas, entre otras, sin necesidad de grandes máquinas costosas. También se tiene utilidad en campos como patología, biotecnología vegetal, toxicología, desarrollo de fármacos y otros [7].

Para proporcionar una alternativa rentable basada en software y una alternativa eficiente para reconocer y analizar las células del tipo sanguínea [7], es necesario aportar una metodología para hacer un conteo automático de células basado en el procesamiento digital de imágenes. Diferentes etapas deben ser consideradas, desde la adquisición de la imagen, preprocesarla, mejorarla, segmentarla, la extracción de características, aplicación de operaciones morfológicas y la detección de anormalidades, se pueden considerar diferentes problemáticas, como la restauración de la imagen, el filtrado de ruido y la reducción del desenfoque. La segmentación de imágenes es también un procesamiento importante aplicado mediante diferentes algoritmos para incluir la separación de los agregados celulares [7].

Para este propósito, se han implementado pocas técnicas de preprocesamiento y posprocesamiento en la imagen de las células sanguíneas para proporcionar una imagen mucho más clara y limpia [3], es aquí donde existe un campo amplio de investigación. Una de las técnicas que ha dado resultados exitosos es la técnica CBIR la cual ha sido probada en identificación y clasificación de imágenes de interés [1], tales como los escenarios naturales y la clasificación de rostros humanos.

Muchos sistemas y herramientas CBIR se han desarrollado para la consulta de imágenes [2], tal como Query By Image Content (QBIC) desarrollado en los años 90s por IBM [4], que permite al usuario consultas dentro de una gran base de datos con base a atributos de la imagen.

CBIR es un sistema que tiene dos etapas principales: la inserción de datos y el procesamiento de consultas. La inserción de imágenes, es un método no supervisado para la extracción de información (tal como color, forma o texturas), de la base de datos para ser guardados y autoagrupados por medio de un algoritmo para su uso futuro. En el procesamiento de consultas, mediante una imagen de muestra, se obtienen sus características, para ser comparadas con métricas especificas hacia los racimos de datos auto agrupados en la etapa previa, y así obtener el valor dentro de la clasificación previamente hecha de la base de datos [6].

Anteriormente la clasificación de las imágenes se llevaba acabo de forma manual y por medio de metadatos insertados en las imágenes tales como etiquetas de texto, pero, en una base de datos muy grande conlleva un costo enorme para su clasificación [12]. Con el uso de CBIR la identificación y clasificación de células sanguíneas se puede automatizar como se ha aplicado a otro tipo de imágenes (rostros y escenarios naturales), ya que se realiza por medio de características de la misma imagen sin necesidad de metadatos, etiquetas o intervención externa, esta propuesta es presentada en el presente trabajo.

Por definición, las enfermedades neurodegenerativas (Parkinson, Alzheimer, Huntington, Esclerosis lateral amiotrófica y tumores cerebrales entre las principales) comprenden tanto la pérdida progresiva e irreversible de neuronas; así

como la proliferación descontrolada de algunos otros tipos celulares en el caso de cáncer en el Sistema Nervioso. Todo esto conlleva a una pérdida del control óptimo de aquellas funciones reguladas a nivel nervioso. La determinación microscópica de muestras de pacientes con padecimientos neurodegenerativos sigue siendo considerada una herramienta muy útil para el diagnóstico y evolución de estas enfermedades. Bajo técnicas diversas de histoquímica e inmunofluorescencia contra marcadores específicos es posible obtener un estimado en biopsias de estos pacientes sobre la disminución en el número de neuronas o la proliferación descontrolada de oligodendrocitos o astrocitos en casos de tumores cerebrales [9–11].

3. Metodología

Para el problema de la identificación y localización de las células gliales en imágenes digitales se propone el modelo mostrado en la Figura 2, en el que se plantean las siguientes etapas:

Etapa 1

- *Adquisición* de la base de datos de imágenes de tejido cerebral, todas ellas se encuentran en el espacio de color RGB.
- *Preprocesamiento*:
  1. Se aplica a cada unan de las imágenes un filtro gaussiano.
  2. Ecualización de imagen.
  3. Exportación de las imágenes del espacio de color RGB original, al espacio HSI, con el objetivo de trabajar con más información independiente del color de la imagen y buscando tener información de las texturas de las imágenes.

Etapa 2

- El usuario selecciona una sola imagen (de referencia) de la pila de imágenes previamente procesadas en la etapa 1, y en dicha imagen selecciona una región de interés (ROI) con el cursor.
- A continuación se procede a obtener los atributos, el sistema obtiene los atributos CBIR de la ROI, los cuales los disponemos en un vector de la forma: $V_R = [\mu_H, \sigma_H, H_H, \mu_S, \sigma_S, H_S, \mu_I, \sigma_I, H_I]_{9x1}$, de tamaño 9×1, ya que se tienen tres capas de la imagen y por cada capa se extraen los atributos: media, desviación estándar y homogeneidad.

Etapa 3

- En la imagen de referencia que el usuario seleccionó anteriormente se genera un emparrillado con dos casos: regular de 100 puntos, o aleatorio de 300 puntos distribuidos uniformemente.
- En cada punto generado se abre una ventana de tamaño 10×10 pixeles.

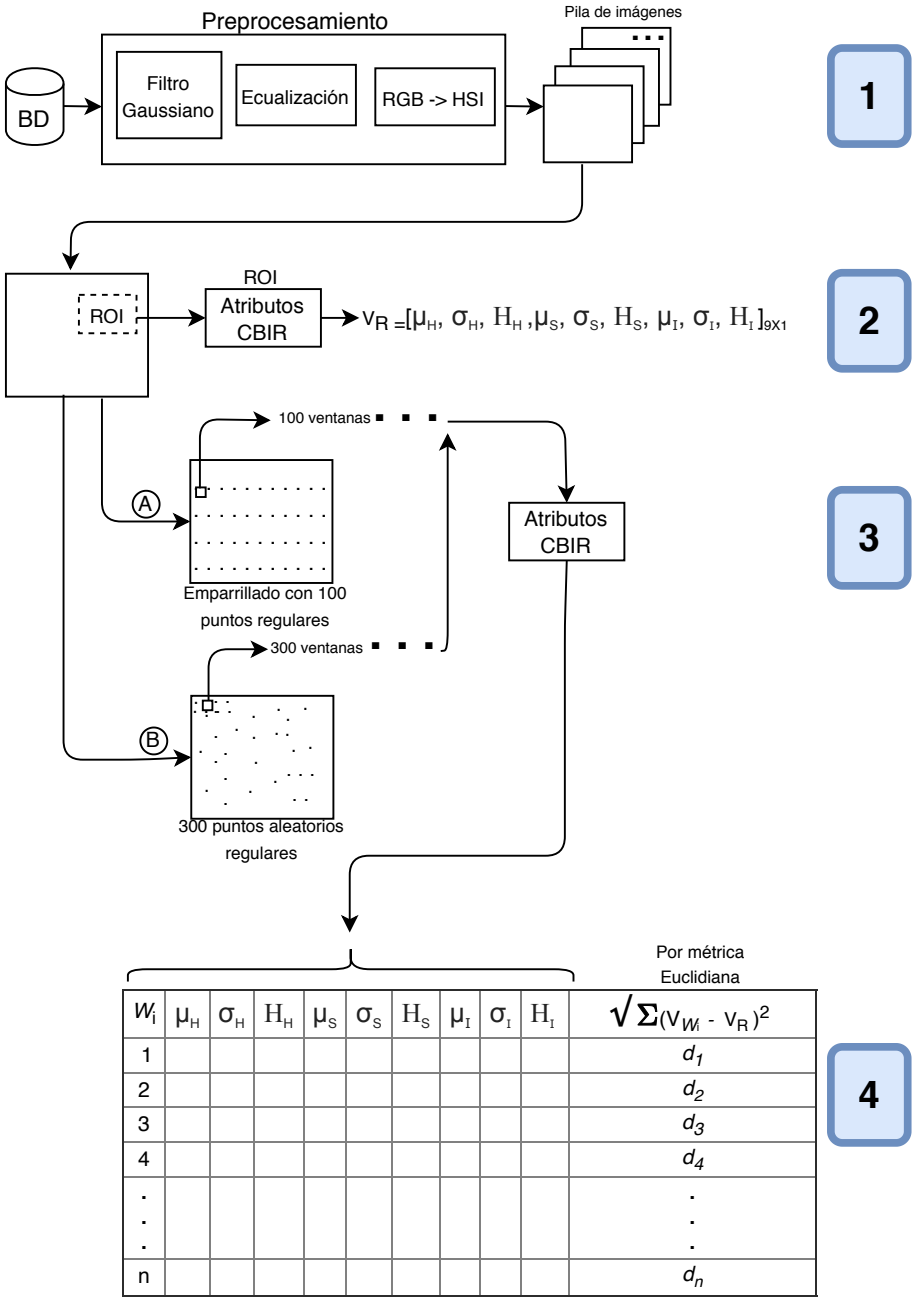


Fig. 2. Metodología.

– A cada una de las (100 o 300) ventanas se extraen sus atributos CBIR teniendo como resultado una tabla de tamaño (9×100, o 9×300), nueve de los atributos CBIR y (100 o 300) por cada ventana según el caso.

Etapa 4

– Aplicación de un clasificador de distancia K-NN, con $K = 3$, 5, 11 o 31 [5]. A cada vector de la tabla obtenida se le aplica la métrica euclidiana contra el vector V_R y se guardan esas distancias en dicha tabla para después aplicar un ordenamiento ascendente y tomar las 3, 5, 11 o 31 (según considere el especialista) regiones más parecidas a la ROI.
Buscamos un número de k-vecinos de cantidad impar, para lograr discriminar mejor acorde a la literatura reportada, y dada la naturaleza de la imagen local analizada, que puede replicarse en muchas otras subáreas de la imagen de estudio, es que se llega a considerar 11 u 31 vecinos cercanos.

4. Experimentos

Se uso una computadora tipo PC con:

– Procesador: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (12 CPUs), 2.2GHz.
– Sistema Operativo: MS Windows 10, 64 bits.
– Memoria RAM: 8 GB.
– Lenguaje: Python 3.8.1.
– Librería: Open CV 4.1.2.

Base de datos cuenta con un total de 312 imágenes, 190 de campo claro y 122 de fluorescencia, en formato TIF:

Tipo	Resolución
Campo Claro	1392×1040 pixeles
Fluorescencia	355×402 a 1280×960 pixeles

Las primeras pruebas que se hicieron son probando con dos formas de emparrillado la primera es un emparrillado regular de 100 puntos con subventanas de tamaño 10×10 pixeles y las segundas pruebas fueron con un emparrillado de 300 puntos aleatorios con subventanas de tamaño 10×10 y de 30×30 pixeles.

Se realizaron diferentes pruebas con las imágenes de campo claro y de fluorescencia, lo que se quiere realizar es tener una imagen de células gliales y ya sea seleccionando una ROI de manera manual o una de manera aleatoria buscar ese objeto de interés en la misma imagen y mostrar esos objetos parecidos.

Por lo cual se tienen dos tipos de pruebas una es de manera supervisada y otra de forma no supervisada. En la primera forma supervisada lo que se hace es que el médico especialista selecciona una ROI de forma manual y el da su punto de vista al decir si los objetos localizados son semejantes o no. Y la segunda forma de manera no supervisada, la ROI se selecciona de manera aleatoria en la imagen y se muestran los más parecidos.

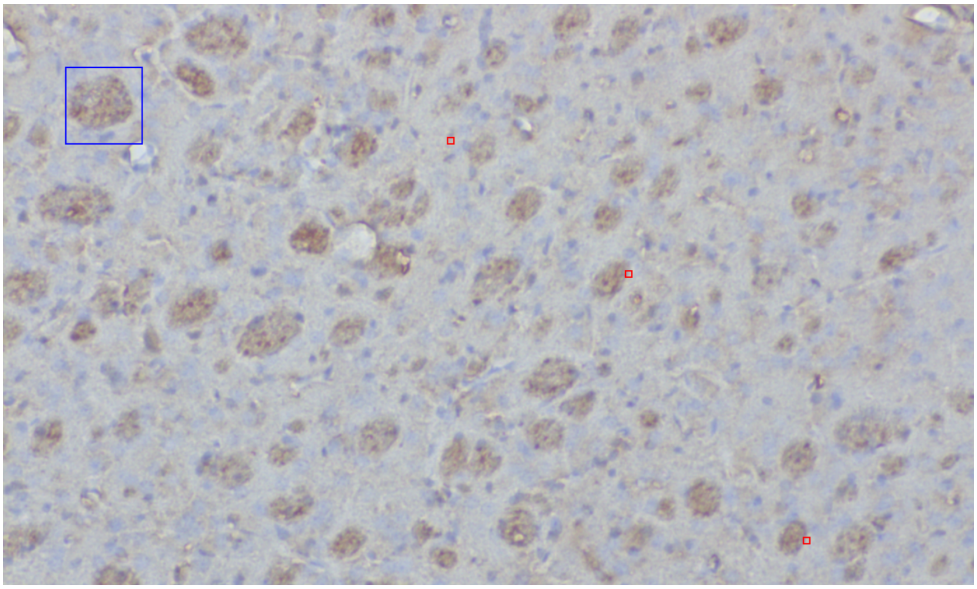


Fig. 3. Se muestra una imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 10x10 más parecidas, considerando un emparrillado regular de la imagen.

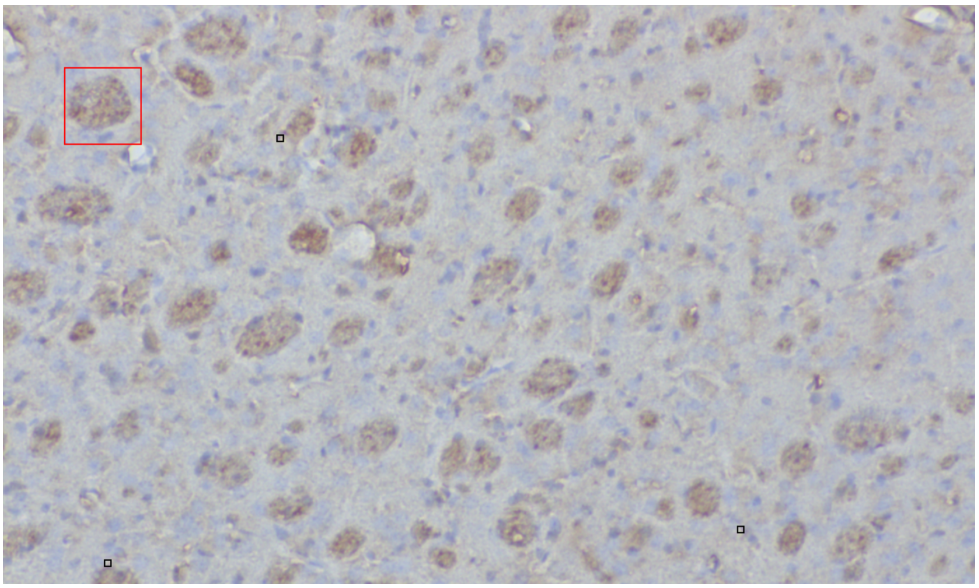


Fig. 4. Se muestra una imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 10×10 más parecidas, considerando un emparrillado aleatorio de la imagen.

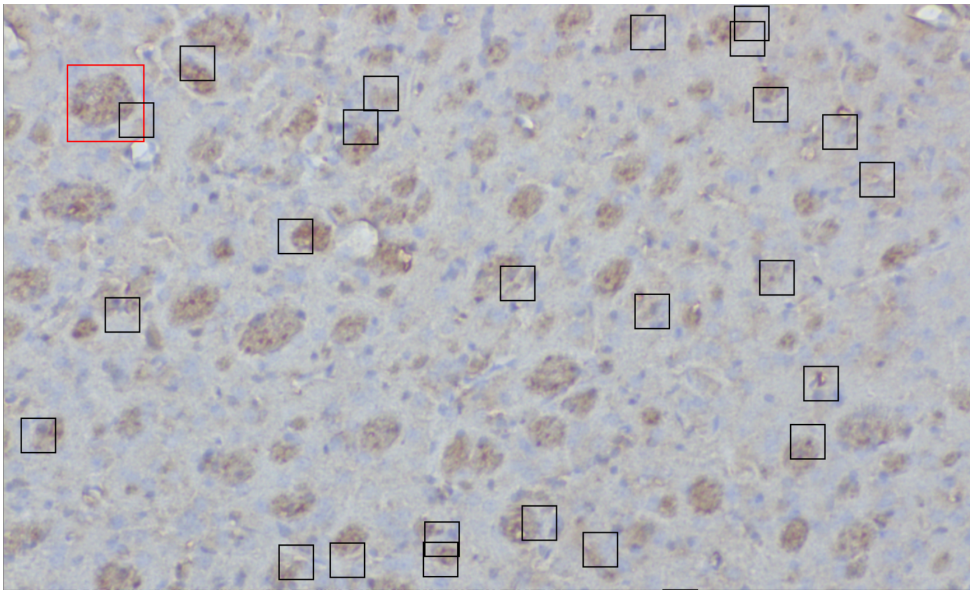


Fig. 5. Se muestra una imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 30×30 más parecidas, considerando un emparrillado aleatorio de la imagen.

4.1. Análisis de resultados

En los primeros experimentos que se realizaron se eligió un emparrillado de 100 puntos regulares en la imagen, se selecciona la ROI y a partir de eso se buscaban las 3 ROI que más se parecían a los atributos CBIR de la selección, como se muestra en la Figura 3.

Las ROI más parecidas se fijaron como ventanas de tamaño 10×10, pero al observar los resultados estas ventanas eran muy pequeñas para el tamaño de la imagen que aproximadamente es de 1040×1392.

El segundo experimento fue realizado con un emparrillado aleatorio de 300 puntos y de igual forma se obtuvieron las tres ROI que más se parecían con respecto a las características CBIR obtenidas de la región seleccionada, pero de igual forma las ventanas de tamaño 10×10 eran muy pequeñas ver Figura 4.

Por último, se ajustó el tamaño de las ventanas a un espacio de 30×30 pixeles y así fue más fácil de visualizar los resultados, como se muestra en la Figura 5.

Por el momento se está trabajando en tener una aplicación amigable para el profesional de la salud, y también estamos recabando la mayor cantidad de imágenes probadas en el laboratorio del Dr. Edgar, para validar nuestros resultados con su experiencia.

4.2. Criterio de pruebas semi-supervisadas automáticas

Con la finalidad de poder tener un criterio para analizar la posible efectividad de nuestra metodología, se partió de marcar o generar una ROI aleatoria sobre cada una de las imágenes de prueba, de tal forma que se tiene la suerte de que caiga o no encerrando o delimitando una región verdadera de interés, i.e. una célula tipo glía, tal como se observa en la Figura 6 la ROI seleccionada ya que es aleatoria selecciona más de un objeto y los objetos que se muestran son similares, pero en la Figura 7 la ROI seleccionada contiene muchos objetos, esto debido a que la imagen tiene más objetos o fue realizada con un acercamiento más detallado, por lo cual no podemos decir si los objetos que encuentra son o no son parecidos.

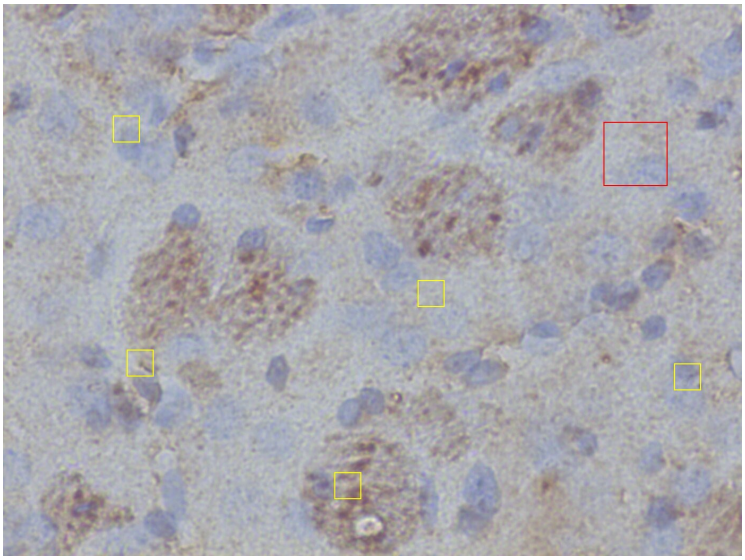


Fig. 6. Se muestra una imagen celular con la ROI seleccionada aleatoriamente y las cinco ROI de tamaño 50×50 más parecidas, considerando un emparrillado aleatorio de la imagen.

4.3. Discusión de resultados

El tamaño de las subventanas depende mucho del tamaño de las imágenes, ya que el tamaño que manejamos de las mismas es muy grande, una ventana de tamaño 10×10 pixeles es muy pequeña para visualizar los resultados. El tamaño de las ventanas para señalar los objetos parecidos al objeto señalado por el usuario en la ROI debería ser de un tamaño aproximado al tamaño del objeto de interés. Se puede ver en las muestras presentadas que por el momento

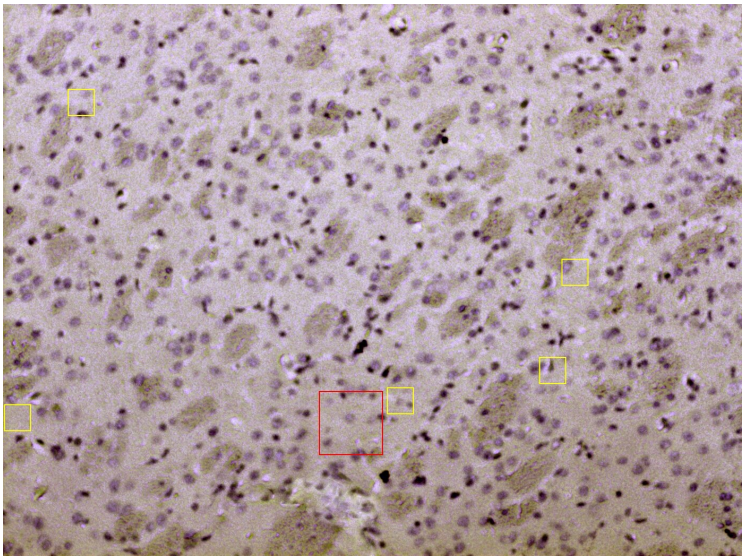


Fig. 7. Se muestra una imagen celular con la ROI seleccionada aleatoriamente y las cinco ROI de tamaño 50×50 más parecidas, considerando un emparrillado aleatorio de la imagen.

la ROI aleatoria para la batería de pruebas no precisamente cae encerrando una célula tipo glia, pero los resultados de las más parecidas aparentemente son satisfactorias para el profesional de la salud según su criterio de supervisión de las pruebas y su resultado.

Si el tamaño de la ROI después de aplicarle el preprocesamiento es de (n, m) se tomará el máximo de ellos por lo tanto el tamaño de las ventanas será de $(max(n, m) \times max(n, m))$, esto para que se busquen objetos de las mismas dimensiones.

5. Conclusiones

Hemos presentado una metodología para estudiar las células tipo glia en tejido cerebral a partir de sus atributos de textura, dentro de una imagen obtenida experimentalmente en microscopio, de tal forma que este enfoque no ha sido reportado en la literatura y este es nuestro principal aporte.

A partir de estos primeros resultados vemos que sí es posible poder brindar un apoyo para la localización de este tipo de células dentro de las imágenes sin importar si es de tipo campo claro o fluorescencia, ya que no impacta el color y sus variaciones dado que hacemos uso de la textura y no del color como atributos para conformar el patrón de caracterización de nuestro objeto de estudio. Se observa que con los tres atributos CBIR el algoritmo es capas de encontrar objetos parecidos al seleccionado por el usuario.

Como trabajo futuro se espera que en la ROI seleccionada por el usuario en ella se haga un preprocesamiento para solo encerrar el contorno deseado con un algoritmo de detección automática y así no se busquen regiones que no sean de nuestro interés, para esto se pretende aplicar un procesamiento morfológico y en base a esto se pretende tener una nueva capa de procesamiento automático de la imagen.

Referencias

1. Benavides, C., Villegas, J., Roman, G., Aviles, C.: Face classification by local texture analisys through cbir and surf points. IEEE Latin America Transactions 14(5), 2418–2424 (2016)
2. Benavides Alvarez, C.: Sistema no-supervisado de clasificación de rostros con técnicas basadas en CBIR. Universidad Autónoma Metropolitana, Unidad Iztapalapa (2015)
3. Bhamare, M.G., Patil, D.S.: Automatic Blood Cell Analysis By Using Digital Image Processing: A Preliminary Study. International Journal of Engineering Research & Technology (IJERT) 2(9), 3137–3141 (2013)
4. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Qian Huang, Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: the QBIC system. Computer 28(9), 23–32 (1995)
5. Fukunaga, K.: Introduction to statistical pattern recognition (2nd ed.). Academic Press Professional, Inc., San Diego, CA, USA (1990)
6. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall (2002)
7. Morales, R.O., Valentín, J., Ginori, L.: Cell Microscopy Imaging: A Review on Digital Image Processing Applications (2015)
8. Reyes H., D., Bulavina, L., Pivneva, T.: La glia el pegamento de las ideas. Ciencia pp. 12–18 (2014)
9. Saxena, S., Caroni, P.: Selective neuronal vulnerability in neurodegenerative diseases: From stressor thresholds to degeneration. Neuron (2011)
10. Subramaniam, S.: Selective neuronal death in neurodegenerative diseases: The ongoing mystery. Yale J. Biol. Med. (2019)
11. Verma, D., Gupta, S., Biswas, J., Joshi, N., Singh, A., Gupta, P., Tiwari, S., Sivarama, R.K., Chaturvedi, S., Wahajuddin, M.: New therapeutic activity of metabolic enhancer piracetam in treatment of neuro-degenerative disease: Participation of caspase independent death factors, oxidative stress, inflammatory responses and apoptosis. Biochim Biophys Acta Mol Basis Dis (2018)
12. Vogel, J., Schiele, B.: Semantic modeling of natural scenes for content-based image retrieval. International Journal of Computer Vision 72, 133–157 (2007)

ISSN 1870-4069

Implementación de una red neuronal profunda en tres etapas paralelas para el reconocimiento de actividades humanas e imágenes

Yafte A. Flores-Morales[1], Juan Villegas-Cortez[2], Graciela Román-Alonso[1], Arturo Zúñiga-López[3], César Benavides-Álvarez[1], Salomón Cordero-Sánchez[4]

[1] Universidad Autónoma Metropolitana, Unidad Iztapalapa, Departamento de Ingeniería Eléctrica, México
[2] Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Departamento de Sistemas, México
[3] Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Departamento de Electrónica, México
[4] Universidad Autónoma Metropolitana, Unidad Iztapalapa, Departamento de Química, México

{yafteaaron, scs, grac}@xanum.uam.mx, cesarbenavides32@gmail.com, {juanvc, azl}@azc.uam.mx

Resumen. La importancia de la aplicación del reconocimiento de patrones para el reconocimiento de imágenes, se hace notar cuando el número y la complejidad de las imágenes aumenta. Las redes neuronales artificiales han demostrado su efectividad para atacar problemas de patrones complejos en los últimos quince años aproximadamente y han evolucionado hacia arquitecturas híbridas y profundas. En este artículo presentamos la implementación de una red neuronal profunda en tres capas paralelas, con una alimentación paralela de los patrones a tres niveles de granularidad: fina, mediana y gruesa; buscando conformar un análisis del patrón característico robusto aplicado a imágenes. La red profunda de la cual partimos la hemos modificado y orientado, de patrones de reconocimiento de actividades humanas (HAR), hacia atacar la complejidad de las imágenes, con resultados prometedores del 99 % de reconocimiento en las pruebas realizadas.

Palabras clave: RNC, aprendizaje profundo, clasificación, reconocimiento de patrones, HAR, reconocimiento de imágenes.

Yafte A. Flores-Morales, Juan Villegas-Cortez, Graciela Román-Alonso, Arturo Zúñiga-López, et al.

Implementation of Three Parallel Layer Deep Neural Network for Human Activity and Image Recognition

Abstract. The recognition of patterns for the recognition of faces becomes an issue of extreme importance when the number and complexity of images increases. For the last fifteen years, artificial neural networks have shown its effectiveness to address complex pattern problems at the same time that they have evolved to hybrid and deep architectures. This paper presents the implementation of a deep neural network developed in three parallel layers, and with parallel feeding of patterns with three levels of granularity: fine, medium and thick. This results in conforming an analysis of the characteristic robust pattern applied in the images. The initial deep network was modified and oriented to address the complexity of the images by integrating human activity recognition (HAR). The results are promising achieving 99 % of recognition of the performed tests.

Keywords: Deep neural networks, deep learning, pattern recognition, HAR, image recognition.

1. Introducción

El área de reconocimiento de patrones en la inteligencia artificial (IA) está tomando mucha importancia en los últimos años debido al tratamiento automático de grandes cantidades de datos complejos. El reconocimiento de patrones (RP) extrae de un conjunto de datos la información que permite establecer propiedades, vínculos y relaciones entre los datos, estas relaciones como patrones permiten una descripción estructural o cuantitativa de un objeto o de alguna otra entidad de interés en los datos [10, 14].

La aplicación del RP a diferentes estructuras de datos, como señales o imágenes, ha dado lugar al desarrollo de una gran variedad de modelos, técnicas y algoritmos en las áreas de aprendizaje automático (del ingles Machine Learning o ML) y aprendizaje profundo (del ingles Deep Learning o DL), que permiten extraer patrones de las estructuras de datos con cierta eficiencia y eficacia. Algunas aplicaciones del reconocimiento de patrones son, por ejemplo, algoritmos sofisticados para reconocimiento de voz en los dispositivos móviles, o los videojuegos basados en realidad aumentada donde pueden detectar los movimientos del jugador por medio de una cámara. Sin embargo, reconocer movimientos de una persona no es tarea fácil. La tarea de reconocer y clasificar acciones de un grupo de individuos en el área de la IA, se le llama Reconocimiento de Actividad Humana (del ingles Human Action Recognition o HAR). En este artículo se propone implementar una red neuronal profunda con diferentes niveles de extracción de características, basado en la red neuronal grueso-a-fino[5] en [5] para la tarea HAR y el reconocimiento de patrones en imágenes.

[5] Coarse-fine N. del T.

La contribución de este trabajo es dos puntos: *(i)*, se busca evaluar la red neuronal basada en [5] con dos diferentes bases de datos WISDM v2.0 y HAPT, mismas que no se habían considerado previamente; y *(ii)*, se prueba el rendimiento de la arquitectura de la red neuronal en el reconocimiento de imágenes.

El documento está organizado de la siguiente manera: en la Sección 2 presentamos el estado del arte del problema de la caracterización de HAR y el reconocimiento de imágenes en grandes cantidades desde la perspectiva del reconocimiento de patrones y de visión por computadora; en la Sección 3 presentamos la implementación y adecuación de la metodología de las bases de datos escogidas para ejemplificar los alcances. La Sección 4 comparte los resultados de los experimentos y una discusión de los mismos y por ultimo, en la Sección 5, las conclusiones y perspectivas de trabajo futuro son compartidos.

2. Estado del arte

En la literatura existen dos enfoques principales para tratar la tarea HAR, el primer enfoque se basa en reconocer diferentes actividades por medio de cámaras de video, Jalal [16] obtiene buenos resultados con este enfoque, utilizando dispositivos de vídeo para reconocer actividades realizadas por un individuo. Sin embargo, el costo de procesar video reduce la eficiencia de los algoritmos por el alto consumo de memoria y cómputo, debido a esto muchos autores utilizan diferentes técnicas capaces de procesar el gran volumen de datos producidos por sensores de video, Martinez [22] utiliza visión por computadora y otros enfoques para el reconocimiento de actividades obteniendo resultados arriba del 90 %. El segundo enfoque se basa en la utilización de sensores como giroscopio y acelerómetro colocados en diferentes partes del individuo, trabajando con señales obtenidas de estos sensores. En esta propuesta nos centraremos en este segundo enfoque.

El área de investigación de HAR ha recibido atención debido a la tendencia creciente de sus aplicaciones en diferentes áreas, la reducción en el precio de los sensores integrados en los dispositivos portátiles y la masificación de su producción [18]. Diferentes artículos realizan estudios sobre características importantes de los sensores; tipo, costo, capacidad, cantidad. sin embargo, en la mayoría de estos se utilizan teléfonos inteligentes debido a la relación calidad-precio y la accesibilidad [27].

El HAR es un problema tratado por diferentes técnicas de la IA, el ML y el DL. Existen métodos basados en ML como máquinas de soporte vectorial o árboles de decisión, estos métodos utilizan características estadísticas como media, mínimo, máximo, desviación estándar, asimetría, curtosis, ángulos, entropía, etc. [2, 29].

Un enfoque diferente para la tarea de extracción de características se basa en DL donde se centra en la utilización de Redes Neuronales Convolucionales (del inglés Convolution Neural Network o CNN), un tipo particular de las redes neuronales artificiales caracterizada por el uso de la convolución entre señales o sucesiones de valores discretos.

Existen diferentes tipos de redes neuronales artificiales, la CNN utiliza como operación principal la convolución para la extracción de características de los datos, estas características representan rasgos particulares de los datos.

La importancia de las CNN radica en el tratamiento de dos puntos principales: *(i)* La extracción de características más robustas a partir de la implementación de la operación de convolución, y *(ii)* la posibilidad de trabajar con una mayor cantidad de patrones complejos en longitud o cantidad de atributos, así como el manejo de valores en punto flotante.

Existen diversos conjuntos de datos para el HAR que ayudan a medir el rendimiento de los modelos. En este artículo se ocuparon cuatro conjuntos de datos: UCI HAR [3], WISDM v1.1 [17],WISDM v2 [17, 32] y HAPT [25].

2.1. Métodos basados en ML

UCI HAR, Anguitia [4] mediante máquinas de soporte vectorial logra atacar el problema HAR obteniendo buenos resultados, Xiangbin Zhu [35] utiliza cadenas de markov. UCI HAPT, Taufeeq [28] obtiene excelentes resultados utilizando arboles de decisión. Zheng [34] utiliza máquinas de soporte vectorial combinado un modelo de agrupación dispersa de dos capas. WIDSM v1.1 Kishor Walse [30] utiliza árboles de decisión. Cagatay Catal [6] realiza una combinación de diferentes técnicas, árboles de decisión, regresión logística y perceptrón multicapa. WIDSM v2, Majid Ali Khan Quaid [23] por medio de un algoritmo genético logra alcanzar buenos resultados.

2.2. Métodos basados en DL

HAR, Avilés Cruz *et al.* [5] obtienen el 100 % utilizando una CNN con 3 diferentes niveles de alimentación paralela. Cho *et al.* [7] utilizaron el paradigma de divide y vencerás y una CNN para identificar acciones realizadas por humanos. Yong Zhang *et al.* [33] construyeron una CNN capaz de adaptarse a diferentes longitudes de datos. En HAPT, Yong Zhang *et al.* [33], el único trabajo encontrado con enfoque DL, construyeron una CNN capaz de adaptarse a diferentes longitudes de datos.

WIDSM v1.1 Avilés Cruz *et al.* [5] obtienen el 100 % utilizando una CNN con 3 diferentes niveles de alimentación paralela. Xinxin Han [13] utiliza una CNN con tres niveles de alimentación. Daniele Ravi [24] crea una CNN tradicional probando diferentes capas de convolución. WIDSM v2 Girmaw Abebe [1] obtiene un buen resultado combinando una CCN con memorias de largo plazo (LSTM). Ignatov [15] utiliza una CNN tradicional con métodos de preprocesamiento. MNIST, Li Wan [31] por medio de una CNN y un enfoque Dropconnect obtiene un buen resultado. Dan Claudin Ciresan [8] construye una red neuronal multicolumna. Ming Liang [21] por medio de una red neuronal recurrente logra mejores resultados que una CNN clásica.

En las Tablas 1 a la 5 se presentan los trabajos mencionados de las diferentes técnicas de ML y DL.

Tabla 1. Comparación de exactitud de UCI HAR [3].

Nombre	Año	Exactitud %	Método
Avilés Cruz [5]	2019	100.0	CNN
Yong Zhang [33]	2018	98.4	U-CNN
San-Segundo [26]	2016	98.0	Marcov+ Estadistica
Andrey Ignatov [15]	2018	97.63	CNN

Tabla 2. Comparación de exactitud de HAPT [25].

Nombre	Año	Exactitud %	Método
Taufeeq [28]	2016	100.0	Random Forrest
Zheng [34]	2018	96.26	TASG+SVM
Zheng [34]	2018	95.83	TASG+RNN
Yong Zhang [33]	2018	93.1	U-CNN

Tabla 3. Comparación de exactitud de WISDM 1.1v [17].

Nombre	Año	Exactitud %	Método
Avilés Cruz [5]	2019	100	CNN
Xinxin Han [13]	2020	98.83	CNN+FusionTrial
Daniel Ravi [24]	2017	98.2	CNN
kishor walse [30]	2016	98.09	Ramdom forrest
Yong Zhang [33]	2018	97.0	U-CNN

Tabla 4. Comparación de exactitud de WISDM 2.0v [17] [32].

Nombre	Año	Exactitud %	Método
Girmaw Abebe [1]	2017	97.9	CNN+LSTM+Handcrafted
Majid Ali Khan Quaid [23]	2019	94.02	Genético
andrey Ignatov [15]	2018	93.32	CNN + tamaño 200
Daniele Ravi [24]	2017	92.7	CNN

3. Propuesta

La propuesta es implementar una CNN con diferentes niveles de extracción paralela de características, estos niveles de extracción están divididos en grueso, medio y fino. En la Figura 1 se muestra la arquitectura de la red neuronal [5], la idea principal está basada en extraer diferentes niveles de características, donde el nivel fino extrae características más sutiles de los datos, el nivel medio extrae

Tabla 5. Comparación de exactitud de MNIST [19].

Nombre	Año	Exactitud %	Método
Li Wan [31]	2013	99.79	CNN Dropconnect
Dan Claudin Ciresan [8]	2012	99.77	CNN Multi-Column
Benjamin Graham [12]	2014	99.68	CNN Fraccional
Dan Claudin Ciresan [9]	2010	99.65	Big Deep CNN

características más robustas y el nivel grueso extrae características más burdas como lineas, círculos, colores, etc.

La arquitectura presentada en la Figura 1 consta de tres niveles:

1. **Fino:** Está compuesto por cuatro capas de convolución, cada capa de convolución es seguida por una capa de agrupación máxima, los mapas de características son reducidos en cada capa. Las primeras dos capas de convolución cuentan con 18 filtros y las ultimas dos capas cuentan con 36 filtros, el tamaño del filtro es de (1×2) para todas las capas de convolución. La ventana de agrupación máxima es de (1×2) y el paso es de 2.
2. **Medio:** Está compuesto por dos capas de convolución, cada capa de convolución es seguida de una capa de agrupación máxima, los mapas de características son reducidos en cada capa. La primera capa de convolución cuenta con 18 filtros y la ultima cuenta con 36 filtros, el tamaño del filtro es de (1×2) para todas las capas de convolución. La ventana de agrupación máxima es de (1×4) y el paso es de 4.
3. **Grueso:** Está compuesto por una capa de convolución, la capa de convolución es seguida por una capa de agrupación máxima. la capa de convolución cuenta con 36 filtros de tamaño (1×2). La ventana de agrupación máxima es de (1×16) y el paso es de 16.

La salida de los tres niveles se agrupan y se aplanan, formando un vector que pasa a una etapa de clasificación por medio de una red neuronal totalmente conectada. Finalmente la salida de la capa totalmente conectada se pasa a una capa softmax que calcula la distribución de probabilidad de las clases [11]. En la Tabla 6 se muestran los parámetros utilizados por la red neuronal propuesta.

4. Implementación

La implementación se realizó en una computadora tipo Workstation con las siguientes características: sistema operativo Linux, 8 GB RAM, procesador Intel Core i7, GPU NVIDIA GTX-1050. Esta sección describe y muestra las pruebas realizadas de la red neuronal profunda gruesa-fina usando diferentes conjuntos de datos, como se detallan a continuación en su generalidad.

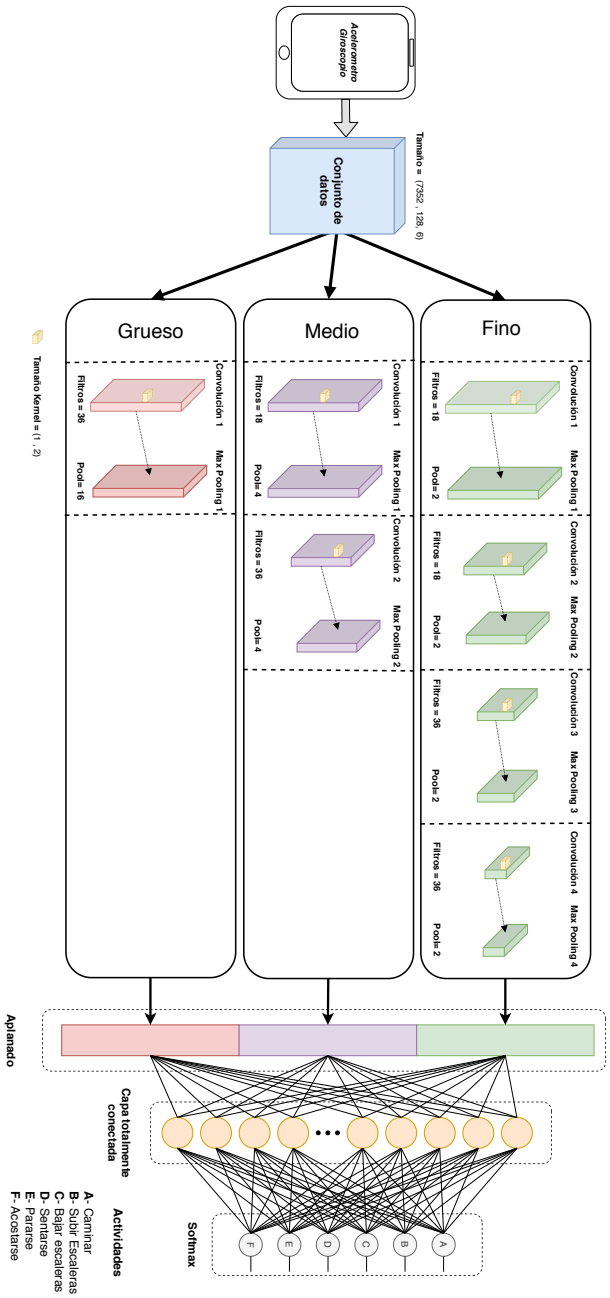


Fig. 1. Arquitectura de la red neuronal propuesta aplicada al reconocimiento de actividad humana [5].

Tabla 6. Parámetros establecidos para la red neuronal profunda gruesa-fina.

Parámetros	Valor	Tipo
Número de épocas	500	Entero
Tamaño de lote	300	Entero
Dropout	0.8	Flotante
Función activación	RELU	Función
Función perdida	Entropía cruzada	Función
Numero de filtros	18,36	Entero
Dimensión filtro (fila)	1	Entero
Dimensión filtro (columna)	2	Entero
Dimensión maxpooling (fila)	1	Entero
Dimensión maxpooling (columna)	2,4,16	Entero
Paso de maxpooling	2,4,16	Entero
Optimizador	Adadelta	Función

4.1. Conjuntos de datos

En los experimentos se utilizaron dos tipos de conjunto de datos: 1. Reconocimiento de actividad humana; UCI HAR [3], WISDM v1.1 [17], WISDM v2 [17] [32] y HAPT [25]. 2. Reconocimiento de imágenes; MNIST [19].

UCI HAR es un conjunto de señales preprocesadas de seis distintas actividades humanas: caminar, subir escaleras, bajar escaleras, sentarse, pararse, acostarse. Estas señales fueron tomadas con un teléfono inteligente colocado en la cintura de treinta voluntarios, se tomaron muestras del giroscopio y acelerómetro integrados en el teléfono con una velocidad de 50 Hz. Las señales del giroscopio y del acelerómetro están compuestas por tres ejes, (X, Y, Z). En la Tabla 7 se muestra la cantidad de instancias por clase.

Tabla 7. Número de instancias por clase del conjunto de datos HAR.

Actividad	Entrenamiento	Porcentaje	Prueba	Porcentaje
Bajar escaleras	986	13.4 %	420	14.25 %
Subir escaleras	1,073	14.6 %	471	15.98 %
Caminar	1,226	16.7 %	496	16.83 %
Sentarse	1,286	17.5 %	491	16.66 %
Pararse	1,374	18.7 %	532	18.05 %
Acostarse	1,407	19.1 %	537	18.22 %
Total	7,352	100 %	2,947	100 %

Human Activities and Postural Transitions (HAPT) es una actualización de HAR, tiene doce clases: caminar, subir escaleras, bajar escaleras, sentarse, pararse, acostarse, pararse a sentarse, sentarse a pararse, sentarse a acostarse, acostarse a sentarse, pararse a acostarse, acostarse a pararse. HAPT proporciona un conjunto de datos no procesados, se realizó un preprocesamiento aplicando el método de ventana deslizante con empalme de 50 % y tamaño de ventana de 128

muestras equivalente a 6.4 segundos, se aplicó un filtro Butterworth de tercer orden con frecuencia de corte de 10 Hz. En la Tabla 8 se muestra la cantidad de instancias por clase del conjunto de entrenamiento y del conjunto prueba.

Tabla 8. Número de instancias por clase del conjunto de datos HAPT procesada y particionada.

Actividad	Procesada	Entrenamiento	Prueba
Sentarse-Pararse	123	93	30
Pararse-Sentarse	155	108	47
Acostarse-Sentarse	169	111	59
Acostarse-Pararse	170	118	51
Sentarse-Acostarse	195	142	53
Pararse-Acostarse	223	157	66
Bajar escaleras	1,691	1,202	489
Subir escaleras	1,817	1,227	590
Caminar	1,905	1,314	591
Sentarse	1,983	1,421	562
Acostarse	2,144	1,503	641
Pararse	2,167	1,523	644
Total	12,742	8,919	3,823

Wireless Sensor Data Mining (WISDM) v1.1, contiene $1,098,204$ señales no preprocesadas de seis diferentes actividades humanas: caminar, subir escaleras, bajar escaleras, sentarse, pararse, trotar. Estas señales fueron tomadas con un teléfono inteligente que midió la aceleración en tres ejes (X,Y,Z) con una frecuencia de 20 Hz. Para el preprocesamiento de las señales se utilizó el método de ventana deslizante con empalme de 50 % y un tamaño de ventana de 128 muestras equivalente a $6.4s$, se aplicó un filtro de Butterworth de tercer orden con frecuencia de corte de 10 Hz. La Tabla 9 muestra la cantidad de instancias.

Tabla 9. Número de instancias por clase del conjunto de datos WISDM v1.1.

Actividad	Sin procesamiento	Procesada	Entrenamiento	Prueba
Pararse	48,395	757	544	213
Sentarse	59,939	936	650	286
Bajar escaleras	100,427	1,565	1,089	476
Subir escaleras	122,869	1,927	1,327	600
Trotar	342,176	5,346	3,737	1,609
Caminar	424,398	6,627	4,663	1,964
Total	1,098,204	17,158	12,010	5,148

WISDM v2.0 es una actualización del conjunto de datos WISDM v1.1, es un conjunto de $2,980,765$ señales no preprocesadas de seis diferentes actividades humanas: caminar, escaleras, sentarse, pararse, trotar, acostarse. Estas señales fueron tomadas con un teléfono inteligente el cual midió la aceleración en tres ejes (X,Y,Z) con una frecuencia de 20 Hz. Se utilizo el mismo preprocesamiento de las señales aplicado en WISDM v1.1. La Tabla 10 muestra la distribución de estas instancias por clase. La Tabla 11 compara las clases contenidas en cada conjunto de datos.

Tabla 10. Número de instancias por clase del conjunto de datos WISDM v2.0 procesada y particionada.

Actividad	Sin procesar	Procesada	Entrenamiento	Prueba
Escaleras	57,425	896	644	252
Acostarse	275,967	4,315	2,972	1,343
Pararse	288,873	4,512	3,141	1,371
Trotar	438,871	6,859	4,833	2,026
Sentarse	663,706	10,373	7,336	3,037
Caminar	1,255,92	19,618	13,675	5,943
Total	2,980,765	46,573	32,601	13,972

Tabla 11. Clases contenidas en los conjuntos de datos.

Dataset	HAPT	UCI HAR	WISDM v1.1	WISDM v2.0
Pararse-Sentarse	*			
Sentarse-Pararse	*			
Sentarse-Acostarse	*			
Acostarse-Sentarse	*			
Pararse-Acostarse	*			
Acostarse-Pararse	*			
Subir escaleras	*	*	*	
Bajar escaleras	*	*	*	
Acostarse	*	*		*
Caminar	*	*	*	*
Sentarse	*	*	*	*
Pararse	*	*	*	*
Trotar			*	*
Escaleras				*
Total clase	12	6	6	6

Modified National Institute of Standards and Technology (MNIST) [20], es una base de datos de números escritos a mano, consta de 10 clases que representan los números del 0 al 9. Los dígitos están representadas por medio de imágenes en escala de grises con un tamaño de 28×28 píxeles. En la Tabla 12 se muestra la cantidad de instancias por clase.

Tabla 12. Número de instancias por clase de la base de datos MNIST.

Dígito	Entrenamiento	Prueba
Cero	5,923	980
Uno	6,742	1,135
Dos	5,958	1,032
Tres	6,131	1,010
Cuatro	5,842	982
Cinco	5,421	892
Seis	5,918	958
Siete	6,265	1,028
Ocho	5,851	974
Nueve	5,949	1,009
Total	60,000	10,000

4.2. Análisis de resultados

El análisis de resultados se llevó a cabo por medio de dos etapas; entrenamiento y prueba. La primera consiste en entrenar la red neuronal propuesta con el conjunto de entrenamiento. Por otro lado, la segunda consiste en evaluar la red neuronal por medio del conjunto de prueba, el cual es ajeno al conjunto de entrenamiento. Para medir el rendimiento de la red neuronal gruesa-fina se utilizaron métricas de exactitud.

Resultados del conjunto de datos UCI HAR: En la Figura 2a se muestra la evaluación de los conjuntos de entrenamiento y prueba. Se alcanzó el 100 % de recuperación y el parámetro de perdida alcanzó el valor 0 % al evaluar el conjunto de prueba. El tiempo de entrenamiento de la red neuronal con el conjunto de entrenamiento fue de 4.83 minutos.

Resultados del conjunto de datos HAPT: HAPT resulta un conjunto de datos difícil, ya que no cuenta con un equilibrio en el número de instancias por clase (como se ve en la Tabla 8), sin embargo, se alcanzó una exactitud del 95.27 % con una perdida de 4.73 % al evaluar el conjunto de prueba. En la Figura 2b se muestra la evaluación de los conjuntos de entrenamiento y prueba. El tiempo de entrenamiento de la red neuronal con el conjunto de entrenamiento fue de 5.30 minutos.

Resultados del conjunto de datos WISDM v1.1: Se alcanzó una recuperación de 98.12 % y valor de perdida de 1.88 % al evaluar el conjunto de prueba.

En la Figura 2c se muestra la evaluación de los conjuntos de entrenamiento y prueba. El tiempo de entrenamiento de la red neuronal con el conjunto de entrenamiento fue de 7.06 minutos.

Resultados del conjunto de datos WISDM v2.0: WISDM v2.0 es una versión más grande que la v1.1, esto aumenta la dificultad. Los resultados obtenidos con el conjunto de datos WISDM v2.0 alcanzaron un valor de recuperación de 94.75 % y un valor de pérdida de 5.21 % al evaluar el conjunto de prueba. En la Figura 2d se muestra la evaluación de los conjuntos de entrenamiento y prueba. El tiempo de entrenamiento de la red neuronal con el conjunto de entrenamiento fue de 14.85 minutos.

Resultados del conjunto de datos MNIST: MNIST es una base de datos muy popular para la evaluación de modelos. Los resultados obtenidos alcanzaron un valor de exactitud de 99.47 % y un porcentaje de error de 0.53 % al evaluar el conjunto de prueba. La Figura 2e muestra la evaluación del conjunto de entrenamiento y del conjunto de prueba. El tiempo de entrenamiento de la red neuronal con el conjunto de entrenamiento y con 30 etapas fue de 30.75 minutos.

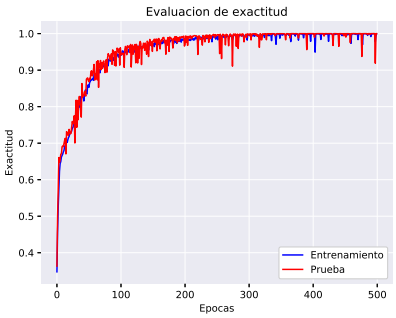
La Tabla 13 muestra una comparación de los resultados obtenidos con otros resultados en el estado del arte.

Tabla 13. Comparación de exactitud de diferentes técnicas del estado del arte

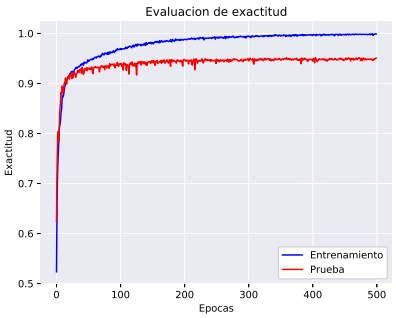
Base Datos	**Propuesta**	Avilés [5]	Zhang [33]	Ravi [24]	Abebe [1]	Taufeeq [28]	Li Wan [31]
UCI HAR	**100**	100	98.4	-	-	-	-
HAPT	**95.27**	-	93.1	-	-	100	-
WISD v1.1	**98.12**	100	97.0	98.6	-	-	-
WISD v2.0	**94.75**	-	-	92.7	97.9	-	-
MNIST	**99.47**	-	-	-	-	-	99.79

5. Conclusiones

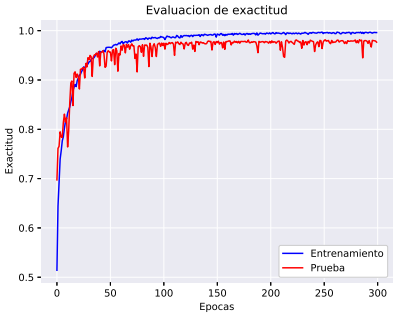
En este artículo se presentó una nueva implementación de la propuesta de una red neuronal profunda previamente reportada en la literatura para aplicarse sólo para reconocimiento y clasificación de patrones HAR, tal que la estrategia se basa en la extracción paralela en tres etapas: fina, media y gruesa. Sin embargo, es necesario una mejora en la optimización de los parámetros, ya que como hemos presentado en la implementación para imágenes, como meta más ambiciosa y de mayor alcance para aplicaciones en visión por computadora, es necesario poder hacerle modificaciones que mejoren en rapidez y capacidad de extracción de características propias para imágenes como entidades numéricas más complejas. La implementación aquí presentada está ubicada en el nivel 25 del *Ranking*



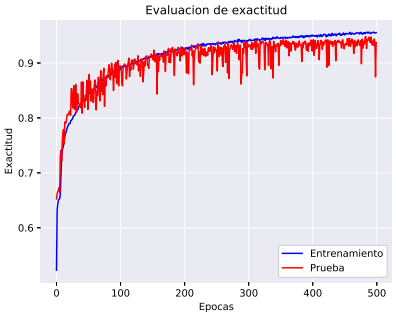
(a) UCI HAR: Exactitud de 100 %



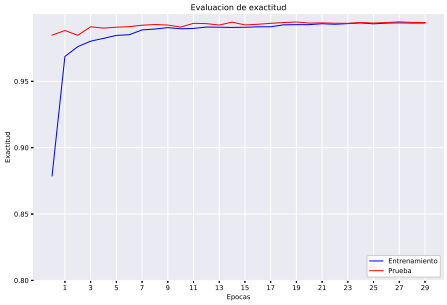
(b) HAPT: Exactitud de 95.27 %



(c) WISDM v1.1: Exactitud de 98.12 %



(d) WISDM v2.0: Exactitud de 94.75



(e) MNIST: Exactitud de 99.47 %

Fig. 2. Evaluación de la red neuronal propuesta con las bases de datos seleccionadas.

classification datasets results[6], esta página se tomó como referencia para la

[6] Ranking classification datasets results: `https://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html`

comparación de resultados de diferentes técnicas y enfoques del conjunto de datos MNIST, ya que para aplicaciones prácticas es factible su uso, pero consideramos que es posible su mejora si realizamos, como trabajo futuro, modificaciones a la capa de extracciones paralelas, buscando robustecer a los patrones orientados a problemas de imágenes más complejas.

Si bien la base de datos MNIST es una buena aproximación para validar posibles aplicaciones reales, existen problemas de imágenes más complejas como lo son el rostro humano y los escenarios naturales, donde en un futuro trabajo podremos mostrar los avances en tales campos.

Referencias

1. Abebe, G., Cavallaro, A.: Inertial-vision: Cross-domain knowledge transfer for wearable sensors. pp. 1392–1400 (10 2017)
2. Anguita, D., Ghio, A.: A public domain dataset for human activity recognition using smartphones. 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (April 2013)
3. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.: A public domain dataset for human activity recognition using smartphones. pp. 24–26. Computational Intelligence and Machine Learning (01 2013)
4. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J.: A public domain dataset for human activity recognition using smartphones. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2013)
5. Avilés-Cruz, C., Ramírez, A., Zúñiga López, A., Villegas Cortez, J.: Coarse-fine convolutional deep-learning strategy for human activity recognition. Sensors 2019 (03 2019)
6. Catal, C., Tufekci, S., Pirmit, E., Kocabag, G.: On the use of ensemble of classifiers for accelerometer-based activity recognition. Applied Soft Computing 46 (01 2015)
7. Cho, H., Yoon, S.: Divide and conquer-based 1D CNN human activity recognition using test data sharpening. Sensors 1055 (06 2017)
8. Cireşan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (02 2012)
9. Cireşan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Deep, big, simple neural nets for handwritten digit recognition. Neural Computation 22(12), 3207–3220 (2010), pMID: 20858131
10. Dutt, V., Chaudhry, V., Khan, I.: Pattern recognition: an overview. American Journal of Intelligent Systems 2, 23–27 (01 2012)
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press (2016)
12. Graham, B.: Fractional max-pooling (12 2014)
13. HAN, X., YE, J., LUO, J., ZHOU, H.: The effect of axis-wise triaxial acceleration data fusion in cnn-based human activity recognition. IEICE Transactions on Information and Systems E103.D(4), 813–824 (2020)
14. Hancock, E., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.: Advances in pattern recognition methodology and applications. Pattern Recognition Letters 34, 359–360 (03 2013)

15. Ignatov, A.: Real-time human activity recognition from accelerometer data using convolutional neural networks. Applied soft computing 62, 915–922 (9 2017)
16. Jalal, A., Kim, Y.H., Kim, Y.J., Kamal, S., Kim, D.: Robust human activity recognition from depth video using spatiotemporal multi-fused features. Pattern Recognition 61 (08 2016)
17. Kwapisz, J., Weiss, G., Moore, S.: Activity recognition using cell phone accelerometers. SIGKDD Explorations 12, 74–82 (11 2010)
18. Lane, N., Miluzzo, E., lu, H., Peebles, D., Choudhury, T., Campbell, A.: A survey of mobile phone sensing. ieee commun mag. Communications Magazine, IEEE 48, 140 – 150 (10 2010)
19. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278 – 2324 (12 1998)
20. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. ATT Labs 2 (2010)
21. Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. pp. 3367–3375 (06 2015)
22. Martinez, F., González-Fraga, J., Cuevas-Tello, J.C., Rodriguez, M.: Activity inference for ambient intelligence through handling artifacts in a healthcare environment. Sensors (Basel, Switzerland) 12, 1072–99 (12 2012)
23. Quaid, M.A., Jalal, A.: Wearable sensors based human behavioral pattern recognition using statistical features and reweighted genetic algorithm. Multimedia Tools and Applications pp. 1–23 (12 2019)
24. Ravì, D., Wong, C., Lo, B., Yang, G.Z.: A deep learning approach to on-node sensor data analytics for mobile or wearable devices. IEEE journal of biomedical and health informatics PP (12 2016)
25. Reyes-Ortiz, J., Oneto, L., Ghio, A., Anguita, D., Parra, X.: Human activity recognition on smartphones with awareness of basic activities and postural transitions (01 2014)
26. San-Segundo, R., Lorenzo-Trueba, J., Martínez-González, B., Pardo, J.: Segmenting human activities based on hmms using smartphone inertial sensors. Pervasive and Mobile Computing 30 (01 2016)
27. Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T., Kjærgaard, M., Dey, A., Sonne, T., Jensen, M.: Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. pp. 127–140 (11 2015)
28. Uddin, M.T., Billah, M.M., Hossain, M.F.: Random forests based recognition of human activities and postural transitions on smartphone. In: 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). pp. 250–255 (2016)
29. Voicu, R.A., Dobre, C., Bajenaru, L., Ciobanu, R.I.: Human physical activity recognition using smartphone sensors. Sensors 19, 458 (01 2019)
30. Walse, K., Dharaskar, R., Thakare, V.M.: Performance evaluation of classifiers on wisdm dataset for human activity recognition (03 2016)
31. Wan, L., Zeiler, M., Zhang, S., Lecun, Y., Fergus, R.: Regularization of neural networks using dropconnect (01 2013)
32. Weiss, G., Lockhart, J.: The impact of personalization on smartphone-based activity recognition. AAAI Workshop - Technical Report (01 2012)
33. Zhang, Y., Zhang, Y., Zhang, Z., Bao, J., Song, Y.: Human activity recognition based on time series analysis using u-net (09 2018)
34. Zheng, Z., Du, J., Sun, L., Huo, M., Chen, Y.: Tasg: An augmented classification method for impersonal har. Mobile Information Systems 2018, 1–10 (12 2018)
35. Zhu, X., Qiu, H.: High accuracy human activity recognition based on sparse locality preserving projections. PLOS ONE 11(11), 1–18 (11 2016)

ISSN 1870-4069

Red neuronal convolucional de baja latencia para la estimación de profundidad monocular

Edgar Rodrigo López Silva, Luis Rogelio Román Rivera,
Jesús Carlos Pedraza Ortega, Marco Antonio Aceves Fernández,
Juan Manuel Ramos Arreguín

Universidad Autónoma de Querétaro,
Facultad de Ingeniería,
México

edgar.lopez06@outlook.com,
luis.rogelio.roman.rivera@gmail.com, caryoko@yahoo.com,
marco.aceves@gmail.com, jsistdig@yahoo.com.mx

Resumen. La estimación de profundidad monocular se ha convertido recientemente en una de las técnicas emergentes y de creciente interés para la obtención de la estructura espacial de una escena. En gran parte, los recientes avances en el área de aprendizaje profundo han permitido obtener resultados prometedores en esta línea de investigación. No obstante, los métodos del estado del arte se caracterizan por utilizar redes neuronales convolucionales muy complejas cuya etapa de inferencia no puede llevarse a cabo en tiempo real en sistemas con recursos computacionales limitados. Desde esta perspectiva, pocos trabajos se han propuesto para la estimación de profundidad monocular mediante arquitecturas de baja latencia. En el presente trabajo, se propone una arquitectura de red neuronal convolucional basada en el extractor de características ShuffleNet V2. Los resultados obtenidos en el conjunto de datos NYU Depth V2 muestran que la arquitectura propuesta es capaz de reducir el tiempo de inferencia en un 37.5% con respecto a métodos relacionados del estado del arte.

Palabras clave: Baja latencia, profundidad, monocular, ShuffleNet V2.

Low-latency Convolutional Neural Network for Monocular Depth Estimation

Abstract. Monocular depth estimation has recently become one of the emerging and increasingly interesting techniques for obtaining the spatial structure of a scene. To a large extent, recent advances in the area of deep learning have allowed promising results in this research area. However, state-of-the-art methods stand out for using highly complex convolutional neural networks whose inference stage cannot be deployed in real-time on systems with limited computational resources. From this perspective, few studies have been proposed to tackle the monocular depth estimation problem using low-latency architectures. In this paper, a convolutional neural network architecture based on the ShuffleNet V2 feature extractor is proposed. The results on the NYU Depth

V2 dataset show that the proposed architecture is able to reduce the inference time by 37.5% compared to related state-of-the-art methods.

Keywords: Low-latency, depth, monocular, ShuffleNet V2.

1. Introducción

La estimación de profundidad se considera como uno de los problemas fundamentales en el campo de la visión por computadora al tratarse de una función imprescindible para la realización de tareas tales como la localización, mapeo, detección de objetos 3D y aplicaciones de realidad aumentada [1]. En general, la estimación de profundidad se refiere al conjunto de técnicas y algoritmos diseñados con el propósito de obtener una representación de la estructura espacial de una escena. En otras palabras, el objetivo primordial de los métodos para la estimación de profundidad es obtener una representación de la distancia absoluta o relativa desde el sensor 3D a cada punto de la escena de interés [2].

La estimación de profundidad a partir de imágenes digitales se ha basado principalmente en un enfoque de visión estereoscópica, que requiere de un par de imágenes capturadas de la misma escena desde distintos ángulos a fin de triangular la posición 3D de cada pixel de las imágenes [2]. No obstante, en los últimos años, ha habido un creciente interés en la estimación de profundidad monocular relativa, es decir, utilizando una sola imagen RGB. Esto último se refiere a un problema inherentemente ambiguo, ya que no existe ninguna correlación entre la saturación de color y la profundidad relativa asociada a cada pixel en una imagen RGB [3].

Por un lado, el creciente interés en la estimación de profundidad monocular relativa se debe a la alta eficiencia energética, el tamaño reducido y el bajo costo de las cámaras monoculares. Por otro lado, los recientes avances en el campo del aprendizaje automático (en inglés, machine learning) han mostrado que es posible obtener mapas de profundidad a nivel de pixel a partir de una sola imagen, factor que también ha incrementado el interés por parte de la comunidad científica [1].

Desde la perspectiva del aprendizaje automático, se ha buscado resolver el problema de estimación de profundidad monocular por medio de técnicas tradicionales y de aprendizaje profundo (en inglés, deep learning). Los trabajos iniciales en esta línea de investigación propusieron métodos clásicos para la extracción manual de características de la imagen de entrada y modelos de Markov para la predicción del mapa de profundidad. Sin embargo, no fue hasta la implementación de técnicas de aprendizaje profundo cuando se lograron avances significativos [1]. Para tal efecto, las redes neuronales convolucionales (CNN por sus siglas en inglés) de tipo autoencoder se han convertido en el enfoque estándar para la resolución del problema en cuestión. En este sentido, el problema de estimación de profundidad monocular puede considerarse como un problema de regresión continua a nivel de pixeles que puede formularse de la siguiente manera [3]:

Sea I el espacio de imágenes RGB y D el codominio de mapas de profundidad de valor real. Dado un conjunto de datos de entrenamiento:

$$\mathcal{T} = \{(\mathbf{I}_i, \mathbf{D}_i)\}_{i=1}^{M}, \mathbf{I}_i \in I \quad y \quad \mathbf{D}_i \in D, \tag{1}$$

el objetivo es obtener la transformación no lineal dada por (2):

$$\varphi: I \rightarrow D \, . \tag{2}$$

Las redes neuronales convolucionales de tipo autoencoder que se utilizan actualmente para la estimación de profundidad monocular relativa están diseñadas para aproximar, de forma supervisada o no supervisada, la transformación no lineal descrita por la ecuación (2) [1]. La función de mapeo obtenida es capaz de reconstruir un mapa denso de profundidad relativa a partir de una sola imagen RGB. Las arquitecturas convolucionales de tipo autoencoder están conformadas por un codificador (en inglés, encoder) y un decodificador (en inglés, decoder). El codificador permite extraer los mapas de características que contienen la representación global de la imagen de entrada. Por su parte, el decodificador es la parte encargada de reconstruir un mapa denso de profundidad relativa a partir de la representación global extraída [4].

A pesar de que los resultados obtenidos han sido prometedores, los algoritmos del estado del arte están basados en arquitecturas de redes profundas muy complejas, cuya etapa de inferencia no puede llevarse a cabo en tiempo real en sistemas con recursos computacionales limitados. Este hecho impide su uso en campos de aplicación donde las características de las cámaras monoculares pudieran ser explotadas [1, 5, 6].

En este trabajo, se propone una arquitectura convolucional de tipo autoencoder de baja latencia que emplea el codificador ShuffleNet V2 [7] como el extractor de características. De acuerdo a la revisión de la literatura, este codificador originalmente diseñado para tareas de clasificación no ha sido evaluado en el problema de estimación de profundidad monocular relativa.

2. Trabajos relacionados

Saxena *et al.* [8] propuso el primer trabajo relacionado con la estimación de profundidad monocular mediante la aplicación de técnicas de extracción manual de características y un entrenamiento discriminativo de campos aleatorios de Markov. No obstante, el método en [8] se caracteriza por un bajo desempeño en ambientes no controlados. Por su parte, Karsch *et al.* [9] propuso un método basado en el algoritmo K-NN (K Nearest Neighbors) para la estimación de profundidad en imágenes con fondo estático. A pesar del avance que representaron dichos trabajos [8, 9], ambos requieren de procedimientos laboriosos de alineación.

No fue hasta la implementación de técnicas de aprendizaje profundo cuando se lograron avances significativos en esta línea de investigación. El primer trabajo que impulsó este enfoque [10], utilizó dos redes neuronales convolucionales acopladas para inferir la profundidad asociada a cada pixel en una imagen. Este trabajo sobresalió al ser el primero en introducir el concepto de información multiescala en la estimación de profundidad monocular.

Estudios posteriores, tal como [11], se enfocaron en obtener transiciones visuales más nítidas al incorporar campos aleatorios condicionales (en inglés, conditional random fields) como una etapa de regularización. Por su parte, Laina *et al.* [4] fue uno de los primeros grupos de investigación en proponer el uso de redes neuronales convolucionales de tipo autoencoder, las cuales se caracterizan por no requerir etapas de postprocesamiento u otros pasos de refinamiento como los métodos anteriormente

descritos. A partir de este punto, la arquitectura de red neuronal convolucional de tipo autoencoder se comenzó a establecer como uno de los métodos de mayor desempeño en esta línea de investigación.

Los trabajos posteriores al propuesto en [4] se han enfocado en mejorar la métrica de exactitud de diversas maneras, p. ej., a través de la inclusión de información semántica en la etapa de entrenamiento [12]. En particular, el trabajo de Fu *et al.* [13] fue bien recibido por parte de la comunidad científica al proponer una arquitectura robusta que, haciendo uso de una técnica de convolución dilatada, es capaz de proveer mapas de profundidad de alta resolución. De manera similar a [13], otros trabajos, tal como el de Kumari *et al.* [14], han explorado la optimización del modelo mediante funciones de pérdida perceptivas con el propósito de mejorar la retención de detalles locales. Común a los trabajos arriba mencionados, es la estrategia de aprendizaje supervisada, la cual requiere datos etiquetados para el entrenamiento de las redes neuronales convolucionales.

Por otro lado, la estrategia de aprendizaje no supervisada es el enfoque ortogonal al mencionado anteriormente. Este enfoque explota el potencial de las redes neuronales convolucionales sin requerir un conjunto de datos de entrenamiento etiquetados. En este sentido, los autores en [15] abordaron la estimación de profundidad monocular como un problema de reconstrucción de imágenes, para el cual definen una nueva función de pérdida capaz de imponer consistencia a las disparidades producidas entre las imágenes derecha e izquierda. Recientemente, investigadores de la Universidad de Oxford [16] propusieron un método novedoso para la estimación de profundidad monocular basado en una red generativa cíclica antagónica (CGAN o CycleGAN por sus siglas en inglés). Dicho método no supervisado, aprovecha la síntesis de imágenes para la medición indirecta de la disparidad.

Cabe destacar que los métodos previamente mencionados, se han enfocado en incrementar la métrica de exactitud de la etapa de validación a expensas de una elevada cantidad de operaciones computacionales [5]. En este sentido, un muy escaso número de estudios se han realizado con el objetivo de explorar arquitecturas de bajo costo computacional para la estimación de profundidad monocular relativa [1, 5, 6]. Por un lado, Poggy *et al.* [5] adopta una estrategia de aprendizaje no supervisada para entrenar una red neuronal convolucional compacta de tipo autoencoder con respecto al conjunto de datos KITTI [17], con la cual, logra una reducción significativa en el número de parámetros y con ello, un decremento en el tiempo de inferencia. De manera similar, Wofk *et al.* [6] propone una arquitectura de red neuronal convolucional de tipo autoencoder, cuya principal característica es el diseño de un decodificador de baja latencia. Asimismo, los autores en [6] sugieren el uso adicional de algoritmos de compresión para disminuir aún más el tiempo de inferencia.

Siguiendo un enfoque similar a [6], este artículo demuestra que el desempeño en el tiempo de inferencia puede ser reducido si se utiliza el codificador ShuffleNet V2 [7]. Este codificador incorpora canales barajados (en inglés, channel shuffle) originalmente diseñados para reducir el costo computacional en tareas de clasificación. En general, la arquitectura convolucional de tipo autoencoder propuesta logra un tiempo de inferencia menor al que se logra con la arquitectura propuesta en [6]. Además, los resultados obtenidos demuestran que es factible obtener un tiempo de inferencia incluso menor a aquel obtenido mediante el uso adicional de algoritmos de compresión.

3. Metodología

En esta sección, se describe la arquitectura de la red neuronal convolucional propuesta, el conjunto de datos de entrenamiento seleccionado, el protocolo de entrenamiento establecido y las métricas de evaluación.

3.1. Arquitectura de la red

Las arquitecturas convolucionales de tipo autoencoder que se utilizan para la tarea en cuestión están conformadas de una etapa de codificación seguida de una etapa de decodificación [4]. La etapa de codificación permite extraer los mapas de características que contienen la representación global de la imagen RGB de entrada (altura x anchura x 3). Estos mapas de características son alimentados a la etapa de decodificación para reconstruir un mapa denso de profundidad (altura x anchura x 1) mediante operaciones de convolución e interpolación. La arquitectura propuesta se ilustra en la Fig. 1.

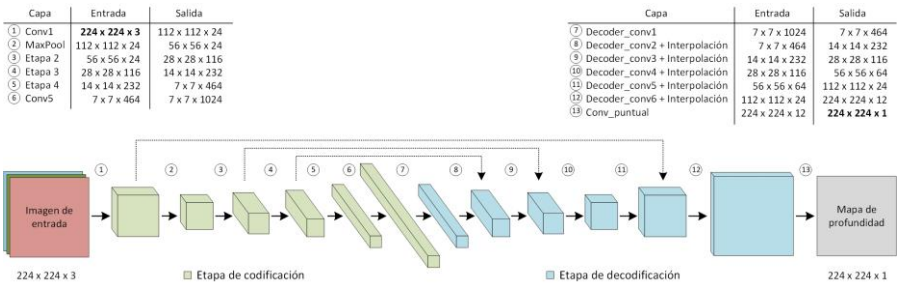


Fig. 1. Propuesta de arquitectura CNN de tipo autoencoder.

Para la etapa de extracción de características, se seleccionó el codificador del estado del arte ShuffleNet V2 [7]. Según los autores en [7], el codificador fue diseñado siguiendo cuatro pautas generales que se listan a continuación:

1. La profundidad/anchura de los canales en ambos extremos de una capa debe mantenerse igual para minimizar el costo de acceso a la memoria (MAC).
2. Se debe evitar el uso excesivo de grupos de convolución ya que esto aumenta el costo de acceso a la memoria.
3. Se sugiere minimizar el uso de operadores fragmentados para mantener un alto grado de paralelismo.
4. Se recomienda reducir el uso de operaciones elemento a elemento ya que su costo computacional no es despreciable.

En general, los autores en [7] sugieren utilizar métricas directas (p. ej., MAC) en lugar de métricas indirectas como el número de operaciones de coma flotante por segundo (FLOPS por sus siglas en inglés) para el diseño de arquitecturas de baja latencia. Nótese que la última capa de propagación hacia delante (en inglés, fully connected) del codificador ShuffleNet V2 debe ser removida para un correcto acoplamiento con la etapa de decodificación.

Además, los autores en [7] mencionan que es posible configurar un factor de profundidad que ajusta el número de canales de cada bloque. En este caso, el factor se estableció en *1x* con el propósito de mantener el número de parámetros de la red en un valor relativamente bajo y de esta manera, minimizar el tiempo de inferencia sin comprometer el valor de la exactitud umbral.

Por otro lado, la etapa de decodificación (ver Fig. 1) está inspirado en el diseño propuesto en [6]. En contraste con este último, el decodificador utilizado está conformado por seis capas de operaciones de convolución e interpolación. Estas capas están encargadas de aumentar gradualmente la resolución y reducir el número de canales. Cada una de estas capas utiliza un filtro de convolución de 5x5 y lleva a cabo la interpolación mediante el método del vecino más cercano. Un filtro de convolución puntual de 1x1 es aplicado al último mapa de características (224 x 224 x 12) para obtener el mapa de profundidad relativa resultante.

Similar a [6], se agregaron tres conexiones residuales aditivas entre el codificador y el decodificador que se ilustran como flechas punteadas en la Fig. 1. Estas conexiones permiten contrarrestar el problema de fuga de gradientes y la consecuente pérdida de nitidez en el mapa de profundidad relativa [14, 6].

3.2. Conjunto de datos de entrenamiento

El conjunto de datos seleccionado para entrenar y evaluar la arquitectura convolucional de tipo autoencoder propuesta es la NYU Depth V2 [18]. Se trata de un conjunto de datos disponible al público a partir del año 2012 y desde entonces, ha sido ampliamente usado en los trabajos del estado del arte relacionados con la estimación de profundidad monocular supervisada [18]. El conjunto de datos NYU Depth V2 está conformado por secuencias de video de diversas escenas de interiores que han sido capturadas mediante una cámara RGB-D. El conjunto de datos oficial contiene 120,000 muestras de entrenamiento y 654 muestras de validación, donde cada muestra consiste en un par de imágenes alineadas RGB y de profundidad. No obstante, algunos trabajos del estado del arte utilizan un subconjunto de 50,000 muestras de entrenamiento seleccionadas aleatoriamente para reducir el tiempo de dicha etapa [19]. En ese sentido, el método propuesto es entrenado bajo dicho esquema.

3.3. Protocolo de entrenamiento

La red neuronal convolucional propuesta es entrenada bajo una estrategia de aprendizaje supervisada. En particular, el proceso de entrenamiento fue llevado a cabo mediante el uso de una laptop Asus X556U 8GB RAM equipada con una tarjeta de video dedicada NVIDIA GeForce 930MX con una VRAM de 2GB. La implementación de la arquitectura propuesta se realizó mediante PyTorch 1.4.0 [20] (CUDA habilitado) en el entorno de desarrollo Spyder 4 y la distribución Ubuntu 16.04 LTS de Linux.

A partir de la interfaz de programación que provee Pytorch, se seleccionó el método del descenso de gradiente estocástico como el optimizador para la etapa de entrenamiento. Como parte de los hiperparámetros del optimizador, el impulso fue definido en 0.95, mientras que el método de regularización L2 fue habilitado con un valor de 0.0001.

Por las limitaciones propias de la VRAM descrita anteriormente, el tamaño del lote para la etapa de entrenamiento (en inglés, *batch size*) fue establecido en 8. Por su parte, el número máximo de épocas se definido en 30. Finalmente, la tasa de aprendizaje se estableció en 0.01 y fue configurada para que disminuyera su valor en un factor de 10 cada 5 épocas transcurridas. Los valores de los hiperparámetros anteriormente descritos fueron obtenidos a través de pruebas experimentales. Similar a otros trabajos del estado del arte [19], los pesos del codificador son inicializados a partir de un modelo preentrenado en el conjunto de datos ImageNet [21].

El error absoluto medio (MAE por sus siglas en inglés) se seleccionó como la función de pérdida a minimizar durante la etapa de entrenamiento [22]. Esta función considera la diferencia entre el mapa de profundidad objetivo que representa la verdad fundamental y (en inglés, ground truth) y la predicción de la red $\hat{y}$. La función de pérdida está dada por la siguiente expresión:

$$L_1(y, \hat{y}) = \frac{1}{n}\sum_{p=1}^{n}\left|y_p - \hat{y}_p\right|, \tag{3}$$

donde y_p es un pixel del mapa de profundidad objetivo y, $\hat{y}_p$ es un pixel en el mapa de profundidad relativa estimado $\hat{y}$, y n es el número total de pixeles del mapa de profundidad relativa.

3.4. Métricas de evaluación

El método propuesto se compara contra [6] mediante el tiempo de inferencia en milisegundos (ms) y dos métricas estándar en la estimación de profundidad monocular que se definen a continuación:

— Raíz del error cuadrático medio (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{p=1}^{n}\left(y_p - \hat{y}_p\right)^2} \tag{4}$$

— Exactitud umbral (δ_1):

$$\%\ de\ y_p \mid \max\left(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}\right) = \delta_1 < 1.25, \tag{5}$$

donde y_p es un pixel del mapa de profundidad objetivo y, $\hat{y}_p$ es un pixel en el mapa de profundidad relativa estimado $\hat{y}$, y n es el número total de pixeles del mapa de profundidad relativa.

La exactitud umbral debe entenderse como el porcentaje de pixeles del mapa de profundidad estimado para los cuales el error relativo es menor o igual al 25% [1, 6].

4. Resultados

En esta sección, se presentan los resultados cuantitativos y cualitativos que se obtuvieron al evaluar el modelo obtenido de la etapa de entrenamiento contra el

conjunto de datos de validación. Los resultados se contrastan con [6] al ser el único estudio relacionado que entrena y evalúa una arquitectura de baja latencia con respecto al conjunto de datos NYU Depth V2.

El proceso de evaluación se llevó a cabo utilizando la misma configuración de hardware que se utilizó para la etapa de entrenamiento.

3.1. Resultados cuantitativos

La Tabla 1 muestra la comparativa entre los resultados cuantitativos obtenidos al evaluar el método relacionado del estado del arte y el método propuesto contra el conjunto de datos de validación que proporciona la base de datos NYU Depth V2. Similar a los trabajos del estado del arte, cada valor en la Tabla 1 representa el promedio obtenido en el conjunto de datos de validación.

Tabla 1. Comparativa de resultados cuantitativos en el conjunto de datos de validación.

Método	$\boldsymbol{\delta_1}$ ↑	**RMSE** ↓	**GPU [ms]** ↓	**fps** ↑
Wofk *et al.* [6]	0.775	591.619	24	41
Wofk *et al.* + PN [6]	0.771	603.741	17	58
Método propuesto	0.757	606.351	15	66

El valor de la exactitud umbral (δ_1) obtenido por el método propuesto resulta ligeramente menor al valor alcanzado por el método del estado del arte (sin compresión). En ese sentido, el valor de la raíz del error cuadrático medio (RMSE) resulta ligeramente menor a favor del método del estado del arte (sin compresión). Dado que la magnitud de las diferencias porcentuales para cada métrica mencionada no es significativa, estas podrían considerarse despreciables.

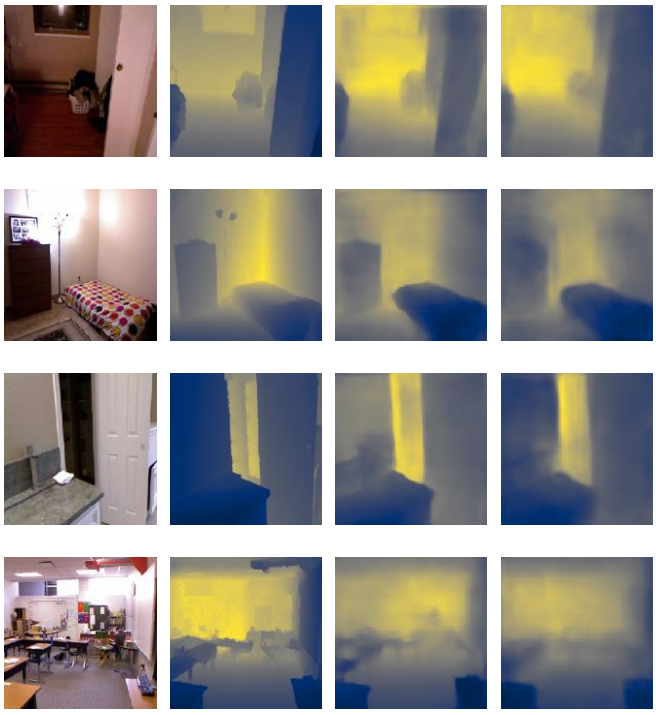
Por su parte, el tiempo de inferencia (ms) logrado por el método propuesto es menor al obtenido por el método del estado del arte (sin compresión) en un 37.5%. Nótese que el tiempo de inferencia del método propuesto es incluso menor al que se obtiene utilizando el modelo comprimido (poda neuronal) del método del estado del arte en un 11.7%.

3.2. Resultados cualitativos

La Fig. 2 muestra una comparativa de los resultados cualitativos con el método relacionado del estado del arte. Debe notarse que la profundidad relativa en cada escena disminuye conforme el color de los pixeles en los mapas de profundidad se oscurece. En otras palabras, entre más oscuro sea el color del pixel, más cercanos están los objetos en la escena.

En general, se aprecia en la Fig. 2 gran similitud entre el mapa de profundidad correspondiente al método propuesto (d) con respecto al método del estado del arte (c), aunque con mínimas diferencias en ciertas regiones locales.

Debe destacarse que ninguna imagen de la Fig. 2 fue utilizada en la etapa de entrenamiento.



(a) (b) (c) (d)

Fig. 2. Comparativa de resultados cualitativos en el conjunto de datos de validación (NYU Depth V2). De izquierda a derecha: (a) imagen RGB de entrada, (b) mapa de profundidad objetivo, (c) Wofk *et al.*+PN [6] y (d) método propuesto.

Desde esta perspectiva, es importante resaltar la buena capacidad de generalización del modelo obtenido en escenas de interiores del conjunto de datos de validación. Nótese que este modelo podría no funcionar de manera adecuada en escenas de exteriores.

5. Conclusiones y trabajo a futuro

El presente trabajo propone una arquitectura de red neuronal convolucional de tipo autoencoder para la estimación de profundidad monocular relativa en sistemas con recursos computacionales limitados. Los resultados experimentales del enfoque propuesto muestran que los filtros de convolución de baja latencia que conforman el codificador ShuffleNet V2 pueden contribuir a sentar las bases para el diseño de arquitecturas convolucionales para la estimación de profundidad monocular relativa en tiempo real. Nótese que el codificador ShuffleNet V2 no había sido evaluado en el problema de estimación de profundidad monocular, hasta ahora.

Debe destacarse que, sin el uso adicional de técnicas de compresión, el modelo obtenido se caracteriza por un tiempo de inferencia menor al que se obtiene al evaluar el método relacionado del estado del arte, manteniendo una exactitud umbral comparable al de dicho trabajo. En particular, el método propuesto logra una reducción del 37.5% en tiempo de inferencia con respecto al método mencionado anteriormente. Asimismo, debe notarse que el tamaño reducido de la arquitectura propuesta permite llevar a cabo las etapas de entrenamiento y validación en una laptop con especificaciones técnicas muy por debajo a las requeridas por los métodos del estado del arte que no son considerados de baja latencia.

Como trabajo a futuro, se tiene considerado entrenar la arquitectura propuesta en el conjunto de datos de entrenamiento KITTI [17] para evaluar su desempeño en escenas de exteriores. Asimismo, se pretende mejorar el desempeño de la métrica de exactitud umbral mediante la implementación de un decodificador que sea diseñado bajo las pautas y lineamientos sugeridos por los autores en [7].

Agradecimientos. Los autores agradecen a la Universidad Autónoma de Querétaro y al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo recibido durante el desarrollo de este proyecto de investigación. (Beca CONACYT No. 2019-000037-02NACF-03968).

Referencias

1. Zhao, C., Sun, Q., Zhang, Tang, Y., Qian, F.: Monocular depth estimation based on deep learning: an overview (2020)
2. Giancola, S., Valenti, M., Sala, R.: A survey on 3d cameras: metrological comparison of time-of-flight, structured-light and active stereoscopy technologies. Springer, Cham (2018)
3. Bhoi, A.: Monocular depth estimation: a survey (2019)
4. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: Proceedings of the IEEE International Conference on 3D Vision (3DV), pp. 239–248 (2016)
5. Poggi, M., Aleotti., F., Tosi, F., Mattoccia, S.: Towards real-time unsupervised monocular depth estimation on CPU. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5848–5854 (2018)
6. Wofk, D., Ma, F., Yang, T., Karaman, S., Sze, V.: FastDepth: fast monocular depth estimation on embedded systems. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 6101–6108 (2019)
7. Ma, N., Zhang, X., Zheng, H.T., Sun J.: ShuffleNet V2: Practical guidelines for efficient cnn architecture design. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – (ECCV) Lecture Notes in Computer Science, 11218, Springer, Cham (2018)
8. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Advances in neural information processing systems 19 (NeurIPS), pp. 1161–1168, Curran Associates, Inc. (2006)
9. Karsch, K., Liu, C., Kang, S.B.: Depth extraction from video using non-parametric sampling. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds) Computer Vision – (ECCV) Lecture Notes in Computer Science, 7576, Springer, Berlin, Heidelberg (2012)

10. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems 27 (NeurIPS), pp. 2366–2374, Curran Associates, Inc. (2014)
11. Liu, F., Shen, C., Lin, G., Reid, I.: Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10), 2024-2039 (2016)
12. Gurram, A., Urfalioglu, O., Halfaoui, I., Bouzaraa, F., López, A. M.: Monocular Depth Estimation by Learning from Heterogeneous Datasets. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 2176-2181, IEEE (2018)
13. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep Ordinal Regression Network for Monocular Depth Estimation. In: Proceedings of the IEEE/CFV Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2002-2011, IEEE (2018)
14. Kumari, S., Jha, R. R., Bhavsar A., Nigam, A.: AUTODEPTH: Single Image Depth Map Estimation via Residual CNN Encoder-Decoder and Stacked Hourglass. In: Proceedings of the 26th IEEE International Conference on Image Processing (ICIP), pp. 340-344, IEEE (2019)
15. Godard, C., Aodha, O. M., Brostow, G. J.: Unsupervised Monocular Depth Estimation with Left-Right Consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6602-6611, IEEE (2017)
16. Pilzer, A., Xu, D., Puscas, M., Ricci, E., Sebe, N.: Unsupervised Adversarial Depth Estimation Using Cycled Generative Networks. In: Proceedings of the IEEE International Conference on 3D Vision (3DV), pp. 587-595, IEEE (2018)
17. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal Robotics Research, 32(11), 1231–1237 (2013)
18. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference from RGBD Images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds) Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol. 7576. Springer, Berlin, Heidelberg (2012)
19. Alhashim, I., Wonka, P.: High Quality Monocular Depth Estimation via Transfer Learning. arXiv preprint arXiv: 1812.11941v2 (2018)
20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32 (NeurIPS), pp. 8024-8035, Curran Associates, Inc. (2019)
21. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248-255, IEEE (2009)
22. Carvalho, M., Saux, B. L., Trouvé-Peloux, P., Almansa, A., Champagnat, F.: On Regression Losses for Deep Depth Estimation. In: Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), pp. 2915-2919, IEEE (2018)

ISSN 1870-4069

Localización de referencias cefalométricas en radiografías laterales de cráneo: Una revisión sistemática

José Luis López-Ramírez, José Ruiz-Pinales,
Juan Gabriel Aviña-Cervantes, Pedro López-Rodríguez,
José Guadalupe Beltrán-Hernández, Mario Alberto Ibarra-Manzano

Universidad de Guanajuato, División de Ingenierías
Campus Irapuato Salamanca, Departamento de Ingeniería Electrónica,
México

{jl.lopezramirez,pinales}@ugto.mx

Resumen. Los sistemas de diagnóstico asistidos por computadora aplicados en la medicina juegan un papel fundamental en la detección de patologías (en etapa temprana y/o progreso), cuyo diagnóstico requiere la toma de muestras de "imagenología diagnóstica". Actualmente existe un número considerable de algoritmos computacionales que han obtenido resultados destacados en la estimación (automática y semiautomática) de referencias cefalométricas aplicada en radiografías laterales de cráneo. Siendo esta, una tarea que requiere de un lapso considerable de tiempo por parte de los médicos especialistas, dado que la identificación de los diferentes puntos de referencia requiere de un conocimiento amplio en el tema a la par de agudeza visual para identificar los diferentes niveles de resolución y clasificación dentro de los cefalogramas. En el área de diagnóstico clínico odontológico la planificación maxilofacial es utilizada comúnmente. Sin embargo, solo existe una base de datos pública presentada en el Simposio Internacional sobre Imágenes Biomédicas (SIIB 2014-2015). Este proyecto consiste en una revisión sistemática de los diferentes trabajos presentados hasta el momento. Adicionalmente, se presenta una comparación (cuantitativa y cualitativa) de los resultados obtenidos por cada uno de los autores en las diferentes técnicas propuestas para dar solución a esta tarea.

Palabras clave: Aprendizaje profundo, aprendizaje máquina, redes neuronales convolucionales (RNC), revisión bibliográfica, análisis cefalométrico.

Cephalometric References' Localization in Lateral Skull Radiographs: A Systematic Review

Abstract. Computer-Aided Diagnostic (CADx) systems applied in medicine play a fundamental role in the detection of pathologies (at an early stage

or progress), which diagnosis requires taking samples of medical images. Currently, there is a considerable number of computational algorithms that have obtained outstanding results in the automatic and semi-automatic cephalometric references localization systems, applied to lateral skull radiographs. Otherwise, this is a time-consuming task for the expert physicians, since it requires knowledge and significant lapse of time for the identification of the reference points and a visual acuity to identify the different levels of resolution and classification within the cephalograms. In the orthodontics field, the maxillofacial treatment plan is used commonly by dentists. Nevertheless, there is only an available public database which was presented at the International Symposium on Biomedical Images (ISBI) 2015. This project consists of a systematic review of the different works presented to date. Additionally, the comparison (quantitative and qualitative) of the results achieved by the authors in the different techniques proposed to solve this task.

Keywords: Deep learning, machine learning, convolutional neural networks, literature review, cephalometric analysis.

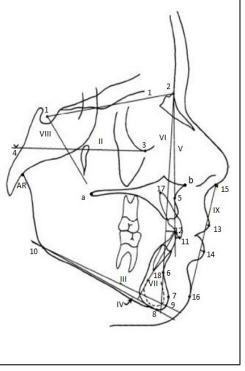
1. Introducción

Los sistemas computacionales han tenido un desarrollo significativo en las últimas décadas. En especial, han mostrado gran influencia en las ciencias médicas. En este campo se desarrollan sistemas de diagnóstico asistidos por computadora cuya finalidad tiene como objetivo la predicción y/o detección temprana de patologías. Estos sistemas ayudan a los médicos a emitir un prediagnóstico con alto grado de exactitud en estudios que requieren "imagenología diagnóstica". En el campo médico de la odontología existen estudios como la cefalometría. Esta última se encarga, de obtener mediciones del cráneo humano a partir de imágenes tomadas con rayos–X, bajo un procedimiento estricto de captura.

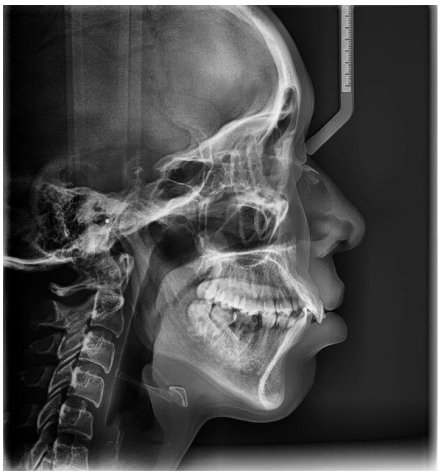
En la Figura 1a, se muestra un diagrama esquemático con los puntos cefalométricos propuestos a detectar. Adicionalmente la Figura 1b, proporciona un ejemplo de una radiografía lateral de cráneo, utilizada en el "Análisis cefalométrico". Para la identificación de referencias cefalométricas se debe contar con máquinas de rayos–X que cuya salida sea una imagen digital o impresa de las condiciones médicas del sistema óseo en la vista lateral del cráneo. También se requiere del tiempo, conocimiento y dedicación de médicos especialistas en el área para la identificación de los puntos cefalométricos. En la Figura 1a, se observan los nombres y ubicación de 19 referencias cefalométricas importantes para el diagnóstico clínico del crecimiento de cráneo, planificación de cirugías maxilofaciales, entre otros.

Actualmente existe una base de datos pública presentada en el SIIB 2014-2015. Esta base de datos ha sido utilizada desde entonces por diferentes autores, donde presentan sus algoritmos computacionales para resolver la identificación automática y semiautomática de las referencias cefalométricas. Se han obtenido resultados sumamente importantes con una alta precisión y exactitud.

Número	Etiqueta	Posición
1	Silla Turca	
2	Nasion	
3	Orbitario	
4	Prion	
5	Subespinal	
6	Supramentoniano	
7	Pogonion	
8	Eminencia	
9	Gnathion	
10	Gonion	
11	Incisal del incisivo inferior	
12	Incisal del incisivo superior	
13	Labrale superius	
14	Labrale inferior	
15	Punto Pm o MN	
16	Pogonion blando	
a	Espina nasal posterior	
b	Espina nasal anterior	
AR	Articular	



(a) Posiciones de los 19 puntos que conforman el análisis cefalométrico [13].



(b) Radiografía lateral de cráneo.

Fig. 1. Imágenes de rayos–X para estudio de Análisis cefalométrico.

En este trabajo se presenta una revisión sistemática de los trabajos que han sido publicados hasta el momento. Es posible obtener una comparación cuantitativa de dichos estudios, debido a que han sido entrenados y evaluados previamente con una misma base de datos, ya que de no ser así, solo es posible comparar las diferentes metodologías.

La evaluación del rendimiento computacional se realiza con una fracción de la base de datos utilizada. Esta última es elegida de manera aleatoria y destinada únicamente para las pruebas de rendimiento en todos los algoritmos propuestos. Las métricas de evaluación empleadas son: el Error Radial Medio (ERM) y la Tasa de Detección (TD) de éxito.

Este trabajo se organiza de a siguiente manera. La Sección **Materiales y Metodología**, donde se presentan las características de la única base de datos pública que utilizaron la mayoría de los autores de los trabajos presentados en esta revisión. Por otro lado se presenta una breve descripción de las diferentes técnicas utilizadas por cada uno de los autores. La Sección **Resultados**, presenta un análisis comparativo de los resultados cuantitativos y cualitativos, basados en las métricas de evaluación y obtenidos por los diferentes autores. En la Sección **Conclusiones** se presentan las deducciones de esta revisión sistemática.

2. Materiales y metodologías

2.1. Base de datos pública (SIIB 2014-2015)

La competición convocada por el Simposio Internacional sobre Imágenes Biomédicas 2015 respaldada por el IEEE (por sus siglas en inglés, "*Institute of*

Electrical and Electronics Engineers"), proporciona un conjunto de imágenes públicas de las competiciones para la detección de puntos de referencia cefalométricos en sus ediciones SIIB 2014 [18] y SIIB 2015 [19]. Esta base de datos se encuentra en el sitio web de la competición SIIB 2014-2015.

La base de datos está compuesta por 400 radiografías de pacientes de entre 7 y 76 años (edad promedio: 27 años; 235 mujeres, 165 hombres). Todos los cefalogramas se adquirieron en formato TIFF con una máquina "Soredex CRANEX Excel CEPH" (Tuusula, Finlandia), utilizando el software "Soredex SorCom 3.1.5, versión 2.0". La resolución de cada imagen es de 1935×2400 píxeles con una resolución espacial de $0{,}1 \times 0{,}1$ mm^2 por píxel. Un ejemplo de imagen de esta base de datos se ilustra en la Figura 2a.

El etiquetaje manual fue realizado por dos ortodoncistas clínicos expertos (con 15 años y seis años de experiencia, respectivamente) del departamento dental del hospital general de Taiwan, el etiquetaje del médico con más experiencia es llamado "Experto 1" y por consecuente el médico con 6 años es reconocido como "Experto 2". Los 19 puntos de referencia marcados fueron elegidos por involucrar estructuras comunes utilizadas en evaluaciones cefalométricas como el análisis cefalométrico de Steiner [17] y el análisis Wits Appraisal [7]. La base de datos fue dividida de manera aleatoria en 150 imágenes de entrenamiento y dos conjuntos de prueba: el primer conjunto también llamado "Prueba 1" está conformado por 150 imágenes con la que se evaluaron los resultados del SIIB 2014-2015 [18] y el segundo conjunto de prueba llamado "Prueba 2" consta de 100 imágenes y de igual forma fueron las imágenes de evaluación del SIIB 2014-2015 [19].

2.2. Métricas de evaluación

Son considerados dos criterios fundamentales para la evaluación del desempeño de los algoritmos computacionales, presentados en la Sección 2.3. Ambos criterios fueron propuestos durante las competiciones SIIB 2014 [18] y SIIB 2015 [19].

Error Radial Medio (ERM). El primer criterio de evaluación es el Error Radial Medio (ERM) en conjunto con la Desviación Estándar (DE) asociada. El primer criterio de evaluación es el error radial medio (ERM) en conjunto con la Desviación Estándar (DE) asociada. El primer parámetro a considerar es la distancia euclidiana entre cada píxel propuesto por el algoritmo y el píxel objetivo (referencia etiquetada por el médico experto) directamente sobre el cefalograma. Ésta distancia es definida matemáticamente por R en la Ecuación (1):

$$R = \sqrt{(\Delta X)^2 + (\Delta Y)^2}, \tag{1}$$

donde, ΔX y ΔY son las distancias en sus respectivos ejes (X y Y). Siendo $\Delta X = x_p - x_o$ y $\Delta Y = y_p - y_o$ (el subíndice "p" es de propuesto y el subíndice "o" de objetivo). Teniendo esto en cuenta, el ERM o distancia promedio es formulada como se muestra en la Ecuación (2):

$$\text{ERM} = \frac{1}{N} \sum_{i=1}^{N} R_i \qquad \text{[mm]}, \tag{2}$$

$$R_i = \frac{1}{M}\sum_{j=1}^{M} R_{ij} \qquad \text{[mm]}, \tag{3}$$

donde, la constante N es el número de cefalogramas del conjunto que se está evaluando y M es el número de referencias cefalométricas a localizar.

La Desviación Estándar (DE), representada en la Ecuación (4). La DE es una medida de la dispersión de datos (referencias cefalométricas), cuanto mayor sea la dispersión mayor es la DE, si no hubiese ninguna variación en los datos, es decir, si fueran todos iguales, la DE sería cero. El cálculo de la desviación estándar se realiza de la siguiente manera:

$$\text{DE} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(R_i - \text{ERM})^2}. \tag{4}$$

Tasa de Detección (TD) Para el cálculo de la tasa de detección, se considera exitosa la detección cuando un punto cefalométrico propuesto por un algoritmo que se encuentra dentro de cierto rango médico aceptable de distancias, tales como (2, 2.5, 3 y 4) mm. De esta forma es como se establece la tasa de detección (TD) dada por la Ecuación (5):

$$p_z = \frac{|\{i \in \mathbb{N} : R_i < z, 1{,}0 \leq i \leq N\}|}{N} \times 100\,\%, \tag{5}$$

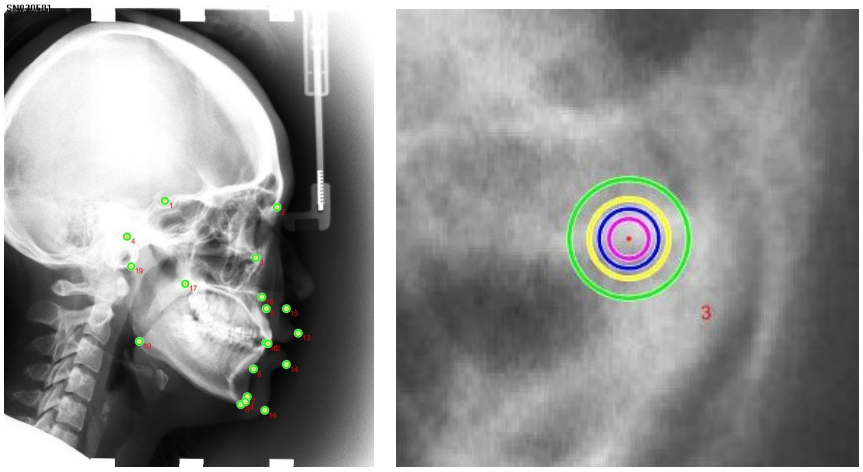
donde $|\cdot|$ simboliza la cardinalidad, R_i representa la distancia radial de la etiqueta propuesta. La literal z expresa el rango de precisión usado en la evaluación (Rangos: {2.0, 2.5, 3.0, 4.0} mm).

La Figura 2a presenta una una radiografía lateral de cráneo, perteneciente al conjunto de datos SIIB 2014-2015, con su respectivo etiquetado manual. Por otro lado la Figura 2b, representa un recorte de la etiqueta llamada "Orbitario". Siendo el punto rojo (centro), el píxel etiquetado por el experto y los anillos concéntricos (magenta, azul, amarillo y verde), los rangos de detección médicamente aceptables {2.0, 2.5, 3.0, 4.0} mm, respectivamente. Estos rangos de detección determinan la capacidad de precisión y exactitud de los sistemas computacionales para resolver este desafío.

En la siguiente Sección presenta una breve descripción de los métodos y técnicas propuestas en la literatura para dar solución a esta importante tarea.

2.3. Revisión literaria

Ibragimov *et al.* [5,6] presentan un marco de trabajo basado en "Teoría de juegos" (GTF) y *"Random Forests"* para detectar las etiquetas cefalométricas. Los puntos, detectados por GTF usando un modelo de apariencia multi-escala, son refinados por medio de *"Random Forest"* usando un modelo de forma. Primero es aplicado el modelo de apariencia multi-escala a la imagen-objetivo escalada y



(a) **(b)**

Fig. 2. a) Imagen cefalométrica de la base de datos. b) Rangos de detección (rojo: Verdad de Referencia ("*Ground-Truth*"), magenta: 2 mm, azul: 2.5 mm, amarillo: 3 mm y verde: 4 mm).

cuando se requiere una posición más precisa se aplica en la imagen original sin escalar. De esta manera, se considera información global al analizar la imagen escalada, e información local al analizar la imagen original, lo que conduce a una reducción de complejidad computacional. El resultado obtenido muestra que al menos el 71.7 % de las etiquetas detectadas se encuentran en el rango de 2mm.

Kaur *et al.* [8] utilizan la detección de características globales basadas en los momentos de Zernike. Este análisis, consiste en la clasificación de objetos obteniendo información sobre los diferentes rasgos geométricos descritos por los polinomios de Zernike debido a su alta capacidad de representación y la pequeña sensibilidad al ruido presentado en el fondo de las imágenes de rayos X. Cabe resaltar que la base de datos utilizada es una base de datos privada, conformada por un total de 135 imágenes y para su validación se utilizó el algoritmo *drop-one-out*, utilizando 134 imágenes para entrenamiento, y una imagen fue excluida para la prueba.

Claudia Lindner and Tim F. Cootes [10]. Proponen solucionar el etiquetaje automático de radiografías laterales de cráneo usando *"Random Forest-Regression Voting"*. Esta técnica cuenta con una alta capacidad de generalización para la solución de muchos problemas. Consiste en mejorar la generalización y aprender los patrones que se presentan en los datos de entrenamiento, compensando los errores de las predicciones de los distintos árboles de decisión mediante la votación por regresión, obteniendo resultados de precisión en el rango de los 2mm.

Claudia Lindner *et al.* [19]. Proponen un sistema llamado FALA (por sus siglas en inglés: *"Fully Automatic Landmark Annotation System"*). FALA es implementado

usando su técnica de *"Random Forest-Regression Voting"*, con la cual detectan la posición escala y orientación del cráneo, para posteriormente aplicar una técnica CML (por sus siglas en inglés: *"Constrained Local Model"*) que detecta las posiciones de los puntos de referencia. El primer paso hace que el sistema sea robusto a cualquier variación en la adquisición de imágenes, y el segundo paso permite la colocación precisa de los 19 puntos de referencia. La combinación de estas dos técnicas muestra el mejor desempeño para resolver esta tarea obteniendo los mejores resultados en la literatura existente. Este sistema logró un ERM de 1.2mm, y el 84.7 % de los puntos de referencia se ubicaron dentro del rango de precisión clínicamente aceptado de 2.0 mm.

Sercan Ö. Arık *et al.* [1]. Propone por primera vez una arquitectura de redes neuronales *"feed-forward"* profundas para el análisis de cefalometría cuantitativa que proporciona la estimación de 19 pares de datos cefalométricos localizados en radiografías laterales de cráneo, utilizando la base de datos públicas propuesta en la competición anteriormente descrita [19]. Los resultados obtenidos muestran que al menos un 76 % de los puntos propuestos de la partición de prueba son etiquetados dentro del rango de 2mm. En este caso, se utilizaron recortes (parches) de imágenes como entrada a la red neuronal.

Hansang Lee *et al.* [9] presentan un sistema basado en aprendizaje profundo, proponiendo una arquitectura de RNC para estimar por medio de regresión los 19 puntos de referencia. Se utilizó la base de datos pública SIIB 2014-2015. La metodología es descrita como el entrenamiento de 38 sistemas de regresión con la misma arquitectura RNC en variables de 19 pares coordenados (x, y), respectivamente. Finalmente, se calculan 38 variables de coordenadas como salida de estos sistemas entrenados a partir de imágenes invisibles. Obteniendo resultados plasmados en la gráfica de caja mostrada en la Figura 3. En el eje horizontal, se enumeran las 19 referencias cefalométricas obtenidas por el algoritmo propuesto. En el eje vertical, se muestra la distancia euclidiana que hay entre los valores que propone el algoritmo computacional y el punto etiquetado por el médico experto. Estos resultados muestran un rendimiento prometedor, sin embargo, no se mencionan resultados cuantitativos, por lo que visualmente no se puede estimar un resultado que pueda ser comparado con los resultados establecidos por las otras metodologías propuestas de esa revisión. Ji–Hoon Park *et al.* [4] comparan el rendimiento computacional de dos algoritmos basados en aprendizaje profundo. La base datos es compuesta por 1311 imágenes. Los sistemas fueron entrenados con un total de 1028 imágenes que se consideraron únicamente para el entrenamiento de las arquitecturas. Para las pruebas de rendimiento se utilizaron las 283 imágenes restantes. Cada imagen es compuesta por 80 etiquetas cefalométricas. Los métodos propuestos son la arquitectura *"You-Only-Look-Once"* versión 3 (YOLOv3) propuesta por Redmon *et al.* . [15] y el método *"Single Shot Multibox Detector"* (SSD) propuesto por Wei Liu *et al.* . [11]. El algoritmo YOLOv3 superó a SSD en precisión para 38 de 80 puntos de referencia. Los otros 42 de 80 puntos de referencia no mostraron una diferencia estadísticamente significativa entre YOLOv3 y SSD.

Runnan Chen *et al.* [2]. utilizan como metodología la combinación de un módulo basado en una RNC simple, que es utilizada para extraer las características de la imagen de entrada y posteriormente se utiliza una técnica llamada "*Attentive*

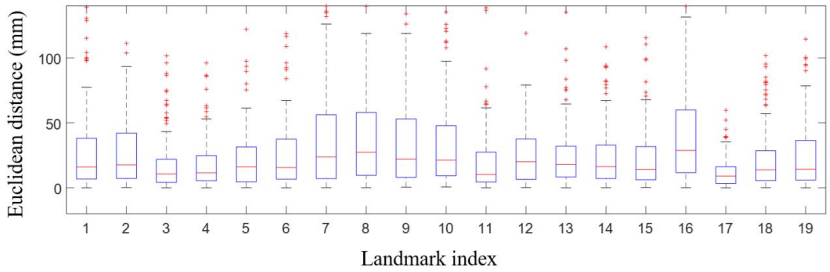


Fig. 3. Diagrama de caja de distancias euclidianas entre puntos de referencia predichos y los puntos objetivo, propuestos por Hansang Lee *et al.* [9].

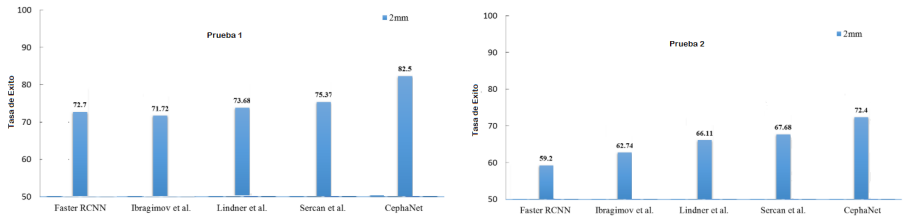
Feature Pyramid Fusion" (AFPF). Esta última, permite resaltar de forma explícita las características de alta resolución. De manera conjunta, se combinaron los "*heatmaps*" (mapas de atención) y mapas de desplazamiento, estas etapas permiten que el rendimiento del sistema mejore. Y así mismo sea más eficaz la regresión por votación de los píxeles. Obteniendo resultados de etiquetado de manera automática en tiempo real, directamente sobre imágenes del dispositivo que realiza la radiografía.

Xiubin Dai *et al.* [3] proponen una arquitectura basada en aprendizaje profundo utilizando redes generativas adversariales. Esta arquitectura está basada en el manejo de mapas de distancia de cada uno de los puntos cartométricos a encontrar. Teniendo la red generadora entrenada se obtienen mapas de distancia que proporcionan información de los 19 puntos cefalométricos. Obteniendo los 19 mapas con mejor aproximación al conjunto de imágenes objetivo son los que por otro lado la red discriminadora entrega como resultado.

Jiahong Qian *et al.* [14] diseñan un método rápido basado en R-CNN *"Region Based Convolutional Neural Network"*, por sus siglas en inglés). Los autores han nombrado a su método como "CephaNET". Esta arquitectura consiste en la construcción de un grafo 2D (no dirigido), donde se plasma la distribución de puntos de referencia en los datos de entrenamiento. Para eliminar los puntos anormales detectados, se adopta la "máxima confidencia" y la transformación Laplaciana. Este resultado es comparando con otras metodologías propuestas como se puede observar en la Figura 4.

Hengameh Mirzaalian *et al.* [12] desarrollaron este algoritmo computacional que extrae un conjunto de características que incluyen: características locales basadas en patrones binarios, coordenadas espaciales (X, Y) de los puntos cefalométricos, *blobness, tubularness* y características de Zernike. Se emplearon estas características para calcular la verosimilitud de cada uno de los puntos de referencia usando un clasificador *"Random Forest"*. En la Figura 5, se presentan los resultados obtenidos en esta investigación.

Yu Song *et al.* [16] proponen un método semiautomático para la detección de puntos de referencia cefalométricos mediante aprendizaje profundo. Primero se extrae manualmente una región ROI (por sus siglas en inglés: "*Region Of*



(a) **(b)**

Fig. 4. Resultados de CephaNET Vs Otros-autores, en el rango clínico de 2mm, para los dos subconjuntos de imágenes de prueba [14].

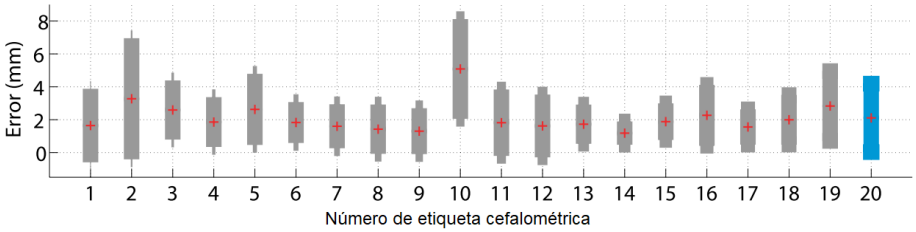


Fig. 5. Resultados obtenidos por el algoritmo computacional propuesto por Hengameh Mirzaalian *et al.* [12].

Interest") para cada punto de referencia de manera manual. Posteriormente, se utiliza una arquitectura Resnet50 para obtener las coordenadas de la referencia cefalométrica que se encuentra dentro de la ROI. Obteniendo resultados que superan cuantitativamente los resultados del estado del arte.

Los diferentes trabajos presentados anteriormente concuerdan en que la cefalometría cuantitativa y precisa toma un papel importante para los diagnósticos o tratamientos odontológicos. Sin embargo, es un trabajo que conlleva mucho tiempo para etiquetar manualmente los puntos de referencia cefalométricos. Dicha implementación requiere de médicos profesionales expertos. En conclusión, el desarrollo de un sistema de diagnóstico que ofrezca una precisión y exactitud comparable a la de un médico experto en el área, agilizaría el diagnóstico para posteriormente dar el tratamiento adecuado y si es el necesaria la planificación de la cirugía.

3. Resultados

A continuación, se presenta la Tabla 1, donde se muestran los resultados obtenidos por los diferentes autores de acuerdo a la métrica ERM.

Por lo tanto no se calculó el ERM para este subconjunto de datos. De igual forma los autores han tomado como regla, sólo entregar sus resultados con el

primer subconjunto de datos llamado “Prueba 1”, por lo tanto no es posible tener un registro de sus resultados para esta métrica y en la Tabla 1, está marcada con SR (Sin Resultado). En el caso específico de los trabajos de Sercan *et al.* y CephaNET. Los autores no presentaron los resultados globales del ERM, sólo se limitaron a presentar la Tasa de Detección (TD). Como previamente se ha mencionado en la Sección 2.3.

Tabla 1. Tabla de resultados del ERM, para autores que utilizaron la base de datos propuesta por el SIIB 2014-2015.

Autor	ERM(Prueba 1)	ERM(Prueba 2)
Ibragimov *et al.* (2014)	1.919	SR
Ibragimov *et al.* (2015)	1.851	1.82
Chu *et al.*	2.679	SR
Chen y Zhang	2.847	SR
Mirzaalian y Harmarneh	2.353	SR
Vandaele *et al.*	2.198	SR
Lidner and Cootes	1.67	SR
Sercan *et al.*	SR	SR
CephaNET	SR	SR
Y. Song *et al.*	1.14746	1.22347
Runnan Chen *et al.*	1.17	1.48

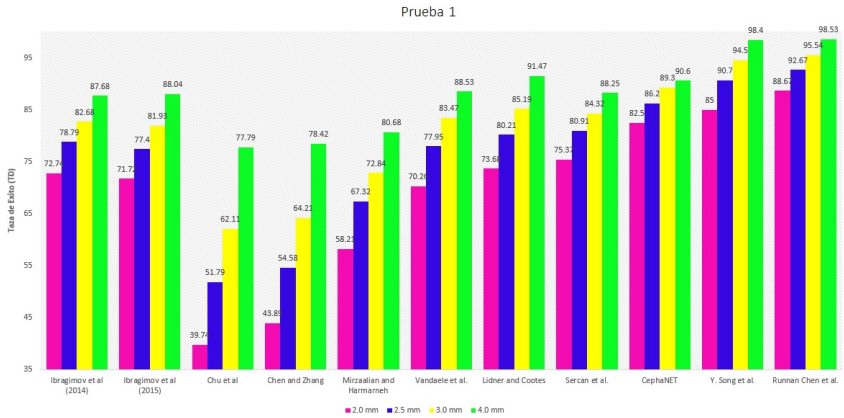
Como se puede observar en la Tabla 1, el mejor resultado es presentado por Y. Song *et al.* [20], seguido de Runnan Chen *et al.* [2]. Por un lado, Song presenta un sistema semiautomático que requiere que el médico seleccione el de manera manual, recortes de ventanas de 128 x 128 píxeles que contengan un punto cefalométrico, por lo que el punto es encontrado con mayor precisión y exactitud. Por otro lado, Chen *et al.* presentan una arquitectura donde, de manera indirecta generan un mapa 2D que contiene la distribución de los puntos cefalométricos de las imágenes etiquetadas.

De igual manera esto representa una ayuda indirecta al algoritmo computacional y posteriormente aplican transformaciones Laplacianas para encontrar el punto objetivo con mayor certidumbre.

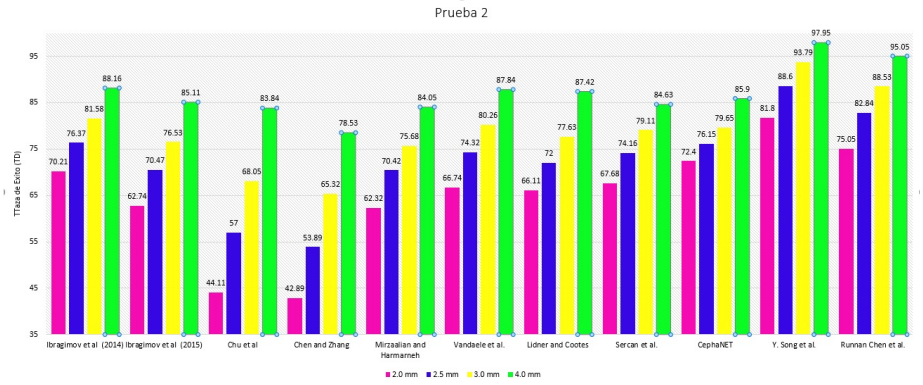
Además, se muestra que el mejor resultado para un sistema completamente automático es el presentado por Claudia Lindner [10]. Este sistema utiliza la técnica computacional *Random Forest-Regression Voting*. Un árbol de decisión representa un modelo de clasificación o regresión aplicado al procesamiento de imágenes en una estructura de árbol. Cada nodo en el árbol representa una característica del espacio de entrada, cada decisión una rama y cada hoja al final de una rama es el valor de salida correspondiente. Esto es realizado múltiples ocasiones para cada una de las 19 etiquetas a identificar. Posterior a que se generalizan y se aprenden patrones, se compensan los errores de predicción con una votación, haciendo un

algoritmo robusto, con un rendimiento alto para superar los resultados de los otros sistemas automáticos presentados anteriormente.

En la Figura 6 se presentan dos gráficas de barras donde se muestra el porcentaje de la tasa de detección para los cuatro rangos médicamente aceptables (2 mm, 2.5 mm, 3 mm, 4 mm). Para los 11 trabajos presentados que utilizaron la



(a) Resultado Experimental 1



(b) Resultado Experimental 2

Fig. 6. Comparación de resultados entre autores para la métrica TD.

base de datos de la competición del SIIB 2014-2015. La Figura 6b representan las TDs, para el subconjunto de imágenes llamado “Prueba 1”, que consta de un total de 150 imágenes. La Figura 6a muestra las TDs para el subconjunto de imágenes “Prueba 2”, de igual manera es un subconjunto de 150 imágenes de prueba.

Por otro lado, existen trabajos que utilizaron su propia base de datos y presentan metodologías importantes que dan solución a la localización de etiquetas

cefalométricas.

Kaur *et al.* obtuvieron una localización del 89.5 % en la tasa de detección, dentro del rango de 2 mm. Siendo este un resultado prometedor y poder ser cuantitativamente comparables los resultados con el obtenido por Claudia Lindner *et al.* , debe ser probado este algoritmo con la base de datos del SIIB 2014-2015.

Por otro lado Ji–Hoon Park *et al.* comparan dos algoritmos computacionales, obteniendo como resultado que entre el *Single Shot Multibox Detector* y la arquitectura YOLOv3 el que obtiene el mejor desempeño es esta última, obteniendo tasas de detección de, 96.2 %, 92 %, 87.4 % y 80.4 % en los rangos de {4, 3, 2.5, 2} mm respectivamente. En su manuscrito comparan los resultados contra los obtenidos por Sercan Ö. Arık *et al.* por ser arquitecturas que utilizan el mismo fundamento científico, pero al no utilizar la misma base de datos los resultados no pueden ser sistemas competitivos cuantitativamente.

La siguiente Sección se presenta las conclusiones obtenidas respecto al diagnóstico clínico de la identificación de etiquetas cefalométricas y así mismo los aspectos que se deben cuidar al intentar resolver esta tarea.

4. Conclusiones

La identificación automática de referencias cefalométricas es un procedimiento médico rutinario para la planificación de cirugías maxilofaciales en el área de la ortodoncia. Este proceso ocupa tiempo y conocimiento de un humano experto, lo cual hace que el realizar un algoritmo computacional para un sistema de diagnóstico asistido por computadora, que tenga características de precisión y exactitud dentro de los rangos médicos aceptables sea aún una tarea pendiente. Los algoritmos presentados en este trabajo reflejan que primero se tiene que implementar un mecanismo automático para la detección de las regiones de interés, donde estas regiones contengan las etiquetas cefalométricas y eliminar las variaciones que afectan a muchos algoritmos. Una de las técnicas que en los últimos años han mostrado un alto rendimiento computacional, son los modelos de aprendizaje profundo. Estos algoritmos están inspirados biológicamente en el comportamiento visual humano y pueden resultar poderosos. Además, los resultados presentados muestran que ya se tiene un TD cercano al 99 % dentro del rango de 4mm, sin embargo, aún no se ha dado por concluida esta tarea.

Referencias

1. Arik, S.Ö., Ibragimov, B., Xing, L.: Fully automated quantitative cephalometry using convolutional neural networks. Journal of Medical Imaging 4(1), 014501 (2017)
2. Chen, R., Ma, Y., Chen, N., Lee, D., Wang, W.: Cephalometric landmark detection by attentive feature pyramid fusion and regression-voting. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.T., Khan, A. (eds.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. pp. 873–881. Springer International Publishing, Cham (2019)

3. Dai, X., Zhao, H., Liu, T., Cao, D., Xie, L.: Locating Anatomical Landmarks on 2D Lateral Cephalograms through Adversarial Encoder-Decoder Networks. IEEE Access 7, 132738–132747 (2019)
4. Hwang, H.W., Park, J.H., Moon, J.H., Yu, Y., Kim, H., Her, S.B., Srinivasan, G., Aljanabi, M.N.A., Donatelli, R.E., Lee, S.J.: Automated identification of cephalometric landmarks: Part 2-Might it be better than human? Angle Orthodontist 90(1), 69–76 (2020)
5. Ibragimov, B.: Automatic cephalometric x-ray landmark detection by applying game theory and random forests. In: Grand Challenges in Dental X-ray Image Analysis (2014)
6. Ibragimov, B., Likar, B., Pernus, F., Vrtovec, T.: Computerized cephalometry by game theory with shape-and appearance-based landmark refinement. In: Grand Challenges in Dental X-ray Image Analysis (2015)
7. Jacobson A: The "Wits" appraisal of jaw disharmony. American Journal of Orthodontics 67(2), 125–138 (1975)
8. Kaur, A., Singh, C.: Automatic cephalometric landmark detection using Zernike moments and template matching. Signal, Image and Video Processing 9(1), 117–132 (2015)
9. Lee, H., Park, M., Kim, J.: Cephalometric landmark detection in dental x-ray images using convolutional neural networks. Medical Imaging 2017: Computer-Aided Diagnosis 10134(March 2017), 101341W (2017)
10. Lindner, C., Cootes, T.F.: Fully automatic cephalometric evaluation using Random Forest regression-voting. Isbi pp. 1–8 (2015)
11. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9905 LNCS, 21–37 (2016)
12. Mirzaalian, H., Hamarneh, G.: Automatic globally-optimal pictorial structures with random decision forest based likelihoods for cephalometric x-ray landmark detection. In: Automatic Cephalometric X-Ray Landmark Detection Challenge 2014, in conjunction with IEEE International Symposium on Biomedical Imaging (IEEE ISBI). pp. 1–12 (2014), http://www.cs.sfu.ca/~hamarneh/ecopy/isbi2014_chlg_ceph.pdf
13. Ortoface: ANÁLISIS DEL TRAZADO: REFERENCIAS CEFALOMÉTRICAS. url http://ortoface.com/wp-content/uploads/2016/12/An %CC %81alisis-del-trazado.pdf (2016)
14. Qian, J., Cheng, M., Tao, Y., Lin, J., Lin, H.: Cephanet: An improved faster r-cnn for cephalometric landmark detection. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 868–871 (2019)
15. Redmon, J., Farhadi, A.: YOLO v.3. Tech report pp. 1–6 (2018), https://pjreddie.com/media/files/papers/YOLOv3.pdf
16. Song, Y., Qiao, X., Iwmoto, Y., Chen, Y.W.: Semi-automatic cephalometric landmark detection on x-ray images using deep learning method. In: Liu, Y., Wang, L., Zhao, L., Yu, Z. (eds.) Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery. pp. 585–592. Springer International Publishing, Cham (2020)
17. Steiner, C.C.: Cephalometrics for you and me. American Journal of Orthodontics 39(10), 729–755 (1953)
18. Wang, C.W., Huang, C.T., Hsieh, M.C., Li, C.H., Chang, S.W., Li, W.C., Vandaele, R., Marée, R., Jodogne, S., Geurts, P., Chen, C., Zheng, G., Chu, C., Mirzaalian, H., Hamarneh, G., Vrtovec, T., Ibragimov, B.: Evaluation and Comparison of Anatomical Landmark Detection Methods for Cephalometric X-Ray Images: A Grand Challenge. IEEE Transactions on Medical Imaging 34(9), 1890–1900 (2015)

19. Wang, C.W., Huang, C.T., Lee, J.H., Li, C.H., Chang, S.W., Siao, M.J., Lai, T.M., Ibragimov, B., Vrtovec, T., Ronneberger, O., Fischer, P., Cootes, T.F., Lindner, C.: A benchmark for comparison of dental radiography analysis algorithms. Medical Image Analysis 31, 63–76 (jul 2016), https://linkinghub.elsevier.com/retrieve/pii/S1361841516000190
20. Yu, H.J., Cho, S.R., Kim, M.J., Kim, W.H., Kim, J.W., Choi, J.: Automated Skeletal Classification with Lateral Cephalometry Based on Artificial Intelligence. Journal of Dental Research (2020)

ISSN 1870-4069

Propuesta de asistente virtual basado en inteligencia artificial en la integración de un brazo robótico para aplicaciones en la industria 4.0

Miguel Ángel Hernández Figueroa, Rodrigo Gibran Vázquez Báez, José Orlando Castro Rangel

Universidad de la Salle Bajío,
División de Ingenierías de la Campus Salamanca,
México

{mahfigueroa, rodrigo.vazquezb,orlas1978}@gmail.com

Resumen. Facilitar la interacción entre humanos y máquinas es uno de los desafíos tecnológicos actuales, el despliegue de todo el potencial que la Industria 4.0 puede ofrecer requiere de interfaces cada vez más avanzadas e inteligentes que permitan a las máquinas y robots interactuar de forma natural con el usuario, entender su entorno y llevar a cabo la toma de decisiones con cierto nivel de conciencia. El desarrollo de asistentes virtuales basados en Inteligencia Artificial juega un papel muy importante en la manera de comprender y responder a las necesidades del usuario, asistiendo en el desempeño de labores automatizadas y personalizadas, agilizando la comunicación entre el humano y la tecnología, haciendo más eficiente el manejo de recursos y transformando la forma en que convivimos. Este artículo explora las posibilidades que se abren al integrar un brazo robótico de uso industrial al campo de la Inteligencia Artificial donde la propuesta de un asistente virtual inteligente permite redefinir las capacidades de interacción y comunicación ofreciendo soluciones al usuario dentro de cualquier escenario en la industria 4.0.

Palabras clave: Asistente virtual inteligente, robot autónomo, inteligencia artificial, industria 4.0.

Proposal of Virtual Assistant based on Artificial Intelligence in the Integration of a Robotic Arm for Applications in the Industry 4.0

Abstract. Facilitating the interaction between humans and machines is one of the current technological challenges. Unlocking the full potential that industry 4.0 can offer requires increasingly advanced and intelligent interfaces that allow machines and robots to interact naturally with the user, understand their environment and carry out decision-making with a certain level of awareness. The development of virtual assistants based on artificial intelligence, plays a very important role in the way of understanding and responding to user needs, assisting in the performance of automated and personalized tasks, streamlining

communication between humans and technology, making the resource management and transforming the way we live together. This article explores the possibilities that open up by integrating a robotic arm for industrial use into the field of artificial intelligence where the proposal of an intelligent virtual assistant allows to redefine the capabilities of interaction and communication offering solutions to the user within any scenario in industry 4.0.

Keywords: Intelligent virtual assistant, autonomous robot, artificial intelligence, industry 4.0.

1. Introducción

La arquitectura de la Industria 4.0 [1] comprende de un nivel físico, sensores, actuadores, máquinas y robots, hasta un nivel digital donde una capa de información y comunicación fluye de manera bidireccional a través de sistemas basados en IIOT (Industrial Internet Of Things, Internet Industrial de las cosas) [2], pasando por sistemas de computación en la nube, donde diferentes tecnologías y ramas de la Ciencia Computacional entre ellas la Inteligencia Artificial logran crear soluciones a las cambiantes necesidades del mercado, adaptando los procesos de producción y haciendo un uso más eficiente e inteligente de los recursos.

La forma en que los niveles físicos y digitales se entrelazan de una manera eficiente y fluida puede comprenderse gracias a los avances en los Sistemas Ciber-Físicos conocidos por sus siglas en inglés como CPS (Cyber-Physical Systems) [3] y M2M (Machine to Machine, Máquina a Máquina) [4], sin embargo, la interacción entre el mundo digital y el mundo físico visto desde la perspectiva del usuario presenta un desafío en el que el desarrollo de interfaces inteligentes que permitan la apertura de nuevos canales de comunicación entre el hombre y la tecnología se hacen evidentes, es el caso de las interfaces de usuario tradicionales HMI (Human Machine Interface, Interfaz Humano Máquina) [5] donde se centraliza el control del sistema, la visualización de la información del estado y sus variables, teniendo como único medio de interacción la pantalla táctil, teclado, apuntadores, etc., donde el operador ingresa manualmente los comandos requeridos, generando una disrupción en la cadena productiva.

Facilitar la interacción entre humanos, máquinas, robots y dispositivos mediante una comunicación más natural nos permite obtener un entorno más conectado y colaborativo, los avances en el campo de la Inteligencia Artificial han puesto a nuestra disposición diferentes tecnologías que al ser aplicadas al desarrollo de asistentes inteligentes [6] otorgan al sistema la capacidad de reaccionar y ser consciente de los diferentes estímulos tanto del usuario como de su entorno.

Los asistentes de voz [7] permiten al usuario dictar comandos de manera verbal, el reconocimiento de voz permite al sistema comprender y procesar la información y responder de igual manera. Los sistemas de visión artificial [8] tienen la capacidad de adquirir, procesar, analizar y comprender imágenes del mundo real, por lo tanto reconocer objetos, personas y colores es una tarea común. La integración del IoT (Internet of Things, Internet de las Cosas) permite al usuario establecer comunicación con los dispositivos en su entorno, como de forma remota, enviando y recibiendo

información del sistema a la nube, donde diferentes herramientas son empleadas para asistir al usuario y favorecer al proceso.

El objetivo que el presente trabajo busca es el de proponer una alternativa a los métodos de interacción y de interfaces de usuario tradicionales, mediante el desarrollo de un sistema de asistente virtual inteligente que logre aumentar [9] las capacidades colaborativas y de interacción entre el hombre y los dispositivos, máquinas y robots que conforman su entorno, por medio de la integración de múltiples tecnologías que en conjunto permitan crear un sistema diseñado para asistir al usuario en el análisis, monitoreo y control de un proceso industrial común, para esto, se hace uso de un brazo robótico como dispositivo principal para el despliegue del asistente virtual inteligente cuyos alcances serán definidos y analizados mediante la simulación de un proceso propio de la Industria 4.0.

Este trabajo está organizado por secciones. La Sección 2 presenta y describe la tecnología integrada, así como su aplicación en el sistema. La Sección 3 ejemplifica la metodología del asistente virtual inteligente trabajando en un entorno simulado. La sección 4 discute los resultados y trabajos futuros. En adelante, nos referiremos al asistente virtual inteligente con las abreviaturas AVI.

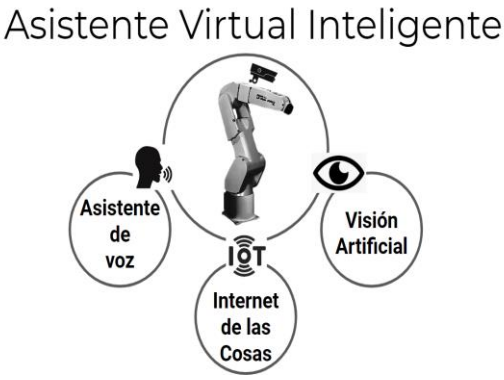


Fig. 1. Asistente virtual inteligente, tecnologías aplicadas.

2. Tecnologías aplicadas

La propuesta del AVI contempla el uso de un brazo robótico de nivel industrial de la marca Fanuc LR Mate 200iD [10] como dispositivo principal para el despliegue de las 3 principales tecnologías integradas, como se muestra en la Figura 1:

El brazo robótico realiza la adquisición de diferentes variables, el procesamiento y visualización de la información del AVI se realiza por medio de un computador que alberga al programa principal desarrollado en Python y es el encargado de orquestar el entrelazamiento de las diferentes tecnologías utilizadas. El sistema se comunica a través de internet con un servidor web alojado en una Raspberry Pi [11] quien por medio de sus pines emite las señales de salida que permiten controlar los movimientos del brazo robot como se muestra en la Figura 2.

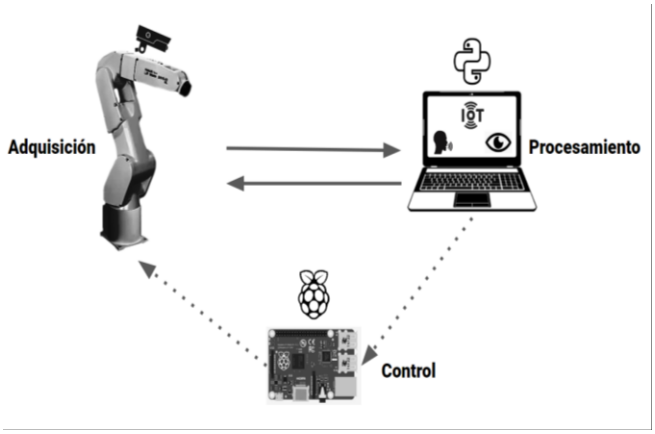


Fig. 2. Arquitectura del AVI.

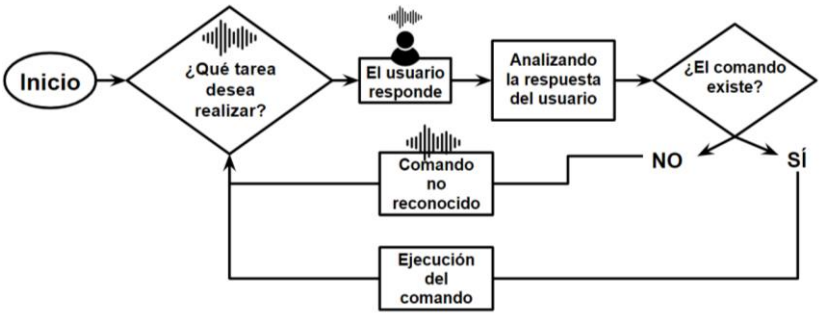


Fig. 3. Diagrama de flujo del comportamiento del asistente de voz.

2.1. Asistente de voz

El asistente de voz es el encargado de establecer el medio de comunicación entre el usuario y el sistema. El asistente de voz está compuesto de diferentes librerías especializadas en reconocimiento de voz y generación de respuesta, entre ellas gTTS (Google-Text-to-Speech) [12] que utiliza redes neuronales de aprendizaje profundo para la generación de salidas de audio que asemejan a la voz humana con gran naturalidad como se muestra en [13].

Dentro del sistema AVI el asistente de voz es utilizado para interactuar con el usuario e identificar en su voz la solicitud de ejecución de ciertos comandos, que generan una respuesta por salida determinada en la programación. Una descripción del comportamiento del AVI puede observarse en la Figura 3:

Las salidas pueden ser de diferentes tipos dependiendo el comando: ejecutar una acción, respuesta audible o salida por pantalla. Así mismo, un comando puede incluir una combinación de distintas salidas, como se muestra en la Figura 4, donde el usuario pregunta al AVI "¿Cuál es tu nombre?", generando la ejecución de un comando con 3 respuestas de salida:

Fig. 4. Respuesta del AVI: a) salida por pantalla, b) respuesta audible, c) movimiento del brazo robot.

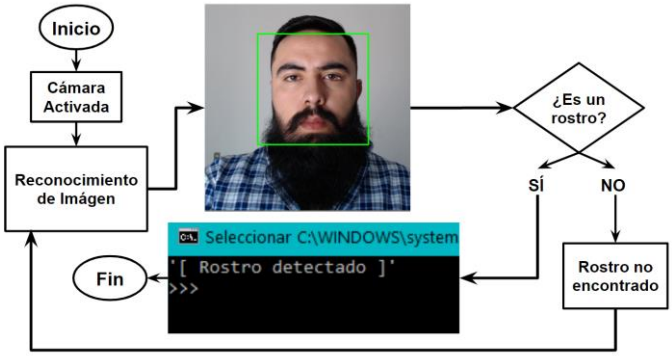


Fig. 5. Diagrama de flujo reconocimiento facial en el AVI.

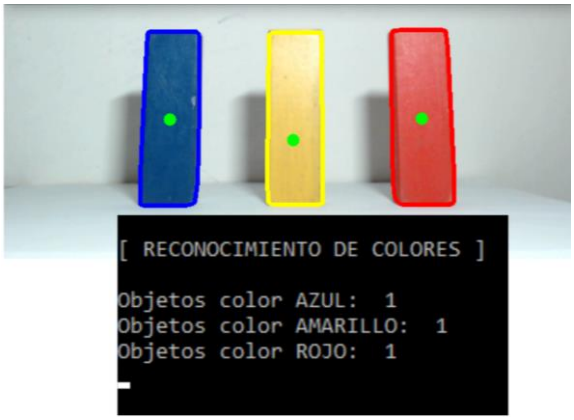


Fig. 6. Reconocimiento y contabilización de AVI.

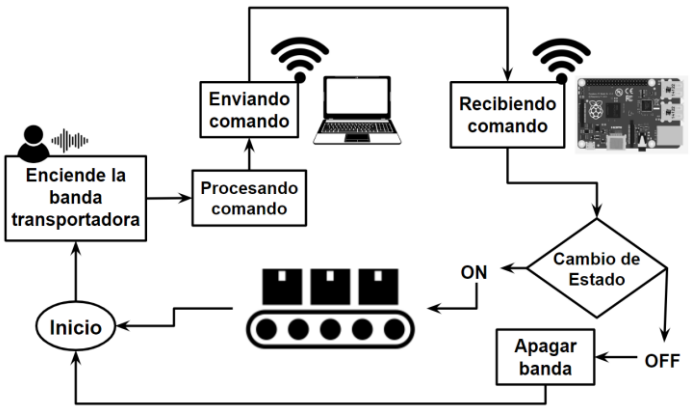


Fig. 7. Diagrama de flujo, encendido de banda transportadora vía internet.

2.2. Visión artificial

El sistema es capaz de adquirir imágenes por medio de una cámara web acoplada al cabezal del brazo robótico, desde donde transmite información en tiempo real al programa. Haciendo uso la librería Opencv [14] especializada en visión artificial, el sistema emplea diferentes algoritmos entre ellos el de reconocimiento facial para identificar el rostro del usuario y producir una respuesta, en este caso una salida por pantalla, como se muestra en la Figura 5:

Bajo el mismo principio el sistema también es capaz de reconocer colores y objetos, así como de contabilizarlos como se muestra en la Figura 6:

De esta manera el AVI integra la visión computacional como una herramienta más para asistir y responder a las necesidades del usuario.

2.3. Internet de las cosas

El sistema está diseñado para desenvolverse en un entorno conectado a internet, mediante una comunicación inalámbrica WiFi, el AVI cuenta con la capacidad de intercambiar información con los diferentes dispositivos conectados a la red, así como de monitorear y controlar su estado. En la Figura 7 el usuario le solicita al AVI de manera verbal que encienda la banda transportadora, el sistema reconoce el comando y a través de internet establece comunicación con el servidor web alojado en la Raspberry Pi, emitiendo una señal de salida por medio de sus pines, iniciando el movimiento de la banda transportadora.

La capacidad de acceder a internet permite al AVI la posibilidad de intercambiar información con una base de datos en la nube, así como de redactar informes en caso de errores o fallos y posteriormente ser enviados por e-mail de manera automática notificando al usuario en todo momento y generando un registro para su análisis y consulta (Figura 8).

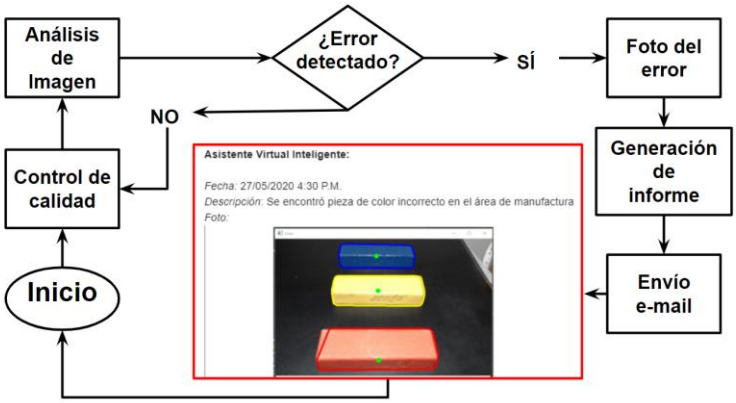


Fig. 8. Diagrama de flujo, generación y envío de informe de error al usuario.

Fig. 9. Línea de producción, brazo robot (izquierda), banda transportadora (derecha).

3. Aplicación del AVI en un entorno simulado

Para la aplicación del AVI se llevó a cabo en las instalaciones de la Universidad, el montaje y simulación de un proceso industrial típico del área de manufactura, donde se puede dar muestra de las capacidades que la integración de las distintas tecnologías empleadas en el AVI ofrece en un escenario propio de la Industria 4.0.

3.1. AVI en los procesos de manufactura y control de calidad

Se montó una pequeña línea de producción, integrada principalmente por el brazo robot y una banda transportadora (Figura 9).

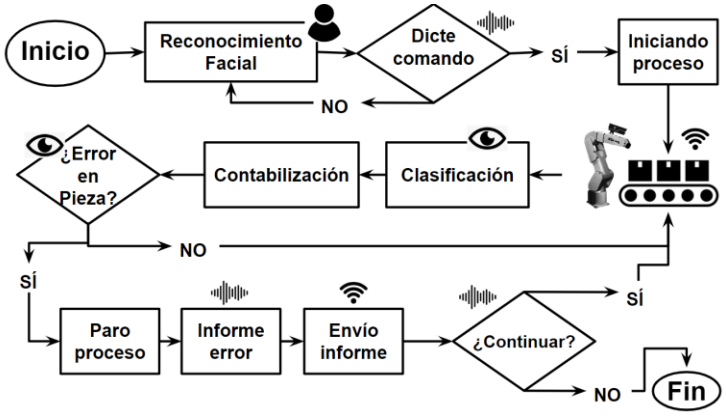


Fig. 10. Diagrama de flujo del proceso de control de calidad.

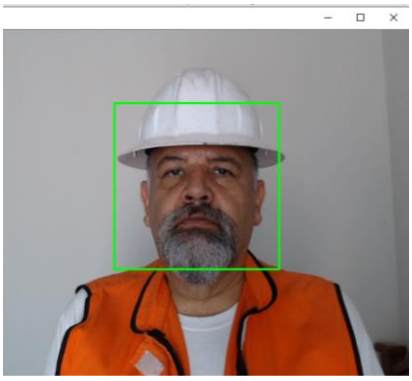


Fig. 11. Reconocimiento facial del operador.

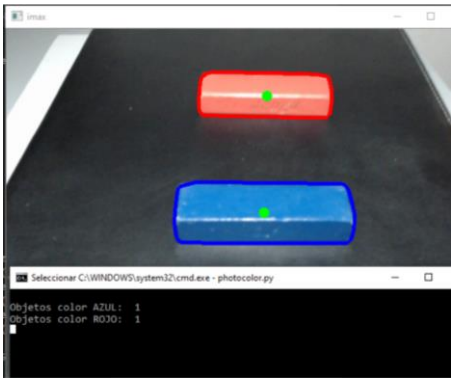


Fig. 12. Clasificación y contabilización de piezas por color.

El AVI es el encargado de supervisar y controlar todos los dispositivos involucrados en el proceso, así como de asistir al operador ante cualquier petición o consulta. [13].

Una vez iniciado el proceso por el operador, las piezas de un producto terminado corren a través de la banda transportadora, el AVI realiza la recopilación de información, contabilización y clasificación por color (rojo y azul) mediante la manipulación del brazo robótico, así mismo, mantiene el control de calidad notificando al operador ante algún error eventual en el color de las piezas. La representación del proceso de la línea de producción se muestra en la Figura 10.

3.2. Puesta en marcha

El AVI inicia su función realizando el reconocimiento facial (Figura 11) una vez que el operador ha sido identificado, le pregunta si quiere encender la línea de producción y el operador confirma con un comando de voz, el proceso entra en operación, la banda

Fig. 13. Contabilización de piezas.

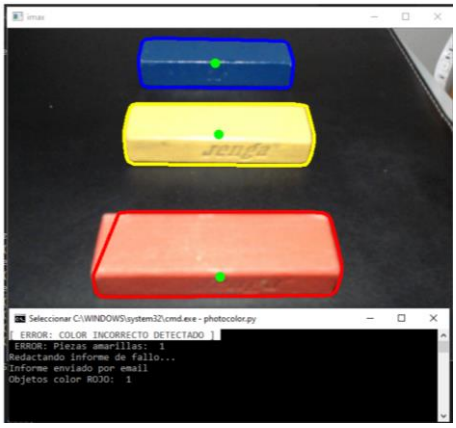


Fig. 14. Pieza de color incorrecto (amarillo).

transportadora avanza y el brazo robot lleva a cabo la selección y manipulación de los productos clasificándolos de acuerdo al color rojo y azul (Figura 12). Al mismo tiempo lleva a cabo la recopilación de información que servirá a la base de datos Figura 13.

Como parte del proceso de control de calidad, el AVI a través del brazo robot es capaz de identificar algún error en el color de las piezas como se muestra en la Figura 14 y actuando en base a su programación ordena la detención de la banda transportadora, notificando del error al operador emitiendo una respuesta audible y redactando un informe con la evidencia que será enviado vía e-mail de manera automática como registro del error (Figura 15). El operador ordena al AVI la continuación del proceso.

Al cambio de turno, un nuevo operador comienza su jornada, le pide al AVI que describa la información del estado del proceso y que muestre por pantalla la información solicitada. El AVI confirma la instrucción y lleva a cabo el despliegue por pantalla al mismo tiempo que genera una respuesta audible con toda la información del proceso (Figura 16).

Fig. 15. Informe de error, e-mail generado por el AVI.

Fig. 16. Generación de respuesta, salida por pantalla y audible.

4. Conclusiones y trabajo a futuro

De las pruebas realizadas en la simulación de un proceso de control de calidad, donde se recreó de manera básica un escenario típico de la Industria 4.0 se puede concluir que la implementación de un asistente virtual inteligente demuestra una importante mejora en:

- La interacción humana con los dispositivos.
- La apertura de nuevos canales de comunicación y control de un proceso.
- Creación de un entorno totalmente conectado, inteligente y colaborativo.

- La aplicación e integración de tecnologías emergentes.
- Oportunidad de escalabilidad.

El asistente virtual inteligente logró integrar con éxito múltiples tecnologías basadas en Inteligencia Artificial y que cumplieron con el objetivo de asistir al usuario, facilitando la comunicación entre el ser humano y los dispositivos que conforman su entorno, dando como resultado una interacción más intuitiva y natural en comparación con las interfaces de usuario tradicionales.

Es de contemplarse para trabajos futuros indagar en el desarrollo e integración de más herramientas y tecnologías especializadas en el campo de la Inteligencia Artificial que logren aumentar las capacidades del asistente, sin embargo, el trabajo obtenido logró estructurar una buena base que facilitará la escalabilidad del proyecto para futuras integraciones, las posibilidades son inmensas.

Finalmente, es importante enfatizar en el desarrollo de propuestas y proyectos que logren la apertura de nuevas vías de comunicación e interacción tecnológica, explorando más allá de lo establecido, apostando a las tendencias de un futuro totalmente conectado, recalcando siempre la valía y vigencia del capital humano en las industrias del futuro.

Referencias

1. Lasi, H., Fettke, P., Kemper, H.: Industry 4.0. Bus Inf Syst Eng 6, pp. 239–242 (2014)
2. Shamim, M., Ghulam M.: Cloud-assisted industrial internet of things (iiot) – enabled framework for health monitoring. Computer Networks, 101, pp. 192–202 (2016)
3. Guato, M., Imacaña, P.: Un breve resumen de lo que son los CPS (Cyber Physical Systems) sus implicaciones y los desafíos futuros. https://www.ticportal.es/expert/cps-cyber-physical-systems-implicaciones-desafios-futuros (2019)
4. Lin, F., Chen, B., Lin, B., Hu, W.: Charging architecture for M2M communications. In: IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, pp. 123–128 (2016)
5. Papcun, P., Kajáti, E., Koziorek, J.: Human machine interface in concept of industry 4.0. In: World Symposium on Digital Intelligence for Systems and Machines (DISA), Kosice, pp. 289–296 (2018)
6. Guzzoni, D., Baur, C., Cheyer, A.: Active, a tool for building intelligent user interfaces (2007)
7. Kim, T.: Short research on voice control system based on artificial intelligence assistant. In: International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–2 (2020)
8. William, P., Marcela. J.: Combined approach using artificial vision and neural networks for facial recognition. In: CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), pp. 1–5 (2017)
9. Azuma, R.: Augmented reality: approaches and technical challenges, fundamentals of wearable computers and augmented reality. pp. 27–63 (2001)

10. Fanuc: LR Mate 200iD. https://www.fanuc.eu/es/es/robots/pagina-filtro-robots/serie-lrmate/lrmate-200-id (2020)
11. Hackaday: Raspberry Pi 3 as a Web Server Using Python IoT https://hackaday.io/project/156072-raspberry-pi-3-as-a-web-server-using-python-iot (2020)
12. Rithika, H., Santhoshi, B.: Image text to speech conversion in the desired language by translating with Raspberry Pi. In: IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4 (2001)
13. Van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: WaveNet: A generative model for raw audio (2016)
14. Jia, X.: Fabric defect detection based on open source computer vision library OpenCV. In: International Conference on Signal Processing Systems, Dalian, V1-342–V1-345 (2010)

ISSN 1870-4069

Análisis comparativo de la calidad de agrupamiento del algoritmo Intuitionistic Fuzzy C-Means

Celia Ramos-Palencia, Dante Mújica-Vargas, Manuel Mejía-Lavalle

Tecnológico Nacional de México,
CENIDET,
México

{celia.ramos17ce, dante.mv}@cenidet.tecnm.mx

Resumen. En este artículo se presenta un estudio comparativo de la calidad del agrupamiento obtenido por el algoritmo Intuitionistic Fuzzy C-Means respecto a los algoritmos Fuzzy C-Means, Gath-Geva, Gustafson Kesel, K-Means y K-Medoid. La evaluación se hace en tareas de reconocimiento de patrones, utilizando las métricas: Xie-Beni, de Dunn, de Partición, Coeficiente de Partición, Entropía de Clasificación, de Separación y alternativo de Dunn. Los resultados indican que el algoritmo Intuitionistic Fuzzy C-Means es más eficiente en la calidad de agrupamiento y convergencia en comparación con los otros algoritmos utilizados para este trabajo.

Palabras clave: Agrupamiento, Intuitionistic Fuzzy C-means, algoritmos clásicos de agrupamiento, calidad de agrupamiento.

Comparative Analysis of the Clustering Quality of the Intuitionistic Fuzzy C-Means Algorithm

Abstract. In this paper we proposed a comparative study of the clustering quality obtained by the Intuitionistic Fuzzy C-Means algorithm about the Fuzzy C-Means, Gath-Geva, Gustafson Kesel, K-Means and K-Medoid algorithms. The evaluation is done in pattern recognition tasks, using the metrics: Xie-Beni, Dunn's, Index, Partition Coefficient, Classification Entropy, Index Separation, and Dunn's Alternative Index. The experimental results show that the Intuitionistic Fuzzy C-Means algorithm is more efficient in the quality of clustering and convergence compared to the other algorithms used in this work.

Keywords: Clustering, Intuitionistic Fuzzy C-means, classic clustering algorithms, clustering quality.

1. Introducción

Clustering o *Agrupamiento* puede ser definido como un método que se utiliza para crear un determinado número de grupos de objetos, de tal forma que los objetos pertenecientes a un grupo son muy similares entre sí y, al mismo tiempo, dichos objetos son distintos respecto a los objetos de otros grupos [1].

La investigación en el área de agrupamiento de datos ha producido una gran cantidad de diferentes métodos que se centran en medir la efectividad de la calidad del agrupamiento y su impacto en el rendimiento de este tipo de algoritmos. En [2], se comparó la efectividad de algunos índices difusos internos ampliamente utilizados, usando el algoritmo Fuzzy C-Means, estando dirigido a investigar los cambios de efectividad de los índices de validez cuando los datos difusos están a diferentes distancias de los centroides. En [3], se mostró un trabajo experimental que compara 30 índices de validez en diversos entornos con características diferentes. Los resultados pueden servir como una guía para seleccionar el índice más adecuado para cada aplicación posible y proporcionar una visión detallada de las diferencias de rendimiento. En [4], se presentó una revisión de los índices de validación difusos disponibles en la literatura, haciendo una comparación exhaustiva junto con el algoritmo Fuzzy C-Means en varias bases de datos ampliamente utilizadas.

En [5], se realizó un estudio del comportamiento de algunos índices en algunas bases de datos con ruido o valores atípicos. Los algoritmos utilizados fueron K-Means, agrupamiento jerárquico, usando el método de Ward y agrupamiento basado en modelos usando mezclas Gaussianas, el algoritmo de maximización de expectativas (EM). El objetivo fue conocer con cual método se obtenían un mejor comportamiento en cuanto al ruido o la presencia de valores atípicos. En [6], se presentó un índice de acuerdo de agrupamiento (CAI) para comparar la disyunción y superposición de la agrupación. Se utilizaron dos índices: ARI y NMI y el propósito fue extender a casos superpuestos, generalizar medidas existentes e introducir medidas novedosas.

En [7], se mencionó que la principal desventaja de K-Means es que el número de los grupos debe ser predifinido, por lo tanto, en este estudio se compararon diferentes índices de validación para la determinación automatizada y reproducible del número óptimo de grupos.

En [8], se evaluaron las características de un conjunto representativo de índices de validación de agrupamiento interno con diferentes bases de datos. El prototipo basado en el marco del agrupamiento incluye estimaciones estadísticas múltiples, clásicas y robustas de la ubicación del grupo, lo que hace este trabajo novedoso. Se utilizaron los algoritmos K-Means, K-medians y K-spatialmeadians En [9], se presentó un análisis comparativo entre las metodologías basadas en máquinas de vectores de soporte y Fuzzy C-Means para agrupar señales de ECG normales y patológicas en un esacio de complejidad de Lempel-Ziv y entropía de Shannon. En [10], se presentó un estudio comparativo para el reconocimiento de patrones con algoritmos de agrupamiento en la clasificación en señales de comunicación analógicas moduladas.

Los algoritmos utilizados fueron K-Means y Fuzzy C-Means. En [11], se hizo una evaluación de las métricas disponibles para comparar algoritmos de agrupamiento. También se definieron algunas restricciones formales intuitivas sobre tales métricas, además de proponer un conjunto de restricciones que una buena métrica de evaluación debe satisfacer en un problema de agrupamiento genérico.

Se pudo observar que la comparación de algunos índices se ha realizado con un solo algoritmo de agrupamiento, en este caso Fuzzy C-Means, o se han hecho estudios comparativos de algunas métricas para su mejora, sin embargo no se ha realizado un estudio para comprobar la calidad de agrupamiento de un algoritmo de agrupamiento difuso intuicionista, por ello se propone realizar un análisis comparativo para medir la calidad de agrupamiento de un algoritmo difuso intuicionista y demostrar que es superior a otros algoritmos clásicos en términos de convergencia y calidad de agrupamiento, para posteriormente poder mejorarlo através de alguna fusión con otras metodologías, por ejemplo, dotarlo de capacidad de aprendizaje con una red neuronal.

La estructura de este documento es de la siguiente manera: en la Sección 2 se presenta una breve descripción de los conceptos utilizados en este documento. En la Sección 3 se presenta la evaluación comparativa de los algoritmos y las métricas en las diferentes bases de datos seleccionadas para este propósito. En la Sección 4 se muestra la descripción de los resultados. Por último se presentan las conclusiones y posibles trabajos futuros.

2. Marco teórico

Dentro de la clasificación de algoritmos de agrupamiento, se encuentran los de agrupamiento difuso, que forman una clase en la cual cada uno de los elementos tiene un cierto grado de pertenencia a los grupos y no solo a uno, como por ejemplo, el clásico K-Means. Este tipo de algoritmos considera que cada elemento se puede agrupar inequívocamente con los elementos de su grupo y que, por lo tanto, no se asemeja al resto de los elementos [12].

2.1. Algoritmo K-Means

El algoritmo K-Means es un método de partición más conocido y utilizado; agrupa los datos en función de su cercanía entre sí de acuerdo con una medida de distancia, por ejemplo, la euclidiana. Minimiza la siguiente función objetivo [1]:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i^{(j)} - c_j\|. \tag{1}$$

En la Figura 1 se muestran los pasos del algoritmo y sus requerimientos.

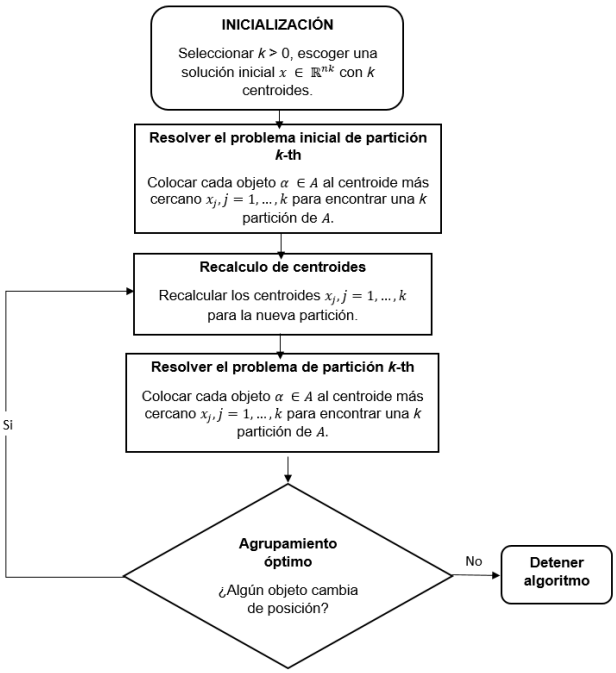


Fig. 1. Algoritmo K-Means.

2.2. Algoritmo Fuzzy C-Means

En Fuzzy C-Means, los datos pueden pertenecer a más de un grupo y también tener diferentes grados de pertenencia en los diferentes grupos. Minimiza la siguiente función objetivo [12]:

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 . \tag{2}$$

En la Figura 2 se observa la descripción de los pasos de este algoritmo.

2.3. Algoritmo Gath Geva

Permite detectar conjuntos de objetos de distintas formas, tamaños y densidades [12,13]. A causa del tipo de distancia que se utiliza (Euclidiana, Manhattan, Mahalanobis, entre otras), los conjuntos no están restringidos a un solo grupo como en el caso de K-Means. Sin embargo, es muy sensible a la inicialización, ya que el algoritmo tiende a converger en óptimos locales. La Figura 3 muestra una descrpción detallada del funcionamiento del algoritmo.

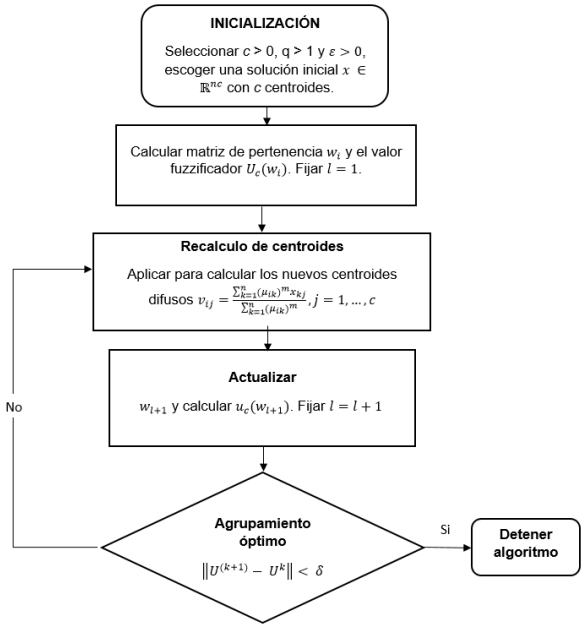


Fig. 2. Algoritmo Fuzzy C-Means.

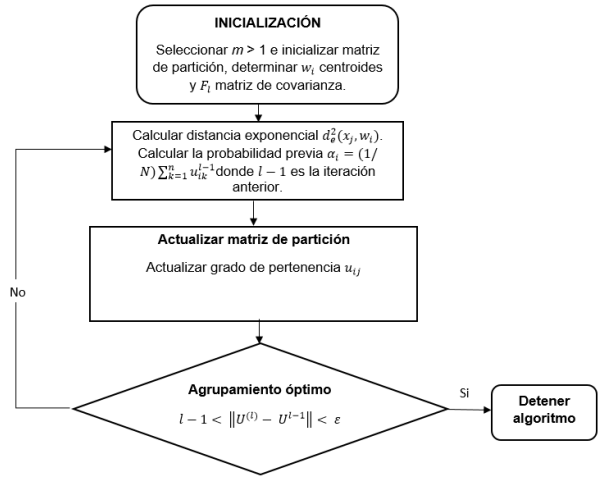


Fig. 3. Algoritmo Gath Geva.

2.4. Algoritmo Gustafson Kessel

Muy similar a FCM, los parámetros m, ε, μ y j son los mismos en los dos algoritmos [14]. Un medoide es un objeto ubicado más al centro del grupo, con

una suma mínima de distancias a otros puntos. En la Figura 4 se presentan los pasos y especificaciones de este método.

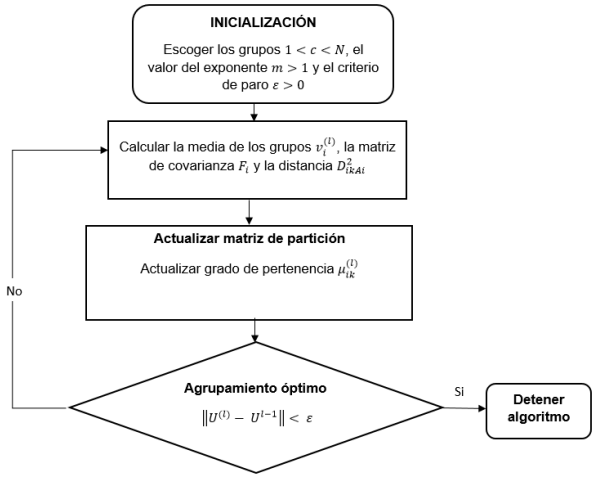


Fig. 4. Algoritmo Gustafson Kessel.

2.5. Algoritmo K-Medoids

Es un algoritmo usado para encontrar los medoides, el cual es un punto situado en el centro de un grupo [15,16]. Además de ser más robusto al ruido y valores atípicos. La Figura 5 muestra los pasos del algoritmo.

2.6. Agrupamiento difuso intuicionista

Gracias a la teoría de conjuntos difusos intuicionistas surge Intuitionistic Fuzzy C-Means (IFCM) o Fuzzy C-Means intuicionista, que además de agregar un grado de pertenencia, asignan a cada elemento un grado de no pertenencia y uno de indecisión, permitiendo que este método sea más poderoso que el tradicional FCM [17].

Los IFS asignan a cada elemento un grado de pertenencia $\mu_F(x)$, un grado de no pertenencia $\nu_F(x)$ y un grado de indecisión $\pi_F(x)$, permitiendo que este método sea más poderoso que el algoritmo difuso tradicional.

Intuitionistic Fuzzy Sets: Dado X como el conjunto de datos, dado $A \subset X$ como un conjunto fijo. Entonces un IFS A en X es expresado de la siguiente manera:

$$A = \{\langle x, \mu_A(x), \nu_A(x)\rangle \,|x \in X\}\,, \tag{3}$$

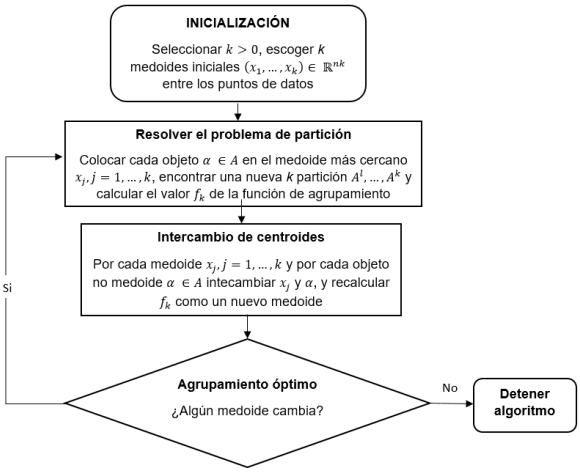


Fig. 5. Algoritmo K-Medoids.

donde las funciones $\mu_A : A \mapsto [0, 1]$ y $A \mapsto [0, 1]$ son los grados de pertenencia y no pertenencia del elemento $x \in X$ al conjunto A, respectivamente, y para cada $x \in X$:

$$0 \leq \mu_A + \nu_A \leq 1. \tag{4}$$

Cada conjunto difuso intuicionista debe ser:

$$A^* = \{\langle x, \mu_A(x), \nu_A(x)\rangle \, | x \in X\}. \tag{5}$$

El siguiente parámetro es el grado de indecisión $\pi_A(x)$, al ser incluido en la expresión anterior, debe tomar la siguiente forma:

$$A^* = \{\langle x, \mu_A(x), \nu_A(x), \pi_A(x)\rangle \, | x \in X\}, \tag{6}$$

donde el grado de indecisión puede ser expresado como $\pi_A = 1 - \mu_A(x) - \nu_A(x)$.

IFCM requiere menor tiempo de ejecución debido a que la cantidad de iteraciones necesarias para lograr un ejercicio de agrupamiento específico corresponde a la precisión requerida. El algoritmo IFCM encuentra una partición difusa de los datos minimizando la siguiente función objetivo:

$$\text{mín}\, J_m\left(X^{IFS}; U, V^{IFS}\right) = \sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij}^m d^2\left(x_j^{IFS} - v_i^{IFS}\right). \tag{7}$$

Sujeto a la condición:

$$\sum_{i=1}^{c} u_{ij} = 1, 1 \leq j \leq N, \tag{8}$$

donde $X^{FIS} = \left(x_1^{IFS}, x_2^{IFS}, \ldots, x_n^{IFS}\right)$ son N datos difusos intuicionistas y cada x_j^{IFS} está representado en términos de conjuntos difusos intuicionistas como $x_j^{IFS} = \{(\mu(x_j), \nu(x_j), \pi(x_j)\}, m(1 < m < \infty)$ es el factor difuso, $c(1 < c < N)$ es el número de grupos, y $V^{IFS} = \left\{v_1^{IFS}, v_2^{IFS}, \ldots, v_c^{IFS}\right\}$ son los centroides de los grupos y cada v_i^{IFS} está representado en términos de conjuntos difusos intuicionistas como $v_i^{IFS} = \{\mu(v_i), \nu(v_i), \pi(v_i)\}, u_{ij}\,(0 \leq u_{ij} \leq 1)$ es el grado de pertenencia y $U = (u_{ij})_{c \times N}$ es la matriz de pertenencia. La matriz de pertenencia U también cumple la condición $0 < \sum_{i=1}^{N} u_{ij} < N, \forall i$.

La pertenencia y los centroides $v_i^{IFS} = \mu(v_i, \nu(v_i, \pi(v_i))$ se calculan de la siguiente manera:

$$u_{ij} = \frac{1}{\sum_{r=1}^{c} \left(\frac{d^2\left(x_j^{IFS}, v_i^{IFS}\right)}{d^2\left(x_j^{IFS}, v_r^{IFS}\right)} \right)^{\frac{1}{m-1}}}, 1 \leq i \leq c, 1 \leq j \leq N. \tag{9}$$

El vector de prototipos está dado por:

$$v_i = \{\mu(v_i), \nu(v_i), \pi(v_i)\}, \tag{10}$$

donde cada uno de los componentes se da de la siguiente manera:

$$\mu(v_i) = \frac{\sum_{j=1}^{N} u_{ij}^m \mu(x_j)}{\sum_{j=1}^{N} u_{ij}^m}, 1 \leq i \leq c, \tag{11}$$

$$\nu(v_i) = \frac{\sum_{j=1}^{N} u_{ij}^m \nu(x_j)}{\sum_{j=1}^{N} u_{ij}^m}, 1 \leq i \leq c, \tag{12}$$

$$\pi(v_i) = \frac{\sum_{j=1}^{N} u_{ij}^m \pi(x_j)}{\sum_{j=1}^{N} u_{ij}^m}, 1 \leq i \leq c. \tag{13}$$

3. Evaluación comparativa

En esta sección se muestra el proceso de experimentación, los datos y las métricas de evaluación utilizados para verificar que el algoritmo IFCM es superior al resto de los otros métodos mencionados, en cuanto a calidad de agrupamiento y convergencia

3.1. Evaluación

Se utilizaron bases de datos reales obtenidas del repositorio de la UCI [18]. En la Tabla 1 se muestran la bases de datos utilizadas y el total de objetos de cada una. Para verificar la calidad del agrupamiento de los algoritmos y hacer el estudio comparativo, se utilizaron diferentes métricas: Coeficiente de Partición (PC), Clasificación de la Entropía (CE), Índice de Partición (SC), Índice de Separación (S), Xie-Beni (XB), de Dunn (DI) y Alternativo de Dunn (ADI).

Tabla 1. Bases de Datos.

Nombre	Número de objetos
Iris	150
Wine	175
Ecoli	336
Breast Cancer	696
RSSI	1420
Skin	9568
Carbon	10721
Bank	45210
Power Plant	245057
Buzz	260059

1. *Coeficiente de Partición*(PC): Mide la cantidad de traslape entre grupos. Fue definido por Bezdek. [19,20]:

$$PC(c) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \left(\mu_{ij}\right)^2, \tag{14}$$

donde μ_{ij} es la pertenencia del i-ésimo grupo. El número óptimo de grupos es indicado por un valor máximo de este índice.

2. *Entropía de Clasificación* (CE): Este índice mide la falta de difusividad de los grupos [9]:

$$CE(c) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij} \log\left(\mu_{ij}\right). \tag{15}$$

3. *Índice de partición* (SC): Es la división de la suma de separación y compactibilidad de los grupos; a su vez, es una suma de medidas individuales de la validez de los grupos que es normalizada a través de la división de la cardinalidad difusa de cada grupo. El número óptimo de grupos se logra al minimizar este índice [9]:

$$SC(c) = \sum_{j=1}^{c} \frac{\sum_{i=1}^{n} \left(\mu_{ij}\right)^m \left\|x_i - c_j\right\|^2}{n_j \sum_{k=1}^{c} \left\|c_k - c_j\right\|^2}. \tag{16}$$

4. *Índice de separación* (S): A diferencia del anterior, este índice utiliza una separación de distancia mínima para validar el agrupamiento [9]:

$$S(c) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{c} \left(\mu_{ij}\right)^2 \left\|x_i - c_j\right\|^2}{n \min_{k,j} \left\|c_k - c_j\right\|^2}. \tag{17}$$

5. *Índice de Xie y Beni* (XB): Ayuda a medir la relación de la variación total dentro de los grupos y la separación de los mismos:[21]:

$$XB(c) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{c} (\mu_{ij})^m \left\| x_i - c_j \right\|^2}{n \min_{i,j} \left\| x_i - c_j \right\|^2}. \tag{18}$$

6. *Índice de Dunn* (DI): Este índice es originalmente propuesto para usarse en la identificación de "grupos compactos y bien separados" [21]:

$$DI(c) = \min_{j \in c} \left\{ \min_{i \in c, i \neq j} \left\{ \frac{\min_{x \in C_j, y \in C_i} d(x,y)}{\max_{k \in c} \{\max_{x,y \in C} d(x,y)\}} \right\} \right\}, \tag{19}$$

el principal inconveniente de este índice es que su carga computacional se vuelve muy expansiva en función del incremento de n y c.

7. *Índice Alternativo de Dunn* (ADI): En este caso, el cálculo se vuelve más simple que en el índice de Dunn tradicional, cuando la función de dissimilaridad entre dos grupos ($\min_{x \in C_i, y \in C_j} d(x,y)$) [21]:

$$d(x,y) \geq \left| d\left(y, v_j\right) - d\left(x, v_j\right) \right|, \tag{20}$$

donde v_j es el centroide del grupo del j-ésimo centroide:

$$ADI(c) = \min_{i \in c} \left\{ \min_{j \in c, i \neq j} \left\{ \frac{\min_{x_i \in C_i, x_j \in C_j} \left| d\left(y, v_j\right) - d\left(x_i, v_j\right) \right|}{\max_{k \in c} \{\max_{x,y \in C} d(x,y)\}} \right\} \right\}. \tag{21}$$

4. Resultados

En esta sección se presenta la evaluación comparativa de algoritmos de agrupamiento clásicos y el algoritmo Intuitionistic Fuzzy C-Means, a través de gráficas comparativas utilizando las bases de datos mencionadas anteriormente, con los siguientes parámetros: en el caso de los difusos m=2, y $\epsilon = 1e-6$. Como nota, este criterio de paro no es aplicable a todos los algoritmos ya que durante los pasos del mismo, algunos de los valores obtenidos son *infinitos* o *NaN*, que significa no un número, por ejemplo, hacer una operación inusual, como dividir entre cero.

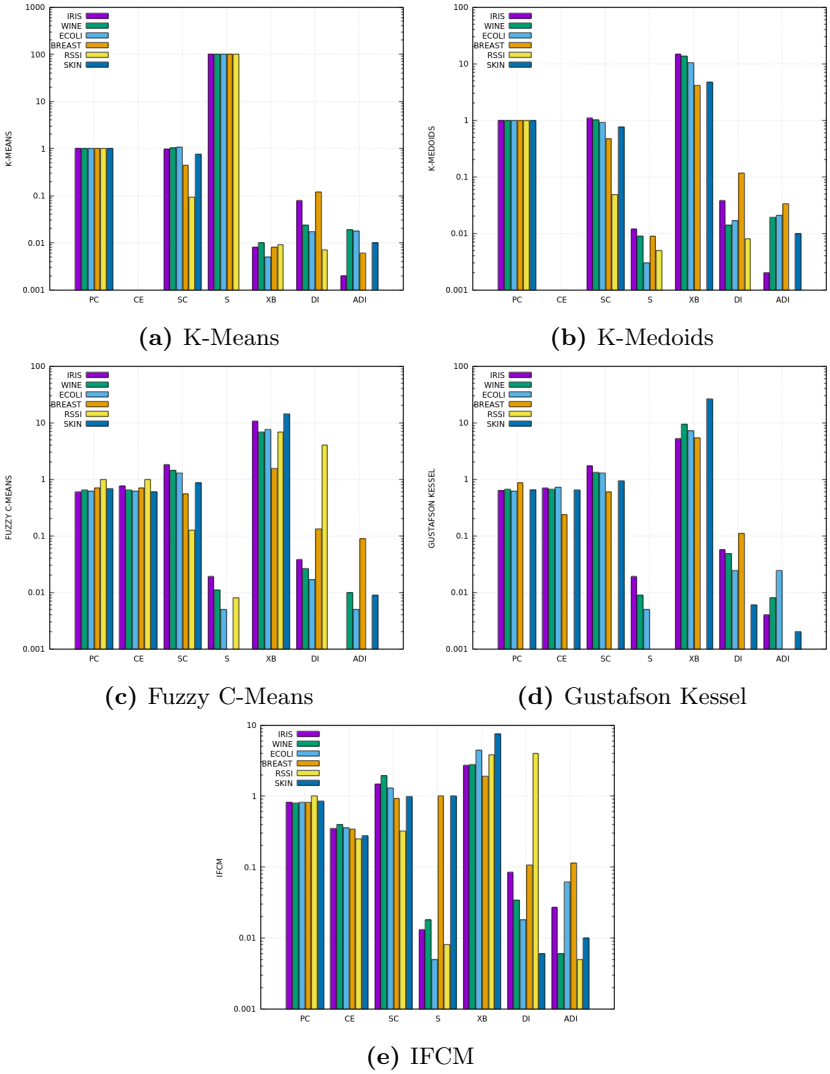
La experimentación se llevó a cabo en MATLAB R2018, usando el *Toolbox Clustering* [22] para los algoritmos clásicos, IFCM fue programado por separado; una computadora con 1 Tb de disco duro de almacenamiento, 8Gb de RAM, procesador iCore i7 de 7ma generación, y Linux Xubuntu 18.04.

4.1. Métricas e iteraciones

En la Tabla 2 se observan los resultados de cada métrica con la ejecución de cada uno de los algoritmos, para cada base de datos; y en las gráficas de la Figura 6 se muestra cómo con el algoritmo IFCM, los resultados obtenidos en cada métrica tienden a alcanzar los valores óptimos. Anteriormente se mencionó

Tabla 2. Resultados cuantitativos.

Base de datos	Metrica	Kmeans	Kmedoids	FCM	GG	GK	IFCM
	PC	1	1	0.59	0.98	0.63	0.82
	CE	NaN	NaN	0.76	NaN	0.69	0.34
	SC	1.27	0.87	1.80	NaN	1.74	1.47
Iris	S	0.009	0.006	0.01	NaN	0.01	0.01
	XB	8.38	Inf	10.58	NaN	5.20	2.70
	D	0.04	0.04	0.03	0.05	0.05	0.08
	ADI	0.03	0.001	0.001	0.003	0.004	0.02
	PC	1	1	0.80	-	0.89	0.79
	CE	NaN	NaN	0.38	-	0.17	0.39
	SC	0.21	0.22	0.49	-	0.000006	1.95
Wine	S	0.001	0.001	0.004	-	0.0000006	0.01
	XB	47.6	Inf	36.44	-	48	2.77
	D	0.03	0.01	0.01	-	0.004	0.03
	ADI	0.00001	0.000007	0.000001	-	0	0.006
	PC	1	1	0.61	-	0.62	0.80
	CE	NaN	NaN	0.73	-	0.72	0.35
	SC	1.07	0.88	1.27	-	1.27	1.30
Ecoli	S	0.005	0.003	0.005	-	0.005	0.005
	XB	11.9	Inf	7.52	-	7.20	4.46
	D	0.01	0.02	0.01	-	0.02	0.01
	ADI	0.01	0.001	0.005	-	0.02	0.06
	PC	1	1	0.70	-	0.86	0.81
	CE	NaN	NaN	0.56	-	0.23	0.33
	SC	0.47	0.52	0.55	-	0.60	0.93
Breast Cancer	S	0.001	0.001	0.001	-	0.001	0.001
	XB	3.75	Inf	1.54	-	5.37	1.89
	D	0.10	0.11	0.13	-	0.11	0.10
	ADI	0.03	0.11	0.09	-	0.00001	0.11
	PC	1	1	0.99	-	-	0.99
	CE	NaN	NaN	0.01	-	-	0.001
	SC	0.04	0.08	0.12	-	-	0.32
RSSI	S	0.000005	0.000008	0.000008	-	-	0.00004
	XB	NaN	Inf	6.92	-	-	3.79
	D	0.082	0.007	4.03	-	-	4.03
	ADI	0	0	0	-	-	0
	PC	1	1	0.67	-	-	0.87
	CE	NaN	NaN	0.60	-	-	0.27
	SC	0.76	0.66	0.87	-	-	0.98
Skin	S	0.00001	0.000009	0.00001	-	-	0.00001
	XB	47.4	Inf	142.16	-	-	7.61
	D	0.001	0.001	0.001	-	-	0.00006
	ADI	0.01	0.005	0.009	-	-	0.001
	PC	1	1	0.64	0.90	0.64	0.82
	CE	NaN	NaN	0.70	0.15	0.65	0.34
	SC	0.61	0.61	0.77	1.55	1.11	0.99
Carbon	S	0.00005	0.000005	0.000007	0.00001	0.00001	0.000009
	XB	44.8	Inf	18.42	1.58	28.09	5.85
	D	0.003	0.002	0.001	0.001	0.001	0.001
	ADI	0.001	0.001	0.004	0.02	0.004	0.02
	PC	1	1	0.74	-	-	0.88
	CE	NaN	NaN	0.47	-	-	0.20
	SC	0.99	1.05	1.44	-	-	0.90
Bank	S	0.000003	0.000003	0.000004	-	-	0.000003
	XB	9.23	Inf	10.76	-	-	7.79
	D	0.001	0.001	0.01	-	-	0.00001
	ADI	0	0	0	-	-	0
	PC	1	1	0.99	-	-	0.88
	CE	NaN	NaN	0.01	-	-	0.20
	SC	0.49	0.50	0.12	-	-	0.90
Power Plant	S	0.0000002	0.0000002	0.000008	-	-	0.000003
	XB	73.02	Inf	6.92	-	-	7.79
	D	0.002	0.028	4.03	-	-	0.00001
	ADI	0.003	0.003	0	-	-	0
	PC	1	1	0.95	-	-	0.34
	CE	NaN	NaN	0.08	-	-	1.18
	SC	0.43	0.50	0.59	-	-	3.43
Buzz	S	0.000002	0.000002	0.000003	-	-	0.12
	XB	159.42	Inf	367.27	-	-	8.78
	D	0.00004	0.0028	0.00005	-	-	0.000001
	ADI	0.0000001	0.0037	0.0000001	-	-	0.0000002



(a) K-Means **(b)** K-Medoids

(c) Fuzzy C-Means **(d)** Gustafson Kessel

(e) IFCM

Fig. 6. Representación gráfica de los resultados cuantitativos.

que debido al criterio de paro, algunos algoritmos se ciclaban y no terminaban de iterar, es por ello, que en la Tabla 2 y en las subfiguras de la Figura 6, se observan lugares vacíos.

Esta situación no se presentó con IFCM; con este método cada una de las bases de datos pudo ser ejecutada sin problema, con un menor tiempo de ejecución (número de iteraciones), las cuales se redujeron en promedio entre un 93 % y un 99 % respecto al número de iteraciones realizadas por los otros algoritmos, y/o con una mayor calidad de agrupamiento.

5. Conclusiones y trabajos futuros

En este artículo se presentó un estudio comparativo de la calidad del agrupamiento obtenido por el algoritmo Intuitionistic Fuzzy C-Means y se realizó una comparación con algunos métodos de agrupamiento utilizados en la literatura. La evaluación fue hecha en tareas de reconocimiento de patrones, utilizando índices de validación. Los resultados indican que el algoritmo IFCM reduce hasta en un 93 % y 99 % la cantidad de iteraciones necesarias para converger, en cuanto a la calidad de agrupamiento, en la mayoría de las métricas se alcanzan los valores cercanos a los óptimos para una buena calidad de agrupamiento. Con estos resultados se pretende, como trabajo futuro, dotar de aprendizaje neuronal a este algoritmo ya que es eficiente en las tareas de agrupamiento y reconocimiento de patrones.

Agradecimientos. Los autores de este trabajo expresan su gratitud al CONACYT, así como al Tecnológico Nacional de México/CENIDET por el financiamiento a través del proyecto "Controlador Difuso para ajuste de coeficientes de rigidez de un modelo deformable para simulación en tiempo real de los tejidos del hígado humano".

Referencias

1. Mújica-Vargas, D.: Segmentación de imágenes utilizando algoritmos robustos de agrupamiento difuso. Tesis Doctoral, Instituto Politécnico Nacional (2013)
2. Cebeci, Z.: Comparison of internal validity indices for fuzzy clustering. Journal of Agricultural Informatics, 10(2) (2019)
3. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), pp. 243–256 (2013)
4. Wang, W., Zhang, Y.: On fuzzy cluster validity indices. Fuzzy Sets and Systems, 158(19), pp. 2095–2117 (2007)
5. Guerra, L., Robles, V., Bielza, C., Larrañaga, P.: A comparison of clustering quality indices using outliers and noise. Intelligent Data Analysis, 16(4), pp. 703–715 (2012)
6. Rabbany, R., Zaïane, O.R.: Generalization of clustering agreements and distances for overlapping clusters and network communities. Data Mining and Knowledge Discovery, 29(5), pp. 1458–1485 (2015)
7. Abualhaj, B., Weng, G., Ong, M., Attarwala, A.A., Molina, F., Büsing, K., Glatting, G.: Comparison of five cluster validity indices performance in brain [18F]FET-PET image segmentation using k-means. Medical Physics, 44(1), pp. 209–220 (2017)
8. Hämäläinen, J., Jauhiainen, S., Kärkkäinen, T.: Comparison of internal clustering validation indices for prototype-based clustering. Algorithms, 10(3), pp. 105 (2017)
9. Villazana, S., Arteaga, F., Seijas, C., Rodriguez, O.: Estudio comparativo entre algoritmos de agrupamiento basado en svm y c-medios difuso aplicados a señales electrocardiográficas arritmicas. Revista Ingeniería, pp. 16–24 (2012)

10. Guldemir, H., Sengur, A.: Comparison of clustering algorithms for analog modulation classification. Expert Systems with Applications, 30(4), pp. 642–649 (2006)
11. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval, 12(5), pp. 613–613 (2009)
12. Ramos-Palencia, C.: Mejora de un algoritmo de agrupamiento del estado del arte mediante heurísticas aplicadas a la mejora del algoritmo K-Means. Tesis de Maestría, Tecnológico Nacional de México (2019)
13. Lange, O., Meyer-Baese, A., Hurdal, M., Foo, S.: A comparison between neural and fuzzy cluster analysis techniques for functional MRI. Biomedical Signal Processing and Control, 1(3), pp. 243–252 (2006)
14. Miller, D.J., Nelson, C.A., Cannon, M.B., Cannon, K.P.: Comparison of fuzzy clustering methods and their applications to geophysics data. Applied Computational Intelligence and Soft Computing, pp. 1–16 (2009)
15. Arora, P., Deepali, D., Varshney, S.: Analysis of k-means and k-medoids algorithm for big data. Procedia Computer Science, 78, pp. 507–512 (2016)
16. Park, H.S., Jun, C.H.: A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications, 36(2), pp. 3336–3341 (2009)
17. Xu, Z.: Intuitionistic fuzzy clustering algorithms. In: Intuitionistic Fuzzy Aggregation and Clustering Studies in Fuzziness and Soft Computing, pp. 159–267 (2012)
18. Lichman, M.: UCI Machine learning repository. (2013)
19. Bezdek, J.C.: Numerical taxonomy with fuzzy sets. Journal of Mathematical Biology, 1(1), pp. 57–71 (1974)
20. Bezdek, J.C.: Cluster validity with fuzzy sets. Journal of Cybernetics, 3(3), pp. 58–73 (1973)
21. Desgraupes, B.: Clustering indices. University of Paris Ouest-Lab Modal'X, 1 (2013)
22. Clustering Toolbox: MathWorks. MATLAB (2012)
23. Gustafson, D., Kessel, W.: Fuzzy clustering with a fuzzy covariance matrix. In: IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes (1978)

ISSN 1870-4069

Implementación en hardware de un algoritmo genético para resolver un problema combinatorio

Eliseo Díaz-Nacar, Katya Rodríguez-Vázquez

Universidad Nacional Autónoma de México,
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
México

{eliseo.diaz, katya.rodriguez}@iimas.unam.mx

Resumen. En el presente trabajo, se realiza la implementación en hardware mediante FPGA´s, de un Algoritmo Genético Simple (GA, por sus siglas en inglés *genetic algorithm*), para resolver problemas de tipo combinatorio, que en este caso, ilustra la solución del problema de ubicar 8 reinas en un tablero de ajedrez sin que haya ataque entre ellas. Así mismo se realiza una comparación de desempeño de la implementación en hardware, con respecto a la implementación del mismo problema en software mediante MATLAB. Los procesos que integran el GA se implementan mediante módulos de hardware, así mismo también se implementaron los módulos que se requieren para operar el GA, tales como Generación Aleatoria de la Población Inicial, Módulo de Despliegue de Resultados, Módulos de Memoria, Buffers para los Módulos de Memoria. Los resul,tados se presentan de manera gráfica a través de un monitor VGA. La implementación de todo el algoritmo se realiza empleando lenguaje VHDL y una tarjeta de desarrollo SPARTAN3 de DIGILENT, la cual contiene un FPGA SPARTAN 3 de XILINIX. También se presentan mediante tablas, los resultados de las pruebas de desempeño realizadas a la implementación en hardware y la implementación en software.

Palabras clave: FPGA, algoritmos genéticos, problemas combinatorios.

A Hardware Implementation of a Genetic Algorithm for Solving Combinatorial Problems

Abstract. In this work, the hardware implementation is carried out by means of FPGA's, of a Simple Genetic Algorithm (GA), to solve combinatorial problems, which, in this case, illustrates the solution of the problem of placing 8 queens on a chess board without any attack between them. Likewise, a comparison of performance of the implementation in hardware is made, with respect to the implementation of the same problem in software through MATLAB. The processes that integrate the GA are implemented through hardware modules. Likewise, the modules that are required to operate the algorithm were also implemented, such as random generation of the initial population, results display module, memory modules, buffers for memory modules. The results are presented graphically through a VGA monitor. The implementation of the entire

algorithm is carried out using VHDL language and a SPARTAN3 development card from DIGILENT, which contains a SPARTAN 3 FPGA from XILINIX. The results of the performance tests of the hardware implementation and the software implementation are also presented in tables.

Keywords: FPGA, genetic algorithm, combinatorial problems.

1. Introducción

Un GA es un proceso de optimización muy poderoso, inspirado en la evolución natural tal como lo establece Goldberg [1]. Los GA fueron creados por John Holland en la década de los 60's, basados en los procesos naturales de evolución y selección natural tales como selección, cruza y mutación. Estos algoritmos encuentran su principal aplicación en problemas de búsqueda y optimización de problemas paramétricos y combinatorios [2].

Los GA simulan el proceso de evolución natural. Tal simulación requiere de una codificación del dominio del problema por lo que se requiere determinar el espacio en el que se encuentran las posibles soluciones al problema que se pretende resolver u optimizar. Para obtener la solución del espacio de búsqueda se requiere de un proceso de evaluación, el cual se aplicará a una población inicial de soluciones. El GA inicia con la generación de la población inicial y el proceso de evaluación, el cual, se realiza a través de una función objetivo que modela la solución del problema. Después del proceso de evaluación, se obtiene el resultado de dicha evaluación para cada miembro de la población inicial.

Con la población inicial ya evaluada, se aplican los siguientes procesos de forma iterativa: *Selección*, de este proceso se obtienen los mejores individuos (soluciones) de la población, de acuerdo a los resultados de la evaluación, el proceso de selección puede emplear varios métodos como ruleta, torneo etc. En este trabajo se empleó torneo, el cual consiste en asignar parejas de forma aleatoria entre los miembros de la población, de cada pareja se selecciona el individuo mejor evaluado. A esta nueva población (padres), se aplica el proceso de *Cruza* por parejas obteniendo una nueva población (hijos). A esta población de hijos se aplica el proceso de *Mutación* obteniendo una población mutada (nueva generación), con características que se espera, se adapten mejor al problema [3]. Este proceso genera en cada iteración una nueva generación de nuevos individuos con características, que se espera, sean cada vez mejores, para la solución del problema, aunque no siempre sucede y puede ocurrir que el algoritmo ya no encuentre una mejor solución, en cuyo caso es conveniente iniciar el algoritmo nuevamente. Este proceso iterativo presume un consumo de tiempo considerable de procesamiento en un equipo de cómputo; dicho tiempo está en función, en gran medida, de la complejidad del problema que se pretende resolver y del software empleado.

La forma de implementación de un GA puede ser mediante un lenguaje de alto nivel o software especializado como MATLAB. Para dar solución al problema del tiempo de proceso, se han realizado implementaciones en hardware a través de diferentes plataformas tal como se describe en la sección 2 del presente trabajo. Para la implementación en hardware se emplea Lógica Programable la cual generalmente se realiza mediante FPGA's.

Un FPGA (FIELD PROGRAM GATE ARRAY) es un dispositivo que contiene miles o incluso millones de transistores cuya interconexión se puede configurar para implementar funciones lógicas, sumas, restas, hasta complejos filtros digitales y sistemas de detección y corrección de errores. Aeronaves, automóviles, radares, misiles, son algunos de los sistemas que emplean FPGA´s [4]. Los principales fabricantes de FPGA's proveen plataformas de desarrollo y herramientas de software para efectuar la implementación física dentro del FPGA. Uno de los lenguajes empleados para el desarrollo de sistemas en FPGA'S es VHDL que es un lenguaje de descripción de hardware.

En la sección 2 del presente trabajo se hace un resumen de los trabajos previos relacionados. En la sección 3 se presenta el diagrama esquemático de la implementación del GA, se detallan las señales de entrada y salida de cada módulo que lo integran, así mismo se presentan los módulos necesarios para la operación del algoritmo tales como el módulo de control y el de despliegue de resultados. En la sección 4 se hace una descripción del funcionamiento del sistema. En la sección 5 se realizan las pruebas correspondientes obteniendo el tiempo requerido para llegar a la solución del problema, así mismo se hace una comparación de desempeño de la implementación del mismo problema empleando MATLAB. En la sección 6 se presenta el trabajo futuro para mejorar el desempeño de la implementación en hardware y poderlo aplicar a problemas más complejos. Finalmente, se presentan las conclusiones del desarrollo del presente trabajo.

2. Trabajos relacionados

Los trabajos que se mencionan a continuación son solo una muestra de la cantidad de trabajos que se han realizado sobre la implementación de un GA en Hardware. Scott *et al.* [5] presentan en 1995 uno de los primeros trabajos relacionados con la implementación en hardware de un GA. Presentan resultados del desempeño de la implementación versus una versión en software. Proporcionan la arquitectura empleada mediante diagramas de bloques, cabe aclarar que emplean pipeline y paralelismo para mejorar el desempeño de la implementación en hardware.

En 2000 Tufte y Haddow [6] presentan la implementación de un filtro digital dinámico cuyos coeficientes de operación se obtienen mediante un GA implementado en un FPGA, en la implementación del GA emplean un arreglo en PipeLine. En el trabajo reportan diagramas de bloques y los resultados los presentan mediante gráficas de respuesta del filtro de amplitud frecuencia y amplitud tiempo aplicando diversas señales de entrada.

En 2001, Aporntewan y Chongsttvatana desarrollaron la implementación de un GA compacto [7]. En este trabajo realizaron la implementación mediante FPGA empleando VHDL, presentando diagramas de bloques y pseudocódigos de la implementación, así mismo presentaron una versión de software realizada en lenguaje "C" con la finalidad de efectuar una comparación de desempeño reportando un *speedup* de 1000.

Barry Shackleford *et al* [8] en 2001 presentan una arquitectura en pipe-line de un GA en FPGA's con la finalidad de acelerar la operación del GA. Presentan dos prototipos de implementación usando diferentes plataformas y frecuencias de reloj.

Para probar la implementación resolvieron algunos problemas como el problema del plegamiento de proteínas. Los resultados reportados presentan una mayor rapidez empleando pipe-line logrando evaluar un cromosoma por ciclo de reloj.

En 2004, Tang y Yip [9] presentan la implementación de un GA usando 2 FPGA's, describen los módulos que integran el algoritmo, proporcionan simulaciones y diagramas de bloques de la implementación. Plantean la implementación en paralelo y pipeline del algoritmo. Para las pruebas emplean una función simple de optimización modificada de la ecuación de línea recta de Witte y Holst.

Mostafa *et al.* en 2004 [10] realizan la implementación de un GA en hardware empleando FPGA's y VHDL, dicha implementación posee una arquitectura paralela y pipeline. Los módulos del algoritmo los diseñaron de forma independiente e implementaron un protocolo de *handshake* para su comunicación. Las pruebas del algoritmo las realizan aplicándolo a 3 problemas diferentes: interpolación lineal, procesamiento de datos de un termistor y la aceleración de un vehículo. Los resultados de las pruebas los presentan mediante gráficas.

Nonel Thirer [11] en 2012 presenta el artículo *About the FPGA Implementation of a Genetic Algorithm for solving Sudoku Puzzles*, en el cual presenta la posibilidad de la implementación para resolver un problema combinatorio como es el Sudoku y de igual forma presenta algunas adaptaciones de los módulos del GA para su implementación en un FPGA.

Tuncer y Yildirim [12] presentaron en 2016 el artículo *Design and Implementation of a Genetic Algorithm IP core on an FPGA for path planning of mobile robots*, en el cual describen la implementación de cada módulo del GA y su implementación física en un robot que ejecuta un recorrido a través de un ambiente de prueba.

En 2016, Alansi *et al.* [13] publicaron un artículo donde plantean el desarrollo de una herramienta adaptativa basada en un GA para sistemas SDMA-OFDM (GASOS, nombré asignado a la herramienta) para mejorar el rendimiento y la complejidad computacional en casos de escenarios multiusuario totalmente cargados y sobrecargados. Presentan los detalles de la implementación con todos los módulos requeridos mediante diagramas de bloques.

Guo *et al.* en 2016 [14] proponen en este artículo un sistema de algoritmos genéticos paralelos en hardware, como Field-Programmable-Gate-Arrays (FPGA's). El enfoque apunta a múltiples FPGA mediante la exploración de diferentes áreas de búsqueda del mismo espacio de solución con diferentes comportamientos. Cada FPGA contiene un GA optimizado y personalizable que puede configurarse utilizando parámetros de tiempo de ejecución, eliminando la necesidad de una compilación costosa. Los experimentos en tres problemas, muestran el alto rendimiento de la propuesta, con una aceleración 30 veces mayor en comparación con una implementación basada en CPU multinúcleo.

En 2017, Thorbole *et al.,* presentaron una implementación en hardware de un GA para la detección y predicción de ataques de epilepsia, lo cual se realiza mediante el uso de métodos analíticos aplicados a las características que se extraen de las señales electroencefalográficas (EEG) [15]. En el trabajo, la detección y predicción de ataques se realiza mediante dos pasos: la extracción de características y la clasificación de características. La extracción de características se realiza mediante el uso de filtrado promedio de pico a pico móvil.

La clasificación de las características de las convulsiones se realiza combinando un enfoque de redundancia mínima de máxima relevancia (mRMR) para la selección de características de las convulsiones y el GA para la clasificación de las convulsiones. El preprocesamiento se realiza en MATLAB y la implementación en hardware del GA se realiza en FPGA. En los resultados se informa sobre el conjunto de datos EEG estándar y la simulación de hardware del GA en FPGA.

Praveena *et al*. en 2017 desarrollan un nuevo algoritmo de segmentación de imágenes médicas tales como imágenes de retina e imágenes cardiacas, con diferentes umbrales el cual se basa en un GA implementado en un FPGA [16]. La arquitectura FPGA propuesta para la segmentación de imágenes es un algoritmo eficiente tanto en tiempo como en energía. Presentan diagramas de bloques de la implementación así como tablas donde presentan el área extraída y el tiempo requerido.

Peker en 2018 propone un IP core de (GA) de propósito general, totalmente personalizable implementado en un FPGA usando una arquitectura de pipeline y arquitecturas paralelas para acelerar el proceso del GA [17]. Los operadores del GA y las funciones de desempeño están diseñados en una estructura modular para permitir el uso de estos módulos de forma asincrónica. El código VHDL de la implementación está escrito con parámetros genéricos para permitir la personalización de casi todos los parámetros del IP Core FPGA propuesto según el problema. El sistema propuesto se utilizó para resolver el problema del agente viajero (TSP). En los experimentos, el autor concluye que el núcleo propuesto mejoró tanto los ciclos de reloj necesarios para iterar una generación como la velocidad de convergencia de las implementaciones de GA existentes.

Torquato y Fernandes proponen una implementación totalmente paralela de un GA implementado en FPGA's [18]. La optimización del tiempo de procesamiento del sistema es el objetivo principal de este trabajo. Se analizan los resultados asociados con el tiempo de procesamiento y la ocupación del área (en FPGA) para varios tamaños de población. Las pruebas se realizaron en la optimización de funciones de dos variables. Los resultados mostraron que la implementación paralela completa de GA logró un rendimiento de aproximadamente 16 millones de generaciones por segundo. También en el trabajo presentan los diagramas de bloques de todos los módulos que integran la arquitectura.

Sunitha y Pradeep presentan la implementación de los módulos que integran el GA basados en una estructura flexible que dinámicamente puede realizar tres tipos de codificación binaria, valor real y enteros [19]. Las estructuras las diseñan y sintetizan usando lenguaje VHDL, simulan con ModelSim y posteriormente lo implementan en FPGA's. Presentan los diagramas de la implementación de cada módulo, así como la simulación de estos mediante la herramienta de implementación en los FPGA´s.

Los trabajos anteriores son solo una muestra de la gran cantidad de aplicaciones en las que se puede usar un GA implementado en hardware, en los cuales, por lo general se emplean FPGA's. Los resultados obtenidos en la mayoría de los trabajos descritos poseen una mayor rapidez en la ejecución del algoritmo, comparando sus respectivas implementaciones en software. La presentación de los resultados es a través de simulaciones, tablas y gráficas, resolviendo generalmente problemas de tipo paramétrico y algunos combinatorios.

3. Implementación en hardware

La implementación del GA se realizó empleando una tarjeta de desarrollo STARTER KIT SPARTAN 3 de DIGILENT [20], la cual contiene un FPGA Spartan 3 de XILINIX así como diversos bloques de memoria, puertos de comunicación paralela y serial, un display de LCD de 2 líneas y 16 caracteres, interruptores diversos y una barra de 8 leds. El software empleado para configuración y comunicación con la tarjeta es proporcionado por XILINIX.

El lenguaje empleado para programar y configurar las funciones requeridas por el algoritmo en el FPGA es VHDL. VHDL es un lenguaje para describir sistemas electrónicos digitales. Se desarrolló bajo los auspicios del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) y se adoptó en forma de Manual de Referencia de Lenguaje VHDL Estándar, IEEE estándar 1076 1987. Como todos los estándares IEEE, están sujetos a revisión cada cinco años. Los comentarios y sugerencias de los usuarios del estándar de 1987 fueron analizados por el grupo de trabajo IEEE responsable de VHDL y en 1992 se propuso una versión revisada del estándar y esta fue finalmente adoptada en 1993 [21].

Los módulos que integran la implementación se enlistan a continuación:

- módulo de control,
- generador de población inicial,
- módulo de lectura de memoria de población,
- módulo de memoria de población,
- módulo de memoria de evaluación,
- módulo de memoria de selección,
- buffer de memoria de población,
- buffer de memoria de evaluación,
- buffer de memoria de selección,
- módulo de evaluación,
- módulo de selección,
- módulo de cruza,
- módulo de mutación,
- módulo de control de monitor VGA.

Se muestra el diagrama esquemático de la implementación (Fig. 1) del GA, en el cual se indica la interconexión de todos los módulos que la integran. Los módulos de memoria a los que se hace referencia fueron implementados con los bloques de memoria RAM interna que posee el FPGA, no se usaron los módulos de memoria externos con los que cuenta la tarjeta, únicamente los recursos propios del FPGA. En el esquema de la Fig. 1 se muestran también líneas de entrada y salida a la implementación las cuales se describen a continuación.

- ***CKop*** Señal de reloj de operación que proviene de un oscilador externo a la tarjeta que es la base de operación de la implementación.
- ***CK*** Señal de reloj de 50MHz que provee la tarjeta de desarrollo. Se emplea para generar números aleatorios requeridos en el funcionamiento del algoritmo.

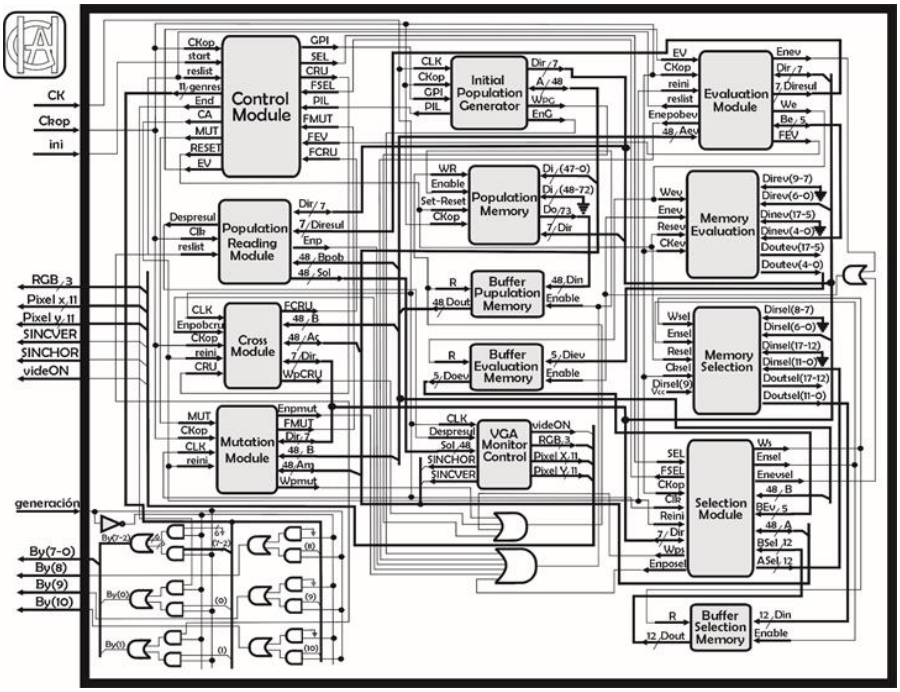


Fig. 1. Diagrama esquemático de la implementación en FPGA.

— ***ini*** Señal externa proveniente de un interruptor que provee la misma tarjeta para dar inicio de forma manual al inicio de la operación del algoritmo.

— ***generación*** Línea de entrada proveniente de un de los interruptores de la tarjeta de desarrollo, que le indica a la implementación la información a desplegar en la barra de 8 leds de la tarjeta de desarrollo más 3 leds adicionales externos para un total de 11 bits. Cuando esta línea se encuentra en cero, los leds correspondientes a los bits "0" y "1" indicarán *"Fin de Algoritmo"* y *"Resultado Listo"* respectivamente, y cuando la línea se encuentra en uno, en los 11 leds se desplegará en binario el número de generación en la cual se encontró la solución.

— ***By(0-10)*** Líneas de salida a la barra de 11 Leds (8 de la tarjeta y 3 externos).

— ***RGB, PIXEL X, PIXEL Y, SINCVER, SINCHOR y VIDEON*** son señales de salida para el control del monitor VGA, cuya conexión se encuentra habilitada en la misma tarjeta de desarrollo.

4. Operación del algoritmo genético en hardware

La operación del algoritmo inicia cuando la señal de entrada *"ini"* adquiere un nivel alto de voltaje, la cual activa la operación del módulo de control. A continuación, se describe la operación de todos y cada uno de los módulos.

Tabla 1. Descripción del Módulo de Control.

MODULO DE CONTROL			
Paso	**Señales de Entrada Activas**	**Señales de Salida Activas**	**Acción**
1	Ini	GPI	Activa la operación del Generador de Población inicial
2	PIL	EV	Termina la generación de la población inicial e inicia el proceso de evaluación
3	FEV	SEL	Después del proceso de evaluación inicia la selección
4	FSEL	CRU	Una vez realizada la selección, procede a realizar la cruza.
5	FCRU	MUT	Terminado el proceso de cruza, inicia la mutación
6	FMUT	EV	Terminada la mutación de la población inicial, ejecuta nuevamente el proceso de evaluación, si encuentra la solución al problema (cero ataques) detiene el proceso y activa el despliegue de resultados, de lo contrario regresa al paso 3 y repite el ciclo para una nueva generación.

Módulo de Control. Este módulo realiza el control de la operación sincronizada de todos los módulos de la implementación. La primera función que realiza es generar una señal de *reset* a las memorias para que se configuren en sus respectivos módulos de acuerdo a los requerimientos de la implementación. Al recibir la señal *"ini"* inicia la operación del algoritmo, para lo cual éste va generando las señales de forma ordenada para que se vayan ejecutando las funciones de cada módulo, así mismo realiza el control de las generaciones a las que se van aplicando el algoritmo, finalmente se encarga de activar el proceso para desplegar los resultados mediante el módulo de control VGA y la barra de leds. En la Tabla 1 se muestra la operación de este módulo.

Módulo Generador de Población Inicial. En este módulo se genera de forma aleatoria la población inicial de soluciones, la cual está constituida por 128 individuos de 48 bits cada uno que se almacenan en la memoria de población cuya capacidad es de 128x48. Cada individuo está codificado como se indica en la Fig. 2. Se propone emplear una población de 128 individuos debido a que, para poder manipular los 100 individuos se requiere de un contador de 7 Bits, y para aprovechar el total de las cuentas se decidió que la población inicial fuese de 128 individuos.

Como se indica en la Fig. 2, cada reina posee un total de 6 bits para su codificación, 3 son para la columna (C) y 3 son para el renglón (R), que indican la posición de cada Reina en un tablero de 8 Columnas y 8 Renglones. La generación de la población inicial se realiza mediante un proceso aleatorio, para lo cual se emplea una base de tiempo diferente a la de operación de la implementación; es decir, provienen de fuentes físicamente diferentes. El dato de 48 bits se forma por 6 contadores de 8 bits en *free runing*, los cuales inician en valores diferentes.

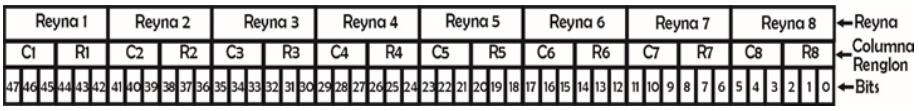


Fig. 2 Descripción de la codificación de una solución en la memoria de población, que contempla la codificación de las columnas y renglones de las 8 Reinas.

Estos contadores se detienen cada que el generador solicita un dato para almacenarlo en la memoria de población hasta completar 128 datos (soluciones o individuos).

La frecuencia de reloj empleada para este proceso aleatorio y todos los procesos aleatorios requeridos en la ejecución del algoritmo es de 50MHz que se obtiene de un oscilador que se encuentra en la misma tarjeta de desarrollo.

Módulo de Evaluación. En este módulo se encuentra implementada la función objetivo la cual determina el número de ataques que se presenta en cada solución (individuo) de la población. El cálculo del número de ataques se realiza por partes, primero se obtienen los ataques por comparación directa de las columnas y renglones de cada reina, enseguida se obtiene el número de ataques en diagonal con pendiente positiva sumando columna más renglón de cada reina, se compara directamente la suma de cada reina y si son iguales se contabiliza un ataque. Finalmente se obtienen los ataques en diagonal con pendiente negativa restando la columna del renglón de cada reina, se compara directamente la resta de cada reina y si son iguales se contabiliza un ataque. Los resultados de la evaluación de cada individuo son almacenados en la memoria de evaluación de forma tal que la dirección del individuo que se evalúa corresponde a la misma dirección en la memoria de evaluación, donde se almacena el resultado. De esta forma, cada solución en la memoria de población posee su evaluación en la misma dirección, pero en la memoria de evaluación. Cuando el número de ataques es cero, se detiene el proceso y este módulo lo indica a través de la señal *"reslist"* y proporciona la dirección de la memoria de población donde se encuentra la solución con ataques cero a través de las líneas *"Dirresul"*.

Módulo de Selección. En este módulo se realiza el proceso de selección mediante el método de torneo, el cual consiste en seleccionar de forma aleatoria parejas y determinar cuál de ellas posee la mejor evaluación (menor número de ataques). Para seleccionar la pareja de la primera solución de la memoria de población, se genera una dirección aleatoria entre 0 y 127. Con esta dirección se obtiene la pareja de la misma memoria de población, con esta misma dirección se obtiene el número de ataques de la pareja en la memoria de evaluación. La dirección de la pareja y el número de ataques se almacena en la memoria de selección en la primera localidad, el proceso lo repite con la segunda dirección de la memoria de población hasta recorrer las 128 direcciones. La codificación del dato que se almacena en la memoria de selección es de 12 bits, los primeros 5 bits corresponden al número de ataques de la pareja y los 7 bits restantes corresponden a su dirección. Para la selección, recorre la memoria de evaluación y la de selección, compara el número de ataques, si el número de ataques en la memoria de evaluación es mayor, se procede a reemplazar la solución de la memoria de población con su pareja, y el proceso se concluye al terminar de recorrer las 128 localidades de las memorias de evaluación y selección. Cabe aclarar que la nueva población resultante, fungirán como padres en el proceso de cruza.

Módulo de Cruza. Este módulo realiza la cruza de cada una de las soluciones (padres) resultado del proceso de selección que se ha almacenado en la memoria de población, el resultado de la cruza son los hijos, los cuales se almacenan nuevamente en la memoria de población obteniéndose ahora una población de hijos. En este proceso se generan 2 números aleatorios que corresponden a la probabilidad de cruza y al punto de cruza. La probabilidad se obtiene de un contador que opera libremente de 0 a 100, el punto de cruza se obtiene de otro contador que opera libremente de 0 a 47 ya que la longitud de cada solución es de 48 bits. La probabilidad de cruza seleccionada es del 80%, siendo el operador de cruza el operador principal y en la práctica, dicha probabilidad generalmente se elige en un rango del 60% al 100%. Con el punto de cruza se genera una palabra de control de cruza de 48 bits, que está formada por ceros del bit más significativo (47) hasta el punto de cruza y después del punto con unos hasta el bit menos significativo (0); de esta palabra también se obtiene su negado. El proceso de cruza se efectúa de la siguiente forma: se leen dos padres contiguos de la memoria de población, de cada padre se obtienen 2 registros; registro a, aplicando una función AND con la palabra de control y registro b, aplicando una función AND con el negado de la palabra de control. Posteriormente se obtiene el hijo uno realizando una función OR del registro 1b con el registro 2a, el segundo hijo se obtiene efectuando una función OR del registro 2b con el registro 1a. De este proceso se generan dos hijos que reemplazan a sus padres y se almacenan nuevamente en sus respectivas direcciones de la memoria de población. El proceso se repite para cada 2 padres hasta recorrer la totalidad de la memoria de población.

Módulo de Mutación. El proceso de mutación se aplica a cada hijo resultante del proceso de cruza, que se encuentran almacenados en la memoria de población. Se lee cada hijo y se recorre bit por bit desde el 0 hasta el 47. En cada bit que se va recorriendo, se obtiene la probabilidad aleatoria de cruza y si es menor o igual al 2% se procede a la mutación del bit en cuestión que consiste en negar dicho bit; si la probabilidad es mayor no se hace la negación y se pasa al siguiente bit. Este proceso se realiza en la totalidad de los hijos de la memoria de población. La probabilidad aleatoria de mutación se obtiene de un contador de 0 a 1000 que opera libremente, cabe aclarar que la base de tiempo de este contador es diferente físicamente a la base de tiempo con la que opera la implementación. La probabilidad de mutación se estableció de acuerdo a la longitud del individuo, definiéndola como *1/Lind*, donde *Lind* corresponde a 48 y es aproximadamente 0.02, lo que corresponde al 2% de mutación, mutando así en promedio un bit de cada individuo en promedio.

Módulo de lectura Memoria de Población. La función de este módulo inicia cuando el Módulo de Evaluación activa la señal "*reslist"* que significa que encontró una solución que genera cero ataques entre las reinas, así mismo le envía la dirección de dicha solución mediante bus "*Dirresul"*, con la dirección se lee la solución de la memoria de población, que es la solución al problema, ésta la envía al Módulo de control del Monitor VGA mediante el bus *"Sol"*. La señal *"reslist"* también la recibe el Módulo de Control con la finalidad de que proporcione el número de generación donde se encontró la solución. Dicho número de generación será desplegado en la barra de leds de 11bits.

Módulo de Control Monitor VGA. Este módulo se encarga de desplegar el resultado del algoritmo mediante un monitor VGA. Envía las señales de control al monitor, el bus RGB (3 Bits), las señales de Pixel "x" y Pixel "y" y las señales de sincronización.

Tabla 2. Resultados de las pruebas de desempeño de la implementación.

N° Pruebas	Generaciones	Pulsos	Tiempo [ms]
42	240.69	$6.1322x10^6$	1985.88
	[4, 1581]	[$6.6560x10^4$, $9.9008x10^7$]	[33.28, 13153.92]

Tabla 3. Resultados de las pruebas de desempeño de la implementación en software.

N° de Pruebas	Generaciones	Tiempo [seg]
42	206.17	507.14
	[69, 1650]	[136, 4548]

El monitor empleado es un VGA HP LV1911 con una resolución de 1366X768. El despliegue de los resultados se realiza de forma gráfica; es decir, el módulo genera un tablero de ajedrez de 8 columnas por 8 renglones, así mismo despliega la figura de las reinas en cada una de las posiciones según lo indique la solución recibida. El diseño de la figura de la reina se realizó pixel por pixel, de forma que el diseño de la figura es propio del presente trabajo.

5. Análisis de resultados

Las pruebas realizadas a la implementación se ilustran en la Tabla 2, se presenta el número de pruebas, el número de generaciones requeridas para llegar a la solución (cero ataques entre las reinas), el total de pulsos empleados, y el tiempo requerido. La cantidad de pruebas se consideraron 42 ya que se observó que los resultados en un número mayor a 42 era muy similar. El tiempo se calculó de acuerdo a la frecuencia del reloj empleado para la operación de la implementación, que fue de 2MHz. Esta señal de reloj se implementó mediante un oscilador integrado de la marca SARONIX NTC040C de 2Mhz, así mismo se calculó el tiempo que requiere la implementación para ejecutar el algoritmo en cada generación, contabilizando el número de pulsos que requiere cada módulo del algoritmo sumando un total de 16640 pulsos por generación.

De acuerdo con las pruebas realizadas, el tiempo promedio que requiere la implementación para llegar a la solución es de 1985.88 ms o 1.98588 seg en 42 pruebas realizadas. La mejor ejecución requirió de únicamente 4 generaciones y un tiempo de 33.28 ms; mientras que la ejecución que requirió más generaciones ocupó 1581 con un tiempo de 13.153 seg, representados en la Tabla 2 con los valores mínimos y máximos entre corchetes.

El desempeño de la implementación en FPGA es comparada con la implementación del mismo GA en software, para resolver el problema de ubicar 8 reinas en un tablero de ajedrez de 8x8 sin que se ataquen entre ellas. Dicha implementación se realizó usando MATLAB, considerando el toolbox de Algoritmos Genético del intérprete MATLAB, para mayor rapidez en la comparación. Sin embargo, el tiempo requerido de ejecución en software mejora si se implementa usando un lenguaje de alto nivel como “C”, dado que es un lenguaje de programación y no un intérprete. Las pruebas de la implementación en software se realizaron empleando una computadora DELL XPS con un sistema operativo WINDOWS 7 de 64 bits, con un procesador INTEL CORE i7-3770 y una frecuencia de reloj de 3.4 GHz. En la Tabla 3 se muestran los resultados

Tabla 4. Comparación de desempeño de las implementaciones en FPGA's y en MATLAB.

Implementación	N° de Pruebas	Promedio Generaciones	Promedio Tiempo [seg]
Hardware (FPGA's)	42	240	1.985
Software (MATLAB)	42	206	507.14

Tabla 5. Proyección del tiempo de la implementación en FPGA si se empleara una base de tiempo de 3.00 GHz.

N° de Pruebas	Generaciones	Pulsos	Tiempo [μs]
42	240.69 [4, 1581]	$6.1322x10^6$ [$6.6560x10^4$, $9.9008x10^7$]	1333.3 [22.18, 8770]

de la prueba, en dicha tabla se muestra el número de pruebas, el número de generaciones en la que encontró el resultado (cero ataques entre las reinas) y el tiempo requerido para llegar al resultado.

De las pruebas realizadas, en la Tabla 3 se muestra el tiempo promedio requerido por la implementación en MATLAB que es de 507.14 segundos. Al igual que en la Tabla 2, las generaciones y tiempos mínimos y máximos de las 42 ejecuciones se presentan en corchetes. En la Tabla 4 se muestra una comparación de los promedios obtenidos de las dos implementaciones.

De los resultados mostrados en las tablas anteriores se puede observar que la implementación en Hardware a través de FPGA's es 500 veces más rápida que la implementación en software empleando MATLAB. Si la implementación en software se realizará utilizando algún lenguaje de alto nivel como "C" o "JAVA", el tiempo requerido se reduciría ya que MATLAB es un software de aplicación y no un lenguaje de programación.

Es importante considerar que la base de tiempo de la implementación en FPGA es de 2MHz y la base de tiempo que emplea la computadora donde se probó la implementación en MATLAB es de 3.4 GHz. Ahora bien, si proyectamos los resultados de desempeño de la implementación en FPGA empleando de forma hipotética una base de tiempo de por ejemplo 3GHz, los resultados de la Tabla 2 serían los que se muestran en la Tabla 5.

De los datos mostrados en la Tabla 5, el tiempo que se emplearía para llegar a la solución es mucho menor, en promedio 1.333 ms, los valores mostrados entre corchetes representan los valores mínimos y máximos para llegar a la solución. Para poder realizarlo se tendría que investigar si las plataformas de FPGA existentes soportan la frecuencia propuesta de 3GHz.

6. Conclusiones y trabajo a futuro

En el presente trabajo se desarrolló la implementación de un GA Simple en hardware mediante FPGA's para resolver un problema ejemplo de tipo combinatorio que consiste en ubicar en un tablero de ajedrez de 8 x 8 casillas 8 reinas sin que se ataquen entre ellas. El algoritmo resuelve el problema ubicando las 8 reinas sin ataques, la disposición

de las reinas no siempre es la misma, siendo este problema combinatorio un problema multimodal dado que se tiene más de una configuración del tablero con cero ataques (solución óptima).

Así mismo se realizó una comparación del desempeño de la implementación en FPGA's, con la implementación del mismo algoritmo en software empleando MATLAB resolviendo el mismo problema, comprobando que la implementación en hardware es más rápida. En este caso particular resultó ser 500 veces más rápida aproximadamente tal como se indica en la Tabla 4. También se presenta una proyección de tiempo si se empleara hipotéticamente una base de tiempo para la implementación con FPGA´s de 3GHz y evidentemente el tiempo es mucho menor, pudiendo existir la posibilidad de poder aplicar el algoritmo en problemas donde el tiempo es un factor muy importante.

Como trabajo futuro se plantea implementar el AG en un lenguaje de alto nivel para comprobar que el tiempo mejora con respecto a la implementación con MATLAB, así mismo agregar al algoritmo un módulo de elitismo, para mejorar aún más el desempeño, usar una frecuencia de reloj externa mayor a 2 MHz. Se plantea la posibilidad de utilizar FPGA's de última generación que permitan un ancho de banda mayor para poder utilizar frecuencias de reloj más altas para tener un desempeño mucho mejor.

Referencias

1. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison Wesley (1989)
2. Sivanandam, S.N., Deepa, S.N.: Introduction to genetic algorithms. Springer, pp. 15–30 (2008)
3. Kuri, A., Galaviz, J.: Algoritmos genéticos. Fondo de cultura Económica, 3(3), pp. 13–30 (2000)
4. Smith, G.R.: FPGA's 101 Everything you need to know to get stated. Elsevier Inc., pp 43–55 (2010)
5. Scott, S.D., Samal, A., Seth, S.: HGA: A hardware-based genetic algorithm. In: Proceedings ACM International Symposium on Field Programable Gate Arrays, pp. 53–59 (1995)
6. Tufte, G., Haddow, P.: Evolving and adaptive digital filter. In: NASA/DoD Workshop on Evolvable Hardware. pp. 143–150 (2000)
7. Aporntewan, C., Chongstitvatana, P.: A hardware implementation of the compact genetic algorithm. In: Proceedings IEEE Congress on Evolutionary Computation Seoul Korea, pp. 624–629 (2001)
8. Shackleford, B., Snider, G., Carter, R.J., Okushi, E., Yasuda, M., Seo, K., Yasuura, H.: A high-performance pipelined, FPGA-based genetic algorithm machine. Genetic Programming and Evolvable Machines, Springer, 2, pp. 33–60 (2001)
9. Tang, W., Yip, L.: Hardware implementation of genetic algorithms using FPGA. In: IEEE International Midwest Symposium on Circuits and Systems, pp. 549–552 (2004)
10. Mostafa, H.E., Khadragi, A.I., Hanafi, Y.Y.: Hardware implementation of genetic algorithm on FPGA. In: National Radio Science Conference (2004)
11. Thirer, N.: About the FPGA implementation of a genetic algorithm for solving sudoku puzzles. In: IEEE 27th Convention of Electrical and Electronics Engineers in Israel (IEEEI) (2012)

12. Tuncer, A., Yildirim, M.: Design and implementation of a genetic algorithm IP core on an FPGA for path planning of mobile robots. Turkish Journal of Electrical Engineering and Computer Sciences, 24(6), pp. 5055–5067 (2016)
13. Alansi, M., Elshafiey, I., Al-Sanie, A., Mabrouk, A.: FPGA implementation of multi-user detection genetic algorithm tool for sdma-ofdm systems. Springer, 86, pp. 1241–1263 (2016)
14. Guo, L., Thomas, D.B., Cross, A.I., Fu, H.: Parallel genetic algorithms on multiple FPGA's. In: ACM SIGARCH Computer Architecture News, 43(4), pp. 86–93 (2017)
15. Thorbole, P.S., Kalbhor, S.D., Harpale, V., Bairagi, V.K.: Hardware implementation of genetic algorithm for epileptic seizure detection and prediction. In: International Conference on Computing, Communication, Control and Automation (ICCUBEA) (2017)
16. Praveena, M., N.B., Naidu, C.D.: FPGA implementation of high-speed medical image segmentation using genetic algorithm. Journal of Theoretical and Applied Information Technology, 95(13), pp. 2981–2988 (2017)
17. Peker, M.: A fully customizable hardware implementation for general purpose genetic algorithms. Applied Soft Computing, 62, pp. 1066–1076 (2018)
18. Torquato, M., Costa-Fernandes, M.A.: High-performance parallel implementation of genetic algorithm on FPGA. Circuits, systems, and signal processing, 38(7), pp. 4014–4039 (2019)
19. Gupta, V., Jain, A., Chourasia, B.: FPGA Based Implementation of Genetic Algorithm Using VHDL. Implementation of generic algorithm using VHDL on FPGA. International Journal for Recent Developments in Science and Technology (2011)
20. Xilinx: http://xilinx.com/products/boards-and-kits/HW-SPAR3E-SK-US-G.htm (2019)
21. Ashenden, P.: The Designer's Guide to VHDL. 3 (1995)

ISSN 1870-4069

Análisis con aprendizaje automático en el proceso de flotación de burbujas para la obtención de minerales

Nelsón Romero-García[1], Sonia Regina Meneses[1], Julio Waissman Vilanova[2], Oscar Mario Rodríguez Elias[1], César Enrique Rose Gómez[1], Álvaro Rendón Montoya[3]

[1] Tecnológico Nacional de México,
Instituto Tecnológico de Hermosillo,
México
[2] Universidad de Sonora,
México
[3] ECN Automation,
México

nelsonromero219@gmail.com, julio.waissman@unison.mx,
{sonia.menesesm, omrodríguez}@hermosillo.tecnm.mx,
arendon@ecn.com.mx

Resumen. El análisis de capa superficial de espuma ha sido clave para entender el comportamiento general de los sistemas de flotación de burbujas para la obtención de minerales en la industria minera. En el estado del arte se han desarrollado varios analizadores de capa superficial de espuma de las burbujas en las celdas de flotación con el objetivo de clasificar mejor el proceso, adición de reactivos y control del flujo de aire. El uso de la inteligencia artificial, siendo más específico el aprendizaje automático, en la industria ha tenido un uso creciente y ha demostrado su aplicabilidad y buenos resultados mediante aplicación de técnicas para la clasificación de imágenes. En este artículo se presenta un caso aplicado a la industria minera donde se hace uso de métodos y herramientas de aprendizaje automático para la clasificación de imágenes, con el fin de obtener el modelo que brinde mejores resultados para estimar el estado del proceso a partir de las imágenes tomadas en las celdas de flotación, con lo que se pretende mejorar el proceso, reducir costos, aumentar productividad y calidad en la extracción de minerales.

Palabras clave: Flotación, burbujas, industria minera, celdas, inteligencia artificial, aprendizaje automático, imágenes.

Analysis with Machine Learning in the Bubble Flotation Process to Obtain Minerals

Abstract. The analysis of the surface foam layer has been key to understanding the general behavior of bubble flotation systems for obtaining minerals in the mining industry. In the state of the art, several surface foam layer analyzers of the bubbles in the flotation cells have been developed in order to better classify the process, the addition of reagents and control of airflow. The use of artificial intelligence, machine learning is

more specific, in the industry has had an increasing use and has demonstrated its applicability and good results by applying techniques for image classification. This article presents a case applied to the mining industry where machine learning methods and tools are used to classify images, in order to obtain the model that provides the best results to estimate the state of the flotation process from of the images taken in the flotation cells, with the intention of improving the process, reducing costs, increasing productivity and quality in the extraction of minerals.

Keywords: Flotation, bubbles, mining industry, cells, artificial intelligence, machine learning, images.

1. Introducción

De acuerdo con [16] el sector minero-metalúrgico en México contribuye con el 4% del Producto Interno Bruto (PIB) nacional. México se ubica entre los 10 principales productores de 16 diferentes minerales en el mundo, es el 1er destino en inversión en exploración minera en América Latina y el 4to en el mundo; y es el 5to país con el mejor ambiente para hacer negocios mineros.

Sonora ha sido considerado tradicionalmente como un Estado minero de gran importancia [17], siendo así que el desarrollo e implementación de nuevas tecnologías resulta de gran importancia para la mejora de los procesos de obtención de minerales. La modernización de la industria minera a nivel mundial se encuentra en pleno desarrollo y la industria de minera mexicana debe mantenerse cerca de este desarrollo.

La investigaciones y desarrollos en el campo de la inteligencia artificial haciendo uso de la visión artificial en sistemas de flotación de burbujas, cubre la necesidad de obtener resultados más eficientes para el análisis de las mismas. Este proceso de análisis es un problema mal definido puesto que en cada mina se realiza mediante la observación de un operador de planta que valora el comportamiento de las burbujas y determina la calidad del proceso, por lo que la caracterización del proceso de flotación de burbujas no es exacta y por tanto las pérdidas en la obtención del mineral es un problema que afecta a la industria.

La visión artificial puede extraer con precisión y rapidez las características de las burbujas, tanto físicas como dinámicas, a través de imágenes digitales y presentar estos resultados a los operadores y / o utilizar los resultados como entradas para el control del proceso.

La comprensión de estos procesos de flotación ha sido durante mucho tiempo clave para entender el comportamiento general de los sistemas de flotación [10], [11]. Buscando resolver esta problemática, se han desarrollado analizadores de la burbuja derramada en las celdas de flotación, buscando mejorar el control de nivel, la adición de reactivos y el control de flujo de aire.

Los objetivos principales de control de las celdas de flotación son los factores metalúrgicos, es decir, recuperación y el grado de concentración del mineral. La estimación y medición de estas variables generalmente requieren instrumentos sofisticados que son caros de comprar y mantener [8]. Estudios anteriores demuestran que las características visuales de la espuma reflejan cambios en las condiciones del proceso y pueden usarse para predecir los factores metalúrgicos y el grado de recuperación del mineral [5, 6].

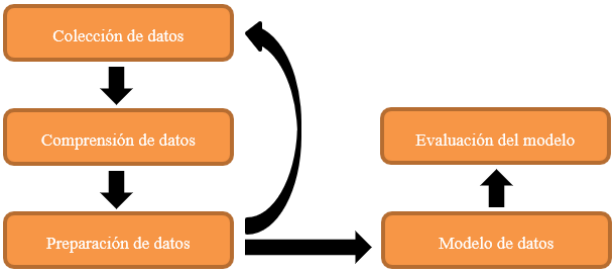


Fig. 1. Metodología de trabajo.

Las características de la capa superficial de espuma formada por las burbujas más importantes incluyen el tamaño de burbuja, el color de la espuma, la velocidad de la espuma y la velocidad de colapso de la burbuja. Varios algoritmos de procesamiento de imágenes eficientes fueron desarrollados para cuantificar estas características. La distribución del tamaño y el color son variables estáticas que se calculan a partir de una sola imagen, mientras que la velocidad de la espuma y la velocidad de colapso de la burbuja son variables dinámicas que se calculan a partir de un par de imágenes.

Para la medición del tamaño de burbuja se han desarrollado varias técnicas las cuales incluyen, la segmentación [7, 8], análisis de textura [7], enfoque de espectro de textura modificado [6] y uso de información morfológica interfacial [20]. En la práctica, cada uno de estos métodos tiene sus respectivas ventajas y desventajas.

El presente trabajo se plantea dada la necesidad de reducir pérdidas minerales y económicas, facilitar la caracterización de burbujas en el proceso de flotación y proveer un análisis de los métodos y herramientas de aprendizaje automático (ML) que se pueden utilizar para resolver un problema como el planteado, mediante imágenes tomadas en las celdas de flotación. Se muestra cómo se pueden tratar las imágenes para resolver problemas de clasificación de burbujas en celdas de flotación e implementar modelos de aprendizaje automático, donde las máquinas de soporte vectorial pueden producir mejores resultados.

Además, al trabajar en la optimización de estos modelos para identificar los diferentes estados de la capa superficial de espuma generada en las celdas de flotación, podría mejorarse aún más por un margen considerable. El resultado final de esto es que es posible realizar estimaciones marcadamente más confiables para la clasificación de burbujas.

2. Metodología

Con demasiada frecuencia existe la tendencia de pasar por alto la metodología y saltar directamente a resolver un problema con ML. Sin embargo, hacerlo de esta manera no se establece una base sólida y no es posible asegurar criterios de calidad para que el proceso de ML pueda ser integrado en un proceso productivo. Por tanto, seguir una metodología más a menudo nos acerca al problema que estábamos tratando de resolver.

A continuación, se muestra la metodología seguida para el desarrollo del presente trabajo (véase Fig. 1).

3. Exploración de datos

Las técnicas de ML han tenido en una amplia gama de aplicaciones, como visión por computadora, procesamiento del habla, comprensión del lenguaje natural, neurociencia, salud e internet de las cosas; dado que en la era de los grandes datos el ML ha despertado amplios intereses. Por una parte, los grandes datos proporcionan información sin precedentes para algoritmos de ML para extraer patrones subyacentes y construir modelos predictivos; por otra parte, los algoritmos tradicionales de ML enfrentan desafíos como la escalabilidad para liberar realmente el significado de los grandes datos.

3.1. Colección de datos

Los datos de imagen de las burbujas fueron tomadas en una celda de flotación de una planta minera de Sonora mediante el uso de cámaras Giga Ethernet (GigE) con un lente de 16mm. Durante el período de toma de las imágenes, se hizo variación sobre la adición de reactivos y el control de flujo de aire para obtener diferentes tipos de burbujas.

Las imágenes de 499 x 499 píxeles se asociaron con tres diferentes clasificaciones de burbujas según la clasificación de los operadores, burbuja pequeña (clase 1), burbuja mediana (clase 2) y burbuja grande (clase 3) (véase Fig. 2). Estas clases fueron esencialmente distinguidas por el tamaño de la burbuja en el banco de flotación. Un total de 553 imágenes en los datos fue lo obtenido al final del proceso.

Fig. 2. Imágenes muestras de cada una de las clases de capa superficial de espuma.

Como se indica en el parte inferior de la Fig. 2, cada clase de burbuja está asociada con un valor, es decir, Clase 1 burbuja pequeña compuesta por 200 imágenes, Clase 2 burbuja mediana compuesta por 141 imágenes y Clase 3 burbuja grande compuesta por 212 imágenes. El conjunto de datos de entrenamiento consistió en 75% de las imágenes seleccionadas al azar, mientras que el resto sirvió como datos de prueba para validar el modelo.

3.2. Coincidencia de histogramas

La coincidencia de histogramas [1] manipula los píxeles de una imagen fuente con una imagen de referencia, haciendo uso del histograma de colores de ambas imágenes. El número de canales de colores entre la imagen fuente y la imagen de referencia deben

Fig. 3. Imagen resultada de aplicar la normalización de colores sobre la imagen fuente.

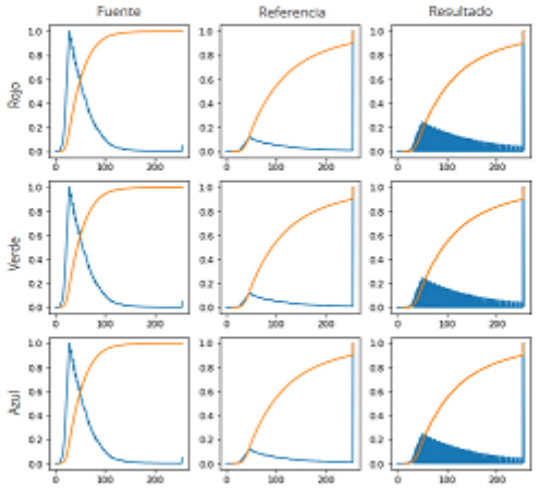


Fig. 4. Histogramas de colores trazado por cada canal RGB.

ser igual en todo momento, por tanto, la coincidencia se realiza de forma independiente por cada canal. A continuación, se muestra el resultado de esta normalización ligera de colores (véase Fig. 3) y el histograma de colores trazado por cada canal RGB, donde se observa como la imagen coincidente tiene el mismo histograma cumulativo que la imagen de referencia (véase Fig. 4).

3.3. Reagrupamiento de atributos usando PCA

El análisis de componentes principales (PCA, por sus siglas en inglés) es una de las principales técnicas para la reducción de dimensionalidad en un conjunto de datos mientras que conserva la mayor parte de la información [18]. Se utiliza para conocer cómo están correlacionados los datos y proporciona un número mínimo de variables que mantengan la máxima cantidad de variación o información sobre cómo se distribuyen los datos originales.

Aplicando PCA a nuestro conjunto de datos y graficando las dos primeras variables que más explican la distribución de los datos, se obtuvo el siguiente resultado (véase Fig. 5).

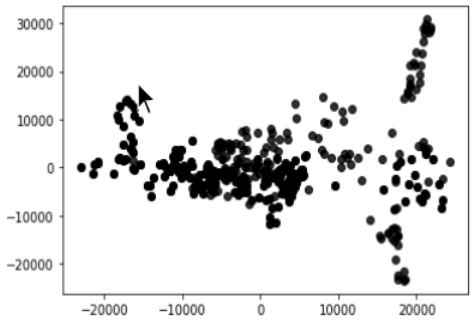


Fig. 5. PCA sobre las dos primeras variables.

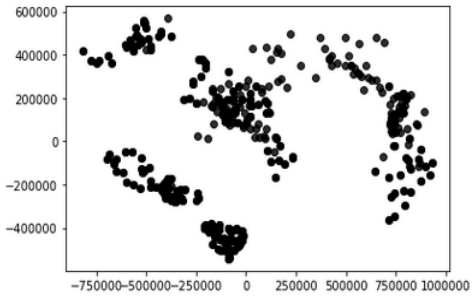


Fig. 6. PCA sobre las dos primeras variables usando la transformada de Fourier.

Con este resultado no se obtuvo una separación clara de los datos, por lo que se aplicó la transformada Fourier para analizar de otra forma los datos con el PCA. La transformada de Fourier [2] analiza los componentes de frecuencia en una señal bidimensional, como el sonido o la onda de radio.

Toma datos bidimensionales como entrada y devuelve otros datos bidimensionales como números complejos, lo que permite encontrar variaciones en las frecuencias. En el procesamiento de imágenes se utiliza mucho pues permite encontrar variaciones de colores. Al aplicar esta técnica primeramente a nuestros datos y posteriormente PCA, obtuvimos la siguiente distribución (véase Fig. 6).

3.4. Reagrupamiento de ejemplos usando el método de mezcla de gaussianas

El reagrupamiento de ejemplos tiene como objetivo encontrar diferentes grupos dentro de los elementos en los datos. Para esto, los algoritmos encuentran estructuras en los datos para que los elementos del mismo grupo sean más similares entre sí que con el resto de los grupos.

El método de mezcla de Gaussianas [14] es un modelo probabilístico que supone que todos los puntos de datos se generan a partir de una mezcla de un número finito de distribuciones gaussianas con parámetros desconocidos. Pertenece al grupo de algoritmos de agrupación suave en los que cada punto de datos pertenecerá a cada grupo existente en el conjunto de datos, pero con diferentes niveles de pertenencia a cada grupo,

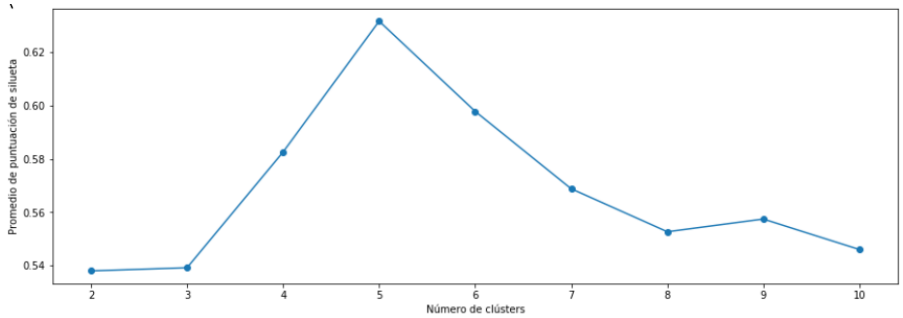


Fig. 7. Método de silueta sobre mezcla de Gaussianas**.**

Fig. 8. Método de mezcla de Gaussianas para 5 grupos.

donde se asigna la probabilidad, que va de 0 a 1, de cada dato de pertenecer a un determinado grupo.

Para conocer el número óptimo de grupos a formar tras aplicar la técnica de mezcla de gaussianas, se utilizó el método de validación de silueta [15]. Este método trabaja sobre la distancia media entre una muestra y todos los demás puntos en el mismo grupo, además de la distancia media entre una muestra y todos los demás puntos en el siguiente grupo más cercano. El método de silueta permite conocer cuán compactos y separados se encuentran los grupos, por lo que mientras más se acerque al puntaje uno, mayor es la agrupación (véase Fig. 7).

La gráfica nos permite observar cómo para 5 grupos podemos establecer que es el número óptimos a formar durante la agrupación usando la mezcla de Gaussianas. A continuación, se muestra el resultado obtenido (véase Fig. 8).

3.5. Matriz de relación entre grupos formados y clasificación de expertos

Una vez realizado el reagrupamiento de los datos y con los resultados obtenidos por parte de la clasificación previa de los expertos, se realizó una matriz de relación entre ambos para analizar los resultados, véase Tabla 1.

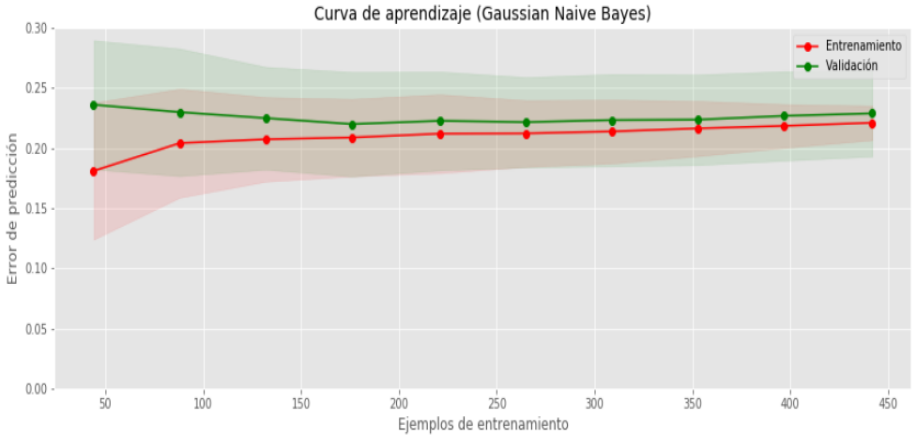


Fig. 10. Curva de aprendizaje para el algoritmo Gaussian Naive Bayes.

Tabla 1. Matriz de relación entre GruposFormados/ClasificaciónExpertos.

GruposFormados/ClasificaciónExpertos	1	2	3
1	100	0	0
2	0	48	80
3	0	0	60
4	0	93	72
5	100	0	0

Como se puede observar en la imagen, la clase 1 perteneciente a las burbujas pequeñas se pueden identificar perfectamente con los grupos 1 y 5 de los grupos formados por la agrupación. Para identificar las demás clases no es suficiente con el análisis realizado hasta el momento por lo que se puede observar, por tanto, se aplicaron métodos de aprendizaje automático para resolver este problema.

4. Métodos de aprendizaje

Se muestra a continuación el resultado del uso de los métodos de aprendizaje, donde la aplicación de la metodología a los modelos datos para clasificación de imágenes se ve reflejada.

4.1. Gaussian naive Bayes

El algoritmo Naive Bayes es uno de los algoritmos más antiguos de aprendizaje automático, que se basa en el teorema de Bayes. La teoría de Bayes y los fundamentos de las estadísticas en la que se basa este algoritmo se desarrollaron en el siglo XVIII. Antes de que apareciera la primera implementación de este algoritmo por computadora, todos los cálculos se hacían manualmente.

El Gaussian Naive Bayes es un modelo clasificador basado en Naive Bayes pero haciendo una suposición de distribución normal de las verosimilitudes [9].

Se muestra a continuación los resultados arrojados por el modelo Gaussian Naive Bayes, se eligió porque es el más simple y el más popular (véase Fig. 10).

4.2. Regresión logística

Uno de los algoritmos más utilizados actualmente en el aprendizaje automático es el de regresión logística [4]. Por su eficacia y simplicidad es muy utilizada por científicos de datos, pues no es necesario disponer tanto en entrenamiento como en ejecución de grandes recursos computacionales.

Dado el propósito del presente trabajo sobre la clasificación de imágenes de burbujas, se procedió hacer uso de este algoritmo de regresión logística para analizar su rendimiento en este tipo de problema. En la siguiente imagen se muestra su desempeño (véase Fig. 11).

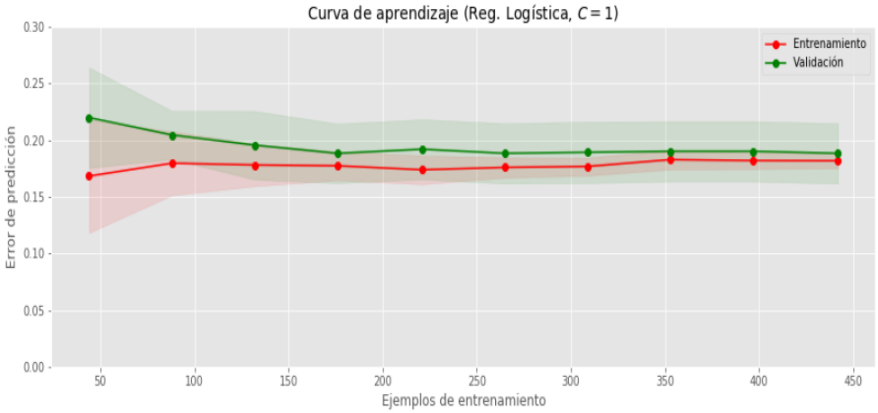


Fig. 11. Curva de aprendizaje para el algoritmo de Regresión Logística.

4.3. Máquina de vectores de soporte con kernel gaussiano

Las máquinas de vectores de soporte (Support Vector Machine, SVM por sus siglas en inglés) constituyen algoritmos de aprendizaje automático muy utilizados. SVM fue desarrollado en los años 90 [3] por Vladimir Vapnik y su equipo de trabajo. Su objetivo se centra en la separación de clases mediante un hiperplano en un espacio extendido donde maximiza el margen de clasificación, por lo que en el espacio extendido la separación entre clases es lineal mientras que no es lineal en el espacio original. Su aplicación actual se refleja en el reconocimiento de imágenes y textos, aplicaciones médicas, y muchos más; y son utilizadas tanto en problemas de clasificación como de regresión.

Para trabajar con SVM hay que tener en cuenta el kernel a utilizar, sus hiperparámetros y el parámetro de regularización. Los kernel más utilizados son el lineal, polinomial y gaussiano, donde cada uno sigue una aproximación distinta en la

que se realiza una expansión de la dimensionalidad. El ajuste de una SVM con kernel gaussiano se ve determinado esencialmente por el parámetro γ (Gamma) y el parámetro de regularización C. El parámetro γ es muy importante para desarrollar una SVM con buena generalización ya que el mismo determina el ancho del núcleo.

Si γ es demasiado grande la SVM va tender al sobreajuste y si es pequeño habrá mayor solape entre gaussianas, y el modelo será muy pobre. El parámetro de regularización C se refiere a la penalización y establece una relación entre el error de entrenamiento y la complejidad del modelo; permite la flexibilidad de que se puedan cometer errores a la hora de clasificar ejemplos con la SVM. Si mayor es el valor de C, la penalización de errores es más rígida por lo que tiende al sobreajuste. Si es C es menor, permite mayor error en el entrenamiento por lo que tiende al subajuste.

En esta investigación se trabajó con el kernel gaussiano por sus buenas prestaciones, brinda una expansión de la dimensionalidad más alta y es el método núcleo estándar para clasificación. El primer resultado de la SVM con valores estándares mostró el siguiente comportamiento (véase Fig. 12).

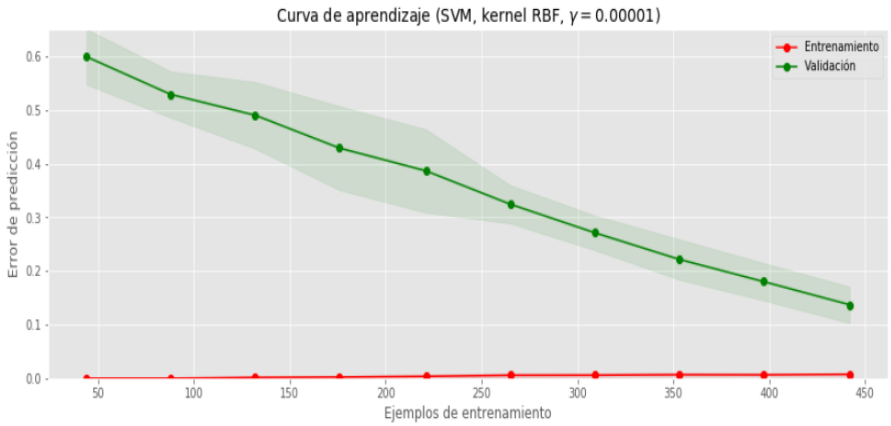


Fig. 12. Curva de aprendizaje para el algoritmo de Máquina de Soporte Vectorial con núcleo Gaussiano.

Se observa en la figura 12 como el error de validación no converge como el error de entrenamiento. Esto se debe por las siguientes cuestiones fundamentales: primero tenemos pocos datos para entrenar y validar nuestro modelo, problema que se detectó desde un inicio en la colección de datos, y segundo no están optimizados nuestros parámetros. A continuación, se trata de optimizar los parámetros γ y C para resolver el problema anterior planteado.

Curvas de validación para los parámetros γ y C

Como se observa en la figura 12, la SVM con núcleo gaussiano muestra mejores resultados que las anteriores técnicas aplicadas. Posteriormente se trabajó en la optimización de los parámetros γ y C para mejorar el comportamiento del modelo.

Los valores óptimos de estos parámetros se pueden determinar por validación cruzada. Los rangos de valores que se exploran se muestran a continuación (véase Fig. 13 y Fig. 14).

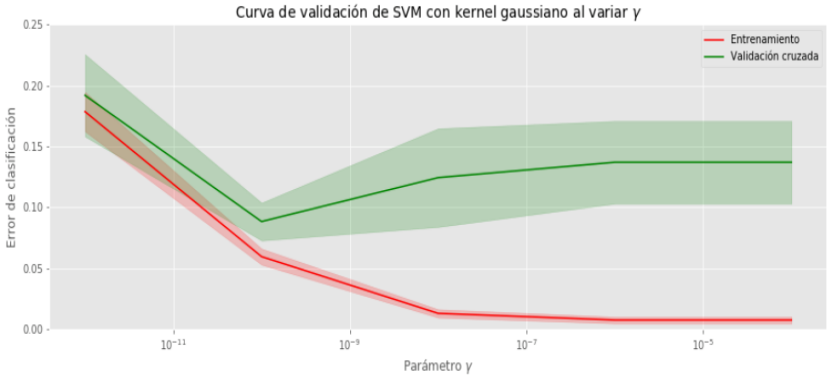


Fig. 13. Curva de validación para el parámetro γ.

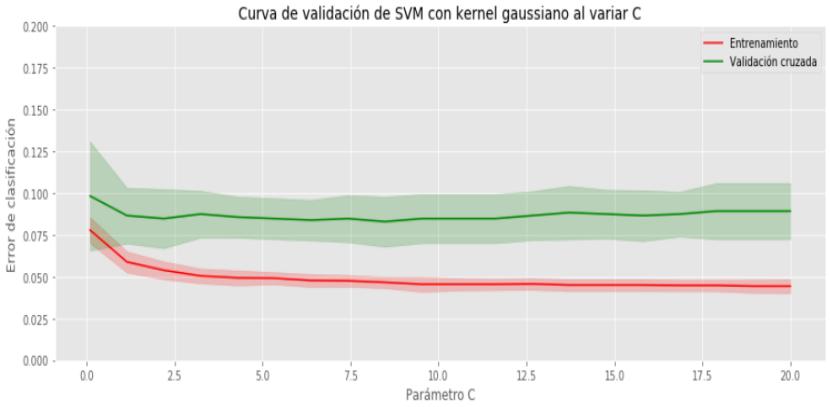


Fig. 14. Curva de validación para el parámetro C.

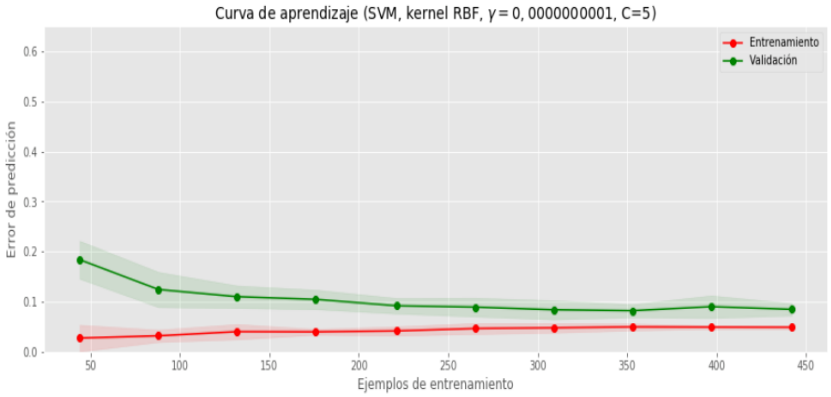


Fig. 15. Curva de aprendizaje para el algoritmo de Máquina de Soporte Vectorial con núcleo Gaussiano y parámetros óptimos.

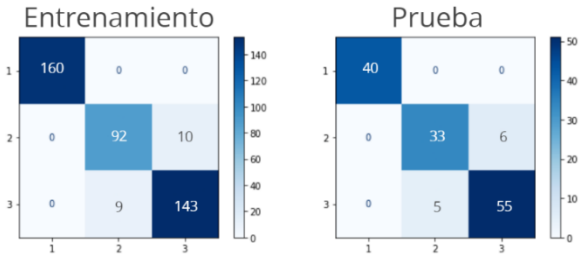


Fig. 16. Matriz de confusión al aplicar SVM con núcleo gaussiano

Tabla 2. Precisión de clasificación de modelos en conjuntos de entrenamiento y validación.

Modelo	Parámetros	Entrenamiento	Validación
Gaussian Naive Bayes	`priors=[0.1, 0.1, 0.1], var_smoothing=1e-9`	78%	77%
Regresión Logística	`penalty=l2, C=1, multi_class=auto`	83%	82%
SVM - Gaussiano	`C=1, kernel=rbf, gamma=scale`	99%	88%
SVM - Gaussiano Optimizado	`C=5, kernel=rbf, gamma=1e-5`	95%	92%

4.4. Máquina de soporte vectorial con núcleo Gaussiano y parámetros óptimos

El rango de valores óptimos para γ se encuentran entre 10^{-11} y 10^{-9}, por lo que se toma el valor 10^{-10} como el óptimo.

Para C los valores se encuentran en el rango de 3 hasta 10 y en este caso se tomó el valor 5 como el óptimo. En la figura siguiente se muestran los resultados de nuestro modelo optimizado (véase Fig. 15).

5. Resultados

Los resultados de los modelos en los conjuntos de datos de imágenes de entrenamiento y validación se resumen en la Tabla 2.

Como se puede observar en la Tabla 2, las precisiones de los modelos Gaussian Naive Bayes y Regresión Logística no son buenas tanto para entrenamiento como para validación. El modelo de SVM con núcleo gaussiano logró resultados mucho mejores de aproximadamente 99% para entrenamiento y 88% para el conjunto de validación lo que indica que tenemos un problema de sobreaprendizaje. Se obtuvieron mejoras significativas cuando se optimizaron los parámetros para la SVM con núcleo gaussiano, lo que resultó en precisiones de clasificación de aproximadamente 95% y 92 % para los conjuntos de entrenamiento y validación respectivamente. Para obtener estos resultados se hace uso de validación cruzada con el 20 por ciento de datos para validación.

A continuación, se muestran los resultados obtenidos por clases de la SVM con núcleo gaussiano en la matriz de confusión (véase Fig. 16).

6. Conclusiones

En esta investigación, se explora el uso de técnicas de aprendizaje automático tanto para el procesamiento como análisis de datos de imágenes. Se trabajó con los modelos de aprendizaje, Gaussian Naive Bayes, Regresión Logística y Máquinas de Soporte Vectorial, para el desarrollo de la clasificación de imágenes en las celdas de flotación.

Con los resultados de este estudio podemos concluir como un análisis detallado y una exploración sobre los datos resulta de gran relevancia para comprensión de este tipo de problemáticas. La implementación de modelos para extraer características de las imágenes de burbujas para clasificar las mismas, resulta en una mejora significativa sobre la toma de decisiones en proceso industrial minero.

De acuerdo con estos resultados preliminares se puede afirmar que el uso de ML puede ser aplicado de manera efectiva para el control y supervisión de procesos de flotación. En trabajos siguientes se pretende hacer uso de aprendizaje profundo para analizar los resultados.

Actualmente el grupo de trabajo se encuentra en el proceso de recolección de imágenes de diferentes celdas de flotación con fines de validación de estos resultados. El etiquetado de datos de entrenamiento, validación y prueba en diferentes plantas impone un desafío.

Referencias

1. Bourke, P.: Histogram Matching. http://paulbourke.net/miscellaneous/equalisation/ (2006)
2. Cooley, J.W., Tukey, J.W.: An Algorithm for the machine calculation of complex fourier series. Math. Comput. 19, pp. 297–301 (1965)
3. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20, 3, pp. 273–297 (1995)
4. Hosmer, D.W., Lemeshow, S., Rodney, X.: Sturdivant Applied logistic regression. Wiley (2000)
5. Kaartinen, J., Hätönen, J., Hyötyniemi, H., Miettunen, J.: Machine-vision-based control of zinc flotation-A case study. Control Eng. Pract. 14(12), pp. 1455–1466 (2006)
6. Lina, B., Reckeb, B., Knudsenb, J.K.H., Bay, S.J.,: Bubble size estimation for flotation processes. Miner. Eng. 21(7), pp. 539–548 (2008)
7. Liua, J.J., MacGregor, J.F., Duchesne, C., Bartolaccic, G.: Flotation froth monitoring using multiresolutional multivariate image analysis. Miner. Eng. 18(1), pp. 65–76 (2005)
8. Liu, J.J., MacGregor, J.F.: Froth-based modeling and control of flotation processes. Miner. Eng. 21(9), pp. 642–651 (2008)
9. Lowd, D., Domingos, P.: Naive bayes models for probability estimation. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 529–536 Association for Computing Machinery (2005)
10. Mathe, Z.T., Harris, M.C., O'Connor, C.T., Franzidis, J.P.: Review of froth modelling in steady state flotation systems. Miner. Eng. 11(5), pp. 397–421 (1998)
11. McKee, D.J.: Automatic flotation control – a review of 20 years of effort. Miner. Eng., 4(7–11), pp. 653–666 (1991)
12. Mehrshad, N., Massinaei, M.: New image-processing algorithm for measurement of bubble size distribution from flotation froth images. Miner. Met. Process, 28(3), pp. 146–150 (2011)

13. Morar, S.H. et al.: The use of machine vision to predict flotation performance. Miner. Eng., pp. 36–38 (2012)
14. Reynolds, D.A.: Gaussian Mixture Models. In: Encyclopedia of Biometrics (2009)
15. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math., 20, pp. 53–65 (1987)
16. Secretaría de Economía: Minería. https://www.gob.mx/se/acciones-y-programas/mineria (2020)
17. SGM: Panorama minero del estado de Sonora. Panor. Min. los Estados, pp. 1–64 (2018)
18. Tipping, M.E., Bishop, C.M.: Probabilistic Principal Component Analysis. J. R. Stat. Soc. Ser. B, Statistical Methodol, 61(3), pp. 611–622 (1999)
19. Wang, W., Bergholm, F., Yang, B.: Froth delineation based on image classification. Miner. Eng., 16(11), pp. 1183–1192 (2003)
20. Yang, C., Xux, C., Mu, X., Zhou, K.: Bubble size estimation using interfacial morphological information for mineral flotation process monitoring. Trans. Nonferrous Met. Soc. China, 19(3), pp. 694–699 (2009)

ISSN 1870-4069

Aplicación del algoritmo K-means para el análisis de resultados de la prueba PLANEA 2017

Israel Gutiérrez González, Doricela Gutiérrez Cruz,
Jenny Elizabeth Juan Ramírez, Liliana Rodríguez Páez,
Ricardo Rico Molina, Mauricio Sánchez Medina

Universidad Autónoma del Estado de México,
Centro Universitario UAEM Nezahualcóyotl,
México

igutierrezg@uaemex.mx

Resumen. El uso de técnicas de minería de datos en el estudio de algunos de los factores involucrados en el rendimiento académico es actualmente un campo de trabajo que se encuentra permeando en los temas de investigación educativa a nivel internacional. Entre estas técnicas se encuentran el algoritmo K-Means el cual es un algoritmo de aprendizaje no supervisado y que se utiliza en este trabajo para analizar los resultados de la prueba PLANEA 2017 de escuelas de nivel medio superior en cada una de los Estados de la República Mexicana en el Área de Lenguaje y Comunicación. En la presente investigación, se analizaron los factores entidad, turno, financiamiento y nivel de marginación de las escuelas evaluadas y se encuentra que hay un importante impacto de los factores estudiados sobre el aprovechamiento por parte de los estudiantes.

Palabras clave: Rendimiento académico, K-Means, PLANEA 2017.

Application of the K-Means Algorithm for the Analysis of PLANEA 2017 Test Results

Abstract. The use of data mining techniques in the study of some of the involved factors in academic achievement is currently a work area that is permeating international educational research topics. Among these techniques is the K-Means algorithm, which is an unsupervised learning algorithm and is used in this work to analyze the results of the PLANEA 2017 test of the high school in each of the States of the Mexican Republic in the Language and Communication Area. In the present investigation, the factors entity, shift, financing and marginalization level of the evaluated schools, were analyzed and it was found that there is an important impact of the studied factors on the students' achievement.

Keywords: Academic achievement, K-Means, PLANEA 2017.

1. Introducción

La evaluación del rendimiento académico es uno de los principales indicadores que toman importancia fundamental dentro de los sistemas educativos a nivel mundial,

donde no solo se mide la efectividad de un sistema educativo, sino que apoyan a los docentes a mejorar la enseñanza y los procesos de evaluación encaminados a alcanzar una educación de calidad [1] e involucrando al estudiante en términos de desarrollo de capacidades o de adquisición de aptitudes generales, así como de competencias curriculares, entre otros [2]. Debido a esto, en el plano internacional, existen diversos organismos especializados en la evaluación y cuyos resultados tienen un impacto importante en las políticas estatales en cuanto a educación se refiere.

En este sentido, la Organización para la Cooperación y el Desarrollo Económico (OCDE) la cual es un organismo internacional que, entre otras funciones, brinda asistencia a sus miembros (México forma parte de esta organización) para la planificación y gestión de sus sistemas educativos, ha establecido la necesidad latente de una educación de calidad como elemento sustancial para mejorar el bienestar social y económico de las personas en todo el mundo [3]. Con miras a comprender el estado de la educación en los países, este organismo se ha encargado de realizar una evaluación trienal mediante el Programa para la Evaluación Internacional de Alumnos (PISA por sus siglas en inglés) la cual consiste en la aplicación de una prueba para evaluar los conocimientos y habilidades que han adquirido los alumnos de entre 15 y 16 años.

En México, también se cuenta con organismos encargados de medir el impacto educativo a través de la aplicación de pruebas a distintos niveles de educación como son el Instituto Nacional para la Evaluación de la Educación (INEE), y con el desarrollo de nuevas iniciativas de la Secretaria de Educación Pública (SEP), en particular con la Evaluación Nacional de Logro Académico en Centros Escolares (ENLACE) [4]. Los resultados que se obtienen de estas pruebas normalmente no son sometidos a un riguroso análisis estadístico y así mismo, estos datos pueden contener una enorme riqueza de información que, con la metodología adecuada, puede ser aplicada para la obtención de patrones interesantes que permitan explicar algunos comportamientos. En este sentido la minería de datos permite aplicar diversos algoritmos de agrupamiento como son COBWEB, EM, y K-means [5]. En el caso de K-means se han utilizado en la predictividad de proceso petitorio en una organización pública [6]. Así como para analizar la ejecución de procesos de negocios [7]. Además, para la identificación de los perfiles de estudiantes [8].

Por lo tanto, el objetivo de este trabajo fue aplicar el algoritmo K-Means para el análisis de resultados de la prueba PLANEA 2017.

2. Trabajos relacionados

Una cantidad importante de trabajos han sido realizados por investigadores del área, de los cuáles se han obtenido resultados que han impactado la forma de crear o transformar las políticas públicas encaminadas al mejoramiento de la educación. El enorme esfuerzo que se requiere para lograr esta transformación ha ido involucrando a su paso a especialistas tales como psicólogos, sociólogos, filósofos, pedagogos, matemáticos, y especialistas del área de la informática y del análisis de datos.

Estos estudiosos del tema han descrito de forma cualitativa y cuantitativa la forma en que, directa o indirectamente, factores de índole social, económico y político impactan sobre el rendimiento académico.

Al respecto, puede mencionarse el trabajo coordinado por Raúl Abreu y David Calderón [9], en donde en su informe del Índice Compuesto de Eficacia de los sistemas Escolares 2007 en México, analizaron algunas variables que resultan relevantes en el aprovechamiento educativo a nivel básico encontrando que la infraestructura de las escuelas, el ausentismo de los profesores y la incorporación de los maestros a la carrera magisterial son factores escolares que presentan una correlación significativa con el logro académico de los estudiantes.

Por otra parte, De Hoyos et al. [10] establecen la importancia que, los antecedentes y recursos escolares, factores institucionales y antecedentes familiares tienen sobre la dinámica del aprendizaje en estudiantes de México. Otros factores estudiados son los institucionales, características personales, ambiente del hogar antecedentes familiares del estudiante, los cuales se cree que son factores asociados al logro cognitivo [11-13]. Todd & Wolpin [11] establecen que la acumulación de conocimiento y logros académicos son el resultado de un proceso acumulativo que depende de la historia y herencia familiares, así como de los insumos escolares.

Así mismo, Glewwe y Kremer [12] definen diferentes actores asociados con el desempeño académico dentro de las instituciones de educación tales como los incentivos a la práctica y el ausentismo docentes. Particularmente en matemáticas y ciencias, en un análisis realizado sobre los resultados de la prueba PISA, Fuchs y Woessman [13] aseguran que existe una importante correlación entre factores tales como el nivel de estudios y la ocupación de los padres, así como el número de libros que se leen en la casa con el logro académico en esas áreas. Hanushek [14] por su parte, establece que los resultados en cuanto a la eficiencia cognitiva se refieren, están en función de las políticas públicas que el Estado implementa y en particular en función de la inversión que se hace en materia de Educación.

Una forma de abordar el análisis de los factores involucrados en la efectividad de los procesos de enseñanza-aprendizaje (como aquellos factores mencionados con anterioridad), es mediante el uso de técnicas de manejo de la información propias de la minería de datos que, en su modalidad de aplicación a los temas educativos, se conoce como minería de datos educativa (EDM) y que, desde ya hace algunos años, se encuentra realizando importantes análisis así como aportando soluciones de los más relevantes problemas educativos [15, 16]. En particular, se ha hecho un uso reiterado de las técnicas propias del aprendizaje automático o aprendizaje de máquina, las cuales permiten la búsqueda de información relevante o de patrones que proporcionan una mejor comprensión del comportamiento de los resultados y eventualmente pueden conducir a la predicción de dichos comportamientos.

En la literatura se pueden encontrar ejemplos de la aplicación de técnicas de minería de datos en investigación educativa [17-19], tales como el uso de algoritmos de agrupamiento (los cuales entran dentro de las técnicas del aprendizaje no supervisado) para el análisis de datos en investigación educativa tales como el estudio realizado por Ivancevic en donde, mediante el algoritmo K-medias, evalúa las implicaciones que tiene la selección de un asiento por un estudiante en el salón de clases con las evaluaciones [20].

Este mismo algoritmo fue utilizado por Chang et al. [21] para saber qué tipo de aprendizaje fue el más apropiado para los distintos tipos de estudiantes, lo que permitió que los profesores adaptaran sus materiales didácticos y sus programas de enseñanza

Tabla 1. Clasificación de niveles de aprovechamiento según PLANEA.

Nivel I	Nivel II	Nivel III	Nivel IV
Logro Insuficiente	Logro apenas indispensable	Logro Satisfactorio	Logro Sobresaliente

de acuerdo con las habilidades y necesidades específicas de cada estudiante y, a su vez, una mayor eficiencia académica.

Los algoritmos de aprendizaje automático también han sido utilizados para evaluar el aprendizaje cuando se hace uso de las plataformas electrónicas mediante lo que se conoce como e-learning [22, 23]. Al respecto, Franki y Moudgalya [24] mencionan que, la forma o formas en que los alumnos hacen uso del software es determinante para saber cómo es el aprovechamiento académico al realizar aprendizaje con recursos e-learning.

En cuanto a la evaluación del aprovechamiento académico y variables o factores involucrados, en México se han llevado a cabo diversas investigaciones de análisis de datos sobre los resultados que se obtienen de la aplicación de las diferentes pruebas a diferentes niveles educativos. Raymundo et al. [25] en 2010 por ejemplo, utilizaron un modelo que compara el grado obtenido por la prueba ENLACE con las calificaciones obtenidas por cada bimestre del ciclo escolar, este análisis lo aplicaron a los alumnos de la Ciudad de México durante los años 2006 al 2009, en la educación primaria. Otros autores que se ha involucrado en el área del análisis del aprovechamiento académico mediante las pruebas estandarizadas son Heredia et al. [26] quienes desarrollaron el sistema informático ANCONE y cual determinó las variables de mayor impacto en pruebas académicas derivado del análisis de los resultados de la prueba PLANEA 2015 y reportando que existen algunos atributos tales como las aspiraciones académicas de los alumnos, el nivel de estudios de los padres o los recursos familiares que toman una importante relevancia en el impacto académico.

Según lo expuesto anteriormente, de importancia radical es el estudio de los distintos factores y variables que son actores vitales en el proceso de enseñanza-aprendizaje y que, eventualmente, pueden conducir al éxito o fracaso del alumnado.

3. Materiales y métodos

En este estudio, se procesan los resultados obtenidos de la prueba PLANEA 2017 para estudiantes del último grado del nivel medio superior de la base de datos del INEE y PLANEA [27, 28] y se analiza el impacto que cada una de las variables como ubicación de una institución, tipo de sostenimiento, nivel de marginación, turno y tipo de sistema educativo tiene sobre el rendimiento académico.

Los datos abiertos contienen un total de 16380 registros que corresponden al número total de escuelas evaluadas y de las cuales, para este análisis, se tomó la información de las columnas “entidad”, “nombre de la escuela”, “turno”, “subsistema educativo”, “grado de marginación”, “sostenimiento”, y “porcentaje de alumnos de la escuela en cada nivel de logro” como atributos y donde se tomaron en cuenta los resultados para el área de Lenguaje y Comunicación para cada uno de los 32 estados que conforman la República Mexicana. En su mayoría los datos utilizados son datos de tipo numérico, para el caso de los atributos no numéricos se agregó una clave numérica, así mismo para depurar los datos innecesarios se realizó la limpieza y depuración.

Particularmente, en este trabajo se estudia el impacto que sobre el aprovechamiento académico tienen los atributos "Subsistema Educativo", "Turno Escolar", "Nivel de marginación" y "Financiamiento de la escuela (sostenimiento)".

En la prueba PLANEA se evalúan 21 distintos sistemas educativos de nivel medio superior que se muestran en la Figura 2. Además, se tienen cuatro valores distintos para los turnos como son: discontinuo matutino, vespertino y nocturno.

Por otra parte, PLANEA clasifica el fenómeno de la marginación en cuanto a las carencias que padece la población en 5 niveles distintos; Muy Alto, Alto, Medio, Bajo y Muy Bajo. Estas carencias están en función de parámetros tales como falta de acceso a la educación, carencia de oportunidades sociales, residencia en viviendas inadecuadas, percepción de ingresos, ausencia de capacidades entre otros factores.

En cuanto al financiamiento o sostenimiento se refiere, PLANEA toma en cuenta cuatro distintas formas de procedencia de los recursos como son: Autónomo, Federal, Estatal y Particular.

Los niveles de logro en la prueba PLANEA son un parámetro de suma importancia en la interpretación de nuestros resultados y estos niveles plantean una forma de categorizar a las escuelas y a los estudiantes con base en los conocimientos y habilidades que demuestren. En el Nivel I corresponde al nivel de menor rendimiento académico y el nivel cuatro corresponde al de mayor aprovechamiento como se muestra en la Tabla 1.

3.1. Desarrollo

El procesamiento de los datos se llevó a cabo mediante la implementación del algoritmo K-medias, es una técnica muy conocida como método de agrupación de datos [29], el nombre viene dado porque representa cada uno de los grupos por la media (o media ponderada) de sus puntos, es decir, por sus "centroides". La representación mediante "centroides" brinda un sentido estadístico inmediato, donde se puede conocer una aproximación de las características por grupos. El algoritmo o criterio resulta más eficiente aplicándolo a atributos numéricos. Cuando esto ocurre, una medida de distancia muy común es la distancia Euclídea [7], para este análisis los datos se procesaron con un software desarrollado en lenguaje de programación JAVA permitiendo el seccionamiento de los datos en diferentes grupos o particiones. La forma de calcular los centroides, fue mediante el promedio de los valores numéricos (porcentajes de alumnos por nivel de logro según PLANEA) de los cuatro niveles de logro (Tabla 2) entre los diferentes objetos o registros de cada grupo; los K grupos que se consideraron para comparar las dispersiones de los datos respecto de los centroides fueron construidos mediante todas las posibles combinaciones entre el total de registros y se tomaron los grupos cuyas combinaciones dieran como resultado un valor menor del error cuadrático.

Posteriormente, se llevó a cabo un análisis de las características de cada grupo con el objetivo de buscar algún comportamiento o patrón que nos pudiera ayudar con la interpretación de los resultados. A pesar de que el cálculo se realizó con K=2, K=3 y K=4, en este trabajo únicamente se reportan los resultados para K=3 grupos; para K=2 y K=4 los resultados y comportamientos de los grupos fueron análogos que en el caso K=3. Para llevar a cabo la clasificación, con cada uno de los tres grupos finales obtenidos se calculó el promedio del total de los valores numéricos (porcentajes de

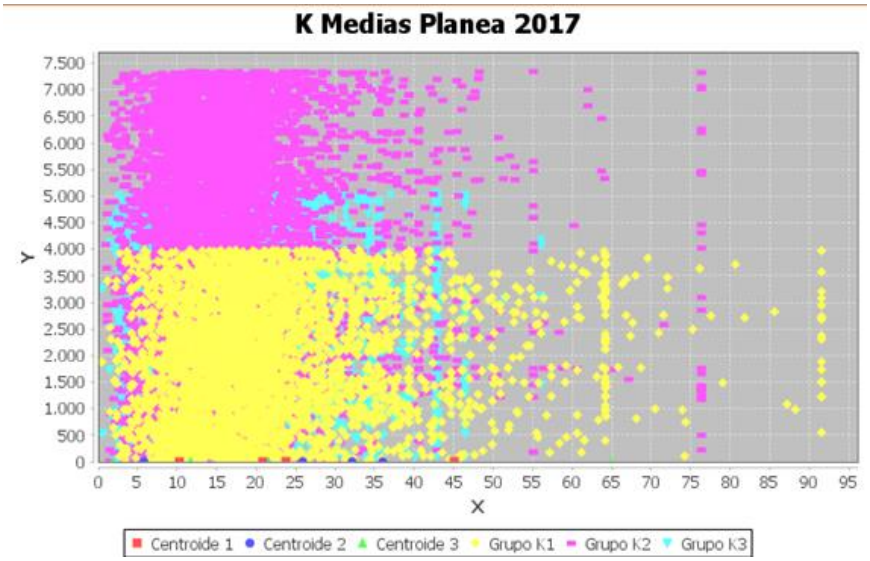


Fig. 1. Visualización de datos de Lenguaje y Comunicación con algoritmo K-medias, con representación de tres centroides y los grupos K1, K2 y K3.

Tabla 2. Distribución por número y porcentaje de escuelas en los tres grupos K1, K2 y K3 para el Área de Lenguaje y Comunicación.

Grupo	Número de escuelas	Promedio de porcentaje por nivel de logro			
		Nivel I	Nivel II	Nivel III	Nivel IV
K1	3988	10.295	20.840	45.084	23.783
K2	7344	32.141	36.012	25.888	5.799
K3	5048	64.900	21.551	11.703	1.845

alumnos) en cada nivel de logro y posteriormente se hizo la clasificación de acuerdo con las cantidades mayoritarias de porcentaje relacionando estas con los niveles de logro de PLANEA. Esto es, de las cuatro variables numéricas que corresponden a los porcentajes de alumnos en cada uno de los cuatro niveles de logro. En tanto que para las variables nominales tales como entidad, turno, subsistema, sostenimiento y marginación fue recuperada su correspondencia una vez que se aplicó el algoritmo a las variables numéricas.

4. Resultados

La visualización de los datos para el Área de Lenguaje y Comunicación en los tres grupos analizados K1, K2 y K3 se puede apreciar en la Figura 1. Para el grupo K=3 se realizaron 22 iteraciones para separar la base de datos PLANEA en donde los individuos comparten características similares.

Para el grupo K1 (3988 escuelas), grupo K2 (7344 escuelas) y grupo K3 (5048 escuelas).

En la Tabla 3 se muestra la cantidad de escuelas o instituciones en las que quedaron las particiones, así como los promedios de cada uno de los niveles de logro (promedios de las columnas) para cada grupo.

Con base en las cantidades mayoritarias de los promedios de escuelas mostrados en la Tabla 2, se hace una correspondencia de cada grupo con un nivel de logro. De esta forma, para el grupo K1 se puede ver que el porcentaje mayor (45.08%) corresponde al nivel III de logro, por lo que se puede decir que el grupo K1 concentra una cantidad importante de escuelas en nivel III y que en conjunto con el nivel IV (23.7%) podemos clasificar este grupo como de logro satisfactorio según los rangos propuestos por PLANEA.

Así mismo, para el grupo K2, la cantidad mayoritaria del promedio se encuentra en el nivel II (36.01%) por lo que se clasifica este grupo como de nivel de logro apenas indispensable. Finalmente, es claro que el grupo K3 tiene que clasificarse como un grupo de nivel de logro insuficiente al encontrarse la mayoría del porcentaje de escuelas (64.9%) dentro del nivel I de logro académico (logro insuficiente según PLANEA). Los grupos ya clasificados, según el criterio tomado para el análisis de datos son: K1 para satisfactorio, K2 logro apenas indispensable y K3 logro insuficiente.

En congruencia con este criterio de clasificación propuesto en este trabajo, adicionalmente en la Tabla 2 se puede observar el siguiente comportamiento: en el nivel IV de PLANEA de logro sobresaliente, se tiene los menores porcentajes de escuelas para cada uno de los grupos K, es decir, son pocas las escuelas con nivel académico destacado. Otro resultado interesante y que resalta esta misma congruencia es que el grupo K3 que se clasifico como de logro insuficiente, tiene un porcentaje mínimo de escuelas en el nivel IV (1.845) de PLANEA mientras, que en el grupo K1 de logro satisfactorio se tiene un mayor porcentaje de escuelas dentro del nivel IV (23.783), ósea, entre mayor es el nivel de desempeño académico (grupo K1), mayor es el porcentaje de escuelas dentro del nivel sobresaliente PLANEA (IV) en comparación con el porcentaje de escuelas para el mismo nivel PLANEA en el grupo K3 de menor nivel de desempeño académico.

En la Tabla 3 se muestra la cantidad y porcentajes de escuelas ordenados alfabéticamente según la entidad federativa y clasificadas en los tres grupos K1, K2 y K3. Se puede ver que el estado que concentra una mayor cantidad de escuelas en el grupo K1 (de logro satisfactorio) es Jalisco con un 40.09% de sus escuelas mientras que Chiapas es el estado que concentra un mayor porcentaje de sus escuelas evaluadas en el nivel K3 de logro insuficiente con un 63.17%.

Así mismo se puede ver que Jalisco, Ciudad de México y Querétaro son los estados que tienen un mayor porcentaje de escuelas dentro del grupo K1 de logro satisfactorio; a pesar de que se encuentran en los primeros lugares, aún tienen menos del 50% de sus escuelas con el nivel de logro satisfactorio. Por otra parte, se pueden identificar los estados de Chiapas, Guerrero y Tabasco como los estados con mayor rezago educativo al ubicarse como los primeros lugares en porcentaje de escuelas dentro del grupo K3 de logro insuficiente.

Así mismo, se realiza un análisis del comportamiento de la distribución de escuelas en cada uno de los tres grupos obtenidos en el caso del atributo “turno”, información que puede ser vista en la Tabla 4.

Tabla 3. Clasificación de los grupos obtenidos de acuerdo con los niveles PLANEA por entidad federativa para el Área de Lenguaje y Comunicación.

Entidad Federativa	Porcentaje de escuelas por grupo		
	K1	**K2**	**K3**
Aguascalientes	28.037	48.131	23.832
B. California	26.722	51.791	21.488
B. California S	33.028	43.119	23.853
Campeche	25.373	47.761	26.866
Chiapas	9.404	27.417	63.179
Chihuahua	18.333	41.042	40.625
CDMX	39.430	42.755	17.815
Coahuila	25.051	35.934	39.014
Colima	35.345	37.931	26.724
Durango	23.077	44.056	32.867
Guanajuato	21.842	52.463	25.696
Guerrero	11.873	27.425	60.702
Hidalgo	25.714	55.102	19.184
Jalisco	40.098	48.533	11.369
México	22.253	53.139	24.608
Michoacán	16.254	36.572	47.173
Morelos	28.829	46.847	24.324
Nayarit	13.580	43.210	43.210
Nuevo León	22.372	35.310	42.318
Oaxaca	14.580	51.347	34.073
Puebla	35.272	44.374	20.354
Querétaro	38.610	50.965	10.425
Quintana Roo	30.244	46.341	23.415
San Luis Potosí	14.004	38.293	47.702
Sinaloa	25.822	40.845	33.333
Sonora	35.802	42.284	21.914
Tabasco	13.095	30.952	55.952
Tamaulipas	22.613	41.709	35.678
Tlaxcala	19.681	56.915	23.404
Veracruz	23.110	48.780	28.110
Yucatán	29.178	49.008	21.813
Zacatecas	23.985	52.768	23.247

Tabla 4. Clasificación de los grupos obtenidos de acuerdo con los niveles PLANEA por turno para el Área de Lenguaje y Comunicación.

TURNO	Porcentaje de escuelas por grupo		
	K1	**K2**	**K3**
Discontinuo	60	34	6
Matutino	27.997	44.499	27.504
Vespertino	14.727	46.166	39.107
Nocturno	8.955	34.328	56.716

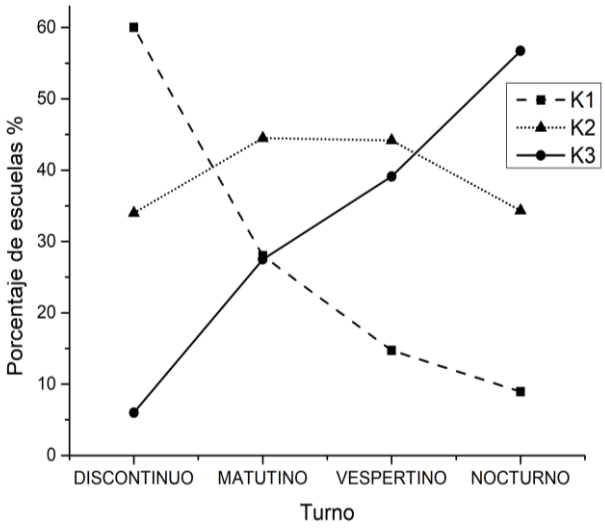


Fig. 2. Porcentaje de escuelas evaluadas por PLANEA por turno para el Área de Lenguaje y Comunicación: para el grupo K1 - logro satisfactorio, k2 - logro apenas indispensable, K3 – logro insuficiente.

Con base en los resultados reportados en la Tabla 4 y considerando los grupos K1 y K3 como representativos (como se hizo en el caso del análisis por entidad), en la Figura 2 se puede ver que el turno matutino es preponderante en el grupo K1 de logro satisfactorio sobre el turno vespertino, mientras que el porcentaje de escuelas evaluadas del turno vespertino es mayoritario en el grupo K3 de logro insuficiente.

Esto indica que existe una importante correlación entre el turno y el aprovechamiento académico siendo el turno matutino asociado a una mayor eficiencia. A pesar de que los turnos discontinuo y nocturno no tienen una cantidad significante de escuelas, por lo que no se pueden comparar con los turnos matutino y vespertino, se puede ver una tendencia que indica que el turno discontinuo tiene una preponderancia general en el grupo K1 y el turno nocturno en el grupo K3 para el caso de la evaluación en Lenguaje y Comunicación.

Análogamente a lo que se hizo en el caso anterior, en la Figura 3 se muestra el comportamiento de los porcentajes de escuelas distribuidas en los grupos K1, K2 y K3 para el análisis por subsistema.

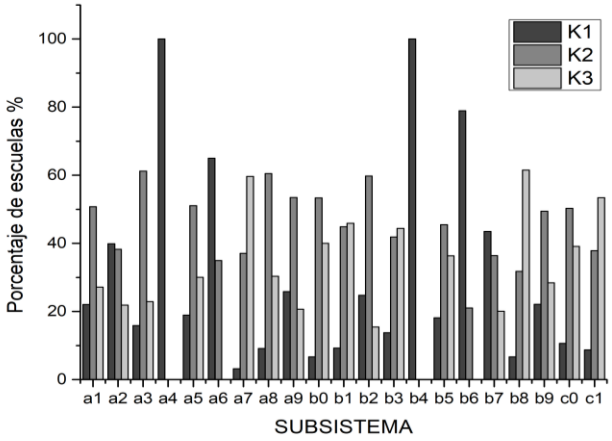


Fig. 3. Porcentaje de escuelas evaluadas por PLANEA por subsistema para el Área de Lenguaje y Comunicación: el grupo K1 - logro satisfactorio, K2 - logro apenas indispensable, K3 - logro insuficiente (a1-BACH ESTATAL DGE-CGE, a2-BACHI AUTONOMO, a3-CECYTE, a4-CETI, a5-COBACH, a6-COLBACH MEX, a7-CONALEP DF-OAX, a8-CONALEP EDOS, a9-DGB, b0-DGECYTM, b1-DGETA, b2-DGETI, b3-EMSAD, b4-IPN, b5-OTRAS ESTATALES, b6-OTRAS FEDERALES, b7-PARTICULARES, b8-PREECOS, b9-PREFECOS, c0-TELEBACHILLERATOS, c1-TELEBACHILLERATOS COMUNITARIOS).

Cabe destacar que, a pesar de que en el subsistema del Instituto Politécnico Nacional solo se evaluaron 35 escuelas en comparación con otras escuelas donde la cantidad fue mucho mayor, la totalidad de escuelas del IPN se encuentran dentro del grupo K1 mientras que sistemas como subsistemas tales como el CONALEP DF-OAX o los sistemas PREECOS (Preparatorias Estatales por Cooperación), tienen una cantidad mayoritaria de escuelas dentro del grupo K3 de nivel insuficiente.

Con respecto al nivel de marginación, en la Figura 4 se puede ver que existe una importante correlación entre el factor pobreza y el nivel de aprovechamiento académico al observar que, si el nivel de marginación es muy bajo, entonces se tiene una cantidad mayoritaria de escuelas en el grupo K1 (de logro satisfactorio) mientras que, si el nivel de marginación es muy alto, las escuelas se distribuyen mayoritariamente en el grupo K3 de nivel insuficiente.

4. Conclusiones y trabajo a futuro

En la presente investigación se ha aplicado al algoritmo K-medias para el análisis de los datos obtenidos a partir de los resultados de la prueba PLANEA llevada a cabo en el año 2017 en estudiantes del último grado del nivel medio superior en el Área de Lenguaje y Comunicación. Mediante el algoritmo se obtuvieron 3 grupos (K=3) a los que, aplicando nuestro criterio de acuerdo, a sus características, podemos clasificar como de logro satisfactorio (K1), logro apenas indispensable (K2) y logro insuficiente (K3).

Con base en nuestros resultados y tomando a los grupos K1 y K3 como representativos, se puede ver que nuestro criterio de clasificación es adecuado ya que se pudo observar una mayor cantidad de escuelas del grupo K1 en el nivel IV de

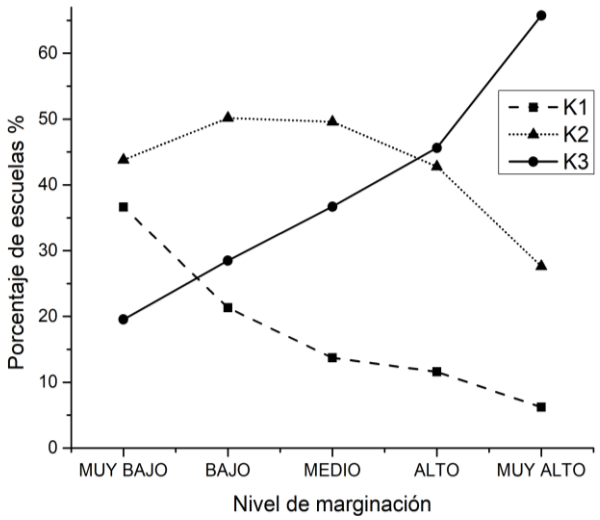


Fig. 4. Porcentaje de escuelas evaluadas por PLANEA por nivel de marginación para el Área de Lenguaje y Comunicación: para el grupo K1 - logro satisfactorio, k2 - logro apenas indispensable, K3 - logro insuficiente.

PLANEA (sobresaliente) en comparación con el número de escuelas del grupo K3 en ese mismo nivel; en contraste a este resultado, se obtiene que el grupo K3 concentra la mayoría de sus escuelas en el nivel I de logro insuficiente.

En cuando al atributo Entidad Federativa, reportamos que los estados de Jalisco, Ciudad de México, Querétaro, Sonora y Colima son aquellos que ostentan un mayor aprovechamiento académico al tener entre el 35 y el 40 por ciento de sus escuelas en el grupo K1, mientras que los estados de Chiapas, Guerrero, Tabasco, San Luis Potosí y Michoacán son los estados que tienen una mayor concentración de sus escuelas en el grupo K3 (entre 47 y 63 por ciento). Estos resultados son congruentes con aquellos que se reportan en la Encuesta Nacional de los Hogares 2016 realizada por el Instituto Nacional de Estadística y Geografía (INEGI) [30] en donde, en particular, los Estados de Chiapas, Michoacán y Guerrero se encuentran entre los estados de mayor rezago educativo, mientras que, en esta misma encuesta, la Ciudad de México y Sonora se encuentran entre los estados de menor rezago tal y como es reportado en este trabajo.

En cuanto al análisis de la relación que tiene el aprovechamiento académico con el turno, en este trabajo se obtiene que el turno matutino domina en el grupo K1 mientras que en el turno vespertino es dominante el grupo K3, esto significa que el turno matutino se caracteriza por tener un mayor aprovechamiento. Por otra parte, respecto al atributo Subsistema, con base en nuestros resultados es de destacar la eficiencia de algunas instituciones educativas como el Instituto Politécnico Nacional o los Centros de Enseñanza Técnica Industrial (CETI) cuya totalidad de escuelas se encontraron en el grupo K1. Respecto del nivel de marginación, otra de las variables estudiadas en este trabajo, se pudo observar que existe una alta correlación entre el nivel marginación y el aprovechamiento académico al ver que un mayor porcentaje de escuelas que pertenecen al grupo K1, de logro satisfactorio, se encuentran en zonas que tienen niveles de

marginación bajo o muy bajo, mientras que aquellas escuelas que se ubican en zonas de alta marginación se clasificaron dentro el grupo K3 de logro insuficiente. Una extensión del presente trabajo sería realizar el mismo estudio, pero ahora tomando en cuenta los resultados que se obtuvieron en el Área de Matemáticas (en el mismo año 2017) y ver si hay un patrón de comportamiento entre ambas áreas y así mismo, realizar el estudio tomando en cuenta los resultados de la prueba de años posteriores para ver la evolución de dicho comportamiento.

Referencias

1. Rivas, T., González, M.J., Delgado, M.: Descripción y propiedades psicométricas del test de evaluación del rendimiento académico (TERA). Interamerican Journal of Psychology, 44(2), pp. 279–290 (2010)
2. Martínez, F.: La evaluación formativa del aprendizaje en el aula en la bibliografía en inglés y francés. Revista Mexicana de Investigación Educativa, 17(54), pp. 849–875 (2012)
3. Organización para la Cooperación y Desarrollo Económico: El trabajo de la OCDE sobre educación y competencias, https://www.oecd.org/education/El-trabajo-de-la-ocde-sobre-educacion-y-competencias.pdf (2019)
4. Martínez, F.: Las evaluaciones educativas en América Latina. México: Instituto Nacional para la Evaluación de la Educación (2008)
5. Garre, M., Cuadrado, J.J., Sicilia, M.A., Rodríguez, D., Rejas, R.: Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. REICIS, Revista Española de Innovación, Calidad e Ingeniería del Software, 3(1), pp. 6–22. https://www.redalyc.org/articulo.oa?id=922/92230103 (2007)
6. Martínez-Abad, F., Hernández-Ramos, J.P.: Técnicas de minería de datos con software libre para la detección de factores asociados al rendimiento. REXE, Revista de Estudios y Experiencias en Educación, 2(2), pp. 135–145. https://www.redalyc.org/articulo.oa?id=2431/243156768012 (2018)
7. Pérez, S., Jaime, J., Espín, R.: Departamento ingeniería industrial, Instituto Superior Politécnico José Antonio Echeverría. Revista investigación operacional, 33(3), pp. 210–221 (2012)
8. Chaparro, A., González, C., Caso, J.: Familia y rendimiento académico: configuración de perfiles estudiantiles en secundaria. Revista electrónica de investigación educativa, 18(1), pp. 53–68. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1607-40412016000100004&lng=es&tlng=es (2016)
9. Abreu, R., Martín del Campo, D.C.: Índice compuesto de eficacia de los sistemas escolares. Mexicanos Primero Visión 2030 A.C, Fundación IDEA A.C. (2007)
10. De Hoyos, R., Espino, J.M., García, V.: Determinantes del logro escolar en México. Primeros resultados utilizando la prueba ENLACE media superior. El Trimestre Económico, LXXIX 4(316), Fondo de Cultura Económica, pp. 783–811 (2012)
11. Todd, P.E., Wolpin, K.I.: On the specification and estimation of the production function for cognitive achievement. Economic Journal, Royal Economic Society, 113(485), pp. 3–33, https://ideas.repec.org/a/ecj/econjl/v113y2003i485pf3-f33.html (2003)
12. Glewwe, P., Kremer, M.: Chapter 16 Schools, Teachers, and Education Outcomes in Developing Countries. In: Hanushek, E., Welch, F. (Eds.), Handbook of the Economics of Education, pp. 945–1017 (2006)
13. Thomas, F., Ludger, W.: What accounts for international differences in student performance? a re-examination using PISA data. Empirical Economics, 32(2-3), pp. 433–464 (2007)

14. Hanushek, E.: The economics of schooling: Production and efficiency in public schools. Journal of Economic Literature, 24(3), pp. 1141–1177 (1986)
15. Romero, C., Ventura, S.: Educational data mining: A survey from 1995 to 2005. Expert Syst. Appl., 33, pp. 135–146 (2007)
16. Romero, C., Ventura, S.: educational data mining: A review of the state of the Art. IEEE Transactions on Systems Man and Cybernetics Part C, (Applications and Reviews) 40(6), pp. 601–6182 (2007)
17. Beck, J. E., Woolf, B.: High-level student modeling with machine learning. In: Proc. 5th Int. Conf. Intell. Tutoring Syst, pp. 584–593 (2000)
18. Baker, R., Corbett, A., Wagner, A.: Off-task behavior in the cognitive tutor classroom: When students game the system. In: Proc. SIGCHI Conf. Human Factors Comput. Syst., pp. 383–390 (2004)
19. Wang, Y., Liao, H.: Data mining for adaptive learning in a TESL-based e-learning system. Expert Syst. Appl., 38(6), pp. 6480–6485 (2011)
20. Ivancevic, V., Celikovic, M., Lukovic, I.: The individual stability of student spatial deployment and its implications. Int. Symp. Comput. Edu. (SIIE), pp. 1–4 (2012)
21. Chang, W., Chen, S., Li, M., Chiu, J.: Integrating IRT to clustering student's ability with K-means. In: Proc. 4th Int. Conf. Innov. Comput., Inf. Control (ICICIC), pp. 1045–1048 (2009)
22. Aher, S., Lobo, L.: Applicability of data mining algorithms for recommendation system in e-learning. In: Conf. Adv. Comput., Commun. Inform. (ICACCI), pp. 1034–1040 (2012)
23. Zheng, Q., Ding, J., Du, J., Tian, F.: Assessing method for E-learner clustering. In: 11th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD), pp. 979–983 (2007)
24. Eranki, K., Moudgalya, K.: Evaluation of Web based behavioral interventions using spoken tutorials. In: IEEE 4th Int. Conf. Technol. Edu., pp. 38–45 (2012)
25. Campos, R., Romero, F.: Desempeño educativo en México: la prueba ENLACE. In: Serie documentos de trabajo del Centro de Estudios Económicos, El Colegio de México, Centro de Estudios Económicos, https://ideas.repec.org/p/emx/ceedoc/2010-19.html (2010)
26. Heredia, A., Chi, A., Guzmán, A., Martínez, G.: ANCONE: An interactive system for mining and visualization of students information in the context of PLANEA 2015. Computación y Sistemas, 24(1), pp. 151–176 (2020)
27. Instituto Nacional para la Evaluación de la Educación: Evaluaciones de Logro referidas al Sistema Educativo Nacional. Último grado de Educación Media Superior (2016-2017), INEE. https://www.inee.edu.mx/evaluaciones/planea/media-superior-ciclo-2016-2017 (2020)
28. Plan Nacional para la Evaluación de los Aprendizajes: Bases de datos de escuelas, PLANEA. http://planea.sep.gob.mx/ms/base_de_datos_2017/ (2019)
29. Tar, J., Bitó, J.F., Rudas, I., Várkonyi, T.A.: Decentralized Adaptive Control with Fractional Order Elimination of Obsolete Information. In: 4th International Conference on Emerging Trends in Engineering and Technology, ICETET 2011 (2011)
30. Instituto Nacional de Estadística y Geografía: Encuesta Nacional de los Hogares. https://www.inegi.org.mx/programas/enh/2016/ (2016)

ISSN 1870-4069

Estudio comparativo de algoritmos de súper resolución de una sola imagen basados en aprendizaje profundo

Leandro José Rodríguez Hernández[1], Humberto de Jesús Ochoa Domínguez[1], Osslan Osiris Vergara Villegas[2], Juan Humberto Sossa Azuela[3]

[1] Universidad Autónoma de Ciudad Juárez,
Departamento de Ingeniería Eléctrica y Computación,
México
[2] Universidad Autónoma de Ciudad Juárez,
Departamento de Ingeniería Industrial y Manufactura,
México
[3] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Robótica y Mecatrónica,
México

{al194726, hochoa}@uacj.mx, overgara@uacj.mx,
hsossa@cic.ipn.mx

Resumen. El problema de la súper resolución comprende un conjunto de métodos de procesamiento digital de imágenes; su objetivo es incrementar la resolución de las imágenes para mejorar su calidad visual. En años recientes, con el auge del aprendizaje profundo y el aumento de la capacidad computacional, han surgido una serie de métodos para súper resolución basados en aprendizaje profundo. Este trabajo presenta un estudio comparativo de cuatro métodos recientes del estado del arte de súper resolución utilizando aprendizaje profundo. Se analiza su fundamento teórico y sus arquitecturas. Se comparan y reportan sus rendimientos usando diferentes bases de datos utilizadas en súper resolución.

Palabras claves: Súper resolución, aprendizaje profundo, redes neuronales convolucionales.

Comparative Study of Single-Image Super Resolution Algorithms Based on Deep Learning

Abstract. The super resolution problem involves a set of digital image processing methods; its objective is to increase the resolution of the images to improve their visual quality. In recent years, with the rise of deep learning and increasing computational capacity, a number of methods for super resolution based on deep learning have emerged. This work presents a comparative study of four recent super resolution state-of-the-art methods using deep learning. Its theoretical foundation and architectures are analyzed. Their performances are compared and reported for different databases used in super resolution.

Keywords: Super resolution, deep learning, convolutional neural networks.

1. Introducción

La degradación de las imágenes digitales es resultado de una serie de factores que influyen en el proceso de adquisición y digitalización de las imágenes. Es deseable que las imágenes digitales contengan la mayor cantidad de información posible. En algunas aplicaciones, como por ejemplo las imágenes médicas, puede significar la detección temprana y eficaz de una enfermedad. Con el objetivo de mejorar la calidad de las imágenes digitales, se producen mejoras continuas en el hardware de los dispositivos de adquisición; sin embargo, las mejoras de hardware son altamente costosas y tienen límites físicos para su implementación. Existen áreas, donde las imágenes que se logran adquirir son de baja calidad, ejemplo: las imágenes médicas. Para abordar este problema han sido propuestos no solo mejoras en el hardware de los equipos. También mediante la utilización de técnicas del procesamiento digital de imágenes, mediante las cuales se intenta mejorar la calidad de imágenes ya adquiridas.

La súper resolución de una sola imagen (*SISR*, por sus siglas en inglés) es un problema clásico en el procesamiento de imágenes digitales. Su objetivo es incrementar la resolución de una imagen de baja resolución (*LR*, por sus siglas en inglés) para obtener una imagen de mayor resolución y recuperar información de alta frecuencia [1, 2, 3]. Existen diversas aplicaciones en diferentes áreas como video vigilancia [4, 5], sensado remoto [6] e imágenes médicas [7, 8]. En general, para *SISR* se supone que la imagen de *LR* es una versión submuestreada y ruidosa de la imagen de alta resolución (*HR*, por sus siglas en inglés). Para abordar el problema de la súper resolución, se han propuesto diferentes métodos. Los métodos basados en interpolación han sido ampliamente utilizados, debido al bajo costo y complejidad computacional [9, 10]. Tienden a difuminar los detalles de alta frecuencia, sin embargo, pueden producir escalas muy grandes.

Otro enfoque clásico de *SISR* se fundamenta en el aprendizaje o algoritmos basados en ejemplos [11-15] donde se explota la propiedad de auto similitud para generar imágenes de *HR* a partir de la entrada de *LR*. Los métodos basados en ejemplos utilizan una base de datos, también llamada diccionarios, que contiene pares de parches (porciones de la imagen) de alta y baja resolución para establecer relaciones de mapeo entre los parches de la imagen de *LR* y los parches de la imagen de *HR*; tal relación de mapeo se utiliza posteriormente para estimar nuevas imágenes de *HR* a partir de la entrada *LR*.

De acuerdo con las formulaciones de mapeo, la *SISR* fundamentada en el aprendizaje utiliza métodos que usan vecindades [16, 17], que supone que los pares de parches *LR-HR* tienen geometrías similares; basados en regresión [18, 19], que construyen un modelo de regresión para estimar las relaciones *LR-HR* y, finalmente, los métodos que utilizan la escasez [11,12], en los cuales se calcula el código escaso del parche de *LR* para estimar el parche de *HR*. Estos métodos requieren que se busque en un amplio conjunto de datos de referencia patrones similares.

Recientemente, se han propuesto algunos métodos basados en redes neuronales convolucionales (*CNN*, por sus siglas en inglés) [20-25], los cuales recuperan imágenes de *HR* aprendiendo un mapeo no lineal entre los parches de *LR* y de *HR*.

Para mejorar los resultados de la reconstrucción, las redes se entrenan utilizando conjuntos grandes de datos. En comparación con los métodos basados en interpolación y en ejemplos, los enfoques que utilizan una *CNN* proporcionan un rendimiento superior en calidad de restauración, dando como resultado mejoras en la relación señal a ruido (*PSNR*, por sus siglas en inglés) y en similitud estructural (*SSIM*, por sus siglas en inglés).

El objetivo del presente trabajo es presentar y describir la teoría y el diseño de cuatro algoritmos de súper resolución basados en aprendizaje profundo, así como comparar su rendimiento. El objetivo del estudio comparativo es proporcionar al investigador, elementos que le permitan formar una base teórica y un acercamiento a las principales arquitecturas y tendencias dentro del área, así como sus posibles aplicaciones.

Las arquitecturas seleccionadas fueron elegidas porque dentro de la gran diversidad que existen, han marcado las pautas, representaron hitos y tendencias de las cuales se derivan muchos otros trabajos, además de que hasta donde sabemos, no existe un estudio comparativo de algoritmos de *SISR* basados en aprendizaje profundo previo al que se muestra en el presente trabajo. El resto del artículo se organiza de la siguiente manera. En la sección 2 se describen los algoritmos utilizados en la comparación. En la sección 3 se describe la experimentación y en la sección 4 los resultados obtenidos son presentados. En la sección 5 se discuten los resultados y finalmente en la sección 6 se presentan las conclusiones.

2. Súper resolución de una sola imagen basada en aprendizaje profundo

El aprendizaje automático ha demostrado su eficacia como herramienta para establecer modelos y relaciones entre conjuntos de datos y se ha aplicado a diversos campos de la ciencia [26, 27]. La súper resolución es un problema inverso, también conocido como problema mal planteado, porque no están disponibles todos los elementos para establecer soluciones únicas, es decir, varias imágenes de alta resolución pueden corresponder a la misma imagen de baja resolución. Algunas técnicas de aprendizaje automático, específicamente el aprendizaje profundo, se han utilizado como herramientas para establecer relaciones óptimas entre una imagen de *LR* y su correspondiente imagen de *HR* [20, 22, 23].

La Fig. 1 muestra el esquema general para súper resolución. La imagen x es la imagen de baja resolución de entrada a la red. Los filtros de convolución W_i son los encargados de extraer los mapas de características $F_i(x)$. Opcionalmente, algunos autores incorporan un término de "*bias*" o sesgo, que se denota como B_i. La salida de la red es la imagen $\hat{y}$, que es la imagen de súper resolución. La imagen y de *HR*, es la imagen ideal o "*ground truth*", utilizada para el entrenamiento.

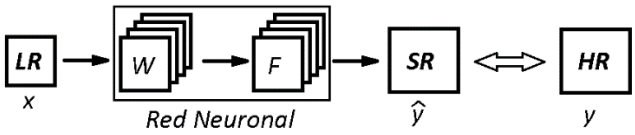


Fig. 1. Esquema general de súper resolución basada en aprendizaje profundo.

2.1. Arquitecturas poco profundas

El término "arquitecturas poco profundas" se utiliza para referirse a redes neuronales con pocas capas. Aunque no existe un límite rígido, una red con menos de diez capas se considera poco profunda. Este tipo de arquitecturas tiene la ventaja de ser más fácil de entrenar, produce resultados superiores a los métodos tradicionales de *SISR* mencionados anteriormente, manteniendo un nivel medio de complejidad computacional. En los trabajos de la Red Neuronal Convolucional para Súper Resolución (*SRCNN* por sus siglas en inglés) [20] y de la Red Neuronal Convolucional Rápida para Súper Resolución (*FSRCNN* por sus siglas en inglés) [22], se utilizaron arquitecturas de redes neuronales poco profundas.

Red Neuronal Convolucional para Súper Resolución (*SRCNN*): La red neuronal *SRCNN* [20] fue el primer trabajo que utilizó una *CNN* para *SISR*, demostrando un rendimiento superior a los métodos anteriores de la literatura, tanto en velocidad como en calidad de restauración. Los autores presentan un método que aprende la relación de mapeo entre las imágenes *LR* y *HR*. El mapeo está representado por una *CNN* que toma la imagen *LR* como entrada y genera la imagen *HR*. La arquitectura se muestra en la Fig. 2. Es una *CNN* de tres capas, donde los tamaños de los filtros son 64 x 1 x 9 x 9, 32 x 64 x 5 x 5 y 1 x 32x 5 x 5 respectivamente. Las funciones de estas tres capas de transformaciones no lineales son: extracción de características, mapeo no lineal y reconstrucción de la imagen súper resuelta.

1. *Capa de extracción de características.* Antes de ingresar a la red *SRCNN*, la entrada de baja resolución se escala al tamaño deseado mediante interpolación bicúbica. Posteriormente, la primera capa realiza una convolución estándar de la imagen escalada con los coeficientes de los filtros entrenados, seguida de una unidad ReLU para obtener el mapa de características $F_1(x)$:

$$F_1(x) = \max(0, W_1 * x + B_1) . \quad (1)$$

2. *Capa de mapeo no lineal.* Es un mapeo del vector n1-dimensional al vector n2-dimensional. El número de características se reduce de 64 a 32. Esta operación de mapeo no lineal se realiza mediante la ecuación:

$$F_2(x) = \max(0, W_2 * F_1(x) + B_2) . \quad (2)$$

3. *Capa de reconstrucción:* Después del mapeo, se obtiene la imagen de súper resolución mediante una capa de convolución final:

$$F(x) = \max(0, W_3 * F_2(x) + B_3) . \quad (3)$$

En esta red se utilizó el error cuadrático medio (*MSE* por sus siglas en inglés) como función de pérdida y el algoritmo gradiente descendiente estocástico (*SGD* por sus siglas en inglés) para resolver el problema de optimización:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \|F(x_i) - y_i\|^2 . \quad (4)$$

La Fig. 2 muestra la arquitectura general de la red *SRCNN*.

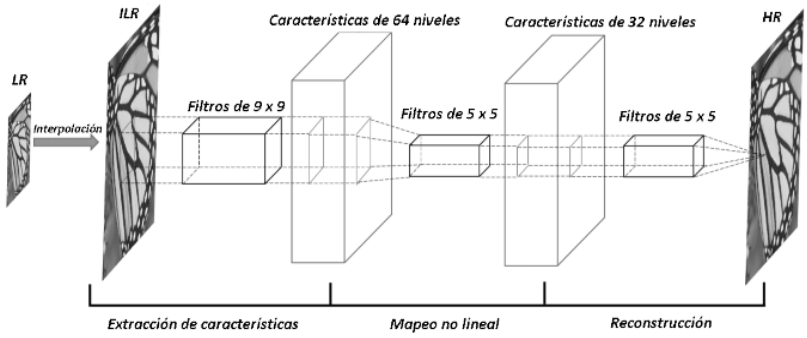


Fig. 2. Red SRCNN [20].

Red Neuronal Convolucional Rápida para Súper Resolución (*FSRCNN*): Dong et al. [19] propusieron la red *FSRCNN* para acelerar la red *SRCNN* [17] a través de una *CNN* de estructura más compacta. El procedimiento de *SISR* se logró más rápidamente y obtuvieron mejores resultados. Rediseñaron la estructura *SRCNN* en tres aspectos principales: introdujeron una capa de deconvolución al final de la red, las capas de mapeo se modificaron reduciendo las dimensiones de las características de entrada antes del mapeo y se usaron filtros de menor tamaño, pero se usaron más capas de mapeo. El modelo propuesto alcanzó una velocidad 40 veces mayor que la red *SRCNN*, con una calidad de restauración de la imagen superior a la de los métodos clásicos. La red *FSRCNN* se organiza en cinco fases principales que son:

1. Extracción de características: mediante la utilización de filtros de 5 × 5 para obtener *d* mapas de características.
2. Reducción de las dimensiones: se utiliza una convolución 1 × 1 para reducir la cantidad de mapas de características de *d* a *s*.
3. Mapeo no lineal: mediante la utilización de filtros de 3 × 3 se realiza un mapeo entre las características baja resolución y alta resolución.
4. Expansión: la convolución 1 × 1 se realiza para aumentar la cantidad de mapas de características de *s* a *d*.
5. Deconvolución: se utilizan filtros de 9 × 9 para reconstruir la imagen de *HR*.

El error cuadrático medio (*MSE*) se usó como la función de pérdida en este método:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \|F(x_i) - y_i\|^2 . \tag{5}$$

La Fig. 3 muestra la estructura general de la red *FSRCNN*.

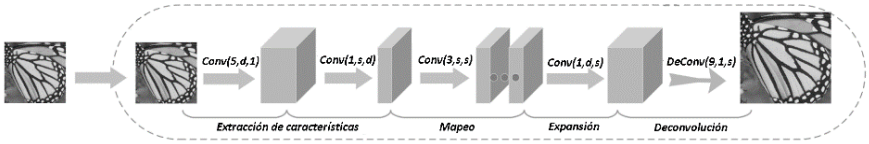


Fig. 3. Red FSRCNN [22].

2.2. Arquitecturas profundas

Una red más profunda tiene una arquitectura más compleja (Tabla 4) y puede lograr mejores resultados, aunque el costo computacional en la etapa de entrenamiento y ejecución aumenta considerablemente. Se considera una red muy profunda si tiene más de diez capas. Los autores de la Red muy profunda para Súper Resolución ("Very Deep Super Resolution" *VDSR* por sus siglas en inglés) [23] y de la Red profunda y recursiva para Súper Resolución **(**"Deep Recursive Convolutional Network" *DRCN* por sus siglas en inglés) [24] utilizaron arquitecturas de redes neuronales muy profundas para aplicarlas a sus trabajos en *SISR*.

Las dos redes, basan su entrenamiento en el aprendizaje residual, el cual permite pasar información de la imagen original magnificada (a la escala deseada mediante un filtro bicúbico), a las capas finales, lo que contribuye a reducir los efectos adversos del desvanecimiento del gradiente [28]. Generalmente, la información se pasa de la primera capa a la última, donde finalmente se suman la información procesada por la red y la información inicial. Para el caso de *SISR*, la imagen escalada a la entrada se asume que es la información de baja frecuencia, ya que carece de detalles y bordes bien definidos. La red neuronal recupera la información de alta frecuencia y, finalmente, suma las dos informaciones para obtener la imagen de súper resolución.

Red muy Profunda para Súper Resolución (*VDSR*): En [23] se propone la red *VDSR* para *SISR*. El método utiliza una red convolucional muy profunda inspirada en la red *VGG* [29], donde el aumento en la profundidad de la red contribuye a una mejora significativa en la precisión. El modelo usa 20 capas con pequeños filtros en cascada para lograr utilizar la información contextual sobre grandes regiones de la imagen de manera más eficiente. Para resolver el problema de la velocidad de convergencia durante el entrenamiento, se propuso utilizar una arquitectura residual, pasando la información de entrada hasta la última capa, para añadirla con la información procesada por la red y tasas de entrenamiento muy altas (104 veces más altas que la *SRCNN*). Los gradientes se acotan entre [- θ / γ; θ / γ], donde γ denota la tasa de aprendizaje actual; θ se ajusta a un valor pequeño para evitar el problema de la explosión del gradiente durante la etapa de entrenamiento. La arquitectura de la red *VDSR* se organiza de la siguiente manera:

1. La imagen de *LR* se interpola y se obtiene la imagen *ILR* que se ingresa a la red.
2. La imagen *ILR* pasa a través de las *(D-1)* capas Conv y ReLU.
3. Continúa la *D-ésima* capa residual (Conv.D (Residual).
4. Finalmente, la salida se suma con la imagen *ILR* y se obtiene la imagen *HR*.

En la red residual, la imagen de entrada de baja resolución se agrega a la salida de la red para obtener la imagen *HR*. Por lo tanto, la función de pérdida es:

$$\frac{1}{2}\|r - F(x)\|^2 , \tag{6}$$

donde $F(x)$ es la salida de la red y $r = Y - X$. Se aprenden los errores residuales entre la salida y la entrada en lugar de aprender la salida *HR* directamente.

La Fig. 4 muestra la arquitectura general de la red *VDSR*.

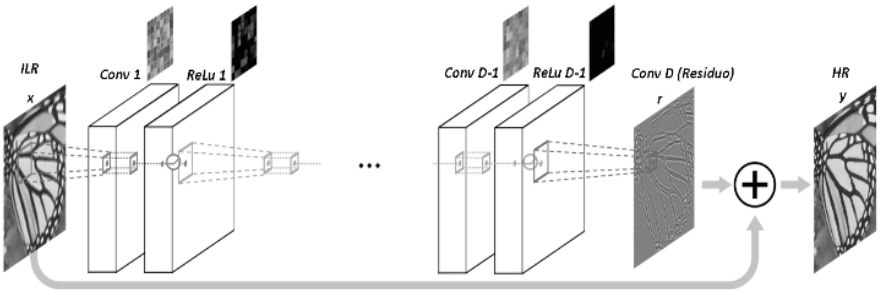


Fig. 4. Red VDSR [23].

Red Profunda y Recursiva para Súper Resolución (*DRCN*): La Red Profunda y Recursiva para Súper Resolución (*DRCN* por sus siglas en inglés) [24] consta de tres etapas: de extracción de características, de inferencia y de reconstrucción. Utiliza una arquitectura residual:

1. *Etapa de extracción de características f_1.* Se toma la imagen de baja resolución interpolada x y se representa como un conjunto de mapas de características F_0 utilizando dos capas convolucionales:

$$F_{-1} = \max(0, W_{-1} * x + B_{-1}) , \tag{7}$$

$$F_0 = \max(0, W_0 * F_{-1} + B_0) , \tag{8}$$

$$f_1(x) = F_0 . \tag{9}$$

2. *Etapa de inferencia f_2.* Los mapas de características de salida de la red pasan por una sola capa recursiva. Cada recursión aplica la misma convolución seguida de una capa ReLU:

$$F_d = g(F_{d-1}) = max(0, W * F_{d-1} + B) , \tag{10}$$

$$f_2(F) = (g \circ g \circ \cdots \circ) g(F) = g^D(F) . \tag{11}$$

3. *Etapa de reconstrucción f_3.* Transforma los mapas de funciones de salida de la red de inferencia en la imagen de alta resolución:

$$F_{D+1} = max(0, W_{D+1} * F_D + B_{D+1}) , \tag{12}$$

$$\hat{y} = max(0, W_{D+2} * F_{D+1} + B_{D+2}) , \tag{13}$$

$$f_3(F) = \hat{y} . \tag{14}$$

La función de pérdida es una combinación de L1 y L2, además de un término de regularización.

Pérdida L1: *MSE* entre las salidas D y la imagen de HR usando supervisión recursiva:

$$L_1(\theta) = \sum_{d=1}^{D} \sum_{i=1}^{N} \frac{1}{2DN} \left\| y^{(i)} - \hat{y}_d{}^{(i)} \right\|^2 . \quad (15)$$

Pérdida L2: *MSE* entre las salidas D ponderadas y la imagen HR.

$$L_2(\theta) = \sum_{i=1}^{N} \frac{1}{2N} \left\| y^{(i)} - \sum_{d=1}^{D} w_d * \hat{y}_d{}^{(i)} \right\|^2 . \quad (16)$$

Resultando finalmente:

$$L(\theta) = \alpha L_1(\theta) + (1 - \alpha) L_2(\theta) + \beta \|\theta\|^2 , \quad (17)$$

donde θ son los parámetros de la red.

La Fig. 5 muestra la arquitectura general de la red *DRCN*.

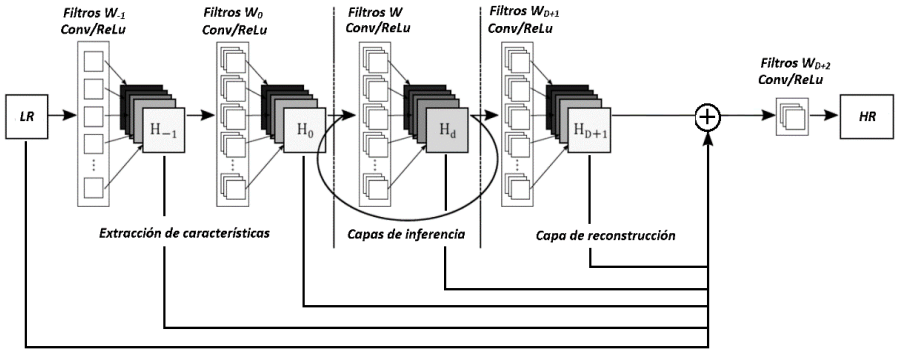


Fig. 5. Red DRCN [23].

3. Estudio comparativo

Los algoritmos utilizados en esta comparación están implementados en Matlab [30] y Pytorch [31], los parámetros de cada método se establecieron de acuerdo con los valores dados en sus reportes correspondientes. Las pruebas se realizaron en una computadora con sistema operativo Windows 10, procesador Intel (R) Core (TM) i7-6700 3.40GHz, con 16 GB de RAM. Para las pruebas, se utilizaron imágenes naturales de bases de datos "Set5" [32], "Set14" [33], "Urban100" [34], y "BSD100"[35, 36] las cuales son ampliamente utilizadas para realizar las pruebas en trabajos de súper resolución [20, 22, 23].

3.1. Métricas

Para comparar el rendimiento de los algoritmos *SISR*, se utilizaron las métricas Relación Señal Ruido ("*Peak-Signal to Noise Ratio*" *PSNR*) e Índice de Similitud Estructural ("*Structural Similarity Index*" *SSIM*), que se calculan entre la imagen súper resuelta y su original. El *PSNR* se calcula a partir del error medio cuadrático (*MSE*),

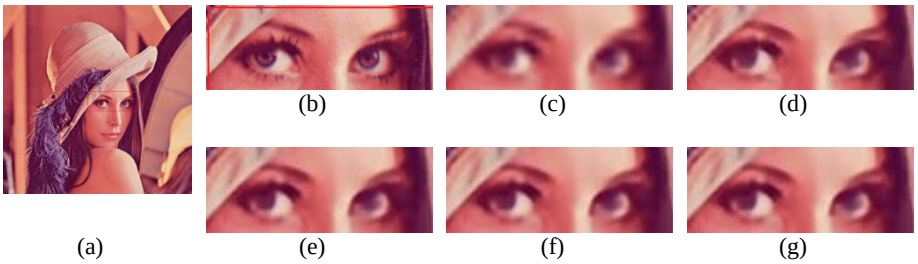


Fig. 6. Comparación visual para una escala de 4x en la imagen de Lena, (a) imagen de alta resolución *HR*, (b) porción de la imagen de *HR*, (c) interpolación bicúbica, (d) método *SRCNN*, (e) método *FSRCNN*, (f) método *VDSR* y (g) método *DRCN*.

que es el error promedio entre la imagen original y la imagen súper resuelta. Dada una imagen súper resuelta $\hat{y}(i, j)$ de dimensiones $m \times n$ y su original $y(i, j)$, *MSE* y *PSNR* se definen como:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [\hat{y}(i,j) - y(i,j)]^2 , \tag{18}$$

$$PSNR = 20\, log_{10} \left(\frac{255}{\sqrt{MSE}} \right). \tag{19}$$

El índice *SSIM* [37] es un modelo basado en la percepción y calcula la similitud entre las imágenes originales y súper resueltas, entre más cercano a 1 es este índice, mayor es el parecido entre las imágenes comparadas. El *SSIM* tiene en cuenta la luminancia, el contraste y los cambios estructurales entre las dos imágenes. El índice *SSIM* se define como:

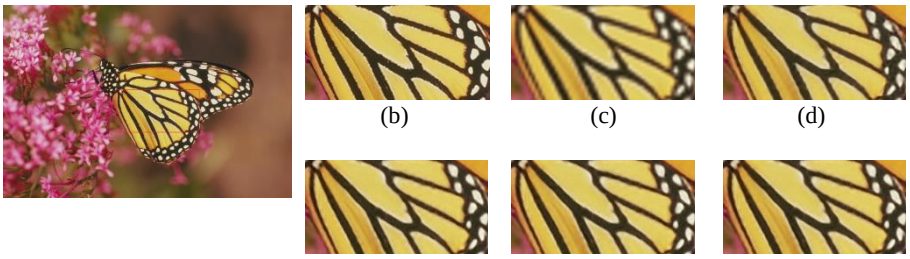
$$SSIM = \frac{(2\mu_y \mu_{\hat{y}} + C_1)(2\sigma_{y\hat{y}} + C_2)}{(\mu_y^2 + \mu_{\hat{y}}^2 + C_1)(\sigma_y^2 + \sigma_{\hat{y}}^2 + C_2)}, \tag{20}$$

donde μ_y , $\mu_{\hat{y}}$ son las medias y σ_y , $\sigma_{\hat{y}}$ son las desviaciones estándar de las imágenes original y súper resuelta, $\sigma_{y\hat{y}}$ es la covarianza entre y e $\hat{y}$. C_1, C_2 son constantes.

3.2. Experimentación

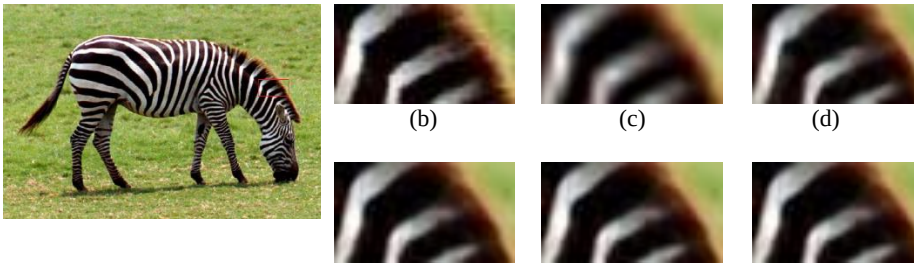
En la experimentación, se utilizó un total de 219 imágenes de las diferentes bases de datos, con el fin de medir el desempeño promedio de *PSNR* y el *SSIM* de cada uno de los algoritmos. Las pruebas se realizaron en las escalas de aumento de 2x, 3x y 4x, reportando para cada escala los resultados promedios del *PSNR* y el *SSIM* en cada una de las bases de datos de imágenes utilizadas.

El algoritmo de interpolación bicúbica se consideró para la comparación, ya que se usa ampliamente como punto de referencia para la comparación en trabajos recientes del estado del arte.



(a) (b) (c) (d) (e) (f) (g)

Fig. 7. Comparación visual para una escala de 4x de la imagen Monarch, (a) imagen de alta resolución *HR*, (b) porción de imagen de *HR*, (c) interpolación bicúbica, (d) método *SRCNN*, (e) método *FSRCNN*, (f) método *VDSR* y (g) método *DRCN*.



(a) (b) (c) (d) (e) (f) (g)

Fig. 8. Comparación visual para una escala de 4x de la imagen Zebra, (a) imagen de alta resolución HR, (b) porción de imagen de alta resolución HR, (c) interpolación bicúbica, (d) método *SRCNN*, (e) método *FSRCNN*, (f) método *VDSR* y (g) método *DRCN*.

Tabla 1. Promedios de *PSNR* y *SSIM* para un factor de escala de 2x.

Método	Tamaño	Bases de datos							
		Set5		Set14		BSD100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicúbico	2x	33.65	0.965	30.20	0.930	29.57	0.928	26.89	0.856
SRCNN		36.52	0.976	32.48	0.933	31.34	0.938	29.56	0.895
FSRCNN		36.98	0.977	32.64	0.932	31.50	0.937	29.83	0.909
VDSR		37.54	0.981	33.05	**0.938**	**31.88**	**0.945**	30.58	0.911
DRCN		**37.62**	**0.981**	**33.08**	0.937	31.84	0.943	**30.64**	**0.915**

Tabla 2. Promedios de *PSNR* y *SSIM* para un factor de escala de 3x.

Método	Tamaño	Bases de datos							
		Set5		Set14		BSD100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicúbico	3x	30.38	0.936	27.57	0.895	27.25	0.853	24.38	0.789
SRCNN		32.59	0.955	29.30	0.911	28.36	0.884	26.29	0.802
FSRCNN		33.10	0.959	29.39	0.911	28.53	0.940	26.41	0.810
VDSR		33.62	0.962	29.68	0.931	28.85	0.940	27.18	0.846
DRCN		**33.80**	**0.962**	**29.74**	**0.932**	**28.85**	**0.942**	**27.20**	**0.851**

Tabla 3. Promedios de *PSNR* y *SSIM* para un factor de escala de 4x.

Método	Tamaño	Bases de datos							
		Set5		Set14		BSD100		Urban100	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicúbico	4x	28.41	0.908	26.02	0.880	25.97	0.834	23.06	0.722
SRCNN		30.38	0.934	27.52	0.896	26.85	0.851	24.48	0.789
FSRCNN		30.70	0.936	27.62	0.896	26.91	0.850	24.60	0.834
VDSR		31.31	0.939	**28.08**	0.901	27.29	0.862	25.14	0.835
DRCN		**31.54**	**0.939**	28.05	**0.911**	**27.35**	**0.868**	**25.25**	**0.847**

Tabla 4. Número de capas y parámetros en cada uno de los métodos.

Método	Número de capas	Número de parámetros
SRCNN	3	57 mil
FSRCNN	8	12 mil
VDSR	20	665 mil
DRCN	5 (recursivas)	1.77 millones

4. Resultados

Los resultados obtenidos en la experimentación se presentan a continuación. Las figuras 6, 7 y 8 muestran los resultados visuales de la comparación en las imágenes de Lena, Monarch y Zebra de la base de datos BSD100, para un factor de escala de 4x. Para una mejor apreciación visual, se muestran sólo porciones de las imágenes mencionadas.

Las tablas 1, 2 y 3 muestran los promedios de *PSNR* y *SSIM* utilizando las bases de datos Set5, Set14, BSD100 y Urban100 para diferentes factores de escala 2x, 3x y 4x.

5. Discusión

En las tablas 1-3 se observa que cualitativamente con el *SSIM* y cuantitativamente con el *PSNR*, el método de interpolación bicúbica proporciona los resultados más modestos. Los métodos muy profundos obtienen mejores resultados en comparación con las arquitecturas poco profundas. Los métodos muy profundos logran, en promedio, mejorar el *PSNR* al compararlos contra la interpolación bicúbica en 2.39 dB, mientras que los métodos poco profundos aventajan a la interpolación bicúbica en 2.17dB.

Para una escala de magnificación de 4x, los métodos profundos aventajan por 2.13 dB a la interpolación bicúbica, mientras que los poco profundos elevan en promedio al *PSNR* en 1.51 dB. Para la escala de 3x, las mejoras son de 2.47 dB para los muy profundos, contra 1.85 dB de los poco profundos y para la escala 2x, los muy profundos logran mejoras de 3.20 dB, contra 2.52 dB de los poco profundos. Podemos ver que a medida en que la escala aumenta, la calidad de las imágenes recuperadas se degrada. La recursividad en la estructura *DRCN* permite que la red *VDSR* sea más compacta,

reduciendo el número de capas y conexiones. Aunque el uso de la recursividad requiere una gran cantidad de memoria *RAM*, lo que hace que el método sea potencialmente inestable.

Los métodos *VDSR* y *DRCN* tienen rendimientos similares, superiores a los métodos poco profundo, la arquitectura *DRCN* es más compacta que *VDSR*. Los menos profundos tienen una menor cantidad de capas como se aprecia en la tabla 4, esto implica una menor complejidad computacional, ya que el número de operaciones y la cantidad de valores a almacenar será menor.

Por lo tanto, son buenos candidatos cuando se requieren métodos computacionalmente livianos, sacrificando calidad de imagen. Estos métodos son los candidatos ideales para aplicaciones de tiempo real, donde se necesitan obtener los resultados en un período relativamente corto de tiempo. Las redes muy profundas son aplicables donde la complejidad no es un factor determinante, por ejemplo, en la súper resolución de imágenes médicas donde la calidad de la imagen, puede ayudar a un mejor diagnóstico.

Finalmente, se puede observar que los valores más altos de *PSNR* no siempre corresponden a una mejor percepción visual de la calidad de imagen, esto indica que esta métrica, a pesar de ser ampliamente utilizada en la literatura, no es suficiente para establecer la calidad absoluta del método de súper resolución. Debido a esto, es conveniente también utilizar la métrica *SSIM*.

6. Conclusiones

Se presentó un estudio comparativo de cuatro algoritmos de súper resolución basados en aprendizaje profundo. Se realizaron mediciones de *PSNR* y *SSIM* de las imágenes en las bases de datos, comúnmente utilizadas en súper resolución, para escalas de magnificación de 2x, 3x y 4x, con el fin de obtener los promedios en cada base de datos. Se muestran comparaciones cuantitativas y cualitativas de cada método. Los resultados demuestran que las arquitecturas más profundas probadas obtienen mejores resultados de *PSNR* y *SSIM*. En general, en los métodos muy profundos *VDSR* y *DRCN* siempre se obtuvieron los mejores resultados.

El uso del aprendizaje residual en las redes muy profundas *VDSR* y *DRCN* permite que el proceso de entrenamiento sea menos complejo, debido a que la cantidad de información que se procesa en la red se reduce y se logra minimizar los efectos del desvanecimiento del gradiente, el cual es un problema que surge en la medida que las redes se hacen más profundas. Los métodos poco profundos tienen más utilidad en aplicaciones que requieran menor complejidad. Si la calidad de las imágenes es lo que interesa, las redes muy profundas pueden valorarse como una opción.

En trabajos posteriores se explorará la influencia de la elección del diseño de la red neuronal, así como la función de pérdida en los rendimientos de estos algoritmos.

Agradecimientos. L. J. Rodríguez agradece a la UACJ por el apoyo brindado y al CONACYT por la beca otorgada para realizar sus estudios doctorales. H. Sossa agradece al Instituto Politécnico Nacional y a la SIP-IPN por el apoyo otorgado a través del proyecto SIP-IPN 20200630 para llevar a cabo esta investigación.

Referencias

1. Singh, A., Singh, J.: Survey on single image based super-resolution implementation challenges and solutions. Multimedia Tools and Applications, 79(3), pp. 1641–1672 (2019)
2. Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J., Liao, Q.: Deep learning for single image super-resolution: a brief review. IEEE Trans. Multimedia, 21(12), pp. 3106–3121 (2019)
3. Li, X., Wu, Y., Zhang, W., Wang, R., Hou, F.: Deep learning methods in real-time image super-resolution: A survey. J. of Real-Time Image Processing, pp. 1–22 (2019)
4. Li, D., Wang, Z.: Video superresolution via motion compensation and deep residual learning. IEEE Trans. on Computational Imaging, 3(4), pp. 749–762 (2017)
5. Huang, Y., Wang, W., Wang L.: Video super-resolution via bidirectional recurrent convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell., 40(4), pp. 1015–1028 (2018)
6. Lei, S., Shi, Z., Zou Z.: Super-resolution for remote sensing images via local–global combined network. IEEE Geosci. Remote Sens. Lett., 14(8), pp. 1243–1247 (2017)
7. Shi, J., Li, Z., Ying, S., Wang, C., Liu, Q., Zhang, Q., Yan, P.: MR image super-resolution via wide residual networks with fixed skip connection. IEEE J. Biomed. Health Inform., 23(3), pp. 1129–1140 (2019)
8. Oktay, O., Ferrante, E., Kamnitsas, K., Heinrich, M., Bai, W., Caballero, J., Cook, S. A., de Marvao, A., Dawes, T., O'Regan, D. P.: Anatomically constrained neural networks (ACNNS): Application to cardiac image enhancement and segmentation. IEEE Trans. Med. Imag, 37(2), pp. 384–395 (2018)
9. Zhou, F., Yang, W., Liao, Q.: Interpolation-based image super-resolution using multisurface fitting. IEEE Trans. Image Process., 21(7), pp. 3312–3318 (2012)
10. Mahmoudzadeh, Pasha, A., Kashou, N.: Interpolation-based super-resolution reconstruction: effects of slice thickness. J. of Medical Imaging 1(3), pp. 1–9 (2014)
11. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution as sparse representation of raw image patches. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, pp. 1–8 (2008)
12. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process., 19(11), pp. 2861–2873 (2010)
13. Timofte, R., De-Smet, V., Van-Gool, L.: A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Cremers D., Reid I., Saito H., Yang MH. (eds) Computer Vision – ACCV. Lecture Notes in Computer Science, 9006, pp. 111–126 (2014)
14. Yang, C. Y., Yang, M. H.: Fast direct super-resolution by simple functions. In: IEEE International Conf. on Computer Vision, Sydney, NSW, pp. 561–568 (2013)
15. Schulter, S., Leistner, C., Bischof, H.: Fast and accurate image upscaling with super-resolution forests. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 3791–3799 (2015)
16. Zhang, K., Gao, X., Tao, D., Li, X.: Single image super resolution with nonlocal means and steering kernel regression. IEEE Trans. Image Process., 21(11), pp. 4544–4556 (2012)
17. Chang, H., Yeung, D. Y., Xiong, Y.: Super-resolution through neighbor embedding. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, pp. 275–282 (2004)
18. Zhang, K., Wang, B., Zuo, W., Zhang, H., Zhang, L.: Joint learning of multiple regressors for single image superresolution. IEEE Signal Process. Lett., 23(1), pp. 102–106 (2016)
19. Zhang, K., Tao, D., Gao, X., Li, X., Xiong, Z.: Learning multiple linear mappings for efficient single image super resolution. IEEE Trans. Image Process., 24(3), pp. 846–861 (2015)
20. Dong, C., Loy, C. C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell., 38(2), pp. 295–307 (2015)

21. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: International Conf. on Computer Vision, pp. 370–378 (2015)
22. Dong, C., Loy, C. C., Tang, X.: Accelerating the superresolution convolutional neural network. In: European Conf. on Computer Vision (ECCV), pp. 1-22 (2016)
23. Kim, J., Lee, J. K., Lee, K. M.: Accurate image superresolution using very deep convolutional networks. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Las Vegas, NV, pp. 1646–1654 (2016)
24. Kim, J., Lee, J. K., Lee, K. M.: Deeply-recursive convolutional network for image super-resolution. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2016)
25. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep Laplacian pyramid networks for fast and accurate super-resolution. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 5835–5843 (2017)
26. Bishop, C. M.: Pattern recognition and machine learning. Springer, New York (2006)
27. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
28. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. on Neural Networks 5(2), pp. 157–166 (1994)
29. Robots: Very deep convolutional networks for large-scale visual recognition. https://www.robots.ox.ac.uk/~vgg/research/very_deep/ (2020)
30. MATLAB: https://www.mathworks.com/products/matlab.html, last accessed 2020/03/20.
31. Pytorch: https://pytorch.org/ (2020)
32. Bevilacqua, M., Roumy, A.; Guillemot, C., M., Alberi-Morel, L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: British Machine Vision Conference BMVC, pp. 1–10 (2012)
33. Zeyde, R., Elad, M., Protter, M.: On single image scaleup using sparse-representations. In: Boissonnat JD. et al. (eds) Curves and Surfaces. Curves and Surfaces 2010, Lecture Notes in Computer Science, 6920, pp. 710–730 (2010)
34. Huang, J. B., Singh, A., Ahuja, N.: Single image superresolution from transformed self-exemplars. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 5197–5206 (2015)
35. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: International Conf. Computer Vision, 2, pp. 416–423 (2001)
36. The Berkeley Segmentation Dataset and Benchmark. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ (2020)
37. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions On Image Processing, 3(4), pp. 600–612 (2004)

ISSN 1870-4069

Detección y clasificación de señales de tráfico mexicanas mediante aprendizaje profundo

Rubén Castruita Rodríguez, Carlos Mendoza Carlos, Osslan Osiris VergaraVillegas, Vianey Guadalupe Cruz Sánchez

Universidad Autónoma de Ciudad Juárez,
Instituto de Ingeniería y Tecnología,
México

{al176515, al176524}@alumnos.uacj.mx,
{overgara, vianey.cruz}@uacj.mx

Resumen. La detección y clasificación automática de señales de tráfico es una tarea para apoyar la seguridad de un conductor e incluso para asistir en la navegación de un automóvil autónomo. El objetivo del artículo es presentar una metodología para la detección y clasificación de señales de tráfico mexicanas mediante aprendizaje profundo. La metodología se divide en cinco etapas: 1) se realiza la recolección de 1284 imágenes de señales de tráfico en un ambiente no controlado, 2) se propone un proceso manual para la detección de señales de tráfico en las imágenes, 3) se entrena una red neuronal convolucional con el conjunto CIFAR-10 para obtener conocimiento amplio de características de diferentes objetos, 4) se utiliza una red neuronal convolucional basada en regiones para detectar las señales, y 5) se realiza un proceso de transferencia de conocimiento y de aumentado de datos para la clasificación con ResNet-50 modificada. De acuerdo con los resultados obtenidos de los experimentos se concluye que la metodología propuesta permite reconocer señales de tráfico mexicanas con una exactitud del 95.33%, lo cual es competitivo con los resultados presentados en la literatura. Además, para demostrar la robustez de la propuesta, se presenta una prueba para clasificar imágenes que no contienen señales de tráfico cuyo resultado de exactitud fue de 99.5%.

Palabras clave: Señales de tráfico, red neuronal convolucional, red neuronal convolucional basada en regiones, detección de regiones.

Detection and Classification of Mexican Traffic Signs Using Deep Learning

Abstract. Automatic detection and classification of traffic signs is a task to support the safety of a driver and even to assist in the navigation of a self-driving car. The goal of this article is to present a methodology for the detection and classification of Mexican traffic signs using deep learning. The methodology comprises five stages: 1) the collection of 1284 images of Mexican traffic signs is carried out in an uncontrolled environment, 2) a manual process for the detection of traffic signs in the images is proposed, 3) a convolutional neural

network with the CIFAR-10 set is trained to obtain a broad knowledge of the characteristics of different objects, 4) a region-based convolutional neural network is used to detect the signals, and 5) a process of knowledge transfer and data augmentation is carried out for signal classification with a modified ResNet-50. According to the results obtained from the experiments, it is concluded that the proposed methodology allows recognizing Mexican traffic signs with an accuracy of 95.33%, which is competitive with the results presented in the literature. Also, to demonstrate the robustness of the proposal, a test to classify images that do not contain traffic signs is presented, obtaining an accuracy of 99.5%.

Keywords: Traffic signs, convolutional neural network, region-based convolutional neural network, region detection.

1. Introducción

Las señales de tráfico son los signos visuales utilizados para ofrecerle información a los conductores y peatones que transitan por un camino [1, 2]. Las señales de tráfico se clasifican de acuerdo con sus formas y colores, de manera que sean llamativas para los automovilistas y se les preste atención [3]. De hecho, los países europeos han realizado trabajos para estandarizar las señales de tráfico, lo que dio lugar a la convención de Viena [4]. Sin embargo, a pesar de los esfuerzos, todavía existen variaciones entre las señales de tráfico utilizadas en las diferentes regiones del mundo.

Particularmente en México, de acuerdo con la Secretaría de Comunicaciones y Transportes (SCT), las señales de tráfico se dividen en dos; las verticales que son construidas con placas e instaladas a través de postes, y las horizontales que son las rayas, palabras, símbolos y objetos, aplicados o adheridos sobre el pavimento [5]. A su vez, las verticales se clasifican en tres tipos que son las restrictivas, preventivas e informativas. Las restrictivas indican prohibiciones reglamentarias que regulan el tránsito en las carreteras y son de color rojo y blanco. Las preventivas advierten a los conductores de la existencia de algún problema o peligro en las carreteras y son de color amarillo y negro. Las informativas les otorgan a los conductores datos de ubicación, nombres o kilometrajes y pueden ser azules, verdes o blancas.

Desafortunadamente, cuando las señales de tráfico se ignoran o no se observan debido a condiciones adversas, como puede ser el clima, mala iluminación u oclusiones parciales, se puede causar un accidente. Por lo que, la posibilidad de construir un sistema automático para la detección y reconocimiento de señales de tráfico podría apoyar a los conductores a tener una conducción más segura cuando están cansados, cuando por descuido las ignoran o cuando no las alcanzan a ver.

Debido a que las señales de tráfico se clasifican por sus formas y colores, se pueden aplicar técnicas de visión artificial para detectarlas y reconocerlas en imágenes o vídeos [6]. Actualmente, dichas técnicas se aplican en automóviles autónomos, vigilancia de tráfico, asistencia a conductores o mantenimiento de las carreteras. Actualmente, hay vehículos que cuentan con sistemas de reconocimiento de señales de tráfico, como BMW Series 5 y 7, Ford Focus y Edge, etcétera [7].

En la literatura, se han presentado diversos trabajos sobre detección y reconocimiento de señales de tráfico [1, 3, 8, 9, 10, 11].

Los trabajos se enfocan principalmente en realizar un análisis de los métodos de detección por la forma o color de las señales [12, 13], y en resolver los retos que conlleva una buena detección (variación de iluminación, distorsiones geométricas, figuras similares a la señal, entre otras) [14, 15, 16], junto con la segmentación [17, 18, 19], reconocimiento [20, 21] y rastreo de señales en secuencias de imágenes [22, 23]. Sin embargo, no se encontraron trabajos enfocados a la detección y clasificación del conjunto de señales de tráfico mexicanas, ni tampoco un trabajo que utilice la fusión de una red neuronal convolucional (CNN, por sus siglas en inglés) y una red neuronal convolucional basada en regiones (R-CNN, por sus siglas en inglés) para resolverlo.

Debido a que todavía hay retos por resolver para diseñar un sistema de detección y reconocimiento completamente exitoso, el presente trabajo tiene como objetivo desarrollar una metodología para la detección y clasificación de señales de tráfico mexicanas, adquiridas en condiciones no controladas, mediante técnicas de aprendizaje profundo que incluyen a la CNN y la R-CNN. Las principales contribuciones del trabajo se resumen a continuación:

- Se presenta un set de datos con señales de tráfico mexicanas, de la ciudad de Monterrey, Nuevo León y Ciudad Juárez, Chihuahua, compuesto de 1028 imágenes RGB que contienen un total de 1126 señales de tráfico.
- Se propone y se prueba una metodología para la detección y clasificación de señales de tráfico mexicanas en ambientes no controlados basada en aprendizaje profundo.

El resto del artículo se encuentra organizado de la siguiente manera: en la sección 2, se describen los materiales y los métodos para el desarrollo de la metodología. La sección 3, describe los experimentos y los resultados obtenidos. Finalmente, la sección 4, discute las conclusiones obtenidas del trabajo.

2. Materiales y métodos

La metodología propuesta para resolver el problema de detección y reconocimiento de señales de tráfico se compone de cinco módulos los cuales se muestran en la Fig. 1, cada módulo se describe a continuación.

Fig. 1. Módulos para la detección y reconocimiento de señales de tráfico mexicanas.

Fig. 2. Señales de tráfico mexicanas. (a) Alto. (b) Ceda el paso. (c) Límite de velocidad. (d) No estacionarse. (e) Vuelta prohibida. (f) Glorieta. (g) Niños jugando. (h) Cruce peatonal. (i) Tope. (j) Estacionamiento para discapacitados. (k) Parada de autobús.

2.1. Recolección de imágenes

Para el módulo de recolección de imágenes se obtuvieron muestras de los tres tipos de señales verticales en dos etapas. En la primera, se capturaron señales de tráfico con la cámara de un celular iPhone 8 y un Samsung S9. Las imágenes se obtuvieron en Ciudad Juárez, Chihuahua, e incluyen avenidas y calles transitadas, parques y áreas recreativas públicas, estacionamientos de escuelas y plazas comerciales, así como fraccionamientos y colonias. Al final, se recolectaron 233 imágenes a color con resolución de 900 x 1200 píxeles (iPhone 8) y 675 x 900 píxeles (Samsung S9). Debido a la poca cantidad de imágenes conseguidas, se procedió a la segunda etapa.

En la segunda etapa, se utilizó la herramienta *Street View* de *Google Maps*. Para la recolección se decidió desplazarse en las calles del mapa de Ciudad Juárez, Chihuahua y Monterrey, Nuevo León. Se escogió Ciudad Juárez ya que es la ciudad donde se desarrolló la metodología, mientras que Monterrey fue seleccionada porque tiene condiciones similares a Juárez, y al ser la tercera ciudad más poblada de México, tiene una gran cantidad de señales de tráfico en sus calles y avenidas.

Tabla 1. Cantidades y tipos de señales de tráfico recolectadas.

Tipo		**Total**
	Señales Restrictivas	
Alto		294
Ceda el paso		71
Límite de velocidad		129
No estacionarse		217
Vuelta prohibida		92
	Señales Preventivas	
Glorieta		59
Niños jugando		91
Cruce peatonal		210
Tope		99
	Señales Informativas	
Estacionamiento para discapacitados		87
Parada de autobús		77
	Total	1426

Fig. 3. Proceso de detección de regiones de interés con su respectiva etiqueta.

Al final, se logró la recolección de 1051 imágenes de diferentes resoluciones en un rango de 147x148 hasta 1512x2016. Por lo que se cuenta con 1284 imágenes que contienen mínimo un señalamiento y máximo cinco, para un total de 1426 señales de tráfico. Es importante destacar que las señales se adquirieron en diferentes perspectivas, tamaños e iluminaciones, como se muestra en la Fig. 2. En la Tabla 1 se muestra el número de señales obtenidas de cada tipo.

2.2. Detección de regiones de interés

Para cada una de las imágenes recolectadas se realizó un proceso manual de detección de regiones de interés (señales de tráfico). Para comenzar, se definieron las etiquetas para representar cada una de las once clases de interés que corresponden a la primera columna de la Tabla 1. Posteriormente, a cada imagen se le insertó un

1028x12 table

	1 imageFilename	2 Alto	3 Peaton	4 NoEstacionarse	5 CedaPaso	6 ParadaCamion	7 Glorieta	8 LimVelocidad	9 EDiscapacitados	10 NinosJugando	11 Tope	12 VueltaProhibida
486	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0486.jpg'	[307,243.07...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
487	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0487.jpg'	[]	[]	[174,97.5309,63.76...	[]	[]	[]	[]	[]	[]	[]	[]
488	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0488.jpg'	[]	[326,156.0140,...	[]	[]	[]	[]	[]	[]	[]	[]	[]
489	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0489.jpg'	[]	[]	[119.0843,130.3146...	[]	[]	[]	[]	[]	[]	[]	[]
490	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0490.jpg'	[]	[753,357.1517,...	[]	[]	[]	[]	[]	[]	[]	[]	[]
491	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0491.jpg'	[497.9775,1...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
492	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0492.jpg'	[]	[]	[]	[]	[169,136.4775,123....	[]	[]	[]	[]	[]	[]
493	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0493.jpg'	[1.1995e+03...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
494	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0494.jpg'	[147,236.73...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
495	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0495.jpg'	[]	[]	[270.2097,117,79,97]	[]	[]	[]	[]	[]	[]	[]	[]
496	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0496.jpg'	[]	[]	[]	[]	[236,296,658...	[]	[]	[]	[]	[]	[212,178,77,124]
497	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0497.jpg'	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[216.0449,136.3876,...
498	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0498.jpg'	[445,117.30...	[1.1913e+03,11....	[]	[]	[99,122,49,66]	[]	[]	[]	[]	[]	[340,124,57,45]
499	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0499.jpg'	[481.2022,9...	[]	[]	[]	[]	[]	[]	[]	[]	[]	[362.99,61,59]
500	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0500.jpg'	[]	[]	[]	[]	[204,151.8717,103....	[]	[]	[200.9270,302.1826,...	[]	[]	[]
501	'C:\Users\Carlos Mendoza\Documents\TrafficSigns\img0501.jpg'	[]	[]	[]	[]	[]	[]	[785.6217,540,11...	[]	[]	[]	[]

Fig. 4. Datos obtenidos en la detección de regiones de interés.

rectángulo delimitador para cada una de las señales de tráfico identificadas. A cada clase se le asignó un color diferente de rectángulo como se muestra en la Fig. 3.

Tras haber etiquetado las 1426 señales se creó una tabla con doce columnas, en la primera se guarda la trayectoria de almacenamiento local de la imagen, y en las columnas de la dos a la doce se almacena un vector de 1x4 con un formato [x, y, ancho, alto], que especifica las coordenadas de la ubicación de la esquina superior izquierda y el tamaño del rectángulo delimitador de la región de interés correspondiente. En caso de que no exista la etiqueta de una señal de tráfico en la imagen, se representa como una celda vacía. En la Fig. 4 se muestra un fragmento de los datos obtenidos.

El conjunto de imágenes fue dividido en 80% (1028 imágenes) para entrenamiento y 20% (256 imágenes) para validación. Posteriormente, se utilizaron los datos de la tabla para obtener de cada imagen una copia individual de las señales de tráfico presentes, por lo tanto, se extrajeron 1126 señales para entrenamiento y 300 para validación. Las imágenes se redimensionaron a un tamaño de 224x224 píxeles, debido a que es el tamaño de las entradas de la red ResNet-50 que se utilizará posteriormente.

2.3. Creación de una CNN con CIFAR-10

Para comenzar la clasificación de señales de tráfico se decidió utilizar una CNN, que se compone principalmente de dos partes: en la primera, se utilizan operaciones de convolución y de pooling para generar características profundas de los datos sin procesar; en la segunda, el perceptrón multicapa utiliza las características para asignar una clase a los datos de entrada [24].

Para realizar el entrenamiento de una CNN desde cero se requiere una gran cantidad de imágenes, debido a que solamente así se podrán aprender las diferentes características que representan a un objeto. Además, para implementar la red se necesita una cantidad considerable de recursos computacionales. Debido a que no se contaba con un conjunto de imágenes amplio, se optó por entrenar una CNN con CIFAR-10 [25], que contiene un conjunto de datos amplio, para posteriormente realizar una transferencia del aprendizaje obtenido.

La transferencia de aprendizaje consiste en utilizar el conocimiento de una red previamente entrenada que será aplicado para realizar una nueva tarea. Una ventaja de la transferencia de aprendizaje es que requiere una menor cantidad de datos y recursos computacionales para el entrenamiento [26].

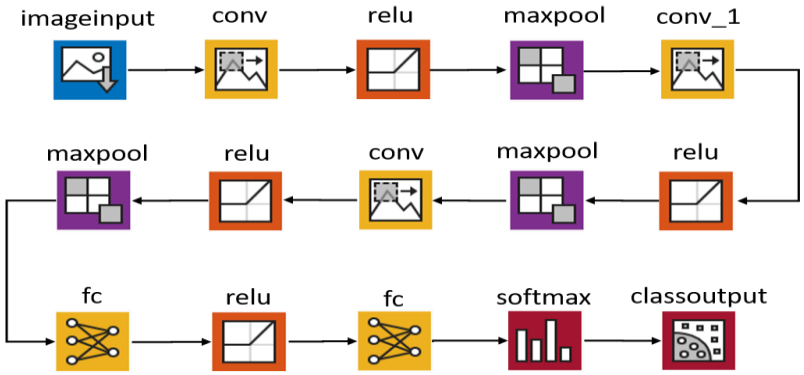


Fig. 5. Arquitectura de CNN para entrenamiento con CIFAR-10.

Se descargó CIFAR-10 con 50,000 imágenes RGB de 32x32 píxeles que incluye aviones, automóviles, pájaros, gatos, venados, perros, ranas, caballos, barcos y camiones. Aun cuando no se incluyen señales de tráfico, es de gran utilidad ya que le permitirá a la CNN aprender conceptos importantes y generalizables para identificar esquinas, bordes, texturas, formas geométricas, entre otros. Para el entrenamiento, y de acuerdo con la experiencia, se diseñó una arquitectura de CNN con 15 capas con una capa de entrada, varias capas de *Convolución*, *ReLU*, *Max Pooling y totalmente conectadas (fully connected)*, una S*oftmax* y una de salida (ver Fig. 5). Además, se utilizó el algoritmo del Descenso del Gradiente Estocástico con Momento (SGDM), porque reduce la oscilación que tiene el SGD en la función de pérdida de la red [15]. En la primera columna de la Tabla 2 se muestra un resumen de los valores establecidos para cada parámetro de la CNN. Una vez que se terminó el entrenamiento, se calculó la exactitud alcanzada por la CNN, que fue de 74.56%. Es importante aclarar que no es necesario alcanzar un 100%, sino la gran cantidad de aprendizaje obtenido con un conjunto de datos tan variado como CIFAR-10.

2.4. Entrenamiento de una R-CNN

El aprendizaje obtenido en el entrenamiento de la CNN con CIFAR-10 fue transferido para entrenar una R-CNN que pudiera aprender a detectar señales de tráfico. Una R-CNN se compone de dos etapas: la primera identifica un subconjunto de regiones en una imagen que puede contener un objeto (señal de tráfico), lo que ayuda a reducir el costo computacional al no buscar en la imagen completa, la segunda realiza la clasificación del objeto identificado [27].

Para el entrenamiento se necesitaron tres entradas: a) la tabla de datos obtenida en la subsección 2.2, b) los valores de los parámetros mostrados en la segunda columna de la Tabla 2 y c) la CNN de la subsección 2.3. Durante el entrenamiento, los pesos de la CNN entrenada con CIFAR-10 son ajustados tomando en cuenta los datos de la Fig. 4. Después, se seleccionan las porciones candidatas a ser señales de tráfico y se etiquetan como positivas o negativas. Las positivas (contienen señales) son aquellas que se superponen con los cuadros generados en la Fig. 4 en un rango de 0.5 a 1, medido por la intersección del cuadro delimitador (coordenadas espaciales) sobre la métrica de

unión. Mientras que las negativas (no contienen señales) son aquellas que se superponen en un rango de 0 a 0.3. Los valores de los rangos fueron seleccionados después de varias pruebas y también se utilizó el SGDM.

La estructura de la R-CNN es idéntica a la que se muestra en la Fig. 5. A la salida de la R-CNN se obtiene el o los objetos detectados enmarcados por un cuadro delimitador (bounding box), la confiabilidad de detección, y la etiqueta de la clase para cada objeto. La confiabilidad genera valores entre cero y uno, y mientras más cerca se encuentre el valor de uno, más confianza hay de que se identificó correctamente la señal de tráfico.

2.5. Creación de una CNN adicional usando como base ResNet-50

Para validar que las señales que clasifica la R-CNN sean las correctas (que detecte una señal de alto y que realmente sea un alto), se aplicó la técnica de transferencia de aprendizaje a la CNN ResNet-50. ResNet-50 es una red residual de 50 capas que fue entrenada con más de un millón de imágenes obtenidas del sitio ImageNet y puede clasificar imágenes en 1000 categorías, tales como teclados, ratones, plumas y muchos animales [28]. Para el entrenamiento, se requirió del repositorio de imágenes de entrenamiento que se creó en la subsección 2.2, el cual contiene 1126 imágenes de señales de tráfico de 224x224 píxeles. Las últimas tres capas de ResNet-50 (Fig. 6(a)), contienen información de cómo combinar las características de las imágenes obtenidas en las capas de convolución para obtener la probabilidad de pertenencia a una clase, el valor de pérdida, y la etiqueta calculada. Para adaptar ResNet-50 al problema de clasificación de señales, se realizó la transferencia de aprendizaje mediante el cambio de las tres últimas capas (ver Fig. 6(b)). Al final, se clasifican las 11 categorías de señales de tráfico, y no las 1000 categorías originales.

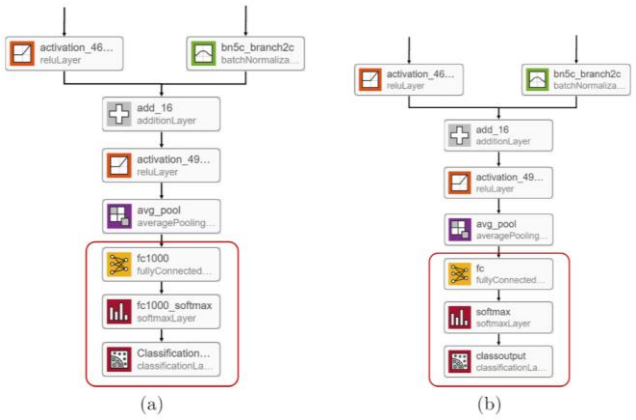


Fig. 6. Cambios realizados a la red ResNet-50. (a) Capas finales originales deResNet-50, (b) Capas finales modificadas a ResNet-50.

Para acelerar el entrenamiento, se congelaron las primeras diez capas al asignarles una tasa de aprendizaje de 0. Adicionalmente, y con el objetivo de evitar un sobreajuste de la red debido a la baja cantidad de imágenes, se realizó un proceso de aumentado de datos.

Tabla 2. Resumen de parámetros y valores utilizados para las redes.

Parámetros	CNN Valor	R-CNN Valor	ResNet-50 Valor
Momentum	0.9	0.9	0.9
InitialLearnRate	0.001	0.001	0.003
LearnRateSchedule	Piecewise	Piecewise	
LearnRateDropFactor	0.1	0.1	
LearnRateDropPeriod	8	100	100
L2Regularization	0.004	0.0001	0.0001
MaxEpochs	40	200	6
MiniBatchSize	128	128	10
Verbose	True	True	False
PositiveOverlapRange		0.5-1	
NegativaOverlapRange		0-0.3	
Shuffle	Once	Once	Every epoch
Validation Data			20%

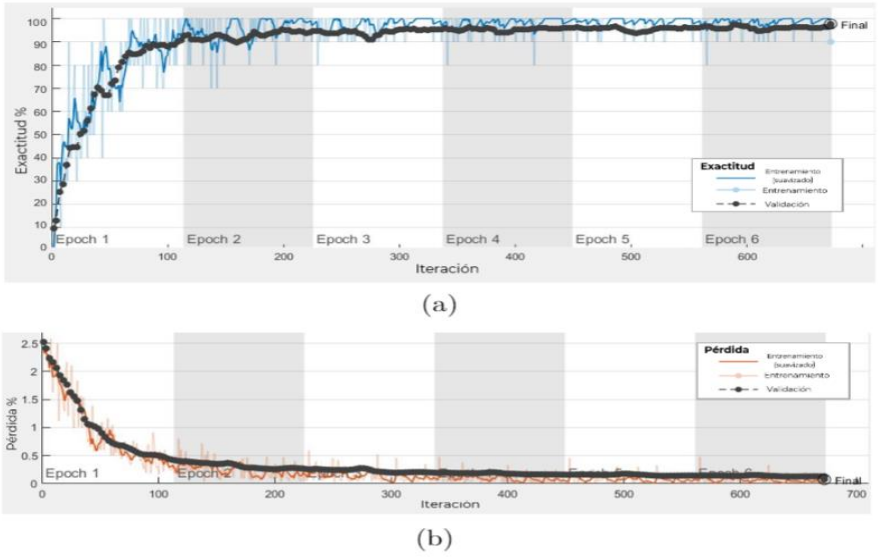


Fig. 7. Gráfica de (a) exactitud y (b) pérdida en el entrenamiento de la CNN.

El aumentado de datos es un proceso artificial para expandir el conjunto original de imágenes de entrenamiento por medio de la aplicación de operaciones de rotación, desplazamiento, y escalamiento, entre otras [29]. Las señales de tráfico fueron rotadas hasta 30 píxeles, y además se obtuvieron espejos verticales y horizontales. La ResNet-50 modificada se entrenó con los valores mostrados en la tercera columna de la Tabla

Tabla 3. Resultados obtenidos con ResNet-50 modificada.

Tipo	Total de señales	Coincidencia positiva	Coincidencia negativa	Confianza media	Exactitud
Señales Restrictivas					
Alto	56	56	0	96.73%	100%
Ceda el paso	13	13	0	99.86%	100%
Límite de velocidad	18	18	0	99.94%	100%
No estacionarse	51	51	0	95.68	100%
Vuelta prohibida	28	25	3	88.72%	89.29%
Señales Preventivas					
Glorieta	7	7	0	81.19%	100%
Niños jugando	14	13	1	93.62%	92.86%
Cruce peatonal	49	48	1	98.41%	97.96%
Tope	21	20	1	99.60%	95.24%
Señales Informativas					
Estac. discap.	29	28	1	89.78%	96.55%
Parada de autobús	14	14	0	97.08%	100%
Total	300	293	7	94.60%	97.67%

2. La gráfica de la exactitud alcanzada durante el entrenamiento se observa en la Fig. 7(a), mientras que la de la pérdida se muestra en la Fig. 7(b).

El entrenamiento se realizó en una computadora HP 15-bs1xx con procesador Intel Core i5 de octava generación y RAM de 8 GB; tomó 288 minutos y 54 segundos y se obtuvo una exactitud del 97.67% tras 672 iteraciones en 6 epochs.

3. Experimentación y resultados

Los experimentos se realizaron para verificar el desempeño de las redes creadas en la sección 2.4 y 2.5. Para la evaluación se utilizaron las medidas de exactitud y confianza media que se muestran en las ecuaciones 1 y 2, respectivamente:

$$\text{Exactitud} = \frac{\text{Coincidencias positivas}}{\text{Núm. de señales de tráfico}}, \tag{1}$$

$$\text{Confianza media} = \frac{\sum \text{Confianza}}{\text{Núm. de señales de tráfico}}. \tag{2}$$

Primero, se analizó la ResNet-50 modificada que se encarga de clasificar la señal de tráfico detectada por la R-CNN. Los resultados de la clasificación de las imágenes del set de validación se muestran en la Tabla 3. Como puede observarse, se obtiene una confianza media total de 94.60% y una exactitud total de 97.67%, los cuáles son valores competitivos con los que se han mostrado en la literatura, por ejemplo, en [3] se menciona una exactitud de 99.75% con una CNN. En la Fig. 8 se muestran ejemplos de la clasificación obtenida con su nivel de confianza correspondiente.

Fig. 8. Resultados de clasificación de señales de tráfico con ResNet-50 modificada.

Tabla 4. Resultados obtenidos con las redes R-CNN y ResNet-50 modificada.

Tipo	Total de señales	Detecciones/ Clasificaciones positivas	Detecciones/ Clasificaciones negativas
Señales Restrictivas			
Alto	56	51	5
Ceda el paso	13	13	0
Límite de velocidad	18	18	0
No estacionarse	51	51	0
Vuelta prohibida	28	26	2
Señales Preventivas			
Glorieta	7	6	1
Niños jugando	14	10	4
Cruce peatonal	49	47	2
Tope	21	21	0
Señales Informativas			
Estac. discap.	29	29	0
Parada de autobús	14	14	0
Total	300	286	14

Posteriormente, se evaluó de forma integral el proceso de detección realizado por la R-CNN y el de clasificación realizado por ResNet-50 modificada.

Fig. 9. Resultados obtenidos en la evaluación de R-CNN y ResNet-50 modificada.

Tabla 5. Resultados obtenidos con las imágenes sin señales de tráfico.

Tipo	Total de señales	Detecciones/ Clasificaciones positivas	Detecciones/ Clasificaciones negativas
Señales Restrictivas			
Alto	200	196	4
Ceda el paso	200	200	0
Límite de velocidad	200	198	2
No estacionarse	200	200	0
Vuelta prohibida	200	199	1
Señales Preventivas			
Glorieta	200	200	0
Niños jugando	200	200	0
Cruce peatonal	200	199	1
Tope	200	200	0
Señales Informativas			
Estac. discap.	200	200	0
Parada de autobús	200	197	3
Total	2200	2189	11

Para realizar la prueba se utilizó los datos de validación que fueran previamente separados de los datos de entrenamiento en la subsección 2.2. Los resultados de la evaluación se muestran en la Tabla 4.

En la Tabla 4, los valores de detección/clasificación negativa incluyen los casos en los que sí hay una señal de tráfico, pero no es detectada y clasificada. Por otra parte, los valores de detección positiva son los casos en los que la señal de tráfico sí es detectada y es clasificada correctamente. Por lo tanto, la exactitud final del sistema es de 95.33%. En la Fig. 9 se muestran cuatro ejemplos de la detección y clasificación de las señales de tráfico con el nivel de confianza correspondiente.

Finalmente, para verificar la robustez de la propuesta, se descargó de la app *Street View* de Google un nuevo conjunto con 62058 imágenes de avenidas, autopistas y centros comerciales ubicados en zonas céntricas de Pittsburgh, Orlando y Manhattan. Sin embargo, lo más importante es que las imágenes no contienen señales de tráfico. De dicho conjunto se seleccionaron aleatoriamente 2200 imágenes, 200 para probar cada una de las 11 clases. El objetivo de la prueba fue identificar los casos en los que se realizan detecciones falsas o un posible comportamiento inesperado en el uso de las redes R-CNN y ResNet-50 modificada. Los resultados obtenidos de la prueba se muestran en la Tabla 5.

Los valores de verdadero negativo hacen referencia al caso en que el sistema infirió que no existen señales de tráfico en la imagen y realmente no se encuentran. Mientras que los valores de falso negativo hacen referencia al caso donde el sistema infirió la existencia de alguna señal de tráfico en la imagen cuando realmente no se encuentra. Como puede observarse en la Tabla 5, la exactitud total de la prueba es de 99.5%, es decir, se cometieron 11 fallos en las 2200 imágenes.

3.1. Discusión

Los resultados obtenidos con la metodología propuesta son competitivos dado que permiten detectar y clasificar un subconjunto de señales de tráfico mexicanas divididas en 11 diferentes clases. Para el caso de la detección y clasificación de señales se observa que la de alto es en la que más se cometen errores, debido a su similitud en las formas y colores con la señal de vuelta prohibida.

También, se detectaron problemas para la señal de glorieta, seguida por la de no estacionarse y tope. Además, es importante resaltar que para las dos señales informativas no se obtuvieron errores.

Por otro lado, para el caso de imágenes que no contienen señales de tráfico, es también la señal de alto en la que más errores se observaron (cuatro), seguida de la señal de parada de autobús y de límite de velocidad. En la clase de señales preventivas es donde se obtienen más éxitos con solamente una equivocación.

4. Conclusiones

En el artículo se presentó una metodología para la detección y clasificación de señales de tráfico mexicanas que utiliza dos CNN y una R-CNN. Las pruebas se realizaron en el conjunto de validación que incluyó 256 imágenes que contienen un total de 300 señales de tráfico mexicanas.

Como resultado de los experimentos se observó que la propuesta obtiene una exactitud de 95.33% para la detección y clasificación de señales, y que además suele no confundirse cuando se le presentan imágenes que no contienen señales de tráfico con una exactitud del 99.5%.

En el futuro, se pretende ampliar el tamaño del conjunto de imágenes y realizar el entrenamiento con unas arquitecturas o detectores distintos con el fin de obtener mejores resultados en una menor cantidad de tiempo. Además, se considera la propuesta de utilizar bases de datos de señales de tráfico internacionales, para poder construir un modelo que pueda en cierta medida generalizar.

Referencias

1. Wali, S., Abdullah, M., Hannan, M., Hussain, A., Samad, S., Ker, P., Mansor, M.: Vision-based traffic sign detection and recognition systems: current trends and challenges. Sensors, 19, pp. 1–28 (2019)
2. Xu, H., Srivastava G.: Automatic recognition algorithm of traffic signs based on convolution neural network. Multimedia Tools and Applications, 79, pp. 11551–11565 (2020)
3. Cao, J., Song, C., Peng, S., Xiao, F., Song, S.: Improved traffic sign detection and recognition algorithm for intelligent vehicles. Sensors, 19, pp. 1–21 (2019)
4. Economic Commission for Europe: Convention on traffic signs and signals. Vienna Convention (1968)
5. Secretaria de Comunicaciones y Transportes: Tipos y Significado de las Señales de Tránsito. http://www.sct.gob.mx/carreteras/direccion-general-de-conservacion-de-carreteras/ publicaciones/senalamiento/ (2019)
6. Jin Y., Yusheng F., Wang W., Guo J., Ren C., Xiang X.: Multi-feature fusion and enhancement single shot detector for traffic sign recognition. IEEE, 8, pp. 38931–38940 (2020)
7. Swathi, M., Suresh, K.: Automatic traffic sign detection and recognition: a review. In: 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), pp. 1–6, Chennai (2017)
8. Gudigar, A., Chokkadi, S., Raghavendra, U.: A review on automatic detection and recognition of traffic sign. Multimedia Tools and Applications, 75, pp. 333–364 (2016)
9. Yang, Y., Luo, H., Xu, H., Wu, F.: Towards real-time traffic sign detection and classification. IEEE Transactions on Intelligent Transportation Systems, 17(7), pp. 2022–2031 (2016)
10. De Oliveira, G., da Silva, F., Pereira, D., de Almeida, L., Artero, A., Bonora, A., de Albuquerque, V.: Automatic detection and recognition of text-based traffic signs from images. IEEE Latin America Transactions, 16(12), pp. 2947–2953 (2018)
11. Alghmgham, D., Latif, G., Alghazo, J., Alzubaidi, L.: Autonomous traffic sign (ATSR) detection and recognition using deep CNN. Procedia Computer Science, 163, pp. 266–274 (2019)
12. Yuan, X., Hao, X., Chen, H., Wei, X.: Robust traffic sign recognition based on color global and local oriented edge magnitude patterns. IEEE Transactions on Intelligent Transportation Systems, 15(4), pp. 1466–1477 (2014)
13. Serna, C., Ruichek, Y.: Classification of traffic signs: the European dataset. IEEE, 6(5), pp. 78136–78148 (2018)
14. Liu, C., Chang, F., Chen, Z., Liu, D.: Fast traffic sign recognition via high-contrast region extraction and extended sparse representation. IEEE Transactions on Intelligent Transportation Systems, 17(1), pp. 79–92 (2016)
15. Lee, H., Kim, K.: Simultaneous traffic sign detection and boundary estimation using convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, 19(5), pp. 1652–1663 (2018)
16. Mannan, A., Javed, K., Rehman, A., Babri, H., Noon, S.: Classification of degraded traffic signs using flexible mixture model and transfer learning. IEEE, 7, pp. 148800–148813 (2019)
17. Zhu, Y., Liao, M., Yang, M., Liu, W.: cascaded segmentation-detection networks for text-based traffic sign detection. IEEE Transactions on Intelligent Transportation Systems, 19(1), pp. 209–219 (2018)
18. García, A., Alvarez, J., Soria-Morillo, L.: Deep neural network for traffic sign recognition systems: an analysis of spatial transformers and stochastic optimisation methods. Neural Networks, 99, pp. 158–165 (2018)

19. Liu, C., Li, S., Chang, F., Wang, Y.: Machine vision based traffic sign detection methods: review, analyses and perspectives. IEEE, 7, pp. 86578–86596 (2019)
20. Greenhalgh, J., Mirmehdi, M.: Recognizing text-based traffic signs. IEEE Transactions on Intelligent Transportation Systems, 16(3), pp. 1360–1369 (2015)
21. Luo, H., Yang, Y., Tong, B., Wu, F., Fan, B.: Traffic sign recognition using a multi-task convolutional neural network. IEEE Transactions on Intelligent Transportation Systems, 19(4), pp. 1100–1111 (2018)
22. Yuan, Y., Xiong, Z., Wang, Q.: An incremental framework for video-based traffic sign detection, tracking, and recognition. IEEE Transactions on Intelligent Transportation Systems, 18(7), pp. 1918–1929 (2017)
23. Li D., Jiang H.: On feature selection in network flow based traffic sign tracking models. Computers & Industrial Engineering, 127, pp. 657–664 (2019)
24. Guangle Y., Tao L., Jiandan Z.: A review of convolutional-neural-network-based action recognition. Pattern Recognition Letters, 118, pp. 14–22 (2019)
25. Krizhevsky, A., and G. Hinton.: Learning multiple layers of features from tiny images. Master's Thesis, University of Toronto (2009)
26. Weiss K., Khoshgoftaar T., Wang D.: A survey of transfer learning. Journal of Big Data, 3, pp. 31–40 (2016)
27. Zhang S., Wu R., Xu K., Wang J., Sun W.: R-CNN-based ship detection from high resolution remote sensing imagery. Remote Sensing, 11, pp. 1–15 (2019)
28. Wen L., Li X., Gao L.: A transfer convolutional neural network for fault diagnosis based on ResNet-50. Neural Computing and Applications, 32, pp. 6111–6124 (2019)
29. Shorten C., Khoshgoftaar, T.: A survey on image data augmentation for deep learning. Journal of Big Data, 6, pp. 1–48 (2019)

ISSN 1870-4069

Análisis de la incidencia delictiva del fuero común mediante la red social Twitter

Rosa G. Hernández Cuacua, Sandra D. Rosas Ortega,
M. Beatriz Bernábe Loranca

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{gabyrocklml1, daana.orteeg,
beatriz.bernabe}@gmail.com

Resumen. En este trabajo se ha hecho uso de técnicas de Procesamiento de Lenguaje Natural PLN, para describir la situación actual en México respecto a la inseguridad (incidencia delictiva del fuero común). El estudio inicia con descubrir opiniones de usuarios en la red social Twitter. Para este propósito, se parte de extraer información aleatoria usando palabras representativas del problema etiquetándolas como hashtag. El desarrollo del trabajo se centra en la creación de un diccionario que permita contabilizar las palabras clave de incidencia delictiva para proceder a representar gráficamente la información semántica de las perspectivas de los asistentes a Twitter sobre delincuencia generalizada. Una vez concentrada la información que ha sido depurada y estructurada, se procesa el conjunto de información con un análisis de sentimientos. A partir de este punto, ha sido posible deducir mecanismos de reacción frente a un problema de violencia. La respuesta denominada reacción tiene al menos 3 interpretaciones en distintos niveles: miedo, terror o pánico.

Palabras clave: Análisis de sentimientos, inseguridad, procesamiento del lenguaje natural, redes sociales.

Analysis of Crime Incidence of Civil Courts through the Social Network Twitter

Abstract. In this work, Natural Language Processing techniques have been used and applied to describe the current situation in México regarding insecurity (crime incidence of civil courts). The study begins discovering the user's reviews on the social network Twitter. For this purpose, random information is extracted using representative words of the problem by tagging them as hashtag. The development of this work is focused in the creation of a dictionary, which allows counting the key words of crime incidence to proceed with representing graphically the semantic information of the different user's perspectives about the generalized delinquency. Once that the information (that has been "cleaned" and structured) is concentrated, it is processed with Sentiment Analysis. From this point, it is possible to deduce mechanisms of reaction face a violence

problem. The so-called reaction response has at least three interpretations in different levels: fear, horror or panic.

Keywords: Sentiment analysis, insecurity, natural language processing, social network.

1. Introducción

La inseguridad social es un tema muy importante en diversos países en vías de desarrollo, lo cual se traduce en un reto a resolver. Tan sólo en México la tasa de inseguridad es de las más altas reflejándose en una de las mayores preocupaciones de los habitantes de cada estado. Se asume que en este problema destaca la angustia que impacta en la integridad física, emocional y la economía.

Existen dos tipos de delincuencia o actos delictivos que son relevantes en estudios de esta magnitud, sin embargo, en este este artículo se aborda la delincuencia menor o común. Esta delincuencia afecta a la mayoría de la población donde no importa la edad, nivel de educación y/o nivel socioeconómico. Esta praxis delictiva da lugar a otras complicaciones dado que sucede diariamente con alto riesgo e impacta en las actividades constantes del riesgo implícito de ser atacado.

Actualmente los actos delictivos suelen presentarse con mayor frecuencia y afectan a casi todos los grupos sociales. En México, estos datos se reportan y registran a los medios de atención a víctimas y consecuentemente, distintas instituciones u organismos gubernamentales organizan esas denuncias con el fin de describir estadísticamente los denominados “focos rojos” de asalto y establecer programas de atención policíaca o de vigilancia.

Desde el enfoque computacional, este tipo de información también puede ser obtenida por otras herramientas de extracción o descubrimiento de información. En este punto, dentro de la Inteligencia Artificial, se encuentra el Procesamiento del Lenguaje Natural (PLN), el cual es conveniente para analizar problemas de tal magnitud. Para el caso que se expone en este artículo, se han identificado las posibles reacciones del problema relacionado con la delincuencia en la red social Twitter. Con el propósito de obtener información asociada con la delincuencia menor, se desarrolló un programa en Python para descargar tweets que identifican palabras y hashtags sobre el problema bajo estudio. Una vez extraída la información, esta es procesada para la creación de diccionarios y análisis de sentimientos.

El trabajo se encuentra estructurado como sigue; la presente introducción como sección 1. La sección 2 se encarga de exponer el marco teórico con los conceptos más relevantes en los últimos cinco años, los antecedentes y estado del arte. En la sección 3 se describen las estrategias para alcanzar el diseño de un algoritmo capaz de soportar descargas de gran tamaño de tweets para crear diccionarios. Esta tarea ha sido posible por la aplicación de un programa desarrollado en Python para PLN. La sección 4 trata el análisis de sentimientos. Y finalmente se presentan las conclusiones y el trabajo futuro.

1.1. Objetivo

El propósito del presente trabajo consiste en recolectar y analizar información sobre la inseguridad en diferentes estados de la república. Este trabajo responde a la necesidad de analizar y atender el problema polarizado de actos delictivos los cuales son cometidos diariamente y que se ha incrementado considerablemente. De esta manera es importante contar con información correcta e insesgada para sensibilizar a los ciudadanos sobre la situación actual que sufre la población en cuanto a delincuencia.

Se entiende que información correcta en este problema, es la generada por el uso de herramientas de PLN en la red social Twitter para obtener datos adecuados sobre la delincuencia. Una vez concentrados datos e información en diccionarios que revelen el problema, es de utilidad su representación en estadística descriptiva para ser comprendida por una comunidad considerable. Posteriormente un análisis de sentimientos se hace necesario para determinar el grado de ansiedad y fragilidad de las personas sobre la delincuencia.

2. Marco teórico

La inteligencia Artificial (IA) es un área reciente que surge alrededor de hace 60 años. Definir un único concepto de Inteligencia Artificial, la cual ha ido madurando a lo largo de los años, no es simple. Sin embargo, en un contexto informal, la IA se expresa como la inteligencia basada en el nivel de conocimientos individuales, y en la habilidad de manipular y reformular apropiadamente los conocimientos de acuerdo con los datos que se proporcionan como requerimientos para resolver algún problema [1]. En términos prácticos se dice que la IA es un campo de estudio que busca explicar y emular el comportamiento inteligente con recursos computacionales [2]. Desde la perspectiva de la informática, la IA trata del diseño de computadoras inteligentes que exhiban las características que se asocian con la inteligencia en el comportamiento humano. Para fines pragmáticos, la IA es una rama de las ciencias computacionales encargada del estudio de la inteligencia con elementos artificiales para llegar a diseñar o plantear sistemas que puedan ser capaces de realizar tareas que hasta hace años sólo la inteligencia humana era capaz de construir manualmente [3].

Dentro de la IA destaca el Procesamiento del Lenguaje Natural (PLN) y se entiende como el uso de los lenguajes naturales para la comunicación de los individuos, ya sea de manera oral, escrita o signada. En este escenario donde el presenta artículo se ubica en la extracción de información a través de los lenguajes naturales, explorando instrumentos computacionales que ayuden a describir el problema de incidencia delictiva desde el enfoque de diccionarios y sentimientos en Twitter.

El análisis de sentimientos en textos es la identificación y extracción de información subjetiva; también llamado "minería de opiniones". Este proceso generalmente involucra el uso de herramientas de PLN y software de análisis de textos para la automatización de estos [4]. El PLN también incluye tratamientos diversos de las redes sociales en donde han tenido gran impacto.

Las redes sociales son lugares en Internet donde las personas publican y comparten todo tipo de información, personal y profesional con terceras personas, conocidos y absolutos desconocidos [5].

En particular, Twitter es una herramienta que comunica principalmente por medio de mensajes cortos, llamados "tweets" que son acompañados de palabras clave, "hashtags" y permiten la agrupación de la información, generalmente suelen tener noticias, opiniones y estados de ánimo. El acceso y estructura de Twitter es ampliamente utilizado dentro del PLN debido a la gran información subyacente que se puede encontrar o rescatar de cada uno de los usuarios que las conforman y de los diversos enfoques que se le puede dar para resolver problemas de carácter social. Dentro de los lenguajes casados con PLN que permiten manipular información destaca Python [6], el cual cuenta con estructuras de datos eficientes y de alto nivel con un enfoque simple pero efectivo a la programación orientada a objetos.

2.1. Estado del arte

En esta sección, se describe una lista de trabajos relacionados con el presente artículo, que en términos amplios representan aplicaciones PLN. Kulkarni Sagar expone una metodología basada en dos aspectos: 1) la detección de publicaciones maliciosas de forma aislada y sin información previa del usuario y 2) la aplicación de un análisis estadístico de lenguaje para detectar publicaciones maliciosas en el lenguaje "hindi". Bajo esta metodología se crea un sistema que se puede aplicar a la detección maliciosa en tiempo real, principalmente porque se enfoca en la detección de tweets maliciosos en lugar de cuentas de spam [7].

Henriquez Miranda centra su atención en un sistema de minería de opiniones en idioma español acerca de los comentarios dados por clientes de diferentes hoteles. Estas normas se basan en las evaluaciones hechas en las dimensiones de valencia, excitación y el dominio. Como resultado, se consiguió un sistema más que aceptable, alcanzando una precisión del 94% para el dominio de turismo en hoteles [8].

Por otra parte, Villena-Roman Julio construyó un sistema automático de almacenamiento, análisis y visualización de información semántica extraída de mensajes de Twitter, diseñado para proporcionar a las administraciones públicas una herramienta para analizar los patrones de comportamiento de los ciudadanos, su opinión acerca de los servicios públicos, la percepción de la ciudad, etc. [9].

Las 3 referencias anteriores, a pesar de no pertenecer a PLN para problemas delictivos, han sido de apoyo para las autoras en la comprensión de las herramientas para análisis de los datos y la interpretación de la información expresada en lenguaje natural.

3. Primera aproximación: extracción de información

Para efectos de comparación, se consideró un reciente trabajo que recoge datos de redes sociales para el análisis de los sucesos delictivos que ocurren diariamente donde se hace uso de PLN. En tal trabajo, los autores exponen un análisis sobre información extraída de Twitter para detección de actos delictivos y sensaciones de inseguridad de la población en Barcelona. Sin embargo, en su conclusión, señalan que "Twitter" no funciona satisfactoriamente como herramienta de extracción de información para el problema que abordaron dado que la información tratada no puede ser exacta ni totalmente confiable debido a que el tiempo de estudio fue de solo una semana [10].

Tabla 1. Frecuencia de palabras.

Palabra	Descripción	Repetición
Mano armada	Herramienta de asalto	21
Disparo	Herramienta de asalto	5
Arma blanca	Herramienta de asalto	7
Arma de fuego	Herramienta de asalto	14
Motocicleta/Coche	Herramienta de asalto	2
Ciudad	Lugar de as alto	6 (Córdoba, Hidalgo, Puebla, Cuernavaca, Apizaco, Xonaca)
Tienda comercial	Lugar de as alto	3
Vía pública	Lugar de as alto	3
Transporte público	Lugar de as alto	1
Inocente asesinado	Resultado	16
Inocente herido	Resultado	11
Captura	Resultado	6
Fuga	Resultado	19
Persecución	Resultado	3
Asalto frustrado	Resultado	4
Resistencia	Defensa	2

Ante esta postura, en este artículo para minimizar tal complicación, se han hecho descargas de Twitter durante más de dos meses, lo cual, si bien no garantiza completamente la calidad de los datos, si ayuda a reforzar la fiabilidad de los mismos.

Esta tarea ha permitido contar con información adecuada y en condiciones de ser procesada para definir tanto diccionarios consistentes como satisfacción del análisis de sentimientos. La información conseguida durante 75 días se comparó con los medios de comunicación comunes (periódicos, páginas oficiales del gobierno mexicano, etc.) y se verificó la similitud con los mismos respecto a su semántica.

3.1. Recopilación de datos

Se inicia el desarrollo del trabajo con una inspección dentro de Twitter sobre la inseguridad reportada por diferentes usuarios de habla hispana alrededor del mundo. Como consecuencia, la búsqueda de información se ubica en el análisis delictivo sobre los asaltos. Tal problema es un reto principalmente para países del tercer mundo porque la inseguridad en estos países se refleja principalmente en el robo y/o hurto a personas en diferentes espacios abiertos (vía pública, en el transporte público, comercios, etc.).

3.2. Propuesta de banco de palabras para la búsqueda

A partir de la recopilación de datos, fue posible construir un banco de palabras para la búsqueda y extracción de información que fueran consistentes con el problema que nos ocupa. Este banco contiene conjugaciones del verbo “asaltar” (una de las palabras

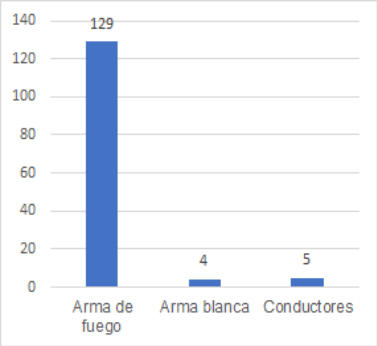


Fig. 1. Gráfica de Herramientas de asalto.

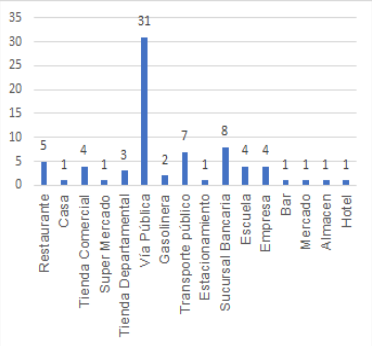


Fig. 2. Gráfica de lugares más asaltados.

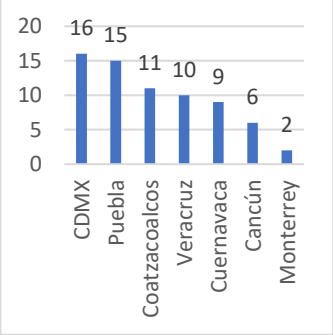


Fig. 3. Gráfica de estados con más asaltos.

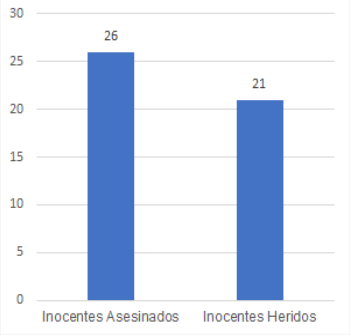


Fig. 4. Gráfica de desencadenamiento de un asalto.

frecuentes), y se suman a estas palabras algunos sinónimos, adjetivos o adverbios relacionados. El vector de palabras para iniciar la extracción formal de información en Twitter se muestra a continuación:

- Palabras: arrestar, arrestaron, asaltantes, asaltar, asaltaron, asalto, atrapar, atraparon, delictivo, delito, inseguridad, ladrones, robar, robaron, robo.
- Hashtags: asaltantes, asaltar, asalto, atrapar, delictivo, delincuentes, delito, inseguridad, ladrones, robar, robaron, robo.

Estas palabras son las candidatas para las búsquedas de los tweets, o bien, ser un hashtag. Para la descarga de los tweets se utilizó la herramienta web Vicinitas, que nos sirve para realizar una descarga de hasta 2080 tweets de los últimos 10 días con claro un parámetro de búsqueda. La información que se obtuvo de la búsqueda fue almacenada en una hoja de cálculo de Excel con los siguientes datos: identificador único del tweet, nombre del usuario, contenido del tweet, fecha de publicación, url, entre otros.

Dado que estas palabras se encuentran en lengua española, los datos recopilados no sólo se concentran en México, sino también, en otros países hispanohablantes. Sin

embargo, como ya se había definido anteriormente, sólo se tomaron en cuenta aquellos Tweets provenientes de la República Mexicana.

Es importante subrayar que, en esta primera aproximación, las descargas de tweets se hicieron de manera periódica durante los meses de enero y febrero de 2020. La información conseguida en estas fechas, permitió el inicio de un análisis del contenido de la descripción de los tweets. Este análisis dio lugar a obtener aquellas palabras que se presentaban de manera repetitiva (sin tomar en cuenta los parámetros de búsqueda). Este vocabulario se consideró para agrupar la información y definir los diccionarios.

De acuerdo a lo obtenido se encontró que se hace mención a: 1) los lugares (espacios públicos o cerrados) en donde ocurren los asaltos/robos, 2) las distintas armas o instrumentos que se usan al llevar a cabo un asalto/robo, así como también el transporte de asalto, 3) la forma de cómo terminan los sucesos después de un asalto/robo y 4) tipo de agresión que reciben las víctimas durante y después del asalto. La frecuencia de las palabras elegidas se muestra en las Figuras 1, 2, 3, 4.

4. Segunda Aproximación: Análisis de Sentimientos

En esta segunda fase se continuó realizando las descargas de los Tweets durante otros dos meses más y se añadieron los registros a los archivos previos. Pero para estas descargas se implementó otro procedimiento distinto al utilizado en las descargas de los Tweets de la primera aproximación. Para ello, se desarrolló un programa desarrollado en Python y usar la API de búsqueda de Twitter, Tweepy. Este programa tiene como ventaja en comparación a Vicinitas que controla con mayor cuidado el filtro de las descargas. La desventaja es el tiempo de uso, por ejemplo, si las descargas rebasan cierto número de peticiones, se deben esperar al menos 15 minutos antes de regresar a obtener información, lo cual puede llegar a generar conflicto para usuarios que requieran descargas masivas de datos. Otro conflicto es la aleatoria negación de la autorización de la cuenta de desarrollador de Twitter.

Una vez que se cuenta con descargas de tweets, se procedió a la de limpieza de datos estableciendo un estándar de la estructura de los tweets. El procedimiento para homogeneizar la información consistió en reunir todos los datos de una palabra en un solo archivo de manera periódica, en plazos de semanas, meses o quincenas. Por ejemplo, si se define un espacio catorcenal de descargas y se encuentra la palabra "asaltar" en ese lapso durante enero, los 14 archivos relacionados se deben conglomerar en uno solo archivo. Después se eliminan los saltos de línea innecesarios de los Tweets o los espacios dobles para ordenarlos de manera ascendente utilizando la fecha. El proceso continúa eliminando los "Re-Tweets" y evitar información repetida (como las tuplas repetidas en el modelo relacional de Bases de Datos).

Este proceso es importante porque nuestro interés es conocer el tipo de asalto, la frecuencia con la que se cometen actos delictivos en vez de contar cuántas veces se compartió un mismo Tweet. Por otro lado, se retiró de los archivos la información adicional que antes fue de utilidad, pero que en este paso ya no es necesaria (Id del Tweet, fecha y hora de este, usuario, etc.), obteniéndose así solo texto del Tweet. De la misma manera se suprimen "emojis" o "caracteres especiales" de cada uno de los Tweets, lo cual fue conseguido desarrollando un programa en C#. Este quehacer fue

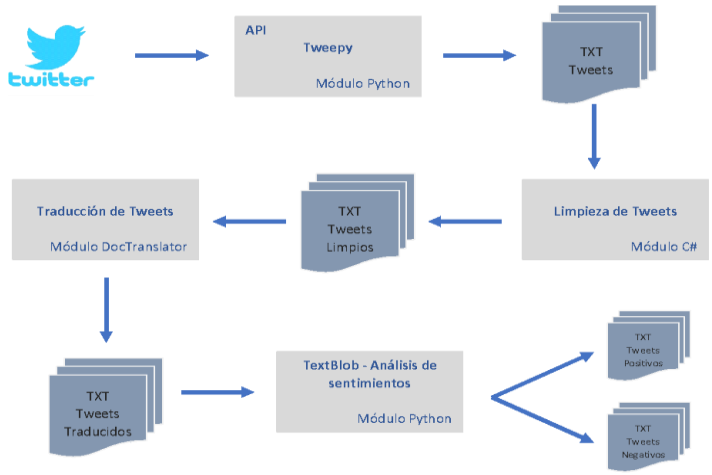


Fig. 5. Proceso de extracción y recolección.

mensual e incluye enero, febrero, marzo y abril de 2020, con 28 palabras de búsqueda en las descargas.

El procedimiento implicó el desarrollo de un programa en Python de utilidad posterior en el análisis de sentimientos. Este programa se encarga de leer en una subrutina los archivos de texto plano (.txt) que se obtuvieron (28 archivos por mes, durante 4 meses). Un proceso posterior toma uno a uno los Tweets y los traduce a inglés, esto dado en consecuencia de que la librería que se utilizó para Analizar Sentimientos (librería TextBlob) necesitaba manejar los Tweets en este lenguaje. Además, es importante enfatizar el hecho de que no se encontraron librerías de PLN que permitieran realizar el análisis de sentimientos para textos en español.

Sin embargo, la traducción de los tweets se interrumpió debido a que la librería utilizada (Google Translate [11]) había alcanzado el número límite de consultas, entonces se optó por traducir manualmente los Tweets en una página en Internet gratuita (onlinedoctranslator.com) todos los archivos de texto (28 archivos por mes, durante 4 meses). Cabe mencionar que se está consciente de que al utilizar herramientas de traducción siempre existe un margen de error en el resultado que proporcionan.

Después del trabajo de obtención, limpieza, concentración de resultados y análisis de sentimientos de la información, se obtuvieron los resultados esperados; algunos irrelevantes y otros sorprendieron a los autores porque no fueron considerados, sin embargo, dieron importancia significativa a la conclusión.

En la figura 1 se presenta de manera general proceso de la extracción y recolección de datos que se llevó en este apartado.

La siguiente subsección se encarga de exponer *Análisis de Sentimientos* y de los resultados obtenidos.

4.1. Búsqueda y análisis de sentimientos

En esta sección se presenta el análisis de sentimientos que surge a partir del primer análisis que arrojó la frecuencia de las palabras dentro del tweets. Un nuevo un enfoque

```
#Programa de análisis de sentimientos
Importar librerías para manipular archivos y Análisis de sentimientos;
var listaTweetsInglés := archivo de excel con Tweets (Inglés);
var listaTweetsEspañol := archivo de excel con Tweets (Español);
var resultadoPositivo := crear y abrir nuevo archivo de excel;
var resultadoNegativo := crear y abrir nuevo archivo de excel;
var resultadoNeutral := crear y abrir nuevo archivo de excel;
var contador := 0;
var nuevaListaTweetsEspañol := listaTweetsEspañol;
repetir para cada Tweet en listaTweetsInglés
    var analisis := calculaSentimientos(Tweet)
    var polaridad := polaridad de: analisis
    si (polaridad es igual a 0)
          agregar en resultadoNeutral := nuevaListaTweetsEspañol en
contador;
    FinSi
    si (polaridad es mayor que 0)
         agregar en resultado Positivo := nuevaListaTweetsEspañol en
contador;
    FinSi
    si (polaridad es menor que 0)
          agregar en resultadoNegativo := nuevaListaTweetsEspañol en
contador
    FinSi
    contador := contador + 1
FinRepetir.
```

Algoritmo 1. Código para la detección de sentimientos.

Tabla 2. Ejemplos de tweets por polaridad.

< 0	En violento #asalto balean a pasajero de la Línea Periférico antes de la #RectaCholula, los asaltantes se dieron a la fuga. Pasajeros lo auxiliaron mientras el chofer los traslado a un hospital de San Pedro Cholula. #TransportePublico #Puebla #PueblaNoticias | Pol: -0.8
	La situación se ha vuelto cada vez peor! Aparentemente un pasajero, que viajaba en la parte trasera, se resistió y discutió con uno de los asaltantes que realizo por lo menos un disparo | Pol: -0.25
	Neutral
> 0	Tras #Persecucion en #Apizaco, policías #Aseguran a probable responsable de #Asalto a #Transeunte | Pol: 0.1
	¡Adoro los finales felices! "Hombre persigue a #asaltantes que le quitaron 200mp; ellos chocan su moto y mueren." | Pol: 0.75

e interpretación ha sido centro de discusión en esta parte: Se señala una reacción al haber sido víctima de algún asalto/robo.

El análisis de sentimientos en este contexto fue desarrollado con Python y para ello es necesario traducir al idioma inglés la información recogida, tal como se describió al principio de esta sección.

TextBlob [12] es una biblioteca dedicada al PLN y es necesaria en el análisis de sentimientos. Esta analiza una cadena de texto generando un peso que se interpreta como polaridad, la cual es la emoción expresada dentro de la oración y tiene un rango de -1.0 a +1.0, lo que significa un resultado interesante y relevante en el artículo.

Tabla 3. Frecuencia de las palabras negativas y positivas.

	Negativas		Positivas	
N°	Palabras	Frecuencia	Palabras	Frecuencia
1	Odio	614	Vive/Vida	254
2	Arma(s)	348	Responsables(s)	230
3	Miserables	342	Justicia	219
4	Mierda	309	Libre	193
5	Corrupción	282	Atrapar	191
6	Violento	267	Cárcel	159
7	Maldito	243	Defender	121
8	Víctima(s)	242	Preso(s)	112
9	Inseguridad	218	Denunciar	111
10	Pinche	167	Arresto(s)	108
11	Terrible	144	Detenido(s)	103
12	Imbécil	142	Libertad	101
13	Peligroso	76	Inocentes	99
14	Dolor	50	Respeto	97
15	Desgracia	44	Feliz	96
16	Infeliz	41	Merecer	94
17	Tristeza	41	Suerte	92
18	Pánico	37	Prevención	91
19	Impotencia	32	Detención	81
20	Coraje	31	Captura	76
21	Ansiedad	20	Asegurar	63
22	Lesiones	20	Protección	42
23	Vándalos	20	Esperanza	41
24	Muerte inocente	20	Vigilancia	22

La polaridad que asigna TextBlob a la información es señalada en tres categorías: Positiva, Negativa y Neutral. Para propósitos de este documento, es válida la etiqueta Positiva y Negativa.

En el Algoritmo 1 se presenta el pseudocódigo implementado para el Análisis de Sentimientos.

Hemos insistido que los Tweets son una cadena de texto que ha sido depurada y expresada de manera adecuada para ser evaluada en el análisis de sentimientos. Cada Tweet presentaba una ponderación diferente.

En la tabla 3 se pueden observar algunos ejemplos representativos. Es oportuno señalar que Tweets neutrales no son relevantes en este estudio, lo cual responde a que, en la creación de los diccionarios, una disyuntiva de positivo y negativo era de utilidad.

De acuerdo con lo anterior se concluye:

Valores positivos:

— Cuando existe un asalto frustrado, la mayoría de las veces son las mismas víctimas las que se logran defender, o estas reciben apoyo de otros civiles.

— Existen casos en los que los asaltantes salen heridos por alguna situación, y las víctimas toman una actitud gratificante ante ello.

— Después de un asalto, es un resultado de alivio si las pérdidas solo fueron cosas materiales.

Valores negativos:

— En la mayoría de los asaltos, las víctimas demuestran su inconformidad con el sistema de vigilancia/seguridad y con los responsables de la supervisión.

— La mayoría de las emociones reflejadas en los Tweets son enojo, tristeza, frustración.

— En casi ninguno de los asaltos, las víctimas presentaron la denuncia ante la fiscalía correspondiente. Cuando las personas asaltadas denunciaron, su emoción es desalentadora porque el proceso fue largo y no satisfactorio.

— Se encontraron algunos casos donde la policía o militares fueron los que agredieron a los civiles.

— En la mayoría de los asaltos, la policía no brinda el apoyo necesario ni actúa de la manera correspondiente. En casos particulares, el apoyo fue completamente nulo.

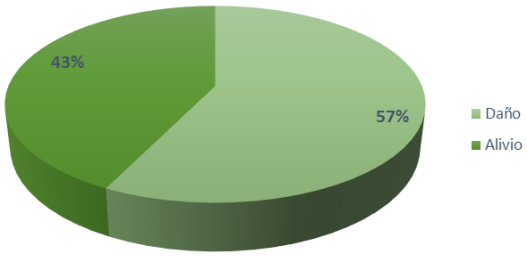


Fig. 6. Gráfica de porcentajes de Daño y Alivio.

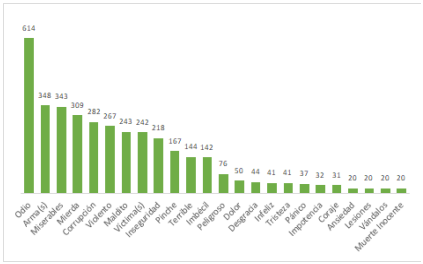


Fig. 7. Gráfica de frecuencias de *Daño*.

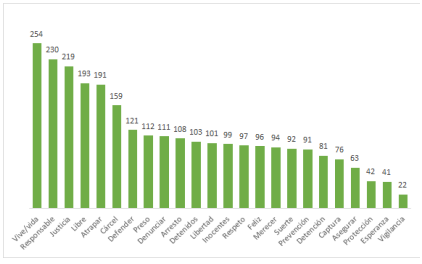


Fig. 8. Gráfica de frecuencias de *Alivio*.

5. Conclusiones y trabajo a futuro

En este trabajo, una investigación sobre los asaltos que ocurren en la sociedad de la república mexicana por medio de la red social Twitter ha sido desarrollado. Considerando que Twitter es flexible y permite generar datos con gran variabilidad de opiniones, se eligió a esta red social como objeto de estudio extracción de datos.

Una bondad de Twitter es la posibilidad es ajustar con PLN las opiniones de usuarios y establecer tendencias, describir sucesos o noticias identificadas como hashtags. propuesta resulta flexible para estos casos.

La gran cantidad de datos que se encontró dentro de los tweets (128,000 seleccionados de 300,000) fue el resultado del esfuerzo de las autoras que permitió alcanzar el objetivo de este artículo.

La gran variabilidad de Tweets, giraban alrededor de mensajes de alerta que se centraban en avisos acerca de asaltos en algún lugar específico y con ciertas características. Este tipo de Tweets son muy importantes porque produce eco en la red para que otros usuarios se protejan y tomen precauciones de un lugar y hora determinados. Otros avisos interesantes son los tweets que escriben personas que fueron espectadoras de un asalto. El banco de preguntas fue el conductor conveniente para una búsqueda de información consistente y correcta.

Alivio y Daño

El análisis de sentimientos hecho en cada uno de los tweets generó dos diccionarios: positivo y negativo. Una revisión y evaluación sobre los diccionarios fue realizada provocando el planteamiento de las siguientes preguntas:

- ¿Qué es “positivo”?
- ¿Qué es “negativo”?

Es sabido que el tema de los asaltos no es algo que podría definirse exactamente como algo bueno o que pueda siquiera acercarse a ser positivo, sin embargo, se admite que palabras aproximadas a positivo puedan ajustarse a la interpretación de Tweets que calificados como “buenos”.

Se ha nombrado Alivio al diccionario positivo, este nombre responde a la acción de consuelo cuando un asalto no fue exitoso para el asaltante o bien, la víctima no sufrió daños graves físicamente o solo fueron pérdidas materiales o cuando en el asalto frustrado el agresor es lesionado.

Para el diccionario negativo denominado Daño, se encontraron diversos identificadores. Este diccionario significa la gravedad de un asalto y los daños, los cuales son traducidos como lesiones/agresiones que ponen en riesgo la vida. Esta situación genera sentimientos negativos similares al dolor, tristeza, odio, resentimiento, impotencia, etc.

Circulan en el análisis de los Tweets de este trabajo, frases alrededor de incertidumbre, decepción, desaliento y pensamientos de ineptitud hacia las autoridades. Estas palabras se encuentran tácitas en las siguientes oraciones: a) Hay más asaltos vía pública que en cualquier otro lugar, b) En la mayoría de los asaltos, el agresor consigue escapar.

En el mes de marzo, algunos delincuentes tomaron ventaja de la situación sanitaria del país y asaltaron casas haciéndose pasar por trabajadores del sector salud o policías. Además, la actividad delictiva no cesó a pesar del escenario actual.

En marzo, algunas víctimas denuncian que fueron asaltados por el hecho de poseer víveres o productos de primera necesidad, esto debido al COVID-19. Se asume que, si hubiera sido posible desarrollar este trabajo en mayo y junio de 2020, se encontrarán muchas expresiones de agresiones a personas portadoras del COVID-19.

La mayoría de los asaltos descritos se llevaron a cabo dentro de México, más que en cualquier otro país hispanohablante.

El trabajo futuro consiste en mapear los resultados de este trabajo en un Sistema de Información Geográfico para mostrar gráficamente la concentración de zonas vulnerables y las frecuencias de las distintas agresiones delictivas.

Referencias

1. Arauz, R.M.L.: Sistema experto de clasificación y apoyo al diseño arquitectónico de edificios inteligentes. México: UNAM (1998)
2. Schalkoff, R. J.: Artificial intelligence: an engineering approach. Michigan: McGraw-Hill (1990)
3. Barr, A., Feingenbaun, E.A.: The handbook of intelligence artificial. Eyrolles, París (1986)
4. Feldman, R., Sanger, J.: The text mining handbook. Cambridge University Press, http:// www. safaribooksonline.com (2009)
5. Celaya, J.: La Empresa en la WEB 2.0. Editorial Grupo Planeta (2011)
6. Python Software Foundation, 201 (2008)
7. Kulkarni, S.: Detection of Malicious Microblogs from Online Social Networking Sites using Natural Language Processing (NLP) Techniques. In: 5th International Conference On Innovations in IT and Management. Sinhgad Institute of Management and Computer Application (SIMCA) (2020)
8. Sociedad Española para el Procesamiento del Lenguaje Natural: Minería de opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles. https://www.redalyc.org/pdf/5157/515754423002.pdf (2020)
9. Villena-Román, J., Luna-Cobos, A., González Cristóbal, J.: Análisis semántico de la opinión de los ciudadanos en redes sociales en la ciudad del futuro. Procesamiento del Lenguaje Natural, 53, pp. 159–162. http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/ article/view/5064/2952 (2014)
10. Puertas-Criado, S.: El Uso de Twitter y Tecnologías SIG como herramienta para la detección de victimización y percepción de inseguridad. Universitat Pompeu Fabra Barcelona (2017)
11. Han, S.: GoogleTrans (Versión 2.4.0) https://pypi.org/project/googletrans/ (2018)
12. Loria, S.: TextBlob (Versión 0.15.3) https://pypi.org/project/textblob/ (2019)

ISSN 1870-4069

Identificación y pesado de términos para la detección de depresión en Twitter

Monserrat Vázquez-Hernández, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez

Instituto Nacional de Astrofísica, Óptica y Electrónica, Laboratorio en Tecnologías del Lenguaje, Puebla, México

{mvazquez, villasen, mmontesg}@inaoep.mx

Resumen. Las redes sociales son parte de la vida diaria de las personas, a través de éstas los usuarios comparten información acerca de cualquier tema incluyendo aspectos emocionales, de ahí el interés de aprovecharlas para la detección de usuarios que sufren ciertos trastornos mentales. En este trabajo se propone un método para la identificación del léxico específico asociado a la depresión en Twitter. Este método identifica los términos más discriminativos y posteriormente construye una nueva representación de los textos al calcular el peso del resto de los términos con respecto al léxico identificado. Para validar el esquema de pesado propuesto se implementó una red neuronal de tipo CNN-LSTM para la detección de tuits relacionados con depresión. Los resultados obtenidos indican que este nuevo esquema de pesado mejora la detección de mensajes de usuarios con depresión en *Twitter*.

Palabras clave: Detección de depresión, Twitter, pesado de términos.

Term Identification and Weighing for Depression Detection in Twitter

Abstract. Social networks are part of people's daily lives, through these, users share information about any topic including emotional aspects, hence the interest in taking advantage of them to detect users suffering from certain mental disorders. This work proposes a method for identifying the specific lexicon associated with depression on Twitter. This method identifies the most discriminating terms and subsequently constructs a new representation of the texts by calculating the weight of the rest of the terms with respect to the identified lexicon. To validate the proposed weighing scheme, a CNN-LSTM neural network was implemented to detect tweets related to depression. The results obtained indicate this new weighing scheme improves the detection of messages from depressed users on *Twitter*.

Keywords: Depression detection, twitter, term weighing.

1. Introducción

Más de 300 millones de personas sufren depresión a nivel mundial siendo ésta la principal causa de ausentismo laboral en todo el mundo. Según la Organización Mundial de la Salud (OMS), en el mundo una de cada trece personas sufre depresión o algún trastorno mental relacionado[1].

Actualmente, las redes sociales son una parte integral en nuestras vidas diarias, a través de ellas los usuarios comparten información sobre sus intereses, opiniones, planes futuros e incluso sobre su estado emocional, por ejemplo, sus alegrías y tristezas. Es por ello que estas plataformas brindan una oportunidad para transformar los servicios de detección temprana de diferentes trastornos mentales como la depresión. Diversos trabajos de investigación se han enfocado en la detección de depresión a partir del contenido de redes sociales siguiendo, en su mayoría, un enfoque supervisado de clasificación de textos [1].

Twitter es un medio dinámico en el cual sus usuarios comparten mensajes frecuentemente durante el transcurso de sus actividades diarias. Esta característica ha convertido a este medio en uno de los más usados para la detección de personas que sufren depresión [2]. Cabe mencionar que la tarea no es sencilla, en primer lugar, los tuits son mensajes muy cortos que normalmente no proveen información de contexto y en segundo lugar, los indicadores de depresión suelen manifestarse de manera muy sutil, por lo tanto, su detección no es obvia para el lector. A pesar de estas características, trabajos recientes han reportando resultados muy alentadores en la detección de usuarios que sufren depresión y de otros trastornos mentales [3].

La mayoría de los métodos previos, orientados a la detección automática de depresión en redes sociales, han seguido un enfoque supervisado que los hace depender de datos etiquetados para su construcción. Estos métodos han explorado distintas representaciones y técnicas de clasificación, desde técnicas tradicionales hasta basadas en aprendizaje profundo [11,13,14,15]. En contraste, en este trabajo proponemos un esquema de pesado de términos orientado al dominio de depresión. Éste se basa en la identificación de los términos más discriminativos relacionados con depresión a partir de un conjunto etiquetado de referencia y posteriormente, el pesado del resto del vocabulario con respecto a su cercanía semántica con los términos identificados. Este nuevo enfoque de pesado, al igual que los tradicionales TF y TF-IDF, puede usarse en conjunto con distintos clasificadores permitiéndoles ponderar de mejor manera la información del dominio de aplicación, depresión en nuestro caso, y con ello su identificación.

El resto del artículo está organizado de la siguiente manera: la sección 2 presenta información acerca de trabajos relacionados donde previamente se ha abordado la tarea de detección de depresión en *Twitter*, en la sección 3 se describe el modelo propuesto para la identificación de los términos más frecuentes y con ellos la definición de pesos del vocabulario, en la sección 4 se evalúa el modelo mediante la implementación de diferentes esquemas de clasificación incluyendo

[1] https://www.who.int/es/news-room/fact-sheets/detail/depression

una red neuronal CNN-LSTM para la tarea de predicción, por último, en la sección 5 se presentan conclusiones y posibles direcciones futuras de trabajo.

2. Trabajo relacionado

La depresión es una enfermedad mental que afecta el equilibrio emocional de las personas. Su detección está dada por diferentes patrones de comportamiento de los individuos que la padecen [4]. En los últimos años, con el surgimiento y expansión del uso de redes sociales, diversas investigaciones se han llevado a cabo para la detección de este trastorno abordando el tema como un problema de *clasificación de textos*.

Estos trabajos han mostrado que el uso de ciertas palabras por parte de un usuario pueden dar pistas importantes sobre su condición psicosocial. Estudios en psicología sugieren que el uso de palabras específicas en nuestro vocabulario revelan indicadores sobre nuestro estado emocional [5]. Diferentes trabajos relacionados con la detección y análisis de depresión revelan que las palabras que usan las personas deprimidas permiten distinguirlas de entre el resto de las personas [3].

Siguiendo esta misma línea de trabajo, Tsugawa et al. así como Nadeem [6,7] realizaron un análisis respecto al vocabulario usado en tuits de usuarios con y sin depresión. Concluyen que observar la frecuencia de ciertas palabras es de gran utilidad para distinguir el comportamiento de una persona con este trastorno. De manera similar, Cavazos-Rehg et al. [8] realizaron un estudio exploratorio de tuits de varios usuarios observando que de acuerdo a las palabras más frecuentes, los usuarios pueden ser asociados al trastorno de estudio.

Por su parte, Stankevich et al. [9] combinaron un enfoque de bolsa de palabras con características estilométricas y morfológicas para la detección de depresión. Usaron representaciones distribuidas (*embeddings*) de palabras seleccionando las n palabras más informativas para cada usuario y obteniendo su representación a partir del promedio ponderado de los vectores de esas palabras. En sus experimentos observaron que el enfoque basado en *embeddings* aportó resultados inferiores a los obtenidos con el enfoque de bolsa de palabras, lo que les permitió reafirmar la utilidad de la frecuencia del uso de palabras para la predicción de depresión a partir de textos de redes sociales.

Basados en la evidencia previa de la utilidad de recursos léxicos, Losada y Gamallo [10] propusieron varios modelos para representar los textos a través de léxicos predefinidos y explotar las ventajas que éstos ofrecen. Después de hacer una experimentación con diferentes léxicos concluyeron que, hacer énfasis en la presencia de adjetivos ayuda a la detección de expresiones relacionadas con la depresión.

En resumen, varias de las investigaciones previas coinciden en que los textos de usuarios que padecen depresión muestran una existencia frecuente de emociones negativas y un constante uso de palabras referidas a síntomas y medicamentos relacionados con la enfermedad, por lo tanto, utilizan un vocabulario muy similar al momento de expresarse.

Esto indica que el análisis de frecuencia en el uso de ciertas palabras es un elemento distintivo para detectar la depresión. Nuestro enfoque se soporta en esta observación para proponer un nuevo esquema de pesado de palabras particularmente orientado a la tarea de detección de depresión.

3. Esquema de pesado propuesto

El propósito de este trabajo es diseñar un nuevo esquema de pesado de términos orientado a la detección de depresión. Este proceso parte de un corpus de textos etiquetados como deprimidos y no-deprimidos. A partir de ellos, en una primera etapa se identifican los términos más discriminativos de la depresión, es decir, se crea un léxico de depresión. Posteriormente, en una segunda etapa, se calcula un peso para cada término del vocabulario indicando su cercanía semántica con las palabras del léxico extraído. Las siguientes subsecciones explican los dos pasos de este proceso y la siguiente sección muestra su aplicación en la identificación de tuits relacionados con la depresión.

3.1. Identificación de un léxico de depresión

Para el análisis del lenguaje se utilizan publicaciones de *Twitter* etiquetadas como deprimidos y no deprimidos. El principal objetivo en esta etapa es detectar aquellas palabras más representativas de los tuits etiquetados como deprimidos.

Para la extracción de este léxico, y para los experimentos posteriormente reportados, se utilizó un conjunto de datos en inglés que consta de 11,595 tuits de los cuales 8,000 son de la clase no-depresivos y 3,595 de la clase depresiva. [2]. Para los experimentos en este trabajo, los datos se dividieron en conjuntos de entrenamiento y prueba de manera aleatoria: el 80 % (9,276 tuits) se utilizó para la extracción del léxico de depresión y el cálculo del peso de los términos, además fueron usados para entrenar los distintos clasificadores; el restante 20 % (2,319 tuits) se utilizó para evaluar el desempeño de los clasificadores.

Los pasos que corresponden al proceso de extracción del léxico de depresión son los siguientes:

1. Preprocesamiento de tuits. El conjunto de datos (entrenamiento y prueba) fue preprocesado de la siguiente manera:
   - Se convierten a minúsculas todos los tuits.
   - Se realiza tokenización de los tuits.
   - Se eliminan espacios redundantes y referencias a URLs.
   - Se eliminan las referencias a usuarios '@' de *Twitter*.
   - Se eliminan signos de puntuación y números.

[2] Se utiliza el conjunto de datos propuesto en el trabajo https://github.com/viritaromero/Detecting-Depression-in-Tweets el cual contiene 10,314 tuits (8,000 no-depresivos y 2,314 depresivos), adicionalmente se agregaron 1,281 tuits de la clase depresiva extraídos de Twitter con la herramienta Twint https://github.com/twintproject/twint

- Se eliminan palabras vacías haciendo uso de las palabras definidas en la librería NLTK de Python para el idioma inglés.
- Se aplica un proceso de lematización.

Extracción de las palabras más frecuentes tanto para el subconjunto de tuits depresivos como para no-depresivos. Algunas de estas palabras se muestran en la Tabla 1.

Tabla 1. Palabras más frecuentes en tuits deprimidos y no deprimidos.

Deprimidos	**No deprimidos**
1. depression	1. love
2. care	2. thank
3. sad	3. day
4. hate	4. good
5. hurt	5. think
6. stress	6. hope
7. problem	7. great
8. suicide	8. well
9. pain	9. fun
10. heart	10. happie

2. Cálculo de la frecuencia normalizada de cada una de las palabras extraídas. Esta frecuencia se calcula por separado para los vocabularios de los tuits depresivos y no-depresivos de acuerdo con la fórmula 1, donde $freq(t, C)$ es la frecuencia del término t en los tuits de la clase C (depresivo o no-depresivo) y $freq_{max}(C)$ es la frecuencia correspondiente al término más común de la clase C:

$$freq_{norm}(t) = freq(t, C)/freq_{max}(C). \quad (1)$$

   El puntaje resultante, expresado por la frecuencia normalizada se obtiene de cada término en cada una de las clases, por lo que, los términos presentes en ambas tendrán dos puntajes. En la Tabla 2 se muestran algunos de los puntajes obtenidos.
3. Selección del conjunto de términos que conforman el léxico de depresión denotado como $\mathcal{L}_{dep}$. Para ello se consideran todos los términos cuya frecuencia normalizada sea mayor a un umbral β predefinido (el umbral β puede tomar valores de 0.01 a 0.06). En caso de que la palabra ocurra en los vocabularios de ambas clases, se mantienen solo aquellas con frecuencia normalizada mayor en la clase depresión que en la de no-depresión.

   La Tabla 3 muestra ejemplos de los léxicos de depresión para dos umbrales distintos.

Tabla 2. Frecuencia normalizada de los términos en tuits depresivos y no depresivos.

Tuits con depresión		Tuits sin depresión	
Palabra	$freq_{norm}$	Palabra	$freq_{norm}$
depression	1.000	love	1.000
care	0.070	thank	0.803
sad	0.058	day	0.699
hate	0.056	good	0.696
hurt	0.056	think	0.445
stress	0.049	hope	0.413
problem	0.043	great	0.385
suicide	0.042	well	0.380
pain	0.041	fun	0.261
heart	0.040	happie	0.259

Tabla 3. Ejemplos de palabras de léxicos de depresión para umbrales β distintos.

$\mathcal{L}_{dep}(\beta = 0{,}02)$	$\mathcal{L}_{dep}(\beta = 0{,}03)$	$\mathcal{L}_{dep}(\beta = 0{,}04)$
depression	depression	depression
people	people	people
someone	life	life
care	care	care
hurt	hurt	hurt
stress	sad	sad
sad	fuck	stress
understand	hate	cause
problem	stress	understand
anxiety	problem	anxiety
pain	suicide	suicide
suicide	anxiety	sometimes

3.2. Pesado de términos orientado a la tarea

Una vez identificado el léxico representativo de usuarios depresivos ($\mathcal{L}_{dep}$), éste es utilizado para calcular el peso del resto de los términos del vocabulario. El peso se determina a partir de la *cercanía semántica* de cada término con respecto al léxico pre-establecido, para lo cual se utilizan representaciones distribuidas (*embeddings*) de los términos. Los *word embeddings* se definen como una técnica para modelar el lenguaje que representa cada palabra a través de un vector que es calculado observando los contextos de uso de cada palabra en grandes corpus de documentos [11]. En nuestro caso, se utilizaron las representaciones de *word embeddings* de GloVe (Global Vector for Word Representation) calculados sobre *Wikipedia* (6B tokens, 400K vocab, uncased 300d vectors) [12].

El cálculo de los pesos de los términos se realiza mediante los siguientes pasos:

1. Se determina el vector (*word embedding*) de cada una de las palabras del léxico de depresión $\mathcal{L}_{dep}$.
2. Se calcula un vector que represente a todo el léxico del lenguaje depresivo. Este vector único lo denominaremos *embedding depression vector* (EDV) y lo definimos como el vector promedio de los vectores de los términos en $\mathcal{L}_{dep}$. El objetivo de este vector es capturar la "semántica"del lenguaje depresivo.
3. Se determina el grado de asociación entre un término cualesquiera y lenguaje depresivo modelado en EDV, para ello se mide la similitud coseno entre el vector del término a ponderar y el vector EDV. De este modo, el peso de cada término corresponde a su similitud con el vector EDV.
   En la Tabla 4 se presenta un listado de algunas de las palabras fuertemente relacionadas, de acuerdo a la similitud coseno calculada, al léxico de depresión $\mathcal{L}_{dep}$, considerando un umbral β =0.01. En este listado se pueden observar términos asociados a emociones negativas además de algunos relacionados con síntomas y tratamientos de la depresión.

Tabla 4. Palabras fuertemente relacionadas al léxico de depresión, correspondientes a $\mathcal{L}_{dep}(\beta$ =0.01)

Término	Similitud con EDV	Término	Similitud con EDV
need	0.697	treatment	0.583
pain	0.648	patient	0.577
suffer	0.644	lose	0.561
fear	0.641	chronic	0.543
treat	0.636	hurt	0.534
rather	0.621	wrong	0.527
care	0.614	trauma	0.523
help	0.613	afraid	0.520
problem	0.612	sick	0.511
stress	0.610	depression	0.499
enough	0.603	cure	0.498
mental	0.585	effect	0.492
heart	0.584	respect	0.490

4. Evaluación del modelo de predicción

Diferentes trabajos se han propuesto para la predicción de depresión en redes sociales. Estos esfuerzos se han realizado en diferentes niveles, es decir, se puede considerar todo el historial de un usuario como un único documento a partir

del cual se estimará la predicción ó se puede dar una estimación para cada tuit individualmente. En este trabajo, la tarea de predicción se realiza a nivel de tuit.

Para concluir sobre el beneficio del esquema de pesado propuesto, éste se comparó contra un esquema de pesado tradicional (*tf-idf*) utilizando diferentes algoritmos de clasificación.

Clasificadores Para evaluar el esquema de pesados propuesto se utilizaron los siguientes algoritmos de clasificación: 1) Clasificador Naïve Bayes, 2) *k-nearest neighbors* y 3) un modelo SVM. En cada caso los textos fueron representados usando una bolsa de palabras (BoW) con (i) pesado *tf-idf* tradicional y (ii) usando el esquema de pesado propuesto (es decir, pesando cada palabra respecto al EDV depresivo). Adicionalmente, se implementó una arquitectura tradicional usada en aprendizaje profundo de tipo CNN-LSTM, los parámetros utilizados para la configuración de la arquitectura están basados en implementaciones anteriores para tareas de clasificación de textos: num. épocas=3, num. de filtros=64, tamaño de los filtros=5, dropout = 0.2 [3]. Usando esta red se realizaron dos experimentos: i) las palabras en los tuits se representan con sus correspondientes *embeddings* pre-entrenado de GloVe y ii) de igual forma, las palabras en los tuits se representan con sus correspondientes *embeddings* pre-entrenados de GloVe, seguido de la multiplicación de cada *embedding* por el peso propuesto en función de su relación con el vocabulario EDV depresivo, de esta forma se premió o castigó a las palabras.

Tabla 5. Comparación de resultados para la clase depresiva entre los esquemas de pesado tradicional (*tf-idf*) y el propuesto, con diferentes tipos de clasificadores. Se muestran diferentes umbrales para determinar el vocabulario del EDV depresivo.

	Precisión	Cobertura	F1-score
Naïve Bayes + tf-idf	0.91	0.87	0.88
Naïve Bayes + EDV $\mathcal{L}_{dep}(\beta$ =0.03)	**0.95**	0.83	**0.89**
Naïve Bayes + EDV $\mathcal{L}_{dep}(\beta$ =0.04)	0.95	0.82	0.88
K-NN (k = 5) + tf-idf	0.58	0.26	0.36
K-NN (k = 5) + EDV $\mathcal{L}_{dep}(\beta$ =0.03)	**0.66**	0.27	**0.38**
K-NN (k = 5) + EDV $\mathcal{L}_{dep}(\beta$ =0.04)	0.59	0.27	0.37
SVM + tf-idf	0.93	0.88	0.90
SVM + EDV $\mathcal{L}_{dep}(\beta$ =0.03)	0.94	0.90	0.92
SVM + EDV $\mathcal{L}_{dep}(\beta$ =0.04)	**0.97**	**0.90**	**0.93**
CNN-LSTM pretrained GloVe	0.69	0.50	0.58
CNN-LSTM pretrained GloVe * peso $\mathcal{L}_{dep}(\beta$ =0.03)	**0.68**	**0.64**	**0.66**
CNN-LSTM pretrained GloVe * peso $\mathcal{L}_{dep}(\beta$ =0.04)	0.68	0.63	0.65

Resultados En la Tabla 5 se muestran los resultados obtenidos en cada clasificador (precisión, cobertura y F1-score) para la clase de tuits depresivos. Como se puede observar, el esquema de pesado propuesto mejora el esquema tradicional *tf-idf*. La Tabla 5 muestra el efecto al considerar diferentes umbrales ($\beta = 0{,}03$ y $\beta = 0{,}04$) para determinar el vocabulario del EDV depresivo. En general, con un umbral más exigente, es decir, con un conjunto menor de palabras para formar el EDV se alcanzan mejores resultados.

5. Conclusiones y trabajo futuro

En este trabajo se propuso un nuevo esquema de pesado de términos para la detección de signos de depresión en publicaciones de *Twitter*. El peso de los términos se estimó en función de su similitud respecto al vector único del léxico asociado al lenguaje depresivo. Para identificar este léxico se presentó un método basado en las palabras más frecuentes. Los resultados alcanzados indican que el método propuesto mejora la clasificación en comparación a las representaciones tradicionales.

Como trabajo futuro se busca implementar el esquema propuesto con representaciones de *embeddings* diferentes como FastText, además evaluar el esquema en otros conjuntos de datos para la tarea de detección de depresión, lo que nor permita comparar nuestro esquema contra otros métodos. Por otro lado, dada la generalidad del método, éste podría evaluarse en otro tipo de tareas de clasificación como perfilado de autor o detección de engaño. Adicionalmente, sería posible aplicarlo a colecciones de datos en otros idiomas, en específico para el idioma español.

Agradecimientos. El primer autor agradece el apoyo otorgado por el CONACYT a través de la beca No. 650291. Los autores también agradecen el apoyo recibido por el CONACYT a través del proyecto de CB-2015-01-257383 para la realización de esta investigación.

Referencias

1. Coello Guilarte, D.L.: Clasificación translingue para la detección de depresión en usuarios de Twitter. Tesis de Maestría en Ciencias Computacionales, INAOE (2019)
2. Tsugawa, S., Mogi, Y., Kikuchi, Y., Kishino, F., Fujita, K., Itoh, Y., Ohsaki, H.: On estimating depressive tendencies of twitter users utilizing their tweet data. In: 2013 IEEE Virtual Reality (VR), pp. 1–4 (2013)
3. Sandhya, G.: Using Word Embeddings to Explore the Language of Depression on Twitter. Master Thesis in Computer Science, The University of Vermont (2019)
4. Spitzer, R. L., Md, K. K., Williams, J. B. W.: Diagnostic and statistical manual of mental disorders (DSM-5 R). Naklada Slap, Jastrebarsko, Croatia (2013)
5. Pennebaker, J. W., Booth, R. J., Francis, M. E.: Liwc 2007: Linguistic inquiry and word count. Austin, Texas, liwc.net. (2007)

6. Tsugawa, S., Mogi, Y., Kikuchi, Y., Kishino, F., Fujita, K., Itoh, Y., Ohsaki, H.: On estimating depressive tendencies of twitter users utilizing their tweet data. In: 2013 IEEE Virtual Reality (VR), pp. 1–4 (2013)
7. Nadeem, M.: Identifying depression on twitter. (2016)
8. Cavazos-Rehg, P. A., Krauss, M. J., Sowles, S., Connolly, S., Rosas, C., Bharadwaj, M., Bierut, L. J. : A content analysis of depression-related tuits. Computers in human behavior, 54, pp. 351–357 (2016)
9. Stankevich, M., Isakov, V., Devyatkin, D., Smirnov, I.: Feature engineering for depression detection in social media. In: ICPRAM, pp. 426–431 (2018)
10. Losada, D. E., Gamallo, P.: Evaluating and improving lexical resources for detecting signs of depression in text. Language Resources and Evaluation, pp. 1–24 (2018)
11. Wohlgenannt, G., Chernyak, E., Ilvovsky, D.: Extracting social networks from literary text with word embedding tools. In: Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), pp. 18–25 (2016)
12. Pennington, J., Socher, R., Manning, C. D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)
13. Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao: Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence (2015)
14. Chunting Zhou, Chonglin Sun, Zhiyuan Liu, Francis Lau: A C-LSTM neural network for text classification. (2015)
15. Mathieu C.: BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Canada (2017)

ISSN 1870-4069

Análisis de sentimiento a nivel de documento en críticas de cine en español

Maritza Flores Domínguez, José Luis Tapia Fabela

Universidad Autónoma del Estado de México,
Unidad Académica Profesional Tianguistenco,
División de estudios de posgrado,
México

maritza_fd@outlook.com,
joseluis.fabela@gmail.com

Resumen. El análisis de sentimiento (AS) determina la polaridad en los textos que contienen opinión. En la actualidad la mayoría de las investigaciones de AS se enfocan al idioma inglés a pesar de que cada día se hace más presente el español en la web. Por lo tanto, en este trabajo se desarrolló un método de AS a nivel de documento utilizando el corpus muchocine en español. El método propuesto emplea la red neuronal back-propagation para clasificar el corpus en cinco clases. Para esto se crearon diferentes muestras de entrenamiento normalizadas con tres tipos de preprocesamiento y se representaron mediante un conjunto de vectores binarios. Los resultados obtenidos muestran que el accuracy obtenido de evaluar la red es similar al de los clasificadores del estado del arte, debido a la muestra de entrenamiento, la configuración de la red, el número de clases utilizadas, así como al número de críticas empleadas.

Palabras clave: Análisis de sentimiento, nivel de documento, clasificación de la polaridad, red neuronal back-propagation, corpus de opiniones en español.

Document-Level Sentiment Analysis in Spanish Film Reviews

Abstract. Sentimental analysis (SA) determines the polarity in texts containing opinion. At present, most SA research focuses on the English language despite the fact that Spanish is increasingly present on the web. Therefore, in this work a method of SA was develop at the document level using the *muchocine* corpus in Spanish. The proposed method uses the back-propagation neural network to classify the corpus into five classes. For this purpose, different standardized training samples were created with three types of pre-processing and represented by a set of binary vectors. The results obtained show that the accuracy obtained from evaluating the network is similar to that of the state-of-the-art classifiers, due to the training sample, the network configuration, the number of classes used, as well as the number of reviews employed.

Keywords: Sentiment analysis, document level, polarity classification, back-propagation neural network, opinion corpus in Spanish.

1. Introducción

Saber qué piensan otras personas es importante durante el proceso de toma de decisiones [1]. Usando las opiniones resumidas de otros podemos decidir con mayor certeza. En la antigüedad para tomar buenas decisiones las personas recurrían a la interpretación de las entrañas de animales, el humo o los sueños. Los chinos usaban la sabiduría poética y el I Ching; los griegos consultaban el oráculo de Delfos [2]. En los 90's las personas para obtener información respecto de algún producto o servicio que deseaban adquirir preguntaban a sus amigos, conocidos, vecinos o familiares.

Con la llegada de la web 2.0 la forma de adquirir información cambió. Con el surgimiento de las redes sociales, foros, aplicaciones web, blogs, etc. [3] miles de usuarios que navegan a través de ellas hacen que la información crezca de manera exponencial, resultando imposible leer una a una todas sus opiniones. De la web 2.0 se pueden distinguir dos tipos textuales: los hechos, que representan oraciones objetivas que no emiten sentimiento y las opiniones, que describen el sentimiento que los usuarios emiten respeto a entidades y que resultan importantes al tomar decisiones [4]. El procesamiento del lenguaje natural PLN tiene entre sus múltiples tareas la clasificación de textos. Partiendo del concepto de clasificar los textos de forma automática con base en la opinión emitida por los usuarios de la web 2.0 surge el análisis de sentimiento (AS) [5].

La mayoría de las investigaciones realizadas en el análisis de sentimiento, se encuentran enfocadas al idioma inglés [6]. Existe una carencia de investigaciones de análisis de sentimiento (AS) en otros idiomas, esto resulta un problema si tenemos en cuenta que el AS nos permite tomar decisiones basados en las opiniones resumidas de otros y que en la web cada día se hace más presente el uso de idiomas diferentes al inglés entre ellos el español.

En este trabajo se desarrolló un método de análisis de sentimiento a nivel de documento en críticas de películas en español utilizando las opiniones contenidas en el corpus muchocine creado por Cruz [7]. Clasificar las opiniones del corpus muchocine es una tarea donde se encontraron áreas de oportunidad al analizar los trabajos del estado del arte. En el trabajo de Cruz [7] se realizaron experimentos aleatorios seleccionando 400 críticas del corpus. Se seleccionaron 200 críticas, puntuadas con 1 y 2 para conformar las críticas negativas. Las otras 200 críticas puntuadas con 4 y 5 se consideraron como críticas positivas. Cruz no utilizó el corpus completo por limitaciones de tiempo debido a que el cálculo de la orientación semántica es un proceso lento que depende del buscador Alta Vista, por lo tanto, clasifica las opiniones del corpus en dos clases. Cada solicitud al buscador tiene un lapso de tiempo de cinco segundos de espera, de esta forma no se satura al servidor de Alta Vista para que no les denegué el servicio.

En el trabajo de [8] se presenta un caso similar con respecto al trabajo de [7]. Las críticas con puntuaciones de 1 y 2 son consideradas como negativas y las críticas con puntuación 4 y 5 son consideradas como críticas positivas, por lo tanto, el número de críticas utilizadas es de 2,625: 1,274 corresponde a críticas negativas y 1,351 corresponden a críticas positivas. En ambos trabajos no utilizan las críticas con puntuación 3 (críticas neutras), limitándose a clasificar las opiniones en dos clases. Utilizar menos clases para clasificar las críticas de cine en español da ventaja al clasificador ya que existe menos confusión en el proceso y se puede obtener un

accuracy superior al 80%. Sin embargo, no representa la realidad del problema debido a que dos clases no aportan la información suficiente a los usuarios para decidir qué película ver.

El método de análisis de sentimiento a nivel de documento en críticas de cine en español desarrollado, se propuso a partir del análisis de las metodologías utilizadas en el estado del arte. Está formado de tres etapas. La primera etapa aplica pre-procesamiento al corpus: se sustituyen las palabras con acentos, se eliminan las stopwords y signos de puntuación. La segunda etapa representa las críticas que conforma el corpus muchocine en un conjunto de vectores binarios, mediante el uso de bolsa de palabras y pesado booleano. En la tercera etapa el corpus se clasifica en cinco clases utilizando la red neuronal back-propagation, mediante una validación cruzada de 10 particiones. Posteriormente se evaluó el accuracy obtenido por la red. Para evaluar accuracy de la red neuronal back-propagation se crearon cinco muestras de entrenamiento diferentes con características específicas. A la primera muestra no se le aplica pre-procesamiento. En la segunda muestra se sustituyen las palabras que están acentuadas. En la tercera muestra se eliminan las stopwords. En la cuarta muestra se eliminan los signos de puntuación. En la quinta muestra se sustituyen los acentos y se eliminan las stopwords. Finalmente, en la sexta muestra se sustituyen acentos, se eliminan los signos de puntuación y stopwords. Concluida la etapa de experimentación con cada muestra de entrenamiento se seleccionó la que obtuvo mejor accuracy y se usó en el comparativo realizado con los trabajos del estado del arte, obteniendo un accuracy superior al 80% en la clasificación.

Es importante resaltar que en nuestro trabajo de investigación se utilizó las 3878 críticas del corpus muchocine del año 2008. El AS se realizó a nivel de documento. Además, no se trabajaron con emoticonos debido a la estructura del corpus.

El presente trabajo se estructura de la siguiente forma. La sección 1 establece el propósito e importancia de la investigación. Pone en contexto la investigación con los trabajos anteriores. En la sección 2 se analizan los métodos de los trabajos del estado del arte relacionados con el análisis de sentimiento. A partir del análisis se sintetiza el método propuesto. La sección 3 explica la estructura del método propuesto para el análisis de sentimiento en críticas de cine en español. Se describe el corpus que se utilizó, así como el procedimiento que siguió el clasificador. En la sección 4 se muestra la experimentación realizada con el método propuesto. En la sección 5 se discuten los resultados obtenidos de la experimentación y posteriormente se comparan con los resultados de los métodos del estado del arte. En la sección 6 se presentan las conclusiones del trabajo. Finalmente, se exponen los trabajos futuros.

2. Estado del arte

Partiendo de los métodos propuestos por Cruz y Martínez para la tarea de análisis de sentimiento en críticas de cine en español, se propuso un método diferente para la resolución de este problema. La Tabla 1 presenta un comparativo entre los trabajos presentados en el estado del arte que utilizan el corpus muchocine.

En la primera columna de la Tabla 1 se muestran los dos trabajos del estado del arte, es decir, el trabajo de Cruz [7] y Martínez [8]. En el trabajo de Cruz [7] el número de críticas que utilizan es menor, debido a que las críticas con puntuación 1 y 2 se

Tabla 1. Estado del arte en el dominio de críticas de cine en español: muchocine.

Trabajos	Clases	No. críticas	Clasificador	Métricas de evaluación
Cruz [7]	Positivo Negativo	200 200	Calculo de la orientación semántica (SO) supervisada.	SO_{v1} accuracy = 0.775
			Calculo de la orientación semántica (SO) no supervisada.	SO_{v2} accuracy = 0.695
Martínez [8]	Positivo Negativo	1274 1351	Máquina de Soporte Vectorial (SVM)	Accuracy = 0.8773 Recuerdo = 0.8769 F1 – measure = 0.8771
			Naive Bayes	Accuracy = 0.8408 Recuerdo = 0.8401 F1 – measure = 0.8404

consideran como críticas negativas y las críticas con puntuación 4 y 5 son consideradas críticas positivas, clasificando las críticas en dos clases. Cruz no utilizó el corpus completo por limitaciones de tiempo debido a que el cálculo de la orientación semántica es un proceso lento que depende del buscador Alta Vista. Cada solicitud al buscador tiene un lapso de tiempo de cinco segundos de espera, de esta forma no se satura al servidor de Alta Vista, de lo contrario les denegaría el servicio.

En el trabajo de Martínez [8] se presenta un caso similar con respecto al trabajo de Cruz [7]. Las críticas con puntuaciones de 1 y 2 son consideradas como negativas y las críticas con puntuación 4 y 5 son consideradas como críticas positivas, trabajan con dos clases. En ambos trabajos no utilizan las críticas con puntuación 3 (críticas neutras). También podemos observar en la Tabla 1 que Cruz [8] utilizan dos clasificadores diferentes para sus experimentos, en el cual SVM es el clasificador que arroja mejores resultados.

Estas áreas de oportunidad encontradas en los trabajos de Cruz [7] y Martínez [8] son tomados en cuenta para este trabajo de investigación.

El método propuesto en esta investigación pretende trabajar con 5 clases: críticas muy negativas con puntuación 1, críticas negativas con puntuación 2, críticas neutras con puntuación 3, críticas positivas con puntuación 4 y críticas muy positivas con puntuación 5. Además, de emplear un clasificador diferente y poco utilizado en la tarea de análisis de sentimiento bajo el dominio de críticas de cine en español.

La idea de desarrollar un nuevo método basado en RNA, surge de revisar los trabajos del estado del arte. Estos trabajos sustentan el uso de un nuevo clasificador en la tarea de análisis de sentimiento en críticas de cine en español.

3. Método

El análisis de sentimiento nos permite asignar automáticamente un conjunto de opiniones a una serie de categorías. Su propósito es determinar cuáles expresan una opinión negativa o positiva. Los resultados del clasificador van a depender del pre-procesamiento del corpus, la forma de representar el texto contenido o incluso del clasificador utilizado.

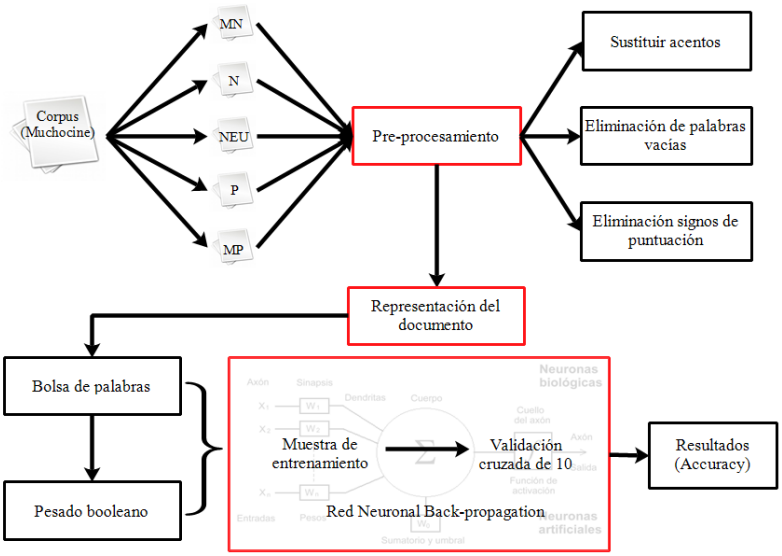


Fig. 1. Método propuesto.

En este trabajo se implementó un método de análisis de sentimiento en críticas de cine en español conformado de tres etapas, Fig.1. La primera etapa tiene como objetivo aplicar pre-procesamiento en el texto del corpus como: sustituir los acentos de las palabras, eliminar las stopwords y los signos de puntuación. En la segunda etapa las críticas son representadas mediante vectores binarios, utilizando bolsa de palabras y pesado booleano. En la tercera etapa, las críticas son clasificadas por la red neuronal back-propagation, mediante una validación cruzada de 10 particiones, evaluando el accuracy de la red.

3.1 Corpus Muchocine

Muchocine es una web [9] destinada a una comunidad de críticas de películas en el idioma español. Las críticas contenidas en la web muchocine se encuentran ordenadas de tres formas: por género, nombre de la película y cine por décadas.

Tabla 2. Distribución del corpus muchocine.

Puntuación	No. De críticas
1	351
2	923
3	1253
4	890
5	461
Total de críticas	3878

El corpus muchocine en español de Cruz [7], fue extraído de las páginas HTML de cada una de las críticas de las películas contenidas en la web muchocine en febrero de 2008.

Dicha colección está constituida por 3878 críticas. Puntuadas con valores de 1 a 5. El valor 1 representa una película muy mala, 2 representa una película mala, las catalogadas en 3 son neutras, 4 representa una película buena y el 5 una película muy buena. El número de documentos sobre los que se han realizado experimentos se muestran en la siguiente Tabla 2.

Las críticas que conforman el corpus son escritas por usuarios comunes añadiendo dificultad a la tarea que se pretende resolver. Los textos pueden contener faltas de ortografía, puede haber incoherencias, error en los signos de puntuación, además de presentar distinto tamaño.

Las páginas HTML que conforman las críticas fueron transformadas en ficheros XML (uno por cada crítica). La estructura de dichos ficheros consta de: nombre del autor que realiza la crítica de la película, el nombre de la película, la puntuación asignada por el usuario, un breve resumen de la película y la crítica como tal.

3.2 Creación de las muestras de entrenamiento

Se crean seis muestras diferentes de entrenamiento para probar la red, así escoger la que da mejores resultados:

1. Muestra_MuchoCine1.xlsx (sin pre-procesamiento),
2. Muestra_MuchoCine2.xlsx (sustituir acentos),
3. Muestra_MuchoCine3.xlsx (eliminar stopwords),
4. Muestra_MuchoCine4.xlsx (eliminar signos de puntuación),
5. Muestra_MuchoCine5.xlsx (eliminar stopwords y signos de puntuación),
6. Muestra_MuchoCine6.xlsx (sustituir acentos, eliminar stopwords y signos de puntuación).

3.3 División de los datos

Concluida la etapa de pre-procesamiento del corpus muchocine, las críticas se transformaron en un conjunto de vectores binarios almacenados en archivos con formato xlsx. Uno con las críticas Muestra-MuchoCine y el otro con las clases Targets_MuchoCine.

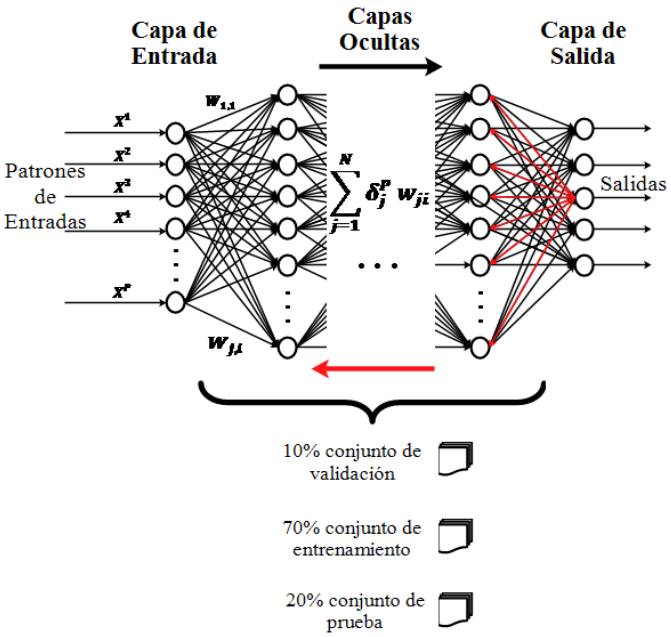


Fig. 2. División de los datos.

La red neuronal back-propagation se configuró para dividir aleatoriamente la Muestra-MuchoCine.xlsx en tres conjuntos de datos: entrenamiento, validación y prueba, Fig. 2:

1. El conjunto de validación utilizó solo el 10% de la Muestra-MuchoCine.xlsx para detener el entrenamiento de la red antes que se sobreajuste (validación cruzada de 10 particiones).
2. El conjunto de entrenamiento utilizó el 70% de la Muestra-MuchoCine.xlsx.
3. Para el conjunto de prueba se utilizó el 20% de la Muestra-MuchoCine.xlsx. Los vectores que la conforman no pertenecen a los dos conjuntos anteriores.

Donde:

1. Se inicializa los pesos (valores pequeños aleatorios).
2. Se escoge un patrón de entrada X^P.
3. Se realiza el proceso de propagación hacia adelante, a través de toda la red hasta que la activación alcance las neuronas de la capa de salida.
4. La red calcula los valores de δ para las capas de salida, $\delta_j^P = \left(t_j^P - O_j^P\right) f'\left(Act_j^P\right)$ usando los valores de las salidas deseadas para el patrón de entrada seleccionado.
5. La red calcula los valores δ para la capa oculta $\delta_j^P = \sum_{j=1}^{N} \delta_j^P \, w_{ji} \, f'\left(Act_j^P\right)$.
6. Se actualiza los pesos utilizando la fórmula: $\Delta_P w_{ij} = y\delta_I^P O_j^P$.

7. La red repite el paso 2 al 6 para todas las entradas que conforman la red.

Aquí $f'\left(Act_j^P\right)$ representa la derivada de la función de activación [10].

3.4 Configuración de la red back-propagation

En la Tabla 3 se muestran los parámetros que se usaron para la configuración de la red back-propagation en MATLAB. Las entradas están formadas por los conjuntos de vectores contenidos en el archivo Muestra_MuchoCine.xlsx. Las salidas están contenidas en el archivo Targets_MuchoCine.xlsx.

Tabla 3. Configuración de los parámetros de la red back-propagation en MATLAB.

Parámetros	
Entradas	Muestra_MuchoCine#.xlsx
Salidas	Targets_MuchoCine#.xlsx
Tamaño de las capas ocultas	3878 neuronas
División de datos	70% conjunto de datos de entrenamiento
(aleatoriamente)	20% conjunto de datos de prueba
	10% validación cruzada
Entrenamiento	Gradiante conjugado escalado (función trainscg back-propagation)
Rendimiento	Entropía cruzada (función crossentropy)
Gráficos	Matriz de confusión (función plotconfusion)
	Curvas ROC (función plotroc)
Número de iteraciones	1000
Métrica de evaluación	Accuracy

El tamaño de las capas ocultas de la red es de 3878 neuronas. Esto es equivalente al número de vectores contenidos en la Muestra_MuchoCine.xlsx. Debemos tener en cuenta que el número de neuronas ocultas determina la capacidad de generalización de la red. El número de neuronas ocultas se tomó de acuerdo con lo dicho por Hernández [11].

Hernández [11] propone que el número de neuronas debe ser menor al número de patrones de entradas.

Esto para evitar problemas de memorización. Se optó por usar 3878 neuronas en las capas ocultas. Se usó una validación cruzada de 10 particiones. La división para los conjuntos de datos se hizo tomando valores aleatorios.

Los parámetros utilizados para la configuración de la back-propagation en MATLAB son:

1. Para el entrenamiento se utilizó la función trainscg que es el gradiente conjugado escalado.
2. Para evaluar el rendimiento de la red se usó la función Cross-Entropy.
3. La red realiza un proceso de 1000 iteraciones para el entrenamiento, prueba y validación.
4. Se calcula el accuracy como métrica de evaluación.
5. Los resultados obtenidos son graficados mediante la matriz de confusión y las curvas ROC.

4. Experimentación

4.1 Pruebas con las muestras de entrenamiento

La Tabla 4 muestra el accuracy de la red neuronal back-propagation con las seis muestras de entrenamiento creadas. Los resultados están ordenandos de forma descendente.

Tabla 4. Accuracy de la red back-propagation al utilizar diferentes muestras de entrenamiento.

Muestra	Pre-procesamiento	Accuracy
Muestra 6 - Muestra_MuchoCine6.xlsx	Sustituir acentos, eliminar stopwords y signos de puntuación	0.8499
Muestra 5 - Muestra_MuchoCine5.xlsx	Sustituir acentos y eliminar stopwords	0.8383
Muestra 2 - Muestra_MuchoCine2.xlsx	Sustituir acentos	0.7891
Muestra 3 - Muestra_MuchoCine3.xlsx	Eliminar stopwords	0.7829
Muestra 1 - Muestra_MuchoCine1.xlsx	Sin pre-procesamiento	0.6877
Muestra 4 - Muestra_MuchoCine4.xlsx	Eliminar signos de puntuación	0.6091

Con la sexta muestra de entrenamiento se obtiene el mejor resultado de la experimentación (un accuracy de 0.8499).

En esta se aplican tres diferentes pre-procesamientos: sustituir acentos, eliminar stopwords y signos de puntuación. El pre-procesamiento ayudó al clasificador a interpretar de forma correcta las críticas debido a la eliminación de stopwords que por su frecuencia semántica no ayudan al entendimiento del texto contenido.

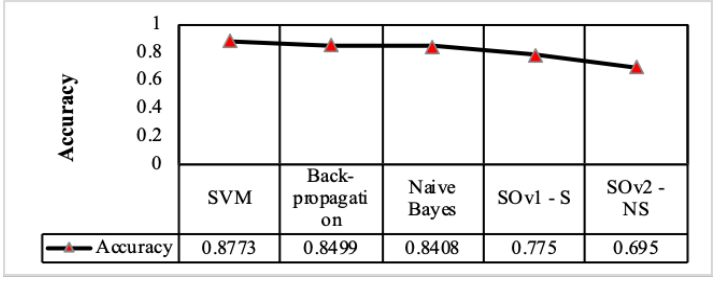


Fig. 1. Comparativa de los resultados obtenidos con el estado del arte.

Tabla 5. Posición del método propuesto en el estado del arte.

Posición	Autores	Clasificador	No. de Clases	No. de críticas	Accuracy
1	Martínez [8]	SVM	2 (Negativo, Positivo)	2625	0.8773
2	Método propuesto	Back-propagation	5 (Muy negativo, Negativo, Neutro, Positivo, Muy Positivo)	3878	0.8499
3	Martínez [8]	Naive Bayes	2 (Negativo, Positivo)	2625	0.8408
4	Cruz [7]	SO_{v1} - S	2 (Negativo, Positivo)	200	0.775
5	Cruz [7]	SO_{v2} - NS	2 (Negativo, Positivo)	200	0.695

Por otra parte, al combinar el pre-procesamiento anterior con sustituir acentos y eliminar signos de puntuación las funciones gramaticales y las relaciones sintácticas permanecen intactas. La quinta muestra de entrenamiento obtiene un accuracy del 0.8383. Los resultados son aceptables, debido a que se aplican dos tipos de pre-procesamiento al corpus, lo cual ayuda a mejorar la calidad en los textos. En la segunda muestra se sustituyen los acentos en el corpus y se obtiene un accuracy de 0.7891, lo anterior coincide con lo reportado en el estado del arte [12-13].

En la tercera muestra se eliminan las stopwords del corpus y da como resultado un accuracy de 0.7829. El accuracy obtenido con la tercera muestra es producto de la eliminación de aquellas palabras que por su frecuencia semántica no poseen valor alguno. Con esto, el clasificador distingue correctamente las críticas.

En la primera muestra el accuracy es de 0.6877. Esto se debe a que el texto sin pre-procesamiento es interpretado incorrectamente por el clasificador. Finalmente, con la cuarta muestra se obtienen un accuracy de 0.6091. El resultado es bajo porque los signos de puntuación ayudan a determinar las funciones gramaticales y las relaciones sintácticas entre los distintos enunciados del corpus, sin ellos es difícil la interpretación del texto.

5. Resultados y discusión

En la Gráfica 1 se muestra la comparativa del estado del arte con los resultados obtenidos, accuracy de 0.8499. La red neuronal back-propagation alcanza el segundo

lugar dentro de los clasificadores usados en la tarea de análisis de sentimiento en críticas de cine en español en el corpus muchocine. La propuesta supera los dos resultados de Cruz [7] en donde utilizan orientación semántica con una semilla ($SO_{V1} - S$) de forma supervisada y orientación semántica no supervisada utilizando dos semillas ($SO_{V2} - S$). También se superan los resultados obtenidos por Martínez [8] donde utilizan Naive Bayes y obtienen un accuracy de 0.8408.

De igual forma, en la Tabla 5 se observa la posición del método propuesto en el estado del arte. Los resultados obtenidos (segunda posición) son satisfactorios considerando que empleamos cinco clases para categorizar las opiniones. Además, de las 3878 críticas que conforman el corpus muchocine en su totalidad, lo cual añade dificultad a la tarea.

En comparación, los trabajos del estado del arte solo usan una parte del corpus y se limitan a clasificar las opiniones en positivas y negativas lo que da ventajas al desempeño del clasificador.

En la tarea de análisis de sentimiento en críticas de cine en español usando el corpus muchocine se han utilizado diferentes clasificadores. El método propuesto en este trabajo emplea la red neuronal back-propagation evaluada con la métrica accuracy.

Otros trabajos como el de Cruz [7] utilizan como clasificador el cálculo de la orientación semántica de forma supervisada y no supervisada. Por otro lado, Martínez [8] utilizan dos clasificadores Naive Bayes y SVM y evalúan accuracy, recuerdo y F1 – measure en ambos.

Se compara a detalle el método propuesto con otros trabajos. Se analiza el número de críticas utilizadas, el número de clases, el pre-procesamiento utilizado, los modelos de selección de términos, pesado de términos, el clasificador y las métricas utilizadas para la evaluación del clasificador. Se puede observar que el método propuesto utiliza 3878 críticas (el 100% del corpus), clasificadas en cinco clases (461 Muy positivo, 890 Positivo, 1253 Neutro, 923 Negativo y 351 Muy Negativo). Mientras que, Cruz [7] utilizan 200 críticas negativas y 200 críticas positivas, de forma similar, Martínez [8] utilizan 1351 críticas positivas y 1274 críticas negativas. Además, los dos trabajos comparados clasifican las críticas en positivas y negativas, a pesar de que el corpus muchocine 2008 está diseñado para cinco clases.

En cuanto a pre-procesamiento, el método propuesto utiliza tres tipos: sustituir acentos, eliminar stopwords y signos de puntuación. Los trabajos del estado del arte emplean diferentes tipos de pre-procesamiento: Cruz [7] tokenizan el corpus, mientras que Martínez [8] utilizan stopper y stemming.

Para el modelo de selección de términos y pesado tenemos que el método propuesto utiliza una bolsa de palabras en conjunto con el pesado booleano, para crear los vectores binarios que representan al corpus. En comparación, Martínez [8] utilizan diferentes pesados de términos (TF-IDF, TF, TO y BTO); de forma similar, Cruz [7] extraen bigramas a través de patrones morfosintácticos simples.

La red neuronal back-propagation, supera al clasificador de orientación semántica supervisada y no supervisada utilizado por Cruz [7]; así como al clasificador Naive Bayes utilizado por Martínez [8]. Pero no supera al clasificador SVM utilizado por Martínez [8]. La diferencia entre el accuracy de la SVM y la red neuronal back-propagation es mínima (0.0274). Sin embargo, es importante tener en cuenta que el método propuesto es el más completo porque utiliza todas las críticas que conforman el corpus y las clasifica en cinco clases.

6. Conclusiones

En este trabajo se desarrolló un método de AS usando la red neuronal back-propagation que permite clasificar las críticas de cine en español a nivel de documento en cinco clases. El método propuesto se aplica para su experimentación en el corpus muchocine del año 2008 obteniendo un accuracy de 0.8499 al clasificar de las críticas de cine en español. Clasificar las opiniones en cinco clases, aporta más información a los usuarios para tomar buenas decisiones.

La red neuronal back-propagation demuestra ser un clasificador que sobresale en el AS al obtener el sentimiento general de las opiniones emitidas en el corpus. La red neuronal back-propagation obtiene el mejor resultado en el proceso de clasificación de las opiniones utilizando: 3878 neuronas en las capas ocultas, divide la muestra en dos subconjuntos de datos una para entrenamiento, otra para prueba y utiliza una validación cruzada de 10 particiones.

En cuanto al pre-procesamiento, se demostró que al aplicar tres tipos de pre-procesamiento: sustituir acentos, eliminar stopwords y signos de puntuación; ayuda al clasificador a interpretar de formar correcta las críticas y a mejorar el desempeño y rendimiento del clasificador.

7. Trabajos futuros

Los trabajos futuros estarán dirigidos a desarrollar un corpus a partir de los resúmenes del corpus muchocine. Con la finalidad de mejorar el accuracy obtenido, se propone incorporar una red neuronal convolucional al método para clasificar en cinco clases el corpus original y el nuevo.

Referencias

1. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. New York (2008)
2. Buchanan, L., O'Connell, A.: Una breve historia de la toma de decisiones (2006)
3. Carrillo de Albornoz-Cuadrado, J.: Un Modelo Lingüístico-Semántico Basado en Emociones para la Clasificación de Textos Según su Polaridad e Intensidad. Departamento de Ingeniería de Software e Inteligencia Artificial, Facultad de Informática, Universidad Complutense de Madrid, Tesis de Doctorado (2011)
4. Agarwal, B., Mittal, N.: Prominent feature extraction for sentiment analysis. Springer (2016)
5. Martínez-Cámara, E., Valdibia, M., Teresa, M., Ureña, A.: Análisis de sentimientos. SINAI, Sistemas Inteligentes de Acceso a la Información (2020)
6. Jiménez, Z.: Salud, M., Martínez, E., Martín-Valdivia, M.T., Ureña-López, L.A.: Desafíos del análisis de sentimientos. Actas de las V Jornadas TIMM, pp. 15–18 (2014)
7. Cruz, F., Troyano, L., Enríquez, F., Ortega, J.: Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. Procesamiento del lenguaje Natural, 4, pp. 73–80 (2008)
8. Martínez-Cámara, E., Perea-Ortega, J.M., Martín-Valdivia, M.T., Ureña-López, L.A.: Técnicas de clasificación de opiniones aplicadas a un corpus en español. Departamento de Informática. Escuela Superior de Jaén. Universidad de Jaén, 47. pp. 163–170 (2011)
9. Trujillo, V.: Muchocine. http://www.muchocine.net/ (s.f.) (2012)
10. Henry-Winston, P.: Inteligencia artificial. Addison Wesley Iberoamericana (1992)

11. Hernández-Orallo, J., Ramírez-Quintana, M.J., Ferri-Ramírez, C.: Introducción a la minería de datos. Madrid: Pearson (2004)
12. Montejo-Raéz, A., Díaz Galiano M.C.: Participación de SINAI en TASS 2016. pp. 41–45 (2016)
13. Dubiau, L., Ale, J.M.: Análisis de sentimientos sobre un corpus en español: Experimentación con un Caso de Estudio. In: 14th Argentine Symposium on Articial Intelligence (2013)

ISSN 1870-4069

Detección de fraude interno utilizando metodologías de aprendizaje máquina

Ulises Jiménez Cardoso, José M. Sarmiento Chávez

Universidad Panamericana,
Facultad de Ingeniería,
México

{0231936, 0231932}@up.edu.mx

Resumen. Este trabajo presenta cómo el aprendizaje de máquina puede detectar comportamientos de empleados fraudulentos a partir de su huella digital. Para generar el modelo se utilizaron datos sintéticos y se probaron cuatro modelos de aprendizaje de máquina, a saber, regresión logística con gradiente estocástico, máquinas de soporte vectorial, bosques aleatorios y XGBoost. El modelo que mejor detecta a los "Insiders" para el conjunto de datos analizado fue el de bosque aleatorio que ha mostrado un buen desempeño para diferentes problemas.

Palabras clave: Fraude, fraude interno, aprendizaje máquina, detección de anomalías.

Insider Detection using Machine Learning Methodologies

Abstract. This document presents how machine learning can detect fraudulent employees behavior from their digital footprint. To generate the model presented, synthetic data simulated from real cases were used and tested in four different machine learning models: Logistic Regression with stochastic gradient optimization, Support Vector Machines, Random Forests and XGBoost. The best model to detect insiders for the given dataset was Random Forest, which shows a better performance for different issues.

Keywords: Fraud, insider trading, machine Learning, anomalies.

1. Introducción

Los empleados fraudulentos representan una amenaza sustancial para los servicios financieros por su conocimiento y acceso a sistemas internos y datos sensibles. El fraude interno es perpetrado por empleado, ex empleado, un proveedor o algún socio comercial que tiene o ha tenido acceso autorizado a la red,

los datos o los sistemas y que intencionalmente ha hecho un mal uso que afecte la confidencialidad, integridad o disponibilidad de los sistemas o información de la empresa.

De acuerdo a [2] el costo promedio de un incidente que involucra a un empleado o proveedor negligente es de $283,281 USD, el costo es más del doble $648,845 si el incidente involucra a un ladrón ("Thief") o impostor y finalmente los incidentes por Hackers le cuestan a las organizaciones en promedio $607,745.

La detección de la amenaza de fraude interno o 'insider' es un reto extremadamente complejo debido a que, primero los insiders hacen cosas no autorizadas con el uso de sus accesos legítimos; por lo tanto, controles de seguridad como detección de intrusos, cortafuego y anti-virus no los pueden detectar; después los ataques de insiders están asociados a distintas tipologías como por ejemplo, un empleado a disgusto que se roba propiedad intelectual o un empleado que filtra información.

Adicionalmente, los incidentes de fraude interno son realizados durante las horas de trabajo, lo cual hace que la detección de su comportamiento inusual se esconda dentro de una gran cantidad de datos de comportamiento usual.

El reto más grande que tiene cualquier modelo enfocado a la detección de crimen financiero es la poca disponibilidad de datos confirmados como insiders y que las empresas del sector bancario no siempre están dispuestas a compartir sus datos con terceros. Por lo tanto, se decidió utilizar una base de datos sintética desarrollada por "CERT® Insider Threat Carnegie Mellon University's Software Engineering Institute (SEI)"[5], dicho instituto se ha encargado de recolectar y analizar información de más de 700 tipos de crimenes cibernéticos de insider, desde seguridad nacional hasta espionaje; más aún, ha establecido lineamientos que las empresas pueden adoptar para mitigar el riesgo.

El objetivo del presente trabajo es detectar a los empleados propensos a realizar fraude a partir de los datos observables de su actividad, con el uso de herramientas de aprendizaje de máquina para finalmente reducir los costos asociados a este riesgo. Se tomo como base el trabajo realizado por [17] y se adicionaron variables de comportamiento como el tiempo que los empleados tienen actividad en los sistemas de la empresa. Adicionalmente se probaron diferentes técnicas de selección de variables y de construcción de datos en tipo panel.

La organización de este artículo es, en la Sección 2 se listan diferentes enfoques para la detección del fraude de insider como el uso de redes neuronales; en la Sección 3 se detallan los algoritmos de aprendizaje de máquina que se utilizaron para elaborar el modelo; en la Sección 4 se explica cómo se trataron los datos para el modelo y cómo se entreno el modelo; en la Sección 5 se presentan los resultados de cada uno de los diferentes algoritmos utilizados, y finalmente en la Sección 6 se expone cual fue el mejor de los algoritmos y se plantean futuras mejoras al análisis.

2. Trabajo relacionado

Se consultaron diferentes artículos con puntos de vista distintos sobre como abordar el problema, [17] genera 42 tipos de modelos distintos, usando como métrica la precisión (Verdaderos Positivos/(Verdaderos Positivos+Falsos Positivos)) de los modelos y el tiempo que tardaron en entrenarse, su mejor modelo fue un Random Forest para un conjunto de datos "oversampled" y una regresión logística con "boosting" para un conjunto de datos "undersampled". Otros artículos [19] abordan el problema al utilizar Redes Neuronales Profundas; mas aún, modela el comportamiento de los "insiders" con "Long Term Short Memory" (LSTM) y "Convolutional Neural Networks" (CNN), lo cual resulta en un área bajo la curva Característica Operativa del Receptor (AUROC) igual a 0.94.

Otros enfoques para la detección de este tipo de anomalías, [13] y [15] proponen utilizar el historial de comandos de un usuario y compararlo con el uso de comandos actual para encontrar un comportamiento anómalo y poder clasificarlo.

Cabe destacar, que el acercamiento al problema que se utilizó para este análisis, toma como base el propuesto por [17] en la ingeniería de algunos atributos, expande la construcción de variables disponibles como los tipos de sitios que se visitan durante la sesión del empleado en el sistema empresarial, cambia el tipo de balanceo de base de datos utilizando una metodología estratificada, prueba con otras técnicas de selección de variables y reduce el conjunto de modelos probados, además de utilizar una muestra fuera de tiempo para la selección de modelos, esto con el fin de simular un modelo productivo.

3. Materiales y método

3.1. Descripción general

El método que se propone es, a través de un algoritmo de aprendizaje, prevenir el tráfico ó robo de información privilegiada y el sabotaje, esto con el fin de ahorrar pérdidas financieras provocadas por aventajamiento o aprovechamiento alevoso de estrategias comerciales, pérdida de información o control de daños por exposición de información fuera de contexto, pudiendo costar en algunos sectores globales hasta $ 11.45 millones de dólares al año [4].

3.2. Materiales

Se utilizaron los datos "Insider Threat Dataset" generados de forma sintética por CERT Division, en colaboracion con ExactData, LLC, y patrocinados por DARPA I2O [1], en un ambiente de cómputo en la nube basado en el servicio de Google Colaboratory con las características de instancia base para Python 3 [3].

CERT Division ha generado diferentes conjuntos de datos sintéticos que modelan comportamiento de empleados fraudulentos, ya conocidos y definidos

en literatura como en [10]. Los datos contienen el comportamiento de cinco escenarios de posible fraude interno, el primero el que trabaja fuera del horario laboral, que utiliza un disco externo, sube información a "wikileaks.org" y deja la organización al poco tiempo; el segundo, un empleado navega en sitios web de búsqueda de trabajo, solicita trabajo a los competidores y antes de dejar la compañía, roba información; el tercero, un administrador de sistemas molesto que descarga un programa "Keylogger" para obtener las contraseñas de su supervisor, manda correos que causan pánico y deja la organización; el cuarto, un usuario se conecta a la computadora de otro en búsqueda de información que se envía a su correo personal, y finalmente, un miembro de un equipo que acaba de ser diezmado por despidos sube información a la nube para su uso personal.

Los datos contienen distintos conjuntos sintéticos de "insiders", este proyecto se enfocará en el conjunto 4.2, donde los "insiders" son más densos (existen mas casos) y se pueden encontrar casos de fuga (escenario 1), robo (escenario 2) y sabotaje (escenario 3). Por un período cercano a 2 años para cada usuario, se tiene la actividad de correos enviados con el texto del correo, la actividad de conexión de dispositivos de almacenamiento, los archivos a los que acceden y el tipo de archivos que son, su comportamiento web (páginas visitadas), sus tiempos de conexión y desconexión al sistema a través de sus PC's y una evaluación psicométrica mediante el instrumento OCEAN realizado por el departamento de RRHH. A pesar de que los datos son sintéticos, dichos datos fueron creados con la intención de emular lo más cercano los datos disponibles en la organización, tienen un tamaño aproximado de 1 TeraByte, los archivos no son totalmente estructurados, contienen datos vacíos, incorrectos y en distintos formatos. El esfuerzo más grande del presente trabajo fue justo en el procesamiento de datos, limpieza y preparación de los datos.

Para la creación de atributos, se agruparon los registros de actividad a partir del usuario, el mes y el año de actividad, con esto, se generaron atributos relacionados al conteo del tipo de sitios que visitaron (descargas, competidores, servicios de hosting, software...), la polaridad del sentimiento a partir del texto en páginas web y correos, la actividad de entrada y salida al sistema, psicométricos, género del usuario, actividad de uso de dispositivos de almacenamiento, acceso a archivos, el cargo del usuario dentro del organigrama de la empresa y su salario. Lo cual resultó en 16567 observaciones y 88 variables (features), algunas de estas se intersectan con los propuestos por [17], con la diferencia de la atomización de datos y la forma de calcularse.

3.3. Métodos de aprendizaje

Regresión lineal con gradiente descendiente estocástico Este método lo que busca es generar aprendizaje para la regresión logística utilizando el método de gradiente descendente estocástico, donde el gradiente de la función de pérdida se estima en cada observación y el modelo se actualiza en cada iteración con cierta función de aprendizaje. La regresión logística [6] estará dada por:

$$P(c|x) = \sigma(w^t \cdot x),$$
$$\sigma(t) = \frac{1}{1+e^{-t}},$$
$$Error = \frac{1}{2} * (y_{real} - y_{pred})^2.$$

Mientras que el método del gradiente descendiente estocástico vendrá dado por el siguiente algoritmo [7].

Para una tolerancia específica, o "hasta la convergencia":

- Escoger aleatoriamente un elemento del conjunto de entrenamiento (random shuffling).
- Actualizar los pesos:

$$w_i = w_i - \eta(y_{real} - y_{pred})^2, \tag{1}$$

con η como el coeficiente de aprendizaje.

Máquinas de soporte vectorial Este método lo que busca es una separación lineal de las observaciones dado un margen y un kernel, suponiendo que las clases son linealmente separables, lo que intenta buscar es un hiperplano que separe las clases. Mientras el problema se haga mas complejo, se llegara al punto de tener que encontrar la envolvente convexa más pequeña para cada clase. Cuando las clases no son separables, se introduce el "truco del kernel", donde se aumenta de dimensión el espacio de interés para encontrar la separación entre clases. El problema entonces se reduce a resolver la siguiente optimización:

$$min_\alpha \frac{1}{2} \sum_i \sum_j y_i y_j \alpha_i \alpha_j k \left\langle \phi(x_i), \phi(x_j) \right\rangle - \sum_i \alpha_i. \tag{2}$$

Sujeto a:

$$\sum_i y_u \alpha_i = 0,$$
$$0 \leq \alpha_i \leq C.$$

Al cual se llamará SVM Dual [14].

Random Forest Un bosque aleatorio (o Random Forest)[9], es una combinación de árboles de decisión tal que cada árbol depende de una muestra aleatoria tomada de los datos de entrenamiento, donde se trata de escoger árboles parcialmente no correlacionados, utilizando un método conocido como "bagging", el cual hace un muestreo con reemplazo, ya que los árboles son muy sensibles a cambios en los datos con los que fueron entrenados. De esta forma, cada árbol dará una predicción de clase funcionando como un ensamble, y así, la clase que haya sido seleccionada mayormente será la predicción de clase del modelo. Con el "bagging" se asegura que los árboles de decisión no estén correlacionados, y por lo tanto sus errores tampoco [18].

XGBoost Extreme Gradient Boosting Trees [12] es un método de ensemble basado en "boosting", el "boosting" [8] es un método, que también es un ensamble, para reducir sesgo y varianza en aprendizaje supervisado usando atributos que están ligeramente correlacionados con la variable independiente convirtiéndolos en una variable predictora fuerte mediante la manipulación de los pesos en el algoritmo y aprendizaje secuencial. XGBoost utiliza el método del gradiente descendente estocástico, paralelización de procesos y regularización para prevenir sesgo o sobreajuste [16].

4. Experimentación

En el proceso de experimentación 1 se generaron los algoritmos previamente citados, generando distintas muestras en los datos para el entrenamiento. Se generó una muestra fuera del tiempo, esto es, una muestra que parta el conjunto de datos por medio de la variable temporal, donde se dejarán los datos fuera del entrenamiento para probar el modelo en una intuición atemporal, para esto se utilizaron los últimos 3 meses. La siguiente partición de datos nace de forma aleatoria, pero para generarla, se hizo un muestreo estratificado a partir de la variable objetivo utilizando el método SMOTE [11] con el cual se pudo generar un "oversampling", ya que la base de datos está desbalanceada, se tiene un 92 % de casos negativos y el resto es positivo ("insiders" observados), generando un 70 % de casos de entrenamiento y el resto como prueba. Esto entonces nos genera tres muestras, la muestra de entrenamiento, la muestra de prueba (Out of Sample), y la muestra de prueba atemporal (Out of Time), los muestreos se pueden encontrar en 2.

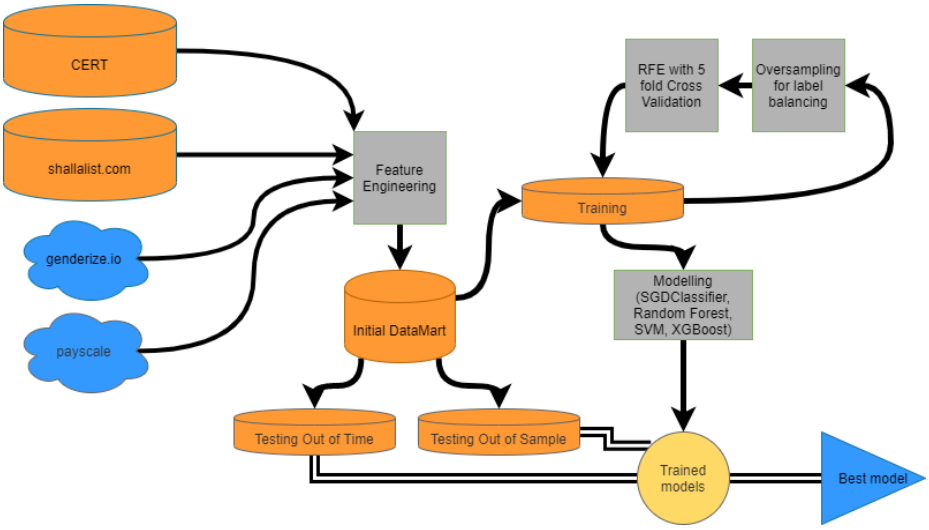


Fig. 1. Flujo de experimentación.

Is Insider \ Muestra	Entrenamiento (oversampled)	Out Of Sample	Out of Time
0	8807	3775	2557
1	8807	418	35

Fig. 2. Observaciones por muestra.

Para cada modelo, se utilizó una reducción de dimensión conocida como Eliminación Recursiva de Atributos, el cual es un proceso iterativo que ordena las variables a partir de la importancia en el modelo a partir de una métrica. Más aún, para cada modelo, se utilizó un método de validación cruzada y selección de hiperparámetros con métodos de búsqueda en red y aleatorio, utilizando la validación cruzada como medio para evitar el sobreajuste, y afinar los métodos para escoger la mejor configuración o arquitectura.

Una vez que se tuvo listo cada modelo, después de entrenado, se aplicó la predicción sobre las muestras "Out of Sample" y "Out of time", verificando la matriz de confusión para seleccionar los mejores modelos a partir de la precisión del modelo pero el ajuste fino, en este caso, será el "recall", ya que por la naturaleza de los datos y su desbalance, será una mejor métrica de selección, ya que los casos son muy escasos a pesar de la gran cantidad de información, lo que debería de afinar el "recall". Se muestra la matriz de confusión esta métrica.

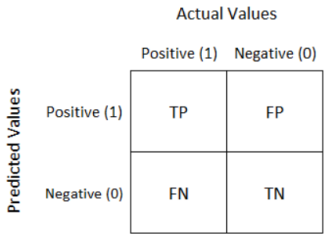


Fig. 3. Matriz de confusión.

De aquí, que nuestro "recall" estará dado por $\frac{TP}{TP+FN}$, esto nos dice de todas las clases positivas, ¿Cuántos predijo correctamente?, de esta forma, mejorar la selección aleatoria de sujetos y reducir el riesgo de tener un "insider" y no clasificarlo debidamente.

5. Resultados y discusión

Se pueden observar variables muy correlacionadas a partir de las etiquetas observadas, podemos observar en la figura 4 algunas de las variables más impor-

tantes, que se reflejan en relación con la etiqueta "isInsider", la cual nos marca las observaciones que han sido detectadas como "insiders".

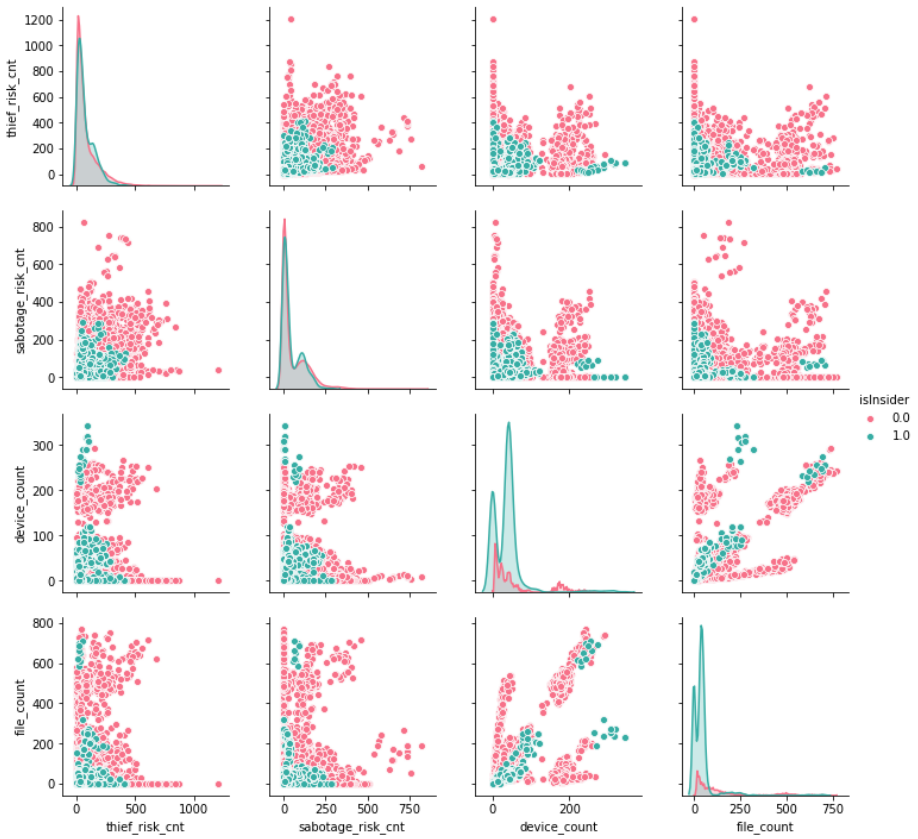


Fig. 4. Relación entre variables y etiqueta.

Si se hace una observación más detenida, podemos ver que los "insiders", mayormente hacen sus transacciones en horario laboral (8 AM a 5 PM), pero, observando a la gente que se queda a trabajar fuera de horario, los perpetradores son mayoría (Figura 5) aunque la muestra es mucho menor. También se puede observar que ningún perpetrador es supervisor (Figura 6) y los temas de sitios que visitan son muy específicos (Figura 7).

Se presentan los resultados de la comparación de los modelos, y se muestra que el método de Random Forest parece ser mejor para detectar perpetradores con la mejor precisión en ambas muestras (OoS [Figura 8] y OoT [Figura 9]), mientras que SVM pareciese tener un buen desempeño (Figura 11), es engañoso, ya que al ver a detalle el reporte y la matriz de confusión, podemos ver que en

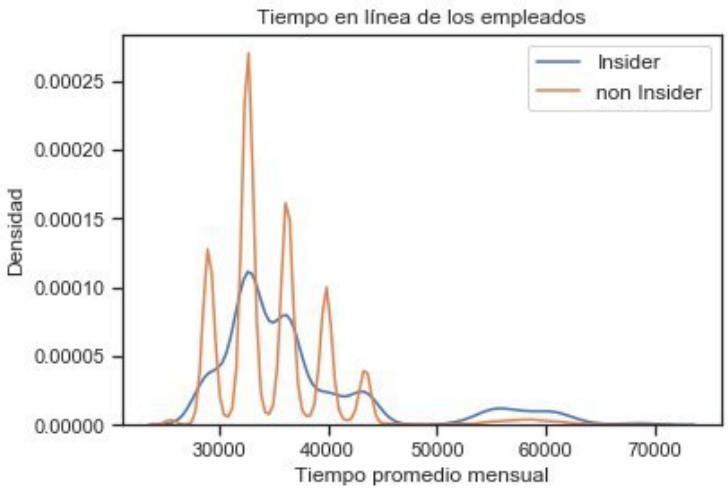


Fig. 5. Conteo de perpetradores y horas de trabajo.

	isSupervisor	
	count	sum
isInsider		
0.0	15139	1139.0
1.0	1428	0.0

Fig. 6. Conteo de supervisores.

	hacking	leaks	spyware	
	sum	sum	sum	count
isInsider				
0.0	0.0	0.0	0.0	15139
1.0	19.0	80.0	29.0	1428

Fig. 7. Sitios que visitan los perpetradores.

la muestra OoT (Figura 12) no funciona, y por tanto quedará descartado en el análisis.

	Out of sample			
Métrica \ Modelo	SGD	Random Forest	SVM	XGBoost
Precision	0.93	0.98	0.96	0.98
Recall	0.9	0.98	0.96	0.98

Fig. 8. Out of sample.

	Out of time			
Métrica \ Modelo	SGD	Random Forest	SVM	XGBoost
Precision	0.99	0.99	0.97	0.99
Recall	0.83	0.99	0.99	0.92

Fig. 9. Out of time.

```
[[2557    0]
 [  35    0]]
```

Fig. 10. Matriz de confusión OoT de SVM.

Es de importancia, en la selección de las muestras para analizar el modelo, utilizar una muestra fuera de tiempo, ya que ha sido un factor para descartar el modelo por su bajo desempeño.

6. Conclusiones

Para este conjunto de datos, pareciese ser que los mejores modelos se basan en árboles de decisión, teniendo un desempeño superior a las SVM y al SGD, compartiendo parte de este resultado con [17]. Se propone extender este trabajo reduciendo la granularidad de los datos y tal vez con datos exógenos, como factores sociodemográficos de los usuarios, situación en burós de crédito y actualizar con cierta periodicidad los artefactos psicométricos de caracterización de la personalidad.

Se puede ver que el comportamiento de los "insiders" parece reflejarse en su huella digital y en su comportamiento en las oficinas, siendo que visitan sitios específicos y algunos se quedan a utilizar el equipo de trabajo después de la hora regular de salida, mientras que ninguno ha tenido un puesto importante dentro del organigrama de la empresa.

	precision	recall	f1-score	support
0.0	0.96	1.00	0.98	3775
1.0	1.00	0.62	0.77	418
accuracy			0.96	4193
macro avg	0.98	0.81	0.87	4193
weighted avg	0.96	0.96	0.96	4193

Fig. 11. Reporte de clasificación SVM OoS.

	precision	recall	f1-score	support
0.0	0.99	1.00	0.99	2557
1.0	0.00	0.00	0.00	35
accuracy			0.99	2592
macro avg	0.49	0.50	0.50	2592
weighted avg	0.97	0.99	0.98	2592

Fig. 12. Reporte de clasificación SVM OoT

Desde el punto de vista de aprendizaje máquina, se podría intentar aprendizaje por refuerzo y aprendizaje profundo, que está fuera del contexto de este proyecto. Aprendizaje profundo nos podría dar la ventaja de detectar a los perpetradores con una granularidad de datos más pequeña, pues se podrían utilizar como observaciones datos no agrupados como se hizo en este proyecto, y generar más varianza en la información, ya que entendido el subtexto de la agrupación mensual, se tendría que calificar las observaciones con una regularidad de un mes mientras que en el otro caso se podrían calificar en tiempo real.

Finalmente, basado en la experiencia profesional de los autores, estos consideran que el crimen financiero debe de ser analizado desde diferentes perspectivas y un único modelo no mitiga el riesgo totalmente, un programa de prevención de dicho riesgo debe contener modelos supervisados para la detección de tipologías, como las presentadas en este trabajo; modelos no supervisados, para la detección de nuevas tipologías; un modelo del comportamiento de clientes y empleados a través de tiempo, y uno que modele las relaciones de clientes o empleados con otros agentes.

Referencias

1. Insider threat test dataset (Nov 2016), `https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099`
2. Data breaches caused by insiders increase in frequency and cost (April 2018)
3. Google colab specifications (Mar 2020), `https://colab.research.google.com/drive/1_x67fw9y5aBW72a8aGePFLlkPvKLpnBl`
4. Insider threat cost 2020 (Jan 2020), `https://www.proofpoint.com/us/resources/threat-reports/2020-cost-of-insider-threats`

5. Software engineering institute (Mar 2020), `https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=21232`
6. Agresti, A.: Categorical data analysis, vol. 482. John Wiley & Sons (2003)
7. Bottou, L.: Online algorithms and stochastic approximations. In: Saad, D. (ed.) Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK (1998), `http://leon.bottou.org/papers/bottou-98x`, revised, oct 2012
8. Breiman, L.: Bias, variance, and arcing classifiers. Tech. rep., Tech. Rep. 460, Statistics Department, University of California, Berkeley ... (1996)
9. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
10. Cappelli, D.M., Moore, A.P., Trzeciak, R.F.: The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes (Theft, Sabotage, Fraud). Addison-Wesley (2012)
11. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357 (2002)
12. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
13. Davison, B.D., Hirsh, H.: Predicting sequences of user actions. In: Notes of the AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis. pp. 5–12 (1998)
14. Deisenroth, M.P., Faisal, A.A., Ong, C.S.: Mathematics for machine learning. Cambridge University Press (2020)
15. Lane, T., Brodley, C.E., et al.: Sequence matching and learning in anomaly detection for computer security. In: AAAI Workshop: AI Approaches to Fraud Detection and Risk Management. pp. 43–49. Providence, Rhode Island (1997)
16. Morde, V.: Towards Data Science XGBoost algorithm: Long may she reign (2020)
17. Noever, D.: Classifier suites for insider threat detection. arXiv preprint arXiv:1901.10948 (2019)
18. Yiu, T.: Towards Data Science understanding random forest (2020)
19. Yuan, F., Cao, Y., Shang, Y., Liu, Y., Tan, J., Fang, B.: Insider threat detection with deep neural network. In: International Conference on Computational Science. pp. 43–54. Springer (2018)

ISSN 1870-4069

Determinación de factores involucrados en la toma de decisiones conductuales mediante minería de datos

Carlos Hiram Moreno Montiel[1], José Luis Bernal López[2], Ezequiel Alpuche de la Cruz[2], Benjamín Moreno Montiel[3], Marisol Sandoval Ríos[4]

[1] Universidad Tecnológica de México,
Sistemas Computacionales,
México
[2] Tecnológico de estudios superiores de Chimalhuacán,
México
[3] Universidad Autónoma Metropolitana Iztapalapa,
Ingeniería Eléctrica,
México
[4] Universidad del Valle de México,
Escuela de Medicina,
México

{hiramoreno, jolubelo1, alpuchedelacruz,
opelo1209, msrios1801}@gmail.com

Resumen. El desarrollo cultural y social en las principales ciudades del mundo genera comportamientos diversos en los individuos, influenciados por múltiples factores de decisión en su vida. Las combinaciones posibles de decisiones son del orden de millones, por lo que analizarlos requiere de altos tiempos, por este motivo deben modelarse en sistemas computacionales. En el presente artículo se analizaron los principales factores reportados en la literatura, que influyen en la toma de decisiones de individuos. Posteriormente, con ayuda de una aplicación móvil se generó un repositorio de datos, obtenidos por encuestas donde se determinaron que factores similares son los indicativos de éxito o fracaso. Estos se obtuvieron mediante un criterio de relevancia, el cual extrae los altos grados de coincidencia mediante minería de datos. Los resultados obtenidos muestran que la toma de decisiones de los individuos está regida por aspectos, emocionales y costumbres sociales, antes que de conocimiento, cooperación o psicológicos.

Palabras clave: Factores, toma de decisiones, minería de datos.

Determination of Factors involved in Behavioral Decision-Making through Data Mining

Abstract. Cultural and social development in the main cities of the world generates diverse behaviors in individuals, influenced by multiple decision factors in their lives. The possible combinations of decisions are of the order of millions, so analyzing them requires long times, for this reason they must be modeled in computer systems. In this article, the main factors reported in the

literature, which influence the decision-making of individuals, were analyzed. Subsequently, with the help of a mobile application, a data repository was generated, obtained by surveys where it was determined that similar factors are indicative of success or failure. These were obtained through a relevance criterion, which extracts the high degrees of coincidence through data mining. The results obtained show that the decision-making of individuals is governed by emotional aspects and social customs, rather than knowledge, cooperation or psychological aspects.

Keywords: Factors, decision making, data mining.

1. Introducción

Un aspecto central para la sociedad gira en torno a la manera en cómo las personas toman decisiones en su vida cotidiana, razón por la cual esta toma de decisiones se vuelven sin duda la materia prima de la cual están formadas las sociedades. Sin embargo, con el incremento del número de personas, se crea también un mayor número de diferentes factores con los que eventualmente tomará decisiones. Se ha tratado de realizar de manera experimental recolección de información, en la que se muestre un importante flujo de decisiones que afectan el comportamiento de las personas y de las mismas sociedades. Sin embargo, al establecer modelos con encuestas, se requiere de mucho tiempo y los análisis no llegan hacer tan confiables, debido a que se necesita de un tiempo largo de estudio, en el cual las decisiones de las personas varían en respuesta a los propios cambios de la sociedad misma. Mediante la utilización de los sistemas de cómputo, la recopilación de datos para un estudio de este tipo se puede hacer de manera más directa y confiable [1], pues cada uno de los factores que influyen en estas decisiones, tiene elementos directamente relacionados con el comportamiento de las personas; además la manera de representar cómo se comportan los individuos al tomar decisiones, sólo se obtiene de manera aislada (de manera individual) y no se encuentra de manera conjunta (como sociedad). Es importante mencionar que el individuo mismo es un sistema complejo [2], el cual está cambiando en todo momento al evolucionar para su mejora [3].

Ciertos autores, han modelado la toma de decisiones considerando las políticas públicas y estrategias empresariales para ver los aspectos de prejuicios y de falta de interés por el trabajo a través del desempeño de las personas. También, existen estudios aislados en donde se demuestra que la complejidad de la conducta humana se basa en los intereses propios de las personas incluyendo factores de egoísmo y de superación. Asimismo, evidencias experimentales [4] demuestran que los estímulos sociales y económicos son los que influyen prioritariamente en que las personas tomen decisiones de manera más objetiva y exitosa. Sin embargo, éstos son estudios individuales sin tomar en cuenta otros factores de manera conjunta.

De acuerdo con lo antes descrito, en el presente estudio se desarrolla un modelado computacional, y, a través de la minería de datos, se identifican diversos factores que influyen en la toma de decisiones de los individuos. Asimismo, se realiza un análisis de datos mediante una aplicación móvil, la cual recaba información por medio de encuestas realizadas a individuos de diferentes edades y características [5]. El principal tipo de información obtenida es sobre los principales factores reportados en la literatura,

como aquellos involucrados cotidianamente en la toma de decisiones que realizan las personas en su vida diaria. La información se almacena en la nube, desde la cual se extrae a un servidor para posteriormente generar una matriz de toma de decisiones con los datos obtenidos.

2. Trabajos relacionados

2.1. Obtención de los factores involucrados en la toma de decisiones (OFITD)

Se realizó una búsqueda en la literatura de los factores reportados de ámbitos variados para realizar un análisis [6]. Los estudios indican que el individuo que participa en una organización en sociedad se encuentra inmerso, en esencia, en una cultura que lo imbuye de elementos simbólicos a través de los cuales ve la realidad circundante, al hacer una compilación de las características que le son inherentes a los individuos en el proceso de toma de decisiones por medio de: sentimientos, emociones, creencias, entre otras, que coexisten con su racionalidad. En [7], se muestra que es necesario tomar en cuenta las emociones y predisposiciones humanas, para lo cual ofrece estados conocidos como emociones inconfundibles, entre los cuales están: el enojo, odio, culpa, vergüenza, orgullo, gusto, pesar, alegría, aflicción, envidia, malicia, indignación, celos, desprecio, disgusto, miedo y amor. Los anteriores deben distinguirse de los "factores viscerales" tales como el dolor, hambre y sueño, los cuales son de origen fisiológico. Además del egoísmo existen muchas otras actividades que están gobernadas por "espíritus animales" como lo nombran en [8] y los estímulos que mueven a las personas cuando persiguen cierto tipo de intereses. Estos aspectos a veces nos paralizan, mientras que en otras ocasiones nos revitalizan y nos llenan de energía, haciendo que superemos nuestros miedos e inseguridades como:

a) La confianza, la cual tiene dos acepciones, en primer lugar, se puede entender como seguridad o fe, es decir se trata del fanatismo donde la racionalidad es mínima. En segundo lugar, está la equidad que muestra que los individuos al ser condescendientes con los demás, se sienten satisfechos con lo realizado [9].
b) Las historias o relatos, muestra que gran parte de las motivaciones de las personas proceden de la historia de sus propias vidas, una historia que se cuentan ellas mismas y que constituyen la base de sus motivaciones [8].
c) Corrupción y mala fe, los seres humanos ante ciertas situaciones tenemos una tendencia que nos llevan a caer en comportamientos antisociales y dificultades.
d) La ilusión monetaria se produce cuando las decisiones están influenciadas por cantidades de dinero.

La evidencia experimental, también muestra que un importante tipo de preferencias sociales son de carácter malicioso y envidioso [10]. Por ejemplo, el egoísmo ha desempeñado un papel esencial en el proceso de selección natural a lo largo de la historia humana [11]. Debe agregarse también, ciertos rasgos característicos del actor, como la codicia, la envidia, egoísmo, hedonismo [12], el miedo, el pánico, indiferencia, así como precaución, desinterés, duda y sospecha, confianza, entusiasmo, empuje y euforia [13], entre otros; pero también es necesario destacar que el actor realiza

elecciones satisfactorias [14] no óptimas ni maximizadas de beneficio, lo hace en ocasiones a través de un efecto imitación, es decir, siguiendo lo que han hecho los líderes del mercado, sin olvidar el entramado simbólico de relaciones: lo ritual y lo ceremonial, así como el deseo de trascender, dejar huella, poder actuar dentro los márgenes de libertad y controlar las zonas de incertidumbre en la organización [15].

De esta forma, los factores que se consideran en este trabajo, como los mínimos indispensables (sin ser exhaustivos), que están involucrados en la toma de decisiones individuales son los que aparecen en la Fig. 1, derivados de los aportes teóricos mencionados anteriormente y con los cuales se procede a la elaboración un modelo computacional.

3. Modelado del sistema propuesto

El modelo propuesto, se realizó mediante encuestas donde se elaboraron preguntas de los 36 factores contemplados en la Fig. 1. Algunas de las preguntas utilizadas se tomaron y adaptaron de la tesis de licenciatura "Sistema de análisis para determinar qué características dentro de los sistemas complejos de humanos ayudan para evolucionar a los individuos." Se siguió un patrón de escala Likert por ser variables cualitativas como se muestra en [16]. Cada una de las preguntas se orientó a obtener información de la muestra poblacional en donde se utiliza un lenguaje coloquial y personal, ya que así lo requiere el nivel de muestra al que va dirigido. Algunas de las preguntas se muestran en la Tabla 1.

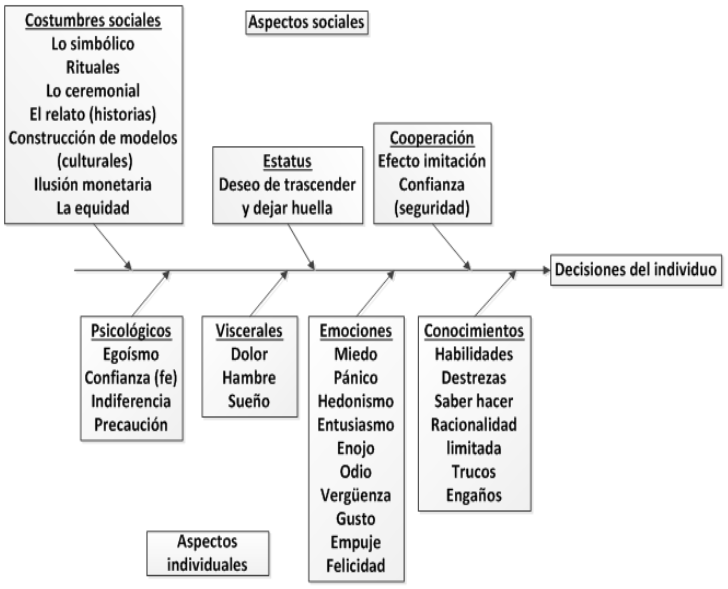


Fig. 1. Factores que influyen en las decisiones conductuales del individuo.

Tabla 1. Preguntas incluidas en las encuestas aplicadas, para la determinación de los factores involucrados en la toma de decisiones.

Preguntas	Factor involucrado
Cuando tomo una decisión, las opiniones de mis familiares, amigos, pareja o vecinos tienen mucha importancia para mí.	(Construcción de modelos)
Mis creencias religiosas me dictan lo que debo elegir.	(Rituales)
Las historias que me han platicado a lo largo de mi vida me indican lo que es bueno o malo para mí.	(El Relato)
Todas las personas deberían contar con las mismas oportunidades para poder tomar las mejores decisiones que los lleve a su bienestar.	(La equidad)
Me gusta elegir toda acción que me permita influenciar en los demás.	(Estatus)
Me encantaría que las siguientes generaciones me recuerden por lo que hice.	(Deseo de trascender)
Siempre hay que pensar en la retribución económica cuando tomamos una decisión.	(Ilusión monetaria)
Para mí es muy importante que mis excompañeros de trabajo o de escuela recuerden mi nombre y lo que me gustaba hacer.	(Dejar huella)
Cuando hay que evaluar lo que se debe hacer para mejorar mi entorno los demás confían en que haré buenas propuestas.	(Confianza)
Me gusta que los demás valoren mis opiniones cuando hay juntas vecinales o de trabajo.	(Cooperación)
Me gusta que los demás imiten lo que hago, digo o pienso.	(Efecto imitación)
Cuando los demás apoyan lo que propongo quiere decir que soy muy bueno para tomar decisiones	(Seguridad)
Al elegir algo sólo me importa mi beneficio.	(Egoísmo)
Confío mucho en mí mismo al tomar decisiones.	(Confianza)
Al tomar una decisión me da igual las repercusiones que pueda tener en los demás.	(Indiferencia)
Siempre soy precavido con mis decisiones.	(Precaución)
Es muy importante no tomar decisiones cuando tengo hambre.	(Hambre)
Soy bueno eligiendo, aún cuando no he dormido bien.	(Sueño)
El dolor físico no influye en mis elecciones importantes.	(Dolor)
Al estar enojado tomo decisiones que me impulsan a salir adelante.	(Enojo)
Siempre pienso en la comodidad y el placer que me pueden traer al momento de elegir algo.	(Hedonismo)

Preguntas	Factor involucrado
Cuando estoy feliz tiendo a tomar buenas decisiones.	(Felicidad)
Cuando elijo algo siempre me da miedo que algo salga mal.	(Miedo)
Considero que cuento con las habilidades necesarias para elegir las mejores opciones al momento de tomar mis decisiones.	(Habilidades)
La vida me ha hecho una persona experimentada, por lo que los demás pueden acudir a mí para darles buenos consejos.	(Saber hacer)
No importa la decisión que tomes, si sabes modificar las normas para que todo sea a tu favor.	(engaños)
Los demás me ven como una persona que analiza muy bien la situación para tomar decisiones.	(Racionalidad)

Para recabar los datos, se utilizó la aplicación móvil desarrollada en App Inventor para un sistema operativo Android, la cual aloja los datos en la nube y posteriormente se extraen a un servidor para el análisis. La aplicación móvil realiza las preguntas, las cuales tienen solo cuatro respuestas posibles como las utilizadas en la escala Likert.

Sus respuestas indican el grado de relevancia determinado por: No Aplica o en desacuerdo (indica que no es relevante para tener éxito), Éxito o totalmente de acuerdo (es importante la cuestión para el éxito en su vida), Fracaso o Totalmente en desacuerdo (en donde implica que esa cuestión la asocian al fracaso en su vida), Ocasiones (indica que en ocasiones existe una relación de la cuestión con el éxito en su vida).

Cada uno de los datos obtenidos a partir de las respuestas en cuanto al Éxito, Fracaso, No Aplica o en Ocasiones, se almacenó en tablas individuales (por persona).

Después, se realizó una clasificación [18] para buscar las coincidencias en cada una de las tablas obtenidas dentro de una matriz general y se procedió a realizar el análisis para mostrar la cantidad o porcentaje de coincidencias que existen entre cada una de las tablas.

A continuación, se muestra a detalle cada una de las etapas para la realización del modelado.

3.1. Minería de datos para el sistema de toma de decisiones propuesto

La minería de datos es el proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias al examinar grandes cantidades de datos, en grandes volúmenes de información, transformado sin duda el proceso de análisis de datos, orientándolo hacia determinadas técnicas especializadas [19].

Para desarrollar el modelado dentro de este trabajo, es importante tomar en cuenta los siguientes procesos: Proceso de Seleccionar (Selecting), Explorar (Exploring), Modificar (Modifying), Modelizar (Modeling) y Valorar (Assessment) grandes cantidades de datos con el objetivo de descubrir patrones desconocidos que puedan ser utilizados [20]. El objetivo principal, es determinar cómo influyen en la toma de decisiones de la vida diaria del individuo desde diversos aspectos analizados en las secciones anteriores del artículo.

3.2. Selección de los datos

Para la selección de los datos de este estudio, se trabajó en dos instituciones educativas: en la Universidad Autónoma Metropolitana unidad Iztapalapa (UAMI) y, en el Tecnológico de Estudios Superiores de Chimalhuacán TESCHI. Las edades de los entrevistados oscilaban entre los 21 y los 39 años, que es la edad del alumnado de licenciatura en ambas instituciones y no hubo distinciones en cuanto al género de los individuos. En ambas instituciones no se tiene un estudio socioeconómico generado hasta el momento, por lo que la variabilidad de los resultados podría resultar beneficiosa en el presente estudio. Se realizaron un total de 1700 encuestas entre ambos planteles con la ayuda de la aplicación móvil.

Para seleccionar los datos obtenidos, se generó una matriz de toma de decisiones individuales y sociales (MTDIS). Esta matriz, contiene los aspectos recabados en la OFITD y se van asignando dentro de la MTDIS para las primeras 3 filas los aspectos individuales y, de la fila 4 a la fila 7, los aspectos sociales como se muestra en la Fig. 2.

Por cada uno de los registros recabados en la encuesta, se obtienen 4 respuestas posibles como se puede observar en la sección del modelado propuesto Cada uno de estos datos se va llenando dentro de la matriz como se observa en la Fig. 2. Para los valores No Aplica se utiliza el acrónimo NA, para el Éxito se utiliza el acrónimo E, para el Fracaso el acrónimo F y para Ocasiones el acrónimo O. De esta manera, se observa cómo se van llenando los valores de las respuestas para cada una de las preguntas realizadas en el estudio.

Costumbres sociales	simbólico NA	ritual E	Ceremonial E	Relato F	Modelos culturales E	Ilusión E	Equidad F		
Estatus	Trascendencia E	Dejar huella NA							
Cooperación	Imitación NA	Equipos E							
Psicológicos	Egoísmo F	Confianza O	Indiferencia F	Precaución NA	Fe E				
Viscerales	Dolor F	Hambre O	Sueño F						
Emocionales	Miedo F	Pánico NA	Hedonismo NA	Entusiasmo E	Enojo F	Odio F	Vergüenza O	Gusto O	Felicidad O
Conocimientos	Habilidades E	Destrezas F	Saber NA	Racionalidad O	Trucos O	Engaños NA			

Fig. 2. Ejemplo de una tabla con las respuestas obtenidas por una persona en la encuesta realizada con los acrónimos pertinentes.

3.3. Limpieza y pre-procesamiento para determinar los factores involucrados en la toma de decisiones de éxito o fracaso en los individuos

En la matriz, se va almacenando cada una de las respuestas que dan los individuos y se hace la limpieza, mediante la agrupación de las respuestas similares tanto de cada una de las tablas (individuales) como de las tablas en conjunto, de manera sucesiva Fig. 3. Mediante esta agrupación de datos se va analizando el porcentaje de respuestas [21], dependiendo del número de valores de: Na, E, O, F.

Para cada una de las tablas que se generen, se analiza el por ciento de coincidencias que se presente por cada renglón, por ejemplo, si se tiene un porcentaje de más del 80% se considera como predominante esa respuesta y el valor del renglón seria NA, E, F, O según sea el caso. Se hace lo siguiente para cada uno de los renglones y sucesivamente para todas las tablas recabadas.

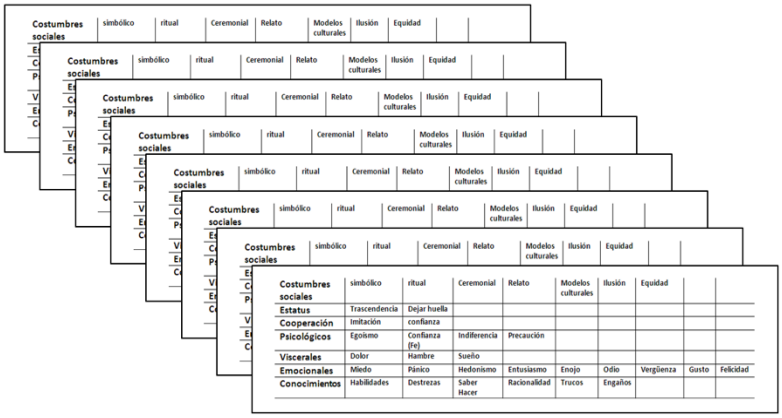


Fig. 3. Representación de la matriz general para determinar los patrones en la toma de decisiones de las personas.

Una vez analizadas todas las encuestas, se generó una matriz tridimensional que concentra el análisis de todas las encuestas tomadas a lo largo del estudio. Al obtener esta matriz general, se buscan los factores similares asociados entre cada una de las MTDIS, de esta manera se van separando cada una de las partes de la matriz, dependiendo del grado de coincidencias que se presente entre ellas [22][23]. Cada una de las tablas que coinciden (TC), dependiendo de los factores similares, representan vertientes en las decisiones Fig. 4. Esta tabla de decisiones determina si las respuestas son: NA, E, F, O, y se generan todas las vertientes similares para todas las tablas generadas Fig. 4.

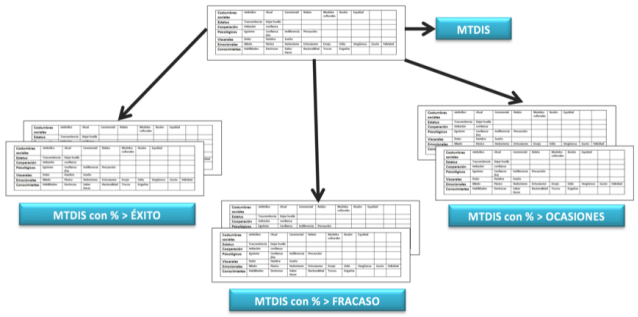


Fig. 4. Derivación de las vertientes en el análisis de coincidencias para determinar los porcentajes obtenidos en los factores para la toma de decisiones.

3.4. Elección del algoritmo de minería de datos

Para obtener las tablas de decisiones generales, el sistema tiene que desarrollar el algoritmo siguiente:

1. Se generan todas las MTDIS de todos los datos recabados.

2. Se analiza cada una de las respuestas que se presentan por persona.
3. Se agrupan y asocian cada una de las respuestas similares para determinar si hay relación entre (NA, E, F, O), de la persona.

Se extienden las vertientes de las TC y se determinan los porcentajes de los factores en una base de datos; dichas tablas serán alimentadas y actualizadas según los datos recibidos.

4. Resultados y discusión

4.1 Aplicación del algoritmo para la evaluación del modelado en la matriz de decisiones

Como primer resultado, se obtuvo la generación de las tablas para el análisis de los datos, tablas que anteriormente no se habían elaborado y se trabajaban de forma manual. En la Tabla 2, se muestran datos experimentales a partir del análisis de 100 y 250 encuestas de manera manual; para el caso de las 500 y 750 encuestas, los valores reportados, son una estimación correspondiente al análisis de ese volumen de datos. Como se puede observar, la recolección de los datos del sistema es mucho más rápida que la recolección que se realiza de manera manual.

Tabla 2. Tiempos de análisis tradicional y con el sistema de las encuestas realizadas.

Número de encuestas	Tiempo de análisis (min) tradicional	Tiempo de análisis (min) con el sistema
100	350	3
250	800	3
500	1600 (+20Hrs)	8
750	2500 (+40Hrs)	10

De acuerdo con el comparativo de la Tabla 2, se sugiere ampliamente realizar la recopilación de este tipo de información haciendo uso de las herramientas tecnológicas de la información y comunicación, pues como se puede observar, el análisis manual de 750 encuestas y su registro en MTDIS, demora un tiempo aproximado de 40 horas; mientras que, con el sistema, el tiempo estimado del mismo análisis es de 10 minutos, incluyendo la generación de las MTDIS.

4.2. Evaluación del algoritmo de clasificación del sistema para determinar el porcentaje de éxito o fracaso en la toma de decisiones

Al elaborar las MTDIS individuales, se analizaron las coincidencias que existían entre ellas y se formaron conjuntos de MTDIS con las cadenas resultantes del análisis. De manera general, se transformaron los datos en un formato específico requerido por el algoritmo (Morales, 2013), formando cadenas de combinaciones por tablas de cada una de las encuestas realizadas para los 7 aspectos y las combinaciones. Los datos se

transforman de la siguiente manera para su manejo: Costumbres sociales = 1, Estatus =2, Cooperación=3, Psicológicos = 4, Viscerales = 5, Emociones = 6, Conocimientos = 7. También, se transforman los valores de las respuestas posibles con los siguientes valores NA = 10, E = 11, F = 12, O = 13. Con esta transformación se obtienen las cadenas para un individuo de la siguiente manera (para el ejemplo de la Fig. 2):

1101111121111122111031011412131210115121312612101011121213131371112
10131310

Esta transformación de los datos se realiza de manera automática en el sistema y se tienen las cadenas numéricas de todos los individuos encuestados. En el siguiente ejemplo, extraído de la base de datos de las combinaciones, se observa el porcentaje de coincidencias de 10 registros con la transformación correspondiente del análisis de los datos, Fig. 5.

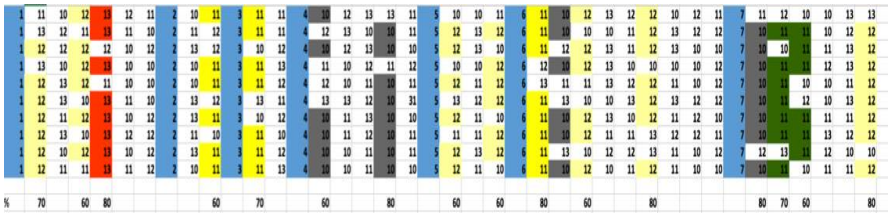


Fig. 5. Porcentajes obtenidos sobre las cadenas numéricas de las coincidencias en los factores involucrados en la toma de decisiones.

Se puede observar en la Fig. 5, aunque son solo 10 registros, que existe una tendencia notoria de las respuestas proporcionadas por los individuos. En algunos factores, la tendencia es más significativa, alcanzando hasta un 80 % de los 10 registros mostrados, Mientras que en otros casos el porcentaje de coincidencias no es representativo, por lo que no se observa en la Fig. 5. Por otra parte, se observa que para el índice 12, que está relacionado con los factores que los individuos asocian al fracaso, se obtiene un 80 %, al igual que con los factores relacionados al conocimiento en su campo engaño y también para el factor emocional en su campo odio. De igual forma, se puede ver que aparece el valor 11 asociado al éxito con un 80 % para el factor emocional en su campo miedo y, el factor cooperación en su campo imitación como un elemento del éxito de los individuos, se registra un 70 %

En la Tabla 2 se muestra con los porcentajes más altos, determinados por el sistema del modelado de factores que en sus campos obtuvieron NA, E, F, O. Asimismo, se puede observar que los factores de costumbres sociales tuvieron un mayor porcentaje de éxito, alcanzando hasta un 87% de coincidencias en el campo de los "relatos", el campo de "los rituales" y el campo de "la ilusión". Se puede observar que en ocasiones un "ritual" que realice una persona, le hace pensar que le puede ir mejor en la vida, al igual que los relatos que escucha la gente son importantes para tener éxito en su vida. También le resulta muy importante el campo de la "ilusión" (que es aquello que esperan lograr los individuos), este campo obtuvo un 87 % de coincidencias con respecto a las decisiones de los individuos para tener éxito. El rango de edades que obtuvieron este porcentaje para el factor de costumbres sociales es de los 21 a los 38 años, lo que indica que la mayoría de las personas tienen la misma percepción de ese tipo de factores cuando toman decisiones.

Se puede observar en la Tabla 2 que en el caso del factor de estatus en los campos "dejar huella" y la idea de "trascender", resulto la elección de "no aplica", por lo que no es importante para las decisiones que toman las personas en este estudio, obteniendo un 70 y 80% respectivamente. En este caso, el rango de edades para la respuesta es mucho menor ya que va de los 22 hasta los 32 años. En el caso del factor de cooperación, se puede observar quien tuvo un porcentaje de 75 de éxito en su campo de "imitación", en donde las personas tratan de imitar a otras que consideran exitosas para tomar decisiones en su vida y que en muchas ocasiones les resulta adecuado. El rango de edad para este caso es más amplio, pues va de los 22 a los 36.

Tabla 2. Porcentaje (%) obtenido de los diferentes factores del análisis propuesto.

Factor Asociado	Campo	Valores	% Obtenido	Edades (Años)
COSTUMBRES	Ritual	Éxito	85	21-38
SOCIALES	Relato	Éxito	78	27-38
	Ilusión	Éxito	87	21-39
ESTATUS	Trascendencia	No	80	21-32
	Dejar huella	Aplica	70	22-33
		No		
COOPERACIÓN	Imitación	Aplica	75	22-36
	Equipos		70	23-34
		Éxito		
PSICOLÓGICOS	Egoísmo	Ocasiones	81	27-38
	Confianza		80	23-36
	Indiferencia	Fracaso	80	21-37
		Éxito		
		Fracaso		
VISCERALES	Dolor	Fracaso	84	21-36
	Hambre	Fracaso	85	23-34
EMOCIONALES	Miedo	Éxito	75	22-37
	Entusiasmo	Éxito	82	21-36
	Felicidad	Éxito	78	21-38
CONOCIMIENTOS	Habilidades	No	70	21-33
	Trucos	Aplica	78	21-36
	Racionalidad	Éxito	70	21-34
	Engaños	No	83	24-37
		Aplica		
		Éxito		

Para el factor de las cuestiones psicológicas, se puede observar que los campos de "Egoísmo" e "indiferencia", obtuvieron un alto índice de "fracaso" en la toma de decisiones, alcanzando un 81 y 80% respectivamente. En estos casos, los rangos de personas que coinciden en la mayoría de las objeciones van desde los 27 a los 37 años

para el primer campo y, para el segundo, van de los 21 a los 37 años de edad. Sin embargo, en el mismo factor, el campo de "Confianza" obtuvo un 80% de respuestas, que son consideradas como tener éxito.

En los Factores viscerales como son el "dolor" y "hambre", estos tuvieron una paridad ya que ambos obtuvieron un porcentaje del 84 y 85% respectivamente, para las cuestiones que determinan el fracaso en las personas. El rango de edades va desde los 22 hasta los 37 años en promedio.

En el caso de los factores emocionales, en la mayoría de ellos los resultados reportan éxito en cada uno de sus campos. En los campos de "miedo", "Entusiasmo" y "felicidad", se obtuvo un 80% en promedio de éxito para la toma de decisiones de las personas en la mayoría de ellos, como se observa van desde los 21 hasta los 38 años, cubriendo toda la población de edades de la muestra. El resultado de los factores de conocimiento de la Tabla 2 muestra que los trucos y los engaños sobresalen de los demás, indicando éxito en la toma de decisiones de las personas, con un porcentaje arriba del 70 y 84% respectivamente, con rangos de edades de 22 a 36 años, abarcando una población considerable en la muestra. En la mayoría de los otros campos de ese factor, el resultado de porcentaje alto lo da el valor "no aplica", que indica que las personas no toman importancia en las habilidades o la racionalidad ya que se obtuvieron porcentajes de 70, con rangos de 21 a 33 años en promedio.

5. Conclusiones

Los factores reportados en este trabajo son solo los más representativos de nuestro estudio, considerando el gran número de variantes que influyen en la toma de decisiones en los seres humanos. Sin embargo, con la ayuda de los sistemas computacionales, se pueden buscar patrones similares en múltiples datos de una manera mucho más rápida en comparación con las búsquedas manuales exhaustivas realizadas, como se observa en los resultados.

La toma de decisiones para lograr éxito en los tiempos actuales en México está regida en su mayoría por los factores emocionales, de conocimientos y costumbres sociales. No obstante, los factores que indican fracaso en la toma de decisiones son los viscerales, psicológicos y de cooperación. En el presente estudio, los aspectos más dominantes para la toma de decisiones que conducen al éxito son aquellos que están relacionados con factores como el "Ritual", el "Relato", y la "Ilusión", que resultan muy comunes en cualquier tipo de sociedades. Asimismo, los campos como el "Miedo", "Entusiasmo" y la "Felicidad", se asocian al éxito al momento de tomar decisiones en la actualidad. Al respecto, el entusiasmo ayuda a impulsar la toma de decisiones de una manera más adecuada y el miedo, muchos autores indican que la superación de este al enfrentarlo, resulta muy adecuada y produce la mejora de cualquier actividad para el logro del éxito. Para el caso del hambre y dolor, estos se asocian mayoritariamente con el fracaso.

El ser humano es un sistema muy complejo, por lo que el estudio tanto individual como en sociedad sobre la toma de decisiones llevaría muchos años, además de que el tipo de análisis requiere de modelos muy complejos para la determinación de factores y la relación entre ellos. No obstante, la valiosa aportación de este estudio permite sentar las bases para seguir investigando con mayor profundidad sobre estos temas que

sin duda impactan tanto en el individuo como en las diferentes áreas y contextos que forman parte de una sociedad.

Finalmente, este tipo de trabajos multidisciplinarios son poco abordados por la comunidad científica debido a la complejidad y a la participación de diferentes áreas para su estudio. En este trabajo participaron investigadores del área de las ciencias administrativas los cuales tienen experiencia en la recolección y análisis de datos. También participaron investigadores de ciencias y tecnologías de la información que adecuan los modelos computacionales a este estudio. Se conto con la participación de una investigadora del área de la educación en ciencias, la cual tiene experiencia en la elaboración y validación de instrumentos, así como en la formulación de categorías e indicadores para el análisis de datos.

6. Trabajo a futuro

Como trabajo futuro, se programa extender el número de personas que interactúen con la aplicación móvil y comparar los nuevos datos con los ya obtenidos. También, se pretende realizar las encuestas en zonas vulnerables, según los indicadores de bienestar del INEGI, para determinar los patrones predominantes bajo estos contextos. Asimismo, se pretende abarcar lugares en donde la calidad de vida se considera alta por el INEGI, para comparar la contra posición de ambos lugares. Es importante hacer un análisis más robusto para determinar los patrones con los cuales la sociedad de manera individual y en conjunto realiza la toma decisiones. Finalmente, se pretende incluir más campos en cada uno de los factores determinados para que el estudio abarque una gran diversidad de valores y sea aún más certero.

Agradecimientos. Se agradece de manera muy especial el apoyo de la UAM Iztapalapa, TESCHI y UVM, Campus Hermosillo Sonora, por el apoyo brindado para la realización del proyecto, también a la Lic. Minerva Núñez Cedillo por el apoyo brindado en este artículo.

Referencias

1. Bohanec, M.: Decision making: a computer-science and information-technology view point-Interdisciplinary Description of Complex Systems 7(2), 22–37 (2009)
2. Nicolis, G., Nicolis, C.: Foundations of Complex Systems. World Scientific Pub., Singapore (2007)
3. Dodder, R., Dare, R.: Complex Adaptive Systems and Complexity Theory: Interrelated Knowledge (2000)
4. Luhmann, N. Organización y decisión: Autopoiesis, acción y entendimiento comunicativo. México: Anthropos / Universidad Iberoamericana. (2005)
5. Azollini, S., Depaula, P. Análisis de validez y confiabilidad de una escala para la evaluación de estilos decisorios operativos. Evaluar (13), 38–60 (2013)
6. Alpuche, E., Bernal J.L.: La Institución y la Organización: un análisis centrado en el actor. Intersticios Sociales (10), 1–29 (2015)

7. Elster, J.: Emotions and Economic Theory en Journal of Economic Literature. Vol. XXXV, 47–74 (1998)
8. Akerlof, G., Shiller, R.: Animal Spirits: cómo influye la psicología humana en la economía. Barcelona, 307–313 (2009)
9. Fehr, E., Gätcher, S.: Reciprocity and economics: The economic implications of Homo Reciprocans. European Economic Review no. (42), 854–859 (1998)
10. Fehr, E., Fischbacher, U.: Theeconomics of strongreciprocity. In: Gintis, H. et al. (eds.). Moral sentiments and material interests: The foundations of cooperation in economic life. Boston: MIT Press (2005)
11. Ostrom, E.: Comprender la diversidad institucional. México: Fondo de Cultura Económica/ Universidad Autónoma Metropolitana (2015)
12. Enriquez, E.: L'organisation en analyse. Presses Universitaires de France, Paris (22), 43–145 (1992)
13. Ponce, B.: Identifícalos para ganar más. Conoce los ciclos del mercado. Inversionista, Informe especial, 287, México: IASA Comunicación, SA de CV (2011)
14. Simon, H.: El Comportamiento Administrativo: estudio de los procesos de adopción de decisiones en la organización administrativa. Buenos Aires: Aguilar (1982)
15. Crozier, M., Friedberg, E.: El actor y el sistema: las restricciones de la acción colectiva. México: Alianza (1990)
16. Maciá, A., Barbero I, Pérez Llantada, C., Vila, E., Navas, E., Mandakovic, C.: Estudio interno de una escala de toma de decisión. Rev. de Psicol. Gral y Aplic., 42(2), 267–278 (1996)
17. Seok-Keun, Y., Bo-Young, K.: A Decision-Making Model for Adopting a Cloud Computing System Sustainability 10, 2952 (2018)
18. Medhat, W., et al.: Sentiment analysis algorithms and applications: A survey. Ain Shams Eng J (2014) http:// dx.doi.org/10.1016/j.asej.2014.04.011
19. Pérez, C.: Minería de datos: técnicas y herramientas. Editorial Paraninfo (2007)
20. Han, J., Kamber, M., Ferry, C.: Data Mining Conceps and Techniques. San Francisco: Morgan Kaufmann Publishers (2005)
21. Moreno-Montiel, B., Kinney-Romero, R.: ParalTabs: A Parallel Scheme of Decision Tables. In: Mexican International Conference on Computer Science (2013)
22. Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997) doi:http://dx.doi.org/10.1006/jcss.1997.1504.
23. Houser, D., Xiao, E.: Classification of natural language messages using a coordination game. Experimental Economics 14(1), 1–14 (2011) doi:10.1007/s10683-010- 9254-4

ISSN 1870-4069

Contrastación de algoritmos de aprendizaje automático para la clasificación de señales EEG

Alexis Edmundo Gallegos Acosta, María Dolores Torres Soto,
Aurora Torres Soto, Eunice Esther Ponce de León Sentí

Universidad Autónoma de Aguascalientes,
México

alexisedm@gmail.com,
{mdtorres, atorres, eponce}@correo.uaa.mx

Resumen. El objetivo de este artículo es presentar el ajuste de parámetros y la contrastación del desempeño de dos algoritmos de aprendizaje automático: máquinas de soporte vectorial y redes neuronales artificiales. Ambos algoritmos son aplicados a la clasificación de señales EEG con imaginación motora para discriminar la intensión de abrir y cerrar la mano. Para este estudio se contó con una base de datos propia obtenida por medio del dispositivo comercial Emotiv EPOC+ de catorce señales. Por medio de un diseño de experimentos factorial y un análisis estadístico, se obtuvieron los parámetros para los cuales, ambos algoritmos presentan un mejor desempeño. Así mismo, se determinó el algoritmo más adecuado para la clasificación de las señales EEG de acuerdo con su exactitud, precisión positiva, precisión negativa y las razones de falsos positivos y negativos.

Palabras clave: Aprendizaje automático, señales electroencefalográficas, máquinas de soporte vectorial, redes neuronales.

Machine Learning Algorithms Testing for EEG Signal Classification

Abstract. The purpose of this article is to present the parameter adjustment and the performance contrasting of two machine learning algorithms: support vector machines and artificial neural networks. Both algorithms are applied to the classification of EEG signals with motor imagination to discriminate the intention to open and close the hand. For this study, an own database was obtained through the Emotiv EPOC + device with fourteen signals. By means of a factorial experiment design and a statistical analysis, the parameters for which both algorithms present a better performance were obtained. Likewise, the most suitable algorithm for the classification of EEG signals was determined according to their accuracy, positive precision, negative precision, and false positive and negative rates.

Keywords: Machine learning, electroencephalographic signals, support vector machines, artificial neuronal networks.

1. Introducción

Uno de los principales objetos de estudio que ha tenido la ciencia a lo largo de la historia es el cuerpo humano. Particularmente, el cerebro ha resultado ser uno de los órganos con mayor interés debido a la complejidad de sus funciones [1]. Gracias a este interés nacen las neurociencias, un conjunto de ramas de la ciencia (neurología, física, biología, matemáticas, filosofía, sociología, ciencias computacionales, música, etc.) enfocadas en el estudio de la organización y funciones del cerebro [2]. Las neurociencias han permitido el desarrollo de herramientas y métodos que permiten encontrar soluciones a patologías y trastornos asociados al cerebro [3].

Una de las herramientas desarrolladas es el electroencefalograma (EEG), un dispositivo capaz de registrar la actividad eléctrica del cerebro y que en la actualidad es utilizado de modo rutinario [4]. La señal registrada es compleja pues contiene la suma de la actividad de diferentes poblaciones neuronales registrando incluso, otras actividades fisiológicas como el movimiento ocular, la respiración, la actividad cardiaca, etc. [4, 5]. En consecuencia, se han desarrollado técnicas de procesamiento matemático con el objetivo de mejorar la comprensión del funcionamiento cerebral [4].

El presente artículo se enfoca en la manipulación de señales electroencefalográficas de imaginación motora por medio de algoritmos de aprendizaje automático. Se expone, por medio de un diseño de experimentos factorial, el ajuste de dos algoritmos de clasificación: las redes neuronales artificiales y las máquinas de soporte vectorial. El objetivo es encontrar los parámetros que permiten un mejor desempeño en la clasificación de señales electroencefalográficas correspondientes a imaginación motora. Las señales se enfocan en la intención de abrir y cerrar la mano derecha sin ejecutar el movimiento físicamente. Para la captura de dichas señales se utilizó el dispositivo comercial Emotiv EPCO+ de 14 nodos descrito a detalle más adelante.

2. Temas principales

Este apartado se centra en abordar conceptos importantes utilizados para la elaboración de este artículo. El primero es la imaginación motora, es decir, la planeación mental de un movimiento sin la ejecución física [6]. Más adelante, se enfoca en conceptos concernientes a las ciencias computacionales como el aprendizaje automático y los dos algoritmos utilizados: las redes neuronales y las máquinas de soporte vectorial.

2.1. Imaginación de movimiento

La imaginación motora (MI), también conocida como intención de movimiento o imaginería motora, es un concepto relativamente nuevo que consiste en la evocación de un movimiento o gesto sin ejecutarlo físicamente [6], [7].

La práctica de la MI representa un tema de investigación para las neurociencias combinando el enfoque psicológico y neuropsicológico [8].

Hoy en día, la MI es aplicada en ámbitos deportivos y terapéuticos, especialmente en pacientes con afecciones neurológicas (i.e. accidentes cardiovasculares, lesiones en médula espinal o enfermedad de Parkinson). En la literatura se ha presentado evidencia de que la práctica de MI tiene efectos positivos al mejorar el desempeño de alguna habilidad motora [9, 10].

Al igual que el movimiento real, la intención de movimiento involucra las mismas áreas cerebrales, cuyos patrones de activación se registran en el electroencefalograma en frecuencias específicas en la región sensorial-motora de la corteza cerebral (ritmo mu entre 8 y 13 Hz; ritmo beta entre 13 y 30 Hz) [6].

2.2. Aprendizaje automático

El aprendizaje automático, o machine learning (ML), es una rama de la inteligencia artificial dedicada al desarrollo de técnicas que permitan a las computadoras generalizar comportamientos y reconocer patrones a partir de la información suministrada [11], [12]. De acuerdo con Tom Mitchell en [13], "*se dice que un programa informático aprende de un conjunto de tareas, gracias a la experiencia y a una medida de rendimiento, si su desempeño en dichas tareas mejora con la experiencia*".

Un sistema informático clásico sólo ejecuta una misma operación una y otra vez. En cambio, un sistema ML tiene la capacidad de extraer información automáticamente de los datos que procesa, de modo que el sistema "aprende", mejorando su desempeño en tarea para la que fue programado [14].

Un modelo de aprendizaje automático se conforma de un proceso de selección y un proceso de adaptación. Primeramente, el sistema ML selecciona las características más relevantes de un evento u objeto y las contrasta con las ya conocidas. En caso de que encuentre diferencias significativas, el sistema las adapta al modelo del evento (u objeto). Finalmente, este proceso se traduce en la mejora en la calidad de actuación del sistema [15].

En la literatura existen diversos algoritmos de aprendizaje automático e incluso, variantes derivados de los mismos [16]. En este caso, se hace uso de dos de los más usados y con mejor desempeño: las redes neuronales y las máquinas de soporte vectorial.

2.3. Redes neuronales artificiales

Las redes neuronales artificiales (RNA) son un tipo de sistema de aprendizaje automático capaz de establecer relaciones no lineales entre variables de entrada y salida [17]. De manera general, proporcionan un enfoque sólido capaz de aproximar funciones objetivo de valores reales, valores discretos y vectores [13].

El funcionamiento de las redes neuronales artificiales se basa en el funcionamiento del sistema neuronal humano, un sistema complejo, no lineal y paralelo [18, 19]. Este sistema se conforma de interconexiones entre neuronas formando un punto clave para el procesamiento del conocimiento, cuya estructura es similar a un árbol [19].

En las RNA cada neurona artificial se somete a una función de activación que es modelada por una combinación lineal de las entradas de dicha neurona. Esta función de activación da como resultado la salida de la neurona [20].

Las RNA permiten la resolución de problemas e incluso, pueden trabajar en conjunto con otros métodos [19]. Son modelos del cálculo eficientes que operan en paralelo y se especializan en tareas cognitivas como aprendizaje de patrones, clasificación o la optimización [21].

2.4. Máquinas de soporte vectorial

Las máquinas de soporte vectorial (SVM por sus siglas en inglés support vector machines) son una clase poderosa de algoritmos supervisados de clasificación y regresión [22]. Se originan de la teoría del aprendizaje estadístico cuyo propósito inicial fue la resolución de problemas de clasificación binaria. Sin embargo, se han desarrollado al punto de que pueden utilizarse en problemas de regresión, agrupamiento y multi clasificación mostrando un buen desempeño [23].

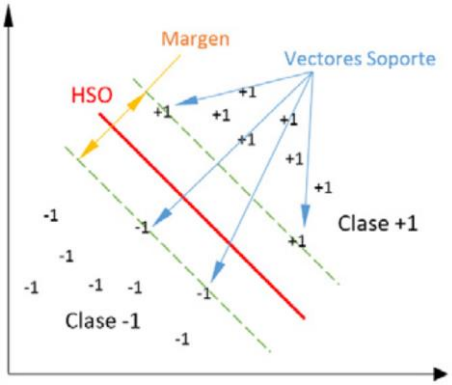


Fig. 1. Funcionamiento de una SVM [24].

En el caso más simple, las SVM determinan el hiperplano que separa los conjuntos de objetos según su clase. Este hiperplano se conoce como hiperplano de separación óptima (HSO) y se distingue de cualquier otro separador posible al maximizar el margen entre las clases [25] como se muestra en la Fig.1. Los vectores de soporte, que dan nombre al algoritmo de aprendizaje, son las características de datos que se encuentran más cercanos al HSO e influyen en su ubicación [24].

La mayoría de los casos no son problemas linealmente separables y, por lo tanto, las SVM no pueden encontrar un HSO. Sin embargo, los SVM hacen uso de métodos kernel que les permite aumentar la dimensionalidad en el mapeo de las características conocido como espacio de características. De esta manera, se puede encontrar una solución lineal en el espacio de características que, a su vez, es una solución no lineal en el espacio de entrada [23, 24].

3. Metodología

La metodología (ver Fig.2) inicia con la obtención de señales EEG orientadas específicamente a la intención de movimiento de abrir y cerrar la mano derecha.

La obtención de las señales se realizó por medio del dispositivo comercial Emotiv EPOC+ el cual, permite la captura y amplificación de ondas cerebrales generadas por acciones mentales [1]. El dispositivo posee catorce receptores (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) distribuidos según el sistema internacional 10-20 [26]. Además, se contó la aplicación Cykit [27] de acceso libre que permitió controlar el tiempo de muestreo y registrar las observaciones en archivos de texto.

Finalmente, Cada muestra tomada tuvo una duración de 15 segundos en la cual, se solicitaba al sujeto de prueba mantener la concentración en el movimiento específico. Al final del muestreo se contó con 1283 registros conformados por 566 correspondientes a abrir la mano y 717 a cerrar la mano.

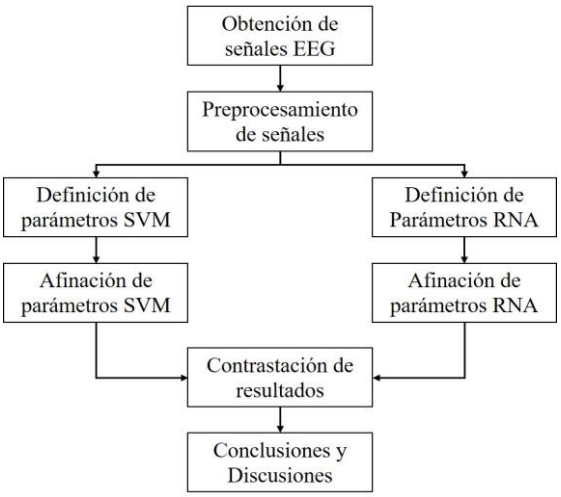


Fig. 2. Metodología.

Tabla 1. Parámetros de entrada para SVM.

Kernel	Gamma	C
rbf	1/14	1E+10
poly	0.08	1500
	0.04	200

Como segundo paso se realizó un preprocesamiento de las señales debido a la complejidad de los datos. Este preprocesamiento consistió en trasladar dichos datos a una escala común por medio de una normalización Z-score [28, 29].

Una vez de contó con los datos, se prosiguió con una definición de parámetros para un modelo de SVM y dos de RNA. Este paso consistió en buscar una serie de valores para cada parámetro con los cuales cada algoritmo de aprendizaje tuviera un buen desempeño. Los valores se basan en literatura y en pruebas empíricas realizadas a los algoritmos.

Tabla 2. Parámetros de entrada para los modelos 1 y 2 de RNA.

Modelo	Neuronas en capa oculta	Función de activación	Neuronas en capa oculta 2	Función de activación	N° de épocas	Tamaño de lote
RNA 1 capa	75	Relu	N/A	N/A	100	10
	50	Tanh	N/A	N/A	50	5
	25					
RNA 2 capas	75	Relu	20	Relu	100	10
	50	Tanh	15	Tanh	50	5
	25					

Para el caso de las SVM, los parámetros fueron el kernel, gamma y el valor C. En el caso del kernel, se seleccionaron el rbf y el poly, de acuerdo con los valores obtenidos en [30] para la clasificación de señales EEG para detectar ojos abiertos y cerrados. Por su parte, el valor gamma (γ) define la anchura del kernel. Si su valor es demasiado grande el modelo tenderá al sobreajuste y en caso de ser muy pequeño la frontera de decisión sería muy rígida [31]. Los valores seleccionados para gamma se basan en la librería sklearn de Python cuyo valor defecto es *1/n*, siendo *n* el número de características que se analizan [32].

Por su parte, el valor C controla las violaciones del margen que se toleran en el proceso de ajuste [33]. Se trata de un peso de regularización que permite que se cometan errores al clasificar. Si el valor de C es muy alto, el modelo SVM permite menor número de violaciones en la clasificación [31]. En este caso, los valores de C se eligieron de manera que pudieran explorar distintos niveles de dicho valor.

En el caso de las RNA se diseñaron dos modelos como se aprecia en la Tabla 2. El primero es una RNA densa con una capa oculta y el segundo, una RNA densa de dos capas ocultas. Pues se afirma que la mayoría de los problemas no requieren de más de dos capas para proveer un buen desempeño e incluso, el uso más capas y neuronas supone un mayor costo computacional [21, 34].

Ambos modelos se componen de una capa de entrada de catorce neuronas, según el número de características a procesar; y una capa de salida con una única neurona con función de activación sigmoide que permite transformar la salida en 0 y 1, donde 0 representa una señal para abrir la mano y 1 una señal para cerrar la mano.

Para establecer los valores para los parámetros expuestos en la Tabla 2 se acudió a la realización de pruebas empíricas para analizar el comportamiento del modelo RNA con los datos de aprendizaje y validación. Para el caso del número de neuronas en la capa oculta se tomó como base la regla de Lippman la cual sostiene que problemas complejos pueden resolverse con una RNA de una capa oculta siempre y cuando, se incluya al menos tres veces el número de nodos de entrada [21]. Sin embargo, existen diversas topologías de RNA y queda a consideración el número de neuronas en cada capa oculta [35].

Para el segundo modelo de RNA se incluye una segunda capa oculta con un número de neuronas menor al de la primera capa oculta. De esta manera, se ajusta a la regla de la pirámide geométrica donde se establece que el número de neuronas ha de ser menor al número de entradas, en este caso la capa oculta anterior [21, 34].

Al igual que en el modelo 1 de RNA, para el modelo 2 se realizaron ejecuciones empíricas para conocer su desempeño y saber qué valores se ajustan mejor a la clasificación esperada.

De acuerdo con la metodología de la Fig. 3, el siguiente paso es la afinación de parámetros para ambos algoritmos de aprendizaje. Para ello, se realizó un diseño de experimentos factorial para cada algoritmo de aprendizaje en el que se ejecutó cada configuración de parámetros y se replicaron treinta veces para formar grupos de experimentos. A partir de estos grupos de experimentos se realizaron pruebas estadísticas para determinar la existencia de diferencias entre los grupos y encontrar las configuraciones que permiten un mejor desempeño, es decir, que maximice la exactitud, la precisión positiva y negativa; y a su vez, minimice las razones de falsos positivos y negativos.

Una vez terminada la afinación de parámetros, se realizó una contrastación de resultados. Este paso en la metodología consistió en determinar la existencia de diferencias significativas entre los mejores grupos de cada algoritmo y así conocer aquel que resulta más adecuado en la clasificación de señales EEG con imaginación motora.

4. Resultados

Como se mencionó anteriormente, cada configuración de parámetros en los modelos SVM y RNA se replicó en un total de 30 ocasiones independientes cuyas salidas fueron las métricas de la matriz de confusión de cada réplica: la exactitud, la precisión positiva, la precisión negativa y las razones de falsos positivos y negativos.

Para las RNA los modelos de una y dos capas ocultas se analizaron en conjunto. El análisis comenzó con una prueba Shapiro Wilk con la que se encontró que las variables de salida no siguen una distribución normal, por lo tanto, se optó por estadística no paramétrica. Dado este resultado, se realizó una prueba Kruskal-Wallis con la que se descubrió la existencia de diferencia significativa entre los grupos de réplicas. Así pues, se seleccionaron empíricamente cuatro grupos de cada modelo RNA de acuerdo con su desempeño, es decir, los grupos que maximizan su exactitud y precisión (positiva y negativa) y minimizan la razón de falsos positivos y falsos negativos.

Dada una nueva prueba de Kruskal-Wallis los grupos seleccionados resultan ser equiparables estadísticamente. Por lo tanto, para el RNA de una capa oculta se seleccionó la configuración de 75 neuronas en su capa oculta, la función de activación relu, 100 épocas y un tamaño de batch de 5. Este modelo fue seleccionado pues maximiza la exactitud con un 95.71% y una precisión positiva y negativa de 95.6% y 95.88%, respectivamente. A su vez, minimiza las razones de falsos positivos y negativos con 4.11% y 4.39%, respectivamente (ver Tabla 3).

En cuanto a los modelos de RNA con dos capas se seleccionó la configuración con 75 neuronas y función de activación relu en la primera capa, 15 neuronas y activación relu en la segunda capa, 100 épocas y un batch de 5. Este modelo se distingue de los otros modelos de 2 capas pues maximiza la exactitud con un 95.81% y la precisión positiva con un 96.07%. A su vez, minimiza únicamente la razón de falsos negativos con un 3.92%.

Tabla 3. Mejores modelos en SVM y RNA.

Modelo	Configuración	E.	P (+)	P (-)	R. F (+)	R. F (-)
RNA 1 capa	C1:75, A: relu, Epoc: 100, Batch: 5	95.71	95.60	95.88	4.11	4.39
RNA 2 capas	C1: 75, A: relu, C2: 15, A: relu, Epoc: 100, Batch: 5	95.81	96.07	95.51	4.48	3.92
SVM	Kernel: rbf, Gama: 0.04, C: 200	93.00	93.06	92.93	7.06	6.93

El mejor modelo una capa y el mejor de dos capas se realizó una prueba U-Mann Whitney resultando en que son equiparables estadísticamente. Sin embargo, se puede observar en la Tabla 3, que el modelo de dos capas es empíricamente superior, aunque requiere un mayor costo computacional. Por lo tanto, se eligen el modelo de una capa al ser un modelo con menor complejidad, menor costo computacional e, incluso, menor tiempo de ejecución observado.

En cuanto a los grupos de experimentos de SVM, el proceso fue similar. Una prueba de Shapiro Wilk determinó que los grupos no siguen una distribución normal y, por lo tanto, se requirió de estadística no paramétrica.

De acuerdo con una prueba Kruskal-Wallis se determinó la existencia de diferencias significativas entre los grupos de experimentos. Dada la diferencia entre los grupos, se seleccionaron de forma empírica los cuatro mejores según la maximización de la exactitud y las precisión positiva y negativa, así como la minimización de las razones de falsos positivos y negativos.

Los cuatro grupos se sometieron a una nueva prueba de Kruslal-Wallis que resultó en la existencia de diferencia significativa respecto a las cinco variables de salida. Entonces se decide optar por contrastar dos grupos:

- Grupo 8 con kernel: rbf, gamma: 0.04, c: 200, exactitud de 93%, precisión positiva de 93.06%, precisión negativa de 92.93%, razón de falsos positivos de 7.06%, razón de falsos negativos de 6.93%
- Grupo 18 con kernel: poly, gamma: 0.04, c: 200, exactitud de 77.66%, precisión positiva de 64.48%, precisión negativa de 94.40%, razón de falsos positivos de 5.59%, razón de falsos negativos de 35.51%

El grupo 8 maximiza la exactitud, la precisión positiva y minimiza la razón de falsos negativos. En cambio, el grupo 18 maximiza la precisión negativa y minimiza la razón de falsos positivos. Ambos grupos se contrastaron por medio de una prueba U-Mann Whitney resultado en que son grupos estadísticamente distintos. Por esta razón, se opta por el grupo 8 (ver Tabla 3) al ser superior en tres de las cinco variables de salida.

Finalmente, se realizó una nueva contrastación entre los modelos SVM y RNA por medio de sus grupos elegidos como mejores de acuerdo con el análisis estadístico aplicado. En este caso, se seleccionó el modelo de una capa oculta de RNA y el modelo de SVM descritos en la Tabla 3.

Por medio de una prueba U-Mann Whitney se determinó la existencia de diferencia significativa entre ambos grupos.

Sin embargo, y como se puede observar en la Tabla 3, el modelo RNA de una capa oculta maximiza la exactitud y la precisión tanto positiva y negativa al tiempo que minimiza la razón de falsos positivos y falsos negativos. Es por ello por lo que puede considerarse como la mejor opción para clasificación de señales EEG.

5. Conclusiones y trabajo a futuro

En conclusión, se tiene que los algoritmos de aprendizaje automático, tanto las máquinas de soporte vectorial como las redes neuronales artificiales, son herramientas muy robustas que pueden adaptarse a distintos problemas como la clasificación de señales EEG. Sin embargo, la tarea de ajustar los parámetros de los algoritmos es crucial para obtener un buen desempeño y uso adecuado de recursos computacionales.

Esto se debe a que es posible obtener un modelo de clasificación con muy buen desempeño y que, a su vez, sea demasiado complejo y sea necesario un reajuste de parámetros u optar por un modelo distinto. Por ejemplo, los modelos de RNA con una y dos capas, descritos en la Tabla 3, donde ambos modelos obtuvieron un buen desempeño, sin embargo, se puede observar que el modelo con dos capas ocultas es superior en sus variables de salida. No obstante, se trata de un modelo con mayor complejidad y mayor costo computacional. Es por ello, que se puede optar por un modelo más sencillo cuyo desempeño puede ser mínimamente inferior, pero con mejor uso de recursos.

La manipulación de señales EEG por medio de la inteligencia artificial provee grandes avances en la detección de enfermedades relacionadas al funcionamiento cerebral. Así como la creación de nuevas herramientas que promueven la inclusión de personas con discapacidades físicas, permitiéndoles manipular dispositivos por medio de la traducción de las señales en comandos de control.

Como trabajo a futuro se tiene propuesto realizar pruebas similares tomando en cuenta el tiempo como factor de decisión para seleccionar el modelo más adecuado para la clasificación de señales electroencefalográficas con imaginación motora.

Referencias

1. Chávez-Saenz, V., Torres-Ramírez, D., Herrera-Ogaz, J., Hernández-Rodríguez, A.: Adquisición y análisis de señales electroencefalográficas utilizando el dispositivo Emotiv EPOC+. 3(7), ECOFARN, pp. 107–118 (2016)
2. Ibanez, A., García, A.: Que son las Neurociencias. Ediciones Paidós (2015)
3. Espinosa, L.: Electroencefalografía inalámbrica: una mirada actual y propuesta de sistema portátil. Universidad Técnica Federico Santa María (2016)
4. Urrestarazu, E., Iriarte, J.: Mathematical analyses in the study of electroencephalographic signals. Rev. Neurol., 41(5), pp. 423–434 (2005)
5. Torres-García, A.A., Reyes-García, C.A., Villaseñor-Pineda, L., Ramírez-Cortés, J.M.: Análisis de señales electroencefalográficas para la clasificación de habla imaginada. Rev. Mex. Ing. biomédica, 34(1), pp. 23–39 (2013)
6. Carino-Escobar, R., Cantillo-Negrete, J., Gutiérrez-Martínez, J., Vázquez, R.A.: Decodificación de imaginación motora en la señal de electroencefalograma mediante mapas auto-organizados. Rev. del Cent. Investig. la Univ. La Salle, 12(45), pp. 107–125 (2016)

7. Fernández-Gómez, E., Sánchez-Cabeza, Á.: Imaginería motora: revisión sistemática de su efectividad en la rehabilitación de la extremidad superior tras un ictus. Rev. Neurol., 66(5), pp. 137–146 (2018)
8. Di-Rienzo, F., Debarnot, U., Daligault, S., Saruco, E., Delpuech, C., Doyon, J., Collet, C., Guillot, A.: Online and offline performance gains following motor imagery practice: a comprehensive review of behavioral and neuroimaging studies. Front. Hum. Neurosci., 10, pp. 315 (2016)
9. Dickstein, R., Deutsch, J.E.: Motor imagery in physical therapist practice. Phys. Ther., 87(7), pp. 942–953 (2007)
10. Marengo, S.: Tipos de imaginación motora y procesamiento predictivo. Poiésis, 1, pp. 58–71 (2018)
11. Sancho-Caparrini, F.: Introducción al aprendizaje automático. http://www.cs.us.es/~fsancho/?e=75 (2019)
12. Soto-Valero, C.: Aplicación de métodos de aprendizaje automático en el análisis y la predicción de resultados deportivos. Retos: nuevas tendencias en educación física, deporte y recreación, 34, pp. 377–382 (2018)
13. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
14. González-García, C.: En qué consiste el aprendizaje automático (machine learning) y qué está aportando a la Neurociencia Cognitiva. Cienc. Cogn., 12(2), pp. 48–50 (2018)
15. Moreno, A.: Aprendizaje automático. Universitat Politècnica de Catalunya, Edicions UPC (1994)
16. Cerón, J.P.: 8 algoritmos de aprendizaje automático data mining más usados. https://www.arame.com.mx/blog/8-algoritmos-aprendizaje-automatico-data-mining-mas-usados-arame-blog/ (2020)
17. Villada, F., Muñoz, N., García-Quintero, E.: Redes neuronales artificiales aplicadas a la predicción del precio del oro. Informacion Tecnologica, (27)5, pp. 143–150 (2016)
18. Izaurieta, F., Saavedra, C.: Redes neuronales artificiales. Concepción (2000)
19. Rivas-Asanza, W., Mazon-Olivo, B., Mejia, F.: Capítulo 1: Generalidades de las redes neuronales artificiales. In Redes neuronales artificiales aplicadas al reconocimiento de patrones, Universidad Técnica de Machala, pp. 11–35 (2018)
20. Pava-Chipol, N.D., Hernández-Gomez, L.H., Beltran-Fernandez, J.A., Pava-Chipol, J.F., Moreno-Garibaldi, P.: Implementación de las redes neuronales artificiales para seleccionar los componentes de una prótesis modular transfemoral. DYNA New Technol., 4(1) (2017)
21. Lévy-Magin, J.P., Flórez López, R., Fernández Fernández, J.M.: Las Redes Neuronales Artificiales. Fundamentos teóricos y aplicaciones prácticas. (2008)
22. VanderPlas, J.: Python Dada Science Handbook. O'Reilly (2017)
23. Carmona, E.: Tutorial sobre Máquinas de Vectores Soporte (SVM) (2016)
24. González, R., Barrientos, A., Toapanta, M., Del Cerro, J.: Aplicación de las máquinas de soporte vectorial (SVM) al diagnóstico clínico de la enfermedad de párkinson y el temblor esencial. RIAI - Rev. Iberoam. Autom. e Inform. Ind., 14(4), pp. 394–405 (2017)
25. Betancourt, G.: Las máquinas de soporte vectorial (SVMs). Sci. Tech., 27 (2005)
26. Soria-Morillo, L., Alvarez-Garcia, J., Gonzalez-Abril, L., Ortega, J.: Discrete classification technique applied to TV advertisements liking recognition system based on low-cost EEG headsets. Biomed. Eng. Online, 15 (2016)
27. Cymaticorps: Cykit Cymaticorp. https://github.com/CymatiCorp/CyKit (2017)
28. Medina, B., Sierra, J.E., Ulloa, A.B.: Técnicas de extracción de características de señales EEG en la imaginación de movimiento para sistemas BCI Extraction techniques of EEG signals characteristics in motion imagination for BCI systems. Espacios, 39(22), pp. 36–48 (2018)
29. Carlos-Cubero, J., Berzal, F.: Introducción a SPSS. Sistemas Inteligentes de Gestión. https://elvex.ugr.es/decsai/intelligent/workbook/D0 SPSS.pdf (2020)
30. Santa-Cruz, O., Del Mar-Ramírez, L., Trujillo-Romero, F.: Técnicas de aprendizaje

automático aplicadas a electroencefalogramas. Research in Computer Science, 113. Instituto Politécnico Nacional, pp. 53–65 (2016)
31. Salamanca-Polo, L.: Generación de conjuntos de máquinas de soporte vectorial mediante técnicas de remuestreo e inyección de ruido en las etiquetas de clase. Universidad Autónoma de Madrid (2017)
32. Scikit-learn developers: Sklearn.svm.SVC, Scikit learn. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (2020)
33. Rodrigo, J.A.: Máquinas de Vector Soporte (Support Vector Machines, SVMs), https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines (2020)
34. Paz-Villaseca, D.I.: Herramientas deep learning para predictive analytics. Universidad de Concepción (2017)
35. Torres-López, S., Lazara, M., Cuza, A., Piñero-Perez, P.Y., Piedra-Diéguez, L.A.: Red neuronal multicapa para la evaluación de competencias laborales Multilayer neural network for assessment of labor competences. Revista Cuba Ciencias Informáticas, 10 (2016)

ISSN 1870-4069

Mecanismo de clasificación para diabetes mellitus en la población de Aguascalientes, México

Pablo Rodríguez de León, María Dolores Torres Soto, Aurora Torres Soto

Benemérita Universidad Autónoma de Aguascalientes,
Posgrado e Investigación,
México

pablo.rdz@live.com, mdtorres@correo.uaa.mx,
atorres@correo.uaa.mx

Resumen. Año con año en México, se pierden muchas vidas debido a diabetes mellitus la cual es una enfermedad crónica asociada a otras enfermedades crónicas como hipertensión, dislipidemia y síndrome metabólico, por esta razón nació la inquietud de identificar tempranamente las variables características relacionadas al desarrollo de esta enfermedad. Esta investigación presenta la creación de un mecanismo de clasificación para seleccionar subconjuntos de características y clasificar pacientes con la patología de diabetes mellitus en la población de Aguascalientes, México. En esta investigación se diseñó, desarrollo y puso a punto un mecanismo integrado por dos mecanismos que fueron ejecutados de forma secuencial, el primero fue utilizando la técnica de selección de variables de testores típicos, con este mecanismo se obtuvieron los sub conjuntos de características interrelacionadas que mejor describen un paciente con la patología de diabetes mellitus, en el segundo mecanismo se empleó la técnica de redes neuronales artificiales con la cual se diseñó, desarrollo, entreno y se validó un clasificador con una precisión de casi el 92%. Cabe mencionar que en esta investigación se contó con información extraída de consultas de pacientes del sistema expediente clínico electrónico del ISSEA, el cual basa sus diagnósticos en el estándar internacional CIE 10. Como resultados de esta investigación se desarrolló un clasificador con una precisión de casi el 92% para identificación de pacientes con diabetes mellitus en la población de Aguascalientes, y por medio de la selección de variables se identificaron variables que no son comúnmente consideradas en conjunto en la literatura, tales como el estado civil, existencia de complicaciones y antecedentes familiares con diabetes.

Palabras clave: Testores típicos, redes neuronales artificiales, diabetes mellitus, catalogo internacional de enfermedades 10.

Classification Mechanism for Diabetes Mellitus in the Population of Aguascalientes, Mexico

Abstract. Year after year in Mexico many lives are lost due to diabetes mellitus which is a chronic disease associated with other chronic diseases such as

hypertension, dyslipidemia, and metabolic síndrome, for this reason, the concern was raised to the early identification of the characteristic variables related to the development of this disease. This research presents the creation of a hybrid classification mechanism to select subsets of characteristics and classify patients with the pathology of diabetes mellitus in the population of Aguascalientes, Mexico. In this research, a hybrid mechanism composed of two mechanisms that were executed sequentially was designed, developed, and developed. The first was using the technique of variable selection of Typical Testers, with this mechanism the subsets of interrelated characteristics that best describe a patient with the pathology of diabetes mellitus were obtained. In the second mechanism, the artificial neural network technique was used, a classifier was designed, developed, trained, and validated with an accuracy of almost 92%. It is worth mentioning that in this research, information was obtained from patient consultations of the ISSEA electronic clinical record system, which bases its diagnoses on the international ICD 10 standard. As a result of this research, a classifier was developed with an accuracy of almost 92% for identification of patients with diabetes mellitus in the population of Aguascalientes, and through the selection of variables, variables were identified that are not commonly considered together in the literature, such as the marital status, existence of health complications and family history with diabetes.

Keywords: Typical testors, artificial neural networks, diabetes mellitus, international catalog of diseases 10.

1. Introducción

En la actualidad, los sistemas de consultas médicas producen una gran cantidad de datos e información, los cuales no son aprovechados y analizados por los organismos que las crean y sustentan, ya que carecen de conocimiento referente a procesamiento, transformación e interpretación de la información.

Para esta investigación se otorgó información de datos de consultas de pacientes de centros de salud del Instituto de Servicios de Salud del Estado de Aguascalientes (ISSEA), esta información se preproceso, limpio y ordeno para conseguir los recursos que se categorizaron o discretizaron para su procesamiento y análisis.

En esta investigación, se diseñó, desarrollo y se puso a punto un mecanismo de clasificación compuesto por dos mecanismos, uno basado en la técnica de selección de variables llamada testores típicos y el otro mecanismo basado en la técnica de aprendizaje automático llamada redes neuronales artificiales.

Mediante la utilización de la técnica testores típicos, se consideraron dos clases: paciente con diabetes y pacientes sin diabetes, se obtuvieron todos los posibles subconjuntos de características que describen a un paciente, mediante el cálculo del peso informacional [1] se evalúa la relevancia de cada variable.

Seguido de esto se empleó la técnica de redes neuronales artificiales utilizando las variables obtenidas por los testores típicos y así se crea un clasificador para clasificar pacientes con diabetes en la población de Aguascalientes.

En parte la motivación de esta investigación es el poder tener un temprano y oportuno diagnóstico de esta patología, ya que la diabetes mellitus es la primera causa de muerte a nivel mundial [2] y la segunda causa de muerte tanto en mujeres como en

hombres en México [3], esta patología ha llegado al nivel de ser comparado con epidemias.

En la actualidad, se cree que en conjunto las variables: Tensión arterial, nivel de glucemia, el índice de masa corporal y el género del paciente son los factores que determinan la presencia de la diabetes mellitus en la población de Aguascalientes.

Un rasgo destacado de esta investigación es que se han considerado los insumos de información manejados en el sistema de salud más importante del Estado de Aguascalientes llamado Expediente Clínico Electrónico, basados en CIE10 [1].

2. Conceptos relacionados

2.1. Diabetes mellitus

La diabetes mellitus es una enfermedad crónica en la cual el cuerpo humano no utiliza bien la insulina producida por el páncreas o el mismo páncreas no produce insulina, la cual es una hormona que interviene como válvula permitiendo que la glucosa de los alimentos ingeridos pase de la sangre a las células para la producción de energía, la glucosa es asistida por la insulina para entrar a las células, es así como la falta de insulina se refleja en un exceso de glucosa [3].

Los bajos niveles de producción de insulina o de utilizarla de manera eficaz lleva a presentar hiperglucemia o niveles elevados de glucosa en la sangre. Manifestar altos niveles de glucosa durante un tiempo prolongado se relacionan con casos de pacientes con daños corporales y fallas en algunos órganos y tejidos [2].

De acuerdo con la Federación Internacional de Diabetes, China, India, Estados Unidos, Brasil, Rusia y México, son en ese orden los países con mayor número de diabéticos [3]. La Organización Mundial de la Salud posiciona a la diabetes mellitus es a nivel mundial como la Octava causa de muerte del año 2000 al 2012, el INEGI sitúa a la diabetes mellitus como una de las 3 causas de muerte desde 2006 a nivel nacional, el Gobierno del Estado de Aguascalientes la posiciona como la tercer causa en mortalidad general y en primer lugar como causa única de muerte [2].

En 2016 se realizó un estudio acerca niveles de diabetes en todo el mundo, dirigido por científicos del Imperial College de Londres, de la escuela de salud pública Harvard T.H. Chan, de la OMS y otros 500 investigadores de todo el mundo este estudio menciona que en 2016 el costo anual del tratamiento de diabetes fue de más de 825 billones de dólares nivel mundial, 170 billones en china, 105 billones en Estados Unidos de América y 73 billones de dólares en la India.

2.2. Testores típicos

Cuando se cuenta con un grupo de datos con demasiadas clases se incrementa la probabilidad de que algoritmos de clasificación encuentren subgrupos de características que no son totalmente correctos [4, 5]. Los métodos de selección pueden acortar el tiempo de entrenamiento de un modelo clasificador [6], de esta manera se simplifica la interpretación a los usuarios [7], se evita el exceso de dimensión y se generaliza por reducción de varianza.

Los métodos de selección de variables se separan en filter, wrapper e híbridos. Los Métodos wrapper: estos se utilizan para medir o conducir el proceso de búsqueda de los diferentes subconjuntos de variables, utilizando un algoritmo de agrupamiento y evaluando los resultados mediante criterios específicos. Estos califican las variables en subconjuntos y no solo cada variable por separado.

Los testores típicos se encuentran ubicados en estos métodos de selección de variables wrapper ya que seleccionan los subgrupos de variables y su relación evaluando todas las variables del subgrupo, estos consideran el subconjunto y su interrelación multivariable.

El concepto de testor aparece a mediados de los años cincuenta, fue empleado en la detección de fallas de circuitos eléctricos por Cheguis y Yablonskii [8], además se utilizó en selección de variables en problemas de Geología [9]. La ponderación de importancia de cada una de las variables relacionadas a un problema puede ser alcanzar mediante el cálculo del peso informacional [10].

Considerando qué U es un conjunto de objetos descritos por N características, agrupados en K clases. En base a la comparación de cada característica de los objetos que pertenecen a una clase contra los que pertenecen a las demás, se crea la matriz de diferencias MD. MD se construye por un criterio de comparación o de igualdad estricta de rasgos [11].

Una vez que se crea la MD, se genera la matriz básica MB, que está constituida por las filas de MD que son básicas, esto es:

Una fila i_q es básica si no existe alguna fila i_p que sea subfila de i_q.

Sean i_p e i_q filas de MD.

Se dice que i_p es una subfila de i_q si para todo elemento de $i_q = 0$ se cumple que $i_p = 0$ y además, existe por lo menos un elemento de $i_q = 1$ en el que $i_p = 0$ [12].

Una fila de la MB es un testor típico, si al eliminar cualquiera de sus características, pierde la propiedad de ser un testor, como citan Santiesteban y Pons en 2003 [13].

Un testor típico es un testor en su mínima expresión o aquel testor al que se le ha eliminado toda redundancia [14].

2.3. Redes neuronales artificiales

En la actualidad las técnicas de machine learning presentan incremento en su utilización tanto en el ámbito empresarial como el ámbito académico, ya que son métodos capaces de detectar automáticamente patrones en la información, ya sea para predecir o tomar decisiones.

Una red característica radica en una secuencia de capas con conexiones entre capas contiguas consecutivas, general mente las redes neuronales artificiales (RNA) están organizadas en capas, de las cuales dos capas son de contacto con el exterior, una capa de entrada en la que ingresan los datos a la red, y una capa de salida la cual entrega la respuesta de la red de entrada, el resto de las capas de la red son llamadas capas ocultas.

Una RNA puede ser o no densamente interconectada, y tiene los siguientes elementos, 1.-Entrada, la cual recibe la información desde el exterior, 2.- Pesos, cada entrada tiene su propio peso relativo o importancia de la entrada, 3.-Suma ponderada y 4.-Salida.

Para desarrollar una RNA no hay que programar ni el conocimiento ni las reglas del procesamiento del conocimiento, la RNA aprende las reglas del procesamiento del

conocimiento a través de ajustes de conexiones ponderadas entre las neuronas de las distintas capas de la red.

Existen dos fases en la modelización de RNA como lo menciona Marín Diazaraque [15].Entrenamiento: se utiliza un conjunto de datos de entrenamiento para determinar los pesos que definen el modelo de red neuronal.

Prueba: es la fase donde se evalúa la precisión del modelo entrenado y para esta fase se utiliza un grupo de datos distintos a los de entrenamiento, ya que puede haber sobre ajuste que es cuando el modelo se ajusta demasiado a las singularidades en los patrones de entrenamiento y por lo cual pierde la habilidad de generalizar su aprendizaje a casos nuevos.

Las neuronas artificiales tienen diferentes estados de activación, comúnmente son dos, pero otras pueden tomar un valor de un conjunto definido. Con esta función de activación se evita que el resultado de una red neuronal sea idéntico a solo tener una sola neurona, para esto se necesita que cada recta pase por una transformación no lineal. Estas transformaciones pueden ser: lineal, relu o unidad lineal rectificada, sigmoide o tangente.

2.4. CIE 10

Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud Décima Revisión o CIE 10, establece un registro sistemático, interpretación, análisis y comparación de los datos de la mortalidad y morbilidad en diferentes países y épocas. Permite clasificar patologías y problemas relacionados con la salud, así como diagnósticos, razones de admisión, afecciones tratadas, motivo de la consulta e incluso las causas de defunción [16].

3. Trabajos relacionados

3.1. Reglas de predicción clínica

Tradicionalmente una gran cantidad de médicos tratan a la diabetes Mellitus y otras patologías mediante las Reglas de Predicción Clínica (RPC), las cuales son herramientas de diagnóstico, pronóstico o intervención que incluyen las características clínicas ponderadas del historial clínico, examen físico y exámenes de gabinete[17], es decir que se basan en la forma en que se han diagnosticado a otros pacientes y el modo en el que han evolucionado.

Esto se hace con base en los conocimientos de casos del médico o mediante casos que han sido publicados, pues la información sobre esta enfermedad está en constante cambio, médicos colaboran en descubrimientos y estudios, información que es relevante, aunque, esto hace que sea complicado tomar en cuenta todos los datos con los que se encuentran disponibles.

Debido a que las reglas de predicción clínica no se encuentran sistematizadas en una herramienta que asista al médico durante la consulta, el desarrollo de un mecanismo clasificador mediante testores típicos y redes neuronales artificiales puede ser una herramienta útil para mejorar la forma en que se diagnostica esta enfermedad.

4. Material y métodos

4.1. Metodología

Esta investigación conto con la metodología representada por medio de la siguiente Grafica en la Fig.1:

Fig. 1. Etapas de la metodología utilizada.

En esta metodología se hicieron los siguientes pasos:

1. Se extrajo la información de consultas de cada uno de los 65 centros de salud urbanos y rurales pertenecientes al ISSEA, la cual se concentró en una sola base de datos.
2. Se preprocesa la información reunida en la base de datos, eliminando registros incompletos, con error de captura además se eliminaron variables con datos fuera de rango, nulos y con datos sensibles de los pacientes. Se identificaron los pacientes con diagnostico confirmado de diabetes mellitus y los pacientes sin diagnóstico de diabetes y sin diagnósticos de alteraciones en glucosa.
3. Se procesa la información por medio del Mecanismo de Clasificación.
4. Se analizaron y presentaron resultados.
5. Se establecieron conclusiones.

4.2. Preparación de la información

Se comenzó filtrando la información de todos pacientes, dejando los confirmados diagnosticados con diabetes mellitus y sin diagnóstico de diabetes mellitus su historial y/o enfermedades relacionadas con alteraciones de glucosa.

Se contó inicialmente con 18 variables de signos vitales y métricas de la consulta, identificación geográfica y de derechohabiencia del paciente y consulta, de las cuales se consideraron solo 12 variables ya que se excluyeron consultas con datos incompletos, nulos y variables con datos sensibles.

La información de consultas obtenida de los centros de salud se concentró en una única base con datos, reuniendo más de 100,000 consultas de todo tipo de patologías, de las cuales se extrajo una muestra de 500 pacientes confirmados con diabetes mellitus y otra de 500 pacientes sin diagnóstico de diabetes mellitus en su historial ni diagnósticos relacionados con alteraciones de glucosa.

4.3. Discretización de la información

Las 12 variables consideradas para esta investigación se enlistan a continuación.

Derechohabiencia: esta variable señala la institución de pertenencia de prestaciones de servicios de salud a la que el paciente es derechohabiente esta variable está categorizada por el Instituto Nacional de Estadística y Geografía (INEGI) de la siguiente manera: 0.- Población Abierta, 1.-Seguro Popular, 2.-IMSS, 3.-ISSSTE, 4.-PEMEX, 5.-SECMAR, 6.-Otra y 7.-SMNG. Este dato es de índole socio económica ya que algunos centros de saludo tienen tendencias marcadas hacia alguna derechohabiencia dependiendo de su ubicación.

Sexo: esta variable representa el género del paciente y está definida como 1.-Masculino y 2.- Femenino.

Municipio de residencia: es el número de municipio catalogado por el INEGI de la siguiente manera, 1.- Aguascalientes 2.-Asientos 3.-Calvillo, 4.-Cosio, 5.-Jesús María, 6.-Pabellon de Arteaga, 7.-Rincon de Romos, 8.-San José de Gracia, 9.-Tepezala, 10.-El Llano, 11.-San Francisco de los Romo. A través de esta variable se puede saber la incidencia poblacional municipal, así como el distanciamiento con respecto a la capital del estado.

Jurisdicción Sanitaria: es una clasificación de las tres jurisdicciones existentes en el estado de Aguascalientes, 1.- Jurisdicción I, 2.- Jurisdicción II y 3.- Jurisdicción III. Con esta variable se agrupan los municipios como tres únicos grupos, así como medio de análisis presupuestal ya que cada uno cuenta con su propio presupuesto anual.

Existencia de antecedentes familiares con diabetes: es una variable booleana la cual señala con 1 cuando el paciente tiene antecedentes y con 0 cuando el paciente no tiene antecedentes.

Nivel de glucemia: es el nivel de riesgo en salud del paciente determinado por la cifra de azúcar o glucosa en una muestra de sangre. el ISSEA la clasifica de la siguiente manera, 1.-Sin Riesgo, 2.- Normal en control, 3.- Con Riesgo.

Estado Civil: esta variable es la categoría de un individuo según el registro civil en función de si tiene o no pareja o su situación legal al respecto. El INEGI la cataloga de la siguiente manera, 1.-Soltero, 2.-Casado, 3.-Union Libre, 4.-Separado, 5.-Divorciado, 6.-Viudo, 7.-Desconocido.

Índice de Masa Corporal: (IMC) es un indicador simple de la relación entre el peso y la talla de un paciente, se utiliza frecuentemente para identificar si se tiene sobrepeso u obesidad. Se calcula dividiendo el peso de una persona en kilos por el cuadrado de su talla en metros (kg/m2). Se utilizó la clasificación establecida por el ISSEA: 1.-Desnutrido, 2.-Normal, 3.-Sobre Peso, 4.- Obesidad.

Complicación: esta variable confirma si el paciente ha tenido alguna complicación en general durante su historial clínico y se describe con 1 con complicación y 0 sin complicación.

Edad: Es la edad del paciente en años. Para esta variable se consideró la clasificación propuesta por Mansilla. A [21] 1.-joven (< 24 años), 2.- Adulto (>= 24 años < 65 años), 3.- Senecto (>= 65 años < 75 años), 4.-Anciano (>= 75 años).

Tensión arterial baja y tensión arterial baja: Casiano Parra ordena los niveles de tensión arterial en el ser humano de la siguiente: 1.-Tensión Optima, 2.-Tensión Normal, 3.- Tensión Normal Alta, 4.-Hipertensión Grado 1, 5.-Hipertensión Grado 2, 6.-Hipertensión Grado 3, 7.- Hipertensión Grado 4 [22].

En la Tabla 1, se listan las variables utilizadas en la investigación.

Tabla 1. Variables utilizadas.

Variable	Nivel	Dominio	Criterio de comparación
Derechohabiencia	Poblacional	0,1,2,3,4,5,6,7	Igualdad Estricta
Sexo	Poblacional	1,2	Igualdad Estricta
Municipio	Poblacional	1,2,3,4,5,6,7,8,9,10,11	Igualdad Estricta
Jurisdicción	Poblacional	1,2,3	Igualdad Estricta
Ant. Fam. Diab.	Clínico	0,1	Igualdad Estricta
Nivel Glucemia	Clínico	1,2,3,4,5,6	Igualdad Estricta
Estado Civil	Poblacional	1,2,3,4,5,6,7	Igualdad Estricta
IMC	Clínico	1,2,3,4	Igualdad Estricta
Complicación	Clínico	0,1	Igualdad Estricta
Edad	Poblacional	1,2,3,4	Igualdad Estricta
Tensión A. Baja	Clínico	1,2,3,4,5,6,7	Igualdad Estricta
Tensión A. Alta	Clínico	1,2,3,4,5,6,7	Igualdad Estricta

4.4. Mecanismo de clasificación

Se desarrolló un mecanismo de clasificación compuesto por dos mecanismos secuenciales, el primer mecanismo fue testores típicos para selección subconjuntos de características, el segundo mecanismo integrado fue un clasificador basado en redes neuronales artificiales.

5. Resultados

Como parte del resultado se diseñó, desarrollo e implemento el siguiente marco de trabajo para el mecanismo de clasificación ilustrado en la Fig. 2.

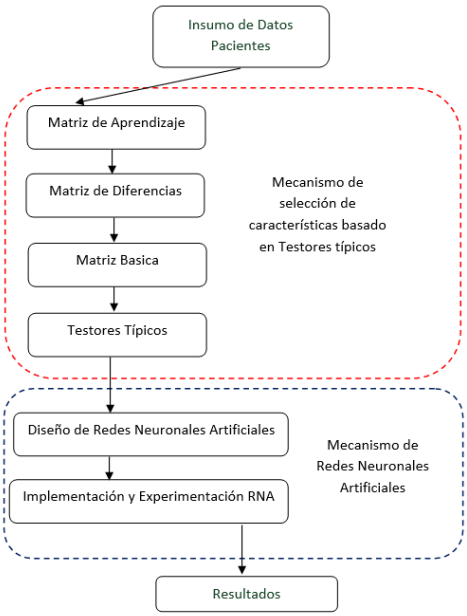


Fig. 2. Etapas de marco de trabajo.

El mecanismo de clasificación está integrado por dos mecanismos:

El primer mecanismo fue testores típicos el cual selecciona subconjuntos de características mediante los siguientes pasos:

1. Se obtuvo la Matriz de aprendizaje a partir de los datos discretizados de las consultas de pacientes de las dos clases establecidas.
2. Se generó la Matriz de diferencias.
3. Se generó la Matriz Basica.
4. A partir de esa matriz básica se obtuvieron todos los testores y se identificó solo a los testores típicos.

Los testores típicos poseen como ventaja la dependencia total de los datos, no tienen límite de clases y de características a analizar, no solo determinan las variables importantes estos además determinan el peso informacional de cada una de las variables de importancia.

Se implementó una selección de características haciendo uso de los testores típicos asociados con la matriz de entrenamiento extraída de un archivo histórico de pacientes con la patología de diabetes, de pacientes que no la padecen de los cuales se analizaron un total de 12 variables que pueden ser consultadas en la Tabla 2.

Se utilizaron 500 registros de la Clase Diabéticos (1) y 500 registros de la clase No Diabéticos (0).

Los datos se procesaron mediante la técnica de testores típicos la cual fue desarrollada en RadStudio 10.4 en el lenguaje C++, y se obtuvieron los siguientes 5

Tabla 2. Variables iniciales para testores.

Todas las Variables	
Variables	**Dato**
1	Derechohabiencia
2	Sexo
3	Municipio
4	Jurisdicción
5	Ant. Fam. Diab
6	Nivel Glucemia
7	Estado Civil
8	IMC
9	Complicación
10	Edad
11	Tensión A. Baja
12	Tensión A. Alta

Tabla 3. Testores típicos obtenidos.

Testor 1	Sexo	Glucemia	Edo. civil	Imc	Comp.	Tensión Alta
Peso inf.	40%	80%	100%	60%	100%	40%
Testor 2	Glucemia	Estado civil	Imc	Comp.		
Peso inf.	80%	100%	60%	100%		
Testor 3	Ant. Fam. Diab.	Estado civil	Comp.	Tensión Alta		
Peso inf.	40%	100%	100%	40%		
Testor 4	Sexo	Glucemia	Edo civil	Imc	Comp.	
Peso inf.	40%	80%	100%	60%	100%	
Testor 5	Ant. Fam. Diab.	Glucemia	Edo civil	Comp.	Edad	
Peso inf.	40%	80%	100%	100%	40%	

testores típicos o subgrupos de características que se muestran en las siguiente Tabla 3. En éstas se presentan las variables que integran el testor y su peso informacional.

Para el mecanismo de clasificación de pacientes diabéticos se utilizaron redes neuronales artificiales ya que son clasificadores superiores o excelentes reconocedores de patrones, estas son usadas donde no funcionan las técnicas tradicionales, manejan un amplio rango de datos, exclusiones y entradas deficientes. Se diseñó e implementó una red neuronal artificial densamente interconectada con una sola capa oculta para procesar la información de cada uno de los testores típicos, tomado como neuronas de

entrada cada una de las variables, implementando la ecuación 1 para calcular las neuronas de la capa oculta la cual propuesta por Piedra Fernández [18] , este autor propone 2 ecuaciones para capas ocultas, una para una sola capa y otra para 2 capas ocultas. Como neurona de salida la predicción de la clase diabético o no diabético:

$$h = n - m\,, \tag{1}$$

donde: $h =$ número de neuronas de la capa oculta, $n =$ número de neuronas de entrada, y $m =$ número de neuronas de salida.

Las redes neuronales se desarrollaron en Python 3.0 utilizando las librerías Keras y Tensorflow. Los parámetros utilizados para el proceso de entrenamiento de las redes neuronales se utilizaron 100 registros de la clase diabéticos y 100 registros de la clase no diabéticos, aplicando activación relu tanto para las neuronas en la capa de entrada como en la capa oculta y activación sigmoide en la capa de salida, en el ajuste se procesó por 80 épocas y lotes de 15, para la fase de validación se utilizaron 30 registros de pacientes diabéticos y 30 registros de pacientes no diabéticos, dado que la selección de variables conlleva un proceso de aprendizaje con el total de los elementos de pacientes con y sin diabetes; las cantidades de registros de pacientes utilizadas para los procesos de aprendizaje y validación del clasificador fueron menores. La configuración topológica y los resultados de precisión de test de cada una de las redes neuronales se presentan en la Tabla 4.

Tabla 4. Configuración de redes neuronales artificiales para testores típicos.

Testor	Variables	Nodos Entrada	Nodos Oculta	Nodos Salida	Precisión de Test
1	6	6	5	1	0.5
2	4	4	3	1	0.883
3	5	5	4	1	0.883
4	5	5	4	1	0.883
5	5	5	4	1	0.916

Analizando los resultados de los 5 clasificadores desarrollados, el mejor resultado es el observado por la red neuronal artificial procesando las variables del Testor 5, el cual obtiene casi el 92% de precisión del test, determinando a las variables Tabla 5 *antecedentes familiares con diabetes, nivel de glucemia, estado civil, existencia de complicación y la edad*, en conjunto como las más importantes para la clasificación de diabetes en la población analizada, las cuales fueron analizadas.

Una de las variables con mayor peso informacional es la de estado civil presenta información relevante ya que se encontró que el 48% de los pacientes diabéticos son de la categoría *casados* y que solo el 22% de diabéticos son de la categoría *solteros*, el 30% restante pertenece a las otras clases restantes de estado civil. La otra variable con el mayor peso informacional fue la de existencia de complicación, en esta variable se muestra que el 90% de los pacientes con diabetes pertenecen a la clase *con complicación* y solo el 10% de los pacientes diabéticos pertenecen a la clase *sin*

complicación. Respecto al nivel de glucemia se pudo notar que el solo el 17% de la población con diabetes corresponde a la categoría *sin riesgo* y el 83% de la población con diabetes presenta categoría *con riesgo*.

En la variable edad los casos de pacientes diabéticos se centran la mayoría en las clases *adulto* y *senecto* que va de los rangos de los >= 24 años a los < 75 años, estos se dividen en 61% de diabéticos en la clase *adulto* y 28% en la clase *senecto*, el otro 11% pertenecen al resto de las clases de edad.

Tabla 5. Peso Informacional de Testor de mejor precisión.

Variable	Descripción	Peso Inf.
1	Antecedentes familiares con diabetes	40%
2	Nivel de glucemia	80%
3	Estado civil	100%
4	Existencia de complicación	100%
5	Edad	40%

Por otro lado, la variable con antecedentes familiares con diabetes presenta un 67% de pacientes con diabetes de la clase *con antecedentes familiares* y un 33% de los pacientes con diabetes son de la clase *sin antecedentes*.

6. Conclusiones

Se encontró con éxito el subconjunto de características ponderadas que mejor describen a un paciente con diabetes mellitus en la población de Aguascalientes clasificado con diagnósticos del CIE10, se diseñó e implementó creó y se puso a punto un clasificador basado en redes neuronales artificiales con una precisión del 91.6%.

Se puede concluir que no se debe prescindir de las variables de estado civil y existencia de complicación ya que fueron las de mayor importancia (100% de peso informacional), además de reafirmar la importancia de las variables nivel de glucemia, antecedentes de familiar con diabetes y la edad, ya que en conjunto las 5 variables del testor resultante considera las relaciones entre todas las variables para identificar pacientes con diabetes mellitus.

Se descubrió que el estado civil tiene una relación muy directa en la descripción de un paciente con diabetes, y queda comprobado que si es una variable importante ya que cuenta con un peso informacional de 100% y la cual en su categoría casado tiene una alta incidencia de pacientes con diabetes y mucho menor incidencia en la categoría soltero, posiblemente la alta incidencia es detonada por los altos niveles de estrés y el marcado sedentarismo que presentan los pacientes de la categoría casado en la población de Aguascalientes.

Se concluye que las variables: historial de familiar con diabetes y la existencia de complicación tienen una alta relación, ya que cuando estas se encuentran en su categoría

con familiar y con complicación, de un total de 65 pacientes 62 fueron diagnosticados con la patología de diabetes. Cuando estas variables se presentan en sus categorías sin familiar y sin complicación de un total de 45 pacientes solo 5 fueron diagnosticados con diabetes.

Se rechaza la creencia respecto a que en conjunto las variables: Tensión arterial, nivel de glucemia, el índice de masa corporal y el género del paciente son los factores determinantes de diabetes mellitus en la población de Aguascalientes.

Para trabajos futuros se está considerando el integrar este tipo de clasificadores como recomendadores auxiliares asistiendo a médicos en el sistema de consultas de especialidad en el módulo de enfermedades crónico-degenerativas, el cual es uno de los módulos de mayor uso del Expediente Clínico Electrónico del Instituto de Servicios de Salud.

Referencias

1. OPS: Clasificación internacional de las enfermedades. Organizacion panamericana de la salud. http://www.paho.org/hq/index.php?option=com_content&view=article&id=3561%253Aclasificacion-internacional-enfermedades-cie&catid=2641%253Acha-clasificacion-internacional-enfermedades-cie&Itemid=2560&lang=en (2015)
2. INEGI: Estadísticas a propósito del día mundial de la diabetes (2011)
3. Herrera,F.: Introducción a los Algoritmos Metaheurísticos. Soft Comput. Intell. Inf. Syst. (2009)
4. Jensen, R., Shen, Q.: Computational intelligence and feature selection : rough and fuzzy approaches (2008)
5. Liu, H., Motoda, H.: Computational methods of feature selection (2008)
6. Bermingham, M.L., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., Wright, A.F., Wilson, J.F., Agakov, F., Navarro, P., Haley, C.S.: Application of high-dimensional feature selection: evaluation for genomic prediction in man. Sci. Rep., 5, pp. 1–12 (2015)
7. Casella, G., Fienberg, S., Olkin, I.: An Introduction to Statistical Learning (2013)
8. Cheguis, I.A., Yablonskii, S.V.: About testors for electrical outlines. Uspieji Mat. Nauk, 4, pp. 188–184 (1955)
9. Alba, C., Santana, R., Ochoa, R., Lazo, C.M.: Finding typical testors by using an evolutionary strategy. In: Proc. V Iberoam. Work. Pattern Recognit., pp. 267–278 (2000)
10. Torres, M., Ponce, E., Ochoa, C., Torres, A., Díaz, E.: Mecanismos de aceleración en selección de subconjuntos de características basada en el peso informacional de las variables para aprendizaje no supervisado (2009)
11. Soto-Torres, M.D., Soto-Torres, A., Sentí, E.P.L.: Algoritmo genético y testores típicos en el problema de selección de subconjuntos de características. Sist. Cibernética E Informática, 3(2), pp. 1–5 (2006)
12. Shulcloper, J.R., Alba, C., Lazo, C.: Introducción a la teoría de testores típicos (1995)
13. Santiesteban, A., Pons, P., Lex, A.: Un nuevo algoritmo para el cálculo de los testores típicos. Revista Ciencias Matemáticas V, 21, pp. 88–95 (2003)
14. Torres, M.D.: Metaheurísticas híbridas en selección de subconjuntos de características para aprendizaje no supervisado. Universidad Autónoma de Aguascalientes (2010)
15. Marín-Diazaraque, J.M.: Introducción a las redes neuronales aplicadas Conceptos básicos, 3 (2003)
16. Vélez, D.: Estructura y uso del CIE10 (2016)
17. Karzulovic, L. García, P., Wozniak, A., Villarroel, L., Hirsch, T., Concha, I., Catalán, S., Cifuentes, L.: Una regla de predicción clínica ¿anticipa el diagnóstico de la faringitis

estreptocóccica en niños de 2 a 15 años?. Rev. Chil. infectología, 35(5), pp. 476–482 (2018)
18. Piedra Fernández, J.A.: Aplicación de los sistemas neurodifusos a la interpretación automática de imágenes de satélite. Universidad de Almeria (2008)

ISSN 1870-4069

Redes neuronales artificiales y árboles de decisión para la clasificación con datos categóricos

C. Guadalupe Origel-Rivas, Eréndira Rendón-Lara,
Itzel María Abundez-Barrera, Roberto Alejo-Eleuterio

Tecnológico Nacional de México,
Instituto Tecnológico de Toluca,
México

{corigelr, erendon, iabundezb,
ralejoe}@toluca.tecnm.mx

Resumen. Las redes neuronales artificiales (RNA) y los árboles de decisión (AD) se han utilizado como clasificadores en distintas áreas, como en el sector salud, financiero, manufactura, social entre otros. Sin embargo, cada uno de estos clasificadores tiene sus ventajas dependiendo del tipo de datos (numérico, categórico o mezclados) de entrada. Por lo tanto, los árboles de decisión trabajan con valores tanto numéricos como categóricos. Teóricamente ambos clasificadores pueden trabajar con todos los tipos de datos, sin embargo, en la práctica no parecer ser así. Por esta razón en este trabajo se realiza un análisis de las ventajas y desventajas de estos clasificadores para valores categóricos. Se utilizaron bases de datos descritas en atributos de tipo categórico, que fueron codificadas, para la RNA. De acuerdo con las pruebas realizadas la eficiencia de clasificación cuando se utilizó la RNA es inferior al clasificador C4.5.

Palabras clave: Redes neuronales artificiales, árboles de decisión, codificación.

Artificial Neural Networks and Decision Trees for Classification with Categorical Data

Abstract. Artificial neural networks (ANN) and decision trees (DT) have been used as classifiers in different fields, such as in the health, financial, manufacturing, and social sectors, among others. However, each classifier has its advantages depending on the type of data (numeric, categorical or mixed) input. Nevertheless, decision trees work with both numerical and categorical values. Theoretically, both classifiers can work with all types of data, however, in practice they do not appear to be so. For this reason: in this work we analyze advantages and disadvantages of these classifiers for categorical values. Databases described in categorical attributes, which were encoded, were used for the ANN. According to the tests carried out, the classification efficiency when the ANN was used is lower than the C4.5 classifier.

Keyword: Artificial neural networks, decision trees, coding.

1. Introducción

El aprendizaje automático es un método de análisis de datos que automatiza la construcción de modelos analíticos. Mediante el uso de algoritmos que aprenden de manera iterativa de los datos, el aprendizaje automático permite a las computadoras encontrar conocimiento oculto sin necesidad de programar explícitamente dónde buscar [1].

Las Redes Neuronales Artificiales (RNA) y los Árboles de Decisión (AD) son algoritmos de aprendizaje automático, que han sido utilizados como clasificadores en diversos sectores: industrial, financiero, salud, manufactura, social entre otros, debido a la capacidad y funcionalidad de estos en el procesamiento y entendimiento de la información. En la mayoría de la literatura actual sobre métodos de clasificación de datos se enfoca principalmente a valores numéricos. Sin embargo, en muchos problemas reales la información se presenta de forma no numérica (categórica). Los datos categóricos son un desafío, debido a sus propiedades inherentes especiales de los datos que describen. Por ejemplo, el atributo categórico, "*color"* puede tener diferentes valores como rojo, verde, azul, etc. Estos valores por su naturaleza son inherentes en ciertas propiedades incertidumbre, ambigüedad y vaguedad, esto hace que la clasificación sea más complicada que cuando se trabaja con valores numéricos [2]. El buen entendimiento de los datos es importante para realizar un análisis correcto de la información y obtener los resultados interesantes.

En el campo de aprendizaje automático es muy popular el uso de la RNA, por los resultados satisfactorios que se han obtenido, sin embargo, cuando la naturaleza de los datos no es numérica, se presentan algunos problemas, es decir es necesario aplicar métodos de trasformación de valores categóricos a valores numéricos lo que no siempre resulta ser lo apropiado. Es decir, trabajar las RNA cuando se tienen datos categóricos, estos deben de ser codificados.

En la presente investigación se presenta un análisis del efecto que se tiene al realizar una transformación de los datos para trabajar con RNA y los resultados obtenidos son comparados con los obtenidos en un clasificador que no realiza trasformación de datos (Un árbol de decisión).

Este trabajo está organizado de la siguiente manera: En la sección II se presenta algunos trabajos relacionados con la utilización redes neuronales con datos categóricos y los árboles de decisión. En la sección III se proporcionan los elementos teóricos relacionados con la propuesta que se presenta. En la sección IV se detalla los pasos que se siguieron en esta investigación. En la sección V se presenta la experimentación realizada y finalmente en la Sección VI se proporcionan las conclusiones a las cuales se llegaron, así como los trabajos futuros.

2. Estado del arte

Las redes neuronales artificiales han sido implementadas como clasificadores para resolver distintos problemas por ejemplo Rajkamal Kesharwani [3] utilizó una RNA en un proceso de manufactura de herramientas de mandril en el área de calidad para clasificarlas en: desperdicio, retrabajo y utilizable.

Donde la red neuronal se programó en Matlab, diseñada para una clasificación binaria de dos fases de los datos, en la primera fase evalúa si la parte en consideración es utilizable y la segunda fase de la clasificación se centra en los datos restantes de la fase 1 (datos no utilizables). Obteniendo un 98% de exactitud en el proceso de clasificación de las herramientas. Se observa que las redes fueron implementadas para la solución de un problema de clasificación. A pesar de ello, no menciona que tipos de datos se utilizaron.

Otra investigación es la de Thomas Küfner [4], donde utilizó una red neuronal supervisada para la clasificación de los estados operativos en las plantas de producción de sistemas embebidos basados en 6 mediciones (características de los estados operativos), las cuales fueron: corriente nominal, potencia nominal, velocidad de rotación nominal, factor potencia, eficiencia y eficiencia de clase. Utilizó 250 mediciones para cada estado de operación en un perceptrón multicapa donde la capa de entrada contiene 701 neuronas, la primera capa oculta tiene 500 neuronas, la segunda capa oculta 150 neuronas y la capa de salida 6 neuronas que corresponden a los estados operativos. Siendo uno de los métodos más costo-efectivo que otros, estas redes tuvieron una clasificación del 99.82% demostrando que esta red es capaz de reconocer los estados operativos de una maquinaria de producción confiable en un ambiente de manufactura. En esta investigación los datos utilizados fueron de tipo numérico por lo que no fue necesaria ninguna codificación en los datos.

Por otro parte Harsh Patel [5] menciona que los algoritmos de clasificación de árboles de decisión obtienen mejores resultados en términos de exactitud, tiempo y precisión que otros algoritmos de clasificación. Además, realiza un estudio comparativo entre los diferentes tipos de algoritmos (ID3, C4.5 y CART) utilizando una base de datos de "*evaluación de automóviles*" para estimar cual es el mejor para este proceso. La exactitud de los resultados se encuentra entre 97.11% y 89.3%.

Sungsu Choi [6] en su investigación identifica productos defectuosos en un proceso de manufactura de equipo electrónico, para ello utilizaron el algoritmo C4.5 únicamente para tres procesos. El resultado de haber construido al árbol de decisión fue que se encontraron las causas que ocasionan que los productos sean considerados como defectuosos.

3. Aspectos teóricos

3.1. Redes neuronales artificiales

Las redes neuronales artificiales están formadas por una serie de procesadores elementales, denominados neuronas artificiales, unidades o nodos, que constituyen dispositivos simples de cálculos que, bien a partir de un vector de entrada procedente del mundo exterior, o bien a partir de estímulos recibidos de otras neuronas, proporcionan una respuesta única (salida) [13]. Hay tres tipos de unidades en las redes neuronales: de entrada, salida y ocultas. Las unidades de entrada reciben señales desde el entorno (entradas a la red), las unidades de salida envían la señal fuera del sistema (salidas de la red), las unidades ocultas son aquellas cuyas entradas y salidas se encuentran dentro del sistema, no tienen contacto con el exterior.

En términos matemáticos se puede definir una neurona *k* por medio de ec.1, ec.2:

$$u_k = \sum_{j=i}^{m} w_{kj} x_j \,, \tag{1}$$

$$y_k = \varphi(u_k + b_{k)} \,, \tag{2}$$

donde $x_1, x_2 \ldots x_m$ son los valores de las señales de entrada: $w_1, w_2 \ldots w_m$ son los pesos sinápticos de la neurona k; u_k es la suma del producto de los pesos sinápticos y los valores de la señal de entrada; b_k es el Bias; φ es la función de activación; y y_k es el valor de la salida de la neurona. La función de activación es la que determina que una neurona sea activa o pasiva de acuerdo con su valor de salida [8].

3.2. Codificación

Las variables en escala nominal no tienen valor cuantitativo, es decir, son variables puramente cualitativas. Algunos ejemplos de datos nominales son variables como el género (masculino, femenino), el estado civil (soltero, casado, divorciado) simplemente denotan categorías [9].

Existen varios métodos de codificación para convertir los datos en valores numéricos, como la Codificación Activa y Codificación Entera.

Codificación activa

Es el esquema de codificación más utilizado. Compara cada nivel de la variable categórica con un nivel de referencia fijo. Una codificación activa transforma una sola variable con *n* observaciones y *d* valores distintos, en *d* variables binarias con *n* observaciones cada una. Cada observación indica la presencia (1) o ausencia (0) de la variable dicotómica [10]

Codificación ordinal o entera

En esta codificación se asigna un número entero a cada categoría, siempre que se conozca el número de categorías existentes, no agrega ninguna columna nueva a los datos, pero implica un orden a la variable que puede no existir realmente [11].

3.3. Árboles de decisión

Los árboles de decisión (AD) son una técnica inductiva mixta que consiste en crear una estructura de árbol que clasifica un conjunto de datos (objetos o ejemplos) en un cierto número de clases, de tal manera que a posteriori pueda clasificar nuevos objetos que le sean presentados, es decir, a partir de conjuntos de objetos iniciales (Muestra de entrenamiento) se construye un árbol que permitirá clasificar o predecir a que clase pertenece a los nuevos objetos que se le van presentando. Los AD constan de dos elementos básicos: los nodos de decisión y los nodos hoja. Los nodos de decisión son aquellos que representan una característica del objeto (por ejemplo, la característica "color") de estos nodos salen tantas ramas como valores tenga la característica (rojo, azul, …) que a su vez se van a conectar a otro nodo de decisión o a un nodo hoja. Los nodos hoja son los nodos finales (último nivel del árbol) y representan la etiqueta de la clase a la que pertenece el objeto.

En la literatura se pueden encontrar diferentes algoritmos para la construcción de un árbol, uno de ellos es el algoritmo ID3 (Iterative Dichotomiser) Dichotomiser Iterativo propuesto en 1986 por Quinlan Ross [12]. El cual trabaja cuando la muestra de entrenamiento tiene dos clases. Otro algoritmo es el C4.5 es utilizado para construir un árbol de decisión también propuesto por Ross Quinlan, que es una generalización del algoritmo ID3, es decir trabaja para *n* clases. Este algoritmo utiliza como criterio de división la *ganancia de la información,* que es una propiedad estadística que mide que tan bien una determinada característica separa los objetos o ejemplos de la muestra de entrenamiento, también puede trabajar con datos numéricos, categóricos o ambos [12].

Para el cálculo de la ganancia *de la información es* necesario calcular la entropía que permite medir la impureza de la muestra de entrenamiento. Los pasos básicos del algoritmo C4.5 son sencillos, ya que se pueden aplicar recursivamente en cada nodo, así el algoritmo inicia seleccionado de entre todas las características al de mayor "Ganancia" que será seleccionado como nodo raíz, que a su vez este tendrá salidas como valores tenga la característica. De esta manera el proceso se repite de igual manera para una de las características hasta llegar a los nodos hojas que serán las clases.

4. Metodología

Los pasos que se siguieron para la realización de este trabajo se presentan en la figura 1. Seguidamente se describen detalladamente cada uno de los pasos.

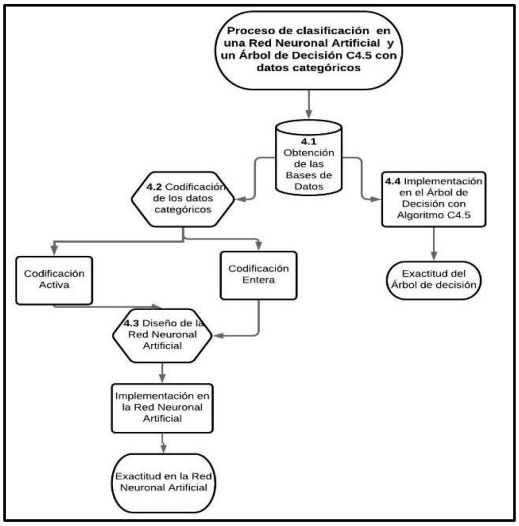


Fig. 1. Metodología.

4.1. Obtención de las bases de datos

Cuatro de las bases de datos utilizadas en este trabajo fueron adquiridas del repositorio de datos UCI "*Machine Learning Repository*" (archive.ics.uci.edu), base de datos "*Defectos en autos*" fue adquirida directamente de una planta automotriz:

a) *Evaluación de carros*: la cual está descrita en 6 características categóricas las cuales son: Compra, Mantenimiento, Puertas, Personas, Arranque y Seguridad con 4 clases (Accesible, Inaccesible, Bueno y Muy bueno) con un tamaño 1728 automóviles (objetos).

b) *Defectos en autos*: se describe en 4 características categóricas las cuales son: Modulo, Coordenada, Modelo y Tipo de defecto, con 3 clases (Ensamble, Carrocerías y Pintura) con 543 objetos.

c) *Juego de ajedrez*: se describe en 36 características categóricas, las cuales representan las posiciones en el juego, con dos clases (ganar o perder) con 3196 jugadas (objetos).

d) *Base de Juego xox:* cuenta con 9 atributos categóricos para describirlo, que son las posiciones del juego, con dos clases (juego positivo o juego negativo), con 958 juegos(objetos).

e) *Cáncer de pecho:* Se describe en 5 características de tipo categórico (Menopausia, Nodos-capas, Pecho, Irradiación) y 4 de tipo numérico (Edad, Tamaño del tumor, Nodos inv, Maligno), con dos clases (recurrente y no-recurrente) con 286 diagnósticos (objetos).

4.2. Codificación de las bases de datos

Se realizaron las codificaciones de los conjuntos de datos utilizando dos métodos, la codificación entera y la codificación activa.

Codificación entera

En las Tablas 1-5 se presenta la codificación entera, en cada una de las tablas se puede ver dos columnas por cada característica, donde la primera columna es el código que le corresponde al valor del dominio de la característica.

Tabla 1. Conjunto de datos "*Evaluación de carros*".

Codificación Entera											
Compra		**Mantenimiento**		**Puertas**		**Personas**		**Arranque**		**Seguridad**	
1	Muy A	**1**	Muy Alta	**1**	Dos	**1**	Dos	**1**	Bajo	**1**	Alta
2	Alta	**2**	Alta	**2**	Tres	**2**	Cuatro	**2**	Medio	**2**	Medio
3	Medio	**3**	Medio	**3**	Cuatro	**3**	Más	**3**	Alto	**3**	Baja
4	Baja	**4**	Baja	**4**	Más						

Tabla 2. Conjunto de datos "*Defectos en autos*".

Codificación Entera							
Módulo		**Coordenada**		**Modelo**		**Tipo de Defecto**	
1	4to D	**1**	A10	**1**	C	**1**	Abollado
2	4tO I	**2**	A11	**2**	L	**2**	Brisa
3	5ta P	**3**	A7	**3**	F	**3**	Contam
40	SI	**41**	K15			**11**	Repegón

Tabla 3. Conjunto de datos *"Juego de ajedrez".*

Codificación Entera											
Posición1		**Posición2**		**Posición3**		**Posición4**		**Posición5**		**Posición36**	
1	F	**1**	F	**1**	F	**1**	F	**1**	F	**1**	N
2	T	**2**	T	**2**	T	**2**	T	**2**	T	**2**	T

Tabla 4. Conjunto de datos *"Juego xox".*

Codificación Entera											
Posición1		**Posición2**		**Posición3**		**Posición4**		**Posición5**		**Posición9**	
1	B	**1**	B	**1**	B	**1**	B	**1**	B	**1**	B
2	O	**2**	O	**2**	O	**2**	O	**2**	O	**2**	O
3	X	**3**	X	**3**	X	**3**	X	**3**	X	**3**	X

Tabla 5. Conjunto de datos *"Cáncer de pecho".*

Codificación Entera											
Edad		**Menopausia**		**Tamaño T**		**Nodos inv**		**NodosC**		**Irradiación**	
1	20-29	**1**	Ge40	**1**	0-4	**1**	0-2	**1**	X	**1**	X
2	30-39	**2**	It4	**2**	5-9	**2**	3-5	**2**	No	**2**	No
3	40-49	**3**	Premeno	**3**	10-14	**3**	6-8	**3**	Yes	**3**	Yes
5	70-79			**11**	50-54	**7**	24-26				

Tabla 6. Conjunto de datos *"Evaluación de carros".*

Codificación Activa											
Compra		**Mantenimiento**		**Puertas**		**Personas**		**Arranque**		**Seguridad**	
0001	Muy Alta	0001	Muy Alta	0001	Dos	001	Dos	001	Bajo	001	Alta
0010	Alta	0010	Alta	0010	Tres	010	Cuatro	010	Medio	010	Medio
0100	Medio	0100	Medio	0100	Cuatro	100	Más	100	Alto	100	Baja
1000	Baja	1000	Baja	1000	Más						

Codificación activa

En las Tablas 6-10 se presenta la codificación activa, en cada una de las tablas se puede ver dos columnas por cada característica, donde la primera columna es el código que le corresponde al valor del dominio de la característica.

Tabla 7. Conjunto de datos *"Defectos en autos"*.

Codificación Activa						
Módulo o Parte	**Coordenada**	**Modelo**			**Tipo de Defecto**	
00000000000000000000000000000001	4to D	00000000000000000000000000000001	A10	001	C	00000000001	Abollado
00000000000000000000000000000010	4tO I	00000000000000000000000000000010	A11	010	L	00000000010	Brisa
00000000000000000000000000000100	5ta P	00000000000000000000000000000100	A7	100	F	00000000100	Contam
10000000000000000000000000000000	SI	10000000000000000000000000000000	K15			10000000000	Repegón

Tabla 8. Conjunto de datos "*Juego de ajedrez"*.

Codificación Activa											
Posición1		**Posición2**		**Posición3**		**Posición4**		**Posición5**		**Posición36**	
01	F	01	F	01	F	01	F	01	F	01	N
10	T	10	T	10	T	10	T	10	T	10	T

Tabla 9. Conjunto de datos *"Juego xox"*.

Codificación Activa											
Posición1		**Posición2**		**Posición3**		**Posición4**		**Posición5**		**Posición9**	
001	B	001	B	001	B	001	B	001	B	001	B
010	O	010	O	010	O	010	O	010	O	010	O
100	X	100	X	100	X	100	X	100	X	100	X

Tabla 10. Conjunto de datos *"Cáncer de pecho"*.

Codificación Activa											
Edad		**Menopausia**		**Tamaño Tumor**		**Nodos inv**		**Nodos-C**		**Irradiación**	
000001	20-29	001	Ge40	00000000001	0-4	000001	0-2	001	X	001	X
000010	30-39	010	It4	00000000010	5-9	000010	3-5	010	No	010	No
000100	40-49	100	Premeno	00000000100	10-14	000100	6-8	100	Yes	100	Yes
.	.			.	.	.	.				
100000	70-79			10000000000	50-54	100000	24-26				

4. 3. Diseño de la red neuronal artificial

La Red Neuronal Artificial utilizada se presenta en la figura 2, donde la estructura de la RNA es utilizada es la misma solo cambiando las entradas y salidas debido a las diferentes características y clases.

Todas las bases de datos se codificaron con los dos métodos anteriormente expuestos para ser implementados en la RNA en Python, la red para la clasificación mostrada en la figura 4.1, representa las neuronas de entrada dependiendo de las características que tenga la base de datos a utilizar, las capas ocultas con 4,10,10 neuronas y la capa de salida con neuronas las cuales son las clases. En la sección V se proporciona más detalle.

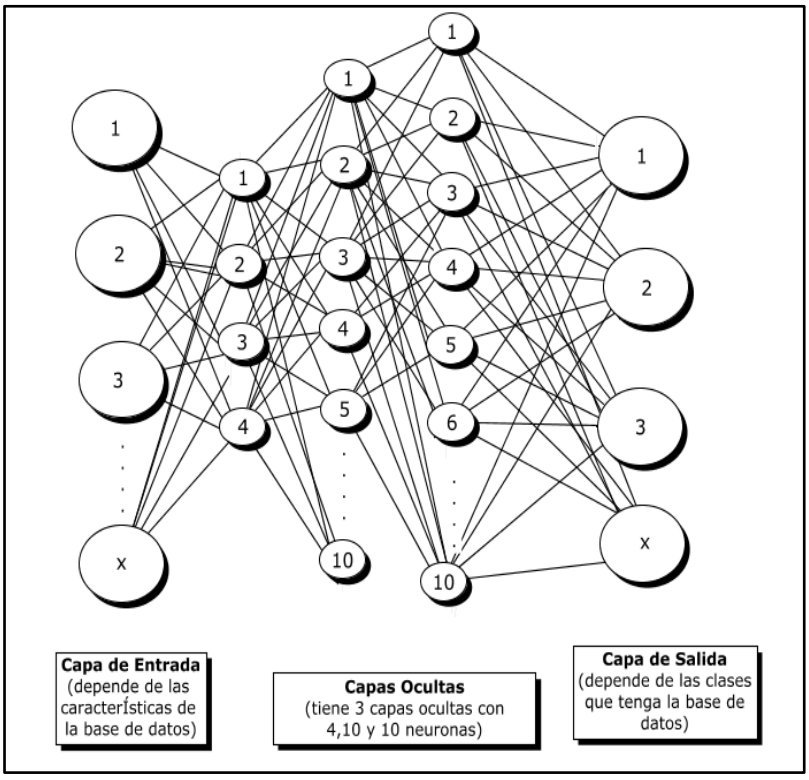


Fig. 2. Red Neuronal Artificial.

4.4. Árbol de decisión con el algoritmo C4.5

Para cada uno de los conjuntos de datos se construyó su respectivo árbol de decisión utilizando el algoritmo C4.5, para esto se trabajó con el software Weka. Por razones de espacio solo se presenta el árbol de decisión obtenido de la base de datos "*evaluación de carros*", mostrado en la figura 3, también se puede observar que el nodo que obtuvo la mayor *"ganancia de información*" fue la característica *"seguridad"*.

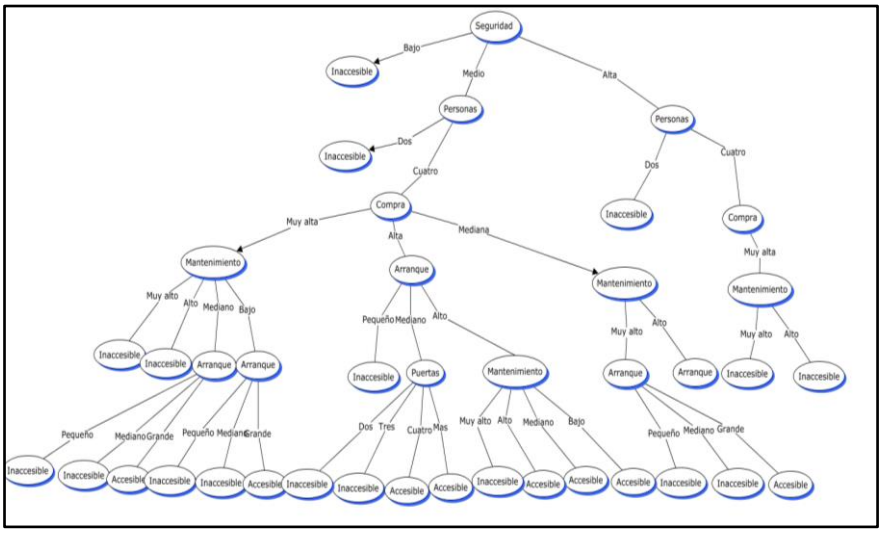


Fig. 3. Ejemplo de árbol de decisión C4.5 en la base evaluación de carros.

5. Experimentación

5. 1. Red neuronal artificial

Implementando dos redes neuronales con diferente número de neuronas en los datos categóricos con codificación entera se obtuvieron los resultados que se muestran en la figura 4. En la figura se indica la configuración de la RNA utilizada para cada una de las bases de datos, tamaño de la muestra de entrenamiento y prueba, así como la eficiencia obtenida. El número de neuronas en la capa de entrada de las redes son las características y en la capa de salida son las clases de cada una de las bases de datos. La función de activación para las capas ocultas fue RELU y para las capas de salida fue SOFTMAX en todas las bases de datos.

En el caso de la base de datos de "*Evaluación de carros*", se utilizaron 1210 objetos para entrenamiento y 518 objetos de prueba. Las redes se conformaron con 6 neuronas de entrada (características), las 3 capas ocultas generales (diferente número de neuronas) y 4 neuronas de salida que fueron las clases. Con esta codificación se obtuvo una exactitud del 68% y 74% de la clasificación.

En la base de "*Defectos en autos*" dado el tamaño, las redes neuronales se conformaron por 4 neuronas de entrada, 3 capas ocultas (con diferente número de neuronas en la red 1 y red 2) y 3 neuronas de salida, se utilizaron 380 objetos para el entrenamiento y 163 para la prueba de esta. Obteniendo una exactitud del 55% y 0.39% en el proceso de determinar la clase a la que pertenece.

Por otra parte, la base de datos de "*Juego de ajedrez*" obtuvo una exactitud del 93% y 83%, de un tamaño de 3196 objetos de los cuales se utilizaron 2237 para el entrenamiento y 959 de prueba, en una red conformada por 36 neuronas de en la capa entrada, 3 capas ocultas y 2 neuronas en la capa de salida.

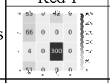
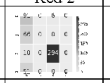
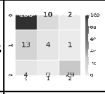
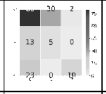
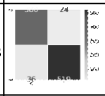
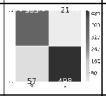
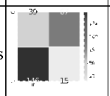
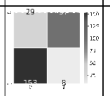
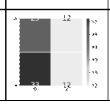
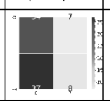
Codificación Entera								
Base de datos	**N° de neuronas en las capas oculta**		**Entrenamiento**	**Prueba**	**Matriz de confusión**		**Resultado de Exactitud**	
	Red 1	Red 2			Red 1	Red 2	Red 1	Red 2
Evaluación de Carro	4 10 10	6 4 9	1210 objetos	518 objetos			0.685	0.74
Defectos en autos	4 10 10	6 4 9	380 objetos	163 objetos			0.55	0.39
Juego de ajedrez	4 10 10	6 4 9	2237 objetos	959 objetos			0.93	0.83
Juego xox	4 10 10	6 4 9	671 objetos	287 objetos			0.16	0.1
Cáncer de Pecho	4 10 10	6 4 9	200 objetos	86 objetos			0.42	0.38

Fig. 4. Configuración de la RNA de los conjuntos de datos para "*Codificación entera*".

En la base de datos "*Juego xox*" se ocuparon 671 objetos para el entrenamiento y 287 objetos de prueba en una red compuesta por 9 neuronas de entrada, 3 capas ocultas (número de neuronas diferentes en cada red) y 2 neuronas de salida.

Teniendo una exactitud del 16% y 10% en el proceso de determinar a qué clase corresponde el objeto.

Por último, en la base de datos "*Cáncer de pecho*" se obtuvo una exactitud del 42% y 38% en una red con 9 neuronas de entrada, 3 capas ocultas y 2 neuronas de salida con 200 muestras de entrenamiento y 86 objetos de prueba. Se observa que las bases con exactitud alta entre 93% y 68% tienen más de 1000 muestras en ellas. Mientras que las otras bases tuvieron una exactitud menor de 60%.

Para la codificación activa se obtuvieron los resultados que se presentan en la figura 5. En la figura se indica la configuración de la RNA utilizada para cada una de las bases de datos, tamaño de la muestra de entrenamiento y prueba y eficiencia obtenida. El número de neuronas en la capa de entrada de las redes son las características y en la capa de salida son las clases de cada una de las bases de datos. La función de activación para las capas ocultas fue RELU y para las capas de salida fue SOFTMAX para todas las bases de datos.

En la base de datos de "*Evaluación de carros"*, las redes se utilizaron con 6 neuronas de entrada (características), las 3 capas ocultas generales (diferente número en las redes) y 4 neuronas de salida que fueron las clases. Se utilizaron 1210 objetos para entrenamiento y 518 objetos de prueba, la exactitud obtenida en ambas redes fue de 0%, debido a que no logro clasificar los datos y presento una falla en este proceso.

Para la base de datos *"Defectos en autos"* las redes neuronales utilizaron 380 objetos para el entrenamiento y 163 para la prueba en una red de 4 neuronas de entrada, 3 capas ocultas (cambia número de neuronas en las redes) y 3 neuronas de salida, presentando

un error al momento de realizar la clasificación, obteniendo una exactitud de 0% en ambas.

En la base de datos de "*Juego de ajedrez*" se obtuvo una exactitud del 90% y 92%,

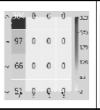
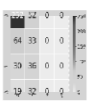
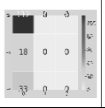
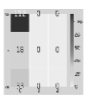
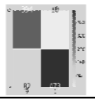
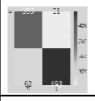
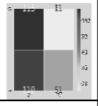
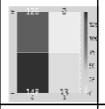
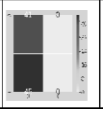
Codificación Activa								
Base de datos	**N° de neuronas en**		**Entrenamiento**	**Prueba**	**Matriz de confusión**		**Resultado de Exactitud**	
	Red 1	Red 2			Red 1	Red 2	Red 1	Red 2
Evaluación de Carro	4	6	1210 objetos	518 objetos			0	0
	10	4						
	10	9						
Defectos en autos	4	6	380 objetos	163 objetos			0	0
	10	4						
	10	9						
Juego de ajedrez	4	6	2237 objetos	959 objetos			0.9	0.92
	10	4						
	10	9						
Juego xox	4	6	671 objetos	287 objetos			0.52	0.28
	10	4						
	10	9						
Cáncer de Pecho	4	6	200 objetos	86 objetos			0	0
	10	4						
	10	9						

Fig. 5. Configuración de la RNA de los conjuntos de datos para *"Codificación Activa"*.

con 2237 objetos para el entrenamiento y 959 de prueba, en una red conformada por 36 neuronas en la capa entrada, 3 capas ocultas y 2 neuronas en la capa de salida. Probando que, de estos dos métodos de codificación, el mejor fue el de codificación Entera.

Por otra parte, la base de datos *"Juego xox"* obtuvo una exactitud de 52% y 28%, mostrando que la codificación Activa es mejor para esta base de datos. Se ocuparon 671 objetos para el entrenamiento y 287 objetos de prueba en las redes compuestas por 9 neuronas de entrada, 3 capas ocultas y 2 neuronas de salida

Por último, en la base de datos "Cáncer de pecho" se utilizaron dos redes con 9 neuronas de entrada, 3 capas ocultas (diferentes neuronas en la red 1 y red 2), 2 neuronas de salida con 200 objetos de entrenamiento y 86 muestras de prueba, los resultados obtenidos en la exactitud de este proceso para determinar a la clase que pertenece no fueron favorables y se tuvo una falla.

5. 2. Árbol de decisión con algoritmo C4.5

En la figura 6 se presentan los resultados de los árboles de decisión obtenidos de las bases de datos.

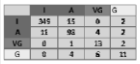
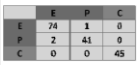
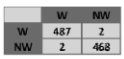
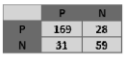
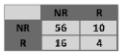
Bases de datos	Entrenamiento	Prueba	Matriz de confusión	Resultado
Evaluación de Carros	1210 objetos	518 objetos		0.9
Defectos en autos	380 objetos	163 objetos	Matriz de confusión 	0.981
Juego de ajedrez	2237 objetos	959 objetos	Matriz de confusión 	0.995
Juego xox	671 objetos	287 objetos	Matriz de confusión 	0.794
Cáncer de Pecho	200 objetos	86 objetos	Matriz de confusión 	0.697

Fig. 6. Resultado de los árboles de decisión con el algoritmo C4.5.

Base de datos	Resultados				
	Red Neuronal Artificial				Árbol de Decisión con algoritmo C4.5
	Codificación Entera		Codificación Activa		
	Red 1	Red 2	Red 1	Red 2	
Evaluación de Carros	68.50%	74%	0%	0%	90%
Defectos en autos	55%	39%	0%	0%	98%
Juego de ajedrez	93%	83%	90%	92%	99%
Juego xox	16%	10%	52%	28%	79%
Cáncer de Pecho	42%	38%	0%	0%	69%

Fig. 7. Resultados de los métodos de clasificación en las 5 *bases de datos.*

En la base de datos "*Evaluación de carros"*, se utilizaron 1210 muestras en el entrenamiento y 518 objetos para la prueba, de los cuales clasifico correctamente 470 objetos, obteniendo una exactitud del 90%.

Para la base de datos "*Defectos en los autos"* se ocuparon 380 objetos para entrenamiento y 163 de prueba, en el cual clasifico 160 objetos dando una exactitud del 98% en este proceso.

En la base de datos "*Juego de ajedrez"* se tuvo una exactitud en la clasificación del 99%, con 2237 objetos para entrenamiento y 959 de prueba.

Como se observa todas las bases de datos obtuvieron una mejor exactitud en los árboles de decisión con algoritmo C4.5 para realizar la clasificación.

6. Conclusión

De los resultados obtenidos en esta investigación podemos concluir que, el algoritmo C4.5 obtuvo mejores resultados en las 5 bases de datos utilizadas, mostrando que para datos categóricos y bases de datos con muestras reducidas se obtiene buenos resultados, mientras que los resultados obtenidos con la red neuronal fueron mejores en bases de más de 1000 muestras con codificación entera.

Las redes neuronales que utilizaron codificación activa, en la mayoría de los datos no fue viable realizarla, debido a que las exactitudes obtenidas para realizar este proceso no fueron óptimas, en la figura 7 se presenta la comparación de los resultados obtenidos por cada método y por cada base de datos.

Finalmente podríamos decir que los árboles de decisión proporcionan mejores resultados en comparación de las redes neuronales artificiales cuando los conjuntos de datos están descritos en características categóricas. Esto debido a que para utilizar una red neuronal se requiere realizar los datos categóricos a numéricos.

Referencias

1. Kedar, P., Taher S., Pardawala, C., Pai, D.: A comparative study of categorical variable encoding techniques for neural network classifiers. International Journal of Computer Applications 175 (2017)
2. Sarkar, J.P., Saha, I., Chakraborty, S., Maulik, U.: Machine learning integrated credibilistic semi supervised clustering for categorical data. Applied Soft Computing Journal (2020)
3. Rajkamal, K., Dagli, C., Sun, Z.: Application of neural network in shop floor quality control in a make to order business. Procedia Computer Science (2016)
4. Küfner, T., Uhlemann T., Ziegler B.: Lean data in manufacturing systems: using artificial intelligence for decentralized data reduction and information extraction. Procedia CIRP (2018)
5. Patel, H.P.: Study and analysis of decision tree based classification algorithms. International Journal of Computer Sciences and Engineering (2018)
6. Choi, Sungsu: A decision tree approach for identifying defective products in the manufacturing process. International Journal of Contents, 13 (2017)
7. Hernandez-Orallo, J.: Introducción a la minería de datos. Pearson Prentice Hall (2004)
8. Haykin, S.: Neuronal networks and learning machines. Pearson Prentice Hall (1999)
9. Damodar, N.G.: Basic econometrics. The McGraw Hill (2004)
10. Brett, L.: Machine learning with R. Packt Publishing Limited (2013)
11. Von-Eye, A., Clifford, C.C.: Categorical variables in developmental research: Methods of analysis. Elsevier (1996)
12. Himani, S., Sunil, K.: A survey on decision tree algorithms of classification in data mining. International Journal of Science and Research (2013)
13. Fernández-Fernández, J.M., Flórez-López, R.: Las Redes Neuronales Artificiales. Netbiblo, (2008)

ISSN 1870-4069

Parametrización de terapia física para falanges de la mano humana

Juan Carlos Belen Luna, José Martin Flores Albino,
Héctor Rafael Orozco Aguirre, Carlos Omar González Morán

Centro Universitario UAEM Valle de México,
México

*metroc12@hotmail.com, coglez@gmail.com,
{jmfloresa,hrorozcoa}@uaemex.mx

Resumen. La mano humana es una de las partes más fascinantes y complicadas del cuerpo humano debido a su estructura, al contar con una precisión notable, con la que las personas pueden realizar un gran conjunto de tareas, desde usar una aguja hasta tocar un instrumento. En este artículo se presenta el análisis de la cinemática directa del movimiento de las falanges de cada dedo, utilizando el algoritmo de Denavit-Hartenberg. Además, la metodología para el diseño paramétrico del guante y su construcción, agregando la amplitud en grados respecto a los movimientos de extensión y flexión, también una proyección en 2D de los movimientos captados en tiempo real, para los valores de conversión de los sensores que contiene el guante, realizando una conexión y comunicación con la placa Arduino MEGA y el software Octave. Por último, se presentan los resultados obtenidos y las conclusiones determinando brindar un apoyo y análisis cuantitativo más que un análisis cualitativo durante una etapa de rehabilitación.

Palabras clave: Mano humana, rehabilitación, Octave, Arduino, sensores, guante.

Parameterization of Physical Therapy for Phalanges of the Human Hand

Abstract. The human hand is one of the most fascinating and complicated parts of the human body due to its structure, having remarkable precision, with which people can perform a great set of tasks, from using a needle to playing an instrument. This article presents the analysis of the direct kinematics of the movement of the phalanges of a finger, using the Denavit-Hartenberg algorithm. In addition, the methodology for the parametric design of the glove and its construction, adding the amplitude in degrees with respect to the extension and flexion movements, also a simulation of the movements captured by the conversion values of the sensors contained in the glove, performing a connection and communication with the Arduino MEGA board, visualizing the whole of everything on the Octave platform. Finally, the results obtained, and the

conclusions are presented, determining to provide support and quantitative analysis rather than a qualitative analysis during a rehabilitation phase.

Keywords: Human hand**,** rehabilitation, Octave, Arduino, sensors, glove.

1. Introducción

Todo ser humano es propenso a sufrir algún tipo de lesión en alguna extremidad del cuerpo, debido a que estas siempre están expuestas y en constante contacto con distintas superficies o esfuerzos en el entorno circundante. Hoy en día, el tratamiento de lesiones que involucran alguna extremidad del cuerpo humano es un enigma que ha generado un amplio espacio de estudio en la medicina, la tecnología y otras ciencias. Se pueden presentar problemas de movilidad por una enfermedad, una lesión traumatológica, o un derrame cerebral, con frecuencia los médicos recomiendan hacer fisioterapia a todos aquellos que presenten alguna de estas condiciones.

La mano humana, al ser una extremidad esencial, podría ser definida como una determinante para la independencia del ser humano, como lo decía Aristóteles "es el instrumento de los instrumentos" o como lo menciono Kapandji, "la mano es la extensión del cerebro". Por lo anterior, al ser una de las extremidades con mayor importancia, de acuerdo con [1] la mano constantemente adopta diversas formas y posiciones que le permiten al ser humano interactuar con el medio ambiente que lo rodea. Algunos ejemplos de posiciones efectuadas por la mano son: la concavidad palmar, que permite tomar y soltar objetos, también se realizan los movimientos de oposición, utilizados para actividades de destreza manual fina, por ejemplo, la manipulación de instrumentos de precisión.

En trabajos relacionados como en [1 y 2], se presenta la arquitectura de la mano, para entender la función prensil dependiendo directamente de la cadena cinemática de la mezcla de huesos y articulaciones que parten desde la muñeca (escafoides) hasta la punta de los dedos (falange distal), en agarres de fuerza y precisión.

En [3], se realiza el análisis del rango de movilidad y la función descriptiva del dedo índice, incluyendo el estudio anatómico y antropométrico concluyendo que la destreza y habilidad en los movimientos realizados también varía de acuerdo con el individuo.

Por otro lado, es posible realizar un caso en específico para el análisis, como se realiza en [4], en el cual se obtienen los rangos de movilidad de los dedos por separado y sus trayectorias a través de software.

Posteriormente, en este trabajo se propone un artefacto basado en los trabajos [5], [6, 7, 8], en conjunto con la composición anatómica de la mano humana y sus grados de libertad, es importante destacar que se asocia al concepto grados de libertad el movimiento de las falanges, debido a que el movimiento es semejante en los sistemas mecánicos para el movimiento de flexión y extensión en cada uno de los dedos, así como el estudio de la cinemática directa realizado por Denavit-Hartenberg, en conjunto con diversos componentes electrónicos como lo son la placa Arduino y el software Octave, aplicados para el modelado y la parametrización, resaltando que su desarrollo lo hace un modelo único y de muy bajo costo, permitiendo que sea accesible su implementación con fines terapéuticos, en pacientes que se encuentren en la etapa de rehabilitación, dando como resultado las ventajas de monitorear y evaluar la ejecución

de los ejercicios durante la rehabilitación, para la valoración de avances en terapias físicas con movimiento activo.

El resto de este trabajo se encuentra organizado como sigue: en la sección 2 se describe parte de la anatomía de la mano, en la tercera sección se incluyen el modelo de parametrización, su diseño y su modelado; posteriormente en la sección 4 se presenta un caso de estudio y finalmente las conclusiones y el trabajo futuro son dados en la sección 5.

2. Anatomía de la mano

La mano humana, al ser una extremidad esencial para el desarrollo de actividades realiza funciones tanto mecánicas, como sensitivas y de comunicación. Es el primer órgano para la manipulación física del entorno [9]. Está compuesta por 27 huesos descritos en la Tabla 1 (catorce falanges, cinco metacarpianos y ocho carpianos), conectados a través de numerosas articulaciones como se muestra en la Fig. 1.

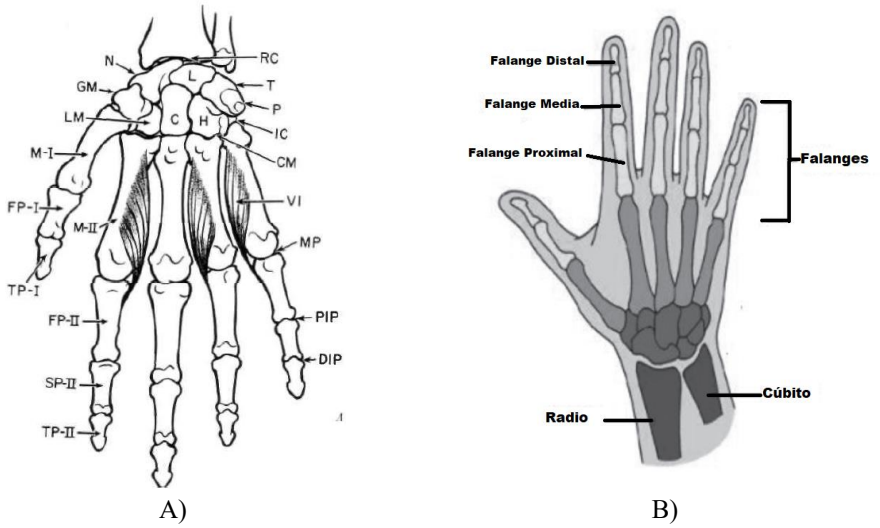


Fig. 1. A) Composición anatómica de la mano human [10]. B) Estructuración de las falanges de la mano humana [9].

Los dedos (meñique, anular, medio, índice) se caracterizan por tener 3 falanges (proximal, media y distal), a excepción del dedo pulgar que carece de la falange media como se observa en la Fig. 1.

Cabe mencionar que las falanges requieren un estudio antropométrico, para cualquier análisis y diseño de un modelo, uno de ellos es el realizado por John W. Garrett (ver Tabla 2), en el cual describe cada una de las longitudes de las falanges con respecto a la longitud total de la mano.

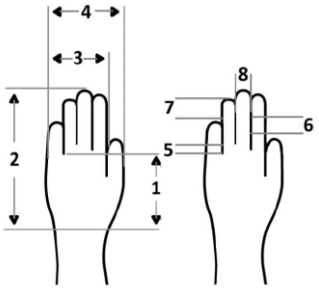


Fig. 2. Obtención de las medidas de la mano [4].

Tabla 1. Composición anatómica de la mano humana.

Huesos del carpo	Huesos metacarpos	Serie de falanges proximal	Serie de falanges media	Serie de falanges distal	Articulaciones
GM, Mayor multiangular (trapecio)	M-I	FP-I	-	TP-I	RC, Radiocarpiano
N, Escafoides	M-II	FP-II	SP-II	TP- II	IC, Intercarpiano
L, Semilunar	M-III	FP-III	SP-III	TP- III	CM, Carpometacarpiano
T, Piramidal (Triquetrum)	M-IV	FP-IV	SP-IV	TP-IV	MP, Metacarpofalángica
P, Pisiforme	M-V	FP-V	SP-V	TP-V	PIP, Proximal interfalángica
LM, Menor multiangular (trapezoide)					DIP, Distal interfalángica
C, Grande (Capitate)					
H, Ganchoso (Hamate)					

Tabla 2. Relación de longitud en milímetros que existe entre los dedos con respecto al porcentaje total de la mano.

Falange	Proximal	Media	Distal
Pulgar	17.1	-	12.1
Índice	21.8	14.1	8.6
Medio	24.5	15.8	9.8
Anular	22.2	15.3	9.7
Meñique	17.2	10.8	8.6

Tabla 3. Valores en milímetros medidos de las falanges de los dedos [4].

Falanges	Pulgar	Indice	Medio	Anular	Meñique
Proximal	37.63	29.53	34.77	30.13	22.78
Media	-	22.78	27.34	26.12	19.21
Distal	31.58	22.39	24.51	23.33	19.58

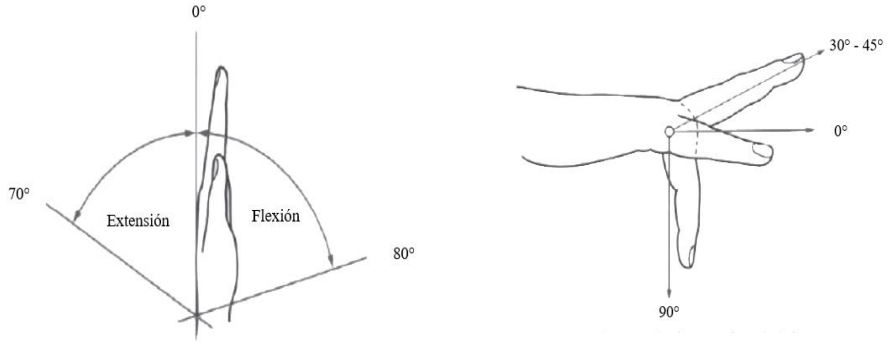


Fig. 3. A) Grados de libertad para la amplitud de movimiento en la muñeca [9]. B) Grados de libertad para la amplitud de las articulaciones metacarpofalángicas [9].

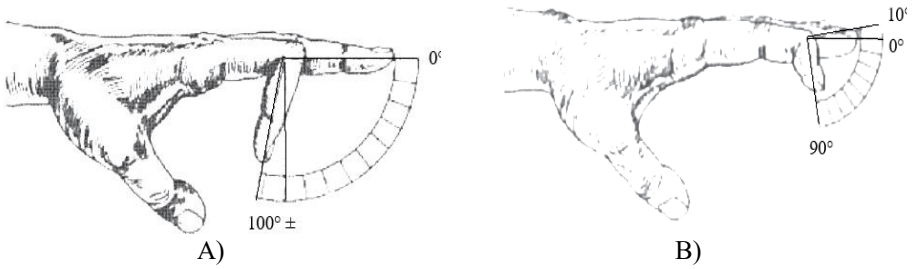


Fig. 4. A) Grados de libertad para la amplitud de movimientos en la falange media [9]. B) Grados de libertad para la amplitud de movimientos en la falange distal [9].

Por otra parte, es posible realizar un caso particular de estudio antropometrico, como se observa en la Fig. 2, en [4] muestran la manera correcta para obtener las medidas necesarias con el apoyo de un vernier, medidas que posteriormente tomaremos en cuenta para la parametrización de las falanges integradas en la Tabla 3.

Es importante mencionar que para que la mano humana pueda realizar algún movimiento, es necesario considerar los grados de libertad que se tiene permitidos para no sufrir algún tipo de lesión, los cuales son fundamentales para la parametrización, descartando la consideración de movimientos innecesarios.

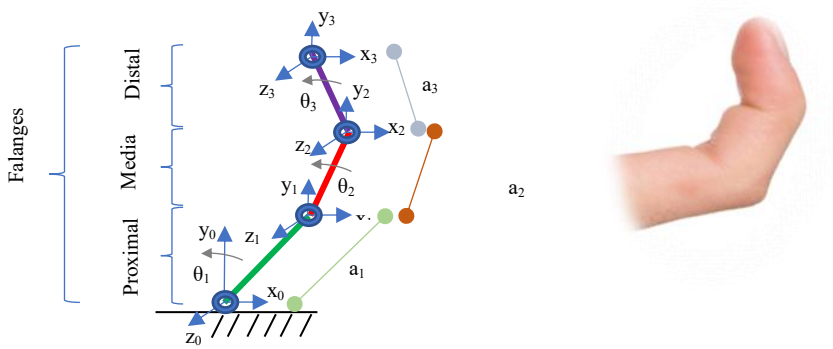
En la muñeca, para los movimientos de flexión y extensión se tienen 80° de amplitud de movimiento y 70° respectivamente [9] como se muestra en la Fig. 3. Por otro lado, para las articulaciones metacarpofalángicas la amplitud para la flexión es de 90° y para la extensión va de 30° a 45° como se muestra en la Fig. 3.

En complemento, en la Fig. 4 se muestra la flexión y extensión de la falange media, mostrando que la amplitud de movimiento interconecta la falange media con la distal, por último, se observa el movimiento de flexión y extensión que puede realizar la falange distal de igual forma en la Fig. 4.

3. Modelo de parametrización por cinemática directa de la mano humana

3.1 Cinemática del dedo

Para obtener la cinemática directa del movimiento de las falanges de un dedo, se hace uso del algoritmo Denavit-Hartenberg. El cual será aplicado a todos los dedos de la mano. En la Fig. 5, se expresan todos los parámetros para poder determinar las rotaciones y translaciones de las falanges del dedo índice en un eje coordenado.



A) B)

Fig. 5. A) Modelo cinemático del dedo con tres grados de libertad en un plano. B) Movimiento del dedo.

Tabla 4. Tabla de Denavit- Hartenberg y sus parámetros de las falanges del dedo índice.

Falanges	θ_i	d_i	a_i	α_i
Proximal	θ_1	0	a_1	0º
Media	θ_2	0	a_2	0º
Distal	θ_3	0	a_3	0º

Obteniendo la información de la Fig. 5, se observa que para d_i la distancia del origen z_i es el mismo punto, por tanto, no hay translación, así también como α_i no tiene rotación debido a que todas las z_i son paralelas, esta información se conforma en la Tabla 4 (Tabla de Denavit- Hartenberg) como se muestra a continuación, donde:

- $\boldsymbol{\theta_i}$ es la rotación sobre el eje z,
- $\boldsymbol{d_i}$ es la translación sobre el eje z,
- $\boldsymbol{a_i}$ es la translación sobre el eje x,
- $\boldsymbol{\alpha_i}$ es la rotación sobre el eje x.

Para describir el movimiento de cada falange se desarrollan las matrices de transformación homogénea de las partes: proximal, media y distal para cada dedo de la mano humana, que describen la posición-orientación relativa en cada una de las falanges.

Proximal: La ecuación (1), está dada por la matriz de rotación en combinación a la matriz de translación dando como resultado la matriz de transformación homogénea de Z_1 con respecto a Z_0, de la cual tenemos:

$$ {}^{0}H_1(\theta_1) = \begin{bmatrix} Cos(\theta_1) & -Sen(\theta_1) & 0 & a_1 \cdot Cos(\theta_1) \\ Sen(\theta_1) & Cos(\theta_1) & 0 & a_1 \cdot Sen(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1) $$

Media: La ecuación (2), está dada por la matriz de rotación en combinación a la matriz de translación dando como resultado la matriz de transformación homogénea de Z_2 con respecto a Z_1, para la cual obtenemos:

$$ {}^{1}H_2(\theta_2) = \begin{bmatrix} Cos(\theta_2) & -Sen(\theta_2) & 0 & a_2 \cdot Cos(\theta_2) \\ Sen(\theta_2) & Cos(\theta_2) & 0 & a_2 \cdot Sen(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2) $$

Distal: La ecuación (3), es la matriz de rotación en combinación a la matriz de translación dando como resultado la matriz de transformación homogénea de Z_3 con respecto a Z_2, para la cual se tiene:

$$ {}^{2}H_3(\theta_3) = \begin{bmatrix} Cos(\theta_3) & -Sen(\theta_3) & 0 & a_3 \cdot Cos(\theta_3) \\ Sen(\theta_3) & Cos(\theta_3) & 0 & a_3 \cdot Sen(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3) $$

Al tener las 3 matrices independientes, se realiza el cálculo de la matriz de transformación homogénea global para el dedo, para la cual obtenemos la ecuación (4):

$$ {}^{0}H_4(\theta_1, \theta_2, \theta_3) = \quad {}^{0}H_1(\theta_1) \cdot \quad {}^{1}H_2(\theta_2) \cdot \quad {}^{2}H_3(\theta_3). \quad (4) $$

El desarrollo para la ecuación (5) es la sustitución de las ecuaciones (1) y (2) de la ecuación (4), manteniendo la ecuación (3) en su forma normal sin sustitución, para facilitar su cálculo:

$$ {}^{0}H_4(\theta_1, \theta_2, \theta_3) = \begin{bmatrix} Cos(\theta_1) & -Sen(\theta_1) & 0 & a_1 \cdot Cos(\theta_1) \\ Sen(\theta_1) & Cos(\theta_1) & 0 & a_1 \cdot Sen(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cos(\theta_2) & -Sen(\theta_2) & 0 & a_2 \cdot Cos(\theta_2) \\ Sen(\theta_2) & Cos(\theta_2) & 0 & a_2 \cdot Sen(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \quad {}^{2}H_3(\theta_3). \quad (5) $$

Al realizar la multiplicación de las ecuaciones ${}^{0}H_1(\theta_1) \cdot \quad {}^{1}H_2(\theta_2)$ se obtiene la ecuación (6):

$$ \begin{bmatrix} Cos(\theta_1)Cos(\theta_2) - Sen(\theta_1)Sen(\theta_2) & -Cos(\theta_1)Sen(\theta_2) - Sen(\theta_1)Cos(\theta_2) & 0 & a_1Sen(\theta_1) + a_2Cos(\theta_1)Cos(\theta_2) - a_2Sen(\theta_1)Sen(\theta_2) \\ Sen(\theta_1)Cos(\theta_2) + Cos(\theta_1)Sen(\theta_2) & -Sen(\theta_1)Sen(\theta_2) + Cos(\theta_1)Cos(\theta_2) & 0 & a_1Sen(\theta_1) + a_2Sen(\theta_1)Cos(\theta_2) + a_2Cos(\theta_1)Sen(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6) $$

Posterior a la obtención de la ecuación (6), se le aplica a la misma una reducción de términos por identidades trigonométricas lo que nos da como resultado:

$$\begin{bmatrix} Cos(\theta_1+\theta_2) & -Sen(\theta_1+\theta_2) & 0 & a_1Cos(\theta_1)+a_2Cos(\theta_1+\theta_2) \\ Sen(\theta_1+\theta_2) & Cos(\theta_1+\theta_2) & 0 & a_1Sen(\theta_1)+a_2Sen(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

Al simplificar la ecuación (5), por el desarrollo en las ecuaciones (6) y (7), se realiza el siguiente cálculo, el cual consta de multiplicar el resultado de la ecuación (7) por la ecuación ${}^2H_3(\theta_3)$, como se muestra a continuación:

$$\begin{bmatrix} Cos(\theta_1+\theta_2) & -Sen(\theta_1+\theta_2) & 0 & a_1Cos(\theta_1)+a_2Cos(\theta_1+\theta_2) \\ Sen(\theta_1+\theta_2) & Cos(\theta_1+\theta_2) & 0 & a_1Sen(\theta_1)+a_2Sen(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} Cos(\theta_3) & -Sen(\theta_3) & 0 & a_3\cdot Cos(\theta_3) \\ Sen(\theta_3) & Cos(\theta_3) & 0 & a_3\cdot Sen(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

Finalmente, en la ecuación (9) se describe el resultado para la ecuación global del dedo en la matriz de transformación homogénea ${}^0H_4(\theta_1,\theta_2,\theta_3)$ del punto 0 al punto 3:

$$\begin{bmatrix} Cos(\theta_1+\theta_2+\theta_3) & -Sen(\theta_1+\theta_2+\theta_3) & 0 & a_1Cos(\theta_1)+a_2Cos(\theta_1+\theta_2)+a_3Cos(\theta_1+\theta_2+\theta_3) \\ Sen(\theta_1+\theta_2+\theta_3) & Cos(\theta_1+\theta_2+\theta_3) & 0 & a_1Sen(\theta_1)+a_2Sen(\theta_1+\theta_2)+a_3Sen(\theta_1+\theta_2+\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

Es importante mencionar que en complemento al algoritmo Denavit-Hartenberg, se puede calcular la posición del dedo desde la base de la falange proximal hasta la falange distal con la ecuación (10), siendo útil el estudio para el agarre, extensión y manipulabilidad en un espacio operacional:

$$|\vec{r}| = \sqrt{x^2+y^2+z^2}. \tag{10}$$

3.2 Diseño del guante

Se hizo un diseño buscando asemejar un exoesqueleto, acoplándole potenciómetros a la altura de las uniones de cada falange como se muestra en la Fig. 6, cabe denotar que como se mencionó en la sección 2, el dedo meñique, anular, medio e índice de la mano se constituyen por 3 falanges (proximal, media, distal), a excepción del dedo pulgar que carece de la media. Para la cual solo serán implementados 14 sensores para las señales de los movimientos, 3 sensores en los dedos meñique, anular, medio e índice y 2 sensores en el dedo pulgar.

Los potenciómetros permiten medir la variación de voltaje que existe en un determinado rango, y mediante programación en la plataforma IDE Arduino con algunos cálculos matemáticos se convierte el voltaje recibido a grados respecto a la variación de voltaje (ver Fig. 7), lo que permite medir y parametrizar la amplitud en grados de libertad para los movimientos de flexión y extensión en cada una de las falanges de la mano humana.

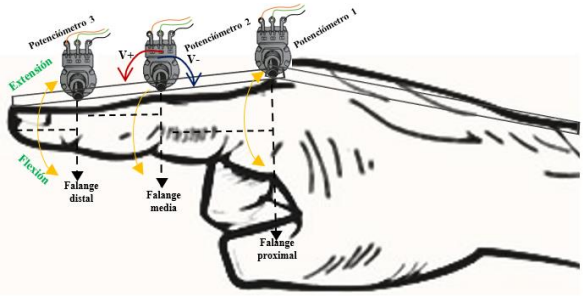


Fig. 6. Posicionamiento de potenciómetros en un guante exoesqueleto en cada unión de las falanges.

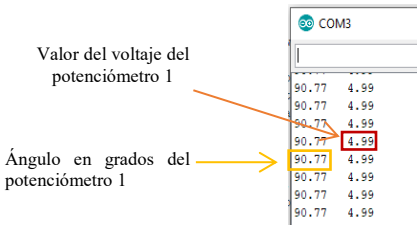


Fig. 7. Salida en el monitor serial de Arduino IDE, donde la primera columna proporciona el ángulo en grados y la segunda columna el valor del voltaje medido de cada falange.

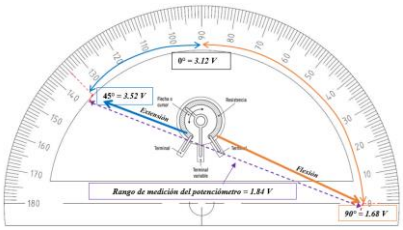


Fig. 8. Valores del potenciómetro logarítmico respecto a los movimientos de extensión y flexión del dedo medio.

A) B)

Fig. 9. A) Vista completa del prototipo del guante exoesqueleto. B) Montado de potenciómetros para cada configuración.

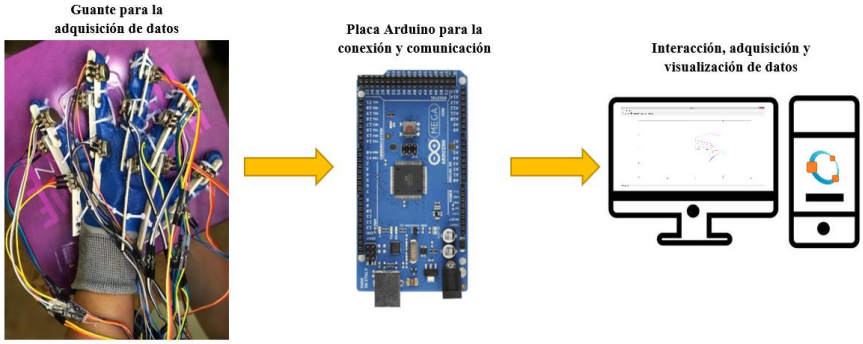


Fig. 10. Diagrama a bloques de la adquisición de datos de un guante exoesqueletico mediante el software Octave.

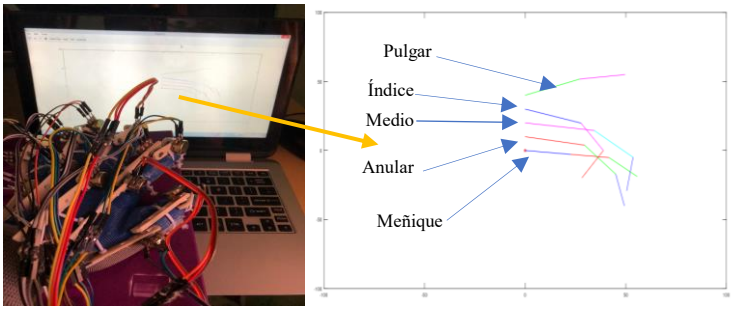


Fig. 11. Adquisición de datos de las falanges usando el software Octave.

Es importante mencionar que los potenciómetros implementados son del tipo logarítmicos, lo que implica determinar el rango de valores que serán tomados en cuenta para las mediciones de los grados de libertad en los movimientos de extensión y flexión, como se muestra en la Fig. 8. Además, cabe decir que en la evolución del prototipo se busca eliminar los saltos no medibles, implementando otro modelo de sensores para mejorar la precisión en la adquisición de los valores.

Se realizó un primer prototipo (ver Fig. 9), el cual contiene catorce potenciómetros de 5KΩ, montados en un material que permite simular cada uno de los huesos de la falange y simular sus movimientos, los sensores fueron colocados aproximadamente sobre la unión de cada falange, para poder determinar la variación de cambio en la amplitud de cada movimiento tanto para la flexión y extensión respectivamente.

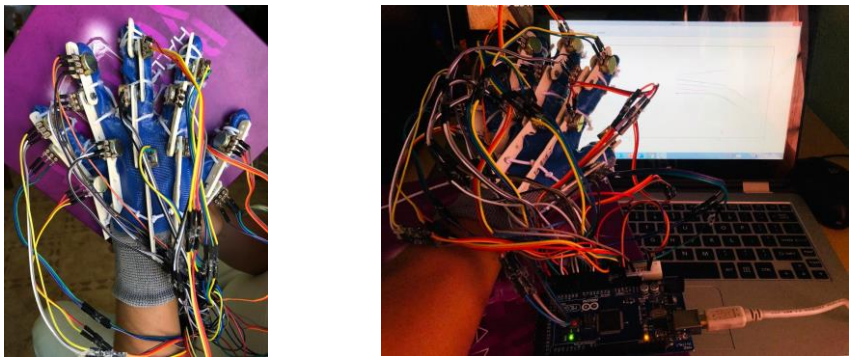
3.3 Simulación en Octave

Para la simulación, se requiere de una conexión entre el guante exoesqueleto, la placa Arduino y el software Octave como se muestra en la Fig. 10, los sensores que se montaron en cada falange envían los datos al Arduino, el cual permite comunicación con Octave.

Posterior a la conexión exitosa de los 3 elementos mencionados, en Octave se realiza un programa que permite la proyección en 2D de los movimientos en tiempo real para cada falange de la mano del usuario, permitiéndole visualizarlos en pantalla como se observa en la Fig.11.

4. Caso de estudio

Para este caso de estudio en particular, se trabajó en conjunto con un paciente que se encuentra en proceso de rehabilitación del miembro superior, montándole el guante exoesqueleto en la mano izquierda, como se puede visualizar en la Fig.12. Así mismo, se muestra el guante exoesqueleto en la mano izquierda del paciente, el cual contiene 14 sensores acoplados a las falanges de la mano humana, los cuales transmiten los datos que permiten el monitoreo de los movimientos de la flexión y extensión.



A) B)

Fig. 12. A) Mano izquierda del paciente utilizando el modelo de guante exoesqueleto con los sensores montados en las falanges de los dedos. A) Montado del guante exoesqueleto en la mano izquierda del paciente y conexión con la placa Arduino MEGA y el software Octave.

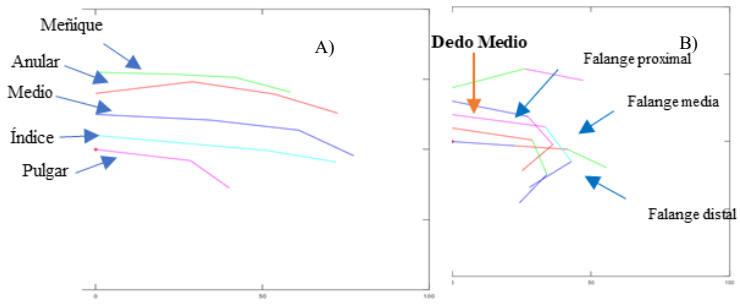
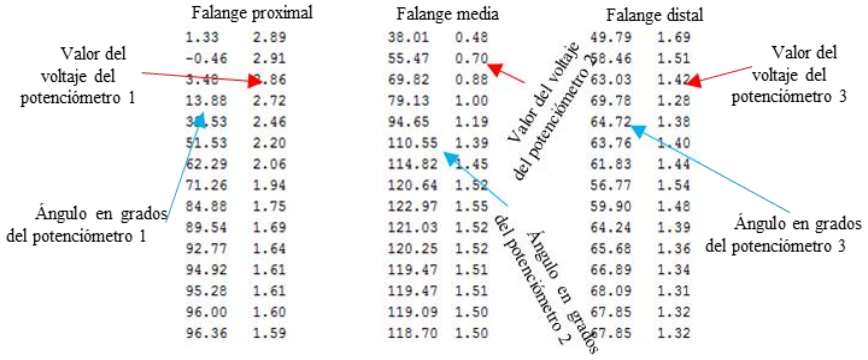
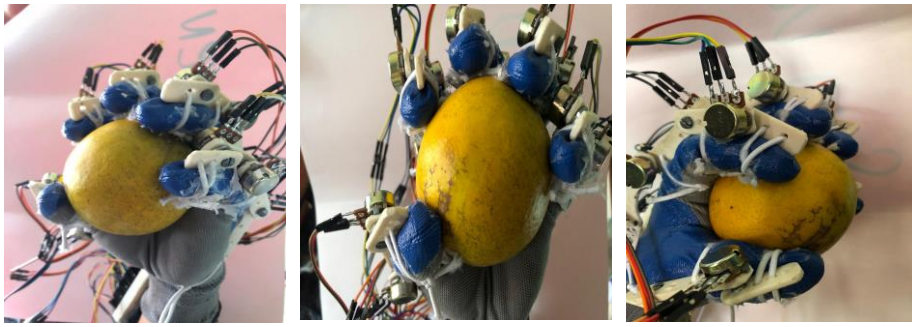


Fig. 13. A) Proyección 2D en tiempo real de las falanges en estado de reposo. B) Secuencia de movimientos de las falanges del dedo medio respecto a los valores obtenidos

Tabla 5. Datos adquiridos del ángulo en grados y del valor de voltaje para los sensores en el guante exoesqueleto, específicamente la falange proximal, media y distal ubicadas en el dedo medio.





A) B) C)

Fig. 14. A) Vista lateral derecha del movimiento de flexión y de agarre de un objeto esférico con el guante exoesqueleto. B) Vista frontal del agarre de un objeto esférico con el guante exoesqueleto. C)Vista lateral izquierda del agarre de un objeto esférico demostrando la libertad de movimiento con el guante exoesqueleto.

Los sensores del guante exoesqueleto se conectan a la tarjeta Arduino MEGA, la cual tiene comunicación mediante su puerto serial con el software Octave, permitiendo realizar una proyección 2D en tiempo real como se puede apreciar en la Fig.13, de la secuencia de movimientos para cada uno de los dedos y falanges de la mano del paciente.

Los valores para los sensores de las falanges (proximal, media y distal) se muestran en el puerto serial del IDE de Arduino (ver Tabla 5). Como se puede apreciar los valores cambian en tiempo real dependiendo directamente de la posición que tenga el sensor del guante exoesqueleto con respecto al movimiento de cada una de las falanges.

Como se sabe en fisioterapia, a los pacientes en etapa de rehabilitación se les proporcionan una seria de ejercicios que deben repetir específicamente por un periodo de tiempo, para los cuales, en algunos de estos se les invita a utilizar algún elemento u objeto extra para su manipulación.

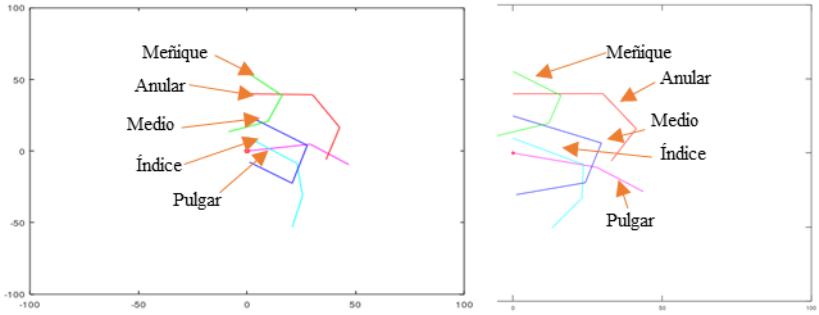


Fig.15. A), B). Proyección 2D en tiempo real de los movimientos de flexión al sostener un objeto esférico con el guante exoesqueleto en la mano del paciente.

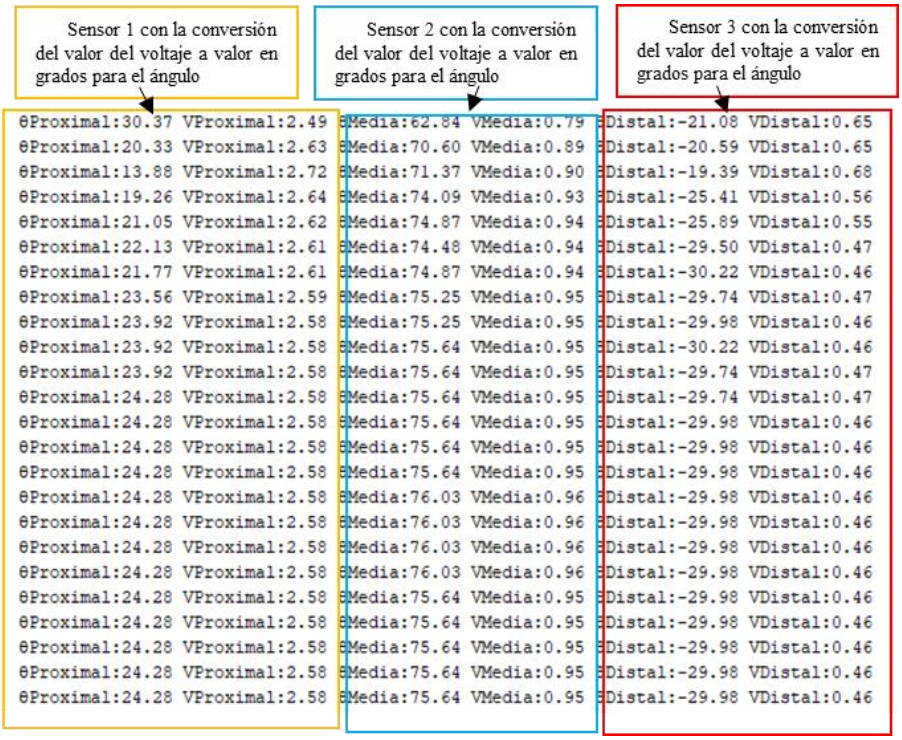


Fig. 16. Valores de los sensores ubicados en las falanges del dedo meñique al realizar el movimiento de flexión para el agarre de un objeto esférico.

En la Fig.14 se puede apreciar que el guante exoesqueleto permite la manipulación de objetos y, en este caso específico la manipulación de un objeto esférico para el movimiento de extensión en las falanges de la mano. En la Fig. 15, se muestran las proyecciones 2D en tiempo real obtenidas de las posiciones para los movimientos de flexión en el agarre de un objeto esférico, los cuales son valores de los 14 sensores

ubicados en el guante exoesqueleto proyectados en Octave de la comunicación serial con la tarjeta Arduino Mega.

Como se puede observar en la Fig.16, los valores de los sensores se actualizan en tiempo real con dependencia directa a los movimientos que realice cada una de las falanges de la mano del paciente, los valores son mostrados en el puerto serial del IDE de Arduino, identificados para la falange proximal, media y distal. En este caso, se tomó una captura para los valores del dedo meñique.

5. Conclusiones y trabajo a futuro

Al finalizar este proyecto, se define que el actual prototipo no es meramente para el ámbito académico, sino que se pretende mejorarlo y escalarlo en cada versión realizada a futuro, se pretende implementarlo como una nueva alternativa para evaluación en el porcentaje de avance para terapia física, ofreciendo un modelo de parametrización que permite al usuario visualizar en pantalla los movimientos realizados por las falanges de su mano mediante el software Octave en conjunto con elementos electrónicos.

Cabe mencionar que realizar un modelo que simule los movimientos hechos por la mano conlleva un alto grado de complejidad, ya que la exactitud y precisión de los movimientos es bastante difícil de replicar por la perfección natural con la que fue diseñada la extremidad. También se debe considerar que para el desarrollo de un guante exoesqueleto se deben tener en cuenta variables como la longitud de las falanges, ya que cada usuario tendrá características antropométricas distintas, por lo que se deben agregar elementos y/o mecanismos que permitan ajustar las posiciones y longitudes de colocación del guante exoesqueleto y sus sensores, de igual modo para simular y replicar los movimientos de flexión y extensión se deben implementar elementos al guante exoesqueleto que le permitan a la mano libertad natural de movimiento, haciendo posible adquirir las señales de los sensores de una manera eficiente para todas y cada una de las falanges de la mano.

Como trabajo a futuro, será perfeccionado el primer prototipo para mayor calidad y eficiencia en la adquisición de datos, para la cual será realizada una base de datos para que el fisioterapeuta pueda estudiar el movimiento de la mano y detectar posibles atrofias o mejoras en la calidad de vida de los pacientes. Estos datos se podrán consultar vía remota para que sin necesidad de que el paciente asista físicamente a su clínica, este sea diagnosticado a distancia. Otra aportación importante es que, a partir de la recolección de datos de muchos pacientes, se pueda generar algunas terapias a manera de video juego para estimular al paciente con estas tecnologías y así el paciente pueda culminar toda su rehabilitación.

Referencias

1. López, L. A.: Biomecánica y patrones funcionales de la mano. Morfolia, 4(1), pp. 14–24 (2012)
2. Abdallah, I., Bouteraa, Y., Rekik, C.: Design and development of 3d printed myoelectric robotic exoskeleton for hand rehabilitation. International Journal on Smart Sensing & Intelligent Systems, 10(2) (2017)

3. Velázquez-Sánchez, A., Merchán-Cruz, E., Hernández-Gómez, L., Urriolagoitia-Calderón, G.: Rango de movilidad y función descriptiva. Científica, 11(4), pp. 177–188 (2007)
4. Barrera-Garcia, A.D., Merchán-Cruz, E.A., Rodriguez-Cañizo, R.G., Hernandez-Lara, D., Hernandez-Valencia, G.: Obtención del rango de movilidad de los dedos índice, medio, anular y meñique. pp. 181–185 (2017)
5. Peña, P.E., Ticó, F.N., López, M.J.A., Romero, C.: Exoesqueleto para mano discapacitada con movimiento y sensibilidad, pero sin fuerza. UPCommons: Portal de acceso abierto al conocimiento de la UPC 76, pp. 1–6 (2012)
6. Iqbal, J., Tsagarakis, N., Fiorilla, A., Caldwell, D.: A portable rehabilitation device for the hand. In: Annual International Conference of the IEEE Engineering in Medicine and Biology, pp. 3694–3697 (2010)
7. Worsnopp, T., Peshkin, M., Colgate, J., Kamper, D.: An actuated finger exoskeleton for hand rehabilitation following stroke. In: IEEE 10th International Conference on Rehabilitation Robotics, pp. 896–901 (2007)
8. Yap, H., Khin, P., Koh, T., Sun, Y., Liang, X., Lim, J., Yeow, G.: A fully fabric-based bidirectional soft robotic glove for assistance and rehabilitation of hand impaired patients. In: IEEE Robotics and Automation Letters, 2(3), pp. 1383–1390 (2017)
9. Tenesaca, D. A.: Diseño y construcción de una mano robótica para la enseñanza del alfabeto dactilológico universal para personas sordomudas. Ingenius 6, pp. 69–86 (2011)
10. Schwarz, R.J., Taylor, C.L.: The anatomy and mechanics of the human hand. Artificial Limbs, 2(2), pp. 22–35 (1955)

ISSN 1870-4069

Extracción de respuestas a partir de ontologías utilizando patrones estructurales en SQWRL

Gabriela A. García-Robledo, Beatriz A. González-Beltrán,
José A. Reyes-Ortiz, Maricela Bravo

Universidad Autónoma Metropolitana,
Unidad Azcapotzalco,
México

{al2181800075,bgonzalez,jaro,mcbc}@azc.uam.mx

Resumen. La extracción de respuestas es una tarea que involucra un reto para la disciplina de Extracción de Información (*EI*). El uso de ontologías aminora la complejidad de esta tarea. Sin embargo, los desafíos vienen con el procesamiento de la petición del usuario (consulta) y su transformación en un lenguaje formal para la extracción de información de las ontologías. Por ello, en este artículo se propone un método para procesar una consulta semiestructurada, transformada automáticamente en el lenguaje *SQWRL* (Semantic Query-Enhanced Web Rule) y extraer respuestas de un sistema de ontologías del dominio académico utilizando patrones. El proceso completo consiste en, primero, extender la consulta semiestructurada con los elementos necesarios para la extracción de información del sistema de ontologías; después, se crean patrones estructurales para mapear entidades identificadas en la consulta a entidades existentes en el sistema de ontologías; finalmente, la respuesta es obtenida con la ejecución de los patrones generados en *SQWRL*. El objetivo principal de las consultas es contestar a preguntas de tipo ¿quién?, ¿dónde? y ¿cuándo? en un dominio académico. Una evaluación del método propuesto ha sido realizada con un conjunto de 258 preguntas formuladas y evaluadas por expertos en el dominio académico, obteniendo resultados alentadores con una precisión cercana al 97 %.

Palabras clave: Extracción de información, ontologías, reconocimiento de entidades, patrones estructurales en SQWRL.

Responses Extraction from Ontologies using Structural Patterns in SQWRL

Abstract. Question-answering is a challenge for the Information Extraction (*EI*) task. The use of ontologies reduces the complexity of this task. However, the challenges are presented in the processing of the user request through a query and how it is converted into a formal language for information extraction from ontologies. This paper proposes a method

that starts with a semi-structured query in order to translate it into the *SQWRL* language (Semantic Query-Enhanced Web Rule) and extract a suitable answer from an ontology system of the academic domain using patterns. The whole process consists of extending the semi-structured query with the necessary elements to extract relevant information from the ontology system; structural patterns are created to match entities identified in the query with existing entities in the ontology system; the answer is obtained with the execution of patterns generated in *SQWRL*. The queries are able to answer questions about who? where? and when? in an academic domain. An evaluation process has been carried out with a set of 258 questions, which were expressed and evaluated by experts in the academic domain, obtaining promising results with precision close to 97%.

Keywords: Information extraction, ontologies, entity recognition, structural patterns in SQWRL.

1. Introducción

En un ambiente académico se realizan diariamente actividades de docencia, investigación y difusión, tales como: asesorías, congresos, seminarios, clases, talleres, cursos, entre otros. También existen instalaciones principales, tales como: laboratorios de cómputo, cafetería, librería, salones de clase, auditorios, entre otros. Finalmente, diversos participantes realizan actividades dentro de ellas como: administrativos, profesores y alumnos.

Una de las formas de almacenamiento y representación de la información generada de estas actividades o eventos, son haciendo uso de ontologías, las cuales permiten estructurar dicha información de tal manera que se facilite su consulta. Sin embargo, la consulta de información sobre eventos y sus aspectos relacionados (personas, tiempo y lugar) involucra una petición de un usuario. Debido a la gran cantidad de información generada en el dominio académico, este proceso de consulta a partir de una petición en lenguaje natural o en un formato estructurado, requiere de un procesamiento computacional y de su transformación hacia lenguajes formales.

Los lenguajes formales de consulta ayudan a obtener información dentro de una ontología, dentro de la literatura los principales son: a) *SPARQL* (Protocol and RDF Query Language), que es un lenguaje de consulta "orientado a objetos" porque solo consulta la información contenida en los modelos[1]; b) *SWRL* (Semantic Web Rule Language) es un formalismo para integrar reglas con ontologías en la Web Semántica y esta basado en una combinación de dos lenguajes *OWL* (Web Ontology Language) y *RuleML* (Rule Markup Language)[3]. *SWRL* introduce las reglas al adaptar la semántica existente por un lenguaje de reglas directamente en la capa de ontología[2]. La interacción entre ontologías y el lenguaje de consulta se integran en átomos antecedentes evaluados por una conjunción y los átomos consecuentes, ambos pueden ser vacíos[3].

El proceso de transformar una consulta estructurada que represente la petición de un usuario es un reto hoy en día. Por ello, el objetivo principal de este trabajo es partir de una petición de un usuario expresada en una consulta semiestructurada obtenida en [15] y extraer una respuesta de un sistema de ontologías del dominio académico. Este proceso involucra el enriquecimiento de una consulta semiestructurada, mapeo de entidades a elementos del sistema de ontologías, la identificación de patrones estructurales (sucesión de elementos estructurados identificados con frecuencia dentro de distintas preguntas) y la ejecución de la consulta en *SQWRL* para obtener una respuesta específica sobre eventos, espacios físicos y actores involucrados (personas) que pertenecen al dominio académico. La elección de este lenguaje de consulta se baso en la afinidad entre la estructura resultante de [15] y la sintaxis de átomos antecedentes y consecuentes. El sistema de ontologías utilizado modela, entre otros conceptos, la noción de personas, tiempo y lugares; con ello es posible contestar preguntas del tipo ¿quién?, ¿dónde? y ¿cuándo?.

El resto del artículo se organiza de la siguiente manera. En la Sección 2 se presentan los avances más importantes y recientes respecto a las área de investigación de este trabajo, tales como: a) el uso de ontologías para representar información de eventos; b) el uso de lenguajes formales para extraer información de ontologías; c) métodos de transformación de consultas en lenguaje natural (no estructurada), semi-estructurada o estructurada. Por su parte, en la Sección 3 se expone el método propuesto para la transformación de la consulta semiestructurada en una consulta en *SQWRL* con la finalidad de extraer información del sistema de ontologías y obtener una respuesta. La evaluación del método propuesto es presentada en la Sección 4 mediante una experimentación con 258 preguntas y mostrando resultados alentadores. Finalmente, en la Sección 5 se presentan las conclusiones de este artículo y el trabajo a futuro.

2. Trabajos relacionados

En esta sección se presenta una revisión de los avances dentro del área de investigación de este trabajo. Existen diversos trabajos que se fundamentan en el uso de ontologías para las consultas de información en lenguaje natural, pero pocos lo han aplicado usando un método de extracción en consultas *SQWRL*.

López, V., et al.[4] en 2005, reportan a Aqualog, donde proponen una arquitectura en cascada, la consulta es traducida a un conjunto de tripletas compatibles con la ontología. El modelo de tripletas que utilizan es de la forma "sujeto, predicado, verbo". Requiere de los recursos de GATE (General Architecture for Text Engineering) para tareas de procesamiento de lenguaje natural. El sistema identifica términos, relaciones, indicadores de preguntas como lo son "qué/quién/cuándo", patrones o tipos de preguntas. Por otro lado, en el mismo año Muhammad, A., et al.[5] desarrollan un método de extracción de información biológica basada en una ontología y sistema de respuesta a consultas con cuatro componentes: una etiqueta basada en una ontología, un etiquetador de nuevos

textos, una base de conocimiento potenciador que mejora la ontología e incorpora nuevos conocimientos y un procesador de consultas de usuarios.

La propuesta de Bernstein, A., et al.[6] en el 2006 describe a Ginseng, una interfaz de lenguaje natural que utiliza reglas gramaticales estáticas que proporcionan las estructuras y frases básicas del idioma. Ginseng utiliza el vocabulario definido por las ontologías, no necesita interpretar las consultas, maneja una gramática multinivel, un analizador incremental y una capa de consulta a la ontología a través de SPARQL.

En el 2007, Tianyong, H., et.al[7] proponen un tipo de patrón semántico, que se puede usar para publicar preguntas y respuestas en sistemas QA interactivos para el usuario. El procedimiento incluye un análisis de estructura de preguntas, coincidencia, generación y clasificación de patrones para la extracción de respuesta. Implementan una interfaz de usuario para hacer uso del patrón semántico en un sistema de control de calidad que permite a los usuarios publicar y responder preguntas.

Tablan, V., et al.[8] en el 2008 presentan QuestIO System, que es una interfaz de lenguaje natural sin un vocabulario predefinido, realiza como primera parte un análisis lingüístico por medio de un tokenizador y etiquetado, después ejecuta el diccionario ontológico que crea anotaciones acerca de la identificación de las partes importantes de la ontología, posteriormente se inicia una transformación de la consulta para generar una formal.

Damljanovic, D., et al.[9] en el 2010 crean FREYA, una interfaz de lenguaje natural que utiliza búsquedas en *SPARQL* en ontologías, ejecuta preguntas en lenguaje natural y genera una respuesta en forma de un grafo. La interfaz también utiliza el conocimiento de la ontología para la identificación de términos, con los que genera tripletas para realizar la búsqueda. Encuentra el tipo de respuesta donde se realiza un análisis sintáctico y localiza la frase relevante que define a la pregunta y lo que se busca.

También, existe un método en el 2015 desarrollado por Eka, A., et al.[10] que se enfoca en analizar preguntas de tipo ¿Por qué? La propuesta adapta el modelo simple de bolsa de palabras, utiliza entidades semánticas para representar una consulta que al ser ejecutada se enriquecen con ayuda de *SQWRL* sobre la ontología de dominio.

Existen desarrollos en la geomática y geografía, Bereta, K., et al.[11] presentan un sistema capaz de responder consultas *GeoSPARQL* (A Geographic Query Language for RDF) sobre base de datos relacionales, traducen *GeoSPARQL* a *SQL* utilizando ontologías y asignaciones.

Una implementación relacionada también con *SPARQL* se presenta en Solís, A., et al.[12]. Una interfaz de lenguaje natural para deducción de información almacenada en ontologías. Utiliza solo *SPARQL* como lenguaje de consulta en ontologías. El módulo procesador obtiene etiquetas gramaticales y lemas del texto utilizando Freeling. La información obtenida es enviada a un módulo de conocimiento capaz de generar consultas y ejecutarlas.

En el 2019 Setyo, F., et al.[13] desarrollan un clasificador de instancias para corpus pequeños aplicados a un sistema de respuesta de preguntas semánticas.

Consta de un pre procesamiento, análisis morfológico y semántico, extracción de características, clasificación de instancias con un algoritmo de redes de función de base radial y un módulo de transformación.

Finalmente, en el sector de la salud, Mourad, S. y Said, O.E.A. [14] en este año desarrollan SemiBioNLQA, que consiste en clasificar preguntas, recuperar documentos y extraer una respuesta. El sistema trata cuatro tipos de preguntas y se basa en: patrones artesanales sintácticos de México, un algoritmo de aprendizaje automático para la clasificación de preguntas, un motor de búsqueda y similitud para la recuperación de documentos, análisis de sentimientos y una métrica de frecuencia para la extracción de la respuesta.

Con la revisión de los trabajos relacionados a las disciplinas involucradas en este artículo, se puede observar la importancia de realizar una adecuada transformación de una consulta no estructurada (lenguaje natural) o semiestructurada hacia una consulta estructurada, apoyándose de un lenguaje formal con la finalidad de extraer información de algún medio por ejemplo las ontologías. Además, se puede observar que la mayoría de los trabajos se enfocan en el inglés, esto conlleva a una necesidad latente de contar, con un método de extracción de información partiendo de una consulta en español, lenguaje en el cual existe una carencia de herramientas computacionales. Por ello, en la siguiente sección se presenta el método propuesto para transformar una consulta semiestructurada en consulta *SQWRL* y obtener una respuesta de un sistema de ontologías del dominio académico.

3. Método propuesto

En esta sección se presenta el método propuesto para transformar una consulta semiestructurada que expresa la petición del usuario en una consulta en *SQWRL*. Este proceso involucra un un mapeo de entidades ontológicas, una identificación de patrones estructurales y una ejecución de una consulta *SQWRL* para obtener una respuesta de un sistema de ontologías. El método propuesto se observa en la Figura 1.

3.1. Consulta semiestructurada creada en [15]

El método propuesto en este trabajo tiene como punto de partida una consulta semiestructurada, la cual es creada en [15] mediante identificación de entidades y componentes de una consulta en lenguaje natural del dominio académico con una precisión del 96 %. El método propuesto en [15] es realizado a partir de lexicones de eventos, espacios físicos y personas que pertenecen a un ambiente académico, disponibles en: https://github.com/UamAISII/AppVoz.git. Ellos analizan un sistema de ontologías llamado *Ambiente Inteligente* desarrollado en [16] con las ontologías Persona, Evento, Red de Sensores, Espacio Físico y Tiempo. En [15] se crea una estructura de recuperación de información y una *tupla* ontológica con cierta estructura, pero que sigue conservando elementos no estructurados como texto libre.

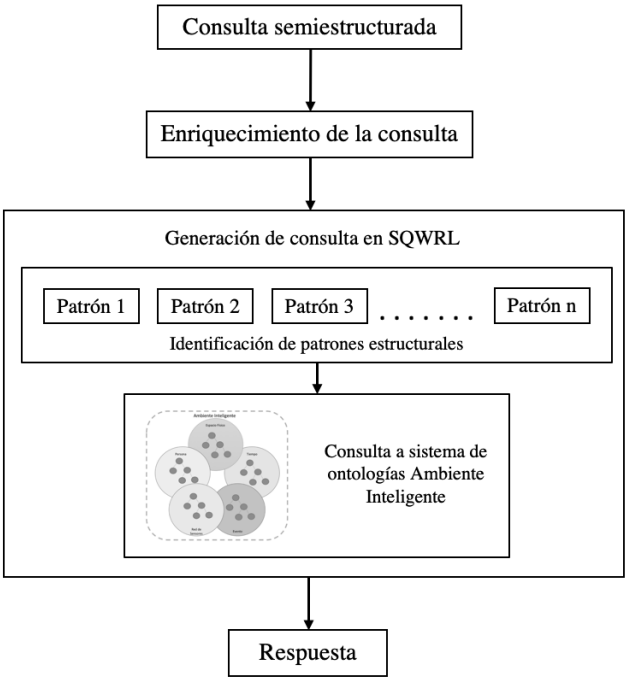


Fig. 1. Método de solución propuesto.

Esta consulta semiestructurada es mejorada en este trabajo para considerar elementos necesarios para responder a la petición del usuario.

La consulta semiestructurada obtenida consta de diversos elementos, tales como el tipo de pregunta expresada y el conjunto de entidades relevantes que contiene la petición del usuario. Las entidades relevantes son aquellas que contienen la información necesaria para responder a la consulta del usuario. Este conjunto de información es representada en una *5-tupla* elementos con la estructura mostrada en la representación mostrada (1):

$$[TP, S, TS, CS, I^*]. \tag{1}$$

Las preguntas en lenguaje natural (texto no estructurado) fueron generadas por miembros del cuerpo estudiantil de la UAM Azcapotzalco (aproximadamente 40 personas formadas por ayudantes del departamento de sistemas y estudiantes de ingeniería en computación e ingeniería industrial), quienes apoyaron mediante la expresión de diversas formas de preguntar a una interfaz de consulta, por ejemplo, la dinámica es: “Preguntar por la ubicación del auditorio W001”, un ejemplo de petición mostrado a la comunidad es: “¿Dónde esta el auditorio W001?”.

De esta manera, se generó un corpus de 300 preguntas, las cuales arrojan la misma estructura general de la representación mostrada (1), donde:

- TP es el tipo de pregunta;
- S el sujeto del que se habla en la pregunta;
- TS el tipo de sujeto, por ejemplo, el nombre de una persona, la matrícula, el nombre del lugar, entre otros;
- CS se refiere a la clase del sujeto como, profesor, edificio, auditorio, estudiante, entre otros;
- I^* es un dato que tiene una ocurrencia de 0 o más, es decir, la información extra que puede existir o no dentro de la pregunta.

Si la consulta es: "¿Cuál es la matrícula de Pedro García?", la estructura generada siguiendo con las caracaterísticas de la representación mostrada (1) se muestra en la Tabla 1.

Tabla 1. Ejemplo de representación 1.

[Persona, pedro garcía, nombre, alumno, matrícula]

En la Tabla 1, *Persona* es TP, "*pedro garcía*" es S, *nombre* equivale a TS, *alumno* es CS y por último nuestra I^* es *matrícula*.

Debido a la gran variedad de preguntas generadas en el idioma español, se ha observado la necesidad de extender la representación mostrada (1) con la finalidad de obtener la mayor cantidad de entidades involucradas en la pregunta para que ayuden a encontrar una respuesta pertinente. Este proceso de enriquecimiento de la consulta semiestructurada se muestra a continuación.

3.2. Enriquecimiento de la consulta estructurada mediante mapeo ontológico

La representación mostrada (1), es mejorada en este trabajo mediante un mapeo de elementos ontológicos (relaciones, nombres de clase e instancias) obteniendo la representación mostrada (2). Esta corresponde a una *7-tupla* de elementos que cumple con todos los datos necesarios para realizar la transformación y ejecución de una consulta en $SQWRL$ sobre el sistema de ontologías del dominio académico:

$$[O, S, P, TP, C, P_2, TP_2], \tag{2}$$

donde:

- O es la ontología u ontologías a las que debe consultarse;
- S es el sujeto que contiene la pregunta;

- P es la propiedad del sujeto, por ejemplo, si es el nombre propio de una persona deberá decir *hasName* o si habla del nombre de un auditorio dirá *hasNamePhysicalSpace*, estas propiedades pertenecen al sistema de ontologías;
- TP se refiere al tipo de propiedad *DataProperty* u *ObjectProperty* según sea el caso;
- C es la clase del sujeto dentro de la ontología que consulta y que contiene la información para responder, por ejemplo, *Professor*, *Building*;
- P_2 es la propiedad que se desea saber, por ejemplo, si queremos saber la matrícula de un alumno este dato contendrá *hasStudentID*;
- TP_2 se refiere al tipo de P_2 y puede ser *DataProperty* u *ObjectProperty*.

Continuando con el ejemplo de consulta mostrado en la sección anterior: "¿Cuál es la matrícula de Pedro García?", la estructura generada siguiendo con las caracaterísticas de la representación mostrada (2) se muestra en la Tabla 2.

Tabla 2. Ejemplo de representación 2.

[person, pedro garcía, hasName, DataProperty, RegularStudent, hasStudentID, DataProperty]

En la Tabla 2, *person* es O, "*pedro garcía*" es S, *hasName* equivale a P, *DataProperty* es TP, *RegularStudent* es C, en el caso de *hasStudentID* que pertenece a P_2 y por último TP_2 es *DataProperty*.

Las clases y relaciones de tipo de dato (propiedades) del sistema de ontologías que pueden ocupar un lugar dentro de la representación mostrada (2) se muestran en la Tabla 3 y Tabla 4 respectivamente.

Las ontologías contienen, también, relaciones de tipo objeto o relaciones semánticas. De esta manera los elementos que conforman la *7-tupla* en la representación mostrada (2) puede contener una relación de este tipo que involucre a más de una ontología. La Tabla 5 muestra estas relaciones de objeto del sistema de ontologías del dominio académico.

3.3. Identificación e implementación de patrones estructurales

La representación mostrada en la sección anterior es transformada en consultas en *SQWRL* como lenguaje formal para la ejecución de la consulta sobre el conocimiento existente en el sistema de ontologías. Mediante un análisis de 300 peticiones con la información semiestructurada en la *7-tupla*, se identificaron tres patrones estructurales con sintaxis *SQWRL*. Los cuales se presentan a continuación.

El Patrón 1 que se centra en las preguntas relacionadas a las categorías de espacio físico y persona, tiene la siguiente sintaxis:

$$\text{Patrón 1: } ON : CL(?var_1) \wedge ON : DP(?var_1, \text{“}VS\text{”}) \wedge ON : DPS(?var_1, ?res) \rightarrow sqwrl : select(?res),$$

Tabla 3. Clases del sistema de ontologías del dominio académico.

Ontología	Nombre de clase
Espacio Físico	ClassRoom
	Cubicle
	CommonArea
	Building
	ComputerLab
	AdministrativeOffice
	Auditorium
	GreenArea
	PublicArea
Tiempo	Instant
Persona	Profesor
	AssistantStudent
	RegularStudent
Evento	DiplomaCourse
	PostgraduateCourse
	UndergraduateCourse
	Colloquium
	Congress
	Demo
	DiscussionPanel
	Presentation
	Seminar
	Workshop

donde: ON es el nombre de la ontología de referencia; CL es el nombre de la clase involucrada; DP es el nombre de la relación de tipo de dato (propiedad); VS es el nombre del sujeto de la relación; DPS es la relación de tipo de dato (propiedad) de la respuesta. La relación entre las variables del patrón estructural 1 y la representación mostrada (2) pueden observarse en la Tabla 6.

El Patrón 2 es el resultado de un análisis de las variantes de preguntas relacionadas a los eventos. Para este tipo de pregunta en particular, se puede contestar sobre su descripción o el tiempo en que es realizado dicho evento. Esto se debe a la información representada en el sistema de ontologías, el cual contiene una ontología *Evento* y una ontología *Tiempo* de propósito general que ayuda a saber los datos que se refieren a cuándo ocurren los eventos académicos. Por esta razón si solo se pregunta por la descripción del evento es utilizado el Patrón 1. Sin embargo, si la pregunta se refiere al tiempo del evento, por ejemplo, "¿Cuándo es el Seminario de Física?", se necesitan más especificaciones y como resultado de ello, se genera el Patrón estructural 2 que se muestra a continuación:

$$\text{Patrón 2: } ON_1 : Event(?var_1) \wedge ON_2 : I(?var_2) \wedge On_1 : DP(?var_1, \text{“}VS\text{”}) \wedge ON_1 : DPS(?var_1, ?var_2) \wedge ON_2 : PT_1(?var_2, ?res_1) \wedge ON_2 : PT_2(?var_2, ?res_2) \rightarrow sqwrl : select(?res_1) \wedge sqwrl : select(?res_2),$$

Tabla 4. Relaciones de tipo propiedad del sistema de ontologías del dominio académico.

Ontología	Nombre de propiedad
Espacio Físico	hasLevel
	hasNamePhysicalSpace
	hasDay
Tiempo	hasHour
	hasMinute
	hasMonth
Persona	hasAcademicTitle
	hasCategory
	hasEconomicNumber
	hasEmail
	hasName
	hasStudentID
Evento	hasEventName
	hasDescription

Tabla 5. Relaciones semánticas (tipo de objeto) del sistema de ontologías.

Propiedad de objeto	Dominio	Rango
isAssignedTo	Persona	Espacio Físico
hasPersonAssigned	Espacio Físico	Persona
isBesideOf	Espacio Físico	Espacio Físico
isLocatedInto	Espacio Físico	Espacio Físico
containsPhysicalSpace	Espacio Físico	Espacio Físico
participatesIn	Persona	Evento
hasPersonInvolved	Evento	Persona
hasTemporalEntity	Evento	Tiempo
happensIn	Evento	Espacio Físico
eventIsPartOf	Evento	Evento
isTemporalEntityOf	Tiempo	Evento
hasEventProgrammed	Espacio Físico	Evento

donde ON_1 es el nombre de la ontología detectada en la tupla ontológica; ON_2 se refiere al nombre de la ontología *Tiempo* (*time*); I es la clase que contiene la información del tiempo del evento (clase $Instant$ de ontología *Tiempo*); PT_1 es la relación de tipo de dato (propiedad) del tiempo que se refiere al día del evento (*hasDay*); PT_2 es la relación de tipo de dato (propiedad) del tiempo que se refiere al mes del evento (*hasMonth*); DP es el nombre de la relación de tipo de dato (propiedad); VS es el nombre del sujeto de la relación; DPS es la relación de tipo de dato (propiedad) de la respuesta. La relación entre las variables del patrón estructural 2 y la *7-tupla* ontológica de la representación mostrada (2) pueden observarse en la Tabla 7.

En el caso de $?var_1$ y $?var_2$ son variables dentro de la consulta. $?res_1$ y $?res_2$ son variables que contienen la respuesta final, que al preguntar por el tiempo, se debe arrojar una fecha conformada por día y mes.

Tabla 6. Relación de variables entre patrón estructural 1 y *7-tupla* ontológica.

Variable de patrón estructural 1	Variable en mapeo ontológico
ON	O
CL	C
DP	P
VS	S
DPS	P_2

Tabla 7. Relación de variables entre patrón estructural 2 y *7-tupla* ontológica.

Variable de patrón estructural 2	Variable en mapeo ontológico
ON	O
DP	P
VS	S
DPS	P_2

El Patrón 3 es el resultado del análisis detallado de las preguntas de tipo espacio físico-evento, espacio físico-persona y evento-persona, del cual se obtuvo el patrón estructural 3 mostrado a continuación:

$$\text{Patrón 3: } PR_1 : ON_1(?var_1) \wedge PR_1 : DP(?var_1, \text{“}VS\text{”}) \wedge PRG : DPS(?var_1, ?var_2) \wedge PR_2 : ON_2(?var_2) \wedge PR_2 : DPS(?var_2, ?res) \rightarrow sqwrl : select(?res),$$

donde PR_1 es el prefijo de ontología 1 y PR_2 es el prefijo de ontología 2. El manejo de prefijos para este tipo de preguntas es relevante ya que se manejan dos ontologías y con ellos se evitan las ambigüedades en los nombres de clases y relaciones, además, en *SQWRL* ayuda a saber la ubicación de cada valor a preguntar dentro del sistema de ontologías; ON_1 es el nombre de la ontología 1 y ON_2 el nombre de ontología 2; DP es el nombre de la relación de tipo de dato (propiedad); VS es el nombre del sujeto de la relación; DPS es la relación de tipo de dato (propiedad) de la respuesta. La relación entre las variables del patrón estructural 3 y la *7-tupla* ontológica de la representación mostrada (2) pueden observarse en la Tabla 8.

Tabla 8. Relación de variables entre patrón estructural 3 y *7-tupla* ontológica.

Variable de patrón estructural 3	Variable en mapeo ontológico
PR_1	O
DP	P
PR_2	O
VS	S
ON_1	O
DPS	P_2
ON_2	O

PRG es el prefijo de sistema de ontologías (*IntelligentEnviroment*) es una parte de la consulta que habla de la unión existente entre las dos ontologías dentro del sistema. En el caso de $?var_1$ y $?var_2$ son variables dentro de la consulta, y $?res$ es la variable que contiene la respuesta final.

Un ejemplo de una consulta utilizando uno de estos patrones se muestra en la Figura 2. Donde se observa que la pregunta de entrada es "¿Cuál es la matrícula de Pedro García?", se lleva a cabo la extracción como en la representación mostrada (1) [15], después se realiza la *7-tupla* como en la representación mostrada (2). En este caso se identificó el tipo de pregunta persona por lo que en la parte de consulta se aprecia que se utilizó el patrón estructural 1. Donde *person* es ON, $RegularStudent$ es CL, $hasName$ es DP, "$PEDRO\ GARCIA$" es el valor de "VS" y $hasStudentID$ es DPS. La variable $?res$, se encuentra en la respuesta, es decir, el valor extraído del sistema de ontologías en este caso 2100000000.

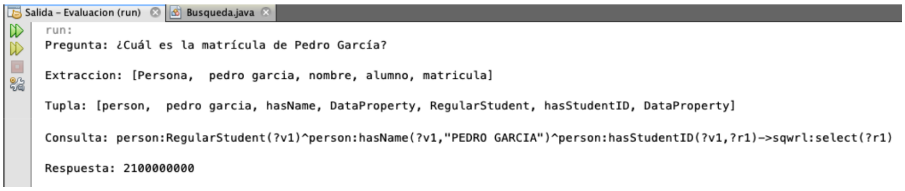


Fig. 2. Ejemplo de consulta utilizando patrón 1.

4. Experimentación y resultados

El método propuesto se evalúa a partir de los individuos existentes en el sistema de ontologías del dominio académico obtenido de [16]. Este modelo consta de 855 individuos representados con la información adecuada para responder cada tipo de pregunta, 91 clases, 10736 axiomas, 36 relaciones de objeto y 51 relaciones de tipo de dato (propiedades). Para la experimentación se realizaron 258 preguntas que fueron evaluadas por expertos del dominio académico (profesores investigadores, estudiantes y ayudantes). El desarrollo del este método esta disponible en: https://github.com/UamAISII/AppVoz.git. La distribución de estas preguntas en cada tipo se observa en la Tabla 9.

Las 258 preguntas son generadas en el idioma español y obtenidas de la comunidad universitaria de la UAM Azcapotzalco, aproximadamente 70 personas entre las cuales se encuentran profesores investigadores del departamento de sistemas, alumnos de ingeniería en computación e ingeniería industrial y ayudantes de profesores, todos con una diversidad de maneras de preguntar. El corpus de cada tipo de pregunta esta disponible en: https://github.com/UamAISII/AppVoz.git, este es formado y evaluado de manera manual por un miembro de dicha comunidad. En la Tabla 10 se muestran los resultados de la evaluación realizada con preguntas en español.

Tabla 9. Distribución de preguntas realizadas.

Tipo de pregunta	Individuos	Preguntas realizadas
Persona	559	111
Espacio Físico	96	19
Evento	200	40
Persona - Espacio Físico	42	8
Espacio Físico - Evento	200	40
Evento - Persona	200	40
Total	**855**	**258**

Tabla 10. Valores de evaluación.

Tipo de pregunta	Respuesta correcta	Respuesta incorrecta
Persona	110	1
Espacio Físico	18	1
Evento	38	2
Persona - Espacio Físico	8	0
Espacio Físico - Evento	38	2
Evento - Persona	38	2
Total	**250**	**8**

Los resultados de la experimentación fueron evaluados utilizando la métrica, bien conocida en la Extracción de Información, denominada precisión y mostrada en la ecuación (3), por un miembro de la comunidad universitaria. Los resultados obtenidos fueron alentadores, logrando un 96.8 % de respuestas correctas y 3.2 % de respuestas incorrectas:

$$P = \frac{resultadosrelevantes}{totaldeconsultas}. \tag{3}$$

5. Conclusiones

En este artículo se ha presentado un método para realizar consultas en idioma español de México dentro de un ambiente académico. Además, se describe un enfoque basado en ontologías como modelo de representación de información, para realizar consultas en *SQWRL* a partir de una consulta semiestructurada utilizando patrones estructurales.

Con respecto a la ejecución de las pruebas realizadas, se obtuvieron resultados prometedores. La arquitectura propuesta conlleva las siguientes contribuciones: (a) la construcción de una *7-tupla* ontológica a partir de técnicas de procesamiento de lenguaje natural en español; (b) la identificación e implementación de tres patrones estructurales con sintaxis en el lenguaje formal *SQWRL* para la consulta en un sistema de ontologías del dominio académico; (c) la adaptación de un método de extracción de información basada en lexicones (conjunto de palabras relacionadas a un mismo tema) en español a una arquitectura de consultas.

Como trabajo a futuro se pretende: a) adaptar el método propuesto a un reconocimiento de voz como entrada y salida de las preguntas realizadas; b) realizar una interfaz gráfica adaptable a distintos dispositivos móviles; c) generación de una respuesta en lenguaje natural.

Referencias

1. SPARQL Tutorial, Apache Jena, Recuperado de https://jena.apache.org/tutorials/sparql.html (2019)
2. Cardoso, J., Lytras, D. M.: Semantic Web engineering in the knowledge society. IGI Global, IRM Press (2009)
3. Villatoro, C.: Desarrollo de una Web API para el Tratamiento Automático de Reglas Contextuales Aplicadas a Servicios Basados en Localización. Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Ciencias Computacionales (2011)
4. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the semantic web. In: Gómez, A., Euzenat, J. (eds) Proceedings of the Second European Semantic Web Conference, ESWC 2005, LNCS, vol 3532, Springer, Berlin, Heidelberg, pp. 546–562 (2005)
5. Muhammad, A., Lipika, D.: An Ontology-Based Pattern Mining System for Extracting Information from Biological Texts. In: Dominik, S., JingTao, Y., James, P., Wojciech, Z., Xiaohua H. (eds) Proceedings of the 10th International Conference, RDFDGrC 2005, LNCS, vol. 3642, pp. 420–429. Springer-Verlag Berlin Heidelberg (2005)
6. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng: A guided input natural language search engine for querying ontologies. Jena User Conference, Citeseer (2006)
7. Tianyong, H., Dawei, H., Liu, W., Qingtian, Z.: Semantic patterns for user-interactive question answering. In: W. Walker, D., Chen, J. (eds) Proceedings of the Second International Conference on Semantics, Knowledge and Grid, SKG2006 2008, LNCS, vol. 20, pp. 783–799. John Wiley & Sons Ltd (2007)
8. Tablan, V., Damljanovic, D., Bontcheva, K.: A Natural Language Query Interface to Structured Information. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds) Proceedings of The Semantic Web: Research and Applications, Proceedings of the 5th European Semantic Web Conference, ESWC 2008, LNCS, vol. 5021, pp. 361–375. Springer, Berlin, Heidelberg (2008)
9. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In: Aroyo, L., Antoniou, G., Hyvönen, E., Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds) Proceedings of the 7th Extended Semantic Web Conference, ESWC 2010, LNCS, vol. 6088, pp. 106–120. Springer, Berlin, Heidelberg (2010)
10. Eka, A., Winarko, E., Azhari, A.: Ontology-based Why-Question Analysis Using Lexico-Syntactic Pattern. International Journal of Electrical and Computer Engineering (IJECE), 5(2), 318–332 (2015)
11. Bereta, K., Xiao, G., Koubarakis, M.: Answering GEOSPARQL queries over relational data. In: Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, LNCS, vol. 42, pp. 43–50. International Society for Photogrammetry and Remote Sensing (2017)

12. Solís, A., Florencia, R., Acosta, J., López, F.: Interfaz de lenguaje natural para deducción de información almacenada en ontologías. Research in Computing Science, 147(6), pp. 189–205 (2018)
13. Setyo, F., Suryana, N., Sanusi, M.: New instances classification framework on Quran ontology applied to question answering system. TELKOMINIKA Indonesian Journal of Electrical Engineering 17(1), pp. 139–146 (2019)
14. Mourad, S., Said, O.E.A.: SemBioNLQA: A semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions. Artificial Intelligence in Medicine, 102, pp. 1–16 (2020)
15. García, G., Reyes, J., González, B., Priego, A.: Extracción de información a partir de preguntas en español basada en lexicones. Academia Journals 11(8), pp. 1054–1059 (2019)
16. Padilla, J.: Detección y representación de eventos en un ambiente académico inteligente. Tesis de Maestría, Universidad Autónoma Metropolitana, Unidad Azcapotzalco (2019)

ISSN 1870-4069

Usando representaciones distribuidas para el análisis de técnicas propagandísticas en noticias falsas

Jennifer Pérez-Santiago, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez

Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE),
Laboratorio en Tecnologías del Lenguaje,
México

{jpsantiago, villasen, mmontesg}@inaoep.mx

Resumen. La propaganda es un mecanismo que busca influir o persuadir al lector manipulando su juicio para fomentar una agenda predeterminada. Este tipo de mecanismos, como las falacias lógicas o apelando a las emociones de la audiencia, son comúnmente usados en las noticias falsas. En este artículo se realiza un análisis sobre la identificación automática de las técnicas propagandísticas más comunes usando diferentes métodos de minería de texto. En específico se aplicaron diferentes representaciones de los textos: *n-gramas* de palabras, *n-gramas* de caracteres, así como representaciones distribucionales (i. e., *embeddings*) de sentencias. La evaluación experimental realizada indica que los *embeddings* de sentencias son una representación adecuada para esta tarea. A través del análisis realizado se distinguen las técnicas más complicadas a identificar y se presentan algunas consideraciones específicas sobre el conjunto de datos utilizados.

Palabras claves: Técnicas de propaganda, *embeddings* de sentencias, clasificación de texto.

Using Distributed Representations for the Analysis of Propaganda Techniques in Fake News

Abstract: Propaganda is a mechanism that seeks to influence or persuade the reader manipulating its judgment to foster a predetermined agenda. These types of mechanisms, such as logical fallacies or appealing to the emotions of the audience, are commonly used in fake news. This article analyzes the automatic identification of the most common propaganda techniques using different text mining methods. Specifically, different representations of texts were applied: n-grams of words, n-grams of characters, as well as distributional representations (eg. embeddings) of sentences. The experimental evaluation carried out indicates that the sentence embeddings are an adequate representation for this task. Through the analysis carried out, the most complicated techniques to identify are distinguished and some specific considerations on the data set used are presented.

Keywords: Propaganda techniques, sentence embeddings, text classification.

1. Introducción

La propaganda es una forma de comunicación, un mecanismo que intenta lograr una respuesta que promueva la intención deseada del propagandista. Este mecanismo es usado, en la mayoría de los casos, en noticias extremadamente sesgadas y/o falsas.

Las noticias en los medios de comunicación pueden ser supuestamente neutrales hasta claramente sesgadas. Un artículo de noticia refleja el sesgo no solo del autor sino también del medio donde se publica. El autor puede no estar consciente de este sesgo o podría ser que el artículo es parte de la propia agenda del autor para persuadir sobre algo o un tema en específico [1]. Esta última situación refleja el uso de la propaganda.

Según el "*Institute for Propaganda Analysis*": la propaganda es un método de comunicación cuyo objetivo es dar a conocer una información con la intención de influir en el público para que actúe de una manera determinada o utilice un determinado servicio o producto. Utiliza técnicas psicológicas y retóricas para alcanzar su propósito. Hace uso del lenguaje emocional para inducir a la audiencia a estar de acuerdo con el hablante solo sobre la base del vínculo emocional que se está creando.

En este trabajo se realiza un análisis para determinar las dificultades en la identificación del uso de técnicas propagandísticas en artículos de noticias. Para ello se considera un conjunto de noticias donde se etiquetan fragmentos de texto identificando el tipo de técnica de propaganda usada. En particular, en este conjunto se distinguen 14 técnicas de propaganda. El análisis se realizó al aplicar técnicas tradicionales de clasificación de textos basadas en representaciones como *n-gramas* de palabras y caracteres, así como *embeddings* de sentencias. El análisis también considera el tamaño del contexto usado: (i) considerando exclusivamente el fragmento propagandístico a clasificar, (ii) considerando el contexto vecino al fragmento; y (iii) considerando la oración completa donde se encuentra el fragmento.

Cabe señalar, que estas técnicas tradicionales nos proporcionan modelos interpretables y la posibilidad de explicar lo que se aprende, para así poder identificar los principales problemas, y posibles alternativas de solución. De ahí, que se dejó fuera de este trabajo las técnicas de aprendizaje profundo actualmente en boga.

El resto del documento está organizado de la siguiente forma. Sección 2 es una revisión de los principales trabajos relacionados, la sección 3 introduce el método propuesto y describe el conjunto de datos utilizado para el desarrollo del experimento, el método de evaluación, los diferentes enfoques de procesamiento del texto y la medida de evaluación utilizada. La sección 4 expone los principales resultados, la sección 5 realiza un análisis de los resultados obtenidos y por último la sección 6 presenta las conclusiones y posibles trabajos futuros.

2. Trabajos relacionados

Debido al desarrollo de los medios de comunicación actuales, como por ejemplo la World Wide Web, el uso de las técnicas de propaganda han experimentado un gran auge.

Es por ello la necesidad de investigación encaminada a la identificación y detección automática de estas técnicas propagandísticas en los diferentes artículos de noticias que se publican a diario en la red.

Barrón-Cedeno et al. (2019)[1] presentaron PROPPY, como un sistema de detección de propaganda para noticias en línea y demostraron que los n-gramas de caracteres y otras características de estilo superan a las alternativas existentes para identificar propaganda basada en n-gramas de palabras.

Por su parte, Li, Ye, and Xiao (2019) [2] desarrollaron una herramienta basada en la regresión logística que clasifica automáticamente si una oración es propagandística o no. Para ello utilizaron una combinación de características lingüísticas y semánticas que arrojaron mejores resultados que varios resultados de referencia. A su vez, Da San Martino et al. (2019) propuso una técnica basada en BERT [3] para identificar problemas de propaganda en los artículos de noticias, considerando múltiples niveles de representaciones semánticas.

Como parte de la tarea compartida de Detección de Propaganda de grano fino: NLP4IF 2019 [4] exiten varios trabajos que abordaron la tarea de detectar y clasificar segmentos textuales que corresponden a una de las 18 técnicas de propaganda indicadas en un conjunto de artículos de noticias. Por ejemplo, Alhindi et al. (2019), propusieron una arquitectura que combina embeddings a nivel de caracteres con diferentes tipos de embeddings de palabras como entrada a un modelo BiLSTM-CRF, agregaron además un total de 30 características conceptuales asociados con palabras específicas, como ofensivo, vulgar, grosero o insulto a las representaciones de e*mbeddings* utilizadas. El mejor modelo obtenido fue un BiLSTM-CRF con G*love embeddings* y características codificadas manualmente, con el que obtuvieron un resultado de F1 = 0.13. De sus experimentos concluyeron que existía ruido en el conjunto de datos y propusieron investigar formas de limpiar aún más los datos. Por su parte, el equipo ganador, [6], propusieron realizar una clasificación a nivel de palabra utilizando BERT, obteniendo un desempeño de F1 = 0.2488. Sin embargo no se analizaron las causas de por qué tan baja puntuación, algo que pretendemos realizar en nuestro estudio.

Cabe señalar que la mayoría de los trabajos que se revisaron, particulamente los de la tarea compartida NLP4IF 2019 [4], abordaron las soluciones propuestas mediante el uso de redes neuronales. Las técnicas de aprendizaje profundo han mostrado altos desempeños en varias tareas de procesamiento de textos, sin embargo estás técnicas son débiles en proporcionar modelos interpretables [7]. Es por ello que nos apoyamos en métodos tradicionales de minería de texto como los *n-gramas* de palabras y caracteres y los *embeddings* para explorar y analizar dificultades de la tarea e identificar posibles problemas sobre el conjunto de datos presentado.

3. Diseño experimental

Los experimentos realizados siguen el enfoque de clasificación supervisada, es decir, dado un fragmento de texto identificado como propaganda y el contexto de su documento, el objetivo es identificar la técnica de propaganda usada de entre las 14 técnicas posibles (detalles en sección 3.1). Posteriormente, a partir de los resultados alcanzados, se analizaron los errores de clasificación para concluir sobre las dificultades de la tarea o las inconsistencias en el conjunto de datos dado.

Las representaciones de los textos usadas son: *n-gramas* de palabras con $n = 1,2,3$ y caracteres con $n = 3,4,5$ respectivamente, así como *embeddings* de sentencias pre-entrenados con USE (*Universal Sentence Encoder*) [8] y LASER (*Language-Agnostic Sentence Representations*) [9], ambos, entrenados y optimizados para texto de más de una longitud de palabra, como oraciones, frases o párrafos cortos. Para USE la entrada es un texto en inglés de longitud variable y la salida es un vector de 512 dimensiones, sin embargo con LASER la salida es un vector de tamaño fijo de 1.024 dimensiones para representar la oración de entrada.

Los algoritmos de clasificación empleados son *Support Vector Machines* (SVM), *Linear Support Vector Machines* (LSVM), y Regresión Logística; para los tres clasificadores utilizamos las versiones implementadas en sklearn con los parámetros establecidos por defecto.

3.1 Conjunto de datos

El conjunto de datos utilizado es proporcionado por la tarea compartida SEMEVAL 2020-TASK 11 "Detección de Técnicas de Propaganda en Artículos de Noticias". Los archivos de entrada para la tarea consisten en artículos de noticias en formato de texto libre recopilados de medios de comunicación propagandísticos y no propagandísticos.

El título está en la primera fila, seguido de una fila vacía y el contenido del artículo comienza desde la tercera fila, una oración por línea. Los fragmentos de propaganda fueron detectados y etiquetados manualmente por expertos.

La distribución del conjunto de datos en entrenamiento, validación y prueba se detallan en la tabla 1.

La Fig. 1 muestra la distribución por cada una de las técnicas de propaganda etiquetadas del número total de instancias, para los conjuntos de entrenamiento y validación.

Tabla 1. Distribución de los artículos y fragmentos etiquetados en el conjunto de datos.

Distribución	**Artículos**	**Fragmentos etiquetados (instancias)**
Entrenamiento	371	6129
Validación	75	1063
Prueba	90	1790
Total	**536**	**8982**

Aunque los datos han sido anotados en un principio con 18 técnicas de propaganda, dada la frecuencia relativamente baja de algunas de ellas, los responsables de la tarea decidieron fusionar técnicas similares en una sola superclase. Convirtiendo la tarea en un problema de clasificación de 14 clases.

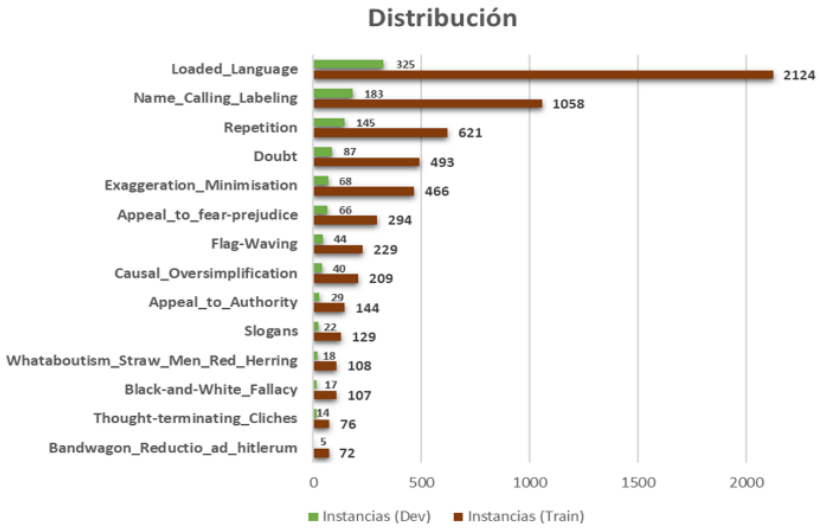


Fig. 1. Estadística sobre el número de instancias por técnica de propaganda para los conjuntos de entrenamiento y validación.

3.2 Técnicas de propaganda

Existen muchos tipos de técnicas de propagandas consideradas en la literatura, en esta sección describimos las 14 técnicas definidas para el desarrollo de esta tarea. La lista solo incluye técnicas que pueden encontrarse en artículos periodísticos y pueden juzgarse intrínsecamente, sin la necesidad de recuperar información de apoyo de recursos externos [3]:

1. **Appeal_to_Authority:** Afirmar que un reclamo es verdadero simplemente porque una autoridad válida o un experto en el tema dijo que era cierto, sin ninguna otra evidencia de respaldo ofrecida. Particularmente poderoso cuando se utilizan referencias religiosas [10]. Ej.: "*Pope Francis explains this again in AL 305 when he says "a small step, in the midst of great human limitations, can be more pleasing to God than a life which appears outwardly in order*"

2. **Appeal_to_fear-prejudice:** Tratar de generar apoyo para una idea, inculcando ansiedad y/o pánico en la población hacia una alternativa, posiblemente basada en juicios preconcebidos. Ej.: "*Ebola victims can be a risk, if mourners have direct contact with the body of the deceased* ".

3. **Bandwagon_Reductio_ad_hitlerum:** Técnica de propaganda que fue fusionada por los responsables de la tarea por dos técnicas diferentes: **Bandwagon:** Intenta persuadir al público objetivo para que se una y tome el curso de acción porque "todos los demás están tomando la misma acción". Está muy relacionada sobre todo con temas políticos [11]. Ej.: "*¿Would you vote for Clinton as president*? *57% say yes*". **Reductio_ad_hitlerum:** Persuadir a una audiencia para que desapruebe una acción o idea sugiriendo que la idea es popular entre grupos odiados por el público objetivo. Puede referirse a cualquier persona o concepto con una connotación negativa [12]. Ej.: "*Only one type of person can think this way: a communist"*. "*Adolf Hitler as a very great man*".

4. **Black-and-White_Fallacy:** Presentar dos opciones, alternativas como las únicas posibilidades, cuando en realidad existen más posibilidades. Como caso extremo, decirle a la audiencia exactamente qué acciones tomar, eliminando cualquier otra opción posible (dictadura). Ej.: "*America will face a choice between accepting a nuclear-armed Iran or acting to destroy as much of this capability as possible the only solution*".

5. **Causal_Oversimplification:** Asume una causa cuando hay múltiples causas detrás de un problema. También incluimos el chivo expiatorio: la transferencia de la culpa a una persona o grupo de personas sin investigar las complejidades de un problema. Ej.: "*the delay signaled the White House was having second thoughts about the nomination"*.

6. **Doubt:** Cuestionando la credibilidad de alguien o algo. Ej.: "*This is evidently of no consequence to a President who cares about nothing about the country and everything about his narrow self-interest*", "*why weren't the police rushing to the scene immediately?*".

7. **Exaggeration_Minimisation:** Representando algo de manera excesiva: haciendo las cosas más grandes, mejores, peores (por ejemplo, "lo mejor de lo mejor", "calidad garantizada") o haciendo que algo parezca menos importante o más pequeño de lo que realmente es (Jowett y O'Donnell, 2012, p. 303). Ej.: " *great big important point*", "*entirely insufficient*".

8. **Flag-Waving:** Jugar con un fuerte sentimiento nacional (o con respecto a un grupo, por ejemplo, raza, género, preferencia política) para justificar o promover una acción o idea. Un truco en el que el propagandista levanta un símbolo, como una bandera, que reconocemos y respetamos [11]. Ej.: "*American people*", "*America First*", "*our democracy*", "*our future*", "*our Republic*", "*our country*".

9. **Loaded_Language:** Usar palabras o frases con fuertes implicaciones emocionales (positivas o negativas) para influir en una audiencia [13]. Ej.: "*hypocritical*", "*filth and degenerate behavior*", "*the nefarious nature of this fascist theocracy*".

10. **Name_Calling_Labeling:** Etiquetar el objeto de la campaña de propaganda como algo que el público objetivo teme, odia, encuentra indeseable o ama o alaba. Apelar al odio o al miedo de una persona para tratar de crear un juicio sobre otra sin evidencia [10]. Truco para hacernos aceptar una conclusión sin tener en cuenta los hechos esenciales del caso [11]. Ej.: "*wanna be statist dictator and naive do-gooder*", "*Muslim Killer*".

11. **Repetition:** Repetir el mismo mensaje una y otra vez, para que la audiencia eventualmente lo acepte. Utilizado por Hitler como una de las principales técnicas fundamentales, si las personas escuchan un mensaje con suficiente frecuencia lo creerán [10]. Ej.: "*autonomy*", se repite alrededor de 25 veces dentro del mismo documento.

12. **Slogans:** Una frase breve y llamativa que puede incluir el etiquetado y los estereotipos. Los lemas tienden a actuar como apelaciones emocionales. Ej .: "*SMASH THE STATE*", "*America First*".

13. **Thought-terminating_Cliches:** Palabras o frases que desalientan el pensamiento crítico y la discusión significativa sobre un tema determinado. Suelen ser oraciones cortas y genéricas que ofrecen respuestas aparentemente simples a preguntas complejas o que distraen la atención de otras líneas de pensamiento [14]. Ej.: "*We all have choices to make*", "*Slowly but surely*", "*too good to be true*".

14. **Whataboutism_Straw_Men_Red_Herring:** Técnica de propaganda que fue fusionada por los responsables de la tarea por tres técnicas diferentes:

    Whataboutism: Desacreditar la posición de un oponente acusándolo de hipocresía sin refutar directamente su argumento. Empleada originalmente por la Unión Soviética y posteriormente continuada por la Rusia post-soviética que busca refutar cualquier crítica

occidental de Rusia por hipocresía [15]. **Straw_Men:** Cuando la propuesta de un oponente se sustituye por una similar que luego se refuta en lugar de la original [16] . Especifica las características de la propuesta sustituida: "caricaturizar una opinión opuesta para que sea fácil de refutar" [13]. **Red_Herring:** Introducir material irrelevante sobre el tema que se está discutiendo, para que la atención de todos se desvíe de los puntos planteados [13].

Los sujetos a un argumento de arenque rojo se alejan del tema que había sido el foco de la discusión y se les insta a seguir una observación o reclamo que pueda estar asociado con el reclamo original, pero que no es muy relevante para el tema en disputa. Ej.: "*don't focus on me, focus on the destructive Radical Islamic Terrorism that is taking place within the United Kingdom*", "*President Trump then reminded everyone of the Obama administration's failures in dealing with Russian meddling in the election*".

3.3 Enfoques de representación de la información de contexto

Para la representación y clasificación de los fragmentos propagandísticos consideramos tres niveles diferentes de información contextual:

- **Enfoque 1, Fragmento:** Solo se usa el fragmento propagandístico definido en los archivos de entrada, sin información de su contexto (i.e. palabras vecinas antes y después del fragmento).
- **Enfoque 2, Contexto:** Se considera el propio fragmento propagandístico y el contexto inmediato a su alrededor conformado desde 1 hasta 3 palabras antes y después de éste.
- **Enfoque 3, Oración:** Se toma en cuenta toda la oración donde se encuentra el fragmento propagandístico a clasificar.

4. Resultados

En las siguientes tablas mostramos los mejores resultados obtenidos para cada uno de los modelos propuestos para los diferentes enfoques de representación del texto propagandístico. El baseline propuesto por la tarea compartida "Detección de Técnicas de Propaganda en Artículos de Noticias, SEMEVAL 2020-TASK 11" usa un clasificador de regresión logística, solo en una característica: la longitud de la oración, obteniendo una medida ***F1 de 0.26529*** para el conjunto de desarrollo.

Para el uso de *n-gramas* de palabras, los mejores resultados fueron representados por los unigramas de palabras, en el caso de los *n-gramas* de caracteres los mejores resultados se obtuvieron para los trigramas de caracteres. (ver Tabla 2). La Tabla 3 ilustra los resultados obtenidos utilizando embeddings de sentencias.

El mejor resultado para todas las evaluaciones fue obtenido mediante embeddings de sentencias (con el modelo USE) usando solo el fragmento propagandístico, y usando Máquinas de Soporte Vectorial Lineal (LSVM). La Tabla 4 muestra las medidas F1 obtenidas para cada una de las técnicas de propagandas etiquetadas con el modelo obtenido con el mejor resultado.

Tabla 2. Resultados con unigramas de palabras y trigramas de caracteres. Se muestran los resultados usando la medida de evaluación micro-F1.

	Clasificador	Oración	Contexto de 3	Contexto de 2	Contexto de 1	Fragmento
Unigrams-palabras	**SVM**	0.309	0.353	0.377	0.406	**0.428**
	LSVM	0.282	0.378	0.407	0.421	**0.457**
	Regresión Logística	0.318	0.381	0.399	0.419	**0.456**
Trigramas-caracteres	**SVM**	0.308	0.375	0.406	0.435	**0.464**
	LSVM	0.276	0.380	0.392	0.435	**0.463**
	Regresión Logística	0.308	0.393	0.415	0.445	**0.459**

Tabla 3. *Embeddings* de sentencias con USE y LASER. Medida de evaluación micro-F1.

	Clasificador	Oración	Contexto de 3	Contexto de 2	Contexto de 1	Fragmento
USE	**SVM**	0.434	0.411	0.434	0.479	**0.517**
	LSVM	0.301	0.423	0.423	0.475	**0.528**
	Regresión Logística	0.296	0.408	0.435	0.475	**0.517**
LASER	**SVM**	0.325	0.458	0.476	0.485	**0.499**
	LSVM	0.326	0.458	0.476	0.492	**0.508**
	Regresión Logística	0.314	0.416	0.435	0.446	**0.475**

Tabla 4. Medidas F1 por clase utilizando *embeddings* de sentencias con USE a nivel de fragmento con LSVM.

Técnicas de Propaganda	F1-LSVM-USE-Fragmento	Datos Entrenamiento	Datos Prueba
Whataboutism_Straw_Men_Red_Herring	0.000	108	18
Black-and-White_Fallacy	0.311	107	17
Flag-Waving	0.000	229	44
Name_Calling_Labeling	0.087	1058	183
Slogans	0.270	129	22
Doubt	0.503	493	87
Thought-terminating_Cliches	0.403	76	14
Appeal_to_Authority	**0.667**	144	29
Exaggeration_Minimisation	**0.673**	466	68
Repetition	**0.574**	621	145
Appeal_to_fear-prejudice	0.284	294	66
Causal_Oversimplification	0.392	209	40
Bandwagon_Reductio_ad_hitlerum	0.000	72	5
Loaded_Language	0.222	2124	325

5. Análisis de resultados

A continuación, se presentan las principales observaciones sobre los distintos aspectos de los modeos elaborados, extraídas a partir del análisis de los errores de clasificación.

Sobre el contexto. Del análisis de los resultados obtenidos lo primero que pudimos apreciar es que para todos los experimentos realizados los mejores valores de F1 se obtuvieron con el enfoque de fragmento, que consiste en observar solo el fragmento propagandístico sin tomar en cuenta su contexto, esto inclusive para fragmentos de una sola palabra. Por otra parte, el uso de toda oración donde se encontraba el fragmento no obtuvo buenos resultados; creemos se debe a que en una misma oración pueden ocurrir más de un fragmento propagandístico para diferentes clases de propaganda.

Sobre la caracterización. Los resultados obtenidos con representaciones tradicionales, n-gramas de palabras y caracteres, fueron en siempre inferiores a los resultados generados por los embeddings de sentencias. De estos últimos USE brindó siempre los mejores resultados, alcanzando un F1 = 0.527 cuando se usó el enfoque de fragmento y el clasificador LVSM.

Sobre la discriminación entre clases. Según los resultados para cada una de las 14 técnicas de propagandas reportadas en la Tabla 4 se puede observar que el modelo tiene problemas distinguiendo prácticamente todas las clases de propaganda. Sin embargo, puede distinguir razonablemente bien clases como: Doubt, Exaggeration and Minimisation, Appeal to Authority, y Repetition. Estas cuatro clases son de las más repetidas o utilizadas en el corpus, así que el tener muchas instancias para el entrenamiento fue un aspecto decisivo.

Sobre la complejidad de las clases. Entre las clases con medida F1 = 0 tenemos el caso de Flag Waving, Whataboutism Straw Men Red Herring y Bandwagon Reductio ad Hitlerum. Para el caso particular de Flag_Waving sucede que su medida F1 es igual a cero para todos los enfoques considerando o no el contexto. Esta técnica de propaganda en particular juega con un fuerte sentimiento nacionalista o de grupo (raza, género, preferencia política) para justificar o promover una acción o idea. Tiende a confundirse con varias clases, pero sobre todo con: Slogans y Doubt.

Slogans, no son más que lemas, frases breves y llamativas que pueden incluir estereotipos y *Doubt*, por su parte es utilizada para cuestionar la credibilidad de alguien o de algo.

Flag_Waving utiliza fragmentos cortos como por ejemplo: *the American People*, *America first*, *our civilization*, *our country*, *our traditions*, our *people*, etc., que tienden a confundirse con fragmentos de *Slogans* como: *America first*, *we will protect our people*, *we will defend our civilization*, *the future is ours and not yours*, *the American people have a* (…).

Para el caso de Doubt, tienden a confundirse con fragmentos como: *the American people are clearly* (…), (…) *prevent it from reaching the American people*, *have the American people been had* (…), (…) *our country and our families*.

Sobre el conjunto de datos. *Whataboutism_Straw_Men_Red_Herring*, es una de las técnicas que es el resultado, como explicamos en la sección 3.1 de la unión de otras

varias técnicas de propagandas, nos referimos a: *Whataboutismo*, que es una técnica que intenta desacreditar la posición de un oponente y constituye la forma más común de evasión de la responsabilidad; *Straw_Men*, es una técnica de propaganda que se propone tergiversar la posición de alguien; y *Red_Herring*, que presenta datos irrelevantes sobre el tema que se está discutiendo para desviar la atención del público y llevar a falsas conclusiones.

Por otra parte, cuando analizamos las instancias del conjunto de entrenamiento correspondiente a esta categoría fusionada observamos fragmentos extremadamente largos que en ocasiones suelen ocupar toda la oración, no siendo así cuando analizamos el conjunto de validación donde los fragmentos son mucho más pequeños, con un tamaño relativo entre 1 a 4 palabras, es por ello que tiende a confundirse con Slogans y Exaggeration_Minimisation.

Bandwagon_Reductio_ad_Hitlerum, constituye el mismo caso de fusión de técnicas de propagandas; *Bandwagon* que consiste en presuadir al público para que se una a una acción o fin común y *Reductio_ad_Hitlerum* que intenta persuadir para que desapruebe una acción o idea, sugiriendo que esta idea es propuesta por grupos, personas o conceptos odiados por el público. La mezcla de ambas técnicas es la de menos instancias tanto para entrenar como para validar en el conjunto de datos, los temas principales de sus fragmentos propagandísticos están encaminados sobre todo a comparaciones o alusiones a la época del fascismo y a la personalidad de Adolfo Hitler. Algunos ejemplos de *Bandwagon_Reductio_ad_Hitlerum* son: *Adolf Hitler as a very great man, to smear Donald Trump as the new Hitler, Hitler exterminating six million Jews, fascist theocracy y Stalinist sense*.

Tiende a confundirse sobre todo con *Slogans*, ya que muchos de sus lemas hacen la misma referencia a esta dictadura y a su principal personaje. Algunos ejemplos de *Slogans*: *long live Hitler*, *to death the Jews*, *Jews are my enemy*.

Loaded_Language, es otra de las técnicas con la que tiende también a confundirse el clasificador en ejemplos como: *Adolf Hitler as a "very great man"*, *the nefarious nature of this "fascist theocracy"*, (…) *most harrowing Stalinist sense*.

Como podemos ver, esta última técnica, *Loaded_Language*, junto con otras como *Slogans* y *Name_Calling_Labeling*, utilizan mucho el contenido enfático, es decir, muchas frases en mayúsculas o entre comillas, y esta es una característica sobre la que no se trabajó y pudiera ser de mucha ayuda en el proceso de clasificación.

6. Conclusiones y trabajo futuro

Para el problema de identificación automática de técnicas de propaganda a nivel de fragmento dado el contexto del documento, experimentamos con varios modelos. Se consideraron diferentes contextos y diferentes caracterizaciones. El modelo de mejor rendimiento obtenido contempla el uso de los embeddings de sentencias de USE a nivel de solo el fragmento propagandístico y con Máquinas de Soporte Vectorial Lineal, logrando un puntaje F1 de 0.528 que supera la línea base propuesta que fue de 0.2653 utilizando Regresión Logística.

A partir del análisis realizado es importante recomponer el corpus equilibrando el número de instancias por clases, o al menos tener una mejor distribución de las 18 técnicas de propagandas con cantidad de instancias suficientes.

Además, es necesario evitar unir clases para evitar ambigüedades. Es necesario evitar oraciones que reúnen varios fragmentos de diferentes técnicas propagandísticas, al menos, para una primera fase del modelo de clasificación. También se descubrió que el tamaño de los fragmentos está en relación a la técnica utilizada y que algunas utilizan marcadores enfáticos, por tanto sería de interés futuro estudiar otras características como: longitud (sentencia, fragmento), uso de contenido enfático (frases en mayúsculas o entre "comillas") y la incorporación del uso de emociones que se pudieran generar de las técnicas usadas.

Agradecimientos. El primer autor agradece el apoyo otorgado por el CONACYT a través de la beca No. 1017548. Los autores también agradecen el apoyo recibido por el CONACYT a través del proyecto de CB-2015-01-257383 para la realización de esta investigación.

Referencias

1. Barrón-Cedeño, A., Jaradat, I., Da San Martino, G., Nakov, P.: Proppy: organizing the news based on their propagandistic content. Inf. Process. Manag., 56(5), pp. 1849–1864 (2019)
2. Li, J., Ye, Z., Xiao, L.: Detection of propaganda using logistic regression. In: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 119–124 (2019)
3. da San Martino, G., Yu, S., Barrón-Cedeño, A., Petrov, R., Nakov, P.: Fine-Grained Analysis of Propaganda in News Articles. In: (EMNLP-IJCNLP) Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, pp. 5636–5646 (2020)
4. da San Martino, G., Barrón-Cedeño, A., Nakov, P.: Findings of the (NLP4IF) shared task on fine-grained propaganda detection. In: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 162–170 (2019)
5. Alhindi, T., Pfeiffer, J., Muresan, S.: Fine-tuned neural models for propaganda detection at the sentence and fragment levels. pp. 98–102 (2019)
6. Yoosuf, S., Yang, Y.: Fine-grained propaganda detection with fine-tuned BERT. In: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pp. 87–91 (2019)
7. Zhou, X., Zafarani, R.: Fake news: a survey of research, detection methods, and opportunities. http://arxiv.org/abs/1812.00315 (2018)
8. Cer, D. et al.: Universal sentence encoder for English. In: EMNLP, Conf. Empir. Methods Nat. Lang. Process. Syst. Demonstr. Proc., pp. 169–174 (2018)
9. Schwenk, H., Douze, M.: Learning joint multilingual sentence representations with neural machine translation. pp. 157–167 (2017)
10. Torok, R.: Symbiotic radicalisation strategies: propaganda tools and neuro linguistic programming. pp. 58–65 (2015)
11. Hobbs, R.: Teaching about propaganda: an examination of the historical roots of media literacy. J. Media Lit. Educ., 6(2), pp. 56–67 (2016)
12. Teninbaum, G.H.: Reductio ad hitlerum: trumping the judicial nazi card. Suffolk Univ. Law Sch. Res. Pap., 09-37(3), pp. 541–78 (2009)
13. Anderson, S.L.: A rulebook for arguments. 10(2) Hackett Publishing (1987)
14. Hunter, J.: Brainwashing in a large group awareness training ? The classical conditioning hypothesis of brainwashing. University of KwaZulu-Natal, Pietermaritzburg (2015)

15. Richter, M.L.: The Kremlin's platform for 'useful idiots' in the west: an overview of RT's editorial strategy and evidence of impact. Kremlin Watch, pp. 53 (2017)
16. Macagno, F., Walton, D., Pragmatics, T. : Interpreting Straw Man. 14, Springer (2017)

ISSN 1870-4069

Selección de características con método wrapper para un sistema de detección de intruso: caso CICIDS-2017

Edson Manuel Ortiz Martínez, Pedro Arguijo, Antonio Hiram Vázquez López, Roberto Ángel Meléndez Armenta

Tecnológico Nacional de México,
Instituto Tecnológico Superior de Misantla,
México

edsonortizm@gmail.com, pedro_arguijo@excite.com,
{jahvazquezl, ramelendeza}@misantla.tecnm.mx

Resumen. Los sistemas de detección de intrusos (IDS) monitorean los eventos de un sistema en una red, siendo responsables de distinguir entre el tráfico normal y el anormal que se considera como algún tipo de ataque. El IDS comúnmente trata con una gran cantidad de tráfico de datos, lo que implica características irrelevantes y redundantes. Consecuentemente, la selección de características es uno de los factores prominentes que influyen en la calidad del IDS. En este trabajo, se analiza el desempeño de un IDS al seleccionar características con dos algoritmos estándar de envoltura (wrapper) Permutation Importance y Boruta, utilizando diversos clasificadores para probar su eficacia. Los resultados obtenidos sugieren que la precisión de los algoritmos considerados no varía al seleccionar distintos subconjuntos de atributos del dataset CICIDS 2017.

Palabras clave: Detección de intrusos, aprendizaje máquina, selección de atributos.

Feature Selection with a Wrapper Method for Intrusion Detection System: Case CICIDS-2017

Abstract. Intrusion Detection Systems (IDS) monitor the events in a network system, being responsible for distinguishing between normal and abnormal traffic which can be considered as an attack. IDS commonly handle a large amount of data traffic, which involves irrelevant and redundant features. Consequently, feature selection is one of the most prominent factors that affect the IDS performance. In this work, IDS performance is analyzed considering feature selection with two standard wrapper algorithms Permutation Importance and Boruta, using various classifiers to test its efficiency. The obtained results suggest that the accuracy of considered algorithms does not vary when selecting different subsets of attributes from the CICIDS 2017 dataset.

Keywords: Intrusion detection, machine learning, feature selection.

1. Introducción

El internet es una herramienta presente en nuestra vida diaria, como resultado de su introducción a diversos ámbitos; entretenimiento, educación, finanzas, etc., un mayor número de usuarios, y en especial empresas, invierten en plataformas de seguridad informática como son antivirus, cortafuegos y otras soluciones para hacerle frente a las amenazas informáticas. Existen antecedentes históricos sobre grandes pérdidas económicas y de datos sensibles debido a amenazas cibernéticas como lo son ataques de denegación de servicio, accesos no autorizados, suplantación de identidad entre muchas otras más amenazas cibernéticas.

Una conocida solución de seguridad informática en un entorno de red es el sistema de detección de intrusos (IDS, por sus siglas en inglés), el cual puede ser un dispositivo o programa que tiene como finalidad detectar amenazas cibernéticas [1]. Un IDS monitorea el tráfico y lo clasifica en dos posibles tipos: normal y malicioso [2] siendo el tráfico malicioso las amenazas cibernéticas y el normal las actividades bien intencionadas. También etiqueta el tráfico y permite a los encargados de la seguridad realizar acciones cuando se presenten ataques cibernéticos [3]. Sin embargo, el IDS funge como un elemento secundario, de apoyo, para otras herramientas orientadas a la seguridad informática como cortafuegos, antivirus, etc. La detección dentro de un IDS se lleva a cabo de dos maneras: mediante firmas que identifican el ataque basándose en patrones previamente establecidos o a través de las características del tráfico de red, ya que el tráfico normal será diferente del anómalo [4].

La detección de tráfico anómalo dentro de los IDS se realiza utilizando diferentes metodologías entre las que sobresale la inteligencia artificial (IA), la cual además es una alternativa útil para detectar ataque desconocidos [5]. La IA utiliza algoritmos de clasificación motivo por el cual la selección de características es un procedimiento útil que evita el sobre entrenamiento, mejora la velocidad de procesamiento y en general el rendimiento de los algoritmos de clasificación [6].

Se han realizado diferentes propuestas que utilizan algoritmos de IA enfocadas a detectar tráfico anómalo en los IDS, para la implementación de éstas destacan los dataset KDD-99 y Kyoto. DARPA propuso KDD-99 y contiene datos sintéticos creados en un entorno militarizado, en este conjunto de datos figuran ataques y tráfico benigno. A pesar de su antigüedad todavía sigue siendo un punto de partida para probar metodologías orientadas a la implementación de IDS, además de existir una versión mejorada NSL-KDD 99 que elimina redundancias y equilibra mejor la presencia de ataques haciéndola más equitativa. Algunas propuestas realizadas involucran la generación de reglas para detección de amenazas utilizando IA con algoritmos genéticos (AG) [7], otras aplican algoritmos evolutivos para mejorar la clasificación ya sea mediante la selección de características [8] o a través de la mejora de los parámetros de clasificadores figurando SVM junto PSO [9-10]. Pawar utiliza NSL-KDD, y realiza la selección de carcterísticas considerando la información de ganancia para mejorar el clasificador [7].

S. Mukherjee utiliza los datos de NSL-KDD e implementa un clasificador bayesiano junto con reducción de características, considerando la correlación entre los atributos seleccionados con la clase objetivo, pero baja entre los mismos atributos, además implementó Information Gain y Gain Ratio [11].En este sentido, otro dataset utilizado para implementar algoritmos de detección es el Kyoto, que tiene como ventaja utilizar

atributos implementados en el dataset KDD-99 pero capturando tráfico real en un amplio periodo de tiempo [13]. Además, Najafabadi clasificó el dataset de Kyoto considerando como entrenamiento el primer día y los demás para la evaluación del modelo, la selección de características la realizó con chi-cuadrada y para clasificar empleó tres algoritmos: k-NN, Decision Tree J48 y Naïve Bayes [13]. Finalmente, Amar sugiere una clasificación utilizando la nomenclatura de -1, 1 y -2. Él dataset lo divide en tres partes para selección de atributos, para aprendizaje y para validación, utiliza PCA y redes neuronales para la selección de atributos [14].

Aunque los dataset mencionados se han utilizado para la implementación de algoritmos de detección, cabe señalar que incluyen ataques cibernéticos desactualizados además de tráfico completamente sintético para el caso de KDD 99 y NSL-KDD, por ello es importante considerar otras alternativas a los dataset mencionados. El dataset CIC IDS 2017 presenta características importantes entre las que figuran un conjunto de ataques variados, un mapeo completo de la red, tráfico de red real, lo cual lo hace ideal debido a que este tipo de tráfico es el que se desea monitorear. En este documento, se analiza el efecto de la selección de características con los algoritmos Boruta y Permutation Importance en el dataset CIC IDS 2017 y se evalúa el desempeño de los subconjuntos de atributos obtenidos con diferentes clasificadores. En la sección dos se describen los métodos de selección de características, así como los algoritmos de clasificación utilizados. Tanto el dataset como el procedimiento de este documento se describen en la sección tres. Los resultados y conclusiones se abordan tanto las secciones cuatro y cinco, respectivamente.

2. Marco teórico

2.1. Selección de atributos

La selección de características es el proceso de seleccionar un subconjunto de características relevantes para construir modelos de aprendizaje robustos. La selección de atributos se clasifica en tres tipos: wrapper, filter e híbridos. El primero emplea un algoritmo de aprendizaje automático para evaluar la fiabilidad de un conjunto de características; entre estos destacan dos algoritmos a mencionar Boruta e Importancia de permutación (Permutation importance). Mientras que el método filter utiliza las características de los datos para evaluar su importancia o rango por medida de distancia, medidas de correlación, medidas de consistencia y medida de información. Finalmente, los métodos híbridos son una combinación de los dos anteriores, siendo útiles cuando existe un gran número de atributos para usar algoritmos del tipo wrapper y el rendimiento del enfoque basado en filtros no es satisfactorio. Este tipo de algoritmo pone en marcha un filtrado para medir la importancia de cada atributo. Posteriormente, dentro del conjunto total se seleccionan diferentes subconjuntos de atributos y se evalúan desde el punto de vista de wrapper.

Boruta. Es un algoritmo de tipo wrapper el cual funciona alrededor de Random Forest y es capaz de trabajar con cualquier método de clasificación que pueda aplicar medidas de importancia de la variable. A continuación, se describe el funcionamiento del algoritmo Boruta:

CONJUNTO ORIGINAL			
Atributo 1	Atributo 2	Atributo 3	Atributo 4
1	2	3	4
9	7	5	4

CONJUNTO SOMBRA			
Atributo 1	Atributo 2	Atributo 3	Atributo 4
2	1	4	3
7	9	4	5

Fig. 1. Conjunto de datos sombra y original.

Atributos		
A	B	C
1	4	3
2	5	4
3	6	5

Atributos		
A	B	C
1	0	3
2	0	4
3	0	5

Fig. 2. Conjunto de atributos, la tabla izquierda contiene el conjunto original y la derecha el conjunto con la variable B permutada.

1. En la Fig. 1 se observa que el algoritmo añade aleatoriedad al conjunto de datos al crear copias barajadas de un número determinado de características (denominadas características sombra u ocultas), por ejemplo, el atributo 1 con valores originales de 1 y 9, son asignados con valores sombra de 2 y 7.
2. El siguiente paso es entrenar un clasificador de Random Forest en el conjunto de datos extendido y aplica una medida de importancia de la característica (el valor predeterminado es Precisión de disminución media) para evaluar la importancia de cada característica, cuando más reducción exista mayor significancia tendrá.
3. En cada iteración, verifica si una característica real tiene mayor importancia que la mejor de sus características sombra (es decir, si la característica tiene una puntuación *Z* más alta que la puntuación *Z* máxima de sus características sombra) y elimina constantemente las características que se consideran poco importantes.
4. Finalmente, el algoritmo se detiene cuando se confirman o rechazan todas las características o cuando alcanza un límite especificado de Decision Tree.

Importancia de la permutación. Es calculada una vez que un algoritmo de clasificación es entrenado. Determina la importancia de los atributos computando el error obtenido en ausencia o ruido de una variable (ver Fig. 2). Si el error es menor o la diferencia es poco significativa cuando se permuta una variable en comparación a cuando no está permutada, entonces la variable no es (muy) importante. Este proceso se lleva a cabo con cada una de las variables para poder calcular la importancia de cada atributo y crear subconjuntos. Los atributos con importancia nula o muy baja pueden ser descartados de manera automática.

2.2. Algoritmos de clasificación

Naïve Bayes. El clasificador ingenuo de Bayes opera bajo un fuerte supuesto de independencia entre los atributos, lo cuales pueden ser tanto continuos como discretos o una combinación de ambos. La independencia de los atributos implica que la probabilidad de un atributo no afecta a la probabilidad del otro. El funcionamiento del clasificador Bayesiano es el siguiente:

1. Los datos de entrenamiento son particionados basándose en las etiquetas de cada clase.

2. Después del proceso de entrenamiento, cuando entre un dato desconocido se buscará la probabilidad a posteriori para cada una de las clases posibles para la tupla desconocida, la clase con mayor posibilidad de existir para una tupla será seleccionada.

Decision Tree. Este tipo de algoritmos crea una estructura en forma de árbol para un determinado conjunto de datos, donde cada nodo representa un atributo y las hojas representan los resultados o clases. Cada nodo debe pertenecer a datos de tipo categórico y en caso de ser continuos se debe discretizar antes del proceso de categorización. La construcción de un árbol se lleva a cabo mediante el método se divide y vencerás, donde se comienza con un nodo principal que contiene el conjunto total de datos y posteriormente se van dividiendo los elementos del nodo en subconjuntos que se localizarán dentro de otros nodos y estos a su vez se dividirán de manera iterativa hasta lograr homogeneidad en los nodos hoja.

Decision Tree tipo CART. Es un tipo de Decision Tree binario que se construye mediante la división de un nodo en dos nodos hijo repetidamente, comenzando con la raíz que contiene todas las muestras para el aprendizaje. Consiste en tres pasos que se repiten de manera iterativa:

1. Buscar el mejor punto para dividir cada atributo.
2. Encontrar el mejor nodo con base en el aumento de la homogeneidad en los nodos hijo.
3. Si las condiciones no son satisfechas, dividir de nuevo comenzando con el paso uno.

El algoritmo continuará si todavía no existe la suficiente homogeneidad dentro de las muestras en los nodos hijo, esto se puede calcular con diversas medidas como lo es la entropía y el índice Gini. También es posible detener el algoritmo si el número de muestras que queda por dividir no es muy grande.

Random Forest. Este algoritmo es un meta clasificador, es decir, ocupa un determinado número de clasificadores débiles para construir uno fuerte. En especial, los Random Forest implementan un conjunto de Decision Tree. Para propósitos de clasificación, al haber un dato entrante cada Decision Tree existente realiza de manera individual un proceso de clasificación votando por una clase en especial, la clase con un mayor número de votos es la ganadora.

AdaBoost. Es un meta clasificador de tipo boosting que permite a clasificadores débiles mejorar su rendimiento mediante el uso de varios clasificadores, ya que adapta el error de clasificación de los algoritmos básicos asignándoles un peso a cada uno.

3. Materiales y métodos

3.1. Dataset

La base de datos CIC IDS 2017 se creó en la Facultad de Ciencias de la Computación de la Universidad de Nueva Brunswick en 2017. Contiene un total de 2,827,876 paquetes con 79 atributos que en conjunto alcanzan 5 días de datos reunidos en un periodo mayor a 7 días de actividad de red. Este dataset contiene tráfico benigno y ataques, los cuales representan datos del mundo real, además tener datos de ataques relativamente recientes como HeartBleed, por ejemplo.

Entre los atributos que contiene se destacan la dirección IP, puerto, protocolo, tipo de servicio, etc. También contiene resultados de análisis de red obtenidos mediante la herramienta CICFlowMeter con sus respectivos atributos dentro de un archivo CSV. Para este dataset su utilizó la representación de la actividad perteneciente a 25 usuarios con protocolos HTTP, HTTPS, FTP, SSH, y protocolos de correo electrónico. Se capturaron datos en diferentes días de la semana y a diferentes horas del día. Cada fecha de captura contuvo diferentes ataques además de tráfico normal de red. Los ataques que se incluyen son: Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS [15].

Los criterios implementados para la realización de esta base de conocimientos incluyeron una amplia variedad de circunstancias las cuales involucran una arquitectura de red con switches, enrutadores, módem, cortafuegos además de múltiples sistemas operativos: MacOS, Windows y Ubuntu. La interacción entre los dispositivos de red fue amplia además de que el tráfico resultante fue mapeado completamente y la cantidad de ataques llevada a cabo fue diversa [15].

3.2. Utilerías

Se utilizó una laptop HP modelo 17z-db000, con sistema operativo Windows 10, procesador AMD Ryzen 5 2500U con 16 GB de memoria RAM (DDR4) y un disco duro de 916 GB. Además, se programó en Python 3.7 para 64 bits. Para el preprocesamiento, división de dataset y clasificación se empleó la librería Scikit-learn. Para implementar el selector de características Boruta se usó la librería BorutaPy, la cual a su vez envolvía al algoritmo Random Forest (perteneciente a Scikit-learn), también, se aprovechó la librería eli5 para el selector de atributos Permutation Importance.

3.3. Procedimiento

El procedimiento propuesto involucró los siguientes pasos: normalización, división de la base de datos, selección de características, entrenamiento y validación tal y como se muestra en la Fig. 3.

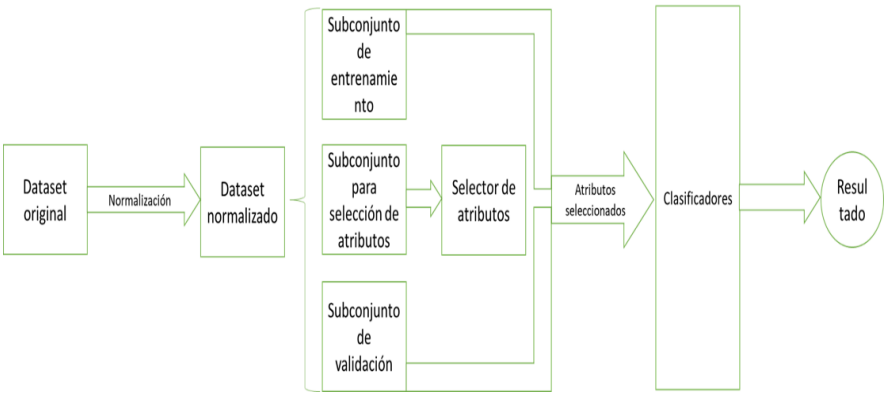


Fig. 3. Descripción del método propuesto.

Tabla 1. Precisión de los algoritmos de clasificación señalados considerando todos los atributos del CIC IDS 2017.

Clasificador	Precisión
Decision Tree	99.81%
Random Forest (64 clasificadores)	99.82%
AdaBoost (100 perceptrón)	85.33%
Multinoamial Naïve Bayes	84.85%
Gaussian Naïve Bayes	70.57%
AdaBoost (100 árboles)	85.69%

El dataset se dividió en tres subconjuntos, 20% para entrenamiento, 20% para selección de características y 60% para pruebas. Para la selección de características se consideraron los algoritmos de Boruta y Permutation Importance, descritos anteriormente. Los clasificadores implementados fueron Decision Tree, AdaBoost, Random Forest y Naïve Bayes. Cabe mencionar que el desempeño de cada algoritmo se analizó con y sin selección de atributos, es decir, con el total de atributos del dataset. Se llevaron a cabo pruebas con los mejores atributos partiendo desde el primer atributo, segundo y de manera iterativa hasta obtener el conjunto total de atributos, excepto con el algoritmo Boruta, ya que éste entrega un conjunto de características óptimo.

4. Resultados

En esta sección se muestran los resultados obtenidos al aplicar los algoritmos previamente descritos al dataset CIC IDS 2017. En la Tabla 1 se puede observar la precisión obtenida en cada uno de los clasificadores al considerar todos los atributos del dataset, dado que la precisión es una métrica utilizada en la eficacia de los

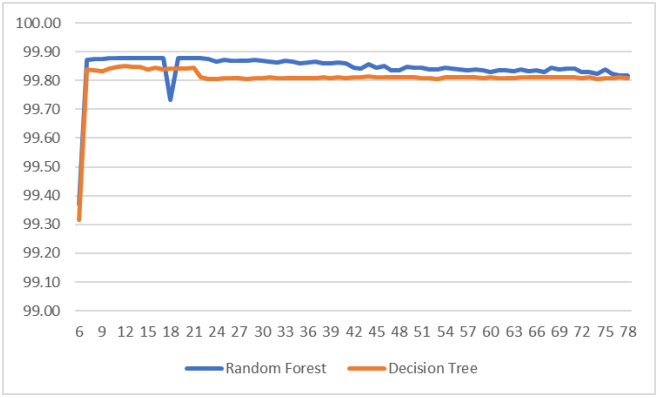


Fig 4. Comparación de la precisión entre algoritmos de Random Forest y Decision Tree.

Tabla 2. Precisión de diferentes algoritmos considerando la cantidad de mejores atributos indicados.

	Atributos						
Clasificadores	20	30	40	50	60	70	Todos
Random Forest	99.88 %	99.87 %	99.86 %	99.85 %	99.83 %	99.84 %	99.82 %
Decision Tree	99.84 %	9981%	99.81 %	99.81 %	99.81 %	99.81 %	99.81 %
Gaussian Naïve Bayes	71.42 %	72.24 %	73.04 %	71.23 %	70.52 %	69.53 %	70.57 %
Multinomial Naïve Bayes	86.47 %	87.12 %	87.04 %	86.56 %	86.60 %	86.35 %	84.84 %
AdaBoost (100 árboles)	85.70 %	85.69 %	85.69 %	85.69 %	85.69 %	85.69 %	85.69 %
AdaBoost (100 Perceptrones)	92.20 %	91.17 %	89.45 %	98.23 %	98.26 %	96.69 %	96.48 %

clasificadores. Los resultados indican que Random Forest arrojó la mejor clasificación seguido por Decision Tree. En estos dos casos la precisión supera el 99%.

En la Tabla 2 se exhibe la precisión de los diferentes clasificadores utilizando las *n* mejores características obtenidas con Permutation Importance. Tanto Decision Tree como Random Forest arrojan resultados de clasificación del 99% a partir de utilizar las 6 mejores características. Bayes Multinomial muestra un desempeño del 87% al considerar los primeros 38 atributos. Mientras que AdaBoost con árboles muestra una precisión constante desde los 27 atributos considerados para la clasificación y manteniéndose sin cambios posteriormente sin importar el número de atributos que se agreguen.

Al agregar cada vez más características los clasificadores tuvieron una tendencia a incrementar su precisión, pero ocasionalmente presentaron intervalos con pequeños pero significativos decrementos y/o incrementos que contenían máximos globales.

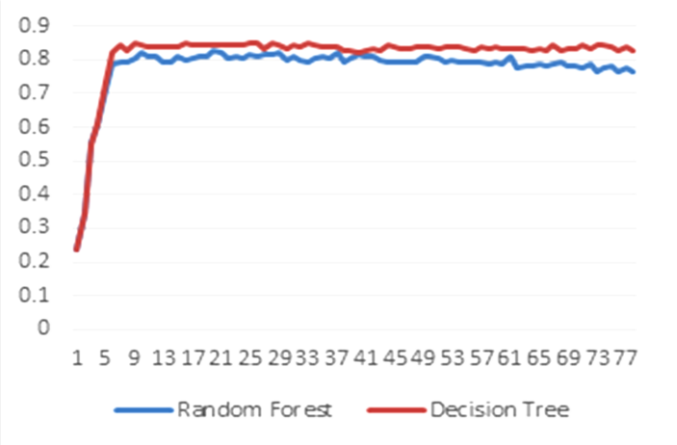


Fig 5**.** Comparación de la sensibilidad entre algoritmos de Random Forest y Decision Tree.

Tabla 3**.** Numero de atributos óptimos para cada clasificador.

Clasificador	**Permutation Importance Precisión, N atributos**	**Boruta**
Decision Tree	99.85%, N=12	99.81%
Random Forest (64 árboles)	99.88%, N=13	99.82%
AdaBoost (100 perceptrón)	92.56%, N=45	74.51%
Multinomial Naïve Bayes	87.32%, N=38	86.00%
Gaussian Naïve Bayes	75.19%, N=26	71.97%
AdaBoost (100 árboles)	85.70%, N=21	85.69%

Como se puede verificar en la Fig. 4, los algoritmos de Decision Tree y Random Forest superaron el 99% de precisión. Ambos exhibieron un desempeño relativamente estable, el cual no variaba demasiado al incrementar el número de atributos. También se puede observar una tendencia por parte del Random Forest a superar levemente a Decision Tree. Por otro lado, el Decision Tree posee una mayor velocidad de procesamiento debido a que Random Forest al ser un meta algoritmo contiene múltiples Decision Tree, por lo cual realiza el proceso de clasificación con menor rapidez.

Por otra parte, Decision Tree mostró una mayor sensibilidad que Random Forest prácticamente en todo momento como se muestra en la Fig. 5.

Por su parte el algoritmo Boruta hizo una valoración de los atributos y únicamente los mejores se toman en cuenta para su uso. En este caso, se obtuvieron 62 atributos. La Tabla 3 muestra el número óptimo de atributos con cada algoritmo, haciendo el uso de Permutation Importance y Boruta.

La matriz de confusión de la Fig. 6 y la Fig. 7 muestran los resultados obtenidos al implementar Boruta junto a Decision Tree y Random Forest.

Fig. 6. Matriz de confusión resultante de la implementación de Decision Tree y Boruta.

5. Conclusiones

Esta investigación ha encontrado que la eficacia de los algoritmos de clasificación en la detección de ataques cibernéticos no está, necesariamente, influenciada a la complejidad de estos.

Los resultados demuestran una mayor eficacia en la clasificación del Decision Tree CART en relación con algoritmos más complejos como AdaBoost y Naïve Bayes, superando a la mayoría de los algoritmos implementados en este estudio, en términos de velocidad de clasificación (excepto de Multinomial Naïve Bayes) y a casi todos en precisión, siendo superado levemente por Random Forest por lo cual se puede argumentar que el Decision Tree CART es un candidato excelente para utilizarse en IDS basados en la detección de actividad anómala al que considera el dataset CIC IDS 2017.

Es importante señalar que se necesitan investigar más selectores de características, ya que no se incluyen los de tipo filter ni híbridos en este trabajo.

Fig 7. Matriz de confusión resultante de la implementación de Random Forest y Boruta.

Referencias

1. Stampar, M., Fertalj, K.: Artificial intelligence in network intrusion detection. In: Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1318–1321 (2015)
2. Gharaee, H., Hosseinvand, H.: A new feature selection IDS based on genetic algorithm and SVM. In: International Symposium on Telecommunications (IST), pp. 139–144 (2016)
3. Kanika, U.: Security of network using ids and firewall. International Journal of Scientific and Research Publications, 3(6), pp. 1–4 (2013)
4. Kanimozhi, V., Jacob, D.T.P.: Calibration of various optimized machine learning classifiers in Network Intrusion Detection System on the Realistic Cyber Dataset Cse-Cic-Ids2018 Using Cloud Computing. International Journal of Engineering Applied Sciences y Technology, 4(6), pp. 209–213 (2019)

5. Li, Z., Qin, Z., Shen, P., Jiang, L.: Zero-shot learning for intrusion detection via attribute representation. Neural Information Processing, Lecture Notes in Computer Science, pp. 352–364 (2019)
6. Zuech, R., Khoshgoftaar, T.M.: A survey on feature selection for intrusion detection. In: Proceedings of the 21st ISSAT International Conference on Reliability and Quality in Design, pp. 150–155 (2015)
7. Pawar, S.N., Bichkar, R.S.: Genetic algorithm with variable length chromosomes for network intrusion detection. International Journal of Automation and Computing, 12(3), pp. 337–342 (2015)
8. Li, J., Zhao, Z., Li, R., Zhang, H.: AI-based two-stage intrusion detection for software defined IOT networks. IEEE Internet of Things Journal, 6(2), pp. 2093–2102 (2019)
9. Gao, F.: Proceedings Qingdao Oulu. Academy Publ., pp. 21–22 (2009)
10. Chakir, E.M., Khamlichi, Y.I.: An effective intrusion detection model based on SVM with feature selection and parameters optimization. Journal of Theoretical and Applied Information Technology, 96(12) (2018)
11. Mukherjee, S., Sharma, N.: Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology, 4, pp. 119–128 (2012)
12. Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BIONETICS) (2016)
13. Najafabadi, M.M., Khoshgoftaar, T.M., Seliya, N.: Evaluating feature selection methods for network intrusion detection with kyoto. International Journal of Reliability, Quality and Safety Engineering, 23(1), pp. 1650001 (2016)
14. Ammar, A.: Comparison of feature reduction techniques for the binominal classification of network traffic. Journal of Data Analysis and Information Processing, 3(2), pp. 11–19 (2015)
15. University of New Brunswick: Datasets. https://www.unb.ca/cic/datasets/ids-2017.html (2020)

ISSN 1870-4069

Evaluación del abandono de clientes de una compañía de telecomunicaciones por medio de cuatro modelos de aprendizaje máquina

Jorge Octavio Castro Rodríguez, Ernesto Pérez Vázquez

Universidad Panamericana,
Facultad de Ingeniería,
México

{0234588,0234686}@up.edu.mx

Resumen. Debido a que los clientes son el activo más preciado de las compañías, la retención de clientes es un objetivo esencial dentro de la estrategia de desarrollo de cualquier organización. En un entorno donde los servicios, productos y valor añadido están en constante evolución se vuelve cuestión de supervivencia contar con una estrategia de retención de clientes. Basados en información histórica existente en combinación con herramientas tecnológicas de procesamiento de información el aprendizaje de maquina (ML) nos permite realizar la extracción de conocimiento de una manera poderosa. Este proyecto implementó las siguientes técnicas de ML: Regresión logística, Random forest, SVM y XGBoost. Los resultados representan las características de los clientes que abandonaron. El negocio puede identificar los abandonos que sucederán usando los resultados de las técnicas empleadas. Con base a dicha información la organización debe implementar estrategias de retención para aquellos clientes más propensos a abandonar al negocio y cuyas características estén apegadas a los objetivos de desarrollo organizacional.

Palabras clave: Aprendizaje máquina, aprendizaje supervisado, ajuste de hiperparámetros, principio de parsimonia, abandono de clientes.

Evaluation of Customer Churn of a Telecom Company Using Four Machine Learning Models

Abstract. Clients are the most valuable asset of a company. Therefore, retaining them is a vital part of the strategy inside any organization. Nowadays, services, products and added value are constantly evolving, which makes it critical to possess a strategy for retaining customers. Based on existing historic data in combination with tech tools of processing data for machine learning (ML), we show how to make knowledge extraction in a powerful manner. In this project, we implement the

following techniques of ML: Logistic Regression, Random Forest, SVM and XGBoost. The results represent features of CHURN customers. The company can identify where CHURN occurs by using the results of those techniques. Based on this, a business may implement strategies for the more likely CHURN customers in which features are attached to the objectives of an organizational development.

Keywords: Machine learning, supervised learning, hyperparameters optimization, principle of parsimony, customer CHURN.

1. Introducción

Haciendo énfasis en la base de que es mucho más costoso captar un nuevo cliente que retener a los clientes actuales y el hecho que a periodos más largos de permanencia los clientes generan mayores ganancias, [1] afirma que la retención de clientes incrementa la rentabilidad.

Hoy día las compañías invierten grandes cantidades de recursos económicos, tecnológicos, humanos y estratégicos para aumentar el número de clientes que se tienen registrados en la base de datos histórica del negocio, sin embargo, no se tiene una estrategia clara y enfocada para retener a los clientes actuales pero sobre todo retener a los clientes ideales (clientes que encajen en el tipo y estrategia de nuestro negocio).

Con base a lo anterior es sumamente importante tener conocimiento de los puntos que se darán a continuación, ya que en cualquier negocio donde los costos de captación son elevados y los márgenes de ganancia del producto o servicio son pequeños, la manera más eficiente para generar rentabilidad es con base al consumo recurrente de los productos y/o servicios de la compañía. Por tanto, minimizar la tasa de abandono de los clientes adecuados se vuelve un pilar de supervivencia de las compañías.

- **Conocimiento de clientes.** Conocer cómo se segmentan nuestros clientes para tener claridad de sus gustos, comportamiento he intereses y de esta manera enfocar la creación de nuevos productos y/o servicios de manera dirigida a captar dichas necesidades.
- **Acciones históricas.** Análisis de las acciones históricas que se han realizado para fidelizar a los clientes y a partir de ello identificar la efectividad de dichas estrategias y de esta manera enriquecer la base de información que alimente tanto al conocimiento de nuestros clientes (segmentación) y de esta manera maximizar las opciones y niveles de retención de nuestros clientes.
- **Optimización.** Mejora continua de nuestros servicios, productos, promociones, ofertas comunicados, etc., que genere un continuo y genuino interés en nuestra compañía y en lo que ofrece (optimizar valor agregado de la compañía).

Dado a lo anterior, el objetivo establecido para este artículo es ejecutar una metodología de ciencia de datos que nos permita llevar a cabo la implementación de cuatro modelos distintos de aprendizaje automático supervisado, a partir de un set de datos históricos en común, y con base a los resultados de cada uno de los modelos realizar un análisis comparativo que nos ayude finalmente a determinar el modelo óptimo.

Para llevar a cabo la implementación de los modelos elaboraremos para cada uno de ellos los siguientes pasos generales (en la sección número cuatro se describe a mayor detalle las etapas de la metodología implementada):

- Recolección de datos (set de datos compartidos).
- Pre-procesamiento.
- Extracción de características.
- Selección de características.
- Creación de modelos.
- Evaluación de modelos.

Este documento se encuentra estructurado de la siguiente manera:

- **Trabajos relacionados.** En esta sección se observan los trabajos históricos de autores distintos que se tomaron como referencia base para el desarrollo de este artículo, dichos trabajos y autores fueron seleccionados debido a las similitudes con respecto a nuestro objetivo inicial (financiero-banca) como con nuestro objetivo final (Telecomunicaciones - considerando las limitaciones)
- **Limitaciones.** En este apartado se describen los elementos restrictivos para llevar el estudio presentado en este documento
- **Material y método.** En este bloque narra la propuesta del trabajo, los materiales utilizados y la metodología llevada a cabo
- **Experimentación.** Se describen los experimentos realizados y las métricas de evaluación
- **Resultados y discusión.** En este módulo se plasman los resultados obtenidos por cada uno de los modelos con la finalidad de abrir el debate dirigido a determinar el modelo más adecuado para el set de datos utilizados
- **Conclusiones.** Sección que se utiliza para detallar las conclusiones del proyecto y consideraciones futuras del estudio.

"La infidelidad siempre tiene como constantes la insatisfacción, desapego y desconfianza".

2. Trabajos relacionados

En Keramati et al. [2] se enfocan en la predicción de abandono de clientes en los servicios de banca electrónica utilizando el método de árbol de decisiones de aprendizaje máquina. Raj [3] utiliza aprendizaje máquina para predecir el abandono de clientes de telecomunicaciones haciendo uso de KNN, random forest,

Gaussian Naive Bayes, Support Vector Machine, LightGBMClassifier, XGBoost. Leon Palafox muestra el proceso para el entendimiento de negocio perfilado al abandono de cliente, entendimiento de datos y su exploración, Guo-en and Wei-dong [4] se enfocan en construir un modelo de predicción de abandono de clientes usando SVM en la industria de telecomunicaciones. De manera tabular los podemos resumir en la figura 1:

Investigador y/o año	Industria	Metodo(s)
AbbasKeramati, HajarGhaneei and SeyedMohammadMirmohammadi (2016)	Banca	Árbol de decisión
Pavan Raj (2018)	Telecomunicaciones	KNN, random forest, Gaussian Naive Bayes, Support Vector Machine, LightGBMClassifier, XGBoost
Leon Palafox	Telecomunicaciones	Análisis exploratorio

Fig. 1. Trabajos relacionados.

3. Limitaciones

Inicialmente se tenía planeado realizar el ejercicio con una base de datos de Nacional Monte de Piedad, sin embargo, no fue posible debido a las restricciones de la institución para compartir información productiva lo que represento un limitante respecto al desarrollo de este estudio, por lo que se optó por trabajar con información pública del sitio Kaggle. La investigación futura tendrá como objetivo realzar este ejercicio con información productiva esperando lograr superar los problemas de privacidad asociados, de igual manera se planea programar un despliegue el cual no se ejecutara debido al corto tiempo con el cual se contó para obtener las autorizaciones correspondientes.

4. Materiales y método

El estudio se llevó a cabo conforme al esquema denominado CRISPDM, (en la figura 2 podemos observar el flujo general utilizado):

- **Entendimiento del negocio** El entendimiento del negocio debe partir del conocimiento de los objetivos estratégicos de las organizaciones ya que a partir de dicho conocimiento se pueden establecer líneas base de observación y/o acción. Debido a las limitantes del set de datos se entiende que este apartado queda pendiente para futuros ejercicios, sin embargo, cabe recalcar

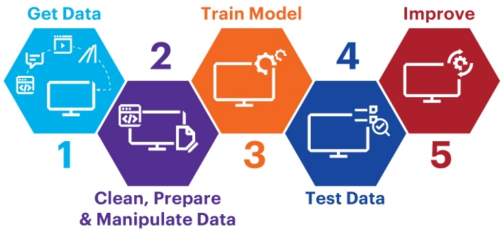
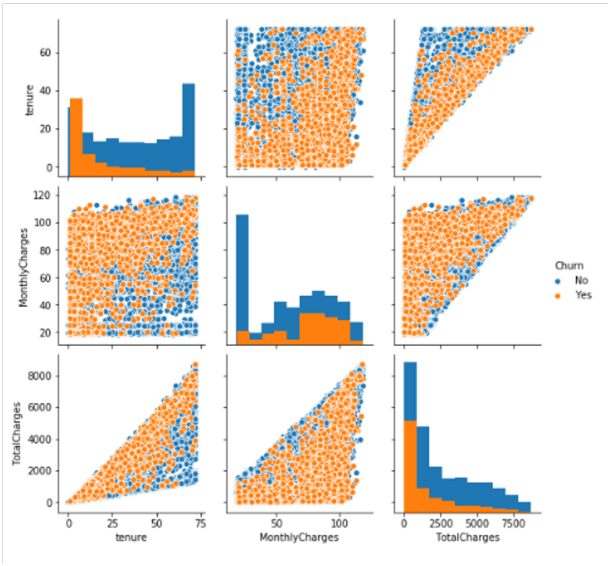


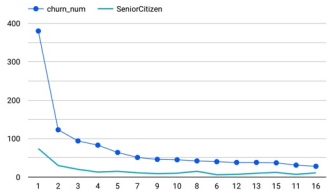
Fig. 2. Flujo de trabajo.

que con base al entendimiento de la metodología empleada este estudio podrá ser ejecutado con mayor exactitud y experiencia en cuanto se logre obtener información productiva.

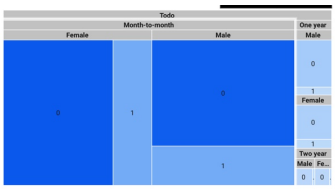
- **Entendimiento de los datos** Como parte de la exploración de datos descubrimos comportamientos valiosos para la toma de decisiones, por ejemplo, figura 3:
  - El abandono temprano de clientes jóvenes.
  - Las personas que pagan menos permanecen por mayor tiempo.
  - Las personas que cuentan con el servicio de telefonía tienden a tener una taza alta de abandono.
  - Las personas que utilizan fibra óptica tienden a presentar una taza alta de abandono.
  - El abandono no se relaciona con el género del cliente.
  - Al finalizar el primer mes de servicio es donde se nota la mayor tasa de abandono.
  - La correlación de las variables pertenecientes al set de datos.
  - Con forme los clientes permanecen mayor tiempo con el servicio la tasa de abandono disminuye.
- **Preparación de los datos** La información se preparó mediante Python. Dado que encontramos casos en los cuales tuvimos que eliminar espacios (TotalCharges) o eliminar valores vacíos (TotalCharges), se opto por excluir 11 observaciones por contener en una columna valores vacíos, normalizamos los valores, excluimos las variables gender y customerID ya que con base al análisis exploratorio observamos que no tienen valor representativo, como se muestra en la figura 3. Para efectos de este estudio el set de datos fue dividido en dos subsets de datos, el primero de ellos contiene el 70 porciento de las observaciones y fue reservado para la etapa de entrenamiento, el 30 porciento de la información fue utilizada para la etapa de evaluación.
- **Modelado** Se detalla en el apartado 5 (Experimentación) de este documento.
- **Evaluación** Se detalla en el apartado 5 (Experimentación) de este documento.
- **Despliegue** Se observa en la sección de limitantes.



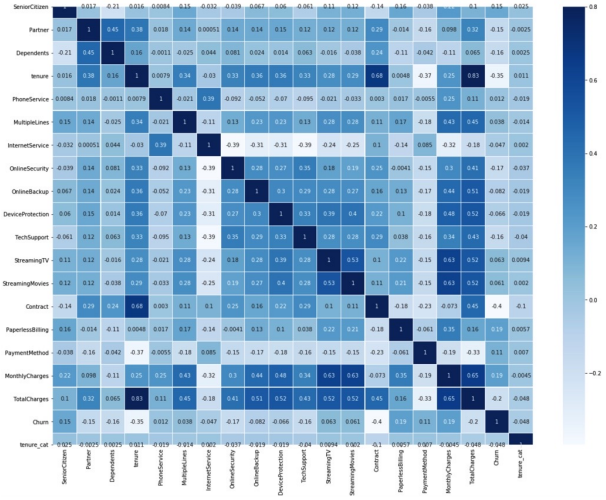
(a) Análisis exploratorio.



(b) Abandono por mes por Seniority.



(c) Relación de abandono por genero.



(d) Matriz de correlación.

Fig. 3. Análisis exploratorio.

4.1. Descripción general

Se programará la ejecución de cuatro modelos de aprendizaje maquina con el objetivo de seleccionar el modelo que nos proporcione la alternativa óptima para apoyar a la toma de decisiones, en la figura 4 se puede observar las variables utilizadas para evaluar los cuatro modelos para el caso de estudio de abandono de clientes.

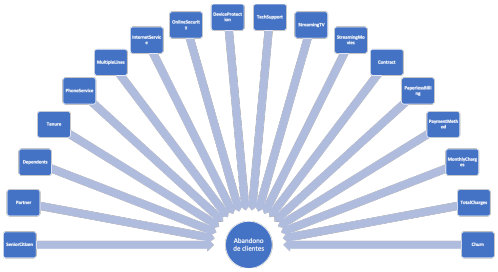


Fig. 4. Mapa conceptual para un modelo de abandono de clientes.

4.2. Materiales

Para llevar a cabo el caso de estudio utilizamos las siguientes herramientas: Python v3, Google Colaboratory, Google data studio, Anaconda.

El resumen del set de datos utilizado se describe en la figura 5.

Artefacto	Descripción
Total de observaciones analizadas	7043
Formato del archivo	csv
Columnas totales	21
Duplicados customerID	0
Columnas homologadas (descripciones)	7
Edición de tipo de dato de columna existente (TotalCharges, String to Int)	1
Variables nulas	11 (TotalCharges)
Entendimiento de columna SeniorCitizen	1=Sr (mayor), 2=Jr (joven)
Entendimiento de columna tenure	Meses de permanencia
Entendimiento de columna PhoneService	Telefonía móvil
Total de variables	19
Variables excluidas (customerID, gender)	2
customerID (string)	7043
Gender (category)	2 tipos
SeniorCitizen (Integer)	2 tipos
Partner (category)	2 tipos
Dependents (category)	2 tipos
Tenure (Integer)	73 tipos
PhoneService (category)	2 tipos
MultipleLines (category)	3 tipos
InternetService (category)	3 tipos
OnlineSecurity (category)	3 tipos
OnlineBackup (category)	3 tipos
DeviceProtection (category)	3 tipos
TechSupport (category)	3 tipos
StreamingTV (category)	3 tipos
StreamingMovies (category)	3 tipos
Contract (category)	3 tipos
PaperlessBilling (category)	2 tipos
PaymentMethod (category)	4 tipos
MonthlyCharges (Integer)	1585 tipos
TotalCharges (Integer)	6531 tipos
Churn (category)	2 tipos

Fig. 5. Resumen del set de datos.

4.3. Métodos de aprendizaje

Los métodos de aprendizaje máquina utilizados en este trabajo se describen a continuación.

Regresión logistica. Chitarroni [5] define la regresión logística como un instrumento estadístico de análisis multivariado, de uso tanto explicativo como predictivo. Resulta útil su empleo cuando se tiene una variable dependiente dicotómica (un atributo cuya ausencia o presencia hemos puntuado con los valores cero y uno, respectivamente) y un conjunto de variables predictoras o independientes, que pueden ser cuantitativas (que se denominan covariables o covariadas) o categóricas.

- **Ventajas:** Al ser un método de clasificación fácil, rápido y simple, tendremos un menor costo computacional que los otros modelos que usaremos para ajustar este set de datos, de esta manera se tiene la opción de usar un rango más amplio de hiperparámetros lo que posiblemente nos dará entrada a tener un mejor desempeño del modelo ajustado.
- **Desventajas:** En este tipo de modelo se requiere de una adecuada selección de características para que este tenga un mayor rendimiento. Dado esto se necesita pensar que al hacer una selección de características mayormente enfocada a la visión de negocio podríamos tener alguna desventaja en este punto.

Random forest. Random Forest es una combinación de árboles de decisión, tal que cada árbol depende de los valores de un vector aleatorio, independiente y con la misma distribución para cada uno de estos [6]. Todos los árboles tienen la misma distribución en el bosque (forest), pero son forzados a ser diferentes. Esto reduce la correlación. Existen diversas técnicas de construcción de conjuntos de clasificadores. En muchas ocasiones, la forma de llevar a cabo estas técnicas es a partir de un conjunto de números aleatorios:

- En Bagging Breiman [7], cada clasificador se construye a partir de una muestra bootstrap, la cual se genera utilizando tantos números aleatorios como elementos tenga el conjunto de entrenamiento.
- En random subspace Ho [8], cada clasificador base utiliza sólo un subconjunto de atributos seleccionado aleatoriamente de entre el total de las variables. De esta forma, cada clasificador se restringe a un subespacio aleatorio de atributos.
- Las técnicas output flipping y class switching, Breiman [9]); Martínez-Muñoz and Suárez [10], se basan en la manipulación aleatoria de las etiquetas de clase, por lo que necesitan de números aleatorios para seleccionar los datos cuyas etiquetas serán cambiadas.
- Randomization Dietterich [11], introduce aleatoriedad en el algoritmo de aprendizaje construyendo conjuntos de clasificadores con árboles de decisión en los que el valor de corte es seleccionado aleatoriamente entre los F mejores cortes posibles.

A continuación, se describen las ventajas y desventajas mas representativas del uso de este modelo:

- **Ventajas:** Este un modelo que funciona bien para métodos de clasificación y parte de sus bondades es que la preparación de los datos que requiere es mínima y al utilizar múltiples árboles se logra reducir el riesgo de sobre ajuste de nuestro modelo.
- **Desventajas:** En este modelo a diferencia del anterior el costo computacional y tiempo de entrenamiento se incrementará con base en los hiperparámetros a usar. Por otro lado, nos enfrentaremos a problemas como el no poder garantizar que el árbol generado sea el óptimo.

Si el clasificador base utilizado es un árbol de decisión, el concepto random forest [12] engloba todas estas técnicas.

XGBoost. Extreme Gradient Boosting (XGBoost), es una técnica de clasificación supervisada que utiliza un conjunto de árboles de decisión. XGBoost es un algoritmo de Machine Learning basado en un árbol de decisiones que utiliza un marco de impulso de gradiente. En los problemas de predicción que involucran datos no estructurados (imágenes, texto, etc.), las redes neuronales artificiales tienden a superar a todos los demás algoritmos o marcos. Sin embargo, cuando se trata de datos estructurados / tabulares de pequeño a mediano, los algoritmos basados en el árbol de decisión se consideran los mejores en su clase en este momento. La figura 6 muestra la evolución de los algoritmos basados en árboles a lo largo de los años.

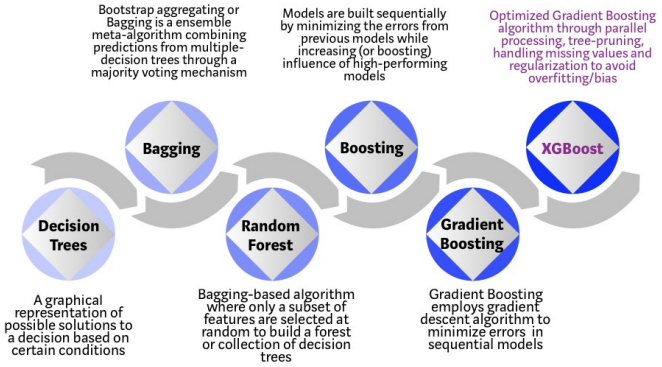


Fig. 6. Evolución del algoritmo XGBoost [13].

- **Ventajas:** Este un modelo que nos funciona bien para temas de clasificación y fundamentalmente nos ayudara para evitar temas de sobre ajuste, permitiendo restringir el crecimiento de los árboles.

– **Desventajas:** Dado que este tipo de modelos es más difícil de afinar que el modelo anterior ("Random Forest"), podemos pensar que entre mejor queramos optimizar temas como; número de árboles, profundidad de árboles y tasa de aprendizaje, tendremos tiempos y costos de entrenamiento más elevados que el modelo anterior.

Máquinas de soporte vectorial. Betancourt [14] explica que una Máquina de Soporte Vectorial (SVM) al ser un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un kernel gaussiano u otro tipo de kernel a un espacio de características en un espacio dimensional más alto, donde se busca la máxima separación entre clases. Esta función de frontera, cuando es traída de regreso al espacio de entrada, puede separar los datos en todas las clases distintas, cada una formando un agrupamiento.

– **Ventajas:** De todos los modelos mencionados anteriormente este modelo tiene una mejor solución debido a que toma ventaja de los siguientes puntos:
  - Utiliza el truco del kernel para resolver soluciones complejas.
  - Utiliza una función de optimización convexa, debido a que los mínimos globales siempre son alcanzables.
  - El concepto de "Hinge loss" proporciona mayor precisión.
  - Los valores atípicos se pueden manejar bien utilizando suavización sobre la constante C.
– **Desventajas:** Si bien es un modelo que nos brinda grandes ventajas, sobre los modelos anteriores, debemos de tomar en consideración los siguientes aspectos:
  - El concepto de "Hinge loss" conduce a la escasez (matriz poblada principalmente con ceros).
  - Los hiperparámetros y los kernel deben ajustarse cuidadosamente para una precisión suficiente.
  - El tiempo y costo computacional de entrenamiento son elevados.

5. Experimentación

En esta sección contrastaremos el desempeño de los modelos:

– **5.1** Regresión Logistica,
– **5.2** Random Forest,
– **5.3** XGBoost,
– **5.4** Máquinas de Soporte Vectorial.

Para generar los modelos realizamos una división del set de datos en el cual 70 porciento de las observaciones se usaron para entrenar el modelo y el 30 porciento restante para probar el modelo, adicional se hizo uso la técnica de

validación por cruce de k-pliegue para mejorar el desempeño de los modelos, el valor de K se definió en 10 (pliegues).

Utilizamos los indicadores de desempeño descritos en la figura 7 para evaluar y contrastar el desempeño entre los modelos y de esta manera definir nuestra recomendación del modelo más óptimo.

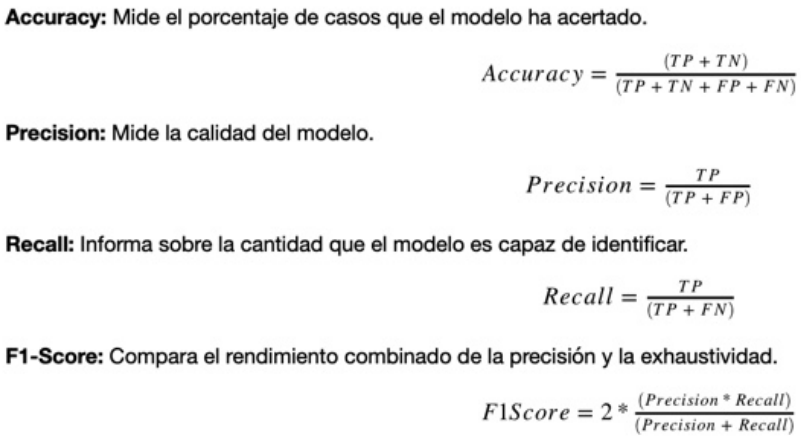


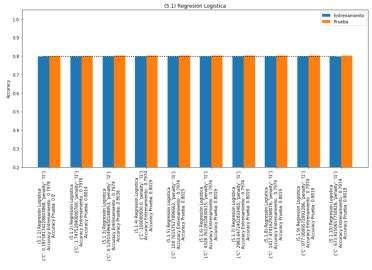
Fig. 7. Interpretación de indicadores.

Se realizó un ranking de desempeño de los hiperparámetros utilizados por modelo con la finalidad de definir los valores óptimos de los hiperparámetros a fijar por modelos, en la figura 8 se muestra un ejemplo de la evaluación de hiperparámetros realizada a la regresión logística de este estudio (las figuras de los modelos restantes se pueden observar en el libro de trabajo que contiene el código que soporta este articulo). Además, en esta misma figura 8 se puede apreciar el listado de hiperparámetros seleccionados a utilizar por cada uno de los modelos.

6. Resultados y discusión

6.1. Resultados

En la figura 9 se observan los resultados obtenidos tras haber evaluado los modelos enunciados en este estudio y el ranking que presentó cada uno de ellos (evaluados con distintos valores en hiperparámetros). Adicional cabe destacar que utilizamos RFE para verificar si el desempeño de los modelos mejoraba, sin embargo no notamos diferencia por lo que la ejecución final no considera RFE (en el libro de trabajo del documento se aprecia el código ejecutado a para esta puntualización).



(a) Ejemplo de comparativa de desempeño de hiperparametro (RL).

Modelo	Hiperparametro
Maquinas de Soporte Vectorial - SVM	{'C': 4, 'gamma': 0.001, 'kernel': 'rbf'}
Regresión Logística	{'C': 0.12915496650148853, 'penalty': 'l2'}
Random Forest	'bootstrap': [True], 'max_depth': [100, 110], 'max_features': ['auto'], 'min_samples_leaf': [3, 4], 'min_samples_split': [8, 10], 'n_estimators': [100, 500, 1000]
XGBoost	'learning_rate': [0.15, 0.20], 'max_depth': [5, 6, 7, 8], 'gamma': [0.4, 0.6], 'colsample_bytree': [0.4], 'n_estimators': [200], 'objective': ['binary:logistic']

(b) Hiperparámetros por modelo.

Fig. 8. Desempeño de hiperparámetros.

Ranking	Modelo	params_cv	Accuracy_Train	Accuracy_test	Recall_test	Precision_test	F1_test
1	(5.4.1) Maquinas de Soporte Vectorial - SVM	{'C': 4, 'gamma': 0.001, 'kernel': 'rbf'}	0.7972	0.8043	0.4957	0.7034	0.5816
2	(5.1.3) Regresión Logistica	{'C': 0.12915496650148853, 'penalty': 'l2'}	0.7974	0.8038	0.5354	0.6813	0.5996
3	(5.4.6) Maquinas de Soporte Vectorial - SVM	{'C': 8, 'gamma': 0.001, 'kernel': 'sigmoid'}	0.7962	0.8033	0.4974	0.6990	0.5812
4	(5.4.3) Maquinas de Soporte Vectorial - SVM	{'C': 7, 'gamma': 0.001, 'kernel': 'sigmoid'}	0.7966	0.8024	0.4853	0.7025	0.5741
5	(5.1.4) Regresión Logistica	{'C': 100000000.0, 'penalty': 'l1'}	0.7974	0.8019	0.5354	0.6754	0.5973
6	(5.1.5) Regresión Logistica	{'C': 104.76157527896662, 'penalty': 'l1'}	0.7974	0.8019	0.5354	0.6754	0.5973
7	(5.2.9) Random Forest	{'bootstrap': True, 'max_depth': 110, 'max_fea...	0.8003	0.8000	0.5095	0.6813	0.5830
8	(5.2.10) Random Forest	{'bootstrap': True, 'max_depth': 100, 'max_fea...	0.8003	0.8000	0.5095	0.6813	0.5830
9	(5.2.1) Random Forest	{'bootstrap': True, 'max_depth': 110, 'max_fea...	0.8009	0.7995	0.5130	0.6781	0.5841
10	(5.3.3) XGBoost	{'colsample_bytree': 0.4, 'gamma': 0.4, 'learn...	0.7926	0.7976	0.5181	0.6696	0.5842
11	(5.3.7) XGBoost	{'colsample_bytree': 0.4, 'gamma': 0.6, 'learn...	0.7899	0.7943	0.5250	0.6566	0.5835
12	(5.3.1) XGBoost	{'colsample_bytree': 0.4, 'gamma': 0.4, 'learn...	0.7936	0.7910	0.5147	0.6507	0.5747

Fig. 9. Ranking de evaluación de performance por modelo.

6.2. Discusión

Con base a la figura 9, podemos argumentar las siguientes observaciones: El mejor indicador de *accuracy* lo tiene el modelo (5.4.1) de Maquinas de Soporte Vectorial - SVM, y en un segundo lugar encontramos que el modelo (5.1.3) de

regresión logística, con base a lo anterior podríamos concluir fría y numéricamente que la opción a utilizar seria la implementación de un modelo de máquina de soporte vectorial SVM, sin embargo si consideramos el uso computacional que conllevó evaluar los modelos, la complejidad de los mismos y la diferencia marginal de resultados (acurracy), nuestra recomendación para este caso de estudio es utilizar el modelo de regresión logística (considerando las limitantes y el uso de hiperparámetros recomendado).

Con base a las condicionantes del set de datos, limitantes del entorno y el tipo de modelos seleccionados al iniciar este estudio es necesario considerar que para determinar un un modelo óptimo no solo se debe contemplar la precisión demostrada por los modelos, dicha decisión debería considerar algunos de los siguientes puntos:

- Tiempo de implementación.
- Infraestructura con la que cuenta la compañía para evaluar un modelo.
- Infraestructura con la que cuenta la compañía para desplegar y operar el modelo.
- Conocimiento del caso de negocio y el objetivo del mismo.
- Conocimiento e interpretación de los datos.
- Entorno.
- Madurez del equipo que desarrolla los modelos.

7. Conclusiones

Para este caso de estudio seguimos los pasos de la metodología CRISPDM para evaluar el abandono de clientes de una empresa de telecomunicaciones, teniendo como objetivo realizar un análisis comparativo entre distintos modelos de aprendizaje máquina que nos permita determinar el modelo más óptimo para el set de datos utilizado. Si bien se siguió la metodología CRISPDM es necesario puntualizar nuevamente que no se logró implementar el paso final de despliegue de los modelos, debido a las limitantes ya descritas en este documento, lo que implica que no tenemos parámetros tangibles para observar el desempeño productivo de dichos modelos y por consiguiente dar el seguimiento correspondiente para realizar los ajustes necesarios a los modelos, set de datos, condiciones de negocio, modificación de hiperparámetros, etc. Los apartados anteriores nos apoyan a soportar que una estrategia de retención de clientes debe considerar diversos elementos como lo son:

- La estrategia y objetivos de negocio.
- Conocimiento de nuestros clientes, claridad en las característcas de clientes objetivo.
- Madurez y estrategia de datos.
- Desarrollo y/o robustecimiento de lineamientos de ética.

Con base a lo anterior podríamos implementar la adecuación de servicios, productos y valor agregado actuales basados en las necesidades del cliente (escuchar

al cliente) o bien la creación de campañas perfiladas de retención de clientes (estas pueden ser digitales o físicas) o la creación de un sistema de calificación del cliente (recompensas), etc.

Referencias

1. Verbeke, W., Martens, D., Mues, C., Baesens, B.: Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Syst Appl 38, pp. 2354–2364 (2011)
2. AbbasKeramati, H., SeyedMohammadMirmohammadi: Developing a prediction model for customer churn from electronic banking services using data mining. (2016)
3. Raj, P.: Telecom Customer Churn Prediction https://www.kaggle.com/pavanraj159/telecom-customer-churn-prediction (2018)
4. Guo-en X., Wei-dong J.: Model of customer churn prediction on support vector machine. Syst Eng Theory Pract, 28(1), pp. 71–77 (2008)
5. Chitarroni, H.: La regresión logística. https://racimo.usal.edu.ar/83/1/Chitarroni (2002)
6. Friedman, J., Hastie, T., Tibshirani, R.: Response to Mease and Wyner, Evidence Contrary to the Statistical View of Boosting, JMLR 9, 131–156, 2008. Journal of Machine Learning Research, 9, pp. 175–180 (2008)
7. Breiman, L.: Bagging Predictors. Machine Learning, 24(2), pp. 123–140 (1996)
8. Ho, T. K.: The random subspace method for constructing decision forests. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(8), pp. 832–844 (1998)
9. Breiman, L.: Randomizing Outputs to Increase Prediction Accuracy. Machine Learning, 40(3), pp. 229–242 (2000)
10. Martínez-Muñoz, G., Suárez, A.: Switching class labels to generate classification ensembles. Pattern Recognition, 38(10), pp.1483–1494 (2005)
11. Dietterich , T. G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine learning, 40(2), pp. 139–157 (2000)
12. Breiman, L.: Random forests. Machine Learning, 45, pp. 5–32 (2001)
13. Vishal Morde: XGBoost Algorithm: Long May She Reign! (2019)
14. Betancourt, G. A.: Las máquinas de soporte vectorial (SVMs). Scientia et technica, 1(27) (2005) 6895/4139
15. Código fuente del estudio realizado[1] (2001)
16. Exploratorio del set de datos [2] (2000)

[1] https://github.com/jorgeo80/UP_MCD/blob/master/MachineLearning/Proyecto/CHURNAnalysis.ipynb,Ingresar->Notebook Python

[2] https://datastudio.google.com/u/0/reporting/1W_R3VelXa8UmWgjxLdlksy7MydMhzD9f/page/YFqGBIngresar->Análisis numero dos realizado con la herramienta google studio

ISSN 1870-4069

Estimación de la frecuencia cardíaca con rPPG: Comparación entre diferentes señales rPPG

Oswaldo Vázquez, Pedro Arguijo, Antonio Hiram Vázquez López, Roberto Ángel Meléndez Armenta

Tecnológico Nacional de México,
Instituto Tecnológico Superior de Misantla,
México

bigosvaap@gmail.com, pedro_arguijo@excite.com,
{ramelendeza, jahvazquezl}@misantla.tecnm.mx.

Resumen. La fotopletismografía remota (rPPG) es una técnica óptica que permite la medición de signos vitales como lo son la frecuencia cardíaca, la respiración y la presión arterial. La medición y monitoreo de estos signos es importante para detectar problemas cardiacos como la hipotensión e hipertensión, ambas relacionadas con la presión arterial o las arritmias en el caso de la frecuencia cardíaca. En este trabajo se estima la frecuencia cardíaca a través de señales rPPG obtenidas usando una cámara web comercial, se realiza una comparación de diferentes formas de obtención de la señal rPPG en diferentes regiones de interés del rostro. Obteniendo como mejor resultados la señal Verde sobre el área total del rostro.

Palabras clave: Fotopletismografía, frecuencia cardíaca, rPPG.

Heart Rate Estimation with Rppg: Comparison between Different Rppg Signals

Abstract. Remote Photoplethysmography (rPPG) is an optical technique that allows the measurement of vital signs such as heart rate, breathing and blood pressure. Measuring and monitoring these signs is important for detecting heart problems such as hypotension and hypertension, both of which are related to blood pressure or arrhythmias in the case of heart rate. In this work the heart rate is estimated through rPPG signals obtained using a commercial webcam, a comparison of different ways of obtaining the rPPG signal in different regions of interest of the face is made. Obtaining as best results the Green signal over the total area of the face.

Keywords: Photoplethysmogram, Heart rate, rPPG.

1. Introducción

La medición de la frecuencia cardíaca por si sola puede ayudar a detectar problemas cardíacos como son la taquicardia y la bradicardia, las cuales se manifiestan por alteraciones del ritmo cardíaco de una persona en estado de reposo. La primera se manifiesta cuando el ritmo cardíaco es superior a 100 pulsaciones por minuto (ppm) mientras que la segunda cuando es inferior a 60 ppm [1,2]. La fotopletismografía (PPG) es una técnica óptica que se utiliza para la detección de la frecuencia cardíaca, respiración y presión sanguínea en diversos dispositivos médicos disponibles comercialmente [3], incluso algunos smartwatchs y teléfonos inteligentes incorporan un sensor para la realización de esta técnica [4]. Aunque es una técnica no invasiva requiere contacto con la piel [5]. Las señales de PPG pueden obtenerse de manera remotas a través de imágenes [6], en este enfoque la persona no necesita tener contacto con el dispositivo. La fotopletismografía remota (rPPG) obtiene la señal, generalmente, de la imagen tomada del rostro de la persona, existen varias aplicaciones asociadas con las mediciones que se pueden llevar a cabo con esta técnica como la detección de emociones [7], la detección de estrés [8], entre otras.

Verkruysse y col., demostraron que la señal PPG se puede obtener de manera remota utilizando luz ambiental [6], previo a este trabajo se empleaba una fuente de luz dedicada.

La mayoría de los trabajos se basan en considerar la intensidad de la imagen [9, 10] con lo cual se supone que la sangre absorbe más luz que los tejidos que la rodean y la variación en el volumen de la sangre afecta la reflexión y transmisión de ésta. Eso lleva a ligeros cambios de color en la piel, los cuales son imperceptibles al ojo humano, pero pueden ser detectados por la lente de la cámara [10], y posteriormente procesados para la obtención de la frecuencia cardíaca, presión sanguínea, entre otros. Generalmente se utilizan cámaras comerciales como cámaras web [11], también se válida la facilidad con la que esta técnica puede implementarse para la obtención de la frecuencia cardíaca de varias personas simultáneamente. Kwon y col., demostraron la factibilidad de utilizar un smartphone para la captura del video y su procesamiento [12].

Diferentes enfoques se han llevado a cabo para la extracción de la frecuencia cardíaca con rPPG, cada uno de estos considera diferentes condiciones ambientales, por lo que es difícil compararlos [10]. En este trabajo estimamos la frecuencia del ritmo cardíaco procesando un solo canal (G, Y, I, V y L) de diversos espacios de color (RGB, YUV, HSI, HSV y HSL), también consideramos los métodos verde-rojo (G-R), CHROM y POS para diferentes regiones de interés. Analizamos un total 10 señales rPPG capturadas con una webcam de voluntarios sanos.

En la sección 2, describimos la metodología empleada. Los resultados y las conclusiones se presentan en las secciones 3 y 4, respectivamente.

2. Metodología

Está sección provee una descripción del algoritmo utilizado para la realización de este trabajo. En la Fig. 1 se muestra el diagrama de la metodología de este proyecto. Para la implementación se utilizó Python 3.7.1 y las librerías de OpenCV y Numpy, para las pruebas se consideraron 10 personas de diferentes edades y tonos de piel.

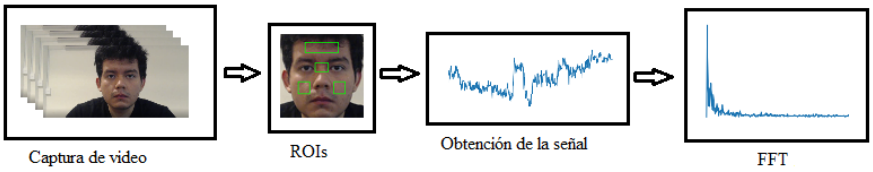


Fig. 1. Metodología utilizada.

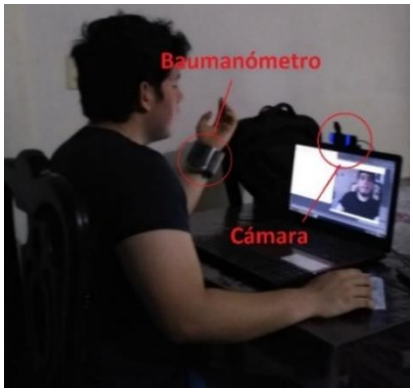


Fig. 2. Captura de video.

2.1. Captura de video

La cámara utilizada es una Logitech C920, durante la grabación del video se tomó la frecuencia cardíaca a un total de 10 participantes sanos a los cuales se les pidió su consentimiento para utilizar sus datos de manera anónima, usando un Baumanómetro digital de muñeca marca Beurer tipo BC 16 de acuerdo con las instrucciones del fabricante [13], las cuales son: no obstrucción de la irrigación sanguínea, mantenerse tranquilo y no hablar, reposo de 5 minutos antes de la medición, el brazalete debe ser sujetado firmemente sin apretar por debajo de 1 cm bajo las eminencias de la pala de la mano y durante la medición se debe doblar y apoyar el brazo teniendo el brazalete a la altura del corazón, como se muestra en la Fig. 2. La duración de cada uno de los videos fue de 30 segundos, como proponen Verkruysse y Poh [6, 11], a una velocidad de 30 fotogramas por segundo con una resolución 1280x720 pixeles grabado a 24-bits en formato MP4. Cada una de las frecuencias cardíacas medidas por el Baumanómetro es la medición testigo y está asociada al correspondiente vídeo de los participantes.

2.2. Regiones de interés

Debido a que del rostro se obtendrán diversas regiones de interés, para la medición de la correspondiente frecuencia cardíaca, éste de detecta con el algoritmo Viola-Jones [14]. Las áreas de interés son: el área del rostro completa, frente, tabique nasal y mejillas, tal como se muestra en la Fig. 3.

Las regiones de interés se calcularon respecto al área del rostro. Para las coordenadas del tabique nasal y las mejillas se consideró lo mencionado por Kopeliovich y Petrushan [15]. El conjunto de coordenada para las regiones de interés son: [x = 0.300, y = 0.100, w = 0.400, h = 0.125] para la frente, mientras que para el tabique nasal [x = 0.420, y = 0.346, w = 0.160, h = 0.104] y las de las mejillas [x = 0.218, y = 0.569, w = 0.138, h = 0.147] y [x = 0.644, y = 0.569, w = 0.138, h = 0.147], para la mejilla derecha e izquierda, respectivamente.

En cada uno de los puntos indicados (x, y) corresponde a la coordenada del vértice superior izquierdo y (w, h) al ancho y altura del área, estos los valores están dados respecto al ancho y alto del área del rostro obtenida por el algoritmo de Viola-Jones. Debido a diferentes constituciones en el rostro de los participantes consideramos que no es posible obtener exactamente la misma ROI, además de considerar el efecto de escalamiento el cual está relacionado con la distancia de la cámara al rostro de la persona. La obtención de todas estas áreas de interés se realizó para cada fotograma del vídeo.

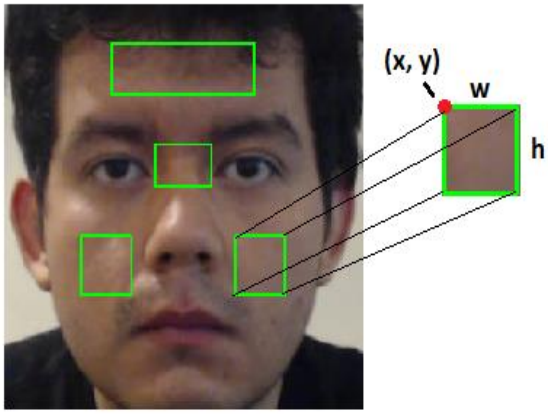


Fig. 3. Regiones de interés, correspondientes a frente, tabique nasal y mejillas.

2.3. Obtención de las señales

Un espacio de color, también conocido como modelo de color (o sistema de color), es un modelo matemático que describe el rango de colores como tuplas de números, típicamente como 3 o 4 valores o componentes de color. Por ejemplo, las imágenes de color típicas se representan como rojo, verde, azul y normalmente se denominan imágenes RGB. Básicamente, el espacio de color es una elaboración del sistema de coordenadas y subespacio.

Cada espacio de color tiene un sistema de coordenadas de color específico y cada punto en el espacio de color representa solo un color específico (ver Fig. 4). Hay una variedad de espacios de color, y cada uno de ellos puede ser útil para aplicaciones específicas, una más amplia información relacionada a los espacios de color se encuentra en Koschan y Abidi [16].

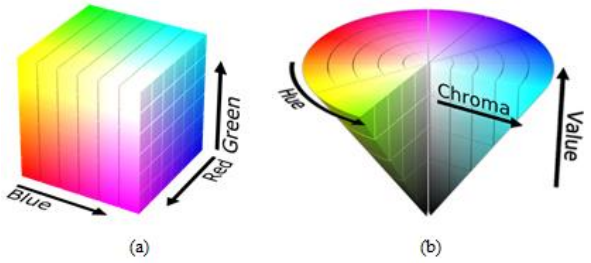


Fig. 4. Espacios de color (a) RGB y (b) HSV. Tomadas de SharkD / CC BY-SA [17].

Para cada fotograma del vídeo se separaron los canales (R, G, B), y se obtuvo el promedio de cada uno, generándose así tres series de tiempo distintas; esto se realizó para cada región de interés indicada en la sección anterior. Cada serie se estandarizó tal como se indica:

$$Z(t) = \frac{X(t) - \mu}{\sigma}, \tag{1}$$

donde *t* indican el tiempo. Las señales rPPG se obtuvieron a partir las series de tiempo estandarizadas. Se consideraron los canales Y, I, V y L de los espacios de color YUV, HSI, HSV y HSL, respectivamente. A continuación, se muestran las expresiones de conversión de RGB a dichos canales:

$$Y = 0.229R + 0.587G + 0.114B\,, \tag{2}$$

$$I = (R + G + B), \tag{3}$$

$$V = \max(R, G, B), \tag{4}$$

$$L = (\max(R, G, B) - \min(R, G, B))/2\,. \tag{5}$$

Aparte de los canales mencionados también se consideró el canal G [18, 19], pues se menciona que de este canal se obtiene la mayor información de las pulsaciones del corazón [6]. Así mismo, se implementó la diferencia entre la señal G y R, éstas señales corresponden a las series de tiempo obtenidas de los canales R y G del espacio de color RGB, debido a que la señal R permite compensar el ruido de ambas señales [20]:

$$p(t) = G(t) - R(t)\,. \tag{6}$$

También se consideró el plano ortogonal al tono de la piel (POS, por sus siglas en inglés) [21]:

$$p(t) = G(t) - B(t) + \frac{\sigma^x}{\sigma^y}[G(t) + B(t) - 2R(t)], \tag{7}$$

donde σ^x, σ^y son las desviaciones estándar de $\sigma^x = G(t) - B(t)$ y $\sigma^x = G(t) + B(t) - 2R(t)$. Igualmente se utilizó un método basado en la señal de crominancia, CHROM [22]:

$$p(t) = 0.77R(t) - 0.51G(t) - \frac{\sigma^x}{\sigma^y}[0.77R(t) + 0.51G(t) - 0.77B(t)]\,, \qquad (8)$$

donde σ^x, σ^y son las desviaciones estándar de $\sigma^x = 0.77R(t) - 0.51G(t)$ y $\sigma^Y = 0.77R(t) + 0.51G(t) - 0.77B(t)$. En la Fig. 5 se muestra las señales rPPG extraídas del área completa del rostro, aunque debe señalarse que la señales no están estandarizadas.

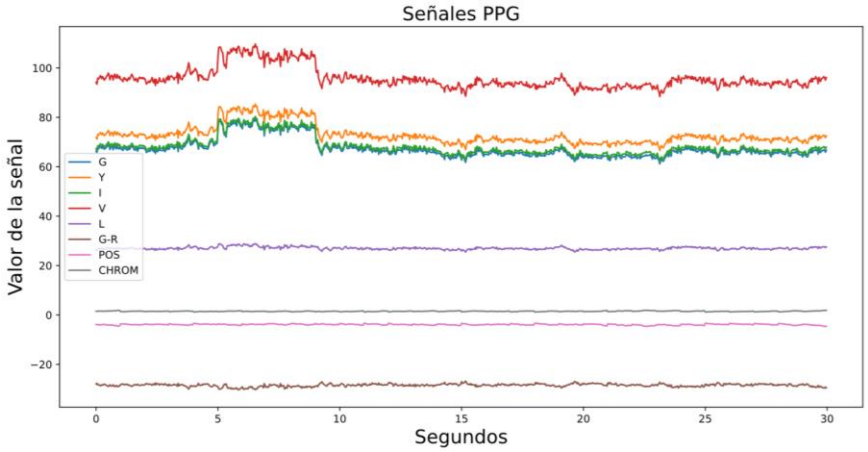


Fig. 5. Ejemplo de señales obtenida.

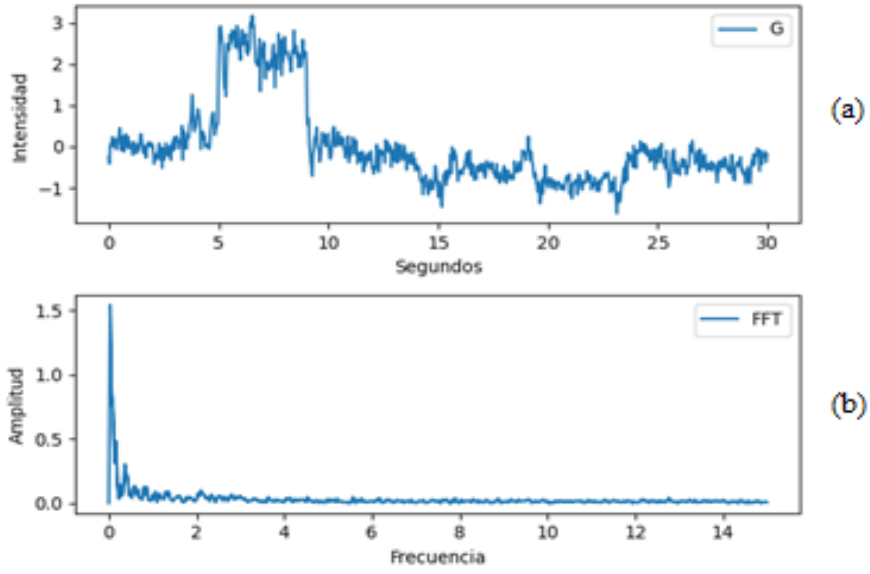


Fig. 6. (a) Señal G del área del rostro completa y su (b) FFT.

Tabla 1. Estimación de la frecuencia cardiaca obtenida del área completa del rostro por diferentes señales.

#	PPM	G	Y	I	V	L	G-R	POS	CHROM
1	80	82	82	60	82	88	60	60	60
2	74	82	82	82	82	78	60	60	60
3	74	74	74	74	74	68	64	74	61
4	69	72	72	72	72	62	60	60	60
5	75	72	72	72	62	74	82	76	66
6	80	80	80	80	80	82	68	76	80
7	82	102	102	102	102	102	108	60	108
8	80	90	62	62	70	72	62	60	76
9	80	82	82	82	82	86	74	60	60
10	86	70	60	60	60	92	74	86	70

2.4. Estimación de la frecuencia cardíaca

La frecuencia cardíaca de los humanos en estado de reposo la cual se encuentra entre 60 a 100 ppm [23], que corresponde a una frecuencia de 1 Hz a 1.6 aproximadamente. Para la estimación de la frecuencia cardíaca a cada señal obtenida (8 por cada una de las 4 región de interés) se le aplicó la FFT obteniendo la frecuencia y amplitud de las señales fundamentales que componen a la señal original, identificándose la señal con mayor amplitud dentro de un rango de frecuencia correspondiente 1 Hz a 2Hz [24]. En la Fig. 6 se muestra un ejemplo de señal G estandarizada correspondiente al área del rostro completa y su respectivo espectro de frecuencias.

3. Resultados

Para todas las señales se estimó la frecuencia cardíaca y fueron agrupadas de acuerdo con la región de interés de la cual fueron tomadas. En la Tabla 1 se muestran frecuencia cardíaca (en ppm) medida y calculada para las diferentes señales mencionadas correspondientes a las diez observaciones, cabe señalar que en estos resultados calculados se consideró el área completa del rostro. La segunda columna corresponde a la frecuencia cardíaca obtenida por el Baumanómetro. Como se puede apreciar existen una serie discrepancia en los resultados de la observación siete, la cual asumimos de debe a cambios bruscos en la iluminación ambiental y este mismo ocurre con las observaciones ocho y diez.

La discrepancia que se observa en las mediciones se analiza a través del error medio absoluto (MAE) y el error cuadrático medio (RECM), las cuales son medidas que permiten comparar series de datos. En nuestro caso, las series a comparar son los datos obtenidos por el Baumanómetro y los datos estimados de las diferentes señales en las diferentes regiones de interés, las cuales están expresadas como:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}, \tag{9}$$

$$RECM = \sqrt{\frac{\sum_{i=1}^{n} (y_i - x_i)^2}{n}}, \tag{10}$$

Tabla 2. Resultados MAE y RECM de la región del rostro.

#	Señal	MAE / ppm	RECM / ppm
1	G	6.4	9.197826
2	Y	8.2	12.20656
3	I	10	13.73317
4	V	8.4	11.94152
5	L	6.8	8.402381
6	G-R	13.4	14.66288
7	POS	11	14.06414
8	CHROM	13	15

Tabla 3. Resultados MAE y RECM de la región de la frente.

#	Señal	MAE / ppm	RECM / ppm
1	G	8.6	10.51665
2	Y	8.6	10.51665
3	I	8.6	10.51665
4	V	8.8	10.59245
5	L	20.2	23.97916
6	G-R	15.6	19.05256
7	POS	9	11.7047
8	CHROM	9	11.23388

Tabla 4. Resultados MAE y RECM de la región del tabique nasal.

#	Señal	MAE / ppm	RECM / ppm
1	G	11.4	12.95376
2	Y	11.4	12.95376
3	I	11.4	12.95376
4	V	9.2	11.80678
5	L	15.2	17.88295
6	G-R	10.6	13.51296
7	POS	10.6	13.21363
8	CHROM	12	15.06652

Tabla 5. Resultados de región de las mejillas.

#	Señal	MAE / ppm	RECM / ppm
1	G	10.4	13.34916
2	Y	11.2	13.70401
3	I	12.4	14.2197
4	V	13.4	14.42914
5	L	11.2	11.89117
6	G-R	14.2	15.43373
7	POS	6.4	9.746794
8	CHROM	8	10.42113

donde y_i son los valores registrados por el Baumanómetro y x_i los valores estimados por la señal, esto se hizo para cada señal en cada una de las regiones de interés. En la Tabla 2 se muestran los resultados del cálculo del MAE y RECM correspondientes a la región del área completa del rostro, mostrando a la señal G con mayor precisión seguida de la señal L.

En la Tabla 3, se muestran los resultados de la región frontal del rostro de las señales. Tanto G, Y e I exhiben el mismo valor MAE y RECM, lo cual nos lleva a considerar necesario aumentar el número de sujetos para comprobar si es algo que se mantiene, y en qué condiciones dichas señales podrían tener ventaja sobre las otras.

En la Tabla 4, muestra los resultados de la región del tabique nasal. Las señales G, Y e I muestran el mismo comportamiento en MAE y RECM que el indicado en la Tabla 3, la similitud con el resultado de ambas tablas podría deberse a que se encuentran en regiones de interés muy cercanas.

En la Tabla 5 correspondiente a la región de las mejillas, la señal con menor error fue POS, seguida de CHROM esto podría sugerir que estas señales son mejores cuando se encuentra un nivel de iluminación desigual, ya que la luz podría incidir más en una mejilla que en la otra.

4. Conclusiones y trabajo a futuro

En este trabajo se calculó la frecuencia cardíaca a través de diversas señales rPPG, el mejor resultado corresponde a la región del área completado del rostro usando el canal G del espacio de color RGB. La discrepancia respecto a entre las regiones de interés puede deberse a que algunas de éstas ofrecen una mejor señal que otras, esto de acuerdo con RECM. De igual manera se asume que la gran discrepancia entre la medida testigo y algunas observaciones se deben a cambios bruscos en la iluminación ambiental, ya que las grabaciones de video no se hicieron bajo una luz dedicada ni en ambientes con luz controlada.

A fin de mejorar los resultados creemos se deben mejorar diversos puntos en la metodología empleada, por ejemplo, en la obtención de las regiones de interés, considerar diversos algoritmos más robustos a la hora de la detección del rostro. Se ha considerado implementar un algoritmo para discriminar los píxeles en los que se encuentra piel de los que no, pudiendo así discriminar barba, bigote, ceja, pestaña o alguna otra oclusión.

Como trabajos futuros se considera la obtención de otras mediciones aparte de la frecuencia cardíaca como lo es la presión sanguínea y la respiración, estos siendo signos

vitales, así como la mejora de la metodología con los puntos mencionados anteriormente y cualquier otro que se encuentre durante la mejora de esta.

Referencias

1. Tachycardia: Fast Heart Rate. https://www.heart.org/en/health-topics/arrhythmia/about-arrhythmia/tachycardia--fast-heart-rate (2020)
2. Arrhythmia: National heart lung and blood institute. https://www.nhlbi.nih.gov/health-topics/arrhythmia (2020)
3. Allen, J.: Photoplethysmography and its application in clinical physiological measurement. Physiological Measurement, 28(3) (2007)
4. Askarian, B., Jung, K., Chong, J.W.: Monitoring of heart rate from photoplethysmographic signals using a Samsung Galaxy Note8 in underwater environments. Sensors, 19(13), pp. 2846 (2019)
5. Ghamari, M.: A review on wearable photoplethysmography sensors and their potential future applications in health care. International Journal of Biosensors & Bioelectronics, 4(4) (2018)
6. Verkruysse, W., Svaasand, L.O., Nelson, J.S.: Remote plethysmographic imaging using ambient light. Optics Express, 16(26), pp. 21434 (2008)
7. Quintana, D.S., Guastella, A.J., Outhred, T., Hickie, I.B., Kemp, A.H.: Heart rate variability is associated with emotion recognition: Direct evidence for a relationship between the autonomic nervous system and social cognition. International Journal of Psychophysiology, 86(2), pp. 168–172 (2012)
8. Vrijkotte, T.G.M., Doornen, L.J.P.V., Geus, E.J.C.D.: Effects of work stress on ambulatory blood pressure, heart rate, and heart rate variability. Hypertension, 35(4), pp. 880–886 (2000)
9. Rouast, P.V., Adam, M.T. P.R., Chiong, D.C., Lux, E.: Remote heart rate measurement using low-cost RGB face video: a technical literature review. Frontiers of Computer Science, 12(5), pp. 858–872 (2018)
10. Wang, C., Pun, T., Chanel, G.: A comparative survey of methods for remote heart rate detection from frontal face videos. Frontiers in Bioengineering and Biotechnology, 6 (2018)
11. Poh, M.Z., Mcduff, D.J., Picard, R.W.: Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. Optics Express, 18(10), pp. 10762 (2010)
12. Kwon, S., Kim, H., Park, K.S.: Validation of heart rate extraction using video imaging on a built-in camera system of a smartphone. Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2012)
13. Handleiding, K.: Manual de uso - Beurer BC 16 tensiómetro, manuall. https://manuall.es/beurer-bc-16-tensiometro/ (2020)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2001)
15. Kopeliovich, M.V., Petrushan, M.V.: Optimal facial areas for webcam-based photoplethysmography. Pattern Recognition and Image Analysis, 26(1), pp. 150–154 (2016)
16. Koschan, A., Abidi, M.A.: Digital color image processing. Hoboken, Wiley-Interscience, (2008)
17. SharkD: CC BY-SA. https://creativecommons.org/licenses/by-sa/4.0 (2012)
18. Li, X., Chen, J., Zhao, G., Pietikainen, M.: Remote heart rate measurement from face videos under realistic situations. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
19. Tasli, H.E., Gudi, A., Uyl, M.D.: Remote PPG based vital sign measurement using adaptive facial regions. In: IEEE International Conference on Image Processing (ICIP) (2014)

20. Unakafov, A.M., Möller, S., Kagan, I., Gail, A., Treue, S., Wolf, F.: Using imaging photoplethysmography for heart rate estimation in non-human primates (2018)
21. Wang, W., Brinker, A.C.D., Stuijk, S., Haan, G.D.: Algorithmic principles of remote PPG. IEEE Transactions on Biomedical Engineering, 64(7), pp. 1479–1491 (2017)
22. Haan, G.D., Jeanne, V.: Robust pulse rate from chrominance-based rPPG. IEEE Transactions on Biomedical Engineering, 60(10), pp. 2878–2886 (2013)
23. Heart: All about heart rate (Pulse) https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/all-about-heart-rate-pulse (2020)
24. Jonathan, E., Leahy, M.J.: Cellular phone-based photoplethysmographic imaging. Journal of Biophotonics, 4(5), pp. 293–296 (2010)

ISSN 1870-4069

Reconocimiento de patrones para la clasificación de componentes argumentales en textos académicos en español

Kenia Nieto-Benitez[1], Noé Alejandro Castro-Sánchez[1], Héctor Jiménez-Salazar[2]

[1] Tecnológico Nacional de México, CENIDET, Cuernavaca, Morelos, México
[2] Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Ciudad de México, México
{kenianieto, ncastro}@cenidet.edu.mx, hgimenezs@gmail.com

Resumen. En este artículo se implementa y experimenta con un método de reconocimiento de patrones para la clasificación de componentes argumentales en español, sobre un corpus de textos académicos. Experimentamos con el uso de características léxicas, estructurales e indicadores junto con una representación basada en tf-idf y binario de estas características. Entre los modelos de aprendizaje automático implementados, se obtuvo un *F-measure* de 0.84 con la Red Neuronal Perceptrón Multicapa con dos clases, premisa y conclusión, y 0.79 con tres clases, premisa, conclusión y no argumental. Ambos resultados al combinar todas las características. La experimentación realizada muestra la importancia y necesidad de determinar características para distinguir segmentos no argumentales, que hasta el momento no se han considerado en la identificación de argumentos.

Palabras clave: Componentes argumentales, reconocimiento de patrones, procesamiento del lenguaje natural, redes neuronales artificiales, minería de argumentos.

Pattern Recognition for Argumentative Components Classification in Academic Spanish Texts

Abstract. This article we implements and experiments with a pattern recognition method for the argumentative components classification in Spanish language, it on an academic texts corpus. We experimented with the use of lexical, structural, and indicator characteristics along with a representation based on tf-idf and binary of these characteristics. Among the machine learning models implemented, an F-measure of 0.84 was obtained with Multilayer Perceptron Neural Network with two classes, premise and conclusion, and 0.79 with three classes, premise, conclusion

and non- argumentative. Both results when combining all the characteristics. The experimentation carried out shows the importance and the need for establish characteristics to distinguish non-argumentative segments, which until now have not been considered in the argument identification.

Keywords: Argumentative components, pattern recognition, natural language processing, artificial neural networks, argument mining.

1. Introducción

Las investigaciones existentes en Minería de Argumentos han implementado modelos que identifiquen de forma automática los componentes y relaciones argumentales en diversos tipos de textos, mediante el diseño e implementación de modelos de aprendizaje automático y procesamiento de lenguaje natural [1]. Sin embargo, se han desarrollado principalmente para el idioma inglés y aplicado en corpus de ensayos persuasivos. Algunas investigaciones han creado nuevos corpus utilizando otras fuentes de información como periódicos en línea, redes sociales, noticias en la Web, entre otras. También han optado por procesar otros corpus que contienen argumentos en diferentes temas, como *Araucaria*, *IBM corpus* o *LiveJournal*.

En el idioma español, hasta ahora se cuenta con un corpus a partir de textos académicos [2]. Sobre este corpus, en su primera versión, se han identificado los componentes argumentales [3] y determinado algunas características para su identificación. Si bien, presenta ciertos tipos de características, sabemos que la forma de representar el texto para la implementación de los modelos puede variar, así como también la selección de características para obtener buenos resultados.

Este trabajo presenta un método de clasificación de componentes argumentales en idioma español, basado en reconocimiento de patrones que incluye dos etapas: extracción de características y reconocimiento de clases. En la primera etapa, experimentamos con características léxicas, estructurales e indicadores. También, exploramos con una representación del texto distinta a [3], basada en valores binarios y tf-idf. En la segunda, la clasificación de componentes se realizó con tres modelos de aprendizaje automático usando un enfoque de clasificación para dos y tres clases.

El artículo está estructurado de la siguiente manera. En la sección 2, presentamos cómo se ha abordado la clasificación de componentes argumentales y los resultados obtenidos. En la sección 3, presentamos el corpus utilizado para los experimentos. En la sección 4, describimos el método implementado basado en reconocimiento de patrones, así como las características y modelos utilizados. En la sección 5, presentados y discutimos los resultados obtenidos en las pruebas con los tres modelos, asimismo la comparación entre los modelos y características. Finalmente, concluimos nuestro trabajo en la sección 6 junto al trabajo futuro.

2. Trabajos relacionados

Los sistemas desarrollados en Minería de Argumentos se han dedicado a implementar modelos que identifiquen de forma automática componentes y relaciones argumentales en diversos tipos de textos [1]. En este trabajo, nos hemos centrado en la identificación de componentes argumentales en textos académicos.

En [4], realizaron la segmentación de textos en unidades de argumento: conclusión, premisa, anécdota y suposición. En el caso de las dos primeras unidades, corresponden a los componentes argumentales. Abordaron esta segmentación como una tarea de clasificación mediante dos modelos de aprendizaje automático y uno de aprendizaje profundo: *Support Vector Machine*, *Conditional Random Fields* y *Bi Long Short-Term Memory* respectivamente, junto con la extracción de características semánticas, sintácticas, estructurales y pragmáticas. Reportaron mejores resultados con la red*Bi Long Short-Term Memory*. Obtuvieron un *F-scores* de 88.54 al utilizar todas las características sobre un conjunto de 402 ensayos académicos [5].

En el caso de [6], identificaron y clasificaron los componentes en tres clases, conclusión principal, conclusión y premisa, empleando el mismo corpus de ensayos que [4], pero en su versión reducida [7] con un total de 90 ensayos. Implementaron un clasificador *Support Vector Machine* con características contextuales, estructurales, léxicas, sintácticas e indicadores. Como resultado obtuvieron un F1 de 0.726.

En [8] también utilizaron el corpus presentado en [5], en este caso, para abordar la identificación automática de componentes y relaciones argumentales juntamente. Emplearon características estructurales, léxicas, sintácticas e indicadores junto a cuatro clasificadores: *Logistic Regression*, *Random Forest*, *Decision Tree* y *Support Vector Machine*. Lograron mejores resultados con *Logistic Regression* al obtener una medida F1 de 0.6081.

En trabajos anteriores, [1] presentó un conjunto de investigaciones que abordaron la identificación de componentes y relaciones argumentales desde 2011 a 2015. En su mayoría, los trabajos citados en [1] se desarrollaron principalmente para el idioma inglés, a excepción de [9] que analizaron y procesaron datos en Alemán y en griego [10,11]. Hasta el momento, se ha encontrado un solo trabajo para la clasificación de componentes argumentales sobre un conjunto de datos en español [3], en el cual se basa el presente artículo.

En [3], clasificaron los componentes en conclusión, premisa y ninguna, con base en la representación del texto *Document Occurrence Representation* (DOR). Utilizaron características léxicas, sintácticas, semánticas e indicadores con modelos de aprendizaje automático como *Support Vector Machine*, *Naive Bayes* y *Random Forest*. Obtuvieron un mejor rendimiento al combinar todas las características sobre el modelo *Support Vector Machine*.

En contraste con el enfoque presentado en [3], en este trabajo, la experimentación se centra en la clasificación de componentes mediante *Support Vector Machine* y dos clasificadores adicionales: *Logistic Regression* y *Multilayer Perceptron*, los tres basados en Reconocimiento de Patrones. En los trabajos mencionados anteriormente utilizaron diversas características, aquí hemos reducido las carac-

terísticas a indicadores, a características léxicas y a características estructurales. De acuerdo con [12] el elemento clave para obtener buenos resultados es la selección de características y no el modelo de clasificación. También, hemos explorado con una representación del texto distinta basada en valores binarios y tf-idf.

3. Datos

Para nuestros experimentos, utilizamos un corpus de 468 tesis y propuestas de investigación compilado por [2]. El corpus contiene anotaciones de componentes argumentales, nivel del argumento, tipo de argumento y tipo de relación entre los componentes. En particular, incluye anotaciones de conclusiones y premisas, las cuales están vinculadas mediante relaciones argumentales de soporte y ataque. El nivel del argumento está relacionado con el número de componentes presentes en cada argumento. Los tipos de argumentos encontrados fueron causal, ejemplos y de autoridad.

De los 468 documentos, seleccionaron 300 secciones pertenecientes al planteamiento del problema, justificación y conclusión. En total, de las 300 secciones obtuvieron 901 párrafos que fueron etiquetados por dos anotadores. Obtuvieron un *inter-annotator agreement* (IAA) de 0.867 en los párrafos pertenecientes al planteamiento del problema; 0.935 pertenecientes a justificación; y 0.866 para conclusión. Posteriormente, seleccionaron solo los párrafos donde los anotadores estuvieron de acuerdo, esto redujo el número de párrafos a 856. En total, el corpus comprende de 856 párrafos (de estos, 584 contienen argumentos y 272 no presentan argumentos) y 2,104 componentes.

<seg type="conclusion"> consideramos que este software será de gran aplicabilidad, </seg>
<seg type="premise"> ya que facilita a los profesores la enseñanza de estos tópicos de manera amena pero con un alto índice de confiabilidad; </seg>
<seg type="premise"> además, sirve de base para la creación de un software hecho en México que se utilice en la creación de modelos tridimensionales de proteínas y su proceso de síntesis, con un amplio contenido técnico que puede ser utilizado tanto en docencia como en la investigación y la industria. </seg>

Fig. 1. Etiquetado de componentes argumentales sobre un texto académico. Obtenido de [2].

De acuerdo con los elementos del corpus, nos centramos únicamente en la identificación de componentes del argumento(ver Fig. 1). Estos se encuentran clasificados en tres clases: conclusión, premisa y ninguno. De los 2,104 componentes identificados, 562 son conclusiones, 1060 premisas y 482 componentes no argumentales.

4. Método

El reconocimiento de patrones se ocupa de la clasificación automática de un objeto dado en un número de categorías o clases. El objetivo del reconocimiento de patrones es el desarrollo de un sistema de clasificación automática que, dándole un objeto desconocido pueda decir a cuál clase pertenece el objeto. Para tomar la decisión analizará las características (variables) de los objetos de las diferentes clases [13].

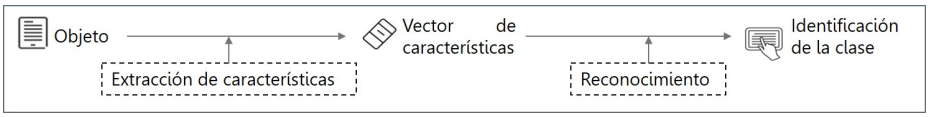


Fig. 2. Método de un sistema de reconocimiento de patrones.

Un sistema de reconocimiento de patrones toma un vector de características como entrada, donde cada objeto es representado por medio de estos vectores obteniendo un patrón P = $\{x, y\}$, donde x es un conjunto de n características $[x_1, x_2, x_3, ..., x_n]$ e y la pertenencia, es decir, a la clase que pertenece el objeto.

Al final, la identificación de la clase se realiza con base en los vectores de características que se procesan mediante modelos de aprendizaje automático (ver Fig. 2). Por tanto, dado un conjunto de datos D = $\{t_1, t_2, t_3, ..., x_m\}$ y un conjunto de clases C = $\{C_1, C_2, C_3, ..., C_o\}$ se emplea una función

$$f : D \rightarrow C, \tag{1}$$

tal que cada t_i sea asignada a una clase C_j. En este trabajo, hemos implementado tres clasificadores para la ecuación (1) y tres tipos de características que nos han permitido encontrar la pertenencia C_j para cada objeto t_i.

4.1. Características

La selección de las características que se utilizaron para representar un objeto fueron con base en [6,3]. Por cada argumento, las características extraídas fueron indicadores, léxicas y estructurales:

- **Indicadores**. Los marcadores del discurso son palabras o conjunto de palabras que permiten identificar los componentes de un argumento y se encuentran divididos en dos categorías: indicadores de premisa e indicadores de conclusión. Por ejemplo, las conclusiones se introducen frecuentemente con 'por lo tanto', 'por ende', 'en consecuencia' o 'por consiguiente', mientras que las premisas tienen marcadores como 'ya que', 'dado que', 'puesto que' o 'viendo que' [3]. Creamos el conjunto de indicadores basados en [14,15] citados por [3] que presentan diferentes marcadores argumentales en español.

- **Léxicas**. Este tipo de característica está basada en los pronombres personales. A menudo, los pronombres personales se introducen con expresiones que hacen referencia a la postura del autor [6]. Seleccionamos los pronombres 'nosotros', 'nosotras', 'nos' y 'se' con el supuesto que este tipo de pronombres son utilizados en el ambiente académico.
- **Estructurales**. Está característica aporta información sobre el tamaño del componente, que de acuerdo con [16] citado por [6], las premisa son más amplias en promedio que las conclusiones. El número de palabras presentes en cada segmento del argumento fue dividido entre 1000 para obtener un valor menor a 1.

Representación de características Al trabajar con texto, las características léxicas e indicadores son categóricas y los algoritmos, dada su forma matemática, no procesan este tipo de características y deben convertirse a números. Uno de los métodos más comunes es la representación basada en los modelos *Bag of Words* o *n-gram* [17]. Si bien, actualmente existen modelos como *word2vec* y *GloVe* que representan las palabras como vectores, decidimos explorar con una representación basada en tf-idf y números binarios.

La primera representación consiste es obtener la relevancia de las palabras mediante la frecuencia del término y la frecuencia inversa del documento con base en

$$tfidf(t,d) = tf(t,d) \times idf(t,d), \tag{2}$$

donde $tf(t,d)$ es la frecuencia del término y se obtiene a partir del número de veces que un término t aparece en un documento d. $idf(t,d)$ es la frecuencia inversa del documento y se calcula del siguiente modo:

$$idf(t,d) = log\frac{n_d}{1+df(d,t)}. \tag{3}$$

En este caso n_d es el número total de documentos y $df(d,t)$ es el número de documentos d que contiene el término t. Para obtener los valores de $tfidf$ utilizamos la clase *TfidfTransformer* de la librería scikit-learn. Esta clase normaliza los valores obtenidos v a partir de (2) aplicando la normalización L2 mediante la ecuación (4) para obtener los valores finales de $tfidf$:

$$v_{norm} = \frac{v}{(\sum_{i=1}^{n} v_i^2)^{\frac{1}{2}}}. \tag{4}$$

Por último, en la segunda representación utilizamos valores binarios para representar las características léxicas, indicadores y estructurales, como una función booleana que indica si un componente argumental contiene un marcador o un pronombre. Para está representación utilizamos el modelo *n-gram* y *Stanford Part-Of-Speech Tagger* como apoyo para obtener los vectores de características.

4.2. Modelos

Para la clasificación de los componentes argumentales implementamos tres modelos. Dos frecuentemente utilizados en el área de Minería de Argumentos:

Máquinas de Vectores de Soporte y Regresión Logística, y la Red Neuronal Perceptrón Multicapa. Si bien, actualmente se utilizan Redes Recurrentes o Convolucionales, ésta red no se ha utilizado para abordar problemas en Minería de Argumentos como la clasificación de componentes argumentales.

Los modelos se implementaron con apoyo de *scikit-learn Toolkit*, conjunto de librerías para aprendizaje automático en *Python*. A continuación presentamos de manera general cómo trabajan estos modelos y cómo se pueden extender, en el caso de Máquinas de Vectores de Soporte y Regresión Logística, para abordar problemas de clasificación multiclase.

Red Neuronal Perceptrón Multicapa Perceptrón Muticapa (*MLP*) es un algoritmo de aprendizaje supervisado para abordar problemas de clasificación no lineal o multiclase. A diferencia del Perceptrón emplea una función

$$f : \mathrm{R}^m \rightarrow \mathrm{R}^o, \tag{5}$$

donde m es el número de dimensiones de entrada y o el número de dimensiones de salida. Además, entre la capa de entra y salida puede haber una o más capas ocultas.

La capa de entrada tiene un conjunto de neuronas, que representa los datos a procesar $\mathrm{X} = \{x_1, x_2, x_3, ..., x_n\}$. Tanto la capa de entrada, oculta y salida tienen un conjunto de coeficientes $W = \{w_1, w_2, w_3, ..., w_m\}$ distintos para cada capa de acuerdo al número de neuronas establecidas por capa.

La clasificación se puede realizar mediante la función logística (sigmoide):

$$\phi(z) = \frac{1}{1 + e^{-z}}, \tag{6}$$

donde z es la suma lineal ponderada de los valores de entrada y coeficientes, como se especifica en (7):

$$z = \sum w_{nm} \times x_{nm} + b. \tag{7}$$

Para una clasificación multiclase, en la capa de salida se implementa una función de activación *Softmax*:

$$\phi(z) = \frac{e^{Z_i}}{\sum_{k=1}^{M} e^{Z_k}}, \tag{8}$$

donde z es la salida de las capas ocultas de acuerdo con (7) y M el número de funciones lineales en el modelo. Entonces, la salida $\hat{y}$ de la red será el $\phi(z)$ con mayor probabilidad.

MLP en la etapa de entrenamiento actualiza los coeficientes w incorporando el descenso del gradiente $\phi(z)(1-\phi(z))$, que permite ajustar y obtener los valores óptimos de w, junto con una tasa de aprendizaje η que controla el tamaño de la actualización de w. Para evitar el sobreajuste del modelo se añade una función de regularización L2: $\|w\|^2$ y una tasa de regularización λ.

Máquinas de vectores de soporte Una máquina de vector soporte (MVS), es un algoritmo de aprendizaje supervisado estadístico para problemas de clasificación lineal de dos grupos de datos como se especifica en (9):

$$f : \mathrm{R}^m \rightarrow \mathrm{R}. \tag{9}$$

Dado un conjunto $\mathrm{S} = \{(x_1, y_1), ..., (x_n, y_n)\}$, donde $x_i \in \mathrm{R}^m$ e $y_i \in \{+1, -1\}$, mediante una función lineal:

$$f(x) = w^T \times x + b = \sum_{i=1}^{m} w_i \times x_i + b, \tag{10}$$

se busca un hiperplano en el espacio de variables que separe las dos clases con un margen de separación óptimo M dado por

$$M = \frac{2}{\|w\|}. \tag{11}$$

El objetivo principal de MVS es maximizar M mediante la minimización de la función objetivo:

$$\begin{array}{c} \text{mín}_{w,b,\zeta} \ \frac{1}{2}\|w\|^2 + \ C \ \sum_{i=1}^{m} \ \zeta_i \\ s.a \quad y_i(w^T x_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0 \end{array}, \tag{12}$$

donde ζ_i son variables de holgura, número de datos no clasificados, y C es un parámetro de regularización para regular el sobreajuste y la proporción del número de datos mal clasificados.

En el caso de un problema no lineal, MVS emplea métodos kernel para proyectar las características hacia un espacio de mayor dimensionalidad, donde este espacio sea un problema lineal. Uno de los kernel más utilizados en la Función de base radial, habitualmente simplificada como:

$$K(x_i, x_j) = \exp\left(-\gamma \ \|x_i, x_j\|^2\right), \gamma > 0, \tag{13}$$

que puede ser interpretada como una función de similitud entre un par de muestras (x_i, x_j) en una matriz K de $n \times n$. El parámetro γ es un parámetro de influencia o alcance de las muestras y juega un papel importante en el sobreajuste del modelo. Por otro lado, MVS puede ampliarse para abordar un problema multiclase mediante las técnicas *OneVsRest* (OVR) o *OneVsOne* (OVO) que permiten construir clasificadores binarios a partir de las clases. OVR transforma un problema de c clases en c problemas binarios. En contraste, OVO construye un clasificador para cada par de clases, transformando un problema de c clases en $c(c-1)/2$ problemas binarios.

Regresión logística (RL) Éste es un modelo lineal para clasificación binaria que también puede ampliarse a multiclase mediante OVR. El objetivo es predecir

la probabilidad de que una determinada muestra x_i pertenezca a una clase concreta $\hat{y}$. Para determinar $\hat{y}$, primero se emplea una función logística (sigmoide) dada en (6) para obtener la probabilidad, a diferencia que z se obtiene como:

$$z = \sum_{i=1}^{m} w_i \times x_i + b \tag{14}$$

Posteriormente, la probabilidad predicha se convierte en un resultado binario mediante una función umbral para determinar $\hat{y}$:

$$\hat{y} = \begin{cases} 1 & \text{si } \phi(z) \geq 0{,}5 \\ 0 & \textit{cualquier otro caso} \end{cases} \tag{15}$$

Por otro parte, así como en otros modelos se tiene el riesgo de un sobreajuste en la etapa de entrenamiento. Para abordar este sobre ajuste, se incorpora una función de regularización L2:$\|w\|^2$ y una tasa de regularización λ en la actualización de los coeficientes w.

5. Experimentación y resultados

La clasificación de componentes argumentales se abordó como una tarea de clasificación binaria/bipolar y multiclase utilizando el corpus detallado en la sección 3. En ambas clasificaciones, se experimentó con dos grupos de características. En el primero, agrupamos las características estructurales e indicadores. En el segundo, consideramos las tres características, léxicas, estructurales e indicadores. Con base en lo anterior, implementamos los modelos SVM, RL y MLP empleando un 70 % de los datos para la etapa de entrenamiento y 30 % en pruebas.

Realizamos cuatro pruebas para cada tipo de clasificación de acuerdo con la representación binaria y tf-idf de las características junto a la combinación de estas características en dos grupos. La Tabla 1 presenta los resultados obtenidos en las dos primeras pruebas pertenecientes a la clasificación binaria/bipolar. Primero, se combinaron las características estructurales e indicadores y se obtuvo un mejor resultado con el modelo MLP junto con la representación binaria. Posteriormente, agregamos la característica léxica como valor discriminante para la clase conclusión. Al agregar esta característica, incrementó el resultado obtenido, a un *F-measure* de 0.84, con respecto a la prueba anterior.

Las dos siguientes pruebas siguieron el mismo enfoque que las anteriores, a diferencia que se introdujo la tercer clase, ninguno. Esta clase agrupa los segmentos no argumentales. Como se observa en la Tabla 2, los resultados obtenidos en cada modelo disminuyeron. Aquí, se obtuvo un mejor resultado, *F-measure* de 0.79, al combinar todas las características con el modelo MLP junto con la representación binaria.

Las premisas del argumento tienden a ser más amplias que la conclusión. Sin embargo, los segmentos no argumentales presentes en el corpus, en su mayoría,

Tabla 1. F-measure para la combinación de características con dos clases.

Grupo de características	SVM	RL	MLP	Representación
Indicadores + Estructurales	0.80	0.79	**0.81**	Binaria
	0.69	0.70	0.69	TFIDF
Indicadores + Estructurales + Léxicas	0.80	0.81	**0.84**	Binaria
	0.76	0.71	0.78	TFIDF

Tabla 2. F-measure para la combinación de características con tres clases.

Grupo de características	SVM	RL	MLP	Representación
Indicadores + Estructurales	**0.67**	0.65	0.65	Binaria
	0.53	0.52	0.53	TFIDF
Indicadores + Estructurales + Léxicas	0.78	0.76	**0.79**	Binaria
	0.61	0.62	0.64	TFIDF

también son muy amplios. Entonces, la disminución del rendimiento de los modelos se debe al tamaño (número de palabras) de los segmentos no argumentales. A esto se añade, la influencia de los marcadores del discurso para distinguir premisas de conclusiones. Algunos segmentos no argumentales tienen marcadores como 'por lo que', 'así que', 'por otra parte' o 'ya que'.

El corpus utilizado en [3] contine menos componentes argumentales en comparación a [2] y no se encuentra disponible al público. Por ello, utilizamos el corpus presentado en [2] para los experimentos. Con este corpus, a deferencia de [3], obtuvimos un *F-measure* de 0.78 con SVM y 0.79 con la red MLP al utilizar una representación binaria. Si bien, hubo una diferencia a favor en los resultados, es necesario considerar el tamaño del corpus aunque se trate del mismo corpus, pero con menos componentes argumentales. Además, observamos que la representación binaria de las características favorece a la clase conclusión representada como 0. Dado el número de ceros que pueden tener los vectores de características con dicha representación.

Los resultados indican la importancia de la selección de características para distinguir y clasificar los componentes argumentales, al igual que en [3]. En este trabajo nos enfocamos en las características estructurales, indicadores y léxicas para distinguir premisas de conclusiones. Sin embargo, dejamos de lado las características para distinguir segmentos no argumentales, así como en los trabajos relacionados. Por lo tanto, consideramos que es conveniente determinar algunas características para distinguir estos segmentos.

6. Conclusiones y trabajo futuro

Presentamos un método basado en el reconocimiento de patrones para la clasificación automática de componentes argumentales en español. A diferencia de trabajos previos que emplearon ensayos, periódicos y discursos en idioma

inglés y noticias así como comentarios en blogs y redes sociales en idioma griego, en este trabajo utilizamos un corpus de 468 tesis y propuestas de investigación en español. Experimentamos con características léxicas, indicadores junto con una representación del texto basada en tf-idf y binaria. Además, utilizamos la Red Neuronal Perceptrón Multica, no empleada en trabajos previos, y dos clasificadores frecuentemente utilizados para abordar problemas existentes en Minería de Argumentos, Máquinas de vectores de soporte y Regresión logística.

La clasificación de los componentes se abordó como una clasificación de dos y tres clases. Con el método presentado, se obtuvieron mejores resultados al combinar todas las características mediante la Red Neuronal Perceptrón Multica con un *F-measure* de 0.84 en la clasificación con dos clases y 0.79 con tres clases. Estos resultados preliminares indican la importancia y necesidad de determinar algunas características para distinguir segmentos no argumentales, que hasta el momento no se han considerado en la identificación de argumentos.

Como trabajo futuro, planeamos experimentar con representaciones de texto actuales como *Word2Vec* y *GloVe*. Ampliar el conjunto de características incorporando características sintácticas y semánticas. Y por último, planeamos implementar otros modelos como Redes Neuronales Recurrentes.

Agradecimientos. Este trabajo fue apoyado parcialmente por el gobierno de México (beca CONACYT, SNI, TecNM 7346.19-P).

Referencias

1. Lippi, M., Torroni, P.: Argumentation mining: State of the art and emerging trends. ACM Transactions on Internet Technology (TOIT) 16(2), pp. 1–25 (2016)
2. Garcia-Gorrostieta, J.M., López-López, A.: A corpus for argument analysis of academic writing: argumentative paragraph detection. Journal of Intelligent and Fuzzy Systems 36(5), pp. 4565–4577 (2019)
3. Garcia-Gorrostieta, J.M., Lopez-Lopez, A.: Argument component classification in academic writings. Journal of Intelligent and Fuzzy Systems 34(5), pp. 3037–3047 (2018)
4. Ajjour, Y., Chen, W.F., Kiesel, J., Wachsmuth, H., Stein, B.: Unit segmentation of argumentative texts. In: Proceedings of the 4th Workshop on Argument Mining, pp. 118–128 (September 2017)
5. Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. Computational Linguistics 43(3), pp. 619–659 (2017)
6. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 46–56 (October 2014)
7. Stab, C., Gurevych, I.: Annotating argument components and relations in persuasive essays. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers. pp. 1501–1510 (August 2014)
8. Du, Y., Li, M., Li, M.: Joint extraction of argument components and relations. In: 2017 International Conference on Asian Language Processing (IALP). pp. 1–4 (December 2017)

9. Eckle-Kohler, J., Kluge, R., Gurevych, I.: On the role of discourse markers for discriminating claims and premises in argumentative discourse. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2236–2242 (September 2015)
10. Sardianos, C., Katakis, I. M., Petasis, G., Karkaletsis, V.: Argument extraction from news. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 56–66 (June 2015)
11. Goudas, T., Louizos, C., Petasis, G., Karkaletsis, V.: Argument Extraction from News, Blogs, and Social Media. In: Likas A., Blekas K., Kalles D. (eds.) Artificial Intelligence: Methods and Applications, LNCS, vol. 8445, pp. 287–29. Springer, Cham (2014). https://doi.org/10,1007/978 − 3 − 319 − 07064 − 3_23
12. Wells, G.: MInería de falacias en el discurso político. Universidad de Barcelona, Barcelona (2018)
13. Theodoridis, S., Koutroumbas, K.: Pattern recognition and neural networks. In: Paliouras G., Karkaletsis V., Spyropoulos C.D. (eds.) Machine Learning and Its Applications, ACAI 1999, vol. 2049, pp. 169–195. Springer, Berlin, Heidelberg (2001). https://doi.org/10,1007/3 − 540 − 44673 − 7_8
14. Fuentes, C.: Diccionario de conectores y operadores del español. Arco/libros, Madrid (2009)
15. Avendaño, C.S.: Los conectores discursivos: su empleo en redacciones de estudiantes universitarios costarricenses. Revista de Filología y Lingüística de la Universidad de Costa Rica 31(2), pp. 169–199 (2005)
16. Biran, O., Rambow, O.:Identifying justifications in written dialogs by classifying text as argumentative. International Journal of Semantic Computing, 5(04), pp. 363–381 (2011)
17. Skansi, S.: Machine Learning Basics. In: Introduction to Deep Learning. Undergraduate Topics in Computer Science. Springer, Cham (2018) https://doi.org/10,1007/978 − 3 − 319 − 73004 − 2

ISSN 1870-4069

Red neuronal Backpropagation para clasificación de pinturas cubistas aplicando segmentación y momentos invariantes de Hu (CIBUrnbp)

Miriam de la Cruz Teomitzi, Ma. Margarita Labastida Roldán

Universidad Autónoma de Tlaxcala,
Facultad de Ciencias Básicas, Ingeniería y Tecnología,
México

dmiriam18@gmail.com,
mariamargarita.labastida.r@uatx.mx

Resumen. En el arte se emplean técnicas de pintura que buscan obtener una composición de formas, colores y texturas. Esta composición es lo que conocemos como las características que pueden determinar el tipo de corriente artística a la que pertenece una obra, sin embargo, estas técnicas pueden dificultar la apreciación de la esencia de las pinturas, es decir, no se logra apreciar con exactitud a qué contexto pertenecen. En el proyecto CIBUrnbp se extraen características de pinturas cubistas aplicando diferentes filtros, entre ellos la segmentación y la extracción de los siete momentos invariantes de Hu para poder realizar una clasificación en tres categorías cotidianas, objetos, paisajes y retratos utilizando una red neuronal Backpropagation.

Palabras clave: Procesamiento de imágenes, segmentación, momentos invariantes, backpropagation, cubismo.

Backpropagation Neural Network for Classification of Cubist Paintings Applying Segmentation and Hu Moments' Invariants (CIBUrnbp)

Abstract. In the art, the usage of techniques that seek obtain a composing of shapes, colors and textures. This composing is known as the characteristics that determine the type of artistic movement a oeuvre to belong. However, these characteristics can hinder assessment of the essence of the painting, i.e., it is not appreciated with accuracy to what context it belongs. The project CIBUrnbp extracts characteristics of cubist paintings applying different filters, between them there are segmentation and extraction of seven invariant moments of Hu to make the classification in three everyday categories: objects, landscapes and portraits, using a neuronal network Backpropagation.

Keywords: Digital image processing, segmentation, invariant moments of Hu, backpropagation, cubism.

1. Introducción

El procesamiento de imágenes es una tarea verdaderamente detallada ya que consiste en extraer características, parámetros o patrones, esto con el fin de clasificar o en todo caso generar nuevas imágenes. Actualmente este procesamiento trae consigo grandes beneficios porque su impacto es de gran utilidad en las áreas como la medicina, el control de procesos industriales, la comunicación, el entretenimiento, la agricultura, la educación, entre otras. Existen precedentes acerca del uso del procesamiento de imágenes en la arqueología, como lo indica un artículo de la Universidad de Murcia [1], donde el principal objetivo era evaluar la utilidad de las técnicas de mejora, realce y restauración de imágenes. Para llevar a cabo esta tarea se eligieron dos objetos, un epígrafe romano y un relieve visigodo; las imágenes elegidas para el proceso tenían defectos, los cuales suelen aparecer ocasionalmente en el material fotográfico de los arqueólogos, dichos defectos no eran más que la fluctuación de la iluminación y circunstancias que dificultaban la visión de los objetos. El resultado de esta investigación fue muy favorable, ya que quedó demostrado que la utilidad del procesamiento digital de imágenes por arqueólogos en la resolución de problemas reales es una forma óptima que se adapta a las necesidades de los trabajos.

Diversos artículos de investigaciones hacen un gran énfasis a la utilidad del procesamiento digital de imágenes y redes neuronales en las áreas antes mencionadas, ya que tiene grandes ventajas en el ámbito industrial y cotidiano, sin embargo, estas áreas de igual manera pueden ser utilizadas para explorar otros ámbitos como puede ser el arte, específicamente el cubismo. El cubismo es una de las corrientes artísticas más importantes del siglo XX, nacido en Francia entre el año 1904 y 1914 y cuyo exponente más conocido es Pablo Picasso. Las pinturas pertenecientes a esta corriente poseen características que pueden permitir la extracción de las mismas, ya que en ellas predominan las líneas rectas, modeladas básicamente por cubos y cilindros.

El objetivo es extraer características de diferentes pinturas cubistas por medio del procesamiento digital de imágenes, específicamente la segmentación y los siete momentos invariantes de Hu, para finalmente clasificar a las pinturas de manera más específica en las categorías: objetos, paisajes y retratos.

2. Trabajos relacionados

En [2] el objetivo principal es desarrollar un software para la interpretación de caracteres. Esta interpretación se aplica al reconocimiento de placas de vehículos y el software desarrollado está compuesto por 3 etapas. La primera consiste en procesar la imagen de partida para resaltar sus características; la segunda etapa realiza medidas sobre la imagen procesada y produce un vector de características, en la etapa final dicho vector se compara con vectores correspondientes a distintas clases. Finalmente se desarrolló un software capaz de interpretar caracteres de placas de vehículos. El principal objetivo de [3] es clasificar automáticamente una serie de objetos que pasan por una banda de producción para posteriormente separarlos y empacarlos. Dicho sistema consta de las siguientes etapas: preprocesamiento, segmentación, extracción de rasgos y finalmente la clasificación.

Los resultados de la implementación son muy favorables, ya que la clasificación fue 100% correcta. La idea principal de [4] es desarrollar aplicaciones para teléfonos móviles que sean capaces de identificar imágenes. A partir de un conjunto de imágenes se extraen las características, se almacenan en una base para que posteriormente un servicio web consuma la información para el reconocimiento y finalmente se utiliza un algoritmo que permite su clasificación definitiva. Los resultados de este proyecto son favorables, ya que el sistema funciona, sin embargo, se menciona que en muchas ocasiones los momentos invariantes de Hu no son suficientes para una clasificación adecuada. En [5] el objetivo principal es reconocer plagas en la planta flor de azúcar ya que esta es atacada por diferentes insectos. Para llevar a cabo la implementación como primera fase se tomaron fotografías en un campo de flores, posteriormente se procesan dichas imágenes aplicando filtros, se hace un conteo de pixeles para analizar la ubicación de los agujeros respecto a la hoja.

Este trabajo de procesamiento de imágenes resulta ser rápido y con ventajas económicas. En [6] se propone la implementación de un algoritmo que combina la umbralización global y adaptativa. Este algoritmo permite la extracción de volúmenes óseos como entidades geométricas únicas de imágenes de tomografía computarizada (TC) para el estudio biomecánico. Se demuestra que la combinación de la umbralización global y la adaptativa constituye un método efectivo. En [7] se desarrollan aplicaciones de detección de bordes por medio de operadores para detectar y reconocer elementos básicos como figuras, líneas y formas. El método de este trabajo consiste en la aplicación de los operadores Roberts, Sobel y Canny, tomando como referencia una imagen de un billete de 100 dólares. Se demuestra que el operador de Roberts presenta un problema y el algoritmo de Canny es más eficiente y preciso.

3. Artefactos propuestos

Se propone el desarrollo del sistema CIBUrnbp para clasificar pinturas cubistas, el cual se detalla a continuación:

3.1. Modelado de contenido

El sistema CIBUrnbp está compuesto por 6 etapas, como se observa en la figura 1. Cada una hace referencia a una fase de todo el proceso de este sistema, la primera consiste en crear una base de datos de 100 pinturas cubistas las cuales son procesadas para finalmente identificar la categoría a la que pertenecen. En la segunda etapa se aplica el método de segmentación por crecimiento de regiones. Durante la tercera se extraen las características de cada pintura basadas en los siete momentos invariantes de Hu, esto para crear la base de datos de dichas características, es decir, la cuarta fase. Posteriormente se procede a entrenar a la red neuronal, quinta fase, para, así como etapa final obtener una clasificación basada en tres categorías, las cuales son: objetos, paisajes y retratos.

La red neuronal tiene como entrada las siete características de las pinturas, las cuales corresponden a los siete momentos invariantes de Hu, una capa interna con 20 neuronas y tres salidas correspondientes a las tres categorías de la clasificación.

Fig. 1. Descripción de la interfaz del sistema CIBUrnbp.

Fig. 2. Imagen en RGB, escala de grises y binaria.

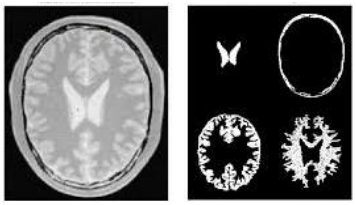


Fig. 3. Segmentación basada en crecimiento de regiones a partir de la rodaja de un cerebro.

Para entrenar a la red se toma la base de datos de las características de las 100 pinturas cubistas y el target, de igual modo se usa la función lineal *logsig*, es decir, *sigmoide* y la función de entrenamiento *trainlm*.

Procesamiento digital de imágenes: como se visualiza en la figura 2, son técnicas de proceso de imágenes que se utilizan para mejorar la apariencia visual de las imágenes. Tiene que ver con la adquisición, transmisión, procesamiento y representación de las imágenes [8].

Segmentación: como se muestra en la figura 3, es el proceso de dividir la imagen en grupos, basado en las similitudes entre los píxeles, cada píxel de la imagen puede asociarse con ciertas propiedades visuales, como el brillo, color y textura [9].

Momentos Invariantes: son un conjunto de momentos a partir de los cuales se pretende reconocer un objeto independientemente de su posición y orientación de la imagen. En la figura 4 se visualizan las fórmulas de dichos momentos [10].

Backpropagation: trabaja bajo aprendizaje supervisado y por lo tanto necesita un conjunto de instrucciones de entrenamiento que le describa cada salida y su valor de salida esperado. En la figura 5 se muestra un diagrama que representa la estructura de la red [11].

Cubismo: movimiento pictórico y escultórico originado por Picasso y Braque reconocido como uno de los grandes puntos de inflexión del arte occidental, nació en París en la primera década del siglo XX. En la figura 6 se visualiza una pintura perteneciente al cubismo [12].

$$H_1 = \bar{\mu}_{20} + \bar{\mu}_{02},$$

$$H_2 = (\bar{\mu}_{20} - \bar{\mu}_{02})^2 + 4\bar{\mu}_{11},$$

$$H_3 = (\bar{\mu}_{30} - 3\bar{\mu}_{12})^2 + (3\bar{\mu}_{21} - \bar{\mu}_{03})^2,$$

$$H_4 = (\bar{\mu}_{30} + \bar{\mu}_{12})^2 + (\bar{\mu}_{21} + \bar{\mu}_{03})^2,$$

$$H_5 = (\bar{\mu}_{30} - 3\bar{\mu}_{12}) \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot [(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - 3(\bar{\mu}_{21} + 3\bar{\mu}_{03})^2] + (3\bar{\mu}_{21} - 3\bar{\mu}_{03}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}) \cdot [3(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + 3\bar{\mu}_{03})^2],$$

$$H_6 = (\bar{\mu}_{20} - \bar{\mu}_{02}) \cdot [(\bar{\mu}_{30} + \bar{\mu}_{12}) - (\bar{\mu}_{21} + \bar{\mu}_{03})^2 + 4\bar{\mu}_{11} \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}),$$

$$H_7 = (3\bar{\mu}_{21} - \bar{\mu}_{03}) \cdot (\bar{\mu}_{30} + \bar{\mu}_{12}) \cdot [(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - 3(\bar{\mu}_{21} + 3\bar{\mu}_{03})^2] + (3\bar{\mu}_{12} - \bar{\mu}_{30}) \cdot (\bar{\mu}_{21} + \bar{\mu}_{03}) \cdot [3(\bar{\mu}_{30} + \bar{\mu}_{12})^2 - (\bar{\mu}_{21} + 3\bar{\mu}_{03})^2].$$

Fig. 4. Siete momentos invariantes de Hu.

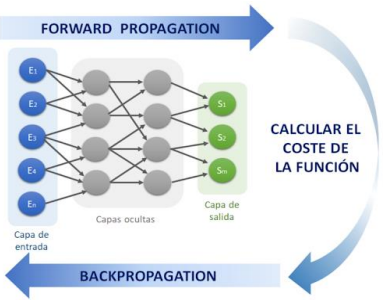


Fig. 5. Estructura de Backpropagation.

Fig. 6. "Las señoritas de Avignon" de Picasso.

A partir de una imagen, se aplican los filtros y algoritmos antes mencionados para obtener los siete momentos invariantes de Hu, mismos que son considerados descriptores los cuales ayudan a la clasificación de las pinturas.

A continuación, se describe el proceso que se sigue en la interfaz cuando se da clic en cada uno de los botones.

- El botón “Abrir imagen” abre un explorador de archivos.
- “Binarización y dilatación” binariza y dilata la imagen elegida y la muestra.
- “Segmentación y erosión” segmenta y erosiona la imagen resultante de la binarización y dilatación.
- “Bordes” detecta los bordes de la imagen previamente segmentada y erosionada.
- “Momentos Hu” obtiene los datos numéricos de los siete descriptores y los muestra en una tabla. Estos datos son la entrada de la red neuronal Backpropagation.
- “Iniciar clasificación” ejecuta la red neuronal y se obtiene el resultado de la clasificación.

En la figura 7. se muestra la interfaz del sistema, en donde se visualizan los resultados de cada una de las fases dentro del procesamiento.

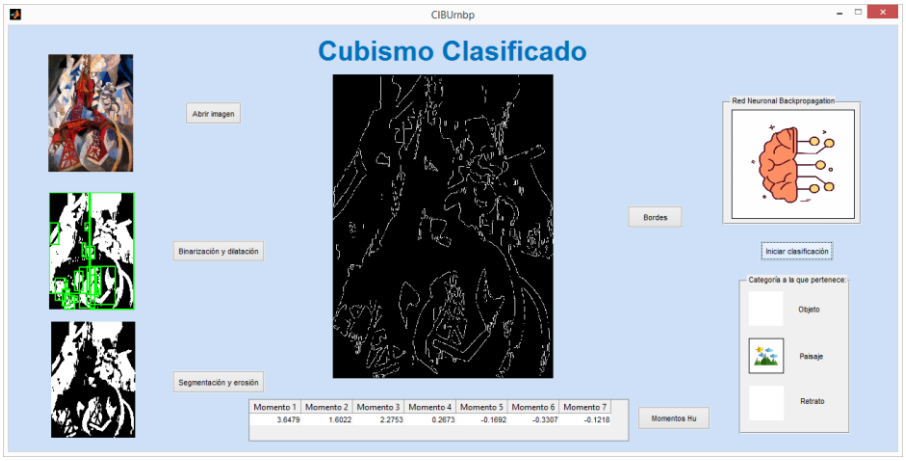


Fig. 7. Interfaz de usuario.

Para evaluar a la red se seleccionaron 20 pinturas, de las cuales 6 fueron objetos, 7 paisajes y por último 7 retratos, en la tabla 1 se muestran las pinturas sometidas a experimentación para evaluar si el resultado de la red fue correcto.

De las 20 pinturas que se sometieron a pruebas 20 casos fueron exitosos, es decir, la red clasificó de manera correcta las 6 pinturas pertenecientes a la categoría objetos con un promedio de 95% de aptitud, de igual modo clasificó correctamente las 7 pinturas de la categoría paisajes con el 95% y por último las 7 pinturas correspondientes a la categoría retratos con un promedio del 92%, dejando como resultado un promedio final del 94% de aptitud.

Tabla 1. Resultados de la clasificación.

Núm.	Pintura	Resultado esperado	% de aptitud	Resultado obtenido
1		Objeto	97%	Objeto
2		Objeto	95%	Objeto
3		Objeto	98%	Objeto
4		Objeto	91%	Objeto
5		Objeto	98%	Objeto
6		Objeto	93%	Objeto
7		Paisaje	98%	Paisaje
8		Paisaje	93%	Paisaje
9		Paisaje	98%	Paisaje
10		Paisaje	92%	Paisaje

Núm.	Pintura	Resultado esperado	% de aptitud	Resultado obtenido
11		Paisaje	100%	Paisaje
12		Paisaje	89%	Paisaje
13		Paisaje	96%	Paisaje
14		Retrato	96%	Retrato
15		Retrato	97%	Retrato
16		Retrato	96%	Retrato
17		Retrato	90%	Retrato
18		Retrato	98%	Retrato
19		Retrato	72%	Retrato
20		Retrato	97%	Retrato

4. Conclusiones y trabajo a futuro

La implementación fue un verdadero reto en cada una de sus etapas, como en la creación de la base de datos de las pinturas, ya que encontrar pinturas digitales no fue sencillo, por otro lado, el crear la estructura de la red tomó un tiempo considerable hasta encontrar la arquitectura más adecuada y con mejores resultados, sin embargo, los resultados finales de este proyecto son completamente aceptables porque se obtuvo en su totalidad el resultado esperado. De igual manera se hizo el hallazgo de la mejor arquitectura de la red neuronal, misma que consiste en el empleo de la función de activación *sigmoide*, *logsig*, que es propia de Backpropagation y de igual modo la función de entrenamiento, *trainlm*, la cual es la más rápida y recomendada para entrenar redes de este tipo, ya que se trata de aprendizaje supervisado. Con el uso de estas dos funciones se obtiene un porcentaje alto de exactitud en la clasificación partiendo del entrenamiento.

Por otra parte, el desarrollo de este trabajo beneficia en gran parte al arte y a las personas que tienen interés en ella, ya que, al realizar una clasificación de pinturas cubistas basada en un contexto cotidiano, se tiene una mejor apreciación artística y se descarta la confusión y desinterés, ya que, al no tener el recurso de la familiaridad o la información previa, es difícil saber qué es y qué significa una obra.

Como trabajo a futuro se puede contar con una base de datos con muchas más pinturas cubistas para evaluar la clasificación, del mismo modo se puede probar la implementación de estos algoritmos para clasificar pinturas de otras corrientes artísticas, como expresionismo y surrealismo, e incluso se pueden clasificar fotografías reales dentro de una galería, por último, se propone realizar una clasificación de pinturas a partir de una aplicación móvil.

Referencias

1. García, I., Storch, J.J., Cisneros, G., Vicente, A.: El proceso digital de imagen en la investigación arqueológica. Universidad de Murcia. https://dialnet.unirioja.es/descarga/articulo/112526.pdf (2019)
2. Rolón, R., Domínguez, V.: Reconocimiento de caracteres en imagen digital, una aplicación al reconocimiento automático de placas de vehículos. Agosto 27, 2019, de Facultad Politécnica, Universidad Nacional del Este. www.une.edu.py/index/php/fpune scientific/article/download (2015)
3. Gómez, W.: Reconocimiento de objetos en fotografías. CINVESTAV-LTI. https://www.tamps.cinvestav.mx/ wgomez/toptamps/tutorial.pdf (2019)
4. García, P.: Reconocimiento de imágenes utilizando redes neuronales artificiales. Universidad Complutense de Madrid. https://eprints.ucm.es/23444/1/ProyectoFinMaster PedroPablo.pdf (2013)
5. Cáceres, C.A., Amaya, D., Ramos, O.L.: Procesamiento de imágenes para reconocimiento de daños causados por plagas en el cultivo de Begonia semperflorens (flor de azúcar). Universidad Nacional de Colombia. Recuperado de: http://www.redalyc.org /articulo.oa?id=169940048012 (2015)
6. Ortega, D.R., Gutiérrez, G., Iznaga, A.M., Rodríguez, T., de Beule, M., Verhegghe, B.: Segmentación de los huesos en imágenes TC empleando la umbralización global y adaptativa. Instituto Superior Politécnico José Antonio Echeverría - Instituto de Tecnología

Biomédica, Universidad de Ghent. https://www.elsevier.es/es-revista-imagendiagnostica-308-pdf-S2171366914000043 (2014)
7. Sánchez, V.M., Caballero, F.G., Pérez, M.Á., Vidal, M., Rodríguez, E.: Detección de bordes de una imagen usando Matlab. Tecnológico Nacional de México/Instituto Tecnológico de Celaya. Recuperado de: http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/download/655/ (2019)
8. González, D.: 2. Procesamiento de imágenes. Universidad de Salamanca Master de Geotecnologías Cartográficas en Ingeniería y Arquitectura. http://ocw.usal.es/eduCommons/ensenanzas-tecnicas/procesamiento-avanzado-de-imagenes-digitales/contenidos/Tema2.pdf (2019)
9. McAndrew, A.: Introduction to Digital Image Processing with Matlab (2004)
10. Cuevas, E., Zaldivar, D., Pérez, M.: Procesamiento digital de imágenes con MATLAB y Simulink, México (2010)
11. Ponce, P.: Inteligencia artificial con aplicaciones a la ingeniería (2010)
12. Lozano, J.M.: Historia del arte (2013)

ISSN 1870-4069

Reconocimiento de caracteres manuscritos a partir de señales EMG usando redes neuronales convolucionales y recurrentes de Elman

José Guadalupe Beltrán-Hernández, José Ruiz-Pinales,
Pedro López-Rodríguez, José Luis López-Ramírez

Universidad de Guanajuato, División de Ingenierías,
Departamento de Ingeniería Electrónica,
México

{jg.beltranhernandez, pinales, p.lopez.rodriguez,
jl.lopezramirez}@ugto.mx

Resumen. Hoy en día la tecnología está presente en todos los ámbitos de la vida diaria, y la escritura es sin duda un medio muy importante de comunicación. Además, la escritura y la lectura juegan un papel importante en el desarrollo cognitivo de los niños, tanto así que éstas determinan su posterior éxito académico. Este documento presenta una nueva metodología para el reconocimiento de caracteres manuscritos. La metodología propuesta consiste en el uso de un brazalete Myo para capturar las señales EMG que se generan durante el proceso de escritura y el uso de redes neuronales convolucionales y redes recurrentes para el procesamiento y reconocimiento de caracteres. Los resultados usando una base de datos propietaria grande muestran un porcentaje de aciertos promedio de 94.92 %.

Palabras clave: Señales electromiográficas, aprendizaje profundo, redes neuronales convolucionales, redes de Elman.

Handwritten Character Recognition from EMG Signals using Convolutional and Elman Neural Networks

Abstract. Nowadays, technology is present in almost every aspect of our lives. Reading and writing continue to be an important means of communication. Besides that, they play en important role in the cognitive development of children and later on determine their academic success. This paper presents a new methodology for the recognition of handwriting. The poposed methodology consists in using a Myo armband to capture the EMG signals generated during the writing process and the use of Convolutional Neural Networks and Elman Networks for processing and recognizing characters. The results using a large propietary database show an average accuracy of 94.92 %.

Keywords: Electromyographic signals, deep learning, convolutional neural networks, Elman networks.

1. Introducción

La escritura y la lectura, son tareas complejas y habilidades importantes para los seres humanos. De hecho, existen estudios que han mostrado la importancia de tales habilidades para el desarrollo cognitivo y el éxito académico de los niños [15]. Sin embargo, resulta que la escritura manuscrita a partir de señales bioeléctricas ha sido poco abordada en el área de las interfaces humano-computadora. Estas interfaces son importantes porque ofrecen una alternativa en aplicaciones clínicas y de asistencia para las personas con o sin discapacidades.

Ya se han logrado desarrollar algunas prótesis neurales capaces de hacer los mismos movimientos que un brazo o una pierna humana. Incluso, ya existen interfaces para juguetes y videojuegos que permiten el control por medio de las señales bioeléctricas de los usuarios. En particular, existen dispositivos comerciales tales como el brazalete Myo que puede usarse para controlar presentaciones, vehículos, etc.

Las señales bioeléctricas tales como las señales EMG (electromiográficas), han sido aplicadas a la detección de trastornos neurológicos [10]. Por ejemplo, el análisis de patrones de escritura se ha utilizado para el diagnóstico de la enfermedad de Parkinson y la disgrafía en niños.

En la literatura hay un cierto número de publicaciones que tratan con las señales EMG generadas por los músculos de las manos y los antebrazos durante la actividad de la escritura [14,6]. En general, el reconocimiento de la escritura a partir de las señales EMG se han abordado utilizando diferentes enfoques: concordancia de plantillas [14], Dynamic Time Warping (DTW) [6,13], Redes Neuronales RBF [10]. Igualmente, la reconstrucción de la escritura a partir de las señales EMG ha sido abordada usando: filtros de Wiener [14], mínimos cuadrados recursivos [8], Redes Neuronales RBF [18], y filtros de Kalman [16].

Las señales EMG son bioseñales complejas, cuya forma se ve afectada por muchos factores externos así como por propiedades de los músculos, la instrumentación que se utiliza para detectarlas, etc. La complejidad puede explicarse por el simple hecho de que hay muchas maneras de realizar una tarea motora (fenómeno de redundancia motora) [16,4,1].

Siempre se ha sabido que las señales EMG contienen una gran cantidad de información sobre los movimientos de las extremidades del cuerpo humano. Aprovechando esta información, varias aplicaciones en la literatura han efectuado reconstrucción de movimientos de la escritura, modelado, simulación de movimientos, reconocimiento de movimientos. Con respecto al reconocimiento de caracteres existen muy pocas bases de datos. Sin embargo, muchos de los estudios imponen restricciones de diferentes tipos: colocación de los electrodos sobre músculos del antebrazo, número y complejidad de los trazos manuscritos.

En nuestro caso, proponemos abordar el problema del reconocimiento de caracteres manuscritos desde un punto de vista más general. En lugar de imponer restricciones, se utiliza una pulsera Myo en el antebrazo y el usuario puede usar su estilo de la escritura. Para contrarrestar las variaciones de las señales EMG, se propone el uso de redes neuronales convolucionales (CNN), para el procesamiento y extracción de características. Esto es debido a que se ha demostrado que las CNN son excelentes para extraer características directamente de los datos en bruto [3,12]. Para el procesamiento de las secuencias se propone el uso de redes recurrentes (RNN) con funciones de activación ReLU. Estas redes son menos pesadas computacionalmente y tienen igual o mejor desempeño que otras redes más recientes tales como las LSTM.

En la siguiente sección se presenta una descripción de los materiales y métodos empleados durante este estudio. Finalmente, se presentan los resultados y conclusiones.

2. Métodos y materiales

2.1. Materiales

A fin de validar nuestra metodología, se escogió una gran base de datos que contiene señales EMG. Las señales EMG fueron capturadas por medio de un brazalete Myo colocado en el antebrazo derecho de tres personas. El brazalete Myo es un dispositivo comercial inalámbrico de bajo costo, no invasivo, de 8 canales, y tiene una frecuencia de muestreo de 200 Hz. La Fig. 1 muestra cómo se usa el dispositivo para capturar las señales EMG generadas al momento de escribir con el dedo sobre una pantalla.

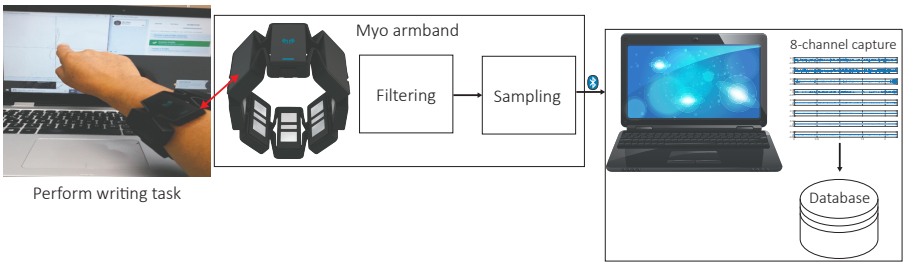


Fig. 1. Diagrama del sistema de captura de las señales EMG.

Por lo tanto, contiene 3,000 capturas por clase. Cada captura comprende 8 señales correspondientes a los 8 sensores del dispositivo. La Fig. 2 muestra una captura para la letra 'b'. En este caso particular, se puede observar más actividad en los 4 primeros canales. En el caso general, el brazalete puede colocarse en cualquier posición, y los canales activos suelen ser diferentes.

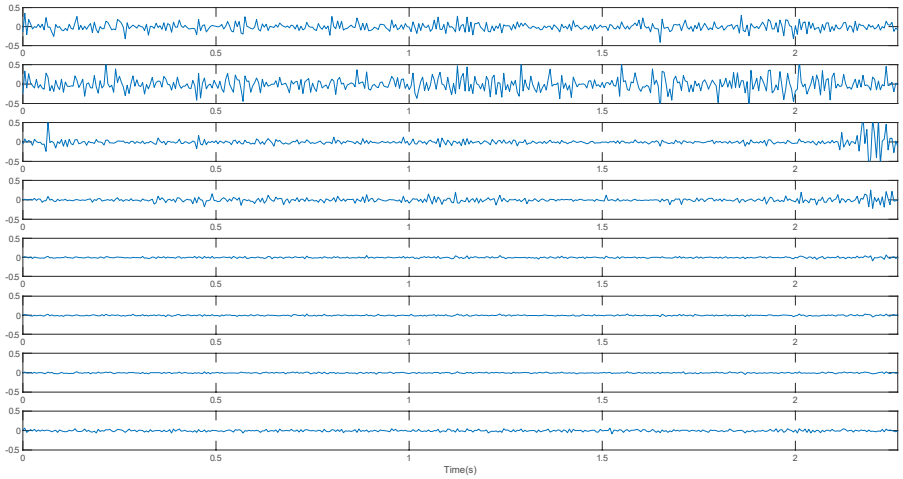


Fig. 2. Ejemplo de una captura correspondiente a la letra 'b'. La parte superior corresponde al canal 0 y la parte inferior al canal 7.

2.2. Métodos

Redes Neuronales Convolucionales Una red neuronal convolucional (CNN), es una red neuronal estática inspirada originalmente de hallazgos en neurociencias. En 1959, Hubel & Wiesel [7], propusieron un modelo jerárquico de red para explicar el comportamiento de las células simples del gato. A partir de ese modelo se propusieron varios otros modelos como el Neocognitrón [2] que quizás constituye una de las primeras CNN, ya que usaba capas convolucionales y capas de submuestreo. Otro modelo, el Cresceptrón [21], utilizó por primera vez el mecanismo de max-pooling.

El modelo LeNet fue el primero en utilizar el algoritmo de retropropagación [12]. Este modelo demostró que una CNN puede procesar la imagen directamente sin ningún o poco preprocesamiento. Lo que abrió el camino para el desarrollo de las arquitecturas actuales de Deep Learning. Aunque el modelo LeNet-5 tuvo gran éxito en el reconocimiento de dígitos, su uso para tareas más complejas estaba muy limitado debido, por un lado, al poco poder de cómputo disponible en su época, y, por otro lado, a algunos problemas que presentaban las CNNs de mayor profundidad (DCNN) [3].

Después de varios años, las CNN resurgieron exitosamente con el modelo AlexNet al ganar por primera vez un concurso internacional (ILVRC2012). El modelo AlexNet [9], tenía una arquitectura más profunda que el modelo LeNet-5, con capas más grandes y nuevos avances como: función de activación ReLU, métodos de inicialización adHoc, mecanismo de regularización llamado Dropout, y mecanismo max-pooling.

El modelo ResNet [5], quien ganó algunos años después el mismo concurso (ILSVRC2015), era mucho más profunda que el modelo AlexNet, e incorporó innovaciones como el bloque residual, y capas batch-normalization.

Redes Neuronales Recurrentes Las redes Neuronales recurrentes (RNN) son un tipo de red neuronal dinámica capaz de procesar secuencias [17]. La redes de Elman y Jordan constituyen algunas de las primeras redes recurrentes. Su característica principal es que tienen la capacidad de memorizar patrones por medio de la retroalimentación de las neuronas. Su entrenamiento utiliza un algoritmo llamado retropropagación en el tiempo (BPTT). En realidad, la red recurrente es desenvuelta en el tiempo dando lugar a una variante de red estática con pesos compartidos que se entrena usando el algoritmo de retropropagación. Estas primeras redes recurrentes tuvieron un éxito limitado debido a que presentaban problemas para aprender secuencias largas (desvanecimiento del gradiente). Se podría decir que estas redes sólo eran capaces de memoria de corto plazo. La Fig. 3 muestra una RNN y su desenvolvimiento en el tiempo.

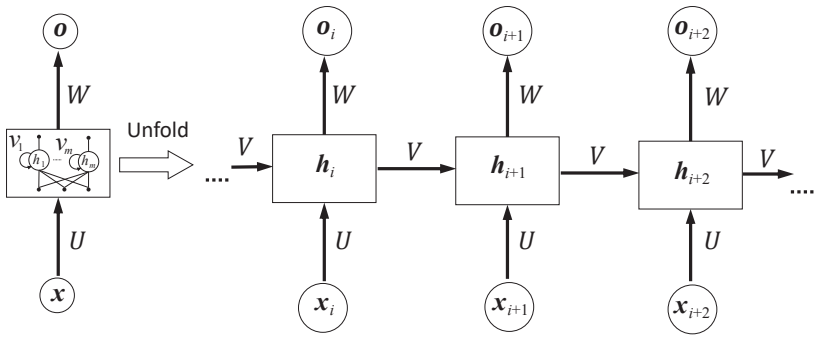


Fig. 3. Arquitectura de una RNN y su desenvolvimiento en el tiempo.

Las redes recurrentes LSTM (Long Sort-Term Memory) fueron las primeras en incorporar la capacidad de memoria de largo plazo ya que no presentaban más el problema del desvanecimiento del gradiente. Una red LSTM está compuesta de unidades de memoria, con la capacidad de decidir cuándo recordar, olvidar y adquirir información. La operación de una unidad LSTM está controlada por tres compuertas multiplicadoras: una para olvidar, otra para adquirir información y otra para dar salida al estado de la celda de memoria. El problema de las redes LSTM es que contienen mucho más parámetros que sus predecesoras.

3. Métodología propuesta

En general, el reconocimiento de caracteres a partir de señales EMG se ha basado en el pre-procesamiento, extracción de características, y clasificación [14].

Sin embargo, existen métodos que se basan en el preprocesamiento, medida de similitud usando DTW y clasificación usando distancia Euclidiana. Por otro lado, la mayoría de los estudios anteriores han sido realizados bajo condiciones controladas de tal manera que se facilita la clasificación de las señales EMG. Por ejemplo, colocar cuidadosamente cada electrodo sobre cada músculo de la mano o antebrazo. Otras restricciones son que cada letra conste de un solo trazo y que las letras sean de la misma forma.

En este trabajo, se propone una metodología más general para el reconocimiento de caracteres a partir de señales EMG. En particular, la escritura es multi-usuario, sin restricciones en la forma del caracter y el número de trazos. Además, en lugar de colocar electrodos sobre el brazo de los sujetos, se utiliza un brazalete Myo en el antebrazo. Para hacer frente a la problemática de la variabilidad de las señales EMG, se usan capas convolucionales y max-pooling para la extracción de características, IRNN para el procesamiento de la secuencia y una capa totalmente conectada para la clasificación. La razón por la que se escogió una RNN es que son menos pesadas computacionalmente que otras redes más recientes. Además, el problema del desvanecimiento gradiente se debe al uso sigmoides como función de activación. Por lo tanto, las RNN que utilizan ReLU como función de activación, llamadas IRNN, no presentan el problema del desvanecimiento del gradiente [11]. Se ha observado que en muchos casos hasta han llegado a superar a otras redes tales como las LSTM [11,20].

Sin embargo, las IRNN aún pueden presentar problemas de convergencia debido al uso de ReLU. Afortunadamente, eso se puede resolver fácilmente inicializando los pesos recurrentes a 1 y los demás pesos a valores pequeños generados de una distribución normal [11].

La Fig. 4 muestra la arquitectura de la red propuesta. Se utiliza la función de activación ReLU en todas las capas convolucionales. Para combatir el sobreaprendizaje y mejorar la generalización de la red, se aplicó la técnica de Dropout en cada capa convolucional. La capa totalmente conectada utiliza la función de activación SoftMax para que las 36 salidas puedan corresponder a probabilidades. La longitud máxima de secuencia se puso en $L = 900$.

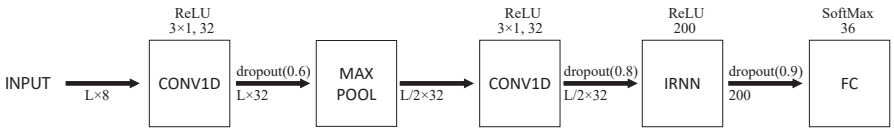


Fig. 4. Arquitectura de la red propuesta. La longitud de la secuencia de entrada está indicada con L.

4. Implementación

Para estimar el desempeño del sistema se utilizó validación cruzada estratificada k-fold. Por lo que la base de datos se dividió al azar en 10 partes disjuntas

de igual tamaño. Es importante mencionar que se tuvo cuidado de preservar la proporciones de cada clase en cada parte. Cada parte se tomó como conjunto de prueba y el resto se utilizó para entrenar la red neuronal. El desempeño de la red se calcula como el promedio de los resultados de las 10 pruebas.

La implementación se efectuó bajo Python 3.6 utilizando el paquete Keras con Tensorflow. El entrenamiento se efectuó en una estación de trabajo con Windows 10, procesador INTEL Core i7, 16 GB de RAM, y GPU nVidia RTX 2080. Para entrenar la red, se utilizó el algoritmo de momento adaptivo acelerado con Nesterov. A fin de acelerar aún más el entrenamiento y además evitar mínimos locales, se utilizó un algoritmo de tasa de aprendizaje cíclico [19] (tasas mínima y máxima de 0.000001 y 0.0001, respectivamente, tamaño de paso igual a 285 o tres veces el número de iteraciones por época). El tamaño de batch se fijó en 1024. El entrenamiento se repitió varias veces con diferentes condiciones iniciales. Al principio se observó que la red presentaba algo de sobre-entrenamiento. Por lo tanto, se optó por regularizar la red usando dropout en las dos capas convolucionales.

5. Resultados

Después de entrenar la red durante 3,500 épocas se encontró que la tasa de aciertos promedio en el conjunto prueba es de 94.92 %.

La tabla 1 muestra una comparación con el estado del arte. A partir de esto, podemos concluir que, en general, la metodología propuesta se compara bien con otros trabajos del estado del arte. Es de resaltar que todos los trabajos anteriores utilizaron bases de datos pequeñas con menos clases y menos variabilidad en las señales EMG.

Tabla 1. Comparación con otros trabajos del estado del arte.

Trabajo	Muestras	Clases	%
Template Matching [14]	350	10	97
DTW [6]	520	26	84.29
DTW [13]	780	26	92.42
Propuesto	108,000	36	94.92

La Fig. 5 muestra la matriz de confusión obtenida para la arquitectura propuesta en el conjunto de prueba. Se puede observar que, en general, todas las clases son bien reconocidas. Sin embargo, hay ciertas clases que presentan más errores, por ejemplo, '0', '3', 'v' y 'x' son confundidas la mayoría de las veces con '6', '5', 'u' y 'y', respectivamente.

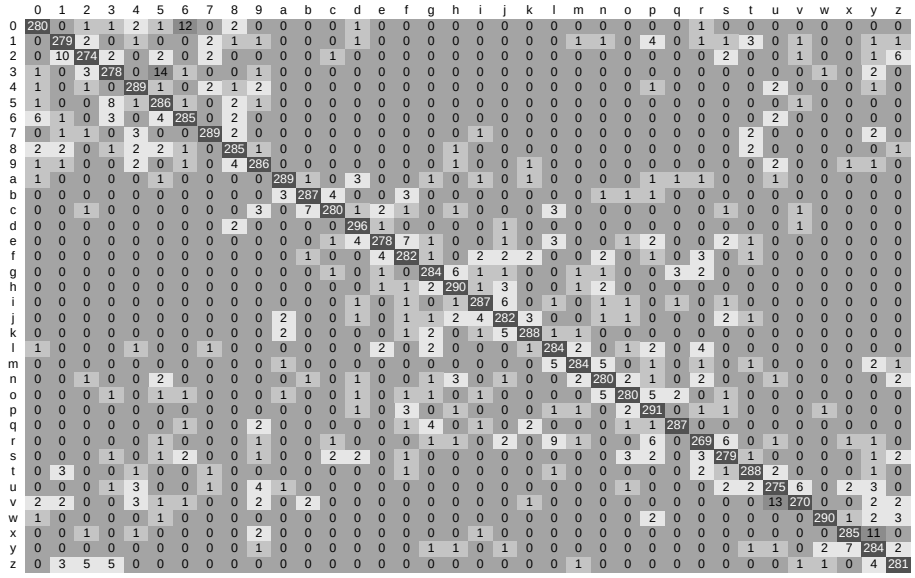


Fig. 5. Matriz de confusión obtenida para la arquitectura propuesta.

6. Conclusiones

Se propuso una nueva metodología para el reconocimiento de caracteres manuscritos a partir de señales EMG. Esta consiste en utilizar capas convolucionales, y max-pooling para la extracción de características, IRNN para el procesamiento de secuencias y totalmente conectada para la clasificación.

Una ventaja de la propuesta es que se pueden escribir caracteres con cualquier instrumento de escritura o simplemente con el dedo. Otra de las ventajas es que los caracteres también pueden ser trazados en el aire. En ese caso, no es necesario tener un soporte plano para la escritura. Una ventaja adicional es que se usan redes recurrentes menos pesadas computacionalmente que otras redes tales como las LSTM.

Se encontró que la tasa de aciertos promedio de la arquitectura propuesta es de 94.92 %. Los resultados obtenidos son comparables a otros resultados del estado del arte. Es importante recalcar que los trabajos anteriores utilizaron bases de datos pequeñas con menos variaciones y menos clases. En nuestro caso, las señales EMG presentan variaciones complejas debidas a la colocación del brazalete Myo y al estilo de escritura de cada sujeto. Además, los resultados comprueban que las IRNN presentan un excelente desempeño para la tarea de reconocimiento de caracteres manuscritos y no presentan ningún problema debido al desvanecimiento del gradiente.

Este trabajo es importante para el área de las interfaces humano-computadora la cuales ofrecen una alternativa para restaurar o mejorar las habilidades de comunicación de las personas con y sin discapacidad.

Trabajos a futuro podrían abordar el problema del reconocimiento de palabras manuscritas a partir de señales EMG. Igualmente, aún cuando las IRNN no son muy pesadas computacionalmente, se considera que es necesario estudiar su implementación en tiempo real.

Agradecimientos. Los autores agradecen a la Universidad de Guanajuato, en particular a la División de Ingenierías Campus Irapuato-Salamanca por el apoyo brindado.

Referencias

1. De Luca, C.J.: Physiology and Mathematics of Myoelectric Signals. IEEE Transactions on Biomedical Engineering BME-26(6), 313–325 (jun 1979)
2. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 193–202 (apr 1980)
3. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition 77, 354–377 (may 2018)
4. Guigon, E., Baraduc, P., Desmurget, M.: Computational Motor Control: Redundancy and Invariance. Journal of Neurophysiology 97(1), 331–347 (jan 2007)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. IEEE (jun 2016)
6. Huang, G., Zhang, D., Zheng, X., Zhu, X.: An EMG-based handwriting recognition through dynamic time warping. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. pp. 4902–4905. IEEE (aug 2010)
7. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology 195(1), 215–243 (mar 1968)
8. Ines Chihi, Afef, A., Mohamed, B.: Analysis of Handwriting Velocity to Identify Handwriting Process from Electromyographic Signals. American Journal of Applied Sciences 9(10), 1742–1756 (oct 2012)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)
10. Lansari, A., Bouslama, F., Khasawneh, M., Al-Rawi, A.: A novel electromyography (EMG) based classification approach for Arabic handwriting. In: Proceedings of the International Joint Conference on Neural Networks, 2003. vol. 3, pp. 2193–2196. IEEE (2003)
11. Le, Q.V., Jaitly, N., Hinton, G.E.: A simple way to initialize recurrent networks of rectified linear units (2015)
12. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation 1(4), 541–551 (dec 1989)
13. Li, C., Ma, Z., Yao, L., Zhang, D.: Improvements on EMG-based handwriting recognition with DTW algorithm. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 2144–2147. IEEE (jul 2013)

14. Linderman, M., Lebedev, M.A., Erlichman, J.S.: Recognition of Handwriting from Electromyography. PLoS ONE 4(8), e6791 (aug 2009)
15. Maldarelli, J.E., Kahrs, B.A., Hunt, S.C., Lockman, J.J.: Development of early handwriting: Visual-motor control during letter copying. Developmental Psychology 51(7), 879–888 (jul 2015)
16. Okorokova, E., Lebedev, M., Linderman, M., Ossadtchi, A.: A dynamical model improves reconstruction of handwriting from multichannel electromyographic recordings. Frontiers in Neuroscience 9, 1–15 (oct 2015)
17. Sinha, N.K., Gupta, M.M., Rao, D.H.: Dynamic neural networks: An overview. In: Proceedings of the IEEE International Conference on Industrial Technology. vol. 1, pp. 491–496. IEEE (2000)
18. Slim, M.A., Abdelkrim, A., Benrejeb, M.: An efficient handwriting velocity modelling for electromyographic signals reconstruction using Radial Basis Function neural networks. In: 2015 7th International Conference on Modelling, Identification and Control (ICMIC). pp. 1–6. IEEE (dec 2015)
19. Smith, L.N.: Cyclical Learning Rates for Training Neural Networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 464–472. IEEE (mar 2017)
20. Talathi, S.S., Vartak, A.: Improving performance of recurrent neural network with relu nonlinearity (2015)
21. Weng, J., Ahuja, N., Huang, T.S.: Cresceptron: a self-organizing neural network which grows adaptively. In: International Joint Conference on Neural Networks (IJCNN). vol. 1, pp. 576–581 (1992)

ISSN 1870-4069

Modelo de agrupamiento de precios en el autoservicio

Leopoldo Javier Gómez Álvarez, Marco Antonio Aristeo García, Luis Alberto Álvarez Ayala

Universidad Panamericana,
Facultad de Ingeniería,
México

{0234446,0152238,0231942}@up.edu.mx

Resumen. En la actualidad el mercado de autoservicios en México presenta un gran dinamismo ya que la mayoría de las cadenas de autoservicios buscan ofrecer el mejor precio a sus clientes con el fin de aumentar su volumen de ventas, lo que las lleva a sacrificar grandes cantidades en margen de utilidad para lograr este objetivo. Por esto decidimos aplicar los siguientes modelos de aprendizaje de máquina que nos permitirán perfilar grupos de artículos que de acuerdo a su rango de precios promedio tengan un comportamiento similar a la diferencia que existe entre su precio y el precio de sus competidores, lo que permitirá generar estrategias puntuales para cada agrupación de productos haciendo más eficientes nuestras inversiones al aplicar rebajas de precio específicas a cada agrupación. Los modelos que empleamos son Agrupamiento Jerárquico,Mean Shift y K-Medias que nos permitirán generar las agrupaciones de acuerdo al comportamiento de los precios y Máquinas de Soporte Vectorial (MSV)que nos ayudará a medir el nivel de precisión de nuestros clusters seleccionados.

Palabras clave: Máquinas de soporte vectorial, clustering, retail, meanshift.

Retail Price Clustering Model

Abstract. Nowadays, the retail in Mexico presents a great dynamism since most of the retailers try to offer the best prices to their clients in order to increase their sales volume, which leads to lost large amounts in profit margin in order to achieve this target. For this reason, we decided to apply the following machine learning models that allow us to create groups of items according to their mean range price and similar patterns with the difference between their price and the competitor's price, this will let us develop unique strategies for each group of products, increasing the profit of their investments by applying specific price reductions to each group. The models used in this article are Hierarchical Grouping,

Mean Shift and K-Means, allowing us to generate clusters according to the patterns of the prices. Vector Support Machines (SVM) will help us to measure the level of precision in our selected groups.

Keywords: Suport vector machine, clustering, retail, meanshift.

1. Introducción

Debido a la estrategia de precios que tienen las cadenas de autoservicio, uno de los principales indicadores al que dan seguimiento es a la diferencia en precios contra sus competidores, lo que es conocido en la industria del autoservicio como diferencial de precios, el cual permite garantizar los mejores precios para el cliente. El cálculo del diferencial de precios se realiza sobre los artículos comparables, es decir aquellos que tienen el mismo código universal de producto o por sus siglas en inglés UPC en ambas cadenas, siendo el UPC un identificador único el cual podemos encontrar en el código de barras de cualquier artículo. El precio promedio de venta de la competencia se obtiene a través de un consolidado de precios del mercado que comparte una agencia de análisis de mercados en donde se detalla el precio de venta promedio a nivel artículo (identificándolo con su código de barras), cadena comercial que lo vende y zona del país en la que se encuentra. La fórmula del diferencial de precios (1)consiste en dividir el precio de venta promedio de la competencia entre el precio de venta promedio de la cadena menos 1, con lo que se obtiene un porcentaje de diferencial:

$$\frac{PrecioPromedioCompetencia}{PrecioPromedioCadena} - 1. \tag{1}$$

A raíz de esto surge la necesidad de identificar la relación que tiene el precio promedio de venta de los artículos con su diferencial de precios, los precios del mercado que usaremos se encuentran a nivel artículo-cadena comercial, les aplicaremos modelos de Agrupamiento Jerárquico,Mean Shift y K-Medias que nos permitirán generar las agrupaciones de acuerdo al comportamiento similar en diferencial de precios en relación a su punto de precio y Máquinas de Soporte Vectorial (MSV)que nos ayudará a medir el nivel de precisión de nuestros clusters seleccionados, todo esto con el objetivo de que una vez identificado el comportamiento del diferencial de los productos analizados se puedan generar estrategias puntuales que permitan hacer más eficientes las inversiones en competitividad de precio. Del set de datos original ejecutaremos los modelos únicamente en la categoría de dulces, ya que es la que representa una muestra más balanceada, como se detallará a continuación.

Existen trabajos que buscan estudiar el comportamiento de los precios en autoservicio [1,2]. En algunos casos desde la óptica de los descuentos basados en la temporalidad, en ese caso para los productos de salsa de tomate y tratando de encontrar la relación misma con la demanda [2] y en otros, bajo circunstancias

específicas para ciertos productos [1] tomando en cuenta una diversidad geográfica en la muestra de las observaciones con las que se desarrolla el artículo. Lo que podemos aprender de ambos enfoques es que siempre se buscará entender el funcionamiento de los precios en el autoservicio.

En [2] se examina los precios en el supermercado para explicar la reducción temporal de precios pasados Basado en el análisis de datos descriptivos, se concluye que el efecto de acumulación de demanda es asimétrico. Afecta la demanda durante precios bajos, pero no durante precios altos. Por lo tanto, las estimaciones de la elasticidad de la demanda de Ketchup basadas en un modelo estático pueden ser sustancialmente sobrevaloradas.

2. Descripción de la propuesta

Nuestra propuesta consiste en el uso de Agrupamiento Jerárquico, Mean Shift y K-medias para perfilar en distintos grupos el comportamiento del precio de los artículos frente a los precios del mercado.

2.1. Descripción General

Con la gráfica del codo calculamos el valor óptimo de centroides. A través de los modelos de Agrupamiento Jerárquico, MeanShift y K-medias buscamos definir los grupos de artículos de la categoría dulces de acuerdo a su precio promedio y diferencial de precios, con el objetivo de definir las etiquetas del set de datos y en el modelo de Máquinas de Soporte Vectorial logramos optimizar esta agrupación.En Agrupamiento Jerárquico con la función agglomerative clustering definimos la localización de los centroides y los productos que se localizarán en cada uno, también graficamos estos clusters en un dendograma, lo que nos permitió identificar el número de artículos que se localizaban en cada uno.Con Mean Shift convertimos nuestro set de datos a un arreglo Nunmpy y obtuvimos las etiquetas y los centroides. Al final lo graficamos para visualizar la localización de estos.En Kmedias se calcularon los centroides a través de la distancia Euclidiana e iteramos hasta que la suma de distancia se redujera hasta que los cambios en distancia fueran mínimos y por último lo graficamos.Con las etiquetas generadas en K-medias usamos el modelo de Máquinas de Soporte Vectorial, optimizando las agrupaciones en las que utilizamos un Kernel lineal por parte de los hiperparámetros, al final evaluamos el modelo. Las etapas a seguir se observan en Fig. 1.

2.2. Materiales

El dataset contiene 3804 productos de una cadena de servicio, con 7 columnas en las cuales podemos encontrar información relacionada a cada uno de los productos.

En primer grado, se valora la distribución de todas las categorías que contiene el set de datos con respecto al precio (ver Fig. 2): La gráfica nos indica que los

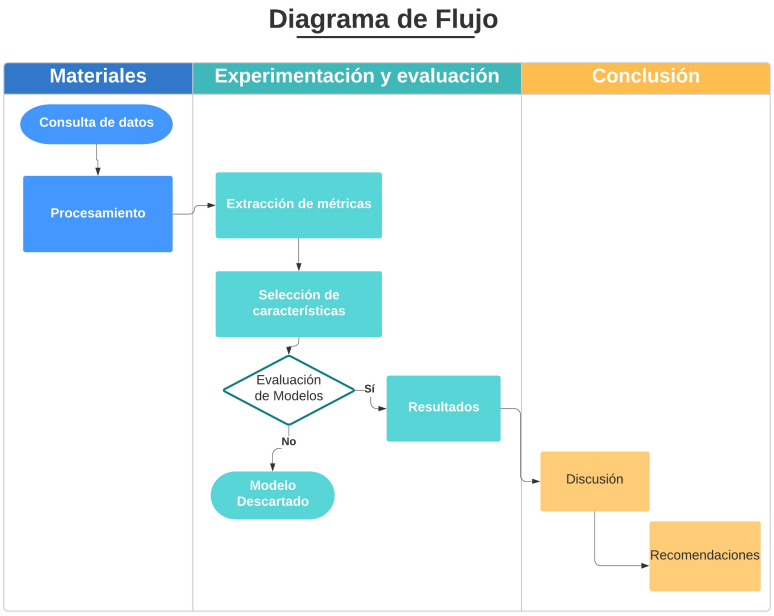


Fig. 1. Diagrama del flujo de trabajo.

Tabla 1. Resumen de datos.

Artefacto	Descripción
Total de observaciones analizadas	3804
Formato del archivo	csv
Columnas Totales	203
Nuevas columnas	2

Tabla 2. Diccionario de datos en columnas seleccionadas.

Columna	Tipo de Dato	No. Valores
Categoría	String	22
UPC	String	3804
Descripción	String	3804
Precio Cadena	String	2391
Último Porcentaje Dif Reportado	String	482
imp_Precio	Float	2391
pct_ult_dif	Float	482

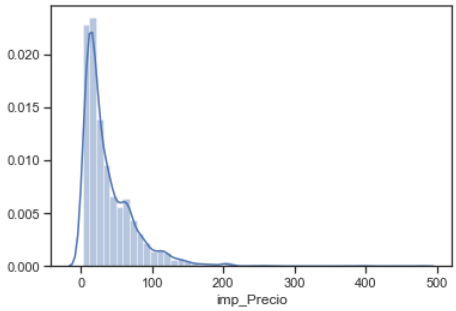


Fig. 2. Distribución de precios de todos los productos.

precios (ver Fig. 3) que mas se repiten en nuestro set de datos, están aproximadamente entre un rango del 0 al 60, por lo cual, decidimos utilizar la categoría de Dulces, debido a que esta categoría alimenta parte de esta gráfica por el rango de precios que contiene la categoría como se puede ver a continuación:

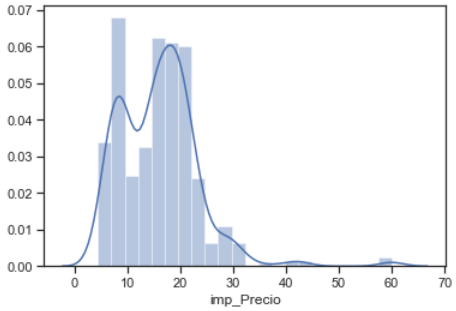


Fig. 3. Distribución de precios de la categoría de ´Dulces.

Realizamos una gráfica de caja (ver Fig. 4) para la categoría de precios, en la cual podemos notar claramente valores atípicos, los cuales pueden afectar algunos de los modelos de clusterización que utilizaremos a continuación. Es importante conservar los valores atípicos encontrados ya que cada punto representa un producto y se busca perfilar la totalidad de los productos.

2.3. Métodos de aprendizaje

Agrupamiento jerárquico aglomerativo Es un algoritmo de agrupación no supervisado, también se le conoce como aglomeramiento anidado ya que cada observación comienza en su propia agrupación y con cada movimiento la une

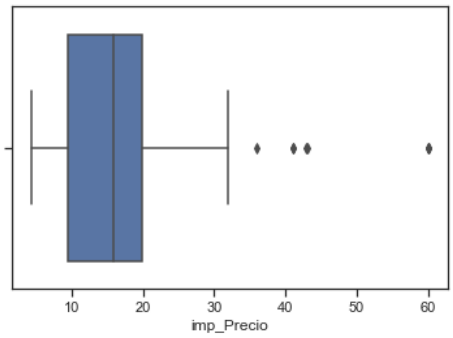


Fig. 4. Boxplot de precios de la categoría de ´Dulces.

con otra agrupación por lo que empieza con K= N clusters y procede uniendo el cluster con el otro que se encuentre más cercano fusinandolos obteniendo así K-1 agrupaciones, esto se repite hasta alcanzar el número deseado de agrupaciones, el criterio para saber que agrupaciones se fusionan es la distancia Euclidiana [4].

Mean shift Es un algoritmo de agrupación no paramétrico que no requiere que se establezca previamente el número de las agrupaciones. Se establece la función $N(X)$ para determinar los puntos cercanos x ∈ X, la medida de distancia usada es la distancia euclidiana y el Kernel $K(d)$ el cuál es un Kernel Gaussiano y la distancia entre dos puntos expresado en la siguiente fórmula:

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}. \tag{2}$$

K-medias Es un algoritmo de agrupación no supervisada que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Primero se define el número de grupos, estableciendose K centroides en el espacio de los datos, después cada grupo de datos se asigna al centoride más cercano, el siguiente paso es que se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo, después se iteran estos últimos dos pasos hasta que los centroides se mueven mínimamente. El algoritmo de k medias se centra en la optimización al minimizar la suma de las distancias cuadráticas de cada objeto al centroide de su agrupación [5]

Máquinas de soporte vectorial Es un modelo de aprendizaje supervisado que tiene como inicio el uso de un espacio de las hipótesis de funciones lineales en un espacio de mayor dimensión inducido por un Kernel, en el cual las hipótesis son entrenadas por un algoritmo tomado de la teoría de optimización, el cuál

utiliza elementos de la teoría de generalización. La forma en la que la MSV aprende la función objetivo es cambiando la representación de la función, es decir se mapean los espacios de entradas X a un nuevo espacio de características $F = \phi(x)|x \in X$ En donde las cantidades que se intoducen para describir los atributos son conocidos como características [3].

3. Experimentación

Gráfica de codo Por medio de la gráfica de codo, podemos obtener el valor óptimo de K, con base en la curva obtenida en la gráfica (ver Fig. 5), usando los parámetros de la tabla 3 se escoge entonces ese punto en donde ya no se dejan de producir variaciones importantes . En este caso, vemos que esto se produce a partir de k >= 3, por lo que evaluaremos los resultados del agrupamiento, con el valore de 3 a fin de observar el comportamiento del modelo.

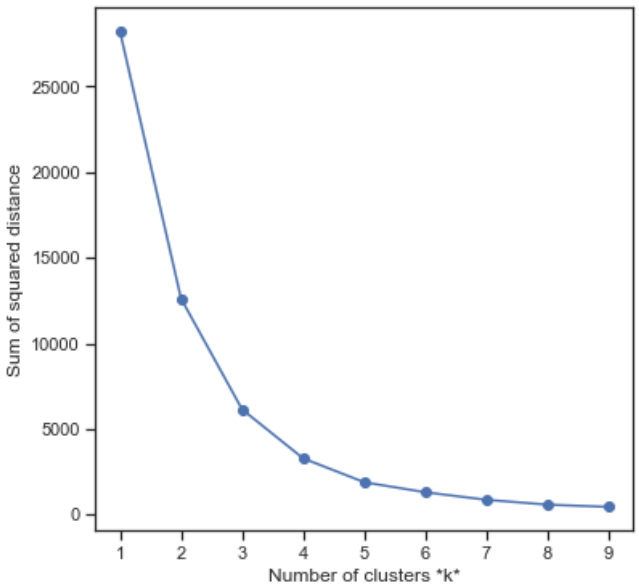


Fig. 5. Gráfica de codo.

Tabla 3. Parámetros para generar gráfica de codo.

Parámetro	Valor
Datos	'imp_Precio','pct_ult_dif'
Longitud de gráfica	range(1, 10)

Agrupamiento jerárquico De la librería Sklearn en Python, importamos la función agglomerative clustering (ver Fig. 6) en la cuál definimos que los datos se debían agrupar en tres clusters (ver Cuadro 4) y lo gráficamos como se muestra a continuación utilizando del set de datos, el precio y último diferencial. Las columnas utilizadas del set de datos son el precio promedio y el último diferencial.

Tabla 4. Parámetros para generar agrupamiento jerárquico.

Parámetro	Valor
Número de clusters	3
Linkage	average

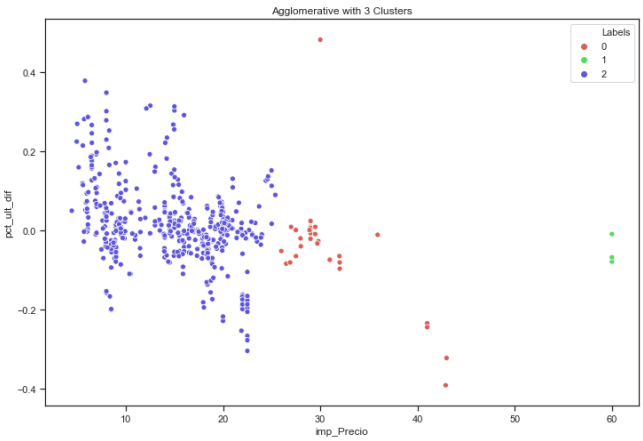


Fig. 6. Cluterización con agrupamiento jerárquico.

En el dendrograma (ver Fig. 7) podemos notar como se van formando los grupos conforme a la siguiente jerarquización con el enfoque de abajo hacia arriba.Se determina la distancia vertical más grande que no se cruza con ninguno de los otros grupos a partir del eje y. Comienza con muchos grupos pequeños y se combinan para crear grupos más grandes:

Mean Shift Primero convertimos el set de datos de un arreglo Pandas a un arreglo Numpy,después utilizamos Mean shift y lo ajustamos al set de datos,en el siguiente paso dejamos que el modelo estableciera las etiquetas y centroides usando los parámetros que se muestran en el cuadro 6. Al final graficamos los centroides y sus respectivos clusters de datos (ver Fig. 8). Las columnas utilizadas del set de datos son el precio promedio y el último diferencial.

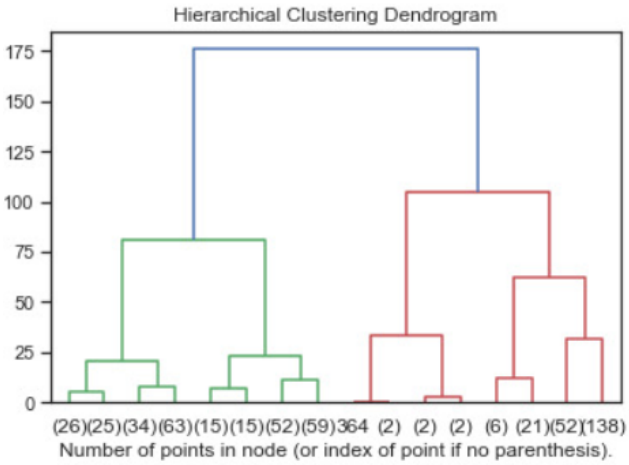


Fig. 7. Dendograma agrupación jerárquica.

Tabla 5. Parámetros para generar MeanShift.

Parámetro	**Valor**
Quantile	0.9
n_samples	500
bin_seeding	True

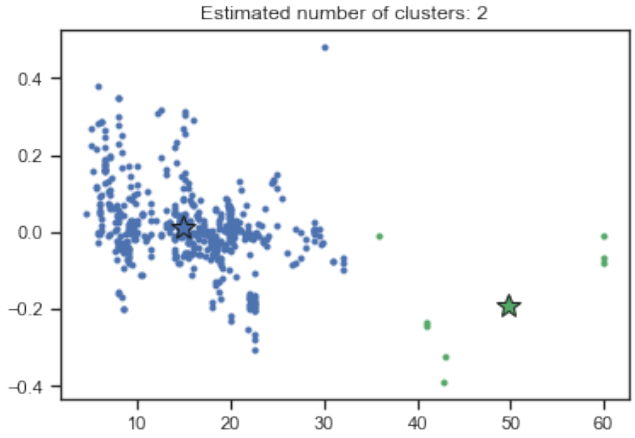


Fig. 8. Cluterización con agrupamiento mean shift.

K-medias En el primer paso cada fila de nuesto conjunto de datos se asigna al centroide más cercano basado en la distancia Euclideana como se aprecia en (3):

$$\arg\min_{c_i \in C} dist(c_i, x)^2. \tag{3}$$

Después recalculamos los centroides tomando una media de los puntos asignados en el paso anterior. Iteramos hasta que la suma de las distancias se minimiza y no hay cambios en los puntos asignados a los grupos. Graficamos los datos agrupados con un scatterplot diferenciandolos con colores (ver Fig. 9). Las columnas utilizadas del set de datos son el precio promedio y el último diferencial.

Tabla 6. Parámetros para generar K-medias.

Parámetro	Valor
Número de clusters	3

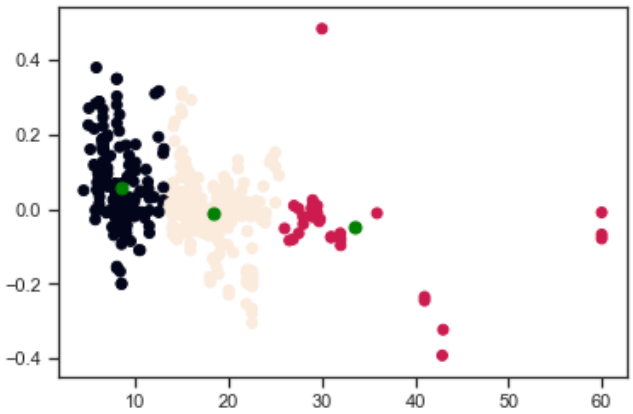


Fig. 9. Cluterización con k medias.

Mediante una tabla de correlación confirmamos la independencia entre el precio promedio y el porcentaje de diferencial. Por último enlistamos los artículos que pertenecen a cada clasificación 7.

Máquinas de Soporte Vectorial Utilizamos el modelo de máquinas de soporte vectorial junto con el modelo de K-medias, ya que SVM funciona por medio de etiquetas y este modelo no es capaz de realizarlo, por lo cual optamos por utilizar las etiquetas obtenidas previamente en el modelo de K-medias.

Tabla 7. Tabla de correlación.

Métrica	UPC	Precio Promedio	Último Diferencial
UPC	1.000000	-0.032489	-0.121398
Precio Promedio	-0.032489	1.000000	-0.386194
Último Diferencial	-0.121398	-0.386194	1.000000

Para utilizar este modelo, dividimos los datos en 70 % entrenamiento y 30 % para pruebas (ver Cuadro 8). En este método, es recomendable estandarizar el subconjunto de datos de entrenamiento y apartir de esto empezar con el modelado, utilizando un kernel lineal por parte de los hiperparámetros. Al final, evaluamos nuestro modelo por medio de una matriz de confusión (ver Fig. 10), la cual nos arrojo una exactitud del 98 %, utilizando un K=3.

Tabla 8. Parámetros para generar MSV.

Parámetro	Valor
Kernel	lineal
test_size	0.3
random_state	0

4. Resultados y discusión

Al realizar el análisis de los modelos planteados en este documento encontramos los siguientes pros y contras en cada método.

En el modelo de Kmedias tuvimos un error alto, ya que nuestro set de datos contenía puntos atípicos, lo que nos llevaría a tener que descartar productos del mismo para disminuir el error. Mientras que en Mean Shift esto no afectó las agrupaciones obtenidas debido a que el modelo define por si mismo el número de agrupaciones y la ubicación de los centroides, permitiendonos conservar todos los artículos de la categoría de dulces sin presentar un error alto en el modelo, lo que asegurará que las estrategias de competitividad de precios que generemos a partir de estas agrupaciones contemplen todo el catálogo de productos. Además de que el no incluir un producto en la estrategia de precios afectaría la percepción que tiene el cliente de la cadena de autoservicio y dejaría en desventaja al artículo frente al precio más bajo ofrecido por la competencia.

Como se muestra en las siguientes 4 graficas, podemos notar que hay consitencia en los grupos definidos por los clusters. Los primeros grupos definidos por el cluster jerarquico y por meanshift son similares respecto a los precios de los productos. Mientras que en K-medias, notamos que ese primer bloque, el algoritmo lo divide en 2 grupos:

	precision	recall	f1-score	support
0	1.00	0.93	0.97	46
1	1.00	1.00	1.00	7
2	0.97	1.00	0.99	101
accuracy			0.98	154
macro avg	0.99	0.98	0.98	154
weighted avg	0.98	0.98	0.98	154

```
[[ 43   0   3]
 [  0   7   0]
 [  0   0 101]]
```

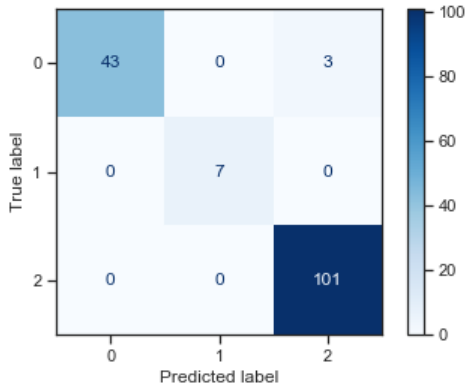
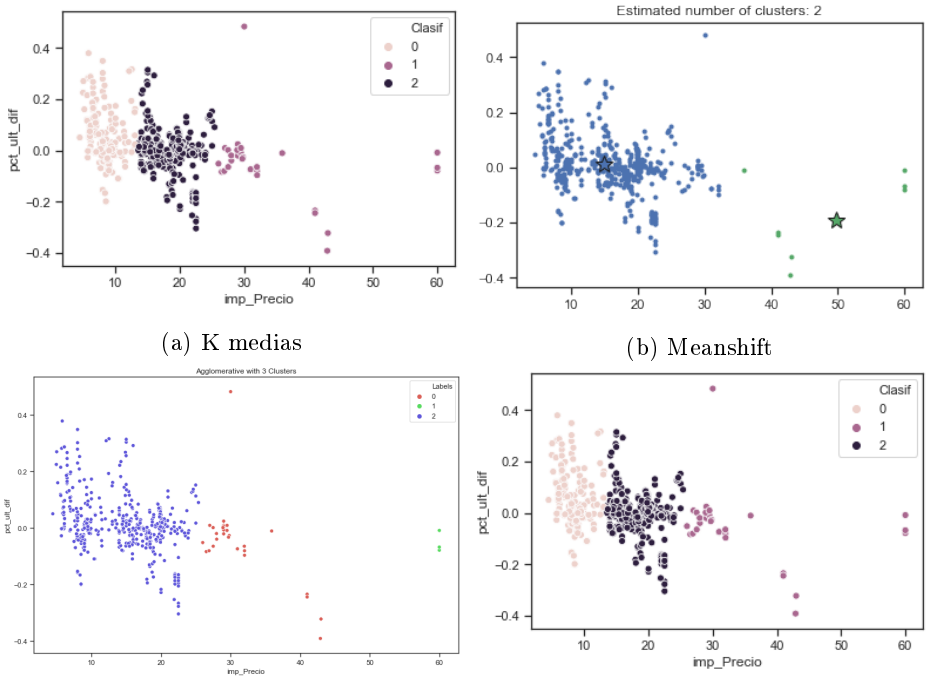


Fig. 10. Matriz de confusión para SVM.

Con los dos grupos identificados en los agloritmos Mean Shift y jerárquico podemos definir que el grupo que se encuentra entre cero y treinta pesos (cluster 0) debe tener una estrategía diferente al que tiene un precio mayor a treinta pesos (cluster 1), ya que los artículos contenidos en el primero presentan una mayor rotación debido a su punto de precio bajo y por lo tanto es importante profundizar la inversión en diferencial de precios para contar con un precio competitivo mientras que los productos del cluster 1 son principalmente importados u orgánicos, que se encuentran dirigidos a un mercado de clientes con un mayor nivel socioeconómico y por lo tanto la inversión en diferencial de precios no debe ser tan agresiva para no afectar los márgenes de utilidad alto que aporta este segmento.

Por lo que podemos afrimar que aquellos productos con un precio menor a treinta pesos (cluster cero) deben continuar con una estrategia inversión agresiva que se traduzca en un diferencial mayor y en los productos con un precio mayor a treinta pesos (cluster 1) la inversión debe de ser mínima, lo que nos ayudará a ser más eficiente con los recursos invertidos y al mismo tiempo a asegurar la competitividad de nuestros precios donde es necesario.



(a) K medias (b) Meanshift

(c) Jerarquico (d) SVM

Fig. 11. Agrupaciones por modelos.

5. Conclusiones

Podemos concluir que los modelos de Agrupamiento Jerárquico y Meanshift son convenientes para identificar el comportamiento de precios de los artículos en el autoservicio lo que se traduce en una herramienta útil para la creación de una estrategia de precios efectiva que nos permita maximizar el margen de los productos e invertir de forma eficiente para tener la competitividad necesaria frente al mercado.

Referencias

1. Hosken, D., Reiffen, D.: Patterns of retail price variation. RAND Journal of Economics pp. 128–146 (2004)
2. Pesendorfer, M.: Retail sales: A study of pricing behavior in supermarkets. The Journal of Business 75(1), 33–66 (2002)
3. Resendiz Trejo, J.A.: Las maquinas de vectores de soporte para identificación en línea. Centro de Investigación Y de estudios avanzado del Instituto Politécnico Nacional Departamento de Control Automático 1(1), 1–68 (2006)

4. Teichgraeber, H., .B.A.R.: Systematic comparison of aggregation methods for input data time series aggregation of energy systems optimization problems. Computer Aided Chemical Engineering 44(1), 955–960 (2018)
5. Wu, J.: Advances in K-means Clustering, Springer Science & Business Media, vol. 1, chap. 1, pp. 1–50. Springer (2012)

ISSN 1870-4069

Clasificación de plantas de maíz y maleza: Hacia la mejora de la fertilización en México

Marcos Yamir Gómez-Ramos, J. Sergio Ruíz-Castilla,
Farid García-Lamont

Universidad Autónoma del Estado de México,
Centro Universitario UAEM Texcoco,
Estado de México,
México

{marcos.yamir, jsergioruizc}@gmail.com,
fglamont@yahoo.com.mx

Resumen. El cultivo del maíz es muy importante en México y aún podemos encontrar métodos tradicionales de labranza desde la siembra hasta la cosecha; sin embargo, se llegan a observar algunos ejemplos de uso de maquinaria sofisticada. El maíz se siembra en hileras llamadas surcos, separados por unos 0.75 m entre sí. Se siembran de 4 a 5 semillas por mata a una distancia de 0.50 m entre cada una. Por diversos factores, no siempre nace el 100 % de semillas, incluso en ocasiones no germina ninguna semilla. En las primeras semanas se lleva a cabo la fertilización, que consiste en depositar una cantidad de fertilizante químico al pie de la planta. Dicho proceso se hace de manera manual, en el que algún trabajador coloca un puñado de la sustancia por mata, buscando hacerlo de la mejor manera posible, aunque no exacta. Por ende, la actividad resulta lenta, tardada y costosa. Actualmente se ha implementado el uso de maquinaria que deposita fertilizante a lo largo de todo el surco. Con este método se hace más rápido y barato. Sin embargo, se esparce fertilizante en donde hay ausencia de mata de maíz e incluso a la maleza. El fertilizante que no es absorbido por la planta se va al subsuelo, generando contaminación de los mantos acuíferos. El presente trabajo propone la clasificación de plantas de maíz y de maleza, pues al identificar la planta de maíz es posible depositar fertilizante a éstas evitando la maleza e incluso donde no hay planta. Para lograr el objetivo, se obtuvieron 1000 imágenes de maíz y maleza. Se aplicaron métodos de procesamiento de imágenes, como segmentación con PCA y Otsu; después se aplicaron algoritmos de clasificación: Naive Bayes, Random Forest, SVM, KNN, CNN y redes neuronales Backpropagation logrando resultados con hasta el 98.98 % de precisión. Con la finalidad de que la solución mejore el proceso de fertilización.

Palabras clave: Visión artificial, extracción de características, clasificación, fertilización.

Marcos Yamir Gómez-Ramos, J. Sergio Ruíz-Castilla, Farid García-Lamont

Classification of Corn and Weed Plants: Towards Improving Fertilization in Mexico

Abstract. Maize cultivation is very important in Mexico. There are traditional methods for growing maize from planting to harvest. However, there are other cases using sophisticated machinery. During the first weeks after the growing the fertilization is carried out, which consists of depositing a quantity of chemical fertilizer at the foot of the plant. This process is made manually and exactly but this activity is slow and expensive. Currently, the sophisticated machinery deposits fertilizer along the furrow uninterruptedly. This method is faster but the fertilizer is deposited in the absence of maize plant and even to the weed. When the fertilizer is not absorbed by the plant goes to the subsoil, generating contamination of the aquifers. The present work proposes the classification of maize and weed plants. First, we obtained 1000 images from a scenario uncontrolled. We process the images as segmentation with PCA and Otsu. After, we apply classification algorithms as: Naive Bayes, Random Forest, SVM, KNN, CNN and Backpropagation neural networks. The results obtained were a 98.98% of precision. The purpose of work is to improve the techniques of fertilization and avoid the contamination of the subsoil.

Keywords: Artificial vision, features extraction, classification, fertilization.

1. Introducción

Los efectos de los fertilizantes químicos sobre el medio ambiente son incuestionables, su uso genera un riesgo elevado de daños ambientales, como la contaminación de las aguas subterráneas y del suelo y subsuelo en los que se aplican. Es relevante aclarar que un *fertilizante* es cualquier sustancia que se utiliza para aportar nutrientes al suelo, cuyo objetivo es aumentar la concentración de nutrientes que favorezcan y propicien el crecimiento de las plantas. Es importante mencionar que "No todos los fertilizantes actúan igual y tampoco todos los fertilizantes son saludables para el suelo y el medio ambiente"[21]. Los productores agrícolas se han centrado en aportar nitrógeno *(N)*, fósforo *(P)* y potasio *(K)* a los sembradíos, práctica que aún se sigue implementando aunque fue descubierta a mediados del siglo XIX. La contaminación por fertilizantes se produce cuando éstos se utilizan en mayor cantidad de la que pueden absorber los cultivos o cuando se eliminan por acción del agua o del viento de la superficie del suelo antes de que puedan ser absorbidos. La contaminación de los mantos acuíferos por los productos y residuos agroquímicos es uno de los problemas más importantes en casi todos los países desarrollados y, cada vez más, en muchos países en vías de desarrollo. Estudios realizados en el Reino Unido han calculado

que se filtran entre 50 y 60 kilogramos de nitrógeno por hectárea al año y que el 58 % de los nitratos que contaminan los mantos acuíferos proceden de la agricultura [21]. El nitrógeno (N), es el fertilizante que más requieren las plantas de maíz y otros cultivos para crecer, además de ser uno de los insumos más costosos dentro de la producción de un agricultor. "Un agricultor de maíz en Sinaloa, México puede llegar a gastar 1700 dolares por hectárea y de esto, un 40 % a 60 % es lo que gasta en fertilizantes; de este porcentaje, hasta un 80 % corresponde al gasto por el nitrógeno"[21].

Hoy en día, el uso de algoritmos de visión computacional ha sido implementado en diversos campos como la agricultura, manufactura, educación, medicina, entre otras. La agricultura es una de las más importantes en México por ser parte del sector primario. Las aplicaciones en este rubro son variadas: detección de plantas [3], reconocimiento de plantas a partir de hojas [18], clasificación de enfermedades [28], reconocimiento de malezas [24] y detección de frutas [8]. Aunque realizar cualquiera de estas tareas resulta complicado, son ampliamente trascendentales para la agricultura, pues la identificación de plantas, frutas o tallos es vital para la toma de decisiones de los agricultores y sus plantíos.

El documento se divide en las siguientes secciones principales: La sección 2 describe el estado del arte en el uso de algoritmos de visión computacional para la agricultura. La sección 3 explica el método propuesto con una descripción detallada de cada conjunto de datos experimentales. En la sección 4 se muestran los resultados obtenidos y la sección 5 presenta las conclusiones de los autores y trabajo futuro.

2. Estado del arte

En los artículos revisados se encuentran diferentes métodos de extracción de características y técnicas de clasificación haciendo uso de diferentes algoritmos. Enseguida se describe la información relevante de los artículos consultados:

Existen algunas características para poder realizar la identificación de plantas y maleza, entre éstas se encuentran: forma, textura y color [4]. Algunas investigaciones actuales se han centrado en características de histogramas y wavelet [5], representación dispersa [24], madurez de las hojas, movimientos rotativos y calibración [14], utilizando SVM [18], CNN [30], FCN [28], KNN [14] como clasificadores con los mejores resultados.

Otros investigadores como [1] propone clasificar una planta en función de su hoja identificando características morfológicas y descriptores de Fourier, usando una Red Neuronal Artificial, ANN por sus siglas en inglés con Backpropagation como clasificador [4]. Utilizó técnicas de selección de características eliminando atributos que introducen ruido, facilita su uso en conjuntos de datos pequeños [3]. Empleó técnicas de extracción de características geométricas invariantes a escalado, traslación y rotación, texturales y cromáticas. Utilizando un clasificador Bayesiano, Backpropagation y SVM [6] propuso un modelo de reconocimiento de características basado en redes de cápsulas y [8] de igual forma propuso

caracterizar frutas empleando descriptores de Fourier y momentos de Hu, la caracterización del color es la contribución de su artículo.

Algunos estudios son muy completos en cuanto al estudio de la segmentación de imágenes por características de color como el de [7], se basó en las características más importantes del color que son la intensidad y cromaticidad, además hace mención de los espacios de color más utilizados en trabajos relacionados y describe las características de estos espacios comparando las ventajas y desventajas de su uso. Ciertos estudios como el de [15] proponen el uso de un algoritmo genético para reducción de ruido con la característica de dimensionalidad. También [29] propone caracterizar imágenes a través de la forma, tipo de margen y geometría ya que estas características tienen patrones comunes en hojas de varias especies. [4] menciona que los métodos de identificación actuales implican algoritmos avanzados para medir las características morfológicas y de textura de los objetos contenidos en la imagen, ya que éstos proporcionan mucha información para clasificarlos.

En [27] et al. se aborda el problema de distinguir las plantas de maíz de las malas hierbas, implementando una red neuronal artificial (ANN) usando el algoritmo backpropagation, teniendo un porcentaje de detección en maiz del 100 % y en maleza del 80 %. [16] plantea el problema de encontrar una herramienta clasificadora para identificar las malas hierbas en los campos de maíz en la etapa temprana de crecimiento, esto a través de una SVM (Máquina de Soporte Vectorial); los resultados arrojaron que los clasificadores SVM podrían identificar con éxito malezas con una precisión del 100 %. [2] expone el problema de reconocimiento de malezas sobre campo con cultivo con técnicas de Deep Learning, el sistema de visión propuesto está basado en Convolutional Neural Networks (CNN) y utiliza una SVM; los resultados fueron la identificación de la maleza en un 91 % con SVM y 84 % con KMEANS. [11] propuso un enfoque de visión artificial para discriminar cultivos y malezas de imágenes de zanahoria logrando un valor de precisión del 93.8 % utilizando el método de Otsu para la segmentación.

[13] desarrollaron un método para discriminar entre cultivos y malezas con un porcentaje de clasificación del 66.7 %. [26] sugiere un método automático para extraer fruta madura de un fondo agrícola complejo obteniendo un valor de precisión del 95 %; se basa en un algoritmo mejorado en el umbral de Otsu que utiliza una nueva función en el espacio de color OHTA. [23] ofrece un método automático para la detección de vegetación en cultivos herbáceos mediante un algoritmo de análisis de imágenes basado en objetos de umbral a través del método Otsu, la clasificación de la vegetación da como resultado el 90 %. [17] desarrolló un método de segmentación de vegetación en imágenes de remolacha azucarera y maíz con una precisión del 87 %; su método se basó en un modelo Bayesiano Naive usando características de espacios de color RGB y HSV.

En los artículos revisados se encuentran variados métodos de extracción de características y técnicas de clasificación, haciendo uso de algoritmos como: SVM (Máquinas de soporte vectorial), Backpropagation, KNN (K- vecino más cercano), CNN (Redes Neuronales Convolucionales), kmeans, Fuzzy c-means,

cadenas de Markov, árboles de decisión y Naive Bayes; además de que el método Otsu para la segmentación es el más utilizado previo a la extracción de características.

La detección de maíz y maleza en un entorno natural busca detectar las regiones de interés y realizar la segmentación. La visión artificial es la tecnología clave para identificar correctamente las plantas de maíz y las puedan diferenciar de la maleza para realizar una adecuada fertilización, haciéndolo de forma automatizada y precisa.

2.1. Dataset

Como ya lo hemos mencionado anteriormente, es difícil encontrar un dataset específico de plantas de maíz con las características necesarias para los experimentos a realizar, las encontradas en la web no cumplen con ello o las imágenes son insuficientes para las pruebas que se necesitan hacer. Debido a esto se creó un dataset propio, que incluye 1000 imágenes de alta resolución, de las cuales obtendremos la información para realizar las pruebas en Matlab y Python.

Tabla 1. Características de imágenes del dataset.

Clase	Planta	Imágenes	Ancho	Altura	Formato
1	Maíz	500	768	1024	JPG
2	Maleza	500	768	1024	JPG

Todas las capturas se hicieron a cielo abierto. Se obtuvieron 1000 fotos para las primeras pruebas, tomadas con un celular iPhone 8 plus con cámara de 12 megapixeles y un selfie stick modelo L01. La toma de fotos se realizó de las 10:00 a las 12:00 horas, el día 9 de mayo del 2020, usando para las muestras todas las plantas de 10 surcos de aproximadamente 100 metros cada uno.

Fig. 1. Imágenes de maíz y maleza.

3. Metodología

En esta sección, se propone un método que permite la identificación con mayor precisión de las plantas de maíz y de maleza en los cultivos, utilizando técnicas de extracción de características y algoritmos de inteligencia artificial; contribuyendo así a la eficiencia en la aplicación y uso de fertilizantes agrícolas en los cultivos de maíz en México, disminuyendo pérdidas económicas a los agricultores, además de evitar la contaminación de suelos y mantos acuíferos. La mayoría de los algoritmos implementados en este estudio han sido codificados internamente en Matlab R2019b a excepción de CNN que fue codificado en Python (versión 3.5) . Los cálculos se implementaron en una computadora MacBook Pro 2014 con procesador Intel core i5 de 2.6 Ghz y memoria RAM de 8 Gb.

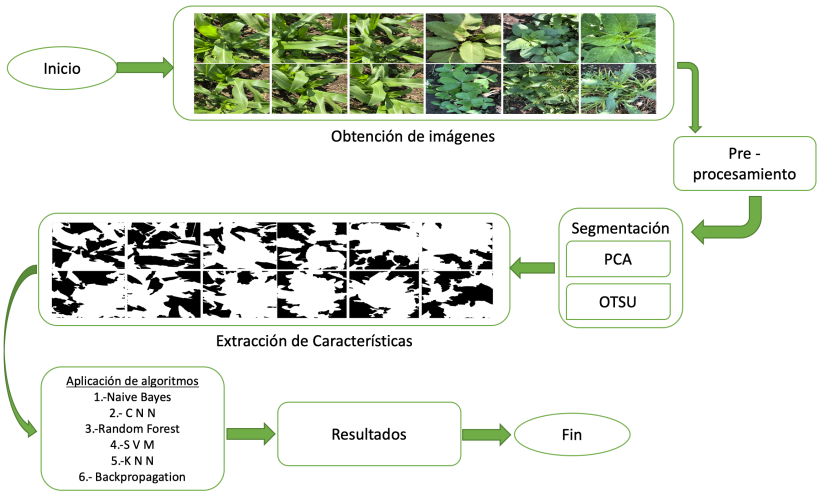


Fig. 2. Metodología usada.

3.1. Pre-procesamiento

El tratamiento de imágenes es una técnica que permite modificar una imagen con el objetivo de lograr una mayor calidad, o bien para obtener una composición totalmente diferente que distorsione la realidad. En nuestro caso, como lo especificamos en el apartado de captura de imágenes, el dataset incluye fotografías de alta resolución. Para un mejor manejo de las mismas y usando el software Matlab se han reducido, eliminando el ruido y opacidad, disminuyendo el brillo, etc. Para reducir el ruido en las imágenes, los métodos de pre-procesamiento son usados antes de la segmentación, pues condicionan la calidad de esta última, por

lo tanto, son necesarios para un diagnóstico preciso. Para mejorar la calidad de la segmentación, el filtro Gaussiano se utiliza para reducir el ruido en la imágenes; esto se realiza mediante una convolución con una función Gaussiana centrada en cada uno de los píxeles de la imagen.

El valor final de cada píxel se obtiene a partir de la contribución de todos sus vecinos ponderados según la función Gaussiana, el efecto que produce es una suavización y una pérdida de detalle de la imagen original. La función Gaussiana se muestra en la ecuación 1:

$$G(x,y) = \frac{1}{\sqrt{(2\pi\sigma)^2}} e^{-}\left(\frac{x^2+y^2}{2\sigma^2}\right), \tag{1}$$

donde x , y y σ son constantes reales; el parámetro x es el valor del punto más alto de la campana; y es la posición del centro de la campana y σ es la desviación estándar que controla el ancho de la campana. El grado de suavizado depende de la desviación estándar σ de la función Gaussiana y los valores mayores de σ implican mayor suavizado de la imagen.

En resumen, el uso del filtro Gaussiano en este apartado es para reducir el impacto de la iluminación y otros factores en la imagen original, logrando así mejorar su calidad, realizando una segmentación efectiva y adecuada para la siguiente etapa.

3.2. Segmentación

PCA (Análisis de Componentes Principales), es una técnica de selección de características concreta que utiliza una transformación ortogonal para convertir un conjunto de observaciones de variables, posiblemente correlacionadas, en un conjunto más reducido de variables que ya no guardan correlación y que se conocen como componentes principales [20]. La razón de usar PCA es porque puede minimizar las correlaciones lineales entre variables y maximizar la entropía de la información sobre distintas extracciones principales; también PCA proporciona una forma directa de reducción de dimensiones.

Otsu es una de las técnicas de binarización de imágenes digitales ampliamente utilizadas en inteligencia artificial, sobre todo cuando se trabaja en imágenes reales tomadas en entornos externos; este método que se adapta a diversas condiciones se basa en conceptos estadísticos, en concreto se utiliza la varianza que es una medida de dispersión de valores (dispersión de los niveles de gris); el objetivo del método es calcular el valor del umbral de forma que la dispersión dentro de cada clase sea lo más pequeña posible, pero al mismo tiempo que la dispersión sea lo más alta posible entre clases diferentes. Otra carácteristica importante de Otsu radica en que es un método no supervisado, es decir, es automático y no necesita supervisión humana ni información previa de la imagen.

Las imágenes de maíz y maleza fueron segmentadas con éxito empleando los siguientes pasos: 1) Cálculo de alto contraste en escala de grises a partir de combinación lineal óptima de los componentes de color RGB [9,22]; 2) Estimar

la frontera óptima ejecutando el algoritmo de segmentación de borde adaptativo *Otsu* [9,22] y una fase del análisis de componentes principales *PCA* [20]; 3) Operaciones morfológicas para rellenar posibles espacios vacíos en la imagen segmentada [9,22].

Todo esto con el fin de obtener la mejor segmentación aun teniendo cambios en las condiciones globales de brillo. Al segmentar la imagen, el sistema puede utilizar únicamente la región de las hojas, determinar sus bordes y calcular las propiedades mediante la extracción de características [3].

3.3. Extracción de características

Cuando una imagen es segmentada con éxito, el sistema hace uso de la región de interés y calcula las propiedades mediante la extracción de características, en este experimento solo se extraen características texturales.

La extracción de características es considerada una piedra angular de los algoritmos de aprendizaje automático; la correcta implementación de los métodos de extracción de características define los descriptores utilizados para el reconocimiento de maiz y maleza.

Los métodos de extracción de características más utilizados son: extracción de características geométricas, texturales y cromáticas. Las imágenes tienen muchas características, las cuales al extraerse se van agregando a un vector que contiene todas las características de la imagen.

En esta sección la imagen está representada a través de un conjunto de valores numéricos con un potencial discriminativo; por lo que, las características obtenidas son invariantes a escala, rotación y traslación, permitiendo al clasificador reconocer objetos a pesar de tener diferentes tamaños, posiciones y orientaciones.

Características texturales: Estás características dan lugar a una propiedad que puede ser rugosa, áspera, suave, entre otras características más. La textura es invariante a los desplazamientos porque repite un patrón a lo largo de una superficie; por lo tanto, allí es donde se explica por qué la percepción visual de una textura es independiente de una posición visual. Las características texturales se extraen de las superficies, en este caso de las hojas del maíz y de la maleza; para este fin se implementó el algoritmo Haralick [10] utilizando matrices de coincidencia de niveles grises. Los descriptores utilizados por Haralick se describen a continuación:

$$f_1 = \sum_i \sum_j \left[p(i,j)^2 \right], \tag{2}$$

$$f_2 = \sum_{n=0}^{N_g - 1} n^2 \left\{ \begin{array}{c} \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \\ |i-j| = n \end{array} \right\}, \tag{3}$$

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left[ijp(i,j) - \mu_x \mu_y \right]}{\sigma_x \sigma_y}, \tag{4}$$

$$f_4 = \sum_i \sum_j (i - \mu_x)^2 p(i,j), \tag{5}$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i-j)^2} p(i,j), \tag{6}$$

$$f_6 = \sum_{i=2}^{2N_g} i P_{x+y}(i), \tag{7}$$

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 P_{x+y}(i), \tag{8}$$

$$f_8 = -\sum_{i=2}^{2N_g} P_{x+y}(i) \log\{P_{x+y}(i)\}, \tag{9}$$

$$f_9 = -\sum_i \sum_j p(i,j) \log\{p(i,j)\}, \tag{10}$$

$$f_{10} = \sum_{i=0}^{N_g - 1} (i - f_8)^2 P_{x-y}(i), \tag{11}$$

$$f_{11} = -\sum_{i=0}^{N_g - 1} P_{x-y}(i) \log\{P_{x-y}(i)\}, \tag{12}$$

$$f_{12} = \frac{HXY - HXY1}{\text{máx}\{HX, HY\}}, \tag{13}$$

$$f_{13} = (1 - e^{[-2(HXY2 - HXY)]})^{\frac{1}{2}}, \tag{14}$$

$$f_{14} = (Segundo\ valor\ más\ grande\ de\ Q)^{\frac{1}{2}}, \tag{15}$$

donde $p(i,j)$ = define el valor de la matriz de co-ocurrencia en la posición (i,j). N_g = Niveles de intensidad trabajados en la región segmentada de la imagen. $p_x(i)$ = $i$ésima entrada en la matriz de probabilidad de co-ocurrencia original que se obtiene sumando las filas en $p(i,j)$. $p_y(j)$ = $j$ésima entrada en una matriz de probabilidad de co-ocurrencia original que se obtiene mediante la suma de las columnas en $p(i,j)$. R = Número total de elementos en la matriz de co-ocurrencia . μ_x,μ_y = Media de P_x y P_y. σ_x,σ_y= Desviación estándar de P_x y P_y. HX, HY = Entropía de P_x y P_y. El vector de características texturales X_t obtenido puede ser representado como:

$$Xt = [x_1, x_2, \ldots, x_{85}], \tag{16}$$

$$Xt = [x_{Rlbp}, x_{RH}, x_{Glbp}, x_{GH}, x_{Blbp}, x_{BH}], \tag{17}$$

donde x_{Rlbp},x_{Glbp},x_{Glbp} representan las características (LBP) [12,25] obtenidas en el canal de color R, G y B respectivamente, x_{RH}, x_{GH} y x_{BH} representan las características texturales de Haralick obtenidas en los canales R, G and B respectivamente.

Una vez obtenidas las características más relevantes de las plantas de maíz y maleza, en la siguiente sección se utilizarán seis algoritmos diferentes para llevar a cabo pruebas y generar los resultados. Los algoritmos fueron elegidos por su uso en experimentos e investigaciones anteriores mencionadas en el estado del arte; las corridas de los algoritmos de clasificación ayudarán al desarrollo de un análisis comparativo de los métodos de extracción de características texturales utilizados en la segmentación (PCA y Otsu).

3.4. Algoritmos de clasificación

Se han utilizado para pruebas los siguientes algoritmos: En la primer experimentación usando Matlab *y el entorno Weka (Waikato Environment for Knowledge Analysis)* se probó con Naive Bayes, Random Forest, SVM, KNN y Backpropagation; para la segunda ronda de pruebas se realizó en Python con el algoritmo CNN.

Naive Bayes: Los clasificadores bayesianos se basan en la teoría de decisión de Bayes, el principio de Bayes provee una metodología fundamental para resolver problemas de clasificación de patrones cuando la distribución de probabilidad de los patrones es conocida.

Random Forest: Es un algoritmo compuesto por los clasificadores de árboles de decisión, cada árbol depende de los valores de un vector aleatorio con muestreo independiente y con la misma distribución para todos los árboles en el bosque.

SVM: Las Máquinas de Vectores Soporte (SVM) son unas técnicas de clasificación muy utilizadas en los últimos años, los puntos clave de las SVM son el uso de kernels al trabajar con conjuntos no-lineales, la ausencia de los mínimos locales, la solución depende de un pequeño subconjunto de datos y el poder discriminativo del modelo obtenido al optimizar el margen de separabilidad entre clases, estas características permiten a las SVM obtener resultados muy competitivos en comparación a otros clasificadores.

KNN: El algoritmo KNN clasifica un nuevo punto en el conjunto de datos basado en la distancia euclidiana, encontrando las k distancias más cercanas al objeto a clasificar.

Backpropagation: De los distintos algoritmos de aprendizaje de las redes neuronales artificiales, uno de los más utilizados es el algoritmo Backpropagation. Este algoritmo cambia iterativamente los pesos entre las neuronas minimizando el error cuadrático entre la salida deseada y la obtenida con los pesos actuales; todo esto para llevar a cabo el proceso de aprendizaje.

CNN: Las Redes Neuronales Convolucionales (Convolutional Neural Networks), es un tipo de red neuronal enfocada al proceso de clasificación de imágenes, texto, audio y discursos. Las CNNs están inspiradas en las redes con perceptron multi-nivel. La CNN permite hacer una convolución mediante un filtro de 3x3 o 5x5 sobre la matriz de píxeles de la imagen fila por fila hasta recorrer

toda la imagen, una activación RELU y un pooling. Como resultado se obtiene una nueva matriz a la cual se le puede aplicar otra vez una convolución.

Es posible aplicar varias convoluciones; de ahí lo de aprendizaje profundo. Después de aplicar las convoluciones se obtiene una red completamente conectada que puede ser clasificada mediante una función como softmax. La arquitectura de una CNN es como la de la Figura 3.

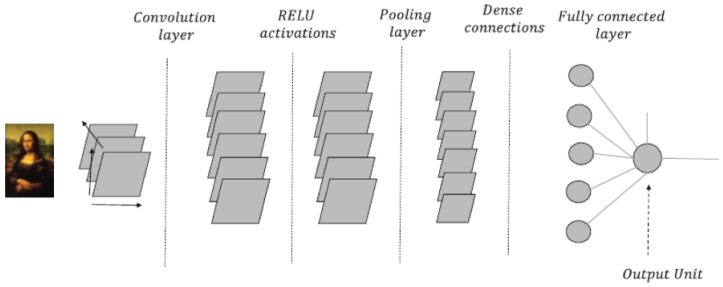


Fig. 3. Arquitectura de una CNN pro deep learning with TensorFlow [19].

Para este trabajo se ha usado Python y la librería Keras de TensorFlow. Para implementar una CNN se requieren dos conjuntos de datos, en este caso se tomaron 800 imágenes para el entrenamiento y 200 para la evaluación. Al final se aplicó validación cruzada con 5 configuraciones de imágenes que dio como resultado un 96 % de efectividad [19].

4. Resultados

En esta sección, se presentan los resultados de los experimentos realizados en las pruebas; mostrando porcentajes de precisión para reconocimiento de planta de maíz y maleza usando diferentes algoritmos de clasificación y la comparación de resultados usando dos métodos de segmentación diferentes. Las abreviaturas en las tablas son: Acc=Accuracy; S=Sensibilidad; E=Especificidad; P=Precisión; R=Recall; F-m=F-measure y MCC=Coeficiente de Correlación de Mattew que son las *medidas de desempeño* generadas en los resultados.

En la Tabla 2 se pueden ver los resultados obtenidos al hacer uso del método de segmentación PCA y la comparación de los algoritmos de clasificación utilizados; siendo el mejor porcentaje de precisión Backpropagation con 98.78 %.

En la Tabla 3 se pueden ver los resultados obtenidos al hacer uso del método de segmentación Otsu y la comparación de los algoritmos de clasificación utilizados; siendo el mejor porcentaje de precisión Backpropagation con 98.98 %.

La figura 4 corresponde a la gráfica donde se muestran los mejores resultados de esta investigación, comparando los dos métodos de segmentación utilizados y mostrando los algortimos probados, Naive Bayes, CNN, Random Forest, SVM,

Tabla 2. Resultados con segmentación PCA.

Algoritmo	Acc	S	E	P	R	F-m	MCC
Naive Bayes	78.44 %	0.909 %	0.658 %	0.803 %	0.784 %	0.781 %	0.587 %
Random Forests	93.11 %	0.961 %	0.90 %	0.933 %	0.931 %	0.931 %	0.864 %
SVM	93.21 %	0.969 %	0.894 %	0.935 %	0.932 %	0.932 %	0.867 %
KNN	93.62 %	0.967 %	0.904 %	0.938 %	0.936 %	0.936 %	0.874 %
CNN	96 %	1.0 %	0.92 %	0.96 %	0.96 %	0.96 %	0.954 %
Backpropagation	**98.78 %**	0.989 %	0.985 %	**0.988 %**	0.988 %	0.988 %	0.976 %
Promedio	92.19 %	0.965 %	0.876 %	0.926 %	0.921 %	0.921 %	0.853 %

Tabla 3. Resultados con segmentación Otsu.

Algoritmo	Acc	S	E	P	R	F-m	MCC
Naive Bayes	77.3 %	0.898 %	0.647 %	0.792 %	0.773 %	0.769 %	0.564 %
Random Forests	91.9 %	0.941 %	0.897 %	0.92 %	0.919 %	0.919 %	0.839 %
SVM	95.6 %	0.975 %	0.935 %	0.956 %	0.956 %	0.956 %	0.912 %
KNN	93.2 %	0.955 %	0.909 %	0.933 %	0.932 %	0.932 %	0.866 %
CNN	96 %	1.0 %	0.92 %	0.96 %	0.96 %	0.96 %	0.954 %
Backpropagation	**98.98 %**	0.987 %	0.991 %	**0.99 %**	0.99 %	0.99 %	0.98 %
Promedio	92.16 %	0.959 %	0.883 %	0.925 %	0.921 %	0.921 %	0.852 %

KNN y Backpropagation en el eje x; y en el eje y podemos ver el porcentaje de precisión obtenido.

La línea azul pertenece a las pruebas realizadas y los resultados obtenidos con el método de segmentación *PCA* y la línea naranja pertenece a las pruebas obtenidas con el método de segmentación *Otsu*.

La precisión más alta se obtuvo con el método de segmentación *Otsu* y el algoritmo *Backpropagation* con un 98.98 % (99 %).

5. Conclusiones y trabajo futuro

Después de concluir nuestra metodología, podemos inferir que, a partir de 1000 imágenes de plantas de maíz y de maleza tomadas directamente del sembradío sin un ambiente controlado, se ha creado un Dataset de 500 imágenes de plantas de maíz y 500 de malezas. A este Dataset se le aplicó un procesamiento de PCA y Otsu para proceder a la clasificación. Mediante las técnicas Naive Bayes, CNN, Random Forest, SVM, KNN y Backpropagation se lograron resultados de clasificación muy interesantes. Los algoritmos de clasificación mostraron resultados que van del 77.27 % hasta un 98.98 % de precisión en la clasificación de plantas de maíz y malezas, y nos permiten determinar que la mejor técnica fue Backpropagation, ya que una precisión del 98.98 % puede ser eficaz en la implementación de tecnologías de detección de plantas de maíz para su fertilización u otros procesos. Si se llevara a cabo su implementación en el proceso de fertilización actual en México, podría hacer la actividad mucho más precisa y eficaz, pudiendo traer consigo beneficios económicos a largo plazo

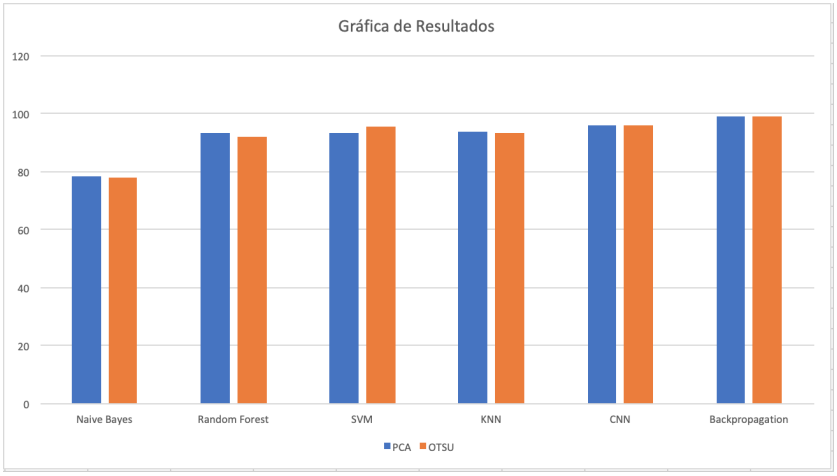


Fig. 4. Gráfica de resultados con los dos métodos de segmentación utilizados.

para este sector productivo, al evitar el desperdicio de fertilizante en la maleza. Finalmente, recalcar que la inteligencia artificial en sus diferentes técnicas debe empezar a utilizarse con mayor frecuencia en todos las áreas con nichos de oportunidad, incluyendo la agricultura. Como trabajo a futuro se realizarán experimentos con caracteristicas geométricas y cromáticas para compararlas con los resultados obtenidos en este artículo y encontrar si existe un mejor porcentaje de efectividad.

Referencias

1. Aakif, A., Khan, M.F.: Automatic classification of plants based on their leaves. Biosystems Engineering 139, pp. 66–75 (2015)
2. Baruffaldi, J.M.: Redes neuronales adversarias para el reconocimiento de malezas. Tesis, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Argentina, pp. 47–61 (2016)
3. Cervantes, J., Taltempa, J., Garca-Lamont, F., Castilla, J.S.R., Rendon, A.Y., Jalili, L.D.: Analisis comparativo de las tecnicas utilizadas en un sistema de reconocimiento de hojas de planta. Revista Iberoamericana de Automatica e Informatica Industrial RIAI 14(1), pp. 104–114 (2017)
4. Cervantes, J., Lamont, F.G., Mazahua, L.R., Hidalgo, A.Z., Castilla, J.S.R.: Complex identication of plants from leaves. In: Intelligent Computing Methodologies,. Springer International Publishing, pp. 376–387 (2018)
5. Chau, A.L., Hernandez, R.R., Mora, V.T., Canales, J.C., Mazahua, L.R., Lamont, F.G.: Detection of compound leaves for plant identication. IEEE Latin America Transactions 15(11), pp. 2185–2190 (2017)
6. Dong, M., Mu, S., Su, T., Sun, W.: Image recognition of peanut leaf diseases based on capsule networks. In: Communications in Computer and Information Science, Springer Singapore, pp. 43–52 (2019)

7. Garcia-Lamont, F., Cervantes, J., Lopez, A., Rodriguez, L.: Segmentation of images by color features: A survey. Neurocomputing 292, pp. 1–27 (2018)
8. Garcia, F., Cervantes, J., Lopez, A., Alvarado, M.: Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets. IEEE Latin America Transactions 14(7), pp. 3434–3443 (2016)
9. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital image processing using MATLAB. Pearson Education India (2004)
10. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics SMC 3(6), pp. 610–621 (1973)
11. Haug, S., Michaels, A., Biber, P. Ostermann, J.: Plant classification system for crop/weed discrimination without segmentation. In: IEEE Winter Conference on Applications of Computer Vision, pp. 1142–1149 (2014)
12. He D.C., Wang L.: Texture Unit, Texture Spectrum, And Texture Analysis. IEEE Transactions on Geoscience and Remote Sensing, 28, pp. 509–512 (1990)
13. Su Hnin Hlaing, Aung Soe Khaing: Weed and crop segmentation and classification using area thresholding. IJRET, pp. 375-382 (2014)
14. Imanov, E., Alzouhbi, A.K.: Machine learning comparative analysis for plant classification. In: 13th International Conference on Theory and Application of Fuzzy Systems and Soft Computing, ICAFS-2018, Springer International Publishing, pp. 586–593 (2018)
15. Jalili, L.D., Morales, A., Cervantes, J., Ruiz-Castilla, J.S.: Improving the performance of leaves identification by features selection with genetic algorithms. In: Communications in Computer and Information Science, Springer International Publishing, pp. 103–114 (2016)
16. Lanlan Wu, Youxian Wen: Weed corn seedling recognition by support vector machine using texture features. African Journal of Agricultural Research 4(9), pp. 840–846 (2009)
17. Krishna, M.: Effective segmentation of green vegetation for resource-constrained real-time applications. Precision agriculture, pp. 257–266 (2015)
18. Niño, D.A., Castilla, J.S.R., Zenteno, M.D.A., Jalili, L.D.: Complex leaves classification with features extractor. In: Intelligent Computing Theories and Application, Springer International Publishing, pp. 758–769 (2019)
19. Pattanayak, S.: Pro Deep Learning with TensorFlow. Bangalore, Karnataka, India (2017)
20. Schor, N., Bechar, A., Ignat, T., Dombrovsky, A., Elad, Y., Berman, S.: Robotic disease detection in greenhouses: Combined detection of powdery mildew and tomato spotted wilt virus. IEEE Robotics and Automation Letters 1(1), pp. 354–360 (2016)
21. Silva, A.R.E., Cobelas, M.A., Gonzalez, E.M.: Impactos del nitrógeno agrícola en los ecosistemas acuaticos. Ecosistemas 26(1), pp. 37–44 (2017)
22. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. Springer US (1993)
23. Jorge Torres. An automatic object-based method for optimal thresholding in UAV images: Application for vegetation detection in herbaceous crops. Computers and Electronics in Agriculture. pp. 43–52 (2015)
24. Feng Wang, X., Wang, Z., Wen Zhang, S.: Segmenting crop disease leaf image by modified fully-convolutional networks. In: Intelligent Computing Theories and Application, Springer International Publishing, pp. 646–652 (2019)
25. Wang, L., He, D.: Texture Classification Using Texture Spectrum. Pattern Recognition, 23(8), pp. 905–910 (1990)

26. Wei Xiangqin: Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. Optik - International Journal for Light and Electron Optics, pp. 5684–5689 (2014)
27. Yang, C.C. Prasher, S.O. Landry, J.A. Ramaswamy, H.S. et al.: Application of artificial neural networks in image recognition and classification of crop and weeds. Can. Agric. Eng. 42, pp. 147–152 (2015)
28. Zhang, S., Wang, X., Wang, Z.: Weed recognition in wheat field based on sparse representation classification. In: Intelligent Computing Theories and Application, Springer International Publishing, pp. 511–519 (2019) https://doi.org/10.1007/978-3-030-26763-6_49
29. Zhao, C., Chan, S.S., Cham, W.K., Chu, L.: Plant identification using leaf shapes – a pattern counting approach. Pattern Recognition 48(10), pp. 3203–3215 (2015)
30. Zheng, Y., Yuan, C.A., Shang, L., Huang, Z.K.: Leaf recognition based on capsule network. In: Intelligent Computing Theories and Application, Springer International Publishing, pp. 320–325 (2019)

ISSN 1870-4069

Análisis comparativo de métodos de desdoblamiento de fase en perfilometría por desplazamiento de fase

Luis Arturo Alvarado Escoto, Jesús Carlos Pedraza Ortega, Juan Manuel Ramos Arreguin, Efren Gorrostieta Hurtado, Marco Antonio Aceves Fernandez

Universidad Autónoma de Querétaro,
Facultad de Ingenierías,
México

{luis.a.alvarado.e, efrengorrostieta, marco.aceves}@gmail.com, {caryoko, jsistdig}@yahoo.com.mx

Resumen. La perfilometría por desplazamiento de fase se encuentra entre las técnicas mas populares de medición óptica debido a sus numerosas ventajas como alta resolución y bajo costo computacional. En perfilometría por desplazamiento de fase una serie de patrones de franjas cuadradas o senoidales son capturados mientras la fase cambia. La combinación de estos patrones desplazados se utiliza para la extracción de fase, la cual contiene la información de altura del objeto. Dado que la fase extraída se encuentra envuelta, los algoritmos de desdoblamiento de fase se utilizan para remover las discontinuidades en fase. El presente trabajo se enfoca en un análisis sobre el desempeño de algunos de los algoritmos de desdoblamiento de fase mas comunes utilizados en perfilometría por desplazamiento de fase. Dada la importancia que tiene el desdoblamiento de la fase se realizo una análisis en desempeño y tiempos de ejecución utilizando tres tipos de objetos: virtuales, reales simétricos y reales con superficies complejas. Como resultados, se encontró que el algoritmo Graph Cuts y el algoritmo Constantini presentan mejores y más rápidos resultados para el desdoblamiento de fase

Palabras clave: Desdoblamiento de fase, proyección de franjas, desplazamiento de fase, perfilometría, reconstrucción 3D.

A Comparative Analysis of Phase Unwrapping Methods in Phase Shifting Profilometry

Abstract. Phase-shifting profilometry is among the most popular techniques in optical measurement due to its numerous advantages like high resolution and low computational cost. In phase-shifting profilometry a series of interferometers are captured while the phase changes. Phase

extraction is used to obtain the height of the object under the interferometers. Given that the extracted phase is wrapped, phase unwrapping algorithms are used to remove the discontinuities caused by phase extraction. This paper focuses on a comparative analysis of some of the most common phase unwrapping algorithms used in PSP. Given the importance of phase unwrapping, an analysis on performance and runtime was made using three types of objects: virtual, real with symmetric dimensions and real objects with complex surfaces. Moreover, we found out that Graph Cuts and Constantini present fast and better performances for phase unwrapping.

Keywords: Phase unwrapping, fringe projection, phase shifting, profilometry, 3D reconstruction.

1. Introducción

El mundo real se visualiza en tres dimensiones. Las cámaras convencionales y los detectores de imágenes únicamente son capaces de obtener la información de intensidad de una escena en dos dimensiones, pero son incapaces de obtener la información de forma y profundidad de un objeto. Al igual que los humanos, que pueden apreciar el mundo tridimensional a través de los ojos, los sistemas de visión inteligentes deben ser capaces de obtener la información tridimensional de una escena a través de imágenes de entrada. Para enfrentar este reto, las tecnologías de medición 3D han desarrollado un sin número de técnicas que permiten obtener de manera cuantitativa la información geométrica en tres dimensiones a partir de imágenes de entrada bidimensionales.

Dentro de las técnicas de extracción de información tridimensional, las técnicas de luz estructurada se encuentran entre las más comunes debido a su alta resolución y bajo costo computacional. Consideradas técnicas de no-contacto, ya que no requieren de un contacto con la superficie bajo estudio, las técnicas de luz estructuradas se basan en dos principios, luz codificada y proyectada, así como en técnicas de patrones de franjas [1]. Los métodos de proyección de franjas consisten en crear un patrón definido de franjas a partir de señales senoidales o periódicas. Estos patrones son proyectados utilizando un proyector digital a la superficie del objeto bajo estudio, creando distorsiones en los patrones debido a la superficie del objeto, las distorsiones son entonces capturadas con una cámara desde un ángulo distinto. Las capturas permiten obtener la información tridimensional del objeto a través de la obtención de la fase [1, 3, 4].

Los métodos por desplazamiento de fase son utilizados ampliamente debido a sus gran cantidad de ventajas, tales como, bajo costo, alta resolución, menor sensibilidad a variaciones en los reflejos de la superficie y menor sensibilidad a la luz ambiente [6]. Los métodos por desplazamiento de fase utilizan al menos tres imágenes de entrada. Cada patrón de franjas proyectado se encuentra desfasado por un valor de $2\pi/N$, donde N representa el número de patrones a utilizar, los métodos más comunes son el *Three-step* y *Four-step*.

En principio, el tener un mayor numero de muestras disminuye el ruido en la intensidad y da pie a tener mayor precisión en las mediciones obtenidas. En perfilometría por desplazamiento de fase, una serie de patrones de franjas son capturados mientras la fase de referencia del patrón de franjas cambia. La fase se encuentra entonces codificada en las variaciones en el patrón de intensidad de las franjas, y un calculo simple punto a punto recupera la fase. La intensidad en cada punto puede varias como una función sinusoidal de la fase desplazada que se introdujo, con un desplazamiento temporal dado por la fase desconocida del frente de onda. El presente trabajo se enfoca en las técnicas de desplazamiento de fase, en particular en el uso del método *three-step* [1–3].

Para obtener la altura de un objeto es necesario extraer la información inmersa dentro de la fase. Para esto, dos pasos básicos son necesarios: la obtención de la fase y el desdoblamiento de la misma. La fase envuelta puede se expresada de manera matemática de la siguiente forma [6]:

$$x_w(n) = W[x(n)], \tag{1}$$

donde $x(n)$ es la fase continua original, $W[\,]$ es el operador de fase envuelta y $x_w(n)$ es la fase envuelta.

El desdoblamiento de fase se realiza para eliminar las discontinuidades generadas por la extracción de fase, la cual debido al principio del valor de la función arco tangente, se encuentra limitada al intervalo $[-\pi \;\; \pi]$. El desdoblamiento de fase se conoce como el proceso de determinar el múltiple entero desconocido de 2π que debe ser agregado a cada píxel del mapa de fase envuelto para hacerlo continuo [3, 5]. En teoría, el desdoblamiento de la fase pareciera ser un calculo trivial para mapas de fase obtenidos de buenos interferómetros. Sin embargo, la presencia de ruido, baja luminosidad, pocas o malas muestras, y las condiciones del objeto bajo estudio, hacen que el desdoblamiento de fase sea un proceso complejo. Debido a esto, existen un sin número de técnicas que permiten obtener distintos resultados en desdoblamiento de fase [6,7]. En este trabajo, se presenta una comparativa en desempeño de algunos de los métodos de desdoblamiento más comunes.

2. Principios del método

Los patrones de franjas utilizados en el método *three-step* pueden ser descritos de la siguiente manera:

$$I_1 = I'(x,y) + I''(x,y)Cos(\phi(x,y)), \tag{2}$$

$$I_2 = I'(x,y) + I''(x,y)Cos(\phi(x,y) + \alpha), \tag{3}$$

$$I_3 = I'(x,y) + I''(x,y)Cos(\phi(x,y) + 2\alpha), \tag{4}$$

donde $I'(x,y)$ es la intensidad promedio, $I''(x,y)$ es la modulación de la intensidad, $\phi(x,y)$ es la fase a resolver y α es el desplazamiento dado en la fase [1,2,4]. Para este caso se utilizó un valor de $\alpha = 2\pi/3$.

Tal y como se muestra en la Fig. 1, los tres patrones de franjas son generados y proyectados sobre la superficie del objeto. La metodología propuesta requiere obtener capturas de los patrones de franjas proyectados sin el objeto, esto permitirá tener una referencia sobre la fase verdadera sin la distorsión generada por el objeto. Generalmente, se utiliza una distancia entre el proyector y la cámara de 18 a 30 cm esto asegura que el ángulo de captura se encuentre alrededor de los 10°.

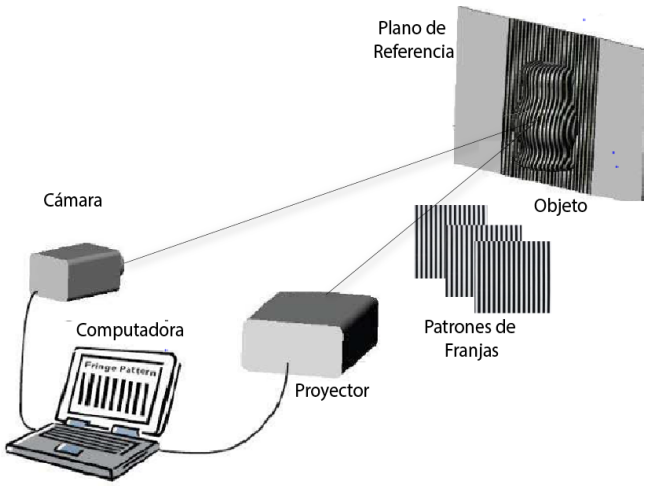


Fig. 1. Sistema utilizado para la proyección y captura de las imágenes.

La fase a resolver esta dada entonces por la ecuación 5 [1], donde $\delta_i = \frac{i2\pi}{N}$ y $i = 1, 2, 3..., N$, mientras que N representa el número de imágenes de entrada, en este caso $N = 3$. Una forma simplificada de para el método *three-step* puede ser descrita con la ecuación 6 [2]. De esta ecuación el mapa de fase puede ser obtenido aplicando el mismo proceso de manera puntual, píxel a píxel. Debido al rango de valores que se obtienen al aplicar la función arco tangente, algunos valores del mapa de fase en encuentran envueltos. Por lo tanto, el siguiente paso debe consistir en aplicar algunos métodos de desdoblamiento de fase:

$$\phi(x,y) = tan^{-1}\left(\frac{-\sum I_i Sen(\delta_i)}{\sum I_i Cos(\delta_i)}\right), \tag{5}$$

$$\phi(x,y) = tan^{-1}\left(\sqrt{3}\frac{I_1(x,y) - I_3(x,y)}{2I_2(x,y) - I_1(x,y) - I_3(x,y)}\right). \tag{6}$$

El proceso completo de perfilometría por desplazamiento de fase se encuentra descrito en la Fig. 2. Además de las capturas de las franjas proyectadas sobre los objetos, se realiza una captura extra de las imágenes de referencia, es decir,

los patrones de franjas. Este plano de referencia pasa por el mismo proceso que las imágenes capturadas conteniendo al objeto bajo estudio. Los tres patrones de franjas descritos por las ecuaciones (2), (3) y (4) son proyectadas sobre el objeto y capturadas al igual que las imágenes de referencia proyectadas. Antes de obtener el mapa de fase de las capturas, se realizó un ajuste de las imágenes a dimensiones similares y se corrigió la perspectiva de estas.

Posteriormente, se realizó una corrección gamma de las imágenes y una ecualización de histograma si es necesario. El mapa de fase del objeto y del plano de referencia son obtenidos utilizando (6). La Fig. 3 muestra un ejemplo de las imágenes obtenidas para el método *three-step* y los mapas de fase obtenidas tanto para el objeto como para el plano de referencia. La fase absoluta envuelta puede ser obtenida obteniendo la diferencia en fase entre el mapa de fase del objeto y el mapa de fase del plano de referencia.

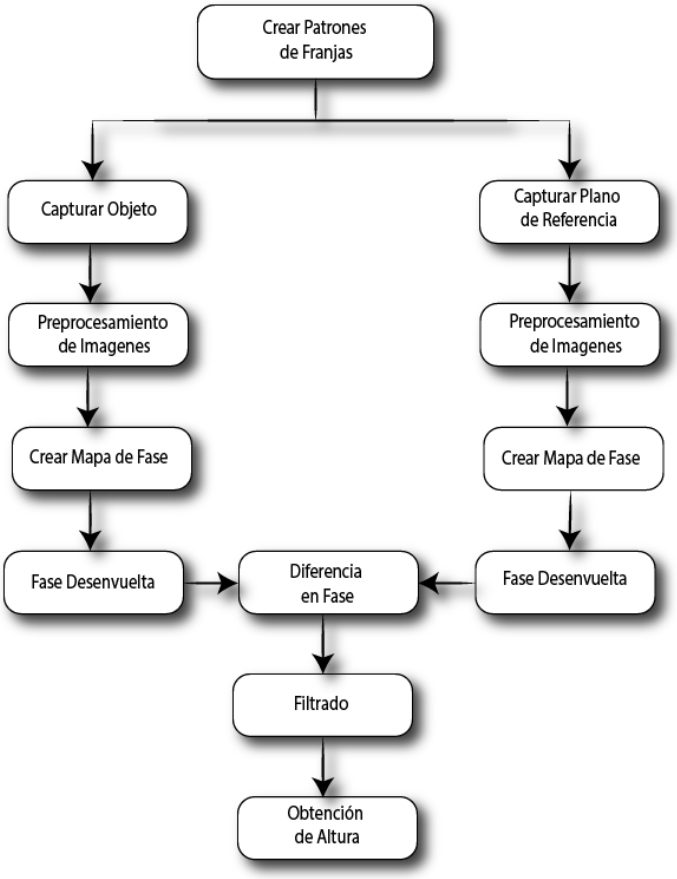


Fig. 2. Metodología utilizada para el proceso de reconstrucción.

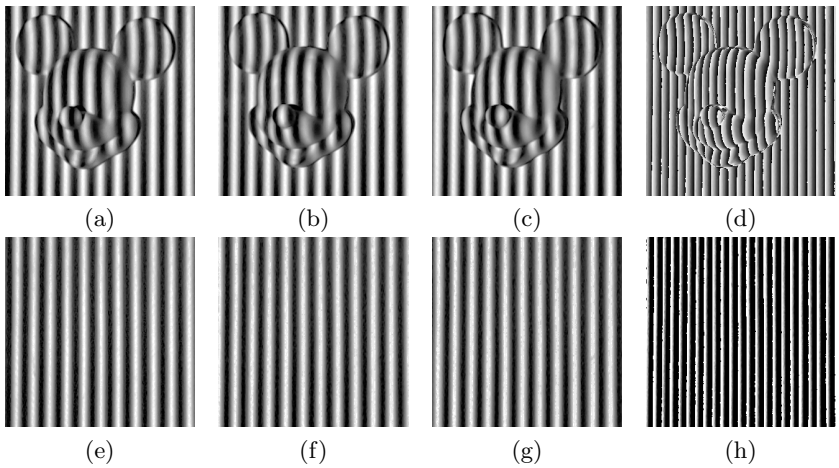


Fig. 3. (a) Paso 1: 0˚. (b) Paso 2: 120˚. (c) Paso 3: 240˚. (d) Mapa de referencia del objeto. (e) Referencia Paso 1: 0˚. (f) Referencia paso 2: 120˚. (g)Referencia paso 3: 240˚. (h) Mapa de fase del plano de referencia.

Algún método de desdoblamiento de fase es aplicado a ambos mapas de fase y finalmente el objeto tridimensional es obtenido a través de la diferencia en fase de ambas fases desenvueltas. De manera opcional, algún filtrado puede ser agregado para eliminar el ruido en la imagen resultante, o para reducir las altas frecuencias generadas por el desdoblamiento de fase. En este trabajo se aplicó únicamente un filtro mediano para reducir estas altas frecuencias.

Finalmente, se obtiene la altura del objeto real multiplicando un factor por la altura obtenida en la reconstrucción. Este factor es obtenido realizando una calibración del sistema con un objeto de dimensión conocida.

Para evaluar el desempeño de los algoritmos de fase se utilizaron más de 30 objetos reales y 10 imágenes de profundidad u objetos virtuales en escala de grises. En el caso de los objetos virtuales no se realizó la captura de imágenes, debido a que las franjas son agregadas de manera simulada al objeto ideal, ningún pre-procesamiento se lleva a cabo debido a que es considerado un proceso ideal. Los objetos virtuales permiten evaluar el desempeño de los algoritmos en condiciones ideales, donde las capturas no presentan problemas de luminosidad o resolución. De igual manera, el error es más sencillo de obtener debido a que se tiene una representación digital del objeto real, lo cual no es el caso de los objetos reales. El desempeño en objetos reales puede ser más complicado de evaluar debido a que las condiciones de luminosidad o captura pueden afectar el desempeño de estas técnicas. Se utilizaron objetos tanto de dimensión conocida, como objetos con una superficie más compleja. Para los objetos de dimensión conocida se evaluó el desempeño de los algoritmos comparando la altura real del objeto con la obtenida del objeto reconstruido en tres dimensiones.

2.1. Métodos de desdoblamiento de fase

El desdoblamiento de fase es un proceso puntual que consiste en localizar las discontinuidades de magnitud cercanas a 2π en cada píxel y agregar o restar este valor de acuerdo al signo de cambio en la fase. Existe una gran cantidad de algoritmos de desdoblamiento de fase, cada uno basado en un principio distinto.

Los algoritmos utilizados en el presente trabajo son mencionados a continuación:

- **Itoh** 2D Este algoritmo corrige los ángulos de fase en un vector agregando múltiples de $\pm 2\pi$ cuando existen saltos absolutos en elementos consecutivos. El algoritmo realiza un desdoblamiento en cada fila, una por una, seguida por un desdoblamiento por columnas, una por una [8].
- **Graph cuts** Desdoblamiento de fase a través de minimización de energía, donde la minimización es alcanzada por una secuencia finita de minimizaciones binarias, cada una resuelta eficientemente por un calculo de max-flujo/min-corte en ciertos grafos [9].
- **Constantini phase unwrapping** Método propuesto que explota el hecho de que las derivadas discretas de la fase desenvuelta son estimadas con un error que es un múltiple entero de 2π. Esto conlleva a que el problema de desdoblamiento de fase sea un problema de minimización con variables enteras. En esta técnica se minimiza la desviación entre las derivadas discretas estimadas y las derivadas desconocida de la fase desenvuelta [10].
- **Quality guided** En esta técnica, la fase envuelta es desenvuelta en partes siguiente un cierto camino dependiendo de la calidad del mapa de fase. El algoritmo divide el mapa completo en n mapas de calidad y realiza el desdoblamiento de estos de manera iterativa. De cada mapa se elige un píxel inicial que tenga la mayor calidad y se agrega a una lista. De esta lista se elige el píxel con mayor calidad y se desenvuelve. Este píxel se elimina de la lista y el proceso se repite nuevamente hasta que la lista quede vacía [11].

3. Resultados

El desempeño de los algoritmos de desdoblamiento de fase discutidos anteriormente fue evaluado utilizando tres diferentes tipos de objetos: objetos virtuales y perfectos, objetos reales de dimensión conocida y objetos reales con superficies complejas. Para todos los experimentos se proyectaron 32 franjas en escala de grises y se obtuvo el mapa de fase utilizando la ecuación 6. El equipo utilizado para los experimentos es una computadora Dell G3 con 16GB de RAM, Intel® Core™ i7-8750GHz y Windows 10 64 bits. El software utilizo para la implementación fue Matlab R2018b. Se utilizó un proyector DLP marca Asus S1 y una cámara web marca Logitech modelo C920 Pro.

Se utilizaron tres métricas para evaluar el desempeño de cada uno de los algoritmos de desdoblamiento de fase. El tiempo de ejecución permitirá evaluar el tiempo de procesamiento requerido por el algoritmo para realizar el desdoblamiento de la fase. Mientras que, *error cuadrático medio* (MSE) y *proporción*

máxima de señal a ruido (PSNR), permitirán evaluar la calidad de la imagen reconstruida con respecto al objeto original. El error cuadrático medio representa la diferencia promedio entre una imagen y una referencia, es decir, a menor MSE, mayor similitud entre ambas imágenes. En este caso, un MSE menor representa un mejor desdoblamiento en fase.

Por el otro lado, PSNR permite evaluar que tanto una imagen degradada o con ruido ha sido reconstruida para coincidir con una imagen de referencia. Por lo tanto, un mayor PSNR indica mayor calidad en la imagen reconstruida resultando en un mejor algoritmo de desdoblamiento de fase.

La Fig. 4 muestra los resultados obtenidos al aplicar los distintos métodos de reconstrucción en un objeto virtual. Aunque a simple vista una diferencia no es notable, la Tabla 1 contiene las métricas obtenidas para cada uno de los métodos de desdoblamiento. En este caso, las franjas fueron agregadas de manera virtual y se utilizó como plano de referencia, las franjas agregadas al objeto virtual. Para los objetos virtuales, las métricas fueron obtenidas realizado una comparativa píxel a píxel entre la reconstrucción obtenida por cada algoritmo y la representación 3D del objeto virtual (ground truth). Para este tipo de objetos el algoritmo Itoh muestra un mejor desempeño ya que además de tener un error menor, el tiempo de ejecución es mucho menor comparado con los de Graph Cuts y Constantini, indicando que en condiciones idoneas, Itoh presenta resultados favorables en menor tiempo. Con objetos virtuales se puede obtener de manera sencilla el error obtenido por el método de desdoblamiento de fase ya que se tiene una representación virtual del objeto real. Sin embargo, debido a que las condiciones de captura son idóneas, la falta de ruido y condiciones reales pueden hacer que las reconstrucciones obtenidas sean casi perfectas, este no es el caso de las pruebas realizadas con objetos reales.

Tabla 1. Métricas obtenidas para el desempeño de los algoritmos de desdoblamiento de fase utilizando imágenes de profundidad virtuales.

Método	MSE	PSNR	Tiempo de ejecución(seg)
Itoh 2D	31248558509e-08	75.0517001178832	0.138235
Graph Cuts	3.12485585093176e-08	75.0517001178832	72.845235
Constantini	3.12485585107644e-08	75.0517001176822	5.9397588

En la Fig. 5 se observan las reconstrucciones obtenidas utilizando un objeto de dimensión conocida. Para objetos reales no se puede realizar una comparación de manera directa entre la imagen reconstruida y el objeto real tal y como se manejó en las imágenes virtuales debido a que no se tiene una representación virtual del objeto. Para esto, es posible seleccionar un perfil de los resultados y examinarlo en 2D. Debido a que el objeto real tiene dimensiones que conocemos, es posible formar un perfil perfecto del objeto real y obtener métricas de error entre estos perfiles, tal y como se muestra en la Tabla 2 y en la Fig. 6. Al igual que en los objetos virtuales, las métricas MSE y PSNR evalúan la calidad

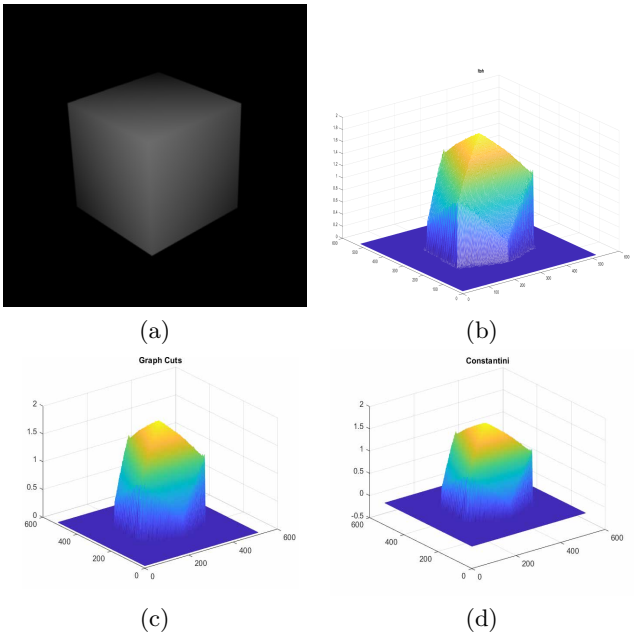


Fig. 4. (a) Imagen Original. (b) Reconstrucción Itoh. (c) Reconstrucción Graph Cuts. (d) Reconstrucción Constantini.

del perfil realizando una comparativa píxel a píxel entre el perfil reconstruido, utilizando PSP y los diferentes métodos de desdoblamiento de fase, y el perfil perfecto del objeto real. En la Tabla 2 se observa que Quality Guided obtuvo un mejor desempeño contra el resto de las técnicas, obteniendo un MSE menor que el resto de los algoritmos y un PSNR de mayor calidad, pero su tiempo de ejecución se eleva muy por encima de los demás. Es importante mencionar que tal y como se observa en la Fig. 5 algunos objetos presentan zonas o regiones con sombras, lo cual produce muchos brincos en la fase que producen valores desconocidos en la reconstrucción.

La Fig. 7 muestra las superficies obtenidas para cada uno de los métodos de desdoblamiento utilizando un objeto con una superficie compleja, es decir, cuyas dimensiones exactas no son conocidas.

Para este tipo de objetos se realizo una evaluación en tiempos de ejecución para cada algoritmo como valoración de su desempeño tal y como se muestra en la Tabla 3. Al igual que en los objetos simétricos, se obtuvo un perfil en dos dimensiones de cada uno de los resultados observado en la Fig. 8. Es importante notar que nuevamente Itoh 2D, aunque es un algoritmo realmente veloz, no presenta buenos resultados en cuanto al desdoblamiento correcto de la fase: no es capaz de identificar los saltos de fase de la imagen envuelta de manera correcta.

En la Tabla 4 se presenta una comparativa entre los resultados obtenidos de cada método para el objeto de superficie compleja. En este caso, se evalúa la

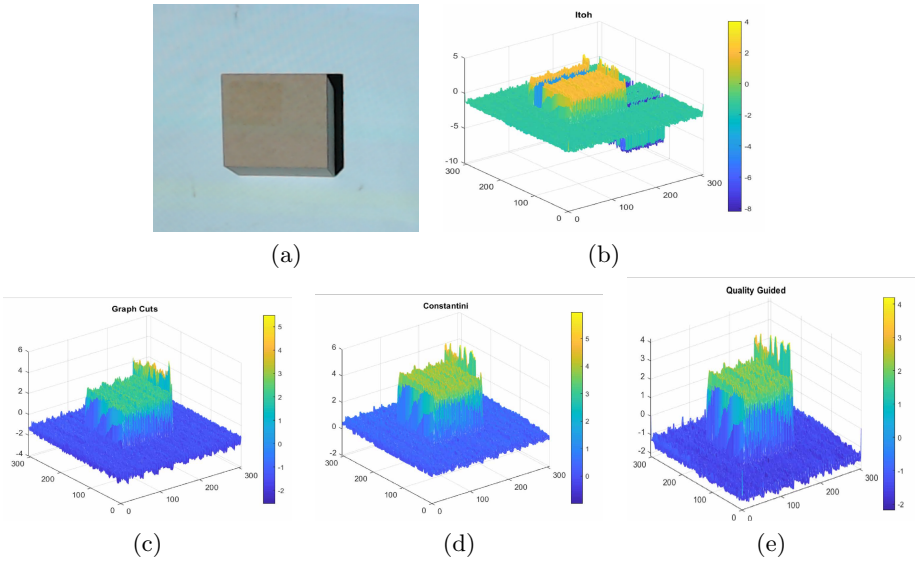


Fig. 5. (a)Objeto real de dimensión conocida. (b)Reconstrucción Itoh. (c) Reconstrucción Graph Cuts. (d) Reconstrucción Constantini. (e) Reconstrucción Quality Guided.

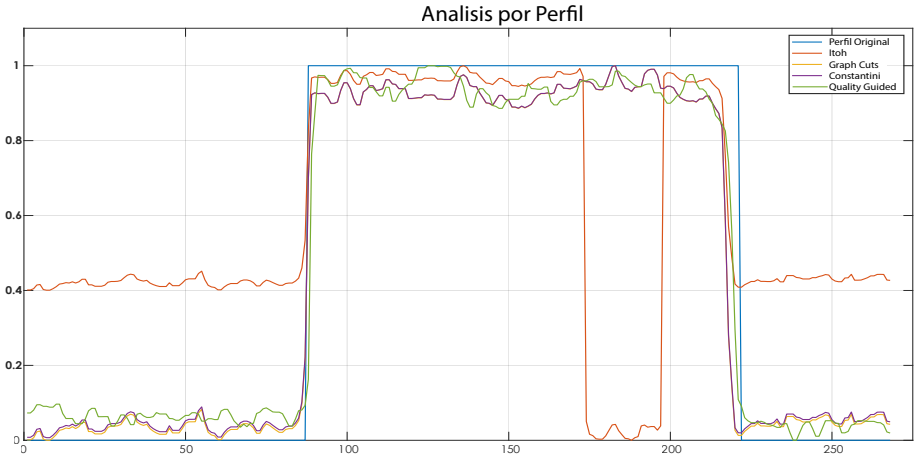
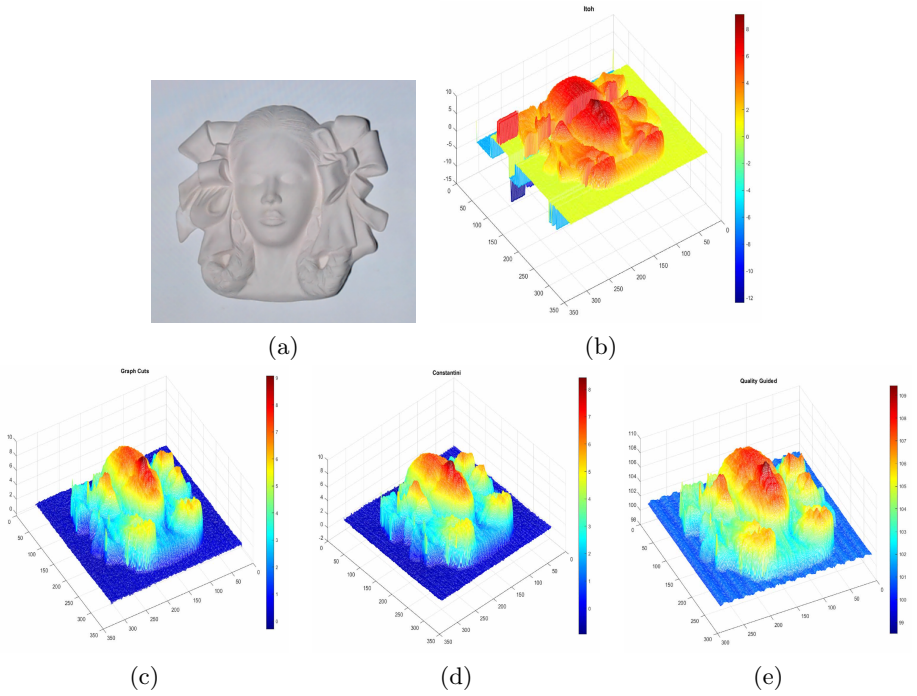


Fig. 6. Análisis por perfil de los resultados de cada método de desdoblamiento utilizando una figura de dimensión conocida.

Tabla 2. Métricas obtenidas para el desempeño de los algoritmos de desdoblamiento de fase utilizando imágenes con dimensión conocida.

Método	MSE	PSNR	Tiempo de ejecución(seg)
Itoh 2D	0.1808	7.4271	0.422698
Graph Cuts	0.0214	16.6868	61.74066
Constantini	0.0214	16.6914	4.971914
Quality Guided	0.0191	17.1974	104.820844



(a) (b) (c) (d) (e)

Fig. 7. (a)Objeto real de superficie compleja. (b)Reconstrucción Itoh. (c) Reconstrucción Graph Cuts. (d) Reconstrucción Constantini. (e) Reconstrucción Quality Guided.

Tabla 3. Tiempos de ejecución promedio para objetos de superficie compleja.

Método	Tiempo de ejecución(seg)
Itoh 2D	0.067956
Graph Cuts	37.43721
Constantini	2.741103
Quality Guided	110.985259

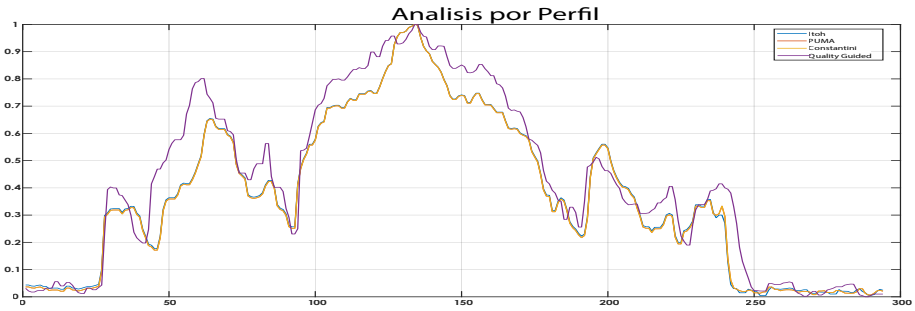


Fig. 8. Análisis por perfil de los resultados de cada método de desdoblamiento utilizando una figura con superficie compleja.

diferencia entre cada algoritmo de reconstrucción. Un menor MSE indica que los algoritmos presentaron resultados similares al realizar el desdoblamiento de fase. Vale la pena mencionar, que el algoritmo Graph Cuts y Constantini presentan diferencias muy bajas entre si obteniendo un MSE relativamente mínimo, lo que indica que ambos algoritmos presentan un desempeño similar, aunque como se vio en tablas anteriores el tiempo de ejecución de Constantini es mucho menor a Graph Cuts.

4. Conclusiones y trabajo futuro

En este trabajo se realizó un estudio de algunos de los métodos más comunes de desdoblamiento de fase y se evaluó su desempeño en términos de efectividad y velocidad.

Las técnicas fueron aplicadas en un proceso de reconstrucción tridimensional utilizando perfilometría por desplazamiento de fase, específicamente el método three-step. El cual consiste en proyectar tres patrones de franjas sobre el objeto y obtener su representación tridimensional virtual a partir de estos patrones.

Se utilizaron tres diferentes tipos de objetos para evaluar el desempeño de los algoritmos: objetos virtuales, objetos reales simétricos y objetos reales con superficies complejas. Los resultados mostraron que en condiciones ideales, todos

Tabla 4. Comparación del error medio cuadrático entre los métodos de desdoblamiento de fase.

Método	MSE
Itoh 2D vs GC	2.88E-05
Itoh 2D vs QG	2.88Ee-05
Itoh 2D vs Constantini	0.0091
GC vs Constantini	1.05E-28
GC vs QG	0.0093
Constantini vs QG	0.0093

los algoritmos son capaces de obtener un buen desempeño, siendo Itoh 2D el algoritmo más veloz. Sin embargo utilizando objetos reales Itoh 2D no es capaz de realizar un buen desdoblamiento debido a los altos saltos de fase que ese encuentran presentes. Graph Cuts y Constantini presentaron resultados similares tanto para los objetos simétricos como para los de superficies no simétricas. Teniendo Constantini un tiempo de ejecución mucho menor a Graph Cuts. El algoritmo Quality Guided aunque presenta resultados favorable en los objetos simétricos tiene un tiempo de ejecución muy elevado en todos los casos.

Como trabajo futuro se espera reducir el error obtenido después del desdoblamiento de fase a través de filtros de suavizado y eliminación de altas frecuencias. Esto con el fin, de agregar mayor suavidad a la imagen resultante y tener una mejor estimación de las dimensiones reales de los objetos. También se trabaja en reducir el ruido producido por las sombras, con el fin de mejorar las reconstrucciones específicamente be las zonas donde se presenta este tipo de ruido. Además, se pretende trabajar con métodos de desdoblamiento de fase basados en técnicas de *machine learning* tal es el caso de K-Means y K-NN.

Referencias

1. Kato, J.: Fringe Analysis. In: Yosizawa, T. (ed), Handbook of optical metrology principles and applications, pp. 541–554, CRC Press, Boca Raton, FL. (2015)
2. Schreiber, H., Bruning, J.: Phase shifting interferometry. In: Malacara, D. (ed), Optical shop testing, pp. 547–667, Wiley Interscience (2007)
3. Gorthi, S., Rastogi P.: Fringe Projection Techniques: Whither we are? Optics and lasers in engineering, 48(2), 133–140 (2010)
4. Garcia, J.: Review in digitization of solid through structured light. Centro de investigaciones en óptica, A.C. (2015)
5. Zuo, C., Feng, S., Huang, L., Tao, T., Yin, W., Chen, Q.: Phase shifting algorithms for fringe projection profilometry: A review. Optics and lasers in engineering (2018)
6. Guerrero, R.E.D., Morales, J.C.M., Arreguin, J.M.R., Soto, J.E.V., Ortega, J.C.P.: Comparative analysis of phase unwrapping in PSP using depth images. International conference in mechatronics, electronics and automative engineering (ICMEAE), Prague, pp. 20–24 (2015)

7. Yuhe, L. Yanxiang, C. Qingxiang, L.: Phase unwraping by k-means clustering in three-dimensional measurement. In: Third international conference on instrumentation, measurement, computer, communication and control, Shenyang, pp. 65–69 (2013)
8. Gdeisat, M., Lilley, F. : Two-dimensional phase unwrapping problem. John Moores University, Liverpool (2012)
9. Bioucas-Dias, J.M., Valadao, G.: Phase unwrapping via max flows. Image processing, IEEE transactions on, pp. 698–709 (2007)
10. Syakrani, N., Mengko, T.L.R., Suksmono, A.B. Baskoro, E.T.: Comparison of PUMA and CUNWRAP to 2-D phase unwrapping. In: Proceeding of the 2011 international conference of electrical engineering and informatics, Bandung, pp. 1–6 (2011)
11. Zhao, M., Huang, L., Zhang Q,. Xianyu, S., Asundi, A., Kemao, Q.: Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies. Applied optics (2011)

ISSN 1870-4069

Sistema multi-agente para el control de tráfico urbano para vehículos de prioridad

Guillermo Fernández Estrella[1], Joel Antonio Trejo Sánchez[2], Mario Renán Moreno Sabido[1]

[1] Tecnológico Nacional de México,
IT de Mérida,
Departamento de Sistemas y Computación,
México
[2] CONACYT-Centro de Investigación en Matemáticas,
México

gfernandezestrella@gmail.com, joel.trejo@cimat.mx,
mario.ms@merida.tecnm.mx

Resumen. El control de tráfico urbano se encarga de controlar el tráfico urbano con el fin de reducir los tiempos de espera de los vehículos involucrados en la red de tráfico. Particularmente, para los vehículos de prioridad, los tiempos de espera son de suma importancia, ya que una mayor espera en dichos vehículos podría tener consecuencias catastróficas. Los sistemas multi-agentes permiten representar el sistema de tráfico urbano y dar capacidades independientes a todas las entidades involucradas en la red de tráfico. En este trabajo se presenta una arquitectura basada en agentes que modela los componentes involucrados en un sistema de tráfico urbano, y se enfoca en reducir el tiempo de espera en los vehículos de prioridad. Experimentalmente, se muestra que la arquitectura propuesta mejora significativamente las estrategias predefinidas en la red de tráfico.

Palabras clave: Control de tráfico urbano, sistemas multi-agentes, vehículos de prioridad.

Multi-Agent Urban Traffic Control for Priority Vehicles

Abstract. Urban traffic control deals with the efficient control of signal traffic lights to reduce the time delays of the involved vehicles in the traffic networks. Particularly, the delay in priority vehicles is very important, because a largest delay could have catastrophic consequences. Multi-Agent systems are a tool to represent an urban traffic system and provide with independent capacities to the involved entities in the urban traffic system. This paper shows a multi-agent system architecture to urban traffic control focus on priority vehicles. Experimentally, this paper shows that the architecture outperforms the predefined strategies in the urban traffic system.

Keywords: Urban traffic control, multi-agent systems, priority vehicles.

1. Introducción

Hoy en día, las ciudades van incrementando su tamaño, lo que genera mayor cantidad de tráfico, principalmente en las grandes ciudades. Está demostrado que la congestión de tráfico urbano produce estrés [1], contaminación [2], repercute directamente en la calidad de vida [3], entre otras cosas.

Con el fin de reducir la congestión de tráfico urbano, existen algunas estrategias que consisten en realizar una asignación predefinida a los semáforos, con el fin de minimizar la espera de los vehículos. Este tipo de estrategia se conoce como control pre-establecido (*pretimed urban traffic control*); por otro lado, aquellas estrategias que buscan asignar dinámicamente los tiempos en los semáforos, se conocen como control dinámico de tráfico urbano [3].

Existen diversas estrategias que pretenden establecer de forma inteligente los tiempos de los semáforos en un sistema de control de tráfico urbano al tomar en consideración la información referente al ambiente en el que está inmerso el sistema de tráfico urbano. El control de tráfico urbano consiste en establecer ciclos, repartos y desfases en una vía o red, de manera tal que los vehículos puedan desplazarse a una cierta velocidad, procurando que las interrupciones generadas por luz roja sean mínimas [4].

Los sistemas multi-agentes están englobados en el campo de la inteligencia artificial, y su propósito es la construcción de sistemas complejos que interactúan entre sí para lograr un objetivo. Los agentes son autónomos y capaces de trabajar en conjunto para la resolución de problemas [5].

2. Trabajos relacionados

Se han propuesto diversas estrategias basadas en agentes para el control de tráfico urbano En [6], Balaji y Srinivasan proponen dos arquitecturas de agentes. Dichas estrategias incluyen políticas de control dinámicas, en la que los agentes se deben adaptar a las condiciones de tráfico. Experimentalmente, los autores muestran que dichas arquitecturas tienen un mejor rendimiento en comparación con otras estrategias, incluyendo estrategias de control de tráfico urbano con tiempos predefinidos.

El dar prioridad a vehículos de prioridad no es nuevo. Existen algunos trabajos que se enfocan en minimizar el tiempo de espera en vehículos de prioridad. En [7] Tlig y Bhouri definen un sistema multi-agentes para dar prioridad a sistemas de transporte público. A diferencia del enfoque que se propone en este trabajo, ellos se enfocan principalmente a vehículos en los cuáles es posible conocer el tiempo aproximado en que pueden pasar por la intersección de tráfico. En el esquema propuesto en esta investigación, los vehículos de prioridad pueden pasar por la red de tráfico en cualquier momento. Recientemente, [8] propone un esquema que da prioridad en la asignación de los tiempos a los sistemas de servicios de transporte urbano. A diferencia de la propuesta que se presenta en este trabajo, en [8], los vehículos de prioridad tienen un calendario estimado del arribo de los vehículos de prioridad en las intersecciones de la red de tráfico urbano.

Este artículo se basa parcialmente en el trabajo realizado por [9 y 10]. Trejo-Sánchez y Fernández-Estrella presentan un controlador de sistemas multi-agentes en la cual se emplea una arquitectura multi-agente cuyo modelo de intersección esté representado por redes de Petri temporizadas, permitiendo analizar un modelo de acuerdo con sus propiedades, tal es el caso del análisis de invariantes, bloqueos, vivacidad, acotamiento, entre otras cosas. Algunos autores han utilizado sistemas de eventos discretos con redes de Petri para el control de tráfico urbano. Los sistemas de eventos discretos [11] son aquellos que están compuestos por elementos que manejan entidades discretas, es decir, numerables y diferenciables entre sí. Su funcionamiento está caracterizado por una sucesión finita o infinita de estados, delimitados por eventos.

Di Febbraro [12] et al. utilizan redes de Petri para modelar la red de tráfico que utiliza una red de Petri híbrida para el control de tráfico urbano. Para una revisión más detallada de cómo se han utilizado las redes de Petri en el control y simulación de tráfico urbano, el lector puede referirse a [13]. Más adelante, cuando se describe la arquitectura propuesta se presenta el módulo encargado del establecer las políticas de control de tráfico urbano. Dicho módulo está basado en sistemas de eventos discretos.

En la actualidad existen controladores de tráfico urbano; entre ellos destaca la plataforma SICE, la cual se especializa en la gestión inteligente de la movilidad, la gestión de la prioridad al transporte público mediante sus equipos RBG, la gestión del *Enforcement* aplicando tecnología tal como sistemas de foto rojo, control de velocidad en travesías, radares por tramo y giros prohibidos [14]. CROSS es una empresa que desarrolla sistemas inteligentes de control de tránsito. Los sistemas incluyen herramientas de despacho para controlar intersecciones, también herramientas para control en coordinación dependiente del tránsito y para control adaptable. Permite la preferencia activa de vehículos de transporte público y crear rutas para el paso de vehículos de servicios de emergencia [15].

SciELO es un controlador de tráfico urbano basado en agentes inteligentes. En su trabajo se presenta el desarrollo de una metodología novedosa que permite incluir un modelo formal basado en agentes autónomos e inteligentes capaces de manipular las fases de los ciclos en una infraestructura de semáforos de acuerdo con las exigencias y limitaciones de la carretera. Este proceso mejora efectiva e inmediata de la calidad del servicio en una intersección, aumentando el rendimiento de la movilidad de los vehículos y mejorando la generación de emisiones, cuando los vehículos se paran en un semáforo rojo [16]. SUMO (Simulation of Urban Mobility) [17] es un simulador de código abierto que permite diseñar una red de tráfico, así como sus diferentes componentes. Dicha herramienta es utilizada por muchos investigadores para poder incluir sus propios algoritmos para el control de semáforos. SUMO ha demostrado permitir la integración de redes de Petri para el control de tráfico urbano [18]. Al ser una herramienta demasiado compleja, se decidió en este trabajo utilizar una herramienta propia que era suficiente para la arquitectura que se propone.

3. Arquitectura basada en agentes

La composición de un sistema es muy importante, ya que permite identificar el propósito de su desarrollo y a su vez la estructura de cada uno de sus componentes con

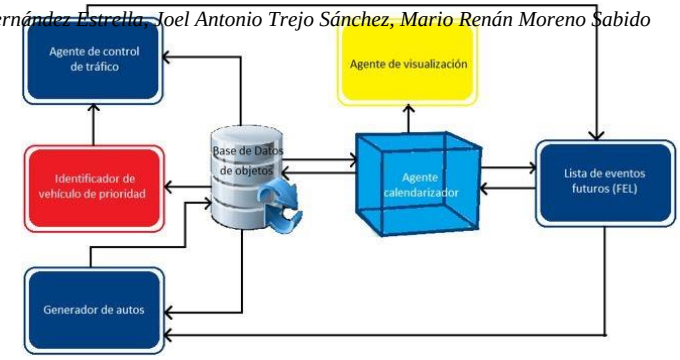


Fig. 1. Arquitectura macroscópica del simulador con las relaciones que existen entre módulos y el orden de las ejecuciones.

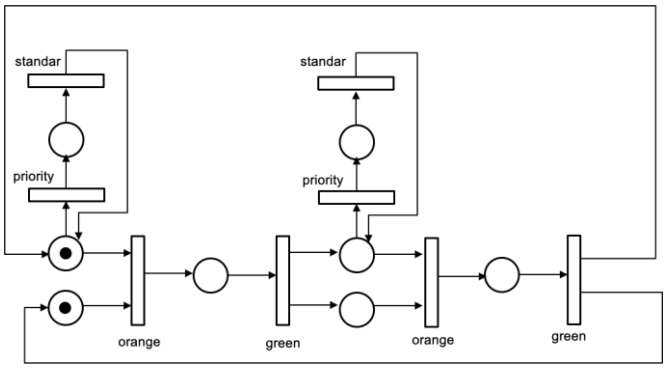


Fig. 2. Red de Petri temporizada.

el fin de justificar por qué son indispensables en el desarrollo del proyecto. La Figura 1 presenta la arquitectura propuesta.

Antes de detallar cada uno de los módulos que componen la arquitectura se presentará la red de Petri temporizada de tres fases empleada para el sistema de eventos discretos. En la figura 2 se describe la lógica del flujo de un vehículo en el simulador; en cada fase existe la posibilidad de que ingrese un vehículo de prioridad influyendo en el flujo de la calle y la coordinación de los semáforos.

A continuación, se describen cada uno de los módulos que componen la arquitectura de agentes.

3.1. Agente de control de tráfico

El agente de control de tráfico coordina las señales individuales de tráfico, por ejemplo, los semáforos. A su vez, un sistema de control tiene como componentes básicos: intersecciones reguladas por semáforos, una red de comunicaciones que permite el envío o intercambio de información, y un computador central que gestiona el funcionamiento de todo el sistema.

Como parte principal del simulador y dado que este módulo lleva por nombre “agente”, éste debe conocer las características que lo rodean.

El agente de control necesita saber el estado en el que se encuentra cada semáforo en cada iteración de tiempo, así como el sentido de la calle, si va de norte a sur, este a oeste, entre otras cosas. La facultad del agente de control de tráfico no solamente repercute en los vehículos que transitan durante la simulación, también determina el permiso del flujo en toda una fila de autos cuando los semáforos cambian y cuando un vehículo de prioridad aparece en la calle.

El agente opera con dos políticas: la política fija toma como referencia valores de tiempo definidos para el cambio de los semáforos en distintas horas del día; en cambio la política adaptativa permite que el agente de control determine una acción, teniendo como valores de entrada el ciclo de cambio de los semáforos, la densidad de vehículos en la calle, el flujo de vehículos en una intersección y la hora del día. Sin embargo, se diseñó para que sea posible integrar nuevas políticas en el futuro.

3.2. Generador de autos

Este módulo conlleva la responsabilidad de la creación e identificación de los vehículos generados; es de suma importancia mencionar que el módulo encargado de generar los autos no puede tener ningún fallo, de lo contrario afectaría a todo el simulador generando un error que termine con la simulación.

También se encarga de generar tanto vehículos de prioridad, como vehículos estándar; éste crea un objeto denominado *"car"* el cual es identificado por un número irrepetible y añadido a la lista de eventos tomando en cuenta el segundo de creación; los vehículos de prioridad se generarán con un identificador para diferenciar de un vehículo estándar. Aparte de solamente generar los autos, este módulo crea de forma aleatoria un vehículo cada segundo, controlando así que la simulación sea de lo más natural a una intersección de vía pública. El movimiento de los vehículos ocurre de forma autónoma ya que el agente generador aplica variables de estado y tiempo para que el vehículo elija si va a moverse o no en cada iteración de simulación.

3.3. Base de datos (objetos)

Es el módulo donde el conjunto de toda la información perteneciente a los objetos, estados y vectores estarán almacenados en la "base de datos de objetos". Aparte de las operaciones básicas que realizará la base de datos como almacenar información, también actualizará los procesos reorganizando los datos generados por el simulador, convirtiendo la base de datos en un proceso dinámico.

3.4. Agente calendarizador

El agente calendarizador es el principal intermediario en el simulador; éste se encarga de obtener información de los vehículos en movimiento como pueden ser su posición que por el momento es asignada por espacios dentro de un carril, su orientación, el estado en el que se encuentra el vehículo, sea en movimiento o detenido. Por otra parte, el agente obtiene la información de la base de datos (FEL) que trabaja en conjunto para poder lograr una actualización real y constante de la simulación; de igual manera, envía la información generada al agente de visualización para su graficación.

La conexión que existe entre los agentes con los módulos de la arquitectura se da por medio de la programación en hilos de Python donde un hilo llamado H se está ejecutando dentro de un método sincronizado y necesita acceso a un recurso llamado R; si éste no se encuentra disponible, H renuncia temporalmente al control del objeto, permitiendo la ejecución de otro hilo hasta que R se encuentre disponible.

3.5. Lista de eventos futuros

Un evento significa un cambio del estado del simulador, por lo tanto, el simulador debe actualizar su contenido. Cada evento tiene asignado un tiempo de ocurrencia como son los semáforos y los vehículos, de tal manera que cuando suceda esto, los eventos generados son inscritos en la lista de eventos futuros (FEL) [9]. El simulador se complementa de la FEL cuando se integra un sistema de control de tráfico urbano.

La FEL almacena la información perteneciente a los objetos generados por el simulador procesándolos e interpretándolos para su lectura por el usuario. Los datos son escorados por el tiempo de creación del evento seguido del tipo de evento y los atributos de cada uno de ellos. El usuario podrá visualizar la FEL en la interfaz de código, pero el simulador también exportará todos los datos almacenados en una tabla ordenada.

3.6. Agente de visualización

Las ventajas de la simulación permiten al usuario detectar el comportamiento de los objetos puestos a prueba; en el simulador se implementó una interfaz gráfica para la proyección de los cambios y movimientos de los vehículos permitiendo una experiencia agradable para el usuario.

El agente de visualización se compone de las calles, los vehículos y los semáforos; éstos son pintados mediante la herramienta Tkinter de Python. Tkinter tiene como ventaja la actualización de la ventana permitiendo ver el movimiento a tiempo real de los vehículos.

3.7. Implementación del sistema multi-agente

Mediante un complejo diseño del simulador microscópico se determinó que es necesario de diversos módulos que sean capaces de tomar decisiones por sí mismo e interactuar entre ellos mismos.

El simulador tiene un componente de tiempo que funciona como un reloj para medir en qué punto se origina cada evento; esto permite a los agentes sincronizarse con base al ciclo de simulación. Todos los vehículos son detectados por el agente de control una vez son sincronizados con la base de datos; esta lectura de datos se da de manera cíclica por cada segundo de iteración de la simulación. En la arquitectura presentada se muestra un módulo el cual se encarga de detectar los vehículos de prioridad; dado que los vehículos son generados por el simulador, la detección de un vehículo de prioridad es mediante un identificador para determinar si el vehículo es de prioridad o es estándar.

El agente de control tiene como prioridad la sincronización de los datos de la lista de eventos futuros y la lectura de los vehículos de prioridad para la elección del nuevo ciclo. Se utilizaron objetos para implementar los agentes.

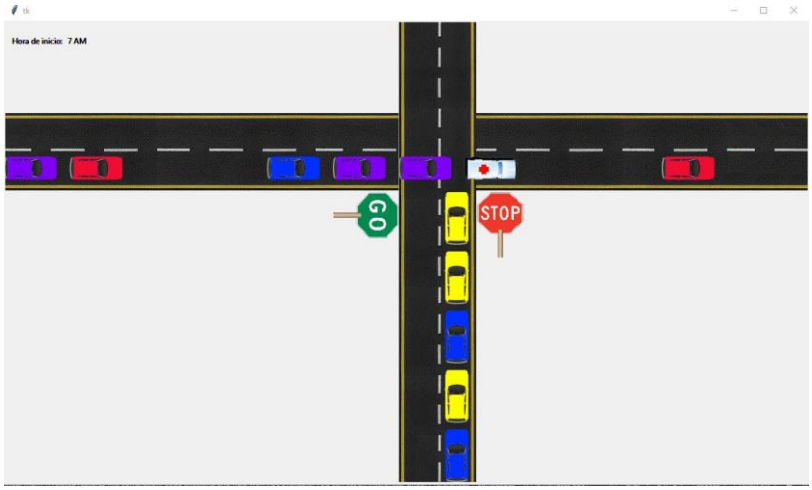


Fig. 3. Interfaz gráfica del simulador.

El agente de control contiene métricas de flujo, densidad y tiempo; también se definieron límites de la calle y conoce el momento de entrada de la calle, así como su salida. Los vehículos tienen variables de tiempo, posición, orientación y el identificador de vehículo de prioridad o estándar.

Los agentes se comunican a través de llamadas directas entre dichos objetos. Actualmente, el sistema que se presenta es centralizado, pero puede extenderse a un ambiente distribuido a través de llamadas a objetos remotos. La comunicación entre estos objetos en la simulación es por medio del *"threading"* que está incluido en una interfaz de alto nivel orientada a objetos para trabajar con concurrencia desde Python, donde los objetos *"Thread"* se ejecutan al mismo tiempo dentro del mismo proceso compartiendo memoria.

En la Figura 3 se presenta la interfaz gráfica del simulador donde se representan los dos tipos de vehículos, la orientación y la hora de la simulación actual. Es importante mencionar que antes de la simulación es posible la modificar la hora inicial y la duración de la simulación; si se requiere el cambio de políticas o los parámetros de generación del vehículo, es necesario modificar el código.

3.8. Experimentación

Para el análisis comparativo de tiempos se emplearon políticas de control que se basan en la hora del día, el número de autos, la duración del semáforo, entre otras cosas.

Con el fin de probar la arquitectura, se realizan varios experimentos utilizando una política adaptativa, y una política fija, sin embargo, por cuestiones de espacio solamente se mencionan los resultados experimentales más relevantes.

La política de control adaptativa da prioridad a los vehículos prioritarios. En esta política, cuando aparecía un vehículo de prioridad, se ajustaban los semáforos y se modificaban los eventos futuros. Para más detalle de esta política referirse a [10].

La política fija funciona con un horario preestablecido que define las fases durante todo el tiempo de la simulación.

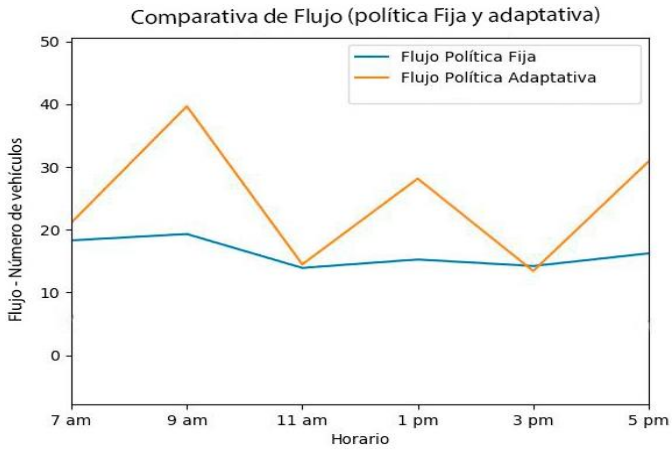


Fig. 4. Comparación del flujo - Política fija y adaptativa.

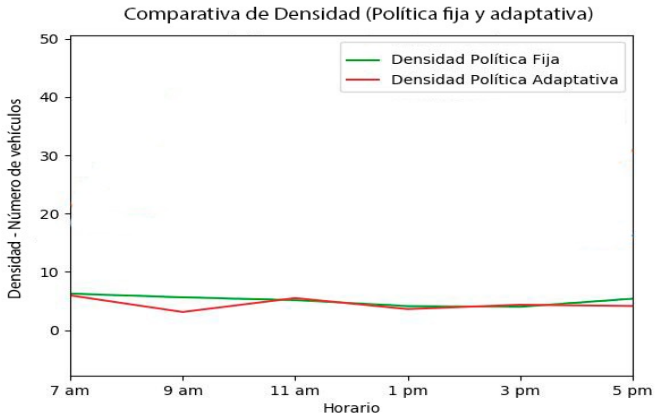


Fig. 5. Comparación de densidad – Política fija y adaptativa.

Comparación de flujo y densidad. Debido a que la política fija tiene definido los parámetros de estado de un semáforo, se vuelve impredecible el comportamiento de variables como el flujo; si en un momento de la simulación existe una congestión vehicular, el flujo mostraría el alto paso de vehículos, sin embargo, no lo haría en un momento óptimo para mantener una estabilidad del flujo en la calle. De igual forma, la densidad se estancaría hasta que los semáforos permitan el paso de los vehículos.

En la Figura 4 se presentan los resultados de la comparación del flujo de la política fija y adaptativa. Al realizar la evaluación del sistema se pudo notar que en ciertas situaciones el sistema de control de tráfico no reduce los tiempos de espera cuando se utilizan políticas fijas tradicionales, por lo tanto, se puede decir que no son convenientes para minimizar la congestión vehicular.

Además, este sistema permite la implementación de políticas de control con el fin de estudiar nuevos métodos para la reducción del tráfico vehicular demostrando las ventajas de este sistema sobre los tradicionales.

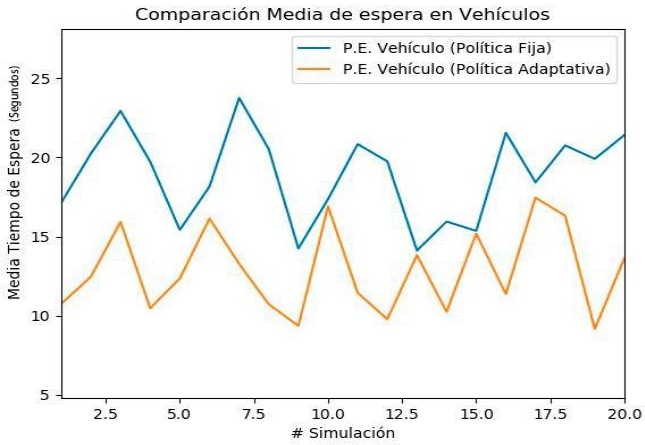


Fig. 6. Comparación media de espera en vehículos.

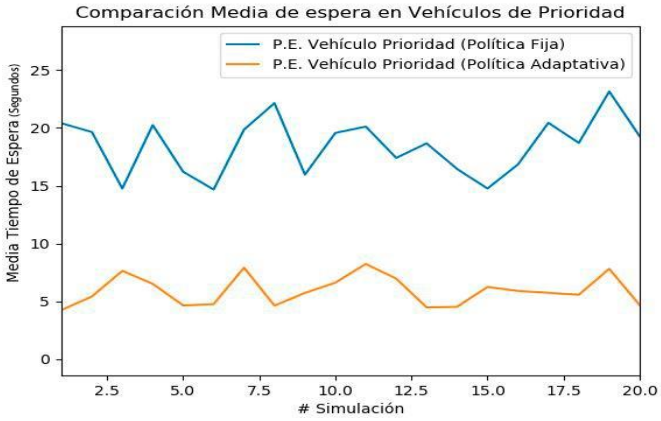


Fig. 7. Comparación media de espera en vehículos de prioridad.

En la Figura 5 se presentan los resultados comparativos de la densidad entre ambas políticas.

Comparación en tiempos de espera. En la gráfica media de espera en vehículos de la política fija se puede observar que el tiempo de espera de los vehículos puede ser muy variante llegando a los 24 segundos, lo cual implica que los vehículos están esperando los ciclos del semáforo completo, y, por lo tanto. puede generarse una congestión vehicular si aumenta la densidad en un sentido de la calle, a comparación de la política adaptativa que el rango de tiempo de espera va desde los 10 segundos, hasta los 16 en promedio (ver Figura 6).

Por otra parte, la media de espera en vehículos de prioridad en la política adaptativa presenta una gran diferencia de tiempo con la política fija, teniendo un rango de espera de entre 4 y 8 segundos en promedio, disminuyendo de 10 a 12 segundos de tiempo en

promedio, solamente con la implementación de la política adaptativa; por lo tanto, se puede decir que la existe una reducción de tiempo de espera notable en los vehículos de prioridad (ver Figura 7).

4. Conclusiones y trabajo a futuro

Con el diseño del sistema de semáforo inteligente se buscó desarrollar un sistema que requiera de poca interacción humana para tomar decisiones y que permita controlar el flujo de tráfico vehicular de una intersección. El sistema permite la detección de vehículos y conteo de éstos; dicha información es almacenada en una memoria, para que posteriormente sea analizada por el agente de control, ya que esto le permitirá tomar decisiones sobre el control del tráfico, y a su vez interactuar con el resto de los módulos para cumplir el ciclo de la simulación.

La implementación de la política de control permite al agente de control elegir la opción óptima que permita el flujo vehicular, dando prioridad a los vehículos designados como importantes en la simulación.

Uno de los elementos primordiales dentro de la gestión del tráfico urbano es el funcionamiento de los semáforos que regulan una intersección vial. El congestionamiento del tráfico representa en la actualidad un gran reto a resolver debido al número de usuarios cada vez mayor que necesitan transportarse para realizar sus actividades económicas, sociales, culturales y de cualquier índole.

El resultado de esta investigación fue la creación de un sistema de control de tráfico urbano, permitiendo analizar en comportamiento del tráfico vehicular durante la simulación gracias a la capacidad de generación, detección y dibujo de vehículos, así como la interacción de los semáforos con el entorno cumpliendo el objetivo principal del presente trabajo, y demostrando en los resultados experimentales la problemática que existe con respecto al tráfico vehicular, y aportando un simulador de eventos discretos con interfaz gráfica que demuestra la reducción de tiempos de espera en vehículos de prioridad mediante políticas de control, con el fin de la experimentación para la contribución a la investigación y mejora del flujo vial.

Se espera que este proyecto de paso a próximas investigaciones relacionadas al tráfico vehicular, dejando como trabajo a futuro en el área la implementación de diferentes políticas de control, el anexo de nuevos módulos al simulador, el incremento en el número de calles simuladas simultáneamente, y la posibilidad de integrar la detección de un vehículo de prioridad a este simulador.

Agradecimientos. Este trabajo fue realizado gracias al apoyo brindado a través de los proyectos FORDECYT 296737 "Consorcio en Inteligencia Artificial" y TecNM 6176.19-P "Análisis de sistemas de flujo urbano para el diseño de ciudades inteligentes".

Referencias

1. Evans, G.W., Carrère, S.: Traffic congestion, perceived control, and psychophysiological stress among urban bus drivers. Journal of Applied Psychology, 76(5), pp. 658 (1991)

2. Rao, A.M., Rao, K.R.: Measuring urban traffic congestion-a review. International Journal for Traffic & Transport Engineering, 2(4) (2012)
3. Roess, R.P., Prassas, E.S., McShane, W.R.: Traffic engineering. Pearson/Prentice Hall (2004)
4. Covenin: Equipos de control de semáforos. Comité de electricidad de Venezuela (1999)
5. Lema, C.S., Pedreira, L.P., Bouza, G., Allende, S.: Estudio de la optimización del tráfico en un cruce a través del ajuste de los ciclos de los semáforos mediante recocido simulado. En: XIX Jornadas ASEPUMA – VII Encuentro Internacional Anales de ASEPUMA (2019)
6. Balaji, P.G., Srinivasan, D.: Multi-agent system in urban traffic signal control. IEEE Computational Intelligence Magazine, 5(4), pp. 43−51 (2010)
7. Tlig, M., Bhouri, N.: A multi-agent system for urban traffic and buses regularity control. Procedia-Social and Behavioral Sciences, 20, pp. 896−905 (2011)
8. Massobrio, R., Nesmachnow, S.: Urban data analysis for the public transportation system of Montevideo. In: Ibero-American Congress on Information Management and Big Data Springer, pp. 199−214 (2019)
9. Trejo-Sánchez, J.: Control de tráfico urbano basado en sistemas multiagentes (2006)
10. Fernández-Estrella, G.: Control de tráfico urbano para vehículos de prioridad (2020)
11. Loborg, P.: Error recovery in automation – an overview. In: AAAI-94 Symposium on Detecting and Resolving Errors in Manufacturing Systems (1994)
12. Di Febbraro, A., Giglio, D., Sacco, N.: Urban traffic control structure based on hybrid Petri nets. IEEE Transactions on intelligent transportation systems, 5(4), pp. 224−237 (2004)
13. Ng, K.M., Reaz, M.B.I., Ali, M.A.M.: A review on the applications of Petri nets in modeling, analysis, and control of urban traffic. IEEE Transactions on Intelligent Transportation Systems, 14(2), pp. 858−870 (2013)
14. SICE: https://www.sice.com/lineas-de-negocio/sistemas-inteligentes-de-trafico/trafico-urbano (2019)
15. CROSS: http://www.cross.cz/es/productos-conteo-y-clasificacion-de-transito (2018)
16. Robles, D., Ñañez, P.: Control y simulación de tráfico urbano en Colombia (2009)
17. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO–simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation (2011)
18. Elidrissi, H.L., Tajer, A., Nait-Sidi-Moh, A., Dakkak, B.: A SUMO-based Simulation for Adaptive Control of Urban Signalized Intersection Using Petri Nets. In: 4th World Conference on Complex Systems (WCCS), pp. 1−6 (2019)

ISSN 1870-4069

Comparación de dos modelos para resolver el problema de implicación textual multilingüe

Darnes Vilariño, Saul Leon Silverio, Beatriz Beltrán

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

{darnes, bbeltran}@cs.buap.mx

Resumen. En el presente trabajo de investigación se presentan dos modelos uno basado en anclas y otro basado en grafos para resolver el problema de implicación textual multilingüe. Los datos con los que se ha trabajado son los ofrecidos en la Conferencia Internacional Semeval 2012 y Semeval 2013. Al probar ambos modelos quedó claro que el modelo basado en grafos supera al modelo basado en anclas, para los 3 corpus utilizados. Se desarrolló una base de conocimientos utilizando ConceptNet5, OpenOffice Thesaurus y WordNet. Esta base de conocimientos se representó mediante un grafo y se diseñó un algoritmo para detectar si dos términos están relacionados dentro del grafo.

Palabras clave: Implicación textual, modelo basado en anclas, modelo basado en grafo, multilingüe.

Comparison of Two Models for Solving the Textual Entailment in Cross Lingual Problem

Abstract. In the present research work, two models are presented, one based on anchors and the other based on graphs, to solve the problem of cross-lingual textual entailment. The data with which we have worked are those offered at the Semeval 2012 and Semeval 2013 International Conference. By testing both models, it was clear that the graph-based model exceeds the anchor-based model, for the 3 corpus used. A knowledge base was developed using ConceptNet5, OpenOffice Thesaurus and WordNet. This knowledge base was represented by a graph and an algorithm was designed to detect if two terms are related within the graph.

Keywords: Textual entailment, model based on anchors, model based on graphs, cross-lingual.

1. Introducción

La vasta cantidad de información disponible en Internet hoy en día supera nuestra capacidad de almacenarla, procesarla y por ende aprovecharla. En los últimos años grandes repositorios de información, como enciclopedias, periódicos, blogs, revistas electrónicas, multimedia y otras fuentes digitales de información han generado contenidos que pueden ser útiles para algún usuario en particular [6,8], sin embargo la disponibilidad de información en varios idiomas, así como la variación léxico-sintáctica del lenguaje suponen un reto a los sistemas de Procesamiento de Lenguaje Natural (PLN) que existen actualmente.

Para explicar los fenómenos de la disponibilidad de información en varios idiomas y la variación léxico-semántica, se analizan los siguientes escenarios:

A. Alger, un usuario alemán, desea buscar información acerca de la cultura Maya de México, hay una gran cantidad de artículos en idioma inglés que tratan de la cultura Maya, no obstante, los artículos que son del interés de Alger se encuentran en español. Alger traduce del idioma alemán al idioma inglés sus inquietudes, pero como se ha notado no haya nada útil y desiste de ser arqueólogo de Chichen Itza. Al igual que Alger, miles de usuarios son limitados por la cantidad de información que encuentran en su idioma.
B. Roberto es un niño que tiene por tarea definir la lluvia, así que busca definiciones de ”Lluvia”, como buen estudiante, busca en más de una fuente para redactar una definición extensa, él se percata de que todas las definiciones denotan lo mismo, pero las palabras, y la forma en las que están escritas son diferentes. Así como Roberto, muchos usuarios buscan información concreta, pero la manera en que está escrita (léxico y sintaxis) impide que sea de fácil acceso. Roberto imagina un Google mejorado que pueda interpretar los textos que hablan de lo mismo, pero con diferentes palabras y escritos de manera diferente.

Estas dos problemáticas habían sido investigadas por separado hasta hace un par de años, ahora se estudian juntas en un problema denominado ”Implicación Textual Multilingüe” (ITML o CLTE por sus siglas en inglés), cuyo fin es solucionar los escenarios dados en A y en B.

En el año 2012, dentro del marco de la conferencia internacional SemEval-2012, se propone a la comunidad científica internacional resolver el problema CLTE[1], éste consiste en determinar si un texto T y una hipótesis H, escritos en diferentes idiomas, se puede inferir el significado de H a partir del significado de T. Formalmente:

Dado un par de fragmentos de texto, tópicamente relacionados, T1 y T2 escritos en diferentes idiomas, la tarea consiste en asignar automáticamente uno de los siguientes juicios de implicación textual:

- Bidirectional: $(T1 \rightarrow T2 \,\&\&\, T2 \rightarrow T1)$ equivalencia semántica.
- Forward: $(T1 \rightarrow T2 \,\&\&\, !T2 \rightarrow T1)$ implicación unidireccional de $T1$ a $T2$.
- Backward: $(!T1 \rightarrow T2 \,\&\&\, T2 \rightarrow T1)$ implicación unidireccional de $T2$ a $T1$.
- No-Entailment: $(!T1 \rightarrow T2 \,\&\&\, !T2 \rightarrow T1)$ sin implicación entre $T1$ y $T2$.

[1] Una extensión multilingüe del problema de Implicación Textual

En esta tarea se asume que tanto T1 y T2 son declaraciones verdaderas (TRUE) y que no existen pares contradictorios [3].

Para desarrollar modelos que solucionan el problema de Implicación Textual Multilingüe, los organizadores ponen a disposición de la comunidad de investigadores a nivel internacional un conjuntos de datos, para entrenar y probar los mismos (train / test). Se ofrecen las siguientes combinaciones de idiomas:

– Español / Inglés,
– Alemán / Inglés,
– Italiano / Inglés,
– Francés / Inglés.

Dotar a los sistemas de un módulo capaz de inferir cuando dos textos expresan lo mismo, enriquece las posibilidades de satisfacer totalmente a la consulta expresada por el usuario. Es importante destacar que esto aún se vuelve más relevante, si la consulta está en un idioma determinado y lo que se está buscando está en otro [4].

Otro resultado implícito que se obtiene con esta investigación, es la posibilidad de desarrollar corpus paralelos de manera no supervisada, que ayude a los sistemas que desarrollan modelos de traducción automática.

Los modelos desarrollados hasta el año 2014 no consideran el juicio de implicación textual contradicción, que en muchas ocasiones es muy difícil de detectar ya que solamente la presencia o ausencia de una palabra hace que dos textos sean totalmente contradictorios. El desarrollo de modelos para resolver este problema, puede ser incorporado a un módulo de procesamiento de lenguaje natural [5].

Detectar el juicio de implicación textual cross lingüe sin el desarrollo de modelos de traducción permite dar solución al problema sin necesidad de disponer de un corpus paralelo totalmente heterogéneo que permita el desarrollo de diccionarios estadísticos de dominio general y dominio particulares.

La mayoría de los modelos desarrollados para resolver este problema son supervisados, lo que obliga a disponer de un corpus de entrenamiento desarrollado por expertos. En muchas ocasiones no es tan fácil disponer de un corpus categorizado, por lo que es importante desarrollar modelos no supervisados, que ofrezcan buen comportamiento independientemente del corpus con el que se está trabajando [10].

El presente trabajo de investigación se encuentra estructurado de la siguiente manera en la sección 2 se explican los modelos que se desarrollaron para realizar las pruebas. En la sección 3 se tiene una breve descripción de las colecciones de datos y de cómo están conformadas dichas colecciones. En la sección 4 se muestran los resultados obtenidos con las pruebas realizadas y finalmente se dan las conclusiones a las que hemos llegado.

2. Modelos desarrollados

Antes de comenzar con la reseña de las 2 metodologías, es necesario definir algunos criterios generales de las mismas. El CLTE asigna juicios de implicación a un par de oraciones en idiomas diferentes, en la Figura 1 se muestran 4 oraciones de la colección español / inglés, cada una etiquetada con su respectivo juicio de implicación.

Ejemplo de CLTE

```
<entailment-corpus languages="spa-eng">

 <pair id="1" entailment="bidirectional">
  <t1>Mozart nació en la ciudad de Salzburgo.</t1>
  <t2>Mozart was born in Salzburg.</t2>
 </pair>

 <pair id="2" entailment="forward">
  <t1>Mozart nació el 27 de enero de 1756 en Salzburgo.</t1>
  <t2>Mozart was born in 1756 in the city of Salzburg.</t2>
 </pair>

 <pair id="3" entailment="backward">
  <t1>Mozart nació en la ciudad de Salzburgo.</t1>
  <t2>Mozart was born on 27th January 1756 in Salzburg.</t2>
 </pair>

 <pair id="4" entailment="no_entailment">
  <t1>Mozart nació el 27 de enero de 1756 en Salzburgo.</t1>
  <t2>Mozart was born to Leopold and Anna Maria Pertl Mozart.</t2>
 </pair>

</entailment-corpus>
```

Fig. 1. Ejemplo de CLTE con idiomas Inglés / Español.

El CLTE ha sido parcialmente resuelto utilizando algoritmos basados en reglas, que toman patrones repetitivos a niveles léxicos y/o sintácticos. Algunas de estas aproximaciones utilizan el conjunto de entrenamiento para poner a punto las reglas que evalúan sobre el conjunto de prueba [2].

Otra vertiente, quizás la más explotada, utiliza algoritmos de aprendizaje automático, que extraen características léxicas y/o sintácticas para descubrir patrones repetitivos en los 4 diferentes juicios de implicación. En este enfoque se construye un modelo para los datos de entrenamiento, que posteriormente permite evaluar los datos de prueba.

De este hecho se pueden distinguir dos perspectivas, la primera toma los cuatro juicios de implicación como clases, es decir que los modelos perciben 4 clases y por ello se nombra a este enfoque multiclase. El segundo punto de vista descompone las oraciones en sus formas más primitivas, reconociendo sólo 2 clases (implicación y no-implicación), esto quiere decir que para obtener el juicio final se realiza una composición de dos clases como exhibe la Tabla 1, este ajuste es denominado como composición.

Tabla 1. Tabla de composición de juicio de implicación.

Juicio de T1 ⇒ T2	**Juicio de T2 ⇒ T1**	**Juicio Final**
Entailment	Entailment	Bidirectional
Entailment	No_entailment	Backward
No_entailment	Entailment	Forward
No_entailment	No_entailment	No_entailment

Independientemente de lo anterior, se tienen dos posibles tratamientos derivados del lenguaje del par de oraciones, el primero busca utilizar métodos multilingüe que rescatan las técnicas de las investigaciones del área de traducción automática (MT, por sus siglas en inglés) [1]. Mientras que el segundo enfoque utiliza un idioma como pivote para hacer que las dos oraciones se encuentren en el mismo idioma, una vez hecho esto, el problema pasa de ser CLTE a implicación (TE) a ello se puede operar con metodologías del TE. Por su parte TE data del año 2005, por lo que utilizar el pivote supone una ventaja en comparación al CLTE.

A continuación, se explica con detalle cada una de las técnicas desarrolladas.

2.1. Modelo de inferencia basado en anclas

Si existe una gran cantidad de información compartida, quiere decir que las oraciones comunican la misma idea. Esta asunción no es del todo correcta, ya que puede presentarse el caso de que dos oraciones que compartan una gran cantidad de información no transmitan la misma idea, por ejemplo el par de oraciones: ”La becerra de Manuel está en la calle” y ”Esta es la calle de Manuel Becerra” es un par equivalente a nivel de tokens, pero al momento de interpretar ambas oraciones, se descubre que hablan de cosas completamente diferentes.

Ante esta nueva evidencia, es necesario proponer un nuevo modelo, el cual sea capaz de detectar los tokens que comparten un par de oraciones, y descubrir si estos tokens se relacionan de manera similar en ambas oraciones. Como preprocesamiento sólo se lematizan las oraciones.

Términos Anclas Consideremos el par de plantillas ”X es obra de Y” y ”X escribió Y”, veamos que en ambas se repiten las variables X y Y, estas variables pueden ser sustituidas por cualquier par de tokens, como por ejemplo Tokio Blues es obra de Murakami”, ”Murakami escribió Tokio Blues”. Ahora bien, estos tokens son denominados anclas, ya que, al estar presentes en ambas oraciones, nos permiten detectar la posición de los elementos en cada oración, de esta manera se pueden analizar cómo es que se relaciona una ancla con la otra dentro de cada par de oraciones. Siguiendo la metodología del empatamiento de plantillas, una vez detectada la posición de las anclas se recurre a buscar un conjunto de plantillas que contenga el texto que está entre las anclas, si se encuentra un par de plantillas que satisfacen las posiciones de las anclas, así como su relación (el texto que hay entre ellas), se dice que hay implicación textual.

Ahora pensemos de manera inversa. A partir de una oración se buscan sus tokens anclas, si estos distan entre sí, a no más de dos tokens en cada oración, quiere decir que se encuentran relativamente cerca, y sospechamos que sostienen alguna relación. Entonces, si en lugar de buscar la relación en plantillas, pudiéramos inferir qué similitud existe entre ambas relaciones, estaríamos descubriendo la implicación textual sin el uso de plantillas, pero con el mismo principio.

En el siguiente apartado se explica cómo se puede llevar a cabo el proceso de inferencia.

Proceso de inferencia La inferencia de dos relaciones, se puede abordar con diferentes técnicas, en los siguientes incisos se describen cada una de ellas:

- Relaciones Directas: Las relaciones directas corresponden a sinonimia, hiperonimia o hiponimia. Son relaciones tan obvias que basta con sustituir la relación con su sinónimo para detectar si son relaciones iguales.
- Similitud Semántica: Estas relaciones son afines, no necesariamente son sinónimos, pero son empleadas bajo contextos similares. Al tomar dos relaciones y detectar una relación semántica, entonces decimos que son iguales.

Para detectar el juicio de implicación que sostienen dos oraciones, se recurre al concepto de eliminación de información común, pero a diferencia de la eliminación de tokens, ahora se eliminarán relaciones de términos anclas, esta nueva variante garantiza que la información eliminada en ambas oraciones corresponde a un nivel interpretativo, más allá del léxico.

Se plantea el algoritmo de la siguiente manera, primero se obtienen todos los términos anclas, que por simplicidad serán los n-gramas más largos presentes en ambas oraciones, posteriormente se tokenizan ambas oraciones y se verifica la distancia a la que se encuentran las anclas, si los términos anclas no exceden un máximo de dos tokens entre ellos en ambas oraciones, se verifican las relaciones que sostienen en una y en otra. Si las relaciones son similares, se eliminan las relaciones y las anclas involucradas en cada oración, al final se asigna el juicio de implicación, en dependencia del porcentaje de eliminación.

2.2. Modelo de interpretación de oraciones basado en grafos

En general se han propuesto algunas metodologías que intentan resolver el problema de la implicación textual a través de propuestas supervisadas y no supervisadas [11]. Recordando que en un lenguaje natural, las oraciones tienen tres niveles de composición, léxico, sintáctico y semántico. En cada nivel se evalúan aspectos diferentes, al descomponer una oración en sus tres niveles, se puede apreciar ciertas igualdades y diferencias.

Por ejemplo, si se toman dos oraciones que comparten poco léxico, que una oración se encuentra escrita en voz pasiva y la otra en voz activa, y que en cuestión semántica difieren; para un modelo computacional que se basa en similitudes, este tipo de oraciones es claramente un no_entailment, y sin embargo puede ser bidirectional. El humano es capaz de interpretar el contenido de cada oración, y gracias a su sentido común, es capaz de relacionar si las ideas son las mismas, de esta manera oraciones como ”Un hombre sostiene un aparato electrónico” y ”Una cámara digital es aguantada por un transeúnte” son oraciones que reflejan la misma idea.

Hasta el desarrollo de esta investigación, no se ha encontrado reportado en la bibliografía ningún sistema que resuelva TE y mucho menos CLTE a través de una interpretación de oraciones, por ello se considera de suma importancia proponer un modelo no supervisado que interprete las oraciones para identificar qué entidades se encuentran en juego y determinar las relaciones que sostienen en cada oración, posteriormente se debe detectar qué entidades son equivalentes y verificar si sus relaciones también son equivalentes, para así determinar si son la misma idea.

El proceso de interpretación de oraciones que se presenta en esta metodología, tiene su origen en las investigaciones de Preguntas y Respuestas [9]. Dentro de los sistemas de preguntas y respuestas se encuentran sistemas que procesan preguntas factuales y preguntas complejas. Las preguntas factuales, corresponden a información verdadera que se encuentra escrita textualmente en algún lugar, mientras que las preguntas complejas, son preguntas que necesitan valerse de información extra para poder ser respondidas, por ejemplo, la pregunta ”¿Dónde nació Albert Einstein?” es una pregunta factual, y la pregunta ”¿Qué medicamento le puedo dar a un paciente enfermo de gripe si es alérgico al paracetamol?” es una pregunta compleja.

Como en el descubrimiento de la implicación textual involucra un par de oraciones, es posible ver una oración como la fuente de conocimiento y la otra como las preguntas que se le hacen a un sistema de preguntas factuales. En un sistema de preguntas factuales se generan árboles sintácticos de la información que se desea indexar, esta estructura permite generar estructuras atómicas del tipo Sujeto-Verbo-Objeto, o lo que es lo mismo Entidad_1-Accion-Entidad_2, una vez generadas estas tripletas son agregadas al índice, cuando una pregunta es lanzada al sistema, este identifica las entidades y acciones presentes en la pregunta y lanza un buscador con operador OR sobre su índice de tripletas, de modo que todas las tripletas recuperadas son respuestas de la pregunta, por ejemplo consideremos el siguiente ejemplo:

Sean los siguientes incisos, el índice de tripletas de un sistema de preguntas y respuestas:

1. (”Los lobos”, ”comen”, ”conejos”),
2. (”Las tortugas”, ”comen”, ”ranas”),
3. (”Las tortugas”, ”viven”, ”el mar”).

Cuando se le pregunta al sistema ”¿Dónde viven las tortugas?”, este identifica los términos ”viven” y ”tortugas”, los cuales son buscados en sus índices y encuentra que el índice 3 contiene ambos términos, finalmente el sistema asume que ”el mar” es la respuesta de la pregunta. Esta investigación retoma este mecanismo de validación de información para interpretar las oraciones.

Este modelo es una aproximación que emplea parsers sintácticos para interpretar las oraciones, después con ayuda de una serie de reglas se extraen hechos de cada oración, estos hechos se utilizan en la confección de grafos. Cada nodo de un grafo generado corresponde a términos de la oración y las aristas son las relaciones entre los términos. Una vez que se construyen los grafos de cada oración, se aplica un algoritmo que busca empatar subestructuras de los mismos, las cuales denominamos hechos-extendidos, que, con ayuda de una fuente de conocimiento, se puede determinar si los nodos son equivalentes o si tendrían alguna relación. Finalmente, la cantidad de subestructuras compartidas, con respecto a la cantidad de subestructuras presente en cada oración nos proporciona el juicio de implicación textual.

A continuación, se exponen cada una de las etapas involucradas en la confección de este modelo.

Extracción de hechos Para entender que es un hecho, se considera la oración ”El lobo feroz derribó la casa de los cerditos, se comió al cazador y huyó a las Vegas”, a partir

de ella podemos extraer la siguiente información: el lobo es feroz, el lobo derribó una casa, la casa es de los cerditos, el lobo se comió al cazador, el lobo huyó a las Vegas. Toda esta información obtenida es denominada hechos, y se define como información verdadera generada a partir de una oración.

El proceso de extracción de hechos consiste en tomar una oración y generar toda la información relevante a partir de ella, para realizar esta tarea se emplea el parser sintáctico de Stanford[2], quien genera el árbol sintáctico. A partir del árbol sintáctico se visita cada nodo para poder generar nuevos nodos, a continuación, se listan los nombres de los nuevos nodos:

- **ENTIDAD**: Son todos los términos de los nodos hijos que contengan algún sustantivo, se recuperan todos los términos asociados a las etiquetas de PoS del tipo: NN, NNS, NNP, NNPS y NAC.
- **ACTIVIDAD**: Son todos los términos de los nodos hijos que contengan algún verbo, se recuperan todos los términos asociados a las etiquetas de PoS del tipo: VB, VBG, VBZ, VBP, VBN, VBD y MD.
- **CALIDAD**: Son todos los términos de los nodos hijos que contengan algún adjetivo calificativo, se recuperan todos los términos asociados a las etiquetas de PoS del tipo: JJ y JJS.
- **PREPOSICIÓN**: Son todos los términos de los nodos hijos que contengan alguna proposición, se recuperan todos los términos asociados a las etiquetas de PoS del tipo: IN y TO.

Los nombres de los nodos reflejan los roles de la información recuperada para relacionar la información recabada, se buscan los siguientes hechos:

- **Sujeto**: Este hecho es obtenido cuando una entidad está asociada a una, o más, actividades. Se asume que la ENTIDAD es el sujeto de la ACTIVIDAD.
- **Objeto**: Este hecho es obtenido cuando una actividad está asociada a una, o más, entidades. Se dice que la ENTIDAD es el objeto de la ACTIVIDAD.
- **Califica**: Este hecho es obtenido cuando se detecta una cualidad asociada a una entidad, entonces se dice que esa CALIDAD califica a la ENTIDAD.
- **Extensión**: Este hecho es obtenido cuando una ENTIDAD está asociada a una PREPOSICIÓN.
- **Complemento**: Este hecho es obtenido cuando una PROPOSICIÓN está asociada a una ENTIDAD.

Estas reglas son extraídas al recorrer el árbol sintáctico, que contienen las nuevas etiquetas, y en cada nodo se aplican las reglas de la Tabla 2 para extraer los hechos.

Interpretación de hechos sobre grafos Se ha dicho que la estructura empleada por los sistemas de preguntas y respuestas, para representar la información indizada, consiste en una tripleta de la forma Entidad _1-Actividad-Entidad_2. Ahora bien, bajo el contexto de los hechos se aprecia que corresponde a un par de hechos: sujeto(Entidad_1,

[2] https://nlp.stanford.edu/software/lex-parser.shtml

Tabla 2. Reglas para la extracción de hechos.

Regla	Hecho
$X \Rightarrow$ ENTIDAD $\&$ $X \Rightarrow$ ACTIVIDAD	sujeto (ENTIDAD, ACTIVIDAD)
ACTIVIDAD $\Rightarrow$ ENTIDAD	objeto (ACTIVIDAD, ENTIDAD)
$X \Rightarrow$ ENTIDAD $\&$ $X \Rightarrow$ CALIDAD	califica (CALIDAD, ENTIDAD)
ENTIDAD $\Rightarrow$ PREPOSICIÓN	extensión (ENTIDAD, PREPOSICIÓN)
PREPOSICIÓN $\Rightarrow$ ENTIDAD	complemento (PREPOSICIÓN, ENTIDAD)

Actividad) & objeto(Actividad, Entidad_2), de modo que es necesario representar los hechos a través de una estructura que permita preservar tanto la información de los hechos como las relaciones entre ellos. Por ello se recurre a una estructura de grafo.

Al representar los hechos en un grafo surge el concepto de hechos ampliados, un hecho ampliado es la manera en la que se relacionan los términos de alguna oración, bajo el contexto de preguntas y respuestas corresponde a una tripleta, y para el contexto de grafos corresponden a subestructuras del grafo.

Tabla 3. Subestructuras del grafo con sus equivalencias en tripletas.

Subestructura	Tripleta
sujeto(Entidad_1, Actividad) & objeto(Actividad, Entidad_2)	(Entidad_1, Actividad, Entidad_2)
califica(Entidad, Calidad)	(Entidad, califica, Calidad)
extensión(Entidad_1, Preposición) & complemento(Preposición, Entidad_2)	(Entidad_1, Preposición, Entidad_2)

Para representar los hechos a través de grafos, basta con tomar a los textos y relacionarlos mediante sus hechos.

Empatamiento de grafos para el descubrimiento de la implicación textual Dentro de la teoría de grafos, el mecanismo que denota que dos grafos tienen la misma estructura es denominado isomorfismo de grafos, que es una metodología rígida que da mayor peso a la forma del grafo que a su contenido [7]. Por otro lado, en algunas ocasiones los grafos que se desean empatar varían en el número de nodos, y no es posible realizar un estudio de isomorfismo, en estos casos lo ideal sería aplicar un empatamiento de hipergrafos. Los hipergrafos son grafos que en cada nodo puede contener otro grafo, desafortunadamente esta última propiedad provoca que el empatamiento de grafos sea un problema NP-Completo.

Como empatar el grafo de forma matemática no es una opción viable, se propone un algoritmo ávido que recorre cada grafo para generar subestructuras, llamadas hechos-extendidos, que al final estas son validadas con otro conjunto de subestructuras. Para validar una subestructura contra otra es necesario decidir si dos segmentos de textos son equivalentes, llegados a este punto siempre se ha hablado de sinonimia, hiperonimia, hiponimia y similitud semántica.

Sin embargo, para este modelo se propone el uso de una fuente de información que busca capturar el sentido común del mundo real en una base de conocimientos, esta base es llamada Conceptnet[3], adicionalmente se utiliza el tesauro de OpenOffice y WordNet, para conformar una base de conocimiento lo más sólida posible.

El empatamiento de subestructuras, que ahora se denomina empatamiento de hechos extendidos, consiste en visitar cada nodo de ambos grafos, generar una subestructura, y verificar si son equivalentes, para ello se emplea la base de conocimientos, se debe asegurar que los hechos extendidos, que se están analizando, posean la misma información para decir que son hechos equivalentes.

3. Colecciones de datos y análisis de resultados

En esta sección se presentan los resultados alcanzados por ambos modelos. Con dos colecciones extras (RTE1 y SICK) que ayudarán al análisis de resultados de las metodologías desarrolladas.

A pesar de que los modelos desarrollados en esta investigación se basan en dos conjuntos de colecciones, es posible extrapolar las metodologías propuestas a otros conjuntos de datos, esto gracias a que los enfoques de las propuestas toman inspiración de modelos que tratan de resolver TE.

Para probar el desempeño de los enfoques expuestos en la secciones anteriores, se han considerado 3 colecciones de datos:

3.1. Corpus CLTE

La colección de datos del CLTE fue liberada a la comunidad científica en el foro internacional de evaluación semántica SemEval, una primera parte en su edición 2012, y una segunda en su edición 2013.

Como las colecciones del CLTE constan de 4 idiomas, se ha decidido solo reportar la subcolección de Inglés/Español, ya que es la colección en la que mejor se comportan todos los algoritmos del estado del arte. La distribución de esta colección puede apreciarse en la Tabla 4.

Tabla 4. Información de la subcolección Inglés/Español del Corpus CLTE.

Conjunto	Pares de oraciones	Juicio Bi-directional	Juicio Backward	Juicio Forward	Juicio No_Entailment
CLTE12-Train	500	125	125	125	125
CLTE12-Test	500	125	125	125	125
CLTE13-Test	500	125	125	125	1225

[3] http://conceptnet5.media.mit.edu

3.2. Corpus RTE1

La colección de datos del RTE1 (Recognising Textual Entailment 1 [2]) fue liberada en el marco del Pattern Analysis, Statistical Modelling and Computational Learning (SemEval), donde se utilizó por primera vez el término Textual Entailment. El corpus RTE1 consta de 2 clases, las cuales se encuentran balanceada. La Tabla 5 muestra la distribución de las colecciones que conforman al corpus RTE1.

Tabla 5. Información del Corpus RTE1.

Conjunto	Pares de Oraciones	Pares de oraciones con Juicio Entailment	Pares de oraciones con Juicio No-Entailment
RTE1-Train1	287	143	144
RTE1-Train2	280	140	140
RTE1-Test	800	400	400

A continuación, se exponen los resultados alcanzados por cada una de las metodologías propuestas en cada uno de los corpus.

4. Resultados

4.1. Resultados de modelo de inferencia basado en anclas

Es importante comentar que para este modelo el tokenizador que se use es determinante, ya que las anclas cambian totalmente, para todos los corpus el tokenizador que permitió detectar las mejores anclas fue el linguistic, este modelo para todos los corpus, supera los resultados obtenidos por los dos modelos que utilizan el principio de eliminación.

4.2. Resultados del modelo de interpretación de oraciones basados en grafos

Para este modelo solo se maneja una sola configuración, las tablas que a continuación se presentan, corresponden a los resultados obtenidos.

Tabla 6. Información de la subcolección Inglés/Español del Corpus CLTE.

Conjunto	Pares de oraciones	Juicio Bi-directional	Juicio Backward	Juicio Forward	Juicio No_Entailment
CLTE12-Train	500	125	125	125	125
CLTE12-Test	500	125	125	125	125
CLTE13-Test	500	125	125	125	125

La colección de datos del RTE1 (Recognising Textual Entailment 1 [2]) fue liberada en el marco del Pattern Analysis, Statistical Modelling and Computational Learning

(SemEval), donde se utilizó por primera vez el término Textual Entailment. El corpus RTE1 consta de 2 clases, las cuales se encuentran balanceada. La Tabla 5 muestra la distribución de las colecciones que conforman al corpus RTE1.

Tabla 7. Resultados del Modelo de Interpretación de Oraciones con el Corpus CLTE.

Conjunto	Precisión de Bidirectional	Precisión de Backward	Precisión de Forward	Precisión de NoEntailment	Precisión Global
CLTE12-Test	0.336	0.168	0.136	0.728	0.342
CLTE13-Test	0.200	0.096	0.080	0.768	0.286

Tabla 8. Resultados del Modelo de Interpretación de Oraciones con el Corpus RTE1.

Conjunto	Precisión de Entailment	Precisión de NoEntailment	Precisión Global
RTE1-Test	0.380	0.667	0.523

En general para los dos corpus reportados el comportamiento ha sido similar. Debido a que la metodología es muy novedosa, quizá haga falta realizar un estudio posterior, para detectar que debilidades pudiera presentar, ya que no logra empatar de manera correcta los grafos asociados a cada uno de los textos. A continuación, se realiza una comparativa entre todos los modelos desarrollados, presentando el mejor comportamiento de cada uno con respecto a cada colección de datos procesada.

5. Conclusiones

Se desarrollaron 163,622 experimentos diferentes, variando las medidas de similitud, el tokenizador, los modelos, el tipo de corpus, los umbrales para detectar la similitud de tokens y los umbrales empleados por el proceso de eliminación, lo que nos ha permitido llegar a las siguientes conclusiones:

1. Las características del corpus influyen completamente en los resultados de cada uno de los modelos.
2. El modelo basado en anclas está también directamente relacionado al proceso de tokenización, porque de este depende la selección de las anclas de cada parte de los textos y permite la extracción de patrones de manera automática.
3. El modelo de grafos puede utilizarse como una herramienta para extraer información.
4. Se desarrolló una base de conocimientos utilizando ConceptNet5, OpenOffice Thesaurus y WordNet. Esta base de conocimientos se representó mediante un grafo y se diseñó un algoritmo para detectar si dos términos están relacionados dentro del grafo.

Referencias

1. Banerjee, S., Lavie, A.: METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. pp. 65–72. Association for Computational Linguistics, Ann Arbor, Michigan (Jun 2005)
2. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment. pp. 177–190. MLCW'05, Springer-Verlag, Berlin, Heidelberg (2005)
3. Esplà-Gomis, M., Sánchez-Martínez, F., Forcada, M.L.: UAlacant: using online machine translation for cross-lingual textual entailment (2012-06)
4. Jimenez, S., Becerra, C., Gelbukh, A.: Soft cardinality + ML: Learning adaptive similarity functions for cross-lingual textual entailment. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). pp. 684–688. Association for Computational Linguistics, Montréal, Canada (7-8 Jun 2012)
5. Mehdad, Y., Negri, M., C. de Souza, J.G.: FBK: Cross-lingual textual entailment without translation. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). pp. 701–705. Association for Computational Linguistics, Montréal, Canada (7-8 Jun 2012)
6. Och, F.J., Ney, H.: Improved Statistical Alignment Models. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. pp. 440–447. ACL '00, Association for Computational Linguistics, USA (2000)
7. Pazienza, M.T., Pennacchiotti, M.: Textual Entailment as Syntactic Graph Distance: a Rule Based and a SVM Based Approach. In: In Proceedings PASCAL RTE challenge. pp. 528–535 (2005)
8. Turchi, M., Negri, M.: ALTN: Word alignment features for cross-lingual textual entailment. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). pp. 128–132. Association for Computational Linguistics, Atlanta, Georgia, USA (Jun 2013)
9. Vilariño, D., Pinto, D., León, S., Alemán, Y., Gómez, H.: BUAP: N-gram based Feature Evaluation for the Cross-Lingual Textual Entailment Task. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). pp. 124–127. Association for Computational Linguistics, Atlanta, Georgia, USA (Jun 2013)
10. Vilariño, D., Pinto, D., Tovar, M., León, S., Castillo, E.: BUAP: Lexical and semantic similarity for cross-lingual textual entailment. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). pp. 706–709. Association for Computational Linguistics, Montréal, Canada (7-8 Jun 2012)
11. Wäschle, K., Fendrich, S.: HDU: Cross-lingual Textual Entailment with SMT Features. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). pp. 467–471. Association for Computational Linguistics, Montréal, Canada (7-8 Jun 2012)

ISSN 1870-4069

Evaluación del modelo neuronal de atención visual en la descripción automática de imágenes en español

Rafael Gallardo-García, Beatriz Beltrán-Martínez, Darnes Vilariño Ayala

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

rafael.gallardo@alumno.buap.mx,
{bbeltran,darnes,}@cs.buap.mx

Resumen. Este artículo presenta la evaluación del desempeño del modelo de atención visual neuronal presentado por Kelvin Xu, et al. en 2016. Se entrena y prueba el modelo con una nueva versión del conjunto de datos Flickr8k con una traducción al español. Así, se realiza el primer análisis cuantitativo del desempeño de este tipo de modelos en el idioma español. El artículo presenta la puntuación de similitud de las descripciones obtenidas, dicha puntuación se calculó con el algoritmo Bilingual Evaluation Understudy (BLEU) e incluye comparaciones entre las descripciones reales y las generadas de modo que se pueda realizar una evaluación directa. Se provee el conjunto de datos traducido y el código a través de GitHub.

Palabras clave: Análisis cuantitativo, BLEU, codificador-decodificador, lenguaje natural, subtitulado.

Evaluation of the Neural Model of Visual Attention in the Automatic Description of Images in Spanish

Abstract. This paper presents a performance analysis of the neural model of visual attention presented by Kelvin Xu, et al. in 2016. The model was trained and tested with a new Spanish translated version of the Flickr8k dataset. This is the first quantitative analysis of the performance of the visual attention models when captioning images in Spanish. This paper presents an evaluation of the quality of the predicted descriptions, the score was calculated with the Bilingual Evaluation Understudy (BLEU) and includes comparisons between the ground truth captions and the generated ones in order to allow a direct analysis of the results. The Spanish translated dataset and the code for the experiments is provided through GitHub.

Keywords: Quantitative analysis, BLEU, encoder-decoder, natural language, captioning.

1. Introducción

Una de las principales motivaciones que han llevado al humano a realizar investigación en el área de la inteligencia artificial es poder replicar las capacidades humanas en las computadoras. El área de la visión artificial busca replicar aquellas capacidades humanas de comprender el entorno a través de estímulos visuales. Los resultados del trabajo es esta área han sido tan variados como impresionantes: clasificación de imágenes con desempeños impresionantes [17,24], generación de contenido realista [14], traducción de estilos con redes condicionadas [21], súper fluidez y resolución en videos [3,5], entre muchos otros avances. Sin embargo, aunque las computadoras han llegado a dominar una gran variedad de áreas en el contenido visual hacía falta replicar una de las características más remarcables del humano: comprender escenas visuales e interpretar información detallada de dicha escena con exactitud.

El problema de la descripción automática de imágenes, bien conocido en el idioma inglés como *image captioning*, es un problema interesante que debe abordarse desde la interdisciplinaridad de las técnicas de aprendizaje automático: las mejores soluciones se obtienen utilizando híbridos entre las técnicas de visión artificial y procesamiento del lenguaje natural [6,18]. Para que un sistema de inteligencia artificial replique la capacidad humana de interpretar y describir una escena es necesaria más de una técnica de aprendizaje automático: reconocimiento de objetos, clasificación de imágenes, comprensión sintáctica y semántica de las escenas, además de la capacidad de poder expresar todo lo anterior en forma de lenguaje natural a través de técnicas como los modelos del lenguaje.

a little girl in a pink dress going into a wooden cabin .
a little girl climbing the stairs to her playhouse .
a little girl climbing into a wooden playhouse .
a girl going into a wooden building .
a child in a pink dress is climbing up a set of stairs in an entry way .

Fig. 1. Descripción de una escena con lenguaje natural en inglés, tomado de Flickr8k [15].

Uno de las características más interesantes de los sistemas de visión humana es la presencia de la *atención* [7,23], en vez de comprimir toda la información contenida en una escena en una representación estática, la atención permite analizar características sobresalientes de dicha escena de forma dinámica, similar al funcionamiento de la atención humana en ciertas áreas de una escena. Este artículo reporta las evaluaciones sobre el modelo de descripción automática de imágenes basado en atención presentado por Kelvin Xu, et al. [26] pues aunque el modelo basado en atención entrenado en el idioma inglés obtiene resultados fascinantes, es de interés analizar de forma cuantizada su desempeño en el lenguaje español.

2. Estado del arte

Las primeras propuestas para resolver el problema consistían en generar descripciones utilizando características manuales que operan sobre lenguajes formales [27], algunas otras propuestas consistían en utilizar detección de objetos y plantillas de descripciones [11], los enfoques más actuales se han respaldado en arquitecturas como las redes neuronales convolucionales (CNN) con esquemas de codificador-decodificador para codificar la imagen y de redes neuronales recurrentes (RNN) para generar texto a partir de la codificación obtenida con la CNN.

En los últimos 10 años se ha publicado una gran variedad de propuestas para solucionar el problema de descripción automática de imágenes, dichas propuestas afrontan el problema desde diversas áreas del aprendizaje automático: enfoques convolucionales [1], representación visual recurrente [4], con vecinos más cercanos [8], con redes recurrentes convolucionales a largo plazo [9], aprendizaje de instancias múltiples [10], con redes *long-short term memory* (LSTM) bidireccionales [25], aprendizaje reforzado [12,22] o con atención [26,28]. Gómez-Garay et al, reportan haber encontrado sistemas de descripción automática de imágenes en idioma inglés, japonés, chino, alemán y francés [13].

Gomez-Garay et al. presentaron un sistema [13] que presenta descripciones verbales en español sobre el entorno, dicho sistema recibía estímulos del entorno en formato de imágenes. Su sistema tardaba en promedio 0.96 segundos para generar la descripción para una imagen. El sistema generó un total de 8,693 frases y 47,061 palabras (838 únicas) para un conjunto de datos de 103 imágenes. Los autores no proveyeron un puntaje o métrica específica que evalúe las descripciones generadas.

En 2019, Martinez-Gutierrez evaluó el potencial de la API de visión computacional de Microsoft Azure en los idiomas español e inglés [20], concluyendo con un puntaje BLEU de 21.09 para el idioma español, siendo en su experimento, un puntaje mayor al obtenido por la API para el idioma inglés, sin embargo, la autora concluye que es un puntaje BLEU bajo, lo que significa que la correlación entre las descripciones generadas por la máquina y los humanos es débil.

3. Descripción del modelo de generación automática de imágenes

El modelo de descripción automática de imágenes que se evalúa en este artículo recibe el nombre de *Image Caption Generation with Attention Mechanism*, publicado con el nombre *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention* [26]. Los autores introdujeron un modelo basado en atención que aprende a describir el contenido de las imágenes automáticamente, en su artículo, ellos describen como entrenar el modelo de forma determinista utilizando las técnicas estándar de retropropagación y maximizando el límite inferior variacional con técnicas estocásticas.

En el artículo original, los autores describen distintos enfoques para generar las descripciones que intentan incorporar dos variantes de los mecanismos de atención: un mecanismo de atención "duro" y uno "suave". Además, defienden la ventaja de que incluir un mecanismo de atención permite visualizar lo que el modelo "ve" en un momento del análisis. El modelo que presentan puedan atender una parte destacada de la imagen mientras generan la descripción.

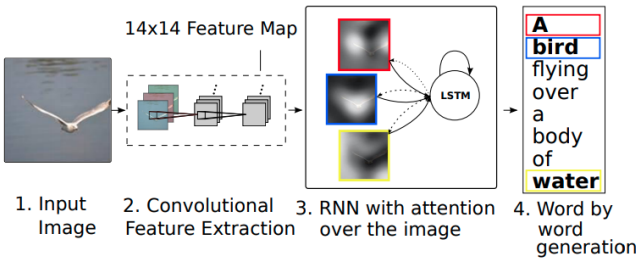


Fig. 2. Modelo de descripción automática con atención [26].

La figura 2 describe el funcionamiento del modelo propuesto en el artículo mencionado. El primer paso consiste en ingresar la imagen de entrada, posteriormente, un decodificador convolucional extrae las características de dicha imagen, el paso 3 consiste en generar las zonas de atención sobre la imagen con un decodificador basado en redes neuronales recurrentes, por último, el sistema genera una palabra basada en el vector de contexto entregado por el mecanismo de atención.

3.1. Codificador

El modelo en cuestión toma una imagen y genera una descripción y codificada como una secuencia de 1 de K palabras codificadas:

$$y = y_1, ..., y_C, y_i \in \mathbb{R}^K, \tag{1}$$

dónde K es el tamaño del vocabulario y C es la longitud de la descripción.

El codificador utiliza una red neuronal convolucional para extraer un conjunto de vectores característicos. El extractor produce L vectores, cada es una representación D-dimensional que corresponde a una parte de la imagen:

$$a = a_1, ..., a_n, a_i \in \mathbb{R}^D. \tag{2}$$

Para obtener la correspondencia entre los vectores característicos y las porciones de la imagen $2-D$, se extraen las características de una capa convolucional menor (a diferencia de trabajos anteriores que utilizan una capa completamente conectada). Esto último permite al decodificador enfocarse de forma selectiva en algunas partes de una imagen seleccionando un subconjunto de todos los vectores característicos.

3.2. Decodificador

El decodificador utiliza una red LSTM que produce una descripción, para generar la descripción, la red LSTM genera una palabra en cada paso de tiempo condicionado por el vector de contexto, el estado oculto y las palabras generadas anteriormente.

En términos generales, cada celda LSTM de la red implica proyecciones con un vector de pesos previamente aprendido. Cada celda aprende como pesar sus compuertas de entrada, mientras aprende como modular esa contribución a la memoria. Además, aprende los pesos que debe borrar de cada celda de memoria y los pesos que controlan como debería ser emitida esa memoria.

Para entender a profundidad la implementación de la red LSTM para el decodificador, se recomienda leer el artículo de Kelvin Xu et al. en la sección 3.1.2.

3.3. Mecanismos de atención

Se introducen dos técnicas de descripción automática de imágenes:

1. Un mecanismo de atención determinista "suave", entrenable mediante los métodos estándar de retropropagación.
2. Un mecanismo de atención estocástico "duro", entrenable mediante la maximización del límite inferior variacional o su equivalente mediante refuerzo.

Los detalles matemáticos y la demostración de cada uno de los mecanismos de atención se pueden encontrar en el artículo de Kelvin Xu et al [26].

4. Metodología de evaluación

Para poder evaluar el rendimiento en el idioma español del modelo descrito en la sección anterior de este artículo, es necesario entrenarlo con un conjunto de datos en pares que contenga imágenes y sus posibles descripciones en el lenguaje deseado, una vez que se cuente con ese conjunto de datos, se puede proceder con un análisis del vocabulario y la estructura del lenguaje contenido en las descripciones de las imágenes, posteriormente, será posible realizar el entrenamiento del modelo y las pruebas para medir cuantitativamente como se desempeña dicho modelo en el idioma español.

4.1. Traducción del conjunto de datos Flickr8k

Como se mencionó anteriormente, para poder evaluar el desempeño del modelo en el idioma español primero es necesario contar con un conjunto de datos que nos permita entrenar el modelo para generar descripciones en español.

No existen conjuntos de datos para descripción automática de imágenes en español que fueran de utilidad para los fines de este artículo, por lo que los autores generaron un conjunto de datos en pares para *image captioning* en

español. El conjunto de datos base recibe el nombre de Flickr8k y es resultado del trabajo de investigación [15] realizado por Hodosh et al. en 2013.

El conjunto Flickr8k contiene un total de 8,092 imágenes con 5 posibles descripciones para cada imagen. Las imágenes pueden contener animales, personas y objetos. Las posibles descripciones de cada imagen se enfocan en distintos aspectos de la escena y cuentan con distintas construcciones lingüísticas.

Proceso de traducción El conjunto de datos tiene un total de 40,460 descripciones de cada imagen, en idioma inglés. Para generar el conjunto de datos con las descripciones en español se utilizó la API de Google Cloud para traducción de texto en su versión básica. Se tradujo automáticamente el total de 40,460 descripciones y se emparejó cada quinteto de descripciones en español con su correspondiente imagen.

Por ahora, el modelo que se entrene a partir de estas descripciones podría heredar los sesgos del modelo de traducción automática de texto que contenga la API de Google. En trabajos posteriores se buscará verificar cada una de las traducciones manualmente para aumentar la fiabilidad del modelo de descripción de imágenes en español. Así mismo, se traducirán conjuntos de datos con una mayor cantidad de ejemplos de entrenamiento, como podría ser el conjunto de datos COCO [19] u 80K Flickr [16].

una niña en un vestido rosa de entrar en una cabina de madera.
una niña que sube las escaleras hasta su casa de juegos.
una pequeña muchacha que sube en una casa de juegos de madera.
una niña de entrar en un edificio de madera.
un niño en un vestido rosa está subiendo por una escalera en una puerta de entrada.

Fig. 3. Descripción de una escena con lenguaje natural en español, tomado de Flickr8k traducido.

La figura 3 es un ejemplo de entrenamiento de la versión de Flickr8k traducida al español automáticamente. Como se puede observar, el contenido de la imagen sigue siendo expresado en términos generales pero se pierden algunas de las características gramaticales y morfológicas del español.

4.2. Análisis de Flickr8k en español

El conjunto de datos de Flickr8k traducido al español varía en cantidad de vocabulario a su versión en inglés. Las descripciones de las imágenes en inglés acumulan un vocabulario total de 8,918 palabras, siendo el artículo *a* el más común con un total de 62,989 apariciones, por otra parte, la versión en español tiene un vocabulario de 12,439 palabras, siendo el artículo *un* el más común con un total de 39,366 apariciones.

Como se puede observar, la traducción del conjunto de datos añadió 3,521 palabras al vocabulario. Solo se removerán signos de puntuación durante la limpieza. Remover palabras cerradas resultaría en descripciones pobres en sintaxis y morfología.

La figura 4 muestra las 50 palabras más frecuentes en todo el conjunto, las más comunes llegan casi a una frecuencia de 40,000 mientras que las menos frecuentes apenas aparecen una vez en todas las descripciones.

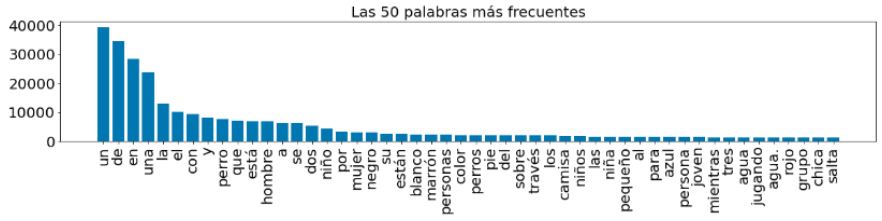


Fig. 4. 50 palabras más comunes en Flickr8k en español.

4.3. Implementación del modelo

El modelo de descripción de imágenes basado en atención se implementó en el lenguaje Python3 con soporte de la plataforma para aprendizaje automático TensorFlow [1]. El código utilizado para la implementación está basado en cuaderno de Jupyter *Image captioning with visual attention* [2], la versión modificada de dicha libreta para el conjunto de datos en español y la versión de Flickr8k en español se pueden encontrar en el siguiente repositorio de GitHub: *https://github.com/gallardorafael/ShowAttendTell_Flickr8k_Spanish*.

Estructura del conjunto de datos El conjunto de datos utilizado cuenta con 8,092 imágenes, como se mencionó anteriormente, cada imagen cuenta con 5 descripciones con el siguiente formato:

nombre_imagen,id_descripción,descripción_textual

En dónde *nombre_imagen* contiene el nombre del archivo .jpg de la imagen a la que esa descripción corresponde, *id_descripción* es el identificador para la descripción en cuestión (5 por cada imagen) y *descripción_textual* contiene el texto crudo que describe a la imagen y sobre el cuál se entrenará el modelo.

El archivo con el conjunto de datos es un CSV que contiene 8,092 líneas, cada línea es un ejemplo de entrenamiento como el descrito anteriormente, cada

[1] https://www.tensorflow.org/
[2] https://www.tensorflow.org/tutorials/text/image_captioning

línea está dividida por una ",", de modo que pueda ser importada con facilidad sin necesidad de realizar un procesamiento adicional.

Implementación del modelo Para implementar el modelo se utilizó un cuaderno de *Jupyter* de modo que la ejecución se pueda realizar con facilidad en una sesión de *Google Colaboratory* [3]. La información detallada sobre el código se encuentra en el repositorio mencionado con anterioridad, pero por motivos de claridad, a continuación se detallan los puntos importantes de la implementación:

- Se utilizó una capa convolucional menor de InceptionV3 (a su vez preentrenada sobre Imagenet [17]), con dimensiones (8,8,2048), mismas que se redimensionaron a (64,2018).
- El codificador implementado es una CNN que consiste en una capa completamente conectada.
- El decodificador implementado es una RNN, específicamente una *Gated Recurrent Unit* (GRU), que será la encargada de predecir la siguiente palabra.
- Se implementó un mecanismo de atención Bahdanau [2].
- Se utilizó el método *teacher forcing* para acelerar de forma eficiente el entrenamiento del modelo.

5. Resultados experimentales

El modelo se entrenó sobre un total de 8,000 imágenes y sus 40,000 descripciones correspondientes, de las cuales el 20 % se utilizó para validación y el otro 80 % para entrenamiento del modelo. Las 91 imágenes y sus 455 descripciones fueron utilizadas para la evaluación del modelo, esto es, calcular los puntajes de secuencia-secuencia y los análisis cualitativos de los resultados.

5.1. Proceso de entrenamiento

La primera parte del entrenamiento consistió en extraer las características de las imágenes utilizando la última capa convolucional (codificador) de InceptionV3, para extraer dichas características se utilizó una GPU NVIDIA Tesla P100 con 16 gigabytes de VRAM. Para extraer las características de las 8000 imágenes fueron necesarias 450 iteraciones y 35 minutos de procesamiento, con un promedio de 4.5 segundos por iteración.

La segunda parte del entrenamiento consistió en entrenar los mecanismos de atención con las características extraídas de las imágenes y de los textos de modo que el sistema aprendiera a describir cada imagen. Sobre la misma plataforma (NVIDIA Tesla P100), el entrenamiento con 20 épocas tomó un tiempo total de 56 minutos. La figura 5 muestra el gráfico de la función de pérdida del entrenamiento a través de las épocas. Es fácil observar que aumentar la cantidad de épocas más allá de 20 en un conjunto de datos reducido como este nos llevaría a un sobre ajuste del modelo.

[3] https://colab.research.google.com/

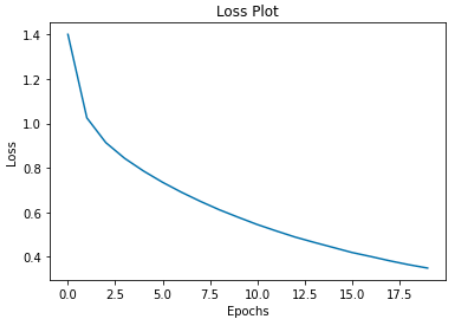


Fig. 5. Gráfico del entrenamiento a través de las épocas.

Una vez concluido el proceso de entrenamiento se puede proceder con la fase de pruebas para evaluar el desempeño del modelo entrenado.

5.2. Análisis cualitativo

El objetivo de este artículo es realizar una evaluación cuantitativa y cualitativa del desempeño del modelo en español. Se presentarán ejemplos de los peores resultados (con puntaje BLEU < 0.3) y de los mejores resultados (con puntaje BLEU > 0.7), así como un ejemplo del desarrollo del mecanismo de atención sobre una imagen tomada del set de prueba.

La figura 6 es una demostración del mecanismo de atención empleado. En la parte superior se encuentra la imagen mostrada al modelo, la parte inferior es la visualización del mecanismo de atención enfocando diversas partes de la imagen para generar una palabra a partir de sus *observaciones*. La descripción real para esta imagen es la mostrada en el primer renglón: *un perro blanco corriendo detrás de una pelota amarilla*, mientras que la generada por el modelo es: *un perro blanco está jugando con una pelota amarilla en su boca*. No es necesaria una métrica para concluir que esta descripción es acorde al contenido de la imagen, describe tanto los objetos como las acciones presentes en la escena.

La figura 7 muestra algunos ejemplos de malas descripciones generadas por el modelo, todas con puntaje BLEU menor a 0.3.

La figura 8 muestra ejemplos de descripciones con puntajes BLEU mayores a 0.7. Al igual que en el ejemplo de la figura 6, no es necesaria la métrica para comprobar que la descripciones coinciden de forma adecuada (aunque no completa) con el contenido de las escenas.

5.3. Análisis cuantitativo

Para realizar el análisis cuantitativo de los resultados se utilizará la medida BLEU. BLEU es un método de evaluación de la calidad de las traducciones realizadas por sistemas de traducción automática, en este sistema, una traducción tiene mayor calidad cuanto más similar es con respecto de otra de referencia que

Fig. 6. Visualización del mecanismo de atención del modelo.

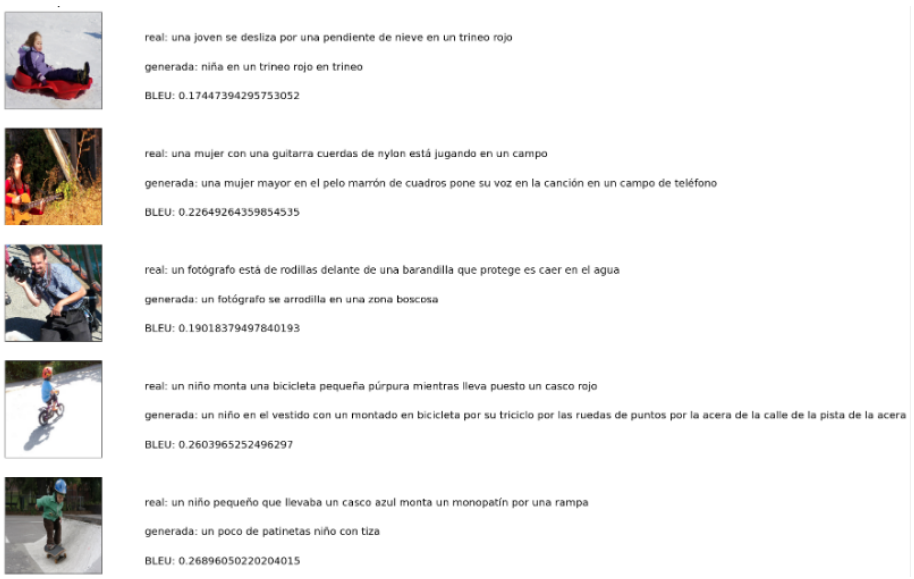


Fig. 7. Descripciones generadas con un puntaje BLEU < 0.3.

Fig. 8. Descripciones generadas con un puntaje BLEU > 0.7.

se supone correcta, BLEU permite evaluar con más de una referencia. Aunque este método surgió para evaluar traducciones, es ampliamente utilizado por las investigaciones del estado del arte mencionados en este artículo para evaluar la similitud de dos textos del mismo idioma.

Para este análisis se describieron 300 imágenes del conjunto de prueba y se calculó el puntaje BLEU para cada una de ellas: las hipótesis son las descripciones generadas y las referencias las descripciones reales obtenidas del conjunto de datos. La Tabla 1 muestra los resultados obtenidos por este método para el español.

Tabla 1. Evaluación de los puntajes BLEU obtenidos por el método.

Media	Mediana	Más alto	Más bajo	Desviación estándar
0.356	0.346	0.816	0	0.192

Una media de 0.356 no es buena, sin embargo es mayor que el reportado por Martinez-Gutierrez para la API de Microsoft Azure (obtuvo un 0.21). El puntaje

BLEU más alto obtenido por el modelo es de 0.816 y el más bajo de 0, dada la dispersión de los puntajes se decidió calcular la varianza, misma que tiene un valor de 0.192, lo que significa que los puntajes no están demasiado alejados de su media.

6. Conclusión

Aunque la media de los puntajes obtenidos es baja, existen descripciones con puntajes BLEU muy altos, lo que indica que una mayor cantidad de datos de entrenamiento podría mejorar satisfactoriamente los puntajes. Por otro lado, las descripciones mejor puntuadas describen con una exactitud alta el contenido de las imágenes tanto en objetos como en las acciones que se están llevando a cabo en la escena, por lo tanto, esas descripciones con puntajes BLEU mayores a 0.7 satisfacen los criterios cualitativos y cuantitativos de la descripción automática de imágenes.

El modelo de atención implementado en este trabajo y entrenado sobre el idioma español obtuvo resultados positivos muy satisfactorios y aunque la media de los puntajes es baja, no es menor que el límite impuesto (0.3) para ser considerada una media mala. Como se mencionó en el párrafo anterior, los resultados de este modelo podrían mejorar añadiendo más ejemplos de entrenamiento y mejorando la calidad de dichos ejemplos. Posiblemente, con una traducción al español realizada por expertos en vez de la automática se puedan obtener descripciones que respeten mejor la morfología y sintaxis del idioma, de forma que suenen coherentes y *más humanas*. Una propuesta para mejorar los resultados del modelo de atención para la descripción automática de imágenes es la utilización de un conjunto de datos con más ejemplos de entrenamiento, como 80K Flickr o MS COCO, aunque esto implique incrementar drásticamente los tiempos de entrenamiento del modelo. De igual forma, la calidad de las descripciones puede mejorar si se cambia la traducción automática de los conjuntos de datos por una traducción manual o semi-supervisada de las descripciones.

Desafortunadamente, el modelo no puede ser comparado de forma cuantitativa con otros modelos de descripción automática de imágenes pues las condiciones de entrenamiento y experimentación no hacen viable la comparación para este idioma, por esta razón, se ha decidido hacer público el conjunto de datos utilizado en este trabajo, de modo que futuros trabajos tengan la opción de evaluar modelos en idioma español que las condiciones de experimentación hagan viable la comparación.

Otra consideración que se deberá tomar en cuenta en trabajos similares o derivados de la descripción automática de imágenes en español, es el tipo de métricas utilizadas, en un futuro, la evaluación a este tipo de modelos debería enriquecerse con métricas que miden secuencia-secuencia como *Syntatic Features for Evaluation of Machine Translation* (SMT), *Metric for MT Evaluation with Improved Correlation with Human Judgments* (METEOR), con técnicas que consideren la complejidad morfológica como *Languaje-independent Model for Machine Translation Evaluation with Reinforced Factors* (hLEPOR), con

métricas que consideren paráfrasis, lemas y sinónimos como *Translation Edit Rate-Plus* (TER-Plus) o con métricas que no necesiten referencias como *Machine Translation Evaluation without Reference Texts* (MEWR).

Referencias

1. Aneja, J., Deshpande, A., Schwing, A.G.: Convolutional image captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5561–5570 (2018)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3703–3712 (2019)
4. Chen, X., Zitnick, C.L.: Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654 (2014)
5. Chu, M., Xie, Y., Leal-Taixé, L., Thuerey, N.: Temporally coherent gans for video super-resolution (tecogan). arXiv preprint arXiv:1811.09393 1(2), 3 (2018)
6. Cui, Y., Yang, G., Veit, A., Huang, X., Belongie, S.: Learning to evaluate image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5804–5812 (2018)
7. Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Annual review of neuroscience 18(1), 193–222 (1995)
8. Devlin, J., Gupta, S., Girshick, R., Mitchell, M., Zitnick, C.L.: Exploring nearest neighbor approaches for image captioning. arXiv preprint arXiv:1505.04467 (2015)
9. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2625–2634 (2015)
10. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., et al.: From captions to visual concepts and back. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1473–1482 (2015)
11. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: Generating sentences from images. In: European conference on computer vision. pp. 15–29. Springer (2010)
12. Feng, Y., Ma, L., Liu, W., Luo, J.: Unsupervised image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4125–4134 (2019)
13. Gomez-Garay, A., Raducanu, B., Salas, J.: Dense captioning of natural scenes in spanish. In: Mexican Conference on Pattern Recognition. pp. 145–154. Springer (2018)
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Advances in Neural Information Processing Systems 3 (06 2014)
15. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research 47, 853–899 (2013)

16. Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., Winnemoeller, H.: Recognizing image style. arXiv preprint arXiv:1311.3715 (2013)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
18. Kumar, A., Goel, S.: A survey of evolution of image captioning techniques. International Journal of Hybrid Intelligent Systems 14(3), 123–139 (2017)
19. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
20. Martinez Gutierrez, M.F.: Automated Image Captioning: Exploring the Potential of Microsoft Computer Vision for English and Spanish. Ph.D. thesis, University of Geneva (2019)
21. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
22. Ren, Z., Wang, X., Zhang, N., Lv, X., Li, L.J.: Deep reinforcement learning-based image captioning with embedding reward. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 290–298 (2017)
23. Rensink, R.A.: Change blindness: Implications for the nature of visual attention. In: Vision and attention, pp. 169–188. Springer (2001)
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision 115(3), 211–252 (2015)
25. Wang, C., Yang, H., Bartz, C., Meinel, C.: Image captioning with deep bidirectional lstms. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 988–997 (2016)
26. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
27. Yao, B.Z., Yang, X., Lin, L., Lee, M.W., Zhu, S.C.: I2t: Image parsing to text description. Proceedings of the IEEE 98(8), 1485–1508 (2010)
28. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4651–4659 (2016)

ISSN 1870-4069

Clasificación de granos de polen: Efecto de la distancia inter-pixel en GLCM

Leyra Martínez, Pedro Arguijo, Antonio Hiram Vázquez López,
Roberto Ángel Meléndez Armenta

Tecnológico Nacional de México,
Instituto Tecnológico Superior de Misantla,
México

leyramartinez.lm@gmail.com, pedro_arguijo@excite.com,
{jahvazquez, ramelendeza}@misantla.tecnm.mx

Resumen. La distancia juega un papel importante en la extracción de características de textura, ya que permite identificar los detalles finos y rugosos del objeto bajo prueba. En este trabajo se estudia el efecto, en la clasificación, de la distancia inter-pixel en las características de textura de segundo orden a partir de las matrices de coocurrencia a nivel de gris (GLCM, por sus siglas en ingles) de imágenes de polen. Las distancias consideras fueron de 1 a 5 pixeles de distancia. Las características se obtuvieron a partir de imágenes de granos de polen considerando su tabla de verdad asociada, en el espacio de color CIELab. Los resultados sugieren que el comportamiento de la Red Neuronal Artificial (RNA) funciona para analizar la textura con detalles finos, puesto que a menor distancia se obtuvo una mayor exactitud, mientras que los RF funcionan para analizar tanto los detalles finos como los detalles rugosos.

Palabras clave: Polen, textura, distancia inter-pixel, matriz de coocurrencia, CIELab, RGB.

Pollen Grain Classification: Effect of Inter-Pixel Distance on GLCM

Abstract. Distance plays an important role in the extraction of texture characteristics, since it allows to identify the fine and rough details of the object under test. This work studies the effect, in the classification, of the inter-pixel distance on the second order texture characteristics from the gray level co-occurrence matrices (GLCM) of pollen images. The distances you consider were 1 to 5 pixels away. The characteristics were obtained from images of pollen grains considering their associated truth table, in the CIELab color space. The results suggest that the behavior of the Artificial Neural Network (RNA) works to analyze the texture with fine details, since at a shorter distance a higher accuracy was obtained, while RF works to analyze both fine details and rough details.

Keywords: Pollen, texture, inter-pixel distance, co-occurrence matrix, CIELab, RGB.

1. Introducción

El objetivo principal de la clasificación de los granos de polen se enfoca en determinar la especie y la familia de las plantas a la cual corresponde. Para diferenciar las especies, un palinólogo entrenado realiza la clasificación de manera manual utilizando un microscopio, midiendo atributos visuales como forma, textura y ornamentación. Sin embargo, para poder hacer un análisis completo a partir de las características observadas, se requiere una gran cantidad de tiempo. A fin de lograr un reconocimiento y conteo rápido y preciso del polen, se han propuesto diversos enfoques de automatización, los cuales proporcionan un mejor resultado, mayor rapidez y objetividad que la identificación manual de polen no puede [1].

Desde que Flenley sugirió la utilización de algoritmos para clasificar y reconocer el polen, se han propuesto diversos algoritmos de extracción y reconocimiento de características para las imágenes de polen. Algunos estudios se han centrado en el análisis de la textura superficial del grano de polen con imágenes de microscopio electrónico de barrido (SEM) [2]. Aunque el sistema reporta una exactitud del 94% para seis taxones de polen distintos no es completamente automático, ya que requiere la preparación de una muestra completa para el análisis SEM. También se ha realizado la clasificación del polen en la miel considerando la textura de las imágenes del polen [3]. Las características de 10 especies diferentes de granos polen se calcularon a partir de matrices de coocurrencia y como clasificador redes neuronales, obteniendo una exactitud del 88%. Así mismo, se han implementado sistemas de clasificación de polen basados en aprendizaje profundo [4]. Aunque su tasa de clasificación fue del 94% en la identificación de 30 tipos de polen se debe considerar que el sistema requiere grandes conjuntos de imágenes para su entrenamiento.

En este trabajo se realiza la clasificación de imágenes de polen, nuestro enfoque se basa en la extracción de características de textura de segundo orden a partir de la matriz de coocurrencia a nivel de gris (GLCM) a escalas múltiples, consideramos cinco distancias inter-pixel [5,6] en el espacio de color CIELab. Evaluamos el desempeño individual de los conjuntos de características obtenidos con redes neuronales (RNA) y bosques aleatorios (RF). Esto con la finalidad de determinar el efecto de los detalles finos y rugosos en la clasificación. En la siguiente sección abordamos la extracción de las características de textura a partir de las matrices de coocurrencia, RNA y RF. En la sección tres se explica la metodología utilizada en este trabajo. Los resultados y las conclusiones se abordan en las secciones cuatro y cinco, respectivamente.

2. Marco teórico

2.1. Matriz de coocurrencia de niveles de gris (GLCM)

La textura, aunque carece de una definición formal, es un patrón visual que contiene información importante sobre la estructura de los objetos y su relación con el entorno circundante en términos de color, forma, tamaño, etc. El análisis de la textura como

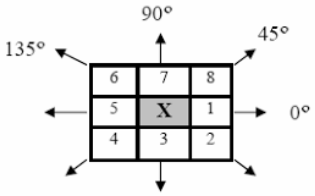


Fig. 1. Dirección de desplazamiento y rotación que definen las relaciones espaciales de los píxeles para la obtención de la matriz de coocurrencia. (Tomada de [8]).

atributo representativo visual se ha estudiado extensivamente en reconocimiento de patrones y visión por computadora, y entre alguna de sus aplicaciones se encuentra la segmentación y clasificación de imágenes, por mencionar algunas. En las imágenes digitales la textura se identifica por la disposición espacial de los pixeles con los niveles de gris. Aunque existen diferentes enfoques para extraer características de textura [7-9], en este trabajo se consideraron las que se obtienen a través de las matrices de coocurrencia (GLCM, por sus siglas en inglés).

La matriz de coocurrencia de una imagen da la probabilidad conjunta de segundo orden, P_d *(i, j)* de los valores de intensidad de dos píxeles (*i* y *j*), separados de una distancia *d* a lo largo de una dirección dada θ; reflejando la textura y las disposiciones espaciales de las intensidades de los píxeles presentes en la región de interés (ROI). Esta probabilidad conjunta; o sea la probabilidad de que *i* y j tengan la misma intensidad, toma la forma de una matriz cuadrada P_d, cuyo tamaño depende del número de niveles de gris discretos (intensidades) en la imagen que se está examinando [10]. Si una imagen de intensidad fuera completamente plana (es decir, no contuviera ninguna textura), la GLCM resultante sería completamente diagonal.

A medida que la textura de la imagen aumenta (es decir, a medida que las variaciones de intensidad de los píxeles locales aumentan), los valores fuera de la diagonal en el GLCM se hacen más grandes. En la fig. 1 se muestran las cuatro direcciones (0°, 45°, 90° y 135°) que se consideran para extraer la GLCM. Es importante señalar que GLCM proporciona una textura fina en distancias y recorridos cortos, y una textura gruesa en distancias y recorridos más largos [11]. El tipo de textura está relacionada con la frecuencia espacial, la textura fina con alta frecuencia espacial, mientras que la textura gruesa con una baja frecuencia espacial.

Haralick [10] propuso extraer 14 valores estadísticos de GLCM para describir las características de la textura de la imagen. En este trabajo solamente consideramos once características, las cuales se describen en la tabla 1, donde $C_{i,j}$ es la probabilidad en la celda *i,j*, *i* es el número de filas y *j* el número de columnas.

2.2. Clasificadores

2.2.1. Redes neuronales artificiales (RNA)

Una red neuronal artificial es un sistema de procesamiento de información que funciona de manera similar a las redes neuronales biológicas [13]. La base de estos modelos son los siguientes supuestos:

Tabla 1. Descripción de las características utilizadas.

Característica	Ecuación	Numero
Contraste	$C_1 = \sum_{i,j} (i-j)^2 . C_{ij} .$	(1)
Energía	$C_2 = \sum_{i,j} {C_{i,j}}^2 .$	(2)
Entropía	$C_3 = \sum_{i,j} - C_{i,j} . \log[C_{i,j}] .$	(3)
Homogeneidad	$C_4 = \sum_{i,j} \frac{C_{i,j}}{1+(i-j)^2} .$	(4)
Correlación	$C_5 = \sum_{i,j} C_{i,j} \left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)}(\sigma_i^2)} \right] .$	(5)
Varianza	$C_6 = \sum_{i,j} (i-\mu)^2 \, C_{(i,j)} .$	(6)
Suma de promedio	$C_7 = \sum_{i=2}^{2Ng} i C_{x+y}(i) .$	(7)
Suma de varianza	$C_8 = \sum_{i=2}^{2Ng} (i - f42)^2 C_{x+y}(i).$	(8)
Suma de entropía	$C_9 = - \sum_{i=2}^{2Ng} C_{x+y}(i) \log[C_{(x+y)}(i)] .$	(9)
Diferencia de varianza	$C_{10} = \sum_{i=2}^{2Ng} i^2 C_{x-y}(i).$	(10)
Diferencia de entropía	$C_{11} = - \sum_{i=2}^{2Ng} C_{x-y}(i) \log[C_{(x-y)}(i)] .$	(11)

— El procesamiento de la información se produce en un gran número de elementos simples llamadas neuronas.

— Las señales se transmiten entre las neuronas a lo largo de los enlaces de conexión.

— A cada enlace de conexión se le asigna un peso que multiplica la señal transmitida.

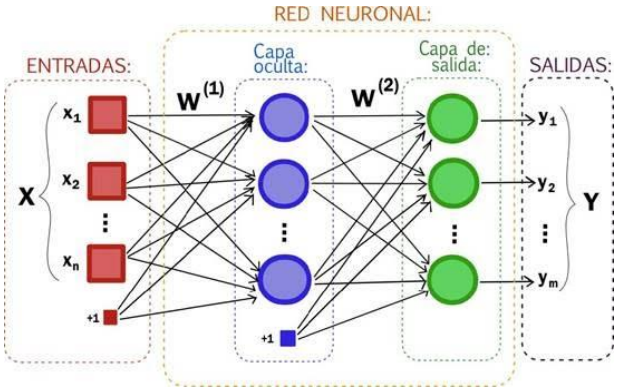


Fig. 2. Arquitectura de una red neuronal.

— Cada neurona aplica una función de activación en su entrada de red (que es la suma de las señales de entrada ponderadas) para obtener su señal de salida.

Existen dos estructuras en el desarrollo de las redes neuronales, redes de una sola capa y redes multicapa. En la fig. 2 se muestra un ejemplo de una red neuronal, en la cual se indican las interconexiones de la red, así como los pesos entre estas.

2.3.2. Random Forest (RF)

También conocidos como *Bosque Aleatorio* es una combinación de árboles predictores, tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. En otras palabras, este clasificador crea un conjunto de árboles de decisión a partir de un subconjunto de características de entrenamiento seleccionado al azar. Luego agrega los votos de diferentes árboles de decisión para decidir la clase final del objeto de prueba [14,15].

Cada árbol individual en el bosque aleatorio da como resultado una predicción de clase y la clase con más votos se convierte en la predicción del modelo (ver fig. 3).

3. Materiales y métodos

3.1. Dataset

El dataset empleado en este documento consta de 390 imágenes de polen divididas en 12 clases diferentes, de las cuales se tienen 618 instancias debido a que algunas imágenes presentan más de un grano de polen [16].

En la fig. 4 se muestra una imagen representativa de cada una de las clases de polen que se analizan. Las clases que tiene el dataset, con su respectivo nombre palinológico y el número de imágenes por clase se muestran en la tabla 2. Tello Mijares y Flores indican el procedimiento utilizado para la obtencion de las imágenes que conforman el dataset [16].

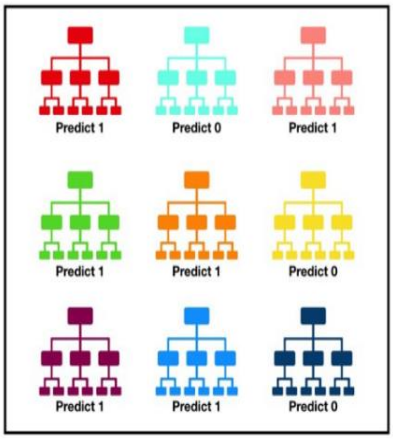


Fig.3. Visualización de un modelo de bosque aleatorio haciendo una predicción.

Tabla 2. Clasificación palinológica del conjunto de imágenes de polen.

Clase	*Nombre polen*	*Imágenes por clase*
1	Huisache	45
2	Alfalfa	29
3	Ortiga	31
4	Pasto	34
5	Hierba	34
6	Morera	24
7	Pacana	31
8	Oliva	28
9	Mesquite	50
10	Sauce	30
11	Pimienta	23
12	Sorgo de Alepo	31

En la fig. 5 se muestra la metodología utilizada en el desarrollo de este trabajo. Como se puede apreciar se inicia con la extracción individual del grano de polen considerando la tabla de verdad asociada. A estas imágenes se les realizó una convirtió de RGB a CIELab, con lo cual se llegó a un total de 1170 imágenes; 390 por cada canal.

A cada una de las imágenes se le extrajeron las características de textura mencionadas en la sección 2.1 considerando distancias de 1 a 5 píxeles, lo cual da un total de cinco distintos dataset; uno por cada distancia.

Para el cálculo de las GLCM no se realizó cuantización de los niveles de gris, esto a pesar de la carga computacional que representa al considerar las cuatro orientaciones. Como paso final se realizó la clasificación de los dataset con RNA y RF. Cabe señalar que en la clasificación se consideraron 2 proporciones de entrenamiento y prueba 50/50 y 70/3, estas proporciones se consideraron en cada uno de los dataset.

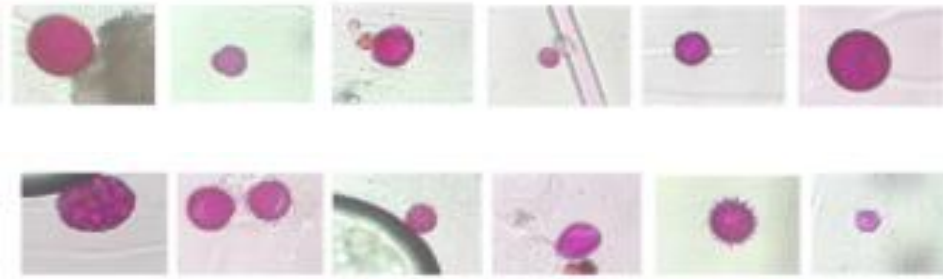


Fig. 4. Granos de polen por clases del 1- 12 y se representan así: 1-Huizache, 2-Alfalfa,3-Ortiga, 4-Pasto, 5-Hierba, 6-Morera, 7-Pacana, 8-Oliva, 9-Mesquite, 10-Sauce, 11-Pimienta, 12-Sorgo.

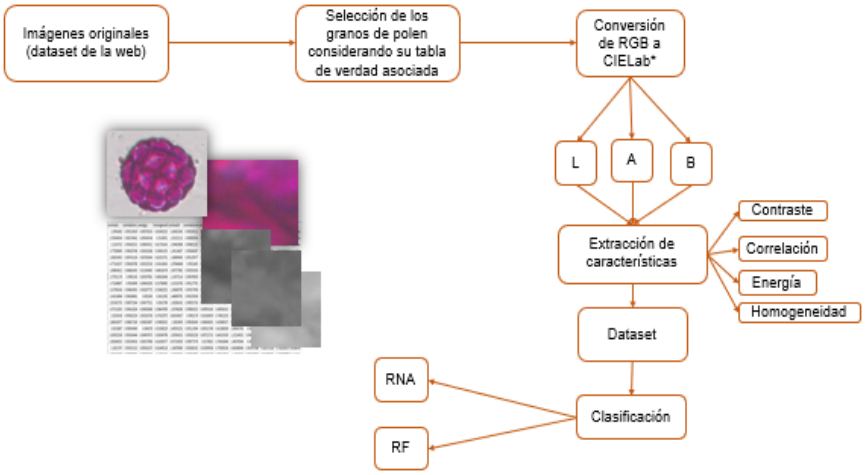


Fig. 5. Metodología grafica implementada en el desarrollo de este trabajo.

4. Resultado

El equipo en que se implementó la metodología descrita anteriormente es una laptop HP con procesador i3, 8 GB de RAM y sistema operativo a 64 bits. Para extraer el grano de polen, así como la conversión de RGB a CIELab y la extracción de características de las GLCM se utilizó MatLab 2016.

Mientras que para la clasificación se aprovechó R con su interfaz gráfica Rstudio. Para analizar el efecto de la distancia inter-pixel en la clasificación consideramos las proporciones 50/50 y 70/30 para el entrenamiento y las pruebas de cada dataset.

Como se buscó maximizar la precisión en la clasificación consideramos diferentes topologías en las redes neuronales [17] con la función arctan como función de activación, aunque se fijó a dos el número de capas ocultas. Mientras que para RF se utilizaron 200 árboles para la clasificación en todos los dataset.

Tabla 3. Resultado de clasificación por distancia para la proporción 50/50.

50/50			
Desplazamiento	**RNA**	**Número de neuronas por capa**	**RF**
1	90.29%	39,39	90.93%
2	90.29%	42,42	88.67%
3	88.02%	42,42	90.61%
4	87.05%	45,45	89.64%
5	89.32%	45,45	90.29%

Tabla 4. Resultado de clasificación por distancia para la proporción 70/30.

70/30			
Desplazamiento	**RNA**	**Número de neuronas por capa**	**RF**
1	93.51%	39,39	92.43%
2	88.64%	42,42	91.89%
3	88.10%	42,42	90.81%
4	88.10%	45,45	92.97%
5	92.43%	45,45	90.81%

En la tabla 3 se muestran los resultados de la clasificación al considerar la proporción 50/50, en la primera columna se indica la distancia inter-pixel considerada. En segunda de indica la precisión alcanzada por la red neuronal, seguida por la topología utilizada y en la cuarta columna se muestra la precisión de RF. Como se puede observar para la red neuronal los mejores resultados se obtienen para las distancias inter-pixel de 1 y 2, al haber llegado a un 90.29% en ambos casos, sin embargo, cambia el número de neuronas por capa oculta. RF alcanzó exactitudes del 90.93%, 90.61% y 90.29% para las distancias de 1, 3 y 5.

En la tabla 4 se muestran los resultados de la clasificación tomando una proporción 70/30, en la primera columna se indica la distancia inter-pixel considerada. Se puede observar que la exactitud va del 88 al 94% para la RNA y del 90 al 93% para RF. En este caso la mayor exactitud para la RNA se obtuvo con la distancia 1, alcanzando el 93.51%, con una topología de red de 2 capas ocultas con 39 neuronas cada una, mientras que para RF la mayor exactitud fue del 92.97% con distancia de 1 y del 92.97% con distancia de 4.

5. Conclusión

En este trabajo se consideraron distancias de 1 a 5 pixeles de separación para obtener características de textura de imágenes de polen, se utilizaron dos clasificadores RNA y

RF, utilizando dos proporciones de entrenamiento y prueba, 50/50 y 70/30, así como también las distintas topologías de red por distancia y un mismo número de árboles para cada distancia. Alcanzando exactitudes que van del 87 al 94%. La RNA alcanzo una exactitud del 90.29% para distancias 1 y 2 con la proporción 50/50 y una exactitud del 93.51% para la proporción 70/30 con distancia de 1.

Mientras que RF alcanzo una exactitud del 90.93%, 90.61% y 90.29% para las distancias de 1, 3 y 5, con una proporción del 50/50 y una exactitud del 92.42% y del 92.97% en distancias de 1 y 4 con una proporción del 70/30. Concluyendo que las RNA funciona para analizar la textura con detalles finos puesto que a menor distancias estos detalles pueden tener una mayor apreciación, mientras que FR funciona para analizar tanto los detalles de textura finos como para detalles rugosos. Consideramos que estos resultados no son concluyentes y es necesario estudiar otros dataset para confirmar el efecto de las distancias en las clasificaciones. De igual manera se necesitan otros extractores de textura como LBP o alguna de sus variantes junto con GLCM.

Referencias

1. Polen: Propiedades, utilización y beneficios. https://mejorconsalud.com/polen-propiedades utilizacion-beneficios/ (2020)
2. Langford, M., Taylor, G.E., Flenley, J.R.: Computerized identification of pollen grains by texture analysis. Review of Palaeobotany and Palynology, 64(1-4), pp. 197–203 (1990)
3. Kaya, Y., Erez, M.E., Karabacak, O., Kayci, L., Fidan, M.: An automatic identification method for the comparison of plant and honey pollen based on GLCM texture features and artificial neural network. Grana, 52(1), pp. 71–77 (2013)
4. Daood, A., Ribeiro, E., Bush, M.: Pollen grain recognition using deep learning. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Porikli, F., Skaff, S., et al., editors. Advances in Visual Computing, 10072, Springer International Publishing, pp. 321–330 (2016)
5. Alvarado-Moreno, J.D.: Defect detection in textiles using special techniques fortexture analysis. Tecnura, 20(47), pp. 97–106 (2016)
6. Molina-Ospina, M.F., Parra-Archila, J.C., González-Gómez, A.F.: Identificación de la enterocolitis necrosante neonatal mediante técnicas de procesamiento digital de imágenes. Universidad Autónoma de Occidente (2008)
7. Satpathy, A., Jiang, X., Eng, H.L.: LBP-based edge-texture features for object recognition. IEEE Transactions on Image Processing, 23(5), pp. 1953–1964 (2014)
8. Castrillon, W.A., Alvarez, D.A., López, A.F.: Técnicas de extracción de características en imágenes para el reconocimiento de expresiones faciales. Scientia et technica, 14(38), pp. 7–12 (2008)
9. Claro, M., Santos, L., Silva, W., Araújo, F., Moura, N., Macedo, A.: Automatic glaucoma detection based on optic disc segmentation and texture feature extraction. CLEI Electronic Journal, 19(2), pp.5 (2016)
10. Presutti, M.: La matriz de co-ocurrencia en la clasificación multiespectral: tutorial para la enseñanza de medidas texturales en cursos de grado universitario. Jornada de Educação em Sensoriamento Remoto no Âmbito do Mercosul, pp. 1–9 (2004)
11. Soni, N., Priya, S., Bathla, G.: Texture analysis in cerebral gliomas: a review of the literature. AJNR Am J Neuroradiol, 40, pp. 928–934 (2019)
12. Haralick, R.M, Shanmugan, K., Dinstein, I.: Textural features for image classification. IEEE Transaction on System, Man and Cybernetics, SMC3, 6, pp 610–621 (1973)
13. Matich, D.J.: Redes neuronales: conceptos básicos y aplicaciones. Universidad Tecnológica Nacional (2001)
14. Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R.: Random forests for land cover classification. Pattern Recognition Letters, 27(4), pp. 294–300 (2006)

15. Biau, G.: Analysis of a random forests model. Journal of Machine Learning Research, 13, pp. 1063–1095 (2012)
16. Tello-Mijares, S., Flores, F.: A novel method for the separation of overlapping pollen species for automated detection and classification. Computational and Mathematical Methods in Medicine, Article ID 5689346, 12 pages (2016)
17. Martínez, L., Arguijo, P., López, A.H.V., Armenta, R.Á.M.: Configuración de red neuronal para el análisis de polen. Research in Computing Science, 148, pp. 91–98 (2019)

ISSN 1870-4069

Predicción del éxito de proyectos de videojuegos en Kickstarter con aprendizaje automático

Salvador Nevarez-Castellanos[1], Vicente García[1], Gilberto Rivera[1], Rogelio Florencia[1], Julia Patricia Sánchez-Solís[1], Alberto Ochoa-Zezzatti[2]

[1] Universidad Autónoma de Ciudad Juárez
División Multidisciplinaria en Ciudad Universitaria, México
[2] Universidad Autónoma de Ciudad Juárez
Departamento de Ingeniería Industrial y Manufactura, México

salva.neva96@gmail.com, vicente.jimenez@uacj.mx

Resumen. Kickstarter se ha convertido en el principal medio de financiamiento para proyectos de desarrolladores independientes. En esta plataforma se han subvencionado exitosamente alrededor de 2,500 proyectos de videojuegos. Aún cuando la de campañas exitosas parece considerable, es pertinente mencionar que 12,500 campañas de esta misma categoría fracasaron en recaudar su meta. Kickstarter establece en sus políticas que el capital recaudado en una campaña debe ser devuelto a sus respectivos contribuyentes, repercutiendo negativamente a los desarrolladores. Por lo tanto, resulta importante encontrar un medio de predección del éxito/fracaso de las campañas de videojuegos. Utilizando una base de datos con datos con variables homologadas de diversas fuentes como Kaggle, Kickstarter, Facebook y otras redes sociales, en el presente artículo se propone el uso de técnicas de desabalance de clases y la regresión logística para la predicción del éxito/fracaso de las campañas. Los resultados experimentales mostraron que es posible predecir el éxito/fracaso de una campaña con una media geométrica de 0.8870.

Palabras clave: Kickstarter, regresión logística, predicción, videojuegos.

Predicting the Success of Kickstarter-Funded Video Games with Machine Learning

Abstract. Kickstarter has become Independent Game Developers' main means of projects financing. This platform has allowed the founding of around 2,500 videogame projects. Despite this being a considerable

amount of successfully founded projects, around 12,500 videogame projects have failed to reach their goal. Kickstarter establishes that the money raised in failed campaign must be returned to their respective contributors, thus, affecting developers negatively. Therefore, it becomes important to find a way of predicting the failure/success of videogame campaigns. In this article, the use of logistic regression is proposed to perform this task. Using a database wich included standarized data from diverse sources such as Kagle, Kickstart, Facebook, Youtube and other social media, the present article proposes unbalanced classes and logistic regression techniques use to predict campaigns success/failure. With a geometric median of 0.8870, experiments' results showed that it is possible to predict campaigns' success/failure.

Keywords: Kickstarter, logistic regression, prediction, video games.

1. Introducción

La industria de los videojuegos tiene gran impacto económico en el mercado internacional. En el año 2017 las ganancias generadas por videojuegos sumaron un total de 121 billones de dólares a nivel mundial (presentando un incremento del 15 % anual) [1]. Dichas ganancias son una suma de aquellas generadas por empresas *triple A* y desarrolladores independientes. Las empresas independientes, también denominadas *indies*, se diferencían de las triple *triple A* por no disponer de presupuestos fijos para la producción de videojuegos. En esta categoría recaen pequeñas empresas, grupos de desarrollo informales o desarrolladores individuales. Para solventar los gastos derivados del desarrollo y exposición de sus videojuegos, las empresas *indies* se han apoyado de medios de financimiento denominados *crowdfunding*. El *crowdfunding* es un método de financiamiento de proyectos conformado pr la inversión de pequeñas cantidades de dinero por parte de un gran número de inversionistas. Existen 4 distintos modelos de financiamiento [3]: 1) basado en donaciones, 2) basados en recompensas, 3) basados en equidad y 4) basados en deudas.

Kickstarter es una plataforma perteneciente al modelo basado en recompensas, en la cual los contribuyentes reciben compensaciones o privilegios relacionados al proyecto en base a la cantidad de dinero invertida. Fue creada el 28 de abril de 2009 por Perry Chen, Yancey Strickler y Charles Adler, bajo la premisa de *ayudar a dar vida a proyectos creativos*. Actualmente, la plataforma aloja inversionistas quienes se presentan como usuarios registrados los cuales pueden consultar proyectos y hacer donaciones que parten desde 1 dólar. Los proyectos que se desean financiar se presentan en forma de *campañas* de financiamiento que conformadas por publicaciones acerca del proyecto en desarrollo o de una idea próxima a desarrollar. Según su estado de financiamiento, a las campañas se les asigna uno de los cinco posibles estados: en vivo, suspendido, cancelado, fallido o exitoso.

El hecho de que Kickstarter sea preferido por los desarrolladores de videojuegos independientes se debe a que hay alrededor de 15 millones de contribuyentes registrados en la plataforma. Asimismo, a partir de esta plataforma se han financiado videojuegos que tuvieron altos índices de ventas al momento de su lanzamiento, como: Shovel Knight (311 mil dólares recaudados de una meta de 75 mil) y Divinity: Original Sin (944 mil dólares de una meta de 400 mil) [5]. A pesar del evidente apoyo hacia proyectos de videojuegos, estadísticas disponibles en Kickstarter.com muestran que se han financiado exitosamente 150,000 de 418,000 campañas pertenecientes a todas las categorías [3]. Hasta el año 2017, del total de las campañas, 13,619 pertenecen a la categoría de videojuegos, siendo exitosas 2,539 (alrededor de un 18 %) [3]. Esto supone un problema para la financiación de proyectos de videojuegos, ya que, de no alcanzarse la meta establecida, la plataforma emplea la política de devolución del dinero recaudado a sus contribuyentes. Por ello, el objetivo principal de este artículo es predecir el éxito y/o fracaso, a partir de datos específicos obtenidos de las campañas de videojuegos y otras fuentes de información, considerando únicamente atributos tales como: meta de la campaña, gráficos del videojuego, título de la campaña. números de comentarios, actualizaciones realizadas, entre otros. El conjunto de datos conformado permitirá tener una descripción completa de la situación más apegada a la realidad de las campañas.

El artículo se estructura de la manera siguiente: Sección 2 presenta una breve revisión de los trabajos relacionados con la predicción del éxito en proyectos de Kickstarter., En la Sección 3 se describe el proceso de recolección, integración y preprocesamiento de los datos. Sección 4 se define el diseño experimental adoptado en este trabajo. Posteriormente, en la Sección 5 se muestran y discuten los resultados. Finalmente, en la Sección 6 se encuentran las conclusiones y el trabajo futuro.

2. Trabajos relacionados

En la literatura se pueden encontrar diversos trabajos orientados a predecir o determinar los factores que influyen en el fracaso o éxito de las campañas. Por ejemplo, Antonio Uribe, fundador de la empresa de desarrollo de videojuegos Hyper Beard, en su artículo sobre la situación del "Desarrollo de Videojuegos en México", expone que existe un apoyo mínimo por parte de la comunidad mexicana hacia los productos de sus compatriotas [6], debido a la preferencia del público en adquirir videojuegos desarrollados en Estados Unidos, sugiriendo que un posible factor de éxito es la preferencia de videojuegos desarrollados en ciertos países. Asimismo, menciona que otro posible factor que influye negativamente es la cantidad monetaria que se establece como meta de la campaña. Ello puede observarse en las campañas de Mulaka (videojuego sobre la cosmogonía tarahumara), que no logró su meta de 77,000 dólares [7], y Neon City Riders, videojuego mexicano, que cumplió su meta de 8,335 dólares [8].

Otra forma de abordar el problema de determinar los factores que han sido determinantes en el éxito, es mediante el uso de técnicas de minería de

datos. Ejemplo de ello, es el trabajo de Michal Trněný [9], en el cual se analizó la información de 4,634 juegos con la finalidad de identificar los factores que predicen el éxito de un videojuego publicado, definiendo el éxito como el número de ventas tales que cubran los costos de desarrollo del producto. La base de datos empleada se obtuvo mediante las API públicas de Steam, considerando para ello sólo juegos creados por compañías AAA (Por ejemplo, Ubisoft). Los atributos que se tomaron en cuenta fueron el género del videjuego, el precio, el desarrollador, así como el alcance en Internet. Para el análisis utilizó máquinas de soporte vectorial, árboles de decisión y redes bayesianas. Trněný [9] concluyó que existe una correlación entre el éxito y la experiencia del desarrollador, número de jugadores concurrentes en los primeros dos meses y los atributos descriptivos.

Por su parte, Li et al.[10] emplearon diversos modelos probabilísticos para la predicción de la posible fecha de éxito de una campaña de Kickstarter mediante un análisis de 18,093 proyectos registrados en la plataforma y que hayan alcanzado su meta de crowdfunding. Los datos fueron obtenidos de Kickspy, página que se dedica a recopilar datos de campañas de la mencionada plataforma, así como utilizar las API de Twitter y Facebook para complementar su colección de datos. Consideraron métricas de la plataforma (tiempo de inicio, tiempo de término, meta, total recaudado) complementadas con estadísticas de publicidad en Twitter y Facebook. Posteriormente, obtuvieron dos modelos, donde uno de ellos consideró únicamente los casos de éxito mientras que el otro consideró los datos en su totalidad. Concluyeron que las variables que determinan la fecha de éxito incluyen: la existencia de publicidad en redes sociales; y la fecha de inicio de la campaña. Asimismo, reconocieron que el modelo que contempla casos de éxito y fracaso presenta un error de predicción menor al 5 %.

Kindler et al. [11], a partir de un ánalisis realizado a una base de datos con datos del 2013, sugieren que es posible predecir en una etapa temprana el éxito de un campaña en Kickstarter, en donde, la comunidad de contribuyentes influye en que se alcance la meta deseada por la campaña.

Los trabajos sobre la predicción del éxito de campañas de Kickstarter, han empleado como fuente de información todos los atributos que describen a las campañas. Sin embargo, esto puede implicar que se utilicen atributos que describen exclusivamente a otros tipos proyectos enfocados al cine, el arte, la moda, el diseño, el teatro, entre otros.

3. Conjunto de datos

El conjunto de datos utilizado en este artículo contiene información de las campañas de videojuegos comprendidas entre los años 2010 y 2017. El proceso de recolección fue realizado de manera manual y automática, donde en este último caso se desarrollaron scripts en PERL y Ruby utilizando las librerías kickscraper de Mark Olson [12] y WWW::Kickstarter del usuario de CPan Ikegami [13], respectivamente.

3.1. Obtención de los datos

Para la conformación del conjunto de datos, se tomó como referencia la base de datos proporcionada por Kaggle [14], la cual está constituida por 378 mil registros de campañas de Kickstarter creadas entre octubre del 2009 y junio del 2018. Las campañas contenidas en la base de datos pertenecen a las 15 categorías de campañas soportadas por la plataforma. Asimismo, contempla un total de 14 atributos para cada campaña. Para cumplir con el objetivo de sólo emplear datos relacionados a los videojuegos se eliminaron registros de campañas que no pertenecieran a las subcategorías *Video Games* y *Mobile Games* de la categoría principal *Games*. Asimismo, registros cuyo estado de campaña fueran *live*, *paused* o *canceled*. Finalmente, todos aquellos registros cuyo año de inicio y finalización estuvieran fuera del rango 2010 a 2017.

Con la finalidad de enriquecer el conjunto de datos inicial con otros atributos, se llevó a cabo un proceso de extracción automática de información sobre la plataforma Kickstarter. Para ello, se apoyó en script desarrollados en PERL y RUBY, con los cuales se pudieron agregar 12 nuevos atributos:

- *Staff pick*: Variable dicotómica con valores [true, false] que indica si una campaña fue marcada como destacada dentro de Kickstarter.
- *Rewards levels*: Número de niveles de recompensas definido por el autor de una campaña.
- *Rewards Min*: Aporte necesario para adquirir el nivel de recompensa de menor valor (en divisa indicada).
- *Rewards Max*: Aporte necesario para adquirir el nivel de recompensa de mayor valor (en divisa indicada).
- *Project count at 2018*: Número de campañas creadas por el autor de la campaña hasta junio de 2018.
- *Project first*: Variable dicotómica con valores [true, false] utilizada para indicar si la campaña fue la primera del usuario.
- *Project count at creation*: Número de campaña que representa la instancia de campaña de Kickstarter.
- *Blurb*: Descripción corta ubicada al inicio de una campaña.
- *Rewards physical*: Indica la existencia de recompensas físicas dentro de los niveles de recompensa definidos.
- *Updates count*: Número de actualizaciones realizadas sobre el contenido de la campaña por parte del autor.
- *Comments count*: Número de comentarios que publicaron los usuarios de Kickstarter y el autor de la campaña a lo largo del tiempo de vida de la campaña.
- *Has video*: Variable dicotómica con valores [true, false]. Indica si la campaña contiene un vídeo como portada. No aplican videos embebidos en el contenido de la campaña.

En una segunda etapa, de forma manual se exploraron otras fuentes de información ligadas a los proyectos tales como Facebook, YouTube y páginas Web.

Adicionalmente se extendió dicha exploración a la plataforma de Kickstarter para obtener aquellos datos que los script no pudieron extraer. De este proceso se obtuvieron 15 atributos adicionales que son:

- *Release console*: Variable dicotómica con valores [true, false], indica si el videojuego que se intenta financiar se encontrará disponible en consolas.
- *Release PC*: Indica si el juego estará disponible en computadora (mediante navegador web, archivo ejecutable o por tiendas virtuales).
- *Release Mobile*: Indica si el videojuego se encontrará disponible en dispositivos móviles con sistema operativo Android (incluye OUYA), iOS, Windows Phone.
- *Facebook*: Variable dicotómica con valores [true, false], indica si existió publicidad del videojuego y/o la campaña en dicha red social.
- *Twitter*: Variable dicotómica con valores [true, false], indica la existencia de publicidad del videojuego y/o la campaña en dicha red social.
- *YouTube*: Variable dicotómica con valores [true, false]. Muestra si existió publicidad del videojuego y/o la campaña en dicha red social.
- *Social landing page*: Variable dicotómica con valores [true, false]. Indica si existió un sitio web del grupo desarrollador que detalle información sobre el videojuego.
- *Press kit*: Variable dicotómica con valores [true, false]. Indica si existió un kit de prensa para el videojuego de una campaña alojado en un sitio web o como un conjunto de archivos descargables.
- *Gameplay*: Variable dicotómica con valores [true, false]. Si la campaña contiene imágenes o videos de las mecánicas del videojuego. Incluye demos y prototipos del juego.
- *Game art style*: Variable categórica que asume los valores [3D, 2D, ND] según el tipo de gráficos del videojuego que se desea financiar.
- *Existing remake*: Variable dicotómica con valores [true, false]. Indica si el juego que se intenta financiar es una versión renovada de otro existente, de la cual no se tiene la licencia o es la modificación.
- *Campaing Target Audience*: Variable categórica que toma uno de los valores [Everyone, Adults, Kids] acorde al tipo de público para el cual se dedican la campaña y el videojuego.
- *Game VR compatibility*: Variable dicotómica con valores [true, false] utilizada para indicar si el juego es compatible con realidad virtual (VR).
- *Multiplayer*: Variable dicotómica con valores [true, false], indicadora de si el videojuego permite cualquier tipo de interacción con personas que dispongan del mismo juego.
- *Backed Projects*: Número de proyectos que el autor de la campaña ha apoyado.

Dado que el objetivo es predecir a priori el éxito o fracaso de una campaña de videojuego, se eliminaron los atributos *usd pleadge real* y *Backers*. Estos dos atributos pertenecen al primer conjunto de datos que se obtuvo de Kaggle. El primero de ellos describe la cantidad monetaria recaudada a lo largo del tiempo de vida de la campaña. El segundo, representa el número de contribuyentes que apoyaron a la campaña.

3.2. Preprocesamiento de los datos

Una vez integrada la información adicional a la base de datos inicial, se codificaron algunos atributos. Siguiendo los hallazgos de la investigación de Downs y Ghauri (investigación en la que demuestra la importancia de la longitud del nombre de una campaña) [15], se crearon nuevos atributos numéricos enfocados a representar el nombre de la campaña (*name*) y la descripción de la campaña (*Blurb*) en términos de la longitud de caracteres (*name chars*, *Blurb chars*) y del número de palabras (*name words*, *Blurb words*).

En el caso de los atributos de tipo fecha, se optó por crear 3 nuevos atributos que representan el mes, el día y el día de la semana. En el caso de los atributos de tipo moneda, debido a que los proyectos utilizan diferentes tipos de cambio, se decidió utilizar el dólar como la moneda estándar. Por lo que se realizó la conversión a dicha moneda, de acuerdo al tipo de cambio que existía en la fecha en la que se estableció la meta de la campaña.

De los 14 atributos originales, se eliminaron atributos que se consideraron irrelevantes para el objetivo de estudio como son: *category*, *main category* y *PID*. El primer y segundo atributo indican la categoría principal y subcategoría a la que pertenece la campaña. Sólo fueron útiles para extraer los datos de las campañas pertenecientes a videojuegos. Finalmente, en el caso de *PID*, es un identificador numérico de la campaña.

El conjunto final está conformado por 40 atributos (uno de ellos es la etiqueta de clase), de los cuales 12 son cuantitativos (numéricos) y el resto nominales o categóricos. Del número de instancias, un total de 1,079 son de la clase éxito y 3,043 de la clase fracaso.

3.3. Variables ficticias

Los modelos de regresión tratan todos los atributos (variables dependientes) como números. Por lo que, los atributos categóricos o nominales (no numéricos) fueron transformados a números con el fin de poder emplear el algoritmo de regresión. Por ejemplo, el atributo *Game art style* que puede tener uno de los tres valores 2D, 3D o ND (no especifica el tipo de gráfico), se utilizaron los números 1, 2 y 3 para representar cada una de los tipos de gráficos, respectivamente. No obstante, asignar un valor numérico a una variable categórica no representa que el 3 sea mayor que 2 ó que se puedan restar/sumar. Para solventar este problema, cada uno de los atributos categóricos fueron codificados a variables ficticias o *dummy*, la cual es un atributo artificial que toma un valor de 0 o 1, para indicar la ausencia y presencia del valor, respectivamente [16]. Si un atributo tiene n valores diferentes, entonces se crearán $n-1$ nuevos atributos ficticios [17]. En la Tabla 1 se muestra un ejemplo para la codificación del atributo *Game art style*.

4. Diseño experimental

El objetivo de este artículo es predecir y determinar los factores de éxito en proyectos de videojuego en Kickstarter.

Tabla 1. Creación de variables ficticias para *Game art style*.

Game art style	Nueva Variable 1	Nueva Variable 2
1 (2D)	1	0
2 (3D)	0	1
3 (ND)	0	0

Para ello, los experimentos se realizaron sobre el conjunto de datos conformado por diversas fuentes de información. La Fig. 1 muestra el diagrama de flujo del proceso empleado para la creación del modelo de predicción del éxito/fracaso de las campañas de videojuegos. Los primeros dos pasos incluyeron las actividades de preparación de la base de datos inicial; y la eliminación, transformación y creación de variables. Posteriormente, se realizaron actividades de construcción y evaluación del modelo, en las cuales se aplicó una validación cruzada de 5 particiones en su versión estratificada, por lo que se preservaron las probabilidades a priori y la independencia estadística entre las particiones de entrenamiento y test.

En el aprendizaje automático, uno de los métodos de clasificación más utilizados en problemas de dos clases es la regresión logística [18,19,20], el cual es modelo clásico que proviene del área de la estadística [19]. En este trabajo el modelo fue tomado de la librería de R denominada caTools [3].

4.1. Métricas de evaluación

La evaluación de un clasificador se suele basar en un tabla de confusión, la cual es una matriz, cuyas entradas (i, j) contienen el número las predicciones correctas/incorrectas [21]. La Tabla 2 muestra una matriz de 2×2 para un problema de dos clases, donde las columnas representan la salida estimada por parte del clasificador y las filas indican las clases reales. Los elementos en la diagonal principal contienen el número correcto de predicciones en la clase positiva y negativa, mientras que el resto de las entradas son los errores.

Tabla 2. Matriz de confusión para un problema de dos clases.

	Predicción Positivos	*Predicción Negativos*
Real Positivos	Verdaderos Positivos (TP)	Falsos Negativos (FN)
Real Negativos	Falsos Positivos (FP)	Verdaderos Negativos (TN)

Por lo general, la métrica empleada para evaluar la efectividad de un clasificador es la exactitud ($A = (TP + TN)/(TP + FN + TN + FP)$).

Sin embargo, se ha demostrado que dicha métrica no es apropiada cuando el conjunto de datos no está balanceado, por lo que, se suelen utilizar otras

[3] https://cran.r-project.org/web/packages/caTools/index.html

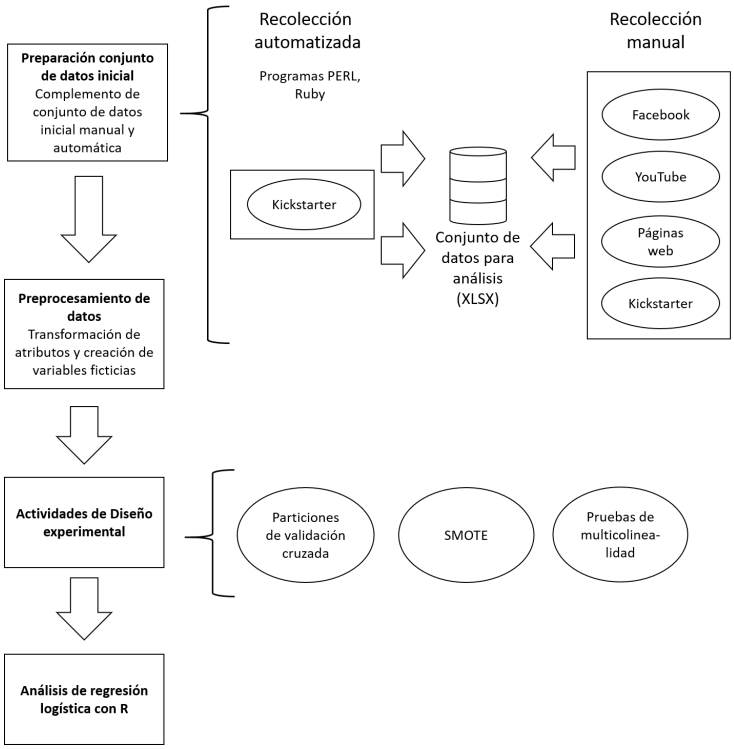


Fig. 1. Diagrama de flujo del proceso experimental.

alternativas [21]. Un ejemplo de esto son las métricas que evaluan la efectividad en cada clase como la tasa de verdaderos positivos ($TPr = TP/(TP + FN)$) y la tasa de verdaderos negativos ($TNr = TN/(TN + FP)$). Una tercera métrica que engloban a las dos anteriores es la media geométrica, la cual, maximiza la efectividad individual de cada clase mientras se mantienen balanceadas ($GM = \sqrt{TPr \times TNr}$) [22]. En este trabajo utilizaremos estas tres últimas métricas.

4.2. Clases no balanceadas

Uno de los mayores obstáculos en la construcción de modelos de aprendizaje automático, es el problema denominado de clases no balanceadas. Un conjunto de dos clases se dice que es no balanceado, si una de las clases (mayoritaria) tiene mayor presencia que la otra (minoritaria). Este tipo de complejidad de los datos puede conducir, en métodos tradicionales de clasificación, a aprendizajes sesgados en perjuicio de la clase minoritaria que, usualmente, contiene los casos de mayor interés [23]. En este trabajo la clase de mayor interés es la denominada como éxito y cuyo número de registros es de 1,079 registros, mientras que en el caso de la clase fracaso los ejemplos son 4,122. Esto supone un ratio de clases no balanceadas ($negativas/positivas$) igual a 3.82.

En la literatura existen diversas estrategias para tratar con el problema. Una de las más populares es aquella que incrementa el tamaño de la clase minoritaria ya sea creando de forma aleatoria copias de las instancias minoritarias o nuevas instancias artificiales a partir de heurísticas. Se ha demostrado que crear copias de los instancias puede conducir a un sobreajuste del modelo. Para evitar este problema algunos métodos crean ejemplos sintéticos, como es el caso de la técnica SMOTE (Synthetic Minority Oversampling TEchnique), la cual genera nuevos datos mediante la interpolación de instancias existentes de la misma clase [24]. Para ello, primero se obtienen los K vecinos más cercanos de la clase minoritaria. Posteriormente, las nuevas instancias son generadas en la dirección de algunos de estos vecinos. La Fig. 2 ejemplifica la interpolación de instancias con SMOTE.

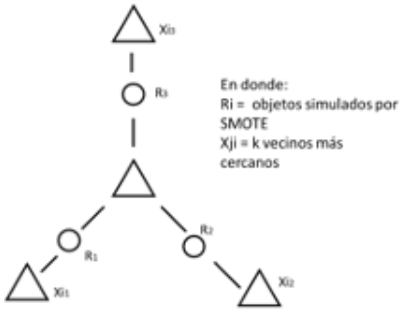


Fig. 2. Interpolación de valores con SMOTE.

En este trabajo se utilizó SMOTE hasta lograr que las clases estuvieran balanceadas, con valor de $K = 8$. Para evitar resultados resultados sobre ajustados o demasiado optimistas, el sobremuestreo de la clase minoritaria se realizó en el conjunto de entrenamiento en cada iteración de la validación cruzada [25].

5. Resultados y discusión

La Tabla 3 muestra los promedios de clasificación con la regresión logística, utilizando el conjunto original (sin los atributos *usd pledge real* y *backer*). Como se puede observar, debido a la problemática de las clases no balanceadas, la exactitud en la clase menos representada (clase éxito) presenta una tasa de 0.7465, mientras que el caso de la clase mayoritaria (clase fracaso) el resultado está por arriba de 0.94. A pesar de esto el resultado de la media geométrica alcanza el 0.8503. Es importante mencionar que, en este conjunto no se consideraron aquellos atributos que puende influir en el éxito de la compaña.

Al llevar a cabo los experimentos sobre el conjunto sobremuestreado por medio de SMOTE, se observa, que existe un incremento en la exactitud de la clase minoritaria (TPr), pasando de 0.7645 a 0.8712. Asimismo, se observe un ligero decremento en la exactitud en la clase mayoritaria .

Tabla 3. Resultados promedios de TPr, TNr y GM usando el conjunto no balanceado.

	TPr	TNr	GM
Original	0.7645	0.9457	0.8503

Con respecto al resultado de la media geométrica, el valor se incrementa hasta llegar a 0.8712 (vea Tabla 4).

Tabla 4. Resultados promedios de TPr, TNr y GM usando el conjunto balanceado con SMOTE para K=8.

	TPr	TNr	GM
Balanceado	0.8378	0.9059	0.8712

Una de las condiciones para usar la regresión logística es la ausencia de colinealidad entre las variables independientes (atributos de entrada) [26]. Por consiguiente, se aplicó una prueba de multicolineadad para detectar aquellos atributos que presenten dicha característica. En total se detectaron los siguientes atributos: *currency*, *deadline weekday*, *launched weekday*, *country*, *game art style*, *campaign target audience*. Existen diferentes formas de resolver la problemática de la multicolinealidad: eliminar las variables, cambiar la escala o combinarlas en una medida única [26]. En este trabajo por sencillez se optó por eliminar los atributos.

En la Tabla 5, se muestran los resultados de clasificación en términos de TPr, TNr y GM. Como se puede observar, el eliminar el problema de la multicolinealidad, los resultados de exactitud en la clase minoritaria se incrementan pasando de 0.8378 (resultados con el conjunto balanceado) a 0.8646. Esto ocasiona que también se de un incremento en la media geométrica hasta llegar a 0.8870

Tabla 5. Resultados promedios de TPr, TNr y GM usando el conjunto balanceado con SMOTE para K=8 y sin colinealidad.

	TPr	TNr	GM
Balanceado S/Col.	0.8646	0.9099	0.8870

6. Conclusiones y trabajo a futuro

El presente trabajo tuvo como objetivo el construir un modelo de regresión logística, para predecir el éxito/fracaso de campañas de videojuegos en Kickstarter. Con este fin, se construyó un conjunto de datos a partir de diversas fuentes tales como YouTube, Facebook, páginas web.

De esta manera, se pudo integrar diversos atributos que proveen información sobre los comentarios, el tipo de videojuego, las redes sociales, niveles de recompensa, entre otros. Posteriormente, se aplicaron diversos preprocesamientos a los datos, tales como convertir de datos nominales a numéricos, crear variables artificiales, y eliminar datos con multicolinealidad. Asimismo, debido a la presentación de clases no balanceadas se sobremuestreo el conjunto para tratar dicha problemática.

Los resultados obtenidos, en términos de la media geométrica, sugieren que la construcción de la regresión logística con la base de datos preprocesada obtiene los mejores resultados de clasificación, con valores cercanos al 0.8870 en términos de la media geométrica. Teniendo, en cuenta este resultado se puede concluir que es posible construir un modelo que sea capaz de predecir el éxito/fracaso de un proyecto de videojuego considerando una base de datos con atributos diversos. Como trabajo futuro, planteamos el uso de otros clasificadores com árboles de decisón, basados en regla, dada la naturaleza del conjunto de datos de tener una mezcla de atributos. Asimismo, aplicar técnicas de selección de características y de balanceo de clases.

Referencias

1. Wijman, T.: NewZoo, https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half (2018)
2. Alonso, A.: HobbyConsolas, https://www.hobbyconsolas.com/noticias/juegos-aaa-vs-juegos-indie-cara-cara-65552 (2018)
3. Reddy, S., Heng Tan, Y.: Crowfounding: Financing Ventures in the Digitial Era. Marketing Intelligence Review 9 (1), 37–41 (2017)
4. Kickstarter Homepage, https://www.kickstarter.com/about (2018)
5. Digital Trends, https://www.digitaltrends.com/gaming/best-kickstarter-funded-video-games/ (2018)
6. Uribe, A.: Medium Corporation, https://medium.com/@Fayer/nos-hace-falta-apoyar-m %C3 %A1s-a-los-h %C3 %A9roes-locales-3eef1163a613 (2018)
7. Kickstarter PBC: Mulaka Origin, https://www.kickstarter.com/projects/lienzo/mulaka-origin-tribes (2018)
8. Kickstarter PBC: Neon City Riders: A Cyberpunk Turf Wars Action Adventure, https://www.kickstarter.com/projects/mechastudios/neon-city-riders-a-cyberpunk-turf-wars-action-adve (2018)
9. Trneny, M.: Machine Learning for Predicting Success of Video Games. Masaryk University, Faculty of Informatics (2017)
10. Li, Y., Rakesh, V., Reddy, C.: Project Success Prediction in Crowdfunding. En: 9th ACM International Conference on Web Search and Data Mining, pp. 247–256. ACM, San Francisco, California (2016)
11. Kindler, A., Golosovsky, M., Solomon, S.: Early prediction of the outcome of Kickstarter Campaigns: is the sucess due to virality. Palgrave Communications, pp. 1–19 (2019)
12. Olson, M.: API Library for Kickstarter.com, https://github.com/markolson/kickscraper (2018)
13. Ikegami: WWW-Kickstarter-v1.12.0, https://metacpan.org/release/WWW-Kickstarter (2018)

14. Mouille, M.: Kickstarter Projects Dataset, https://www.kaggle.com/kemical/kickstarter-projects (2018)
15. Downs, R., Ghauri, M.: Predicting Kickstarter Campaign Success with Classification Models (2018)
16. Fox, J.: Applied Regression Analysis and Generalized Linear Models. SAGE Publications, London (2015)
17. The Odum Institute: Learn about multiple regression with dummy variables in SPSS with data from the general social survey (2012). SAGE Publications, United Kingdom (2015)
18. Esposito, D., Esposito, F.: Introducing Machine Learning. Microsoft Press (2020)
19. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. The MIT Press (2018)
20. Zaki, M. J., Meira, W.: Data Mining and Machine Learning: Fundamental Concepts and Algorithms. Cambridge University Press, United Kingdom (2020)
21. Jadhav, A. S.: A novel weighted TPR-TNR measure to assess performance of the classifiers. Expert Systems with Applications 152, 1–27, (2020)
22. Kuncheva, L. I., Arnaiz-González, Á., Díez-Pastor, J., Gunn, I. A.D.: Instance selection improves geometric mean accuracy: a study on imbalanced data classification. Progress in Artificial Intelligence 8, pp. 215–228 (2019)
23. Fotouhi, S., Shahrokh, A., Kattan,M. W.: A comprehensive data level analysis for cancer diagnosis on imbalanced data. Journal of Biomedical Informatics 90, pp. 1–30 (2019)
24. Fernandez, A., Garcia, S., Herrera, A., Chawla, N. V.: SMOTE for Learning from Imbalanced: Progress and Challenges, Making the 15 year Anniversary. Journal of Artificial Intelligence 61, pp. 863–905 (2018)
25. Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., Santos, J.: Cross-validation for imbalanced datasets: Avoiding Overoptimistic and Overfitting Approaches. IEEE Computational Intelligence Magazine 13(4), pp. 59–76 (2018)
26. Lopez-Roldan, P., Fachelli, S.: Metodología de la investigación social cuantitativa. Universitat Autonoma de Barcelona (2015)

ISSN 1870-4069

Implementación de sistema de visión inteligente para reconocimiento de rostros en robot de código abierto BOB

Carlos Leal-Beltrán[2], Juan Villegas-Cortez[1], Arturo Zúñiga-López[2], Salomón Cordero-Sánchez[3]

[1] Universidad Autónoma Metropolitana, Departamento de Sistemas, Unidad Azcapotzalco, México
[2] Universidad Autónoma Metropolitana, Departamento de Electrónica, Unidad Azcapotzalco, México
[3] Universidad Autónoma Metropolitana, Departamento de Química, Unidad Iztapalapa, México

car-lebe@hotmail.com, {juanvc, azl}@azc.uam.mx, scs@xanum.uam.mx

Resumen. En el presente trabajo se realiza la implementación de una interfaz entre un mecanismo móvil de código abierto y un sistema de autenticación de rostros para personas, ya que el diseño recuerda mucho a un juguete lo vuelve más amigable con niños y jóvenes que otro tipo de mecanismos y robots, lo que lo vuelve mas accesible y en consecuencia hace más fácil la interacción humano-computadora, pudiendo interfasarse con otros proyectos de evaluaciones pedagógicas, de educación o con otros sistemas de reconocimiento de personas, independientemente del lenguaje o propósitos de comunicación. Proponemos utilizar la interconexión, con la posibilidad de reconfiguración y mejora para facilitar la comunicación entre personas, a partir de recabar la información de rostros. El sistema consta de una tarjeta de sistema embebido Raspberry Pi Zero W, una cámara tipo CSI, y la implementación de una red neuronal embebida de código abierto también. Se ha realizado el ensamblado del mecanismo y la integración del sistema, logrando una operación fiable y las pruebas de reconocimiento están basadas en dos bases de datos de rostros probadas por la comunidad internacional de inteligencia artificial.

Palabras clave: Robótica, visión por computadora, redes neuronales profundas, reconocimiento de rostros.

Smart Computer Vision System Applied to Face Recognition in Open Source Robot BOB

Abstract. This work studies the implementation of an interface between an open code movil mechanism and a person face authentication system.

The design of it is similar to a child's play, which makes it more attractive to children and youths than any other type of mechanism and robot. In this sense, it is more accessible and makes human computer interaction easier. Also, stands out its capability to interface with other pedagogical projects and with other face recognition systems, regardless language or communication purposes. Based on collected face information data, it is proposed to utilize the interconnection with the possibility of continuous reconfiguration and improvement to facilitate communication among people. The system consists of a Raspberry Pi Zero W embedded card system, a CSI camera, and the implementation of a neural network embedded in open code. The assembly of the mechanism and its integration achieve a reliable operation and the recognition tests are based on two face databases tested by the international community of artificial intelligence.

Keywords. Robotics, computer vision, deep neural networks, pattern recognition.

1. Introducción

La ciencia informática camina actualmente hacia la creación de computadoras cada vez más rápidas, expertas y autónomas. Una de las empresas más ambiciosas consiste en dotar a las computadoras de la facultad de relacionarse con su entorno del mismo modo que lo hace un humano: a través de los sentidos. El proporcionar capacidad sensorial a una computadora es una tarea difícil ya que entran en juego los periféricos al microprocesador y sus tipos de interfase de comunicación, como son: los sensores, tarjetas adaptadoras de señal o el ruido del entorno. A pesar de los muchos inconvenientes, existe hoy un interés especial por dotar a las computadoras de uno de los cinco sentidos del hombre: la habilidad de ver. Es así que la visión artificial se aplica ya en procesos industriales en los que la simple detección de presencia no resulta una fuente de información suficiente [6].

En este trabajo proponemos implementar una metodología general de reconocimiento de rostros en un sistema embebido portable, con programación en código abierto y usando herramientas par lograr su interfase hacia otros sistemas de propósito específico. Aquí nosotros presentamos la propuesta implementada en un mecanismo de código abierto, BOB, y herramientas ampliamente conocidas por la comunidad científica del área de visión por computadora.

El resto del trabajo está organizado de la siguiente forma. En la Sección 2 presentamos el Estado del arte, en la Sección 3 la metodología de detalla con el propósito que pueda ser replicada por la comunidad científica e ingenieril interesada, en la Sección 4 presentamos la implementación en hardware, las pruebas realizadas hasta ahora y una discusión de resultados, y finalmente en la Sección 5 presentamos las conclusiones de nuestro trabajo.

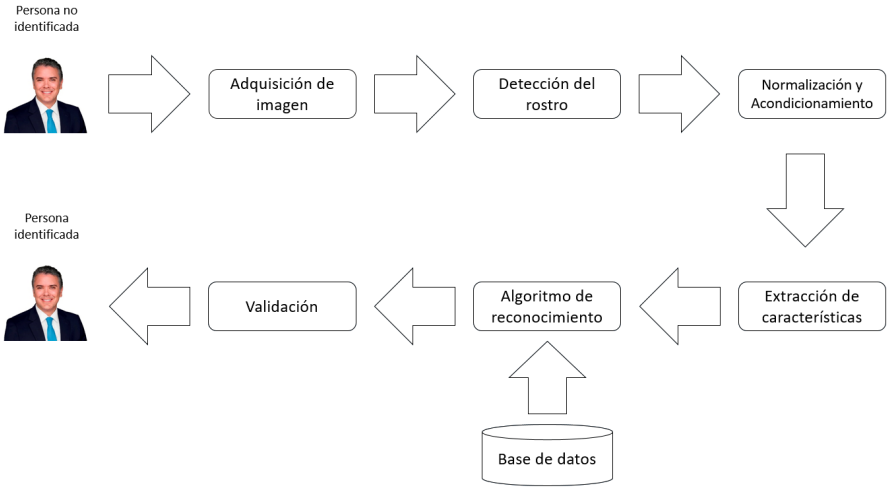


Fig. 1. Diagrama de funcionamiento de un sistema de autenticación de rostros.

2. Estado del arte

Los sistemas de visión integrados o sistemas de visión artificial incluyen sistemas de visión, tales como: cámaras inteligentes, tarjetas de adquisición de imágenes e interfaces de visión diseñados para una amplia variedad de aplicaciones como verificación de ensamble, rastreo de objetos y robótica guiada por visión, como puede verse en la página de referencia de la aplicación de visión por computadora en la industria por David Lowe en [5] y también tenemos ejemplos concretos en [3] [9]. En la Figura 1 se muestra un ejemplo de un sistema o metodología general de validación de rostros por reconocimiento facial. Es así que en general, se procesa la imagen de entrada en un sistema que puede categorizar un rostro mediante un clasificador de distancia no supervisado como por ejemplo el de K vecinos más próximos, $K-NN$ [2], extrayendo características y medidas de la imagen como un patrón para reconocer rostros por medio de una métrica con base a la cercanía de los patrones de los rostros. Estos datos se cotejan contra una base de datos ya existente previamente entrenada, y en el caso que encuentre quién es la persona de la cual se detectó el rostro por medio de una comparación basada en una métrica de distancia, detecte o identifique a la persona.

El reconocimiento de rostros, del inglés *face recognition* (FR), es una tarea que realizamos las personas de manera cotidiana y después de lo cual toman decisiones y actúan de diferente manera.

Por ejemplo, si se reconoce a una persona conocida se puede proceder a saludarla. Es así que el FR un proceso muy complejo que se define en varias partes, por un lado, la primera de ellas es el momento de tomar la observación o en nuestro caso la foto que se usara, ya que actualmente la mayoría de aplicaciones que usan FR ocupan una ventana de tiempo para decidir en qué

momento realizarán su funcionalidad [13]. Por otro lado, en nuestra propuesta aquí presentada, primero se usó sensores de proximidad, para determinar la ventana de observación en la que se realizará una captura del rostro mediante una cámara; posteriormente, después de la observación y adquisición de la foto de la persona, es el FR en la foto, en este paso también es donde se delimitan los parámetros que podrá tener la imagen como lo son el ángulo, cantidad de luz, uso de lentes y demás parámetros. Algo en lo que en la actualidad se realiza ocupando varias librerías de software para la detección de rostros que realizan el trabajo de normalizar y acondicionar la foto.

En este paso se utiliza una red neuronal artificial, que pueden ser del tipo *eigenfaces, fisherface,* etc. [14]. Estos algoritmos de reconocimiento, desde el reconocimiento de patrones, son aquellas instrucciones que están orientadas hacia la agrupación de las características principales en los patrones que nos permitirán distinguir una cara de otra, en este segmento podemos mencionar a los métodos basados en clasificadores bayesianos, redes neuronales artificiales o los algoritmos genéticos y la generalidad de los algoritmos evolutivos [9], que proponen soluciones muy sencillas, desde la generalidad de los Algoritmos Evolutivos, tal como tenemos un ejemplo muy concreto en [11].

El Aprendizaje Profundo, del inglés *Deep Learning* (DL), es una metodología que caracteriza a los objetos de estudio, como puede ser una imagen con todos sus pixeles y capas o canales, en un cúmulo de rasgos obtenidos por etapas sucesivas con un alto costo de cómputo. Desde el enfoque del DL, se considera que sea el propio algoritmo de aplicaciones sucesivas de extracción de características por operaciones de convolución discreta, que se conformen los patrones característicos del objeto de estudio [15]. En esta línea se cuenta con desarrollos abiertos para la comunidad científica en FR aplicando DL, específicamente hablamos de OpenFace [4], y en nuestra propuesta la retomamos para usarla en el sistema embebido o empotrado basado en Raspberry Pi. En el trabajo de [10] se tiene un ejemplo de este tipo de soluciones, basadas en sistemas embebidos de bajo coste y de redes neuronales profundas; pero ahora nosotros presentamos una implementación orientada a un abanico de posibilidades dentro de la educación y de apoyo a tareas más allá de la autenticación de un usuario o persona concreto. La mayor diferencia o aportación de este proyecto, es que implementa este tipo de soluciones en un mecanismo móvil que interactúa directamente con las personas y sus traducen en respuestas físicas.

3. Metodología

El siguiente paso es el procesamiento de todos estos datos y en el caso del trabajo que se realiza se eligió un dispositivo Raspberry que ya ha probado ser un dispositivo de alto rendimiento que después de pasar un entrenamiento en otro dispositivo de mayor capacidad, es capaz de lograr un reconocimiento facial de gran calidad debido a sus prestaciones que son muy buenas para su tamaño

[4] https://cmusatyalab.github.io/openface/

[10]. Como parte final se programó la acción resultante del mecanismo para el movimiento, dependiendo de si el resultado del reconocimiento facial es positivo o en caso contrario tomar otra acción. Para el entrenamiento y pruebas de la red neuronal se optó por las bases de datos de rostros ampliamente conocidas por la comunidad de Visión por Computadora y Reconocimiento de patrones: Yale y Ar, como se muestran en las Figuras 2 y 3 respectivamente. La primera, contiene imágenes de tamaño 320×243 pixeles en formato de archivo con compresión JPEG, con 13 individuos, con variaciones de iluminación y con 8 distintas expresiones faciales tales como de postura normal, feliz, triste, soñoliento, etc. La segunda es de fotografías de 83 individuos, todas en condiciones ideales, de frente y con expresiones faciales, y no es regular en la cantidad de muestras por persona para unos tiene 6 poses, mientras otros tienen 8, y también presentan iluminación variante, con resolución de 576×768 pixeles.

A continuación damos detalles de la implementación de la metodología en su arquitectura, componentes electrónicos y de sistema usados, así como de las etapas de la metodología que se desarrollaron en una computadora tipo PC, y las etapas desarrolladas en el sistema embebido Raspberry.

(a) **(b)** **(c)**

Fig. 2. Ejemplos de rostros de la base de imágenes Yale tomadas por el mecanismo.

(a) **(b)** **(c)**

Fig. 3. Ejemplos de rostros de la base de imágenes AR captadas por el mecanismo.

Arquitectura

El sistema que se implementa se muestra en la Figura 4, en un diagrama a bloques con los procesos a realizar. En este sistema se analiza el movimiento y velocidad del mecanismo una vez que se ensamble para verificar que este movimiento no afecte significativamente la eficiencia del experimento.

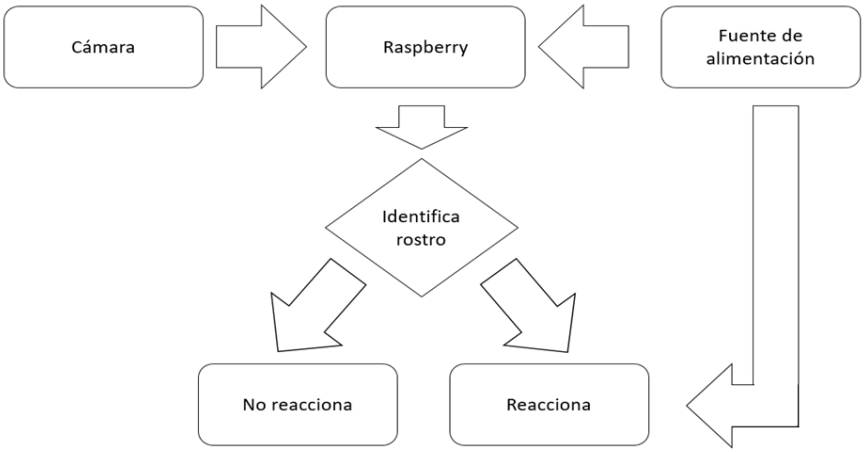


Fig. 4. Diagrama a bloques del funcionamiento del sistema propuesto.

Componentes

– Sistema embebido: Es el cerebro del proyecto, consiste en un sistema miniaturizado de control que mediante circuitos con interfaces y de procesamiento que se pueden adaptar a cualquier proceso que sea necesario.
Este tipo de sistemas compactos son muy útiles cuando se necesita procesamiento de datos en tiempo real y mas usado en aplicaciones móviles. La gran ventaja de estos sistemas es que son programables, lo que lo vuelven en un sistema muy elástico.
En cuanto a programación tenemos muchos lenguajes para programarlo y de diferentes tipos tales como lenguajes ensamblador, lenguajes orientados a objetos, etc. Con lo que tenemos diferentes herramientas para las aplicaciones que necesitemos y del cual se usará un lenguaje orientado a objetos Phyton.
Para aplicaciones con sensores de visión o cámaras útiles para el proyecto propuesto se recomiendan sistemas del tipo *Raspberry Pi Zero W*, que es el más adecuado por sus especificaciones, tales como son memoria, procesador e interfaces como se muestra en la Figura 5.

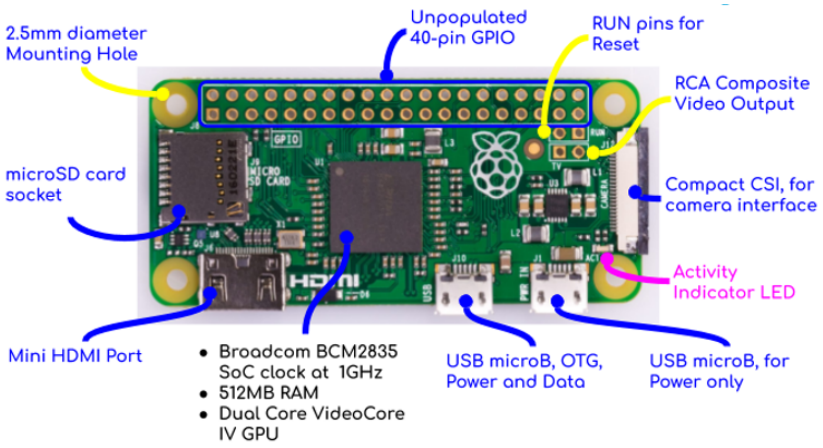


Fig. 5. Componentes de la Raspberry Pi Zero W.

- **Características del sistema embebido**
  - Procesador tipo: BCM 2835 SOC.
  - Velocidad de procesador: 1GHz.
  - Memoria RAM: 512MB.
  - Memoria ROM: micro-SD.
  - Puerto de Audio y Video: mini-HDMI.
  - Puerto USB: 1 x micro-B USB para datos (y alimentación también).
  - Entrada de alimentación: 1x micro-B USB for power (sin datos).
  - Interfaz para Camara: conector CSI para camera.
  - GPIO: con 40-pines.
  - Dimensiones: 65mm x 30mm x 5mm.

- Estructura de ensamble físico: Es la base del experimento y dependiendo de qué estructura se elija el sistema de movimiento. Se optó por una estructura impresa en 3D (Mecanismo Bob de código libre [4]), que servirá como caparazón o estructura externa de montaje para montar experimento. Ver Figura 6.
- Características de la cámara: Tipo: CSI, Resolución: 5 Mpx.
- Sistema de movimiento: Para trasladarse de un punto a otro contamos con un sistema de movimiento, basado en servomotores que en puntos específicos (articulaciones del mecanismo) ayudarán al movimiento de éste. Se proponen servomotores de 1.5 Kg. Véase la Figura 7.

Etapas desarrolladas en la computadora

Se realiza una normalización mediante un algoritmo para preparar los datos para la red neuronal que recibirá una biblioteca de imágenes de cuatro personas, cada

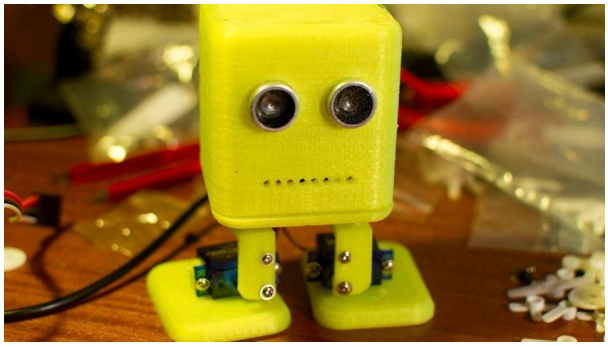


Fig. 6. Estructura física ensamblada del mecanismo Bob.

(a) La cámara usada.

(b) Servomotor de 1.5 Kg.

Fig. 7. Componentes adicionales.

persona de esta biblioteca tendrá cuatro fotos de 320×320 píxeles con diferentes gestos faciales, estas se categorizaran mediante una biblioteca de OpenFace.

Con esta información se entrena una red neuronal que sea capaz de tener una alta fiabilidad para determinar si la persona está dentro de la base de datos y así dar una respuesta positiva o negativa. Esta red neuronal es del tipo supervisada y con activador tipo Relu, lo que hace que tenga un porcentaje de exactitud mayor [10].

Etapas desarrolladas en la Raspberry

A partir de una foto tomada por el mecanismo cuando una persona se acerque a menos de un metro de distancia, empieza el procesamiento de la imagen primero normalizando la imagen e identificando el rostro por medio del algoritmo de la librería OpenCV.

Después de esto se procesara la imagen normalizada por una red neuronal ya entrenada en la computadora que procesara los puntos de interés entregados

por el paso anterior y con eso poder obtener si esta imagen pertenece a alguna persona de la base de datos [10].

Como último paso, en el caso positivo de que el proceso detecte una persona que corresponda a la base de datos el mecanismo realiza movimientos para acercarse a la persona y en caso contrario realiza otro movimiento.

4. Experimentación y resultados

El experimento a realizar se centra en que el mecanismo tome la foto, reconozca el rostro y determine si éste pertenece a una persona que esté en la base de datos aprendida en el entrenamiento avance hacia adelante, caso contrario deberá retroceder. Todo este movimiento lo realizara mediante los servomotores instalados en la parte inferior del mecanismo.

Para ambos escenarios la posición inicial es la misma, la foto es tomada a una distancia menor a un metro y a una altura en que se pueda observar el rostro como se muestra en la Figura 8.

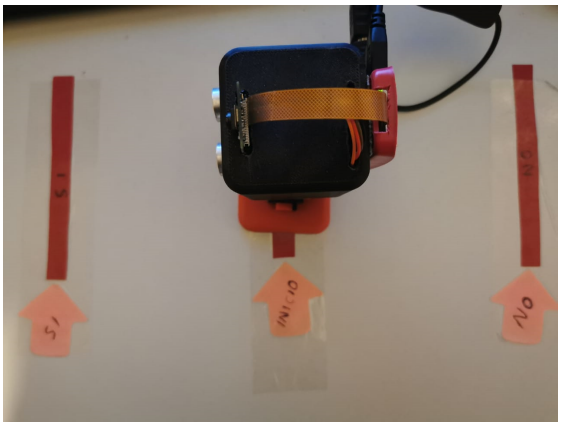


Fig. 8. Posición Inicial del mecanismo para el experimento.

Primer escenario

En este escenario el mecanismo se encuentra con una foto de una persona a menos de un metro de distancia, en este punto toma la captura de la persona como se muestra en la Figura 9.

Cuando la cámara toma el rostro comienza el proceso de reconocimiento, esto ocurre en el algoritmo que ya ha sido entrenado previamente en la computadora y fue migrado a la Raspberry, en cuanto el algoritmo reconoce el rostro el mecanismo avanzara. Como se muestran en la Figura 10a y 10b.

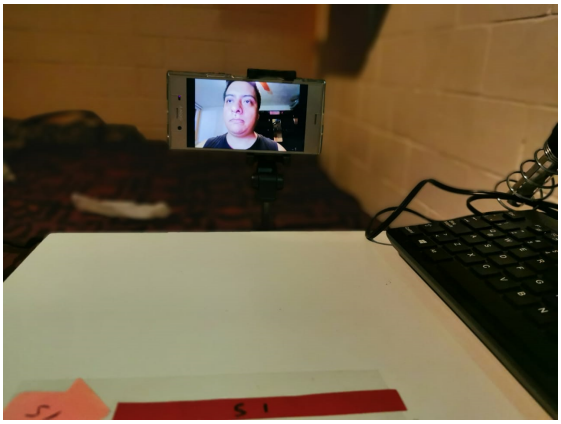
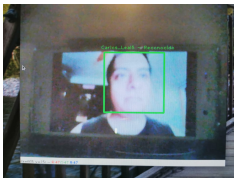
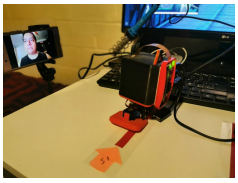


Fig. 9. Captura del rostro para el primer escenario.



(a) Reconocimiento y Validación de una persona reconocida.

(b) Posición inicial.



(c) Posición Final.

Fig. 10. Ejemplo experimental con la fotografía capturada en un dispositivo móvil y el proceso de reconocimiento de rostro en el mecanismo.

De esta manera podemos ver como el algoritmo de reconocimiento pre-entrenado válida que es una persona que está en la base de datos y el mecanismo realiza correctamente su rutina después de esta validación como se puede ver en la Figura 10c.

Segundo escenario

En este escenario el mecanismo se encuentra con una foto de otra persona que también está a menos de un metro de distancia y en este punto toma la captura de la persona como se muestra en la Figura 11.

Nuevamente la cámara captura el rostro y comienza el proceso de reconocimiento, en cuanto el algoritmo pre-entrenado reconoce que el rostro no coincide con alguno de la base de datos el mecanismo retrocederá. Como se muestran en la Figura 12a y 12b. De esta manera podemos ver como el algoritmo de

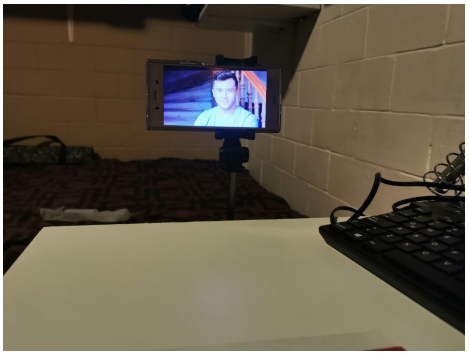
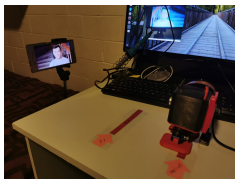


Fig. 11. Captura del rostro para el segundo escenario.

(a) Reconocimiento y Validación de una persona no reconocida.



(b) Posición inicial.

(c) Posición final.

Fig. 12. Proceso de reconocimiento del rostro en la pantalla y respuesta del mecanismo.

reconocimiento pre-entrenado válida que esta persona no está en la base de datos y el mecanismo realiza correctamente su rutina de retroceso después de esta validación como se puede ver en la Figura 12c.

4.1. Resultados

Presentamos los resultados obtenidos para el entrenamiento con las bases de datos consideradas, Yale y AR. Se puede apreciar en la figura 13 que los rostros se reconocen claramente y no se traslapan.

Se pudo validar ambos escenarios, el de reconocimiento de rostro cuando la persona es identificada como parte de la base de datos y en el que la persona es desconocida para el sistema, viendo que el mecanismo valido a la persona y después realizo la acción que tenía programada para cada escenario.

Un ejemplo del porcentaje de éxito del reconocimiento facial con el método seleccionado se puede observar en la siguiente Figura 14.

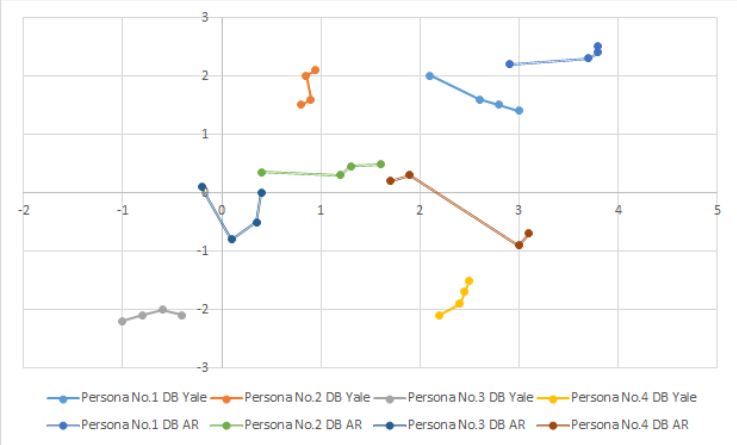


Fig. 13. Proyección de los vectores de los rostros, de la base de imágenes de Yale y AR.

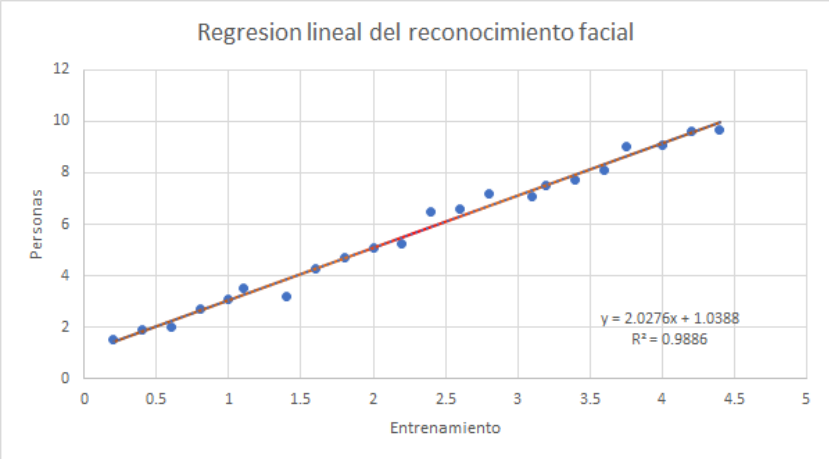


Fig. 14. Regresión lineal

5. Conclusiones

En este trabajo hemos presentado una implementación de un sistema de reconocimiento de rostros basado en una red neuronal convolucional, pre-entrenada y de dominio público, en un sistema embebido portable, de código abierto y con librerías ampliamente conocidas por la comunidad científica de Visión por Computadora y Redes Neuronales artificiales. Todo el sistema se pudo montar en un mecanismo móvil abierto a la comunidad académica para su experimentación libre, con el propósito de poder interactuar con personas, infantes o jóvenes, para

su identificación por medio del rostro y la respuesta con movimiento asociado a consentimiento por parte de las personas.

Si bien consideramos que la implementación puede tener múltiples propósitos, hemos mostrado la factibilidad de la implementación de hardware y software concreto para una tarea práctica de reconocimiento de rostros, y de que el re-entrenamiento de nuevos rostros es posible con corto tiempo y toda la implementación a muy bajo costo en comparación con otros sistemas de mecanismos móviles del tipo en el mercado.

Como trabajo futuro estamos desarrollando la interfaz de nuestro desarrollo con un mecanismo móvil propietario, a fin de poder extender el uso de los mecanismos para nuevos propósitos y también extender su vida útil para fines educativos y empresariales. Así también estamos considerando la implementación de una metodología modificada que permita una actualización de la red entrenada con bases de datos masivas en un corto tiempo bajo el esquema cliente-servidor.

Referencias

1. Ordieres-Meré, J., Limas, M., Ascacibar, F.J., Alba-Elías, F., González-Marcos, A., Pernía-Espinoza, A., Vergara, E.: Técnicas y algoritmos básicos de visión artificial. Recurso electrónico - En línea. Universidad de la Rioja (2006)
2. Fukunaga, K.: Introduction to statistical pattern recognition (2nd ed.). Academic Press Professional, Inc. (1990)
3. The Computer Vision Industry: https://www.cs.ubc.ca/ lowe/vision.html (2020)
4. MakerBob Education: https://www.thingiverse.com/thing:43708 (2019)
5. National Instruments Corp.: http://www.ni.com/es-mx/shop/select/machine-vision-category (2019)
6. Nalwa, V.S.: A Guided Tour of Computer Vision. Addison-Wesley Longman Publishing Co., Inc. (1993)
7. Pérez, M.A.C.: Reconocimiento robusto de rostros en ambientes dinámicos. Master's thesis, Universidad de Chile (2012)
8. Villanueva-Escudero, C., Villegas-Cortez, J., Zúñiga-López, A., Avilés-Cruz, C.: Monocular visual odometry based navigation for a differential mobile robot with android OS. Human-Inspired Computing and Its Applications, Lecture Notes in Computer Science, Vol. 8856, Springer International Publishing, pp. 281–292 (2014)
9. Olague, G.: Evolutionary Computer Vision: The First Footprints. Natural Computing Series. Springer-Verlag Berlin Heidelberg (2016)
10. Zuñiga Lopez, A., Villegas-Cortez, J., Avilés-Cruz, C., Rodríguez-Martínez, E., Ferreyra-Ramirez, A.: Reconocimiento de rostros por medio de Openface en una Raspberry PI. Research in Computing Science, 147, pp. 77–88 (2018)
11. Benavides-Alvarez, C.: Sistema no supervisado de clasificacion de rostros contecnicas basadas en CBIR. tesis para obtener el grado de maestro en ciencias ytecnologías de la información. Universidad Autónoma Metropolitana (2015)
12. Benavides-Alvarez, C., Villegas-Cortez, J., Román-Alonso, G., Aviles-Cruz, C.: Face classification by local texture analysis through CBIR and surf points. Revista IEEE América Latina, 14 (2016)

13. Benavides Alvarez, C., Villegas Cortez, J., Román-Alonso, G., Aviles-Cruz, C.: Reconocimiento de rostros a partir de la propia imagen usando tecnica CBIR. In: X Congreso Español sobre Metaheuristicas, Algoritmos Evolutivos y Bioinspirados (MAEB), Merida, Extremadura, Spain, Universidad de Extremadura, pp. 733–740 (2015)
14. Sulis Setiowati, Z., Legya Franita, E., Ardiyanto, I.: A Review of Optimization Method in Face Recognition: Comparison Deep Learning and Non-Deep Learning Methods. In: ICITEE (2017)
15. Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E.: Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience (2018)

ISSN 1870-4069

Generación de cargas horarias para estudiantes basadas en un algoritmo micro-genético

Brenda Sunuami González López, René Arnulfo García-Hernández

Universidad Autónoma del Estado de México,
Unidad Académica Profesional Tianguistenco,
México

brenda_sunuami@hotmail.com, renearnulfo@hotmail.com

Resumen. El problema de generación de cargas horarias para estudiantes consiste en asignar, entre una oferta de clases, a aquellas que cumplen en su conjunto con criterios que son relevantes para el estudiante y para su desempeño académico. Debido a la complejidad que implica la generación de cargas horarias de forma manual, actualmente varios estudiantes no consideraron una mejor opción. En este artículo se presenta un método basado en un micro algoritmo genético que considera criterios del estudiante, y del avance del estudiante de acuerdo a la curricula. Los resultados cuantitativos con el método propuesto muestran mejores resultados que la generación manual de la carga horaria. También los resultados cualitativos fueron buenos basados en los comentarios de los estudiantes que probaron el método propuesto.

Palabras clave: Carga horaria para estudiantes, optimización multi-criterio, micro algoritmo genético.

Timetable Generation for Students based on a Micro-Genetic Algorithm

Abstract. The problem of timetable generation for students consists of assigning, among a classes offer, those that accomplish as a whole the relevant criteria to the student and to their academic performance. Due to the complexity involved in generating the timetable manually, currently several students did not consider a better option. This article presents a method based on a micro-genetic algorithm that considers the criteria of the students, and the progress of the students according to the curricula. The quantitative results with the proposed method show better results than the manual generation of the timetable. Also, the qualitative results were good based on the comments of the students who tested the proposed method.

Keywords: Timetable for students, multi-criteria optimization, micro-genetic algorithm.

1. Introducción

La solución de problemas en el mundo real se ha visto beneficiada cada vez más por técnicas computacionales que le han permitido al humano, por un lado, tener mejores resultados y, por el otro, realizar la misma tarea en un menor tiempo (Ledeneva & Hernández, 2017). La generación de cargas horarias no es una excepción, los estudiantes enfrentan cada periodo de inscripción el problema de diseñar su carga horaria. Este tipo de problemas también conoce como o programación horaria (en inglés *timetabling*) (Hernández, 2008). Los problemas de esta área consisten en la asignación de recursos logísticos, humanos, materiales y de infraestructura a la programación de eventos en una calendarización que normalmente se repite por semana (González, 2018)

La mayoría de las instituciones universitarias cuentan con planes de estudio flexibles. Por lo que la programación horaria de cursos queda a cargo de los estudiantes donde el estudiante debe de elegir, entre una oferta de Unidades de Aprendizaje (también conocidas como clases o materias), aquellas que va cursar durante el siguiente período (Solano, 2008). La flexibilidad del plan de estudios trae consigo varios beneficios porque puede generarse una carga académica adecuada a las condiciones particulares de cada estudiante. Sin embargo, el número de condiciones individuales y la gran cantidad de Unidades de Aprendizaje (UAs) ofertadas generan una gran combinación de posibilidades.

En nuestra universidad, el estudiante genera su carga horaria de forma manual por lo que demora un largo lapso y en varios casos se obtiene una mala carga horaria por no satisfacer las condiciones del estudiante (UAEM, 2016). Una carga horaria incorrecta produce problemas académicos, alimenticios y económicos. Por ejemplo, si un estudiante selecciona todas las unidades de aprendizaje de manera continua entonces no tiene tiempo para alimentarse adecuadamente. Otro ejemplo es cuando un estudiante tiene que asistir a la universidad más días de los necesarios por lo que afecta en los recursos de tiempo y económicos empleados. Para conocer cuáles son los criterios involucrados en la generación de cargas horaria se hicieron encuestas tanto a los estudiantes como a los tutores encargados de guiar o los estudiantes en su avance en el plan de estudios.

En este trabajo se propone un método para generar automáticamente cargas horarias bajo los criterios que común mente utilizan los estudiantes y los tutores para la generación de cargas horarias.

2. Trabajos relacionados

En nuestra revisión del EA no se encontró un trabajo enfocado en la problemática de asignación de carga horaria para estudiante, sin embargo, se encontraron trabajos que se enfocan en la asignación de recursos para la institución y docentes. En general, los trabajos encontrados trabajan con técnicas evolutivas.

El cómputo evolutivo consiste en un proceso basado en la evolución, con posibles soluciones a un problema, las cuales se modifican o evolucionan obteniendo cada vez mejores resultados (Holland, 1992).

Tabla 1. Trabajos relacionados a la generación de cargas horarias.

Investigación	Técnica empleada	Aplicación
Programación óptima de horarios de clase usando un algoritmo memético (Granada, 2006).	Algoritmo memético	Institución
Programación de horarios de clases y asignación de salas para la facultad de ingeniería de la universidad Diego Portales mediante un enfoque de programación entera (Hernández, 2008).	Programación entera	Institución
Algoritmo basado en búsqueda tabú para el problema de asignación de Horarios de Clases (Ramos, 2012).	Búsqueda Tabú	Institución
Programación de horarios escolares basados en Ritmos Cognitivos usando un Algoritmo Genético con ordenamiento no dominante, NSGA-II (Suárez, 2013).	Ritmos Cognitivos	Profesores

De acuerdo con el autor Andaluz en 2004, el cómputo evolutivo es una rama de la computación que engloba técnicas que simulan la evolución natural, consiste en un proceso basado en la evolución de un número de posibles soluciones a un problema, las cuales se modifican o evolucionan por medio de operadores denominados genéticos, por ejemplo; réplica cruzamiento y mutación (Coello & Zacatenco, 2005). Algunas técnicas empleadas para la asignación de horarios son: camino de hormigas (Díaz, 2013), algoritmos genéticos (Broca, 2016) y algoritmos meméticos (Gómez, 2006)

Existen investigaciones enfocadas en la optimización multi-objetivo con distintas herramientas matemáticas y computacionales. En la Tabla 1 se presentan algunos de los trabajos que emplearon el uso de herramientas computacionales para la generación de horarios. Cabe mencionar que estos trabajos están enfocados en la asignación de cargas horarias de profesores o instituciones, lo cual difiere de las condiciones del estudiante que se describen en la siguiente sección.

3. Descripción formal de las condiciones en la generación de cargas horarias

Los siguientes criterios fueron obtenidos a partir de encuestas realizadas a estudiantes pertenecientes a la afiliación de los autores de este artículo. Además de los criterios obtenidos de los estudiantes, también se obtuvieron dos criterios, uno utilizado por el tutor académico del estudiante que permite guiar y verificar el buen desempeño académico del estudiante a su cargo, y el otro estipulado por el plan de estudios. A continuación, se definen cada uno de los criterios que se deben considerar para realizar la generación de cargas horarias de estudiantes para este trabajo:

a) Condición de Tiempo por Semana (CTS): La condición del tiempo disponible por semana (medido en horas) varía de acuerdo con cada estudiante. Por ejemplo; existen estudiantes que trabajan o tienen labores externas a la institución, por lo que solo pueden destinar un número máximo de horas.
b) Condición de Horas Traslapadas (CHT): Un estudiante no puede tomar dos unidades de aprendizaje al mismo tiempo (horas traslapadas). Un claro ejemplo de

esto es un estudiante que selecciona dos unidades de aprendizaje en el mismo día y la misma hora, lo que provoca un serio problema, porque no puede estar en ninguna de las dos clases al mismo tiempo. Un traslape parcial puede repercutir seriamente en el rendimiento escolar del estudiante.

c) Condición de Unidades de Aprendizaje que Puede y Debe Cursar el estudiante (CUAPDC): El tutor académico debe vigilar el avance curricular de cada estudiante para poderle sugerir posibles cargas horarias a seleccionar. Cuando el avance curricular de un estudiante se desfasa del avance que está programado en el plan de estudios, es importante que las UAs más retrasadas tengan una mayor prioridad para el estudiante.

d) Condición de Créditos Máximos (CCMAX) y Mínimos para cada semestre (CCMIN): El estudiante tiene un mínimo y máximo de créditos que cumplir segú se especifica en el plan de estudios, pero este rango puede acotarse todavía más si el estudiante lo desea. Algunos estudiantes refirieron que este parámetro es importante porque parte del costo de inscripción se basa en el número de créditos de la carga horaria diseñada.

Otras condiciones que se dan por defecto:

e) Condición de generación de respuesta en tiempo real: El tiempo real es un factor de suma importancia, ya que el estudiante en el momento que desee una sugerencia de carga horaria, podrá obtenerla. Esta condición cobra relevancia minutos antes de la inscripción del estudiante debido a que las UAs disponibles pueden cambiar de horario, cancelarse u ofertarse; por lo que es importante contar con la sugerencia de una carga horaria en un tiempo razonable (menor a 60 minutos).

f) Dos unidades de aprendizaje no pueden ser seleccionadas en el mismo día, ni en el mismo horario.

g) La institución brinda los horarios de clases.

h) El estudiante debe especificar el número de horas semanales de clase que puede asistir a la institución.

4. Método propuesto

Los algoritmos genéticos están basados en el proceso genético de los organismos vivos (Zulueta, 1921) porque simulan el proceso de la evolución natural (Kuri, 2000). El algoritmo genético cuenta con una población la cual se divide en individuos de donde se seleccionan los mejores para crear nuevos individuos, realizando así una generación nueva (Pérez, 2011). Estos pretenden realizar la búsqueda de los mejores resultados optimizando todas las posibles combinaciones (Gestal, 2010). Generalmente, en problemas de búsqueda y optimización de parámetros se utilizan métodos adaptativos basados en la reproducción sexual y en el principio de supervivencia del más apto (Andaluz, 2004).

Un micro algoritmo genético tiene la particularidad de tener una población muy pequeña.

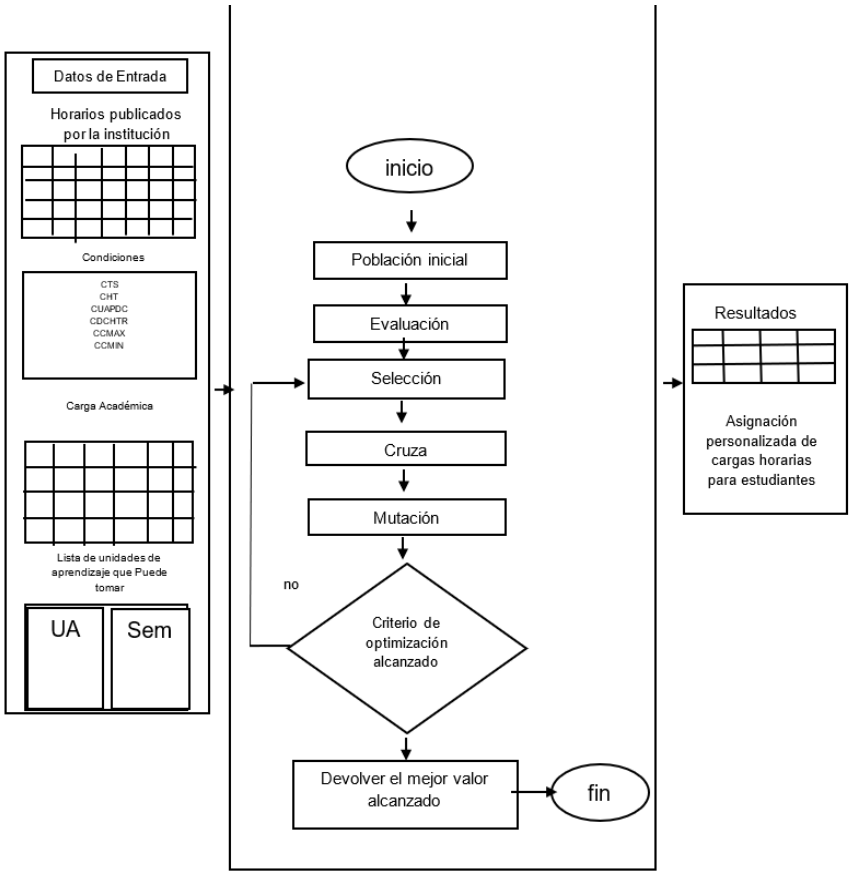


Fig. 1. Método propuesto.

Esta idea fue obtenida por algunos resultados obtenidos por Goldberg & Holland, 1988 en el cual se mostró que con una población de tres individuos era suficiente para converger, sin importar la longitud del cromosoma o individuo.

En el caso de esta investigación, se emplea un micro algoritmo genético con el objetivo de implementarse en dispositivos móviles con poco espacio de almacenamiento sin descuidar la calidad de los resultados. En la figura 1 se muestra el método propuesto implementado para la asignación de cargas horarias para estudiantes.

4.1 Codificación del individuo

Para la población inicial es importante decir que la longitud del individuo está relacionada con el número de UAs ofertadas que el estudiante puede cursar. La codificación del individuo utilizada es binaria, donde cada gen indica si esa unidad de aprendizaje si (1) es o no (0) considerada para formar parte de la carga horaria propuesta.

4.2 Función de evaluación

La función de aptitud considera las condiciones descritas en la sección 3, las cuales son normalizadas en un rango de 0 a 1 para poder llevar a cabo la minimización del error que presenta un individuo (ind). La Función de Aptitud (FA) del micro algoritmo genético se formuló como una función multi-criterio, como se describe en ec. (1):

$$FA\ (ind) = \propto CTS + \beta CHT + \gamma CUAPDC + \lambda CTR + \psi CCMIN + \theta CCMAX\ , \quad (1)$$

donde las variables α, β, γ, ε, λ, ψ, θ, están en un rango entre 0 y 1; y tiene la función de dar mayor o menor peso a las condiciones. Es por ello que la suma de las variables debe dar un resultado igual a 1:

a) ***CTS.*** Para la condición de tiempo por semana ec. (2), el primer error calcula el Número de Horas ($\boldsymbol{NH}$) del individuo ($\boldsymbol{ind}$) que se exceden de acuerdo con las Horas Máximas disponibles por Semana ($\boldsymbol{HMS}$) que tiene el estudiante:

$$\boldsymbol{CTS(ind) = NH(ind) = \frac{\begin{cases} NH(ind) > HMS & NH(ind) - HMS \\ En\ otro\ caso & 0 \end{cases}}{HMS}}. \quad (2)$$

b) ***CHT***. Para la condición de horas traslapadas se modela el error como el Número de las Horas Traslapadas ($\boldsymbol{NHT}$) propuestas por el individuo. Este error se normaliza con $\boldsymbol{HMS}$, es decir:

$$CHT = \frac{NHT}{HMS}. \quad (3)$$

c) ***CUAPDC***. Para la Condición de Unidades de Aprendizaje que Puede y Debe Cursar el estudiante se considera, de acuerdo al plan de estudios, el número de semestre en que debe de ser cursada la UA. La ec. (4) da prioridad a las UAs más rezagadas propuestas por el individuo ($\boldsymbol{ind}$) de tamaño $\boldsymbol{n}$:

$$\sum_{i=1}^{n} Semestre\ (ind[UA_i]). \quad (4)$$

Para normalizar este valor se calcula como límite menor la sumatoria de las n unidades de aprendizaje de Semestres Menores ($\boldsymbol{S.Menores}$) que puede cursar el estudiante de la Lista de UAs Posibles ($\boldsymbol{LUAP}$) y como límite mayor la sumatoria de las n unidades de aprendizaje de Semestres Mayores que puede cursar el estudiante de la $\boldsymbol{LUAP}$:

$$CUAPDC(ind) = \frac{\sum_{i=1}^{n} Semestre\ (ind[UA_i]) - \sum_{i=1}^{n} S.Menores\ (LUAP(i))}{\sum_{i=1}^{n} S.Mayores(LUAP(i)) - \sum_{i=1}^{n} S.Menores\ (LUAP(i))}. \quad (5)$$

d) ***CCMAX&CCMIN***. Condición de créditos máximos ($\boldsymbol{C_{max}}$) y créditos mínimos ($\boldsymbol{C_{min}}$) para generar cargas horarias (de acuerdo con el plan curricular, que además puede ser ajustado por el estudiante, ya que debido al número de créditos puede o no aumentar el costo de pago de inscripción por semestre). El

error de esta condición evalúa en ec. (6) qué tan alejado está el Número de Créditos ($\boldsymbol{NC}$) propuesto por el individuo ($\boldsymbol{ind}$), de los rangos permitidos por $\boldsymbol{C_{max}}$ y $\boldsymbol{C_{min}}$:

$$CCMIN\&CCMAX(ind) = \begin{cases} NC(ind) > C_{max} & \frac{NC(ind) - C_{max}}{C_{max}}, \\ NC(ind) < C_{min} & \frac{C_{min} - NC(ind)}{C_{min}}, \\ C_{min} \leq NC\ (ind) \leq C_{max} & 0\ . \end{cases} \quad (6)$$

5. Experimentación

Para la experimentación se trabajó con 100 estudiantes de la universidad de los autores (UAEMex), con una muestra de 3 generaciones de cargas horarias para estudiantes. Todos los experimentos se generaron con los mismos parámetros del micro algoritmo genético. Cabe mencionar que los estudiantes tienen mínimo dos posibilidades de horario que brinda la institución, del cual se puede realizar la generación de su carga horaria tomando en cuenta las condiciones del estudiante. El E1 forma parte de la comunidad estudiantil de la UAEMex, cursa el 10° semestre de su carrera y tiene una situación académica irregular. Por lo cual el estudiante debe seleccionar Unidades de Aprendizaje de diferentes semestres.

Tabla 2. Las UAs (Unidades de aprendizaje) traslapadas se indica con color gris, el resto de las UAs se muestran con colores cualesquiera.

UA2
UA2
UA4
UA4

Tabla 3. Se muestran las condiciones que requieren los estudiantes.

Condición	**E1**	**E2**	**E3**
Horas máximas semanalmente	37	20	0
Unidades de aprendizaje a cursar	10	5	5
Créditos mínimos	39	42	43
Créditos máximos	17	17	17

Tabla 4. Se muestran las definiciones de las abreviaturas que se muestran enseguida.

UA	**Unidad de aprendizaje**
E1	Estudiante 1
E2	Estudiante 2
E3	Estudiante 3

En la Tabla 5 se muestra el horario estructurado por el primer estudiante dónde en color gris pueden apreciarse las UAs que son traslapadas para el (E1).

Tabla 5. Carga horaria diseñada por el E1.

Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	UA1	UA3	UA6	UA7		
8:00 - 9:00						
9:00 - 10:00	UA2	UA3		UA7		
10:00 - 11:00		UA4		UA5		
11:00 - 12:00		UA4		UA5		
12:00 - 13:00						
13:00 - 14:00		UA5				

Tabla 6. Carga horaria diseñada por el método propuesto para el E1.

Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	UA1	UA3	UA6		UA7	
8:00 - 9:00						
9:00 - 10:00	UA2	UA4				
10:00 - 11:00						
11:00 - 12:00		UA5	UA3			
12:00 - 13:00			UA4			
13:00 - 14:00						

Tabla 7. Carga horaria diseñada por el E2.

Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00	UA6	UA2	UA4		UA1	
8:00 - 9:00						
9:00 - 10:00						
10:00 - 11:00			UA5	UA3		
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00						

En la Tabla 6 se muestra la carga horaria diseñada por el método propuesto para el E1 donde se obtuvo un resultado mejor al del estudiante. En la tabla 7 se muestra la carga horaria realizada por el segundo estudiante (E2). En la Tabla 8 se muestra la carga horaria diseñada por el método propuesto para el E2. El resultado obtenido es equivalente al horario realizado manualmente por el E2.

En la Tabla 9 se muestra la carga horaria diseñada por el tercer estudiante (E3). En la Tabla 10 se muestra la carga horaria diseñada por el método propuesto para el E3 dónde se obtuvo un resultado igual al realizado por estudiante manualmente.

Tabla 8. Carga horaria diseñada por el método propuesto para el E2.

Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00						
8:00 - 9:00		UA1	UA4		UA2	
9:00 - 10:00	UA5					
10:00 - 11:00						
11:00 - 12:00			UA6	UA3		
12:00 - 13:00						
13:00 - 14:00						

Tabla 9. Carga horaria diseñada por el E3.

Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00		UA7				
8:00 - 9:00	UA6		UA3		UA2	
9:00 - 10:00				UA5		UA1
10:00 - 11:00						
11:00 - 12:00		UA4				
12:00 - 13:00						
13:00 - 14:00						

Tabla 10. Carga horaria diseñada por el método propuesto para el E3.

Hora/Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 - 8:00		UA7				
8:00 - 9:00	UA6		UA3		UA2	
9:00 - 10:00				UA5		UA1
10:00 - 11:00						
11:00 - 12:00		UA4				
12:00 - 13:00						
13:00 - 14:00						

6. Desempeño de la evolución

En la siguiente gráfica se muestra la evolución del micro algoritmo genético tomando en cuenta los criterios mencionados en función de evaluación:

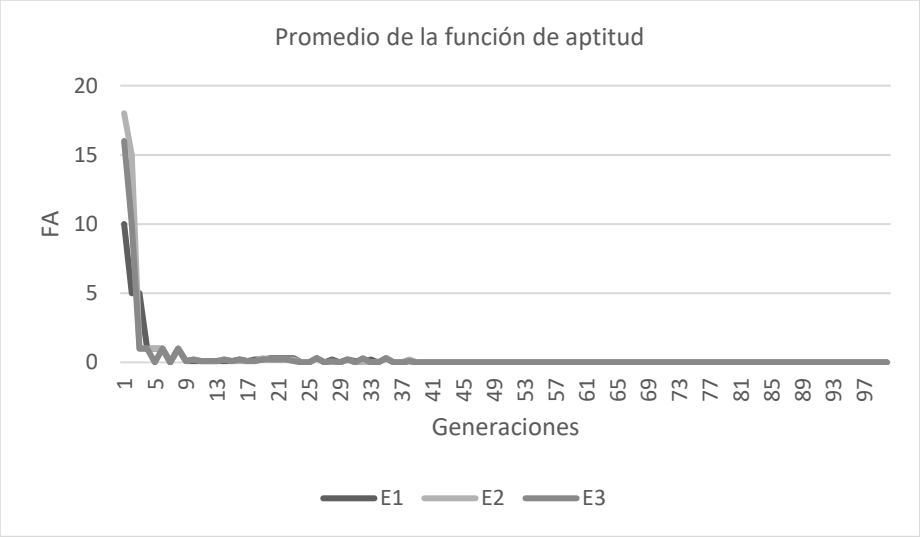


Fig. 1. Población de 5 individuos, 100 generaciones, probabilidad de cruza 80%, probabilidad de mutación 0.2%.

Fig. 2. Con una muestra de 100 estudiantes, el método propuesto generó un horario igual (25%), equivalente (25%) o mejor (50%) que el estudiante.

En la gráfica 2, se puede observar la aptitud que el micro algoritmo genético genera para el conjunto de condiciones que solicitan los tres estudiantes anteriormente mencionados. Con una población de 100 individuos se llegó a los siguientes resultados, en un 50% se asignaciones de acuerdo con los estudiantes, el método propuesto realiza una buena generación de carga horaria, tomando en cuenta los criterios del estudiante.

Estos porcentajes indican que en el 50 % el método propuesto ofrece cargas horarias que satisfacen las condiciones del estudiante, en los porcentajes que corresponde a cargas horarias equivalentes e iguales, indica que quizá no hay más posibles combinaciones en las UAs y con las condiciones que establece el estudiante.

Gracias a la evolución se ha podido conocer que con una población pequeña se obtienen buenos resultados cubriendo los criterios del estudiante (mencionados en la Tabla 3).

No se encontraron trabajos en el estado del arte que atiendan cargas horarias en el estudiante, se encontraron trabajos que realizan asignación de aulas pero en esos trabajos no se toman en cuenta criterios relacionados a los que se emplean en la presente investigación.

Con un micro algoritmo genético, se logró reunir una descripción formal de las condiciones reales que en general pueden ser útiles para aplicar el método propuesto a diferentes instituciones. Todos los criterios empleados en el presente trabajo son importantes en para el desempeño académico del estudiante.

7. Conclusiones

El problema que enfrenta el estudiante de asignar una carga horaria es muy importante, ya que actualmente existen muy pocos trabajos enfocados a los estudiantes, en su mayoría realizan la asignación de horarios para escuelas y profesores.

Con el presente trabajo se demuestra que con el uso de un micro algoritmo genético y optimización multi-criterio es posible asignar cargas horarias para estudiantes. De acuerdo a la experimentación se vio que con el método propuesto en todos los casos asignó cargas horarias mejores, equivalentes, iguales, cabe resaltar que nunca generó una carga horaria peor

Como trabajo futuro se pretende llevar el sistema a una aplicación móvil apta para cualquier plataforma, misma en la que ya se está trabajando.

Referencias

1. Andaluz, A.M.: Algoritmos evolutivos y algoritmos genéticos. Universidad Carlos III de Madrid, pp. 1–14. http://www.it.uc3m.es/~jvillena/irc/practicas/estudios/aeag (2004)
2. Broca, J.M.: Algoritmos genéticos en la generación de horarios escolares. Atizapán de Zaragoza (2016)
3. Coello, C.: Introducción a la optimización multiobjetivo usando metaheurísticas (2013)
4. Díaz, M.: Implementación del algoritmo de las Hormigas MAX-MIN para la selección de rutas de distribución (2013)
5. Gestal, M., Rivero, D., Rabuñal, J., Dorado, J., Coruña, A.P.A.: Introducción a los algoritmos genéticos y la programación genética. Galeon. Com. http://www.galeon.com/dantethedestroyer/algoritmos.pdf (2010)
6. Gómez, S.M.G.: Programación óptima de horarios de clase usando un algoritmo memético. Scientia et Technica, XII (30), pp. 255–260 (2006)
7. Goldberg, H.: Enhancement of xylanase production by sol-gel immobilization of Aspergillus awamori K-1. Bulgarian Journal of Agricultural Science, 19, pp. 117–119 (1988)

8. González, B.S.: Asignación de cargas horarias para estudiantes mediante un algoritmo genetic. 74 (2018)
9. Granada, M., Toro, E.M., Franco, J.F.: Programación óptima de horarios de clase usando un algoritmo memético. Scientia et Technica, XII (30), pp. 255–260 (2006)
10. Hernández, R., Miranda, P,J., Rey, P.A.: Programación de horarios de clases y asignación de salas para la facultad de ingeniería de la universidad diego portales mediante un enfoque de programación entera. Control, pp. 121–141 (2008)
11. Holland, J.: Algoritmos genéticos. Investigación y Ciencia, pp. 38–45 (1992)
12. Kuri, A.: Algoritmos Genéticos. http://cursos.itam.mx/akuri/PUBLICA.CNS/2000/AGS.PDF (2000)
13. Ledeneva, Y.N., Hernández, R.A.G.: Generación automática de resúmenes [Automatic generation of text summaries challenges, proposals, and experiments] (2017)
14. Pérez, C.: Un algoritmo genético para un problema de horarios con restricciones especiales. Revista de Matemática: Teoría y Aplicaciones, 18(2), pp. 215–229. http://www.redalyc.org/articulo.oa?id=45326930002 (2011)
15. Ramos, F.R.: Sistema para la generación de horarios académicos en instituciones universitarias usando algoritmo Tabú, 155. http://cybertesis.unmsm.edu.pe/xmlui/bitstream/ handle/cybertesis/4913/Ramos_mf.pdf?sequence=1 (2012)
16. Solano, Y.: Implementación de un algoritmo genético para la asignación de aulas en un centro de estudio. Uniciencia, 22, pp. 115–121 (2008)
17. Suárez, G.C.: Programación de horarios escolares basados en ritmos cognitivos usando un algoritmo genético de clasificación [Scheduling of school hours based on cognitive rhythms using a non-dominated sorting genetic algorithm]. NSGA-II, 24(1), pp. 103–114 (2013)
18. UAEM: Tutor Académico UAEM. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0ahUKEwi30pmMmLDbAhUUNn0KHf1PD08QFghHMAM&url=http%3A%2F%2Fri.uaemex.mx%2Fbitstream%2Fhandle%2F20.500.11799%2F66704%2Ftesis%2520final-split-merge.pdf%3Fsequence%3D1&usg=AOvVaw3VqNk2x8dxFnZ (2016)
19. Weare, R.F., Burke, E.K., Elliman, D.G.: A hybrid genetic algorithm for highly constrained timetabling problems. Science, pp. 605–610 (1995)
20. Zulueta, A.: El origen de las especies por medio de la selección natural. 9, pp. 55–62 (1921)

ISSN 1870-4069

Aprendizaje automático aplicado en física: Una revisión de la literatura científica

Iris Iddaly Méndez-Gurrola[1], Abdiel Ramírez-Reyes[1], Román Anselmo Mora-Gutiérrez[2]

[1] Universidad Autónoma de Ciudad Juárez, México
[2] Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México

iddalym@yahoo.com.mx, adbiel.ramirez@uacj.mx, mgra@correo.azc.uam.mx

Resumen. El Aprendizaje Automático es una rama dentro de Inteligencia Artificial que utiliza algoritmos y herramientas para lograr predicciones y juicios, analizando y aprendiendo de cantidades de datos históricos y sintéticos, lo cual puede ayudar por ejemplo en la toma de decisiones. Esta rama se ha abierto paso en muchas disciplinas científicas durante los últimos años. En el presente artículo se realiza una revisión de la literatura en la que se utiliza el Aprendizaje Automático en Física, se resaltan los desarrollos que ha habido en cuanto a la interrelación de ambas disciplinas durante los años 2005-2020, esto incluye desarrollos en la aplicación de técnicas de Aprendizaje Automático a varios dominios de la Física. Primeramente, se brinda una noción básica de Aprendizaje Automático la Física. Seguidamente se centra en organizar cada una de las publicaciones encontradas por categorías o áreas de la Física, para brindar un panorama general de áreas con mayores aplicaciones del Aprendizaje Automático en Física. Destaca en la investigación, el crecimiento acelerado de publicaciones de 2017 a la fecha. Finalmente, se discuten los diversos hallazgos en la aplicación de métodos y técnicas de Aprendizaje Automático en Física, y aquellos orientados a la búsqueda de conocimiento de frontera en la Física.

Palabras clave: Aprendizaje automático, inteligencia artificial, física, revisión de literatura, física aplicada.

Machine Learning Applied in Physics: A Review of the Scientific Literature

Abstract. Machine Learning is a branch within Artificial Intelligence that uses algorithms and tools to make predictions and judgments, analyzing and learning from amounts of historical and synthetic data, which it can help for example in decision making. This branch has been used in many scientific disciplines in recent years. In this article a literature review is made in which the Machine Learning is used in Physics, highlights the developments that have been in the

interrelation of both disciplines during the years 2005-2020, this includes developments in the application of Machine Learning techniques to various domains of Physics. First, a basic notion of Machine Learning is made and later of Physics. Then it focuses in organizing each of the publications found by categories or areas of Physics, which allows to provide an overview of areas with higher applications of Machine Learning within Physics. Highlight in the investigation, the accelerated growth of publications from 2017 to the date. Finally, various findings in the application of methods and techniques of Machine Learning in Physics are discussed, and those oriented to the search of new knowledge in frontier Physics.

Keywords: Machine learning, artificial intelligence, physics, literature review, applied physics.

1. Introducción

El Aprendizaje Automático es un área dentro de la Inteligencia Artificial (IA) que abarca una amplia gama de algoritmos y herramientas de modelado utilizados para una amplia gama de tareas de procesamiento de datos, que ha entrado en la mayoría de las disciplinas científicas en los últimos años, las Ciencias Físicas no han sido la excepción. En Física se desea comprender los mecanismos de la naturaleza y se utiliza su propio conocimiento, inteligencia e intuición para formar sus modelos. Por tanto, ambas disciplinas están preocupadas por el proceso de recopilación y análisis de datos para diseñar modelos que puedan predecir el comportamiento de sistemas complejos. El uso de la IA en las ciencias Físicas es algo que se ha planteado desde hace años; se sabe de la importancia de la IA para mostrar fenómenos, comportamientos o propiedades que no pueden ser inferidos a partir de metodologías tradicionales [1,2], pero que inicialmente tenían la dificultad del insuficiente procesamiento de cómputo o el poco desarrollo de los algoritmos.

La intersección entre ambas disciplinas es un tema de amplio interés actual en la Física, pues esto ofrece nuevas estrategias para resolver los problemas de alta complejidad que hay en las ciencias Físicas, pero a la vez importante para el desarrollo de las técnicas del Aprendizaje Automático. Ahora bien, ¿Cómo identificar estás intersecciones?, ¿Qué avances se han hecho al respecto? y ¿Qué hallazgos se han encontrado de esta conjunción? Para responder a estas y otras preguntas es necesario hacer una revisión de la literatura. Construir una investigación y relacionarla con el conocimiento existente es cada vez más una tarea compleja, debido a que la generación de conocimiento es de forma acelerada, por tal motivo, es tan importante realizar una revisión de literatura como método de investigación. La revisión de la literatura es, en términos generales, una forma sistemática de recopilar y sintetizar investigaciones previas y que puede llegar a ser una base firme para el avance en el conocimiento [3].

Este trabajo comienza dando una breve descripción de Aprendizaje Automático, y se mencionan algunos de los paradigmas de aprendizaje que se encuentran dentro de esta área y algunos de los campos donde se aplica. En la sección 3 se define la Física y se muestra una categorización de las áreas que la integran. La sección 4 especifica la metodología utilizada para la revisión de la literatura en este trabajo. En sección 5 se muestra el análisis de datos y los resultados de la investigación, para finalmente en la sección 6 se abordan las conclusiones y perspectivas del trabajo.

2. Aprendizaje automático

La IA se podría decir que es una colección de varias herramientas analíticas que colectivamente intentan imitar el comportamiento del ser humano. En los últimos años, la IA ha madurado a un conjunto de herramientas analíticas que facilitan la resolución de problemas que antes eran difíciles o imposibles de resolver. La tendencia ahora es la integración de estas herramientas, con tecnologías convencionales como el análisis estadístico, para construir sistemas sofisticados capaces de resolver problemas desafiantes [4]. La IA se divide básicamente en dos tipos: IA débil e IA fuerte.

Dentro de la IA débil sobresalen las áreas de: a) Aprendizaje Automático, b) Métodos Probabilísticos, c) Computación Evolutiva, d) Sistemas Difusos, entre otras. Este trabajo se centra en el ***Aprendizaje Automático*** o Machine Learning (ML), el cual es capaz de simular un comportamiento utilizando algoritmos que requieren una gran cantidad de datos. En la mayoría de las ocasiones el algoritmo "aprende" qué decisión adoptar y crea un modelo. Pertenecen a esta rama las redes neuronales, modelos basados en KNN o en regresión logística, árboles de decisión, máquinas de soporte vectorial, entre otros [5].

Con relación al Aprendizaje Automático entre los modelos utilizados para entrenar los algoritmos se ubican los modelos lógicos, los modelos geométricos y los modelos probabilísticos. Además, entre los paradigmas de aprendizaje se encuentran [5]:

- ***Aprendizaje supervisado***. Basándose en el pasado el modelo elabora una función que relaciona las variables predictoras con las de clase. Esta función es utilizada posteriormente para realizar predicciones (clasificación).
- ***Aprendizaje no supervisado***. Tiene el objetivo de la obtención de grupos, tal que en cada uno de ellos haya instancias homogéneas, mientras que los grupos son heterogéneos entre sí. En este aprendizaje no hay información del pasado, es el propio modelo el encargado de realizar sus propias divisiones.
- ***Aprendizaje por refuerzo***. El algoritmo aprende, no con la información previa que se le hubiera proporcionado, sino con su interacción con el mundo que le rodea, por tanto, se produce una retroalimentación que va modificando y refinando su comportamiento.

Los campos de aplicación de la IA y en particular del Aprendizaje Automático son cada vez más diversificados, entre ellos se encuentran las aplicaciones en la Física y sus diferentes áreas que la integran.

3. Física

La Física es una ciencia natural cuyo objeto de estudio es la materia, la energía, los campos y sus interacciones entre sí, en escalas microscópicas o macroscópicas. Posee un marco lógico-matemático, con el que formula y cuantifica principios físicos. Es una ciencia fundamental, sobre la que se fundamentan otras ciencias naturales y aplicadas. La Física tiene aplicaciones extraordinariamente amplias en otras áreas del conocimiento, así como en cuestiones prácticas que han suscitado sustanciales repercusiones tecnológicas. Con un enfoque reduccionista y pragmático, la Física —o Ciencias Físicas— las podemos categorizar como Física Básica y Física Aplicada.

— ***La Física Básica,*** son los conocimientos que permiten comprender los problemas fundamentales en la naturaleza, es decir, busca el conocimiento solo por conocer y comprender la naturaleza.

— ***La Física Aplicada,*** tiene la finalidad de utilizar los conocimientos de la Física Básica para resolver problemas prácticos dentro de la misma Física, así como en otras disciplinas del conocimiento.

Las principales áreas o subdisciplinas que forman parte de la Física Básica o Aplicada se muestran en la Figura 1. Dicha clasificación es tomada en parte de [6], pero no es exhaustiva.

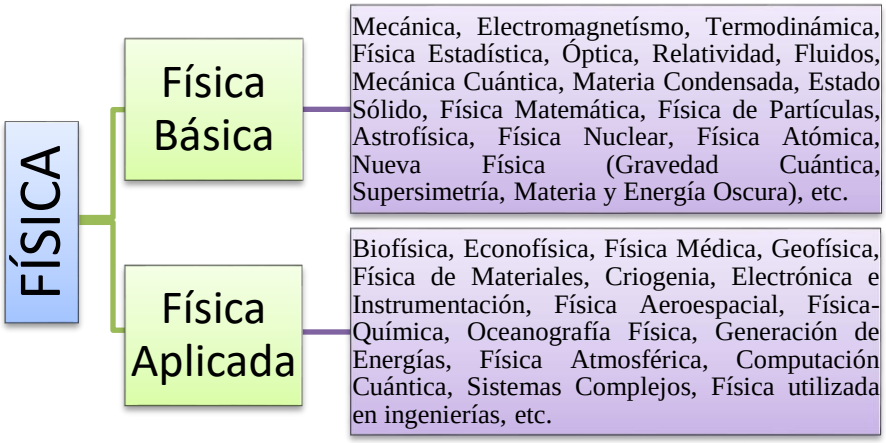


Fig. 1. Síntesis de las principales áreas de la Física.

La Física siempre ha sido, es y será un área de amplio interés para el humano. Los conocimientos generados en experimentos como el Gran Colisionador de Hadrones [7, 10-19], la prometedora Computación Cuántica [7, 22], la existencia de sistemas tan exóticos como agujeros negros [7], la hadronterapia para tratamientos de enfermedades como cáncer [8] y muchos otros experimentos o aplicaciones, hacen evidente la importancia de la Física —todos estos ejemplos hacen uso de la IA—.

Sin embargo, como cualquier otra disciplina científica, posee limitantes y requiere de robustas herramientas de las ciencias computacionales, en particular la IA. Las ciencias interdisciplinarias que surgen de la interacción de la Física con otras ciencias no son nuevas —basta recordar el origen de la Biofísica, Física Médica, Física de Materiales, etc.—. Una hipótesis de este trabajo es que está surgiendo un área nueva del conocimiento producto del cruce de la Física y la IA, con un enorme potencial de generar nuevo conocimiento y nuevas tecnologías con importantes implicaciones en ciencia y tecnología. Es importante identificar las herramientas específicas de IA que se están aplicando en la Física.

Es importante mencionar que existen pocos estudios como el presente, los estudios existentes no son exhaustivos —se enfocan en un área de IA aplicada a una o pocas áreas de la Física y publicados en revistas de Física—.

Esto no permite a los lectores o investigadores interesados en estos tópicos tener una visión general, actual y pronta del tema. De los trabajos existentes destaca [10, 11], quienes revisan los métodos de Aprendizaje Automático Aplicado a la Física de Partículas, Mecánica Cuántica, Física Estadística y Astrofísica. Ninguno de los trabajos revisados hace un estudio de cómo se está dando la interacción entre la IA y la Física.

4. Metodología

De acuerdo con Snyder [3], el proceso para llevar a cabo una revisión de la literatura está divido en cuatro fases: 1) el diseño de la revisión, 2) realización de la revisión, 3) análisis y 4) escribir la reseña. Esta metodología es la generalmente utilizada en el desarrollo de trabajos de este tipo.

Fase 1 Diseño de la revisión. Aquí se indica la pertinencia de la revisión a realizar, considerando preguntas como ¿Existe realmente la necesidad de una revisión de la literatura en esta área? Si es así ¿Qué tipo de revisión de la literatura será la más útil y haría la mayor contribución? Ante estos cuestionamientos, es necesaria una revisión de la literatura, dado que en Física recientemente son muy utilizadas técnicas y herramientas de IA, específicamente de Aprendizaje Automático, existiendo pocos artículos de revisión [10, 11]. Ahora, dentro de los tres diferentes enfoques que señala Snyder [3], en esta investigación se seleccionó la revisión semi-sistemática pues el objetivo es tener una visión general de un tema, en este caso ¿Cómo el Aprendizaje Automático ha contribuido o permeado en varias de las áreas de la Física? Se señala además que en la revisión semi-sistemática se analiza cómo la investigación dentro de un campo seleccionado ha progresado en el tiempo y cómo se ha desarrollado un tema a través de las tradiciones de investigación [9]. Entonces, está investigación busca identificar y comprender todas las áreas de la Física que usan el Aprendizaje Automático y sus implicaciones, además de sintetizarlas utilizando meta-narrativas.

Una vez que se ha identificado la pregunta de investigación y se ha considerado un enfoque de revisión general, es necesario desarrollar una estrategia de búsqueda para identificar la literatura, esto incluye seleccionar los términos de búsqueda y las bases de datos apropiadas y decidir sobre los criterios de inclusión y exclusión [3].

La estrategia de búsqueda utilizada fue la siguiente: se seleccionaron referencias bibliográficas de las bases de datos electrónicas más representativas, como: 1) Biblioteca digital ACM, 2) Annual Reviews, 3) EBSCOHOST, 4) Biblioteca digital IEEE Xplore, 5) Nature, 6) ScienceDirect, 7) Scopus, 8) Wiley, 9) Google Scholar y 10) InSpire. La ventana de observación definida para este estudio fue de 15 años: enero de 2005 hasta marzo del 2020. La búsqueda se efectuó con limitación del idioma inglés. Las palabras clave utilizadas para la búsqueda fueron "machine learning", " physical sciences", “physics” y “review”. Los criterios de inclusión involucran todos los aspectos señados anteriormente y los criterios de exclusión fueron: artículos científicos que se refieren a cualquier otro tipo de ciencia no relacionadas con la Física, y estudios realizados más bajo un enfoque matemático.

Fase II Realización de la revisión. En esta fase se realiza la revisión real, Snyder [3] señala que es preferible utilizar dos revisores, con el objetivo de garantizar la calidad y fiabilidad del protocolo de búsqueda. El presente estudio fue realizado por dos

revisores. Se utilizaron las 10 bases de datos descritas anteriormente, encontrando 25 y 29 artículos potencialmente relevantes por cada revisor, dando un total de 54 artículos en esta fase. La selección real de la muestra se puede hacer de varias maneras, dependiendo de la naturaleza y el alcance de la revisión específica, en este caso, el enfoque utilizado fue realizar la revisión por etapas; se eliminaron elementos duplicados y se realizó un análisis preliminar, basado en el título, resumen y los criterios de exclusión, para seleccionar documentos elegibles. Después de la selección quedaron 42 artículos (ver Figura 2), y que se leyeron por completo.

Fase III Análisis. Aquí se debe utilizar un medio estandarizado para extraer información apropiada de cada artículo. Los datos resumidos pueden ser en forma de información descriptiva, como autores, años de publicación, tema o tipo de estudio, o en forma de efectos y hallazgos. En esta fase se llevó a cabo la Tabla 1, que muestra la información descriptiva de cada artículo, determinando la referencia, el año de publicación, el título del artículo y el área de la Física a la que corresponde.

Fase IV Escribir la reseña. Dependiendo del enfoque, el artículo de revisión final puede estructurarse de diferentes maneras y con y diferentes niveles de detalle. Sin embargo, algunas generalizaciones pueden ser indicadas, por ejemplo, se espera que todos los autores sigan las convenciones aceptadas para informar sobre cómo se realizó el estudio, y describir de manera transparente el proceso de diseño de la revisión y el método para recopilar literatura, es decir, cómo la literatura fue identificada, analizada, sintetizada e informada [3]. En este caso de estudio, cada una de estas indicaciones fue realizada en las fases previas. También en esta fase es importante señalar las diferentes contribuciones que pueden ser valiosas en el campo de estudio.

El proceso de búsqueda y selección de trabajos relevantes se resume en la Figura 2.

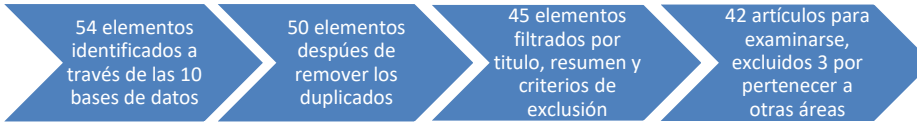


Fig. 2. Esquema del proceso de estrategia de búsqueda y selección.

5. Análisis de resultados y discusión

Siguiendo la metodología previamente planteada en la sección 4, se identificaron 42 publicaciones en la literatura internacional, qué hacen uso explícito del Aprendizaje Automático —Inteligencia Artificial— en la resolución parcial o total de un problema físico, ver Tabla 1. Las publicaciones están bien referenciadas en bases de datos que registran artículos de alta calidad académica; es de resaltar que varios de los artículos pertenecen a las revistas de mayor impacto en la Física como “Nature” (ver item 6, 8, 13, 14, 17, 29, 33, 36, 41 y 47). Aproximadamente 80% de estas publicaciones están citadas en el “Journal Citation Reports 2019” [52]. Dado que algunos artículos aplican el Aprendizaje Automático a dos o más áreas de la Física, se utiliza un asterisco (*)

Tabla 1. Publicaciones en Física que utilizan Aprendizaje Automático.

	Autor	Año	Título	Área de Física
1	Larkoski et al. [10]	2020	Jet substructure at the LHC: A review of recent advances in theory and machine learning	Física de Partículas
2	Carleo et al. [11] *5	2019	Machine learning and the physical sciences	Física de Partículas
3	Calafiura et al. [12]	2019	TrackML: A High Energy Physics Particle Tracking Challenge	Física de Partículas
4	Benson y Gizdov [13]	2019	NNDrone: A toolkit for the mass application of machine learning in High Energy Physics	Física de Partículas
5	Guest et al. [14]	2018	Deep Learning and Its Application to LHC Physics	Física de Partículas
6	Radovic et al. [15]	2018	Machine learning at the energy and intensity frontiers of particle physics	Física de Partículas
7	Albertsson et al. [16]	2018	Machine Learning in High Energy Physics Community White Paper	Física de Partículas
8	Baldi et al. [17]	2014	Searching for exotic particles in high-energy physics with deep learning	Física de Partículas
9	Whitenson et.al. [18]	2009	Machine learning for event selection in high energy physics	Física de Partículas
10	Rem et al. [19]	2019	Identifying quantum phase transitions using artificial neural networks on experimental data	Mecánica Cuántica
11	Carleo et al. [11] *5	2019	Machine learning and the physical sciences	Mecánica Cuántica
12	Dunjko y Briegel [20]	2018	Machine learning & artificial intelligence in the quantum domain: a review of recent progress	Mecánica Cuántica
13	Nieuwenburg et al. [21]	2017	Learning phase transitions by confusion	Mecánica Cuántica
14	Krein et al. [22]	2016	Automated Search for new Quantum Experiments	Mecánica Cuántica
15	Shuld et al. [23]	2015	An introduction to quantum machine learning	Mecánica Cuántica
16	Arsenault et al. [24] *2	2014	Machine learning for many-body physics: The case of the Anderson impurity model	Mecánica Cuántica
17	Carrasquilla y Melko [25]	2017	Machine learning phases of matter	Materia Condensada
18	Ch'ng, et al. [26] *2	2017	Machine Learning Phases of Strongly Correlated Fermions	Materia Condensada

para identificar la multiplicidad del artículo (p.ej. el item 2, aparece 5 veces), esto implicó los 50 items de la Tabla 1.

19	Arsenault et al. [24] *2	2014	Machine learning for many-body physics: The case of the Anderson impurity model	Materia Condensada
20	Qian et al. [27]	2020	Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems	Física Matemática
21	Pei et al. [28]	2020	Machine learning as a contributor to physics: Understanding Mg alloys	Física Matemática
22	Raissi, et al. [29]	2018	Hidden physics models: Machine learning of nonlinear partial differential equations	Física Matemática
23	Zhang et al. [30]	2020	A Review of Research on Pulsar Candidate Recognition Based on Machine Learning	Astrofísica
24	Carleo et al. [11] *5	2019	Machine learning and the physical sciences	Astrofísica
25	Cruz et al. [31]	2019	The use of the Reynolds force vector in a physics informed machine learning approach…	Fluidos
26	Swischuk et al. [32]	2019	Projection-based model reduction: Formulations for physics-based machine learning	Fluidos
27	Nakamura y Hashimoto [33]	2007	Hybrid learning strategy to solve pendulum swing-up problem for real hardware	Mecánica
28	Ruta y Gabris, 2007 [34] *2	2007	A framework for machine learning based on dynamic physical fields	Mecánica
29	Zdeborová et al. [35]	2017	MACHINE LEARNING New tool in the box	Física Estadística
30	Ruta y Gabris, 2007 [34] *2	2007	A framework for machine learning based on dynamic physical fields	Electromag-netismo
31	Zhou et al. [36]	2019	Emerging role of machine learning in light-matter interaction	Óptica
32	D'angolo y Wulzer [37]	2019	Learning new physics from a machine	Nueva Física
33	Collins et al. [38]	2018	Anomaly Detection for Resonant New Physics with Machine Learning	Nueva Física
34	Carleo et al. [11] *5	2019	Machine learning and the physical sciences	Física de Materiales
35	Oviedo et al. [39]	2019	Fast and interpretable classification of small X-ray diffraction datasets using data augmentation	Física de Materiales
36	Butler et al. [40] *2	2018	Machine learning for molecular and materials science	Física de Materiales
37	Rosenbrock et al. [41]	2017	Discovering the building blocks of atomic systems using machine learning: application	Física de Materiales
38	Ch'ng, et al. [26] *2	2017	Machine Learning Phases of Strongly Correlated Fermions	Física de Materiales
39	Li et al. [42]	2020	Reaction diffusion system prediction based on convolutional neural network	Física-Química

40	Lattimer et al. [43]	2020	Using machine learning in physics-based simulation of fire	Física-Química
41	Butler et al. [40] *2	2018	Machine learning for molecular and materials science	Física-Química
42	Carleo et al. [11] *5	2019	Machine learning and the physical sciences (Revisión)	Física-Química
43	Issam, et al. [44]	2015	Machine Learning in Radiation Oncology	Física Médica
44	Zhu et al. [45]	2011	A planning quality evaluation tool for prostate adaptive IMRT based on machine learning	Física Médica
45	Grazzini et al. [46]	2019	Extreme precipitation events over northern Italy. Part I: A systematic classification	Física Atmosférica
46	Xu et al. 2017 [47]	2017	A Novel Ozone Profile Shape Retrieval Using Full-Physics Inverse Learning Machine	Física Atmosférica
47	Hulbert et al. [48]	2018	Similarity of fast and slow earthquakes illuminated by machine learning	Geofísica
48	Ng et al. [49]	2019	Predicting the state of charge and health of batteries using data-driven machine learning	Electrónica e Instrumentación
49	Cheng and Yu, 2019 [50]	2019	A new generation of AI: A review and perspective on machine learning technologies applied to smart energy	Sistemas de energía
50	Alber et al. [51]	2019	Integrating machine learning and multiscale modeling—	Biofísca

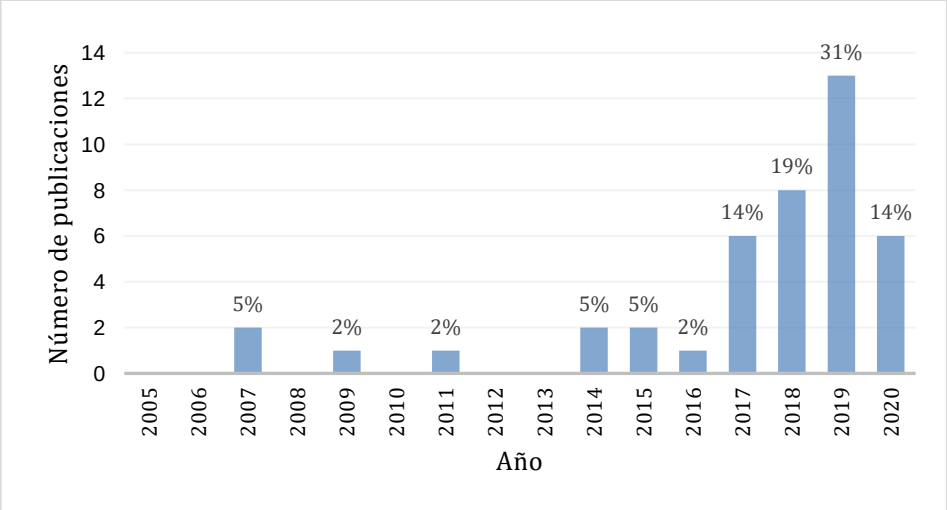


Fig. 3. Núm. de publicaciones en Física que utilizan Aprendizaje Automático.

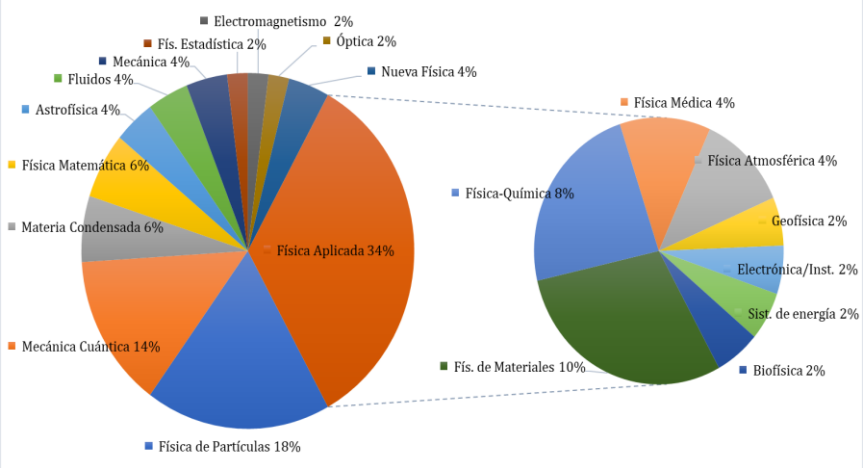


Fig. 4. Áreas de la Física que utilizan Aprendizaje Automático.

La Figura 3 muestra los artículos de investigación publicados en función del tiempo, entre enero-2005 y marzo-2020. Resalta el aumento significativo y acelerado de publicaciones en los últimos 4 años, y que corresponde al 78% de las publicaciones.

Para agrupar los artículos se propuso la siguiente clasificación que consta de 2 categorías que cubren varias áreas de Física: 1) Física Básica, y 2) Física Aplicada. En la Figura 4 se muestra la clasificación realizada en este estudio y el porcentaje de trabajos encontrados por áreas de la Física. Es importante notar que todas las publicaciones hacen uso del Aprendizaje Automático como una herramienta sumamente poderosa para llegar a una conclusión o resultado en la Física y son publicaciones en revistas de Física. No hay en sentido opuesto, de aplicaciones de la Física a IA.

En la Figura 4 se puede observar que las áreas del conocimiento que concentran las publicaciones son en primer lugar Física de Partículas (18%), seguida de Mecánica Cuántica (14%), en tercer lugar Física de materiales (10%), posteriormente Física-química (8%), Materia Condensada (6%), Física matemática (6%), después Física médica, Física atmosférica, Mecánica, Astrofísica, Fluidos, nueva Física con 4%, y finalmente con 2% cada una de las restantes áreas.

Es conveniente mencionar que se logró establecer una metodología para identificar las intersecciones de la Física con el Aprendizaje Automático, mediante ello se identificaron las publicaciones al respecto y con base en esto se hace evidente que la Inteligencia Artificial, a través del Aprendizaje Automático, está volviéndose una herramienta importante en las Ciencias Físicas. Este cruzamiento es relativamente nuevo y muy novedoso, esto puede ser explicado por el robustecimiento del Aprendizaje Automático, por una parte, y por otra, debida a la necesidad de nuevas herramientas para resolver problemas sumamente complejos en las Ciencias Físicas, que por sí solas no pueden resolver con herramientas tradicionales.

Algo que es muy interesante para la Física es que, analizando cualitativamente los ítems 14, 32 y 33, se observa una estrategia nueva para identificar nueva Física (como nuevos experimentos en Mecánica Cuántica o Física más allá del modelo estándar) mediante la Inteligencia Artificial. Esto es algo completamente nuevo, pues la Física nunca se ha construido mediante IA. Podemos sugerir el término de Física de IA.

6. Conclusiones

Se identificaron 42 artículos científicos sobre IA (Aprendizaje Automático) aplicado a la Física, y las formas en cómo se utilizan y aplican las técnicas de Aprendizaje Automático a las diferentes áreas de la Física. De lo anterior sobresalen los siguientes puntos: primero resalta el gran interés creciente que las técnicas de Aprendizaje Automático tienen en la Física, pues en los últimos 4 años están el 78% de todas las publicaciones identificadas. Sobresale Física Básica, con las áreas de Física de Partículas y Mecánica Cuántica pues son las que más hacen uso del Aprendizaje Automático, concentrando el 48% de las publicaciones en esa área en los últimos 4 años. Segundo, en lo referente a Física Aplicada, sobresale Física de Materiales con el 10%.

Es importante indicar como se observa en la Figura 4, que la cantidad de áreas donde ha sido utilizado Aprendizaje Automático es muy amplia, en Física Básica abarca Física de Partículas, Mecánica Cuántica, Materia Condensada, Física Matemática, Astrofísica, Fluidos, Física estadística, Mecánica, Electromagnetismo y Óptica. Ahora bien, la conexión de técnicas de Aprendizaje Automático con la Física Aplicada también es muy amplia, abarca: Materiales, Física-química, Física médica, Física Atmosférica, Geofísica, Electrónica e Instrumentación, Sistemas de energía y Biofísica. Tercero, destacan dos trabajos que aplican Aprendizaje Automático para la búsqueda de nueva Física, lo que implica una nueva forma de construir conocimiento en Física.

Cuarto, el trabajo realizado permite observar que la interacción entre el Aprendizaje Automático y las Ciencias Físicas ha tenido un crecimiento los últimos años, sin embargo, aún hay mucho por hacer, pues en muchas de las áreas no se ha aplicado grandemente y que se podrían generar resultados interesantes. Como trabajo futuro se realizará un análisis detallado de la revisión por áreas de la Física, con el objetivo de encontrar específicamente las herramientas y técnicas de Aprendizaje Automático que son más utilizadas dentro de cada área en particular.

Referencias

1. Hogg, T., Huberman, B.A.: Artificial intelligence and large scale computation: A physics perspective. Physics Reports, 156(5), pp. 227–310 (1987)
2. Rossel, C., Nordberg, M.: Physics and its current and future applications. Europhysics News (2018)
3. Snyder, H.: Literature review as a research methodology: An overview and guidelines. Journal of Business Research, 104, pp. 333–339 (2019)
4. Gordon, B.M.: Artificial intelligence: approaches, tools, and applications. New York: Nova Science Publishers, Inc. (2011)

5. Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á.L., Heredia, I.: Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review, pp. 77–124 (2019)
6. Sears, F.W., et al.: Física Universitaria. 1 (2009)
7. Russell, S.: Artificial intelligence: The future is superintelligent. Nature, 548(7669), pp. 520 (2017)
8. Akgun, U.: CARNA with artificial intelligence-use of machine learning for a proton imaging. In: SPIE Medical Imaging Conference (2019)
9. Wong, G., Greenhalgh, T., Westhorp, G., Buckingham, J., Pawson, R.: RAMESES publication standards: realist syntheses. BMC medicine, 21 (2013)
10. Larkoski, A.J., Moult, I., Nachman, B.: Jet substructure at the Large Hadron Collider: a review of recent advances in theory and machine learning. Physics Reports, 841, pp 1–63 (2020)
11. Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Zdeborová, L.: Machine learning and the physical sciences. Reviews of Modern Physics, 91(4), 045002 (2019)
12. Calafiura, P., Farrell, S., Gray, H., Vlimant, J.R., Innocente, V., Salzburger, A., Germaint, C.: TrackML: A high energy physics particle tracking challenge. In: 14th International Conference on e-Science, IEEE. pp. 344–344 (2018)
13. Benson, S., Gizdov, K.: NNDrone: A toolkit for the mass application of machine learning in High Energy Physics. Computer Physics Communications, 240, pp. 15–20 (2019)
14. Guest, D., Cranmer, K., Whiteson, D.: Deep learning and its application to LHC physics. Annual Review of Nuclear and Particle Science, pp. 161–181 (2018)
15. Radovic, A., Williams, M., Rousseau, D., Kagan, M., Bonacorsi, D., Himmel, A., Wongjirad, T.: Machine learning at the energy and intensity frontiers of particle physics. Nature, 560(7716), pp 41–48 (2018)
16. Albertsson, K., Altoe, P., Anderson, D., Anderson, J., Andrews, M., Espinosa, J.P.A., Bonacorsi, D.: Machine learning in high energy physics community white paper (2018)
17. Baldi, P., Sadowski, P., Whiteson, D.: Searching for exotic particles in high-energy physics with deep learning. Nature communications, pp 1–9 (2014)
18. Whiteson, S., Whiteson, D.: Machine learning for event selection in high energy physics. Engineering Applications of Artificial Intelligence, pp. 1203–1217 (2009)
19. Rem, B.S., Käming, N., Tarnowski, M., Asteria, L., Fläschner, N., Becker, C., Weitenberg, C.: Identifying quantum phase transitions using artificial neural networks on experimental data. Nature Physics, 15(9), pp 917–920 (2019)
20. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics, 81(7), 074001 (2018)
21. Van Nieuwenburg, E.P., Liu, Y.H., Huber, S.D.: Learning phase transitions by confusion. Nature Physics, 13(5), pp. 435–439 (2017)
22. Krenn, M., Malik, M., Fickler, R., Lapkiewicz, R., Zeilinger, A.: Automated search for new quantum experiments. Physical review letters, 090405 (2016)
23. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemporary Physics, 56(2), pp 172–185 (2015)
24. Arsenault, L.F., Lopez-Bezanilla, A., von Lilienfeld, O.A., Millis, A.J.: Machine learning for many-body physics: The case of the Anderson impurity model. Physical Review B, 90(15), 155136 (2014)
25. Carrasquilla, J., Melko, R.G.: Machine learning phases of matter. Nature Physics, 13(5), pp. 431–434 (2017)
26. Ch'Ng, K., Carrasquilla, J., Melko, R.G., Khatami, E.: Machine learning phases of strongly correlated fermions. Physical Review X, 7(3), 031038 (2017)

27. Qian, E., Kramer, B., Peherstorfer, B., Willcox, K.: Lift & learn: physics-informed machine learning for large-scale nonlinear dynamical systems. Physica D: Nonlinear Phenomena, 406, 132401 (2020)
28. Pei, Z., Yin, J.: Machine learning as a contributor to physics: Understanding Mg alloys. Materials & Design, 172, 107759 (2019)
29. Raissi, M., Karniadakis, G.E.: Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics, 357, pp.125–141 (2018)
30. Zhang, C.J., Shang, Z.H., Chen, W.M., Xie, L., Miao, X.H.: A review of research on pulsar candidate recognition based on machine learning. Procedia Computer Science, 166, pp. 534–538 (2020)
31. Cruz, M.A., Thompson, R.L., Sampaio, L.E., Bacchi, R.D.: The use of the Reynolds force vector in a physics informed machine learning approach for predictive turbulence modeling. Computers & Fluids, 192, 104258 (2019)
32. Swischuk, R., Mainini, L., Peherstorfer, B., Willcox, K.: Projection-based model reduction: Formulations for physics-based machine learning. Computers & Fluids, 179, pp. 704–717 (2019)
33. Nakamura, S., Hashimoto, S.: Hybrid learning strategy to solve pendulum swing-up problem for real hardware. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1972–1977 (2007)
34. Ruta, D., Gabrys, B.: A framework for machine learning based on dynamic physical fields. Natural Computing, 8(2), pp. 219–237 (2009)
35. Zdeborová, L.: Machine learning: New tool in the box. Nature Physics, 13(5), pp 420–421 (2017)
36. Zhou, J., Huang, B., Yan, Z., Bünzli, J.C.G.: Emerging role of machine learning in light-matter interaction. Light: Science & Applications, pp. 1–7 (2019)
37. D'Agnolo, R.T., Wulzer, A.: Learning new physics from a machine. Physical Review D, 99(1), 015014 (2019)
38. Collins, J., Howe, K., Nachman, B.: Anomaly detection for resonant new physics with machine learning. Physical review letters, 121(24), 241803 (2018)
39. Oviedo, F., Ren, Z., Sun, S., Settens, C., Liu, Z., Hartono, N.T.P.: Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks. Computational Materials, 5(1), pp. 1–9 (2019)
40. Butler, K.T., Davies, D.W., Cartwright, H., Isayev, O., Walsh, A.: Machine learning for molecular and materials science. Nature, pp. 547–555 (2018)
41. Rosenbrock, C.W., Homer, E.R., Csányi, G., Hart, G.L.: Discovering the building blocks of atomic systems using machine learning: application to grain boundaries. Computational Materials, 3(1), pp. 1–7 (2017)
42. Li, A., Chen, R., Farimani, A.B., Zhang, Y.J.: Reaction diffusion system prediction based on convolutional neural network. Scientific reports, pp. 1–9 (2020)
43. Lattimer, B.Y., Hodges, J.L., Lattimer, A.M.: Using machine learning in physics-based simulation of fire. Fire Safety Journal, 102991 (2020)
44. El Naqa, I., Murphy, M.J.: What is machine learning? In: machine learning in radiation oncology. Springer, pp. 3–11 (2015)
45. Zhu, X., Ge, Y., Li, T., Thongphiew, D., Yin, F.F., Wu, Q. J.: A planning quality evaluation tool for prostate adaptive IMRT based on machine learning. Medical physics, 38(2), pp. 719–726 (2011)
46. Grazzini, F., Craig, G.C., Keil, C., Antolini, G., Pavan, V.: Extreme precipitation events over northern Italy. Part I: A systematic classification with machine-learning techniques. Quarterly Journal of the Royal Meteorological Society, 146(726), pp. 69–85 (2020)
47. Xu, J., Schüssler, O., Rodriguez, D.G.L., Romahn, F., Doicu, A.: A novel ozone profile shape retrieval using full-physics inverse learning machine. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10(12), pp. 5442–5457 (2017)

48. Hulbert, C., Rouet-Leduc, B., Johnson, P.A., Ren, C.X., Rivière, J., Bolton, D.C., Marone, C.: Similarity of fast and slow earthquakes illuminated by machine learning. Nature Geoscience, 12(1), pp. 69–74 (2019)
49. Ng, M.F., Zhao, J., Yan, Q., Conduit, G.J., Seh, Z.W.: Predicting the state of charge and health of batteries using data-driven machine learning. Nature Machine Intelligence, pp. 1–10 (2020)
50. Cheng, L., Yu, T.: A new generation of AI: A review and perspective on machine learning technologies applied to smart energy and electric power systems. International Journal of Energy Research, 43(6), pp. 1928–1973 (2019)
51. Alber, M., Tepole, A.B., Cannon, W.R., De, S., Dura-Bernal, S., Garikipati, K., Kuhl, E.: Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. NPJ digital medicine, 2(1), pp. 1–11 (2019)
52. Clarivate Analytics: Journal Citation Reports Social Sciences Edition, https://clarivate.com/webofsciencegroup/ (2018)

ISSN 1870-4069

Nuevo enfoque para la extracción de características en la clasificación de textos para la atribución de autoría

Omar González Brito[1], José Luis Tapia Fabela[1], Silvia Salas Hernández[2]

[1] Universidad Autónoma del Estado de México,
Unidad Académica Profesional Tianguistenco,
México
[2] Universidad Autónoma del Estado de México,
Centro Universitario Atlacomulco,
México

gonzalezbritoomar@gmail.com, joseluis.fabela@gmail.com,
salashernandezsilvia@gmail.com

Resumen. La tarea de atribución de autoría se ha realizado bajo dos enfoques: basado en perfil e instancias, a través del análisis de características textuales o rasgos lingüísticos que permiten encontrar el estilo de escritura del autor. El conjunto de datos es uno de los principales problemas que enfrentan los enfoques y métodos de atribución de autoría debido a que no siempre se cuenta con un conjunto de documentos representativos de cada autor. En los enfoques de perfil e instancias, al extraer las características a partir del conjunto de documentos de los autores, se genera una alta dimensionalidad de características. Por lo que se propone un nuevo enfoque que no dependa del conjunto de documentos para la extracción de características, la clasificación de textos se realizó con máquina de soporte vectorial como método de aprendizaje supervisado. Este determina si están contenidas todas las características de un autor en un solo documento que describan su estilo de escritura. Para la experimentación se trabajó con tres *corpus* (C10, C50 y PAN12), estos fueron seleccionados basándose en la revisión de la literatura. De acuerdo a los resultados obtenidos se concluyó que el enfoque muestra resultados superiores al estado del arte en muestras desbalanceadas, resultados consistentes cuando es evaluado en diferentes contextos y robusto al analizar 10 y 50 autores. A partir de este enfoque se puede determinar que con 500 palabras sin repetir se puede identificar el estilo de escritura de un autor, presentando una exactitud de clasificación del 79.68%.

Palabras clave: Atribución de autoría, clasificación de textos, máquina de soporte vectorial, extracción de características, enfoques de autoría.

New Approach for the Extraction of Characteristics in the Classification of Texts for Attribution of Authorship

Abstract. The task of attribution of authorship has been carried out under two approaches: based on profile and instances, through the analysis of textual features or linguistic features that allow finding the author's writing style. The

data set is one of the main problems faced by attribution approaches and methods of authorship because it does not always have a set of documents representative of each author. In profile and instance approaches, by extracting the features from the authors' set of documents, a high dimensionality of features is generated. Therefore, a new approach is proposed that does not depend on the set of documents for the extraction of features, the classification of texts was carried out with a vector support machine as a supervised learning method. This determines if all the features of an author are contained in a single document that describes her writing style. For experimentation, we worked with three corpora (C10, C50 and PAN12), these were selected based on the literature review. According to the results obtained, it was concluded that the approach shows results superior to the state of the art in unbalanced samples, consistent results when evaluated in different contexts and robust when analyzing 10 and 50 authors. From this approach, it is determined that an author's writing style is contained in 500 words without repeating, presenting a classification accuracy of 79.68%.

Keywords: Attribution of authorship, classification of texts, vector support machine, extraction of characteristics, approaches of authors.

1. Introducción

El problema computacional de identificación de autoría ha sido abordado bajo dos enfoques; basado en perfil e instancias, utilizando modelos de clasificación generativos o discriminativos [1]. En el enfoque basado en perfil se extrae el estilo de escritura de cada autor, este es creado a través de la concatenación de los documentos obteniendo un solo documento por autor, del cual son extraídas las características para el entrenamiento del modelo [1, 2]. A diferencia del enfoque basado en instancias, la extracción de características se realiza por cada documento del autor, una vez extraídas forman un solo conjunto con el que se realiza el entrenamiento del clasificador y se genera el modelo de atribución de autoría. El enfoque más utilizado es el basado en instancias mediante la clasificación de textos [1-5].

En [5] se implementa un enfoque de instancias mediante los modelos de aprendizaje: máquina de soporte vectorial (*Support Vector Machine*, por sus siglas en inglés SVM) y máquina de soporte tensorial implementado el método del término más frecuente para seleccionar los 2500 términos más frecuentes. En este trabajo analizan las muestras balanceadas y desbalanceadas volviéndose uno de los trabajos más referenciados con el *corpus* C10. El proceso de extracción de características se realiza con el conjunto de documentos de los autores. Sin embargo, no siempre existe el número de documentos para cada autor por lo que se requiere determinar si en un solo documento se puede encontrar el estilo de escritura de un autor.

En la presente investigación se propone un nuevo enfoque para atribución de autoría, este propone utilizar un método de clasificación de textos extrayendo las características de un solo documento del conjunto de entrenamiento sin la implementación de métodos de selección de características, analizando este nuevo enfoque con características léxicas (*n-gramas* y bolsa de palabras), con una representación booleana y máquina de soporte vectorial como método de aprendizaje supervisado.

Para la experimentación se utilizaron tres diferentes *corpus:* C50, C10 y PAN12, analizando diferentes contextos, tamaños de muestras y diferentes números de autores.

2. Trabajos relacionados

Actualmente los métodos del estado del arte analizan ambos enfoques: perfil e instancias como se muestra en los trabajos de [2, 6]. En [2] realizaron una clasificación mediante los enfoques de perfil e instancias con el modelo LDA (*Latent Dirichlet Allocation,* por sus siglas en inglés LDA) para el análisis de datos dispersos y de alta dimensionalidad, utilizando la medida de similitud de coseno para determinar la autoría. Este modelo utilizó el conjunto de documentos de entrenamiento para la extracción de las características, posteriormente implementa el modelo LDA para realizar la selección de características.

En el trabajo presentado en [6] se realizó la construcción de grafos sintácticos bajo los enfoques de instancias y perfil debido que al implementar las técnicas de bolsa de palabras o *n-gramas* se pierde información sintáctica y semántica, caso que en los grafos sintácticos se conserva, para determinar la autoría usaron la métrica de similitud de coseno, esta ha sido implementado en las dos tareas de autoría; verificación y atribución. En los enfoques de perfil e instancias, al extraer las características a partir del conjunto de documentos de los autores, se genera una alta dimensionalidad de características, debido a esto surge la necesidad de implementar métodos de selección de características para extraer las más representativas, como lo realizaron [5, 7, 8].

El conjunto de datos es uno de los principales problemas que enfrentan los métodos de atribución de autoría [6]. En el análisis de las muestras desbalanceadas los resultados no son los mismos que en muestras balanceadas [9], esto se debe a que dependen del número de documentos para extraer las características.

3. Artefactos propuestos

El enfoque propuesto se analiza a partir de la clasificación de textos; que se compone de seis etapas: adquisición de datos, análisis y etiquetado de datos, construcción y ponderación de características, selección o proyección de características, entrenamiento del modelo y evaluación de la solución [10]. Los métodos de clasificación han sido utilizados en los enfoques de instancias y perfil de autor. En la presente investigación la etapa del análisis y etiquetado de datos se realizó considerando la extracción de características de un solo documento, la etapa de selección o proyección fue omitida. Para la etapa de experimentación el conjunto de datos con el que se trabajó fueron los *corpus* C50, C10 y PAN12. Los *corpus* C50 y C10 están compuestos por documentos del *corpus* REUTERS volumen uno, que se encuentra dentro de la categoría CCAT formado de noticias acerca de la industria, por cada autor tiene 50 documentos de entrenamiento y 50 de validación, la diferencia entre el *corpus* C50 y el C10 es el número de autores, en el C50 se consideran 50 autores y en el C10 se consideran 10 autores. El *corpus* PAN12 está compuesto por 14 autores cada uno con tres documentos en idioma ingles del contexto novelas [1, 11].

3.1. Adquisición de datos

En 2009 surge la primera competición internacional de detección de plagio en el marco del taller PAN (CLEF, *Conference and Labs of the Evaluación Forum*). Su objetivo ha sido fomentar el desarrollo de herramientas automáticas para la detección de plagio y actualmente también la identificación de autoría y otros usos abusivos del *software* social [1]. En la página web de PAN (https://pan.webis.de/) se encuentran los *corpus* C50, C10 y PAN12, estos corpus han sido utilizados para la tarea de atribución de autoría en diferentes trabajos de la literatura, el acceso a estos es de dominio público.

3.2. Análisis y etiquetado de datos

Para la extracción de características únicamente se consideraron las características de un solo documento por autor. El análisis del método se realizó con dos modelos de representación que han sido utilizados en trabajos de la literatura [3, 5]: bolsa de palabras y trigramas de carácter. Para la representación con bolsa de palabras se eliminaron los siguientes caracteres especiales; signos de puntuación, signos de admiración e interrogación, y con el modelo de trigramas no se realiza ningún pre procesamiento.

3.3. Construcción y ponderación de características

La ponderación booleana o binaria consiste en asignar un valor a un término dentro de un documento, el valor asignado al término refleja la importancia del mismo, se asigna un valor de uno cuando el término se encuentra presente en caso contrario se asigna un valor de cero. A través de esta ponderación se puede conocer la importancia de cada uno de los términos. Lo anterior se representa a través de la ecuación uno, Donde: t_j representa la frecuencia de término j que tiene la oración p_i [12, 13]:

$$p_i\,(t_j) = \begin{cases} 0, si\ aparece, \\ 1, en\ otro\ caso. \end{cases} \quad (1)$$

La Tabla 1 muestra la representación vectorial utilizada en la presente investigación, donde en una columna se coloca el documento de cada autor y en la fila el termino, posteriormente se va llenando la tabla con la ponderación, donde 1 significa que el término está presente dentro del documento del autor, caso contrario se coloca un cero, este proceso se realiza hasta terminar con todos los documentos de los autores.

Tabla 1. Representación vectorial con ponderación booleana.

Documento/autor	Término 1	Término2	Término n-1	Término n
Documento1 autor1	0	0	1	1
Documento n-1 autor1	0	1	0	0
Documento n autor1	0	0	1	1
Documento n-1 autor n-1	1	1	0	1
Documento n autor n	1	0	0	1

3.4. Entrenamiento del modelo

El modelo se construyó a partir de aprendizaje supervisado implementando máquina de soporte vectorial, los parámetros fueron un *kernel* lineal, el parámetro C igual a uno, utilizando una clasificación de uno contra todos, la máquina de soporte vectorial fue entrenada con las características de cada uno de los autores, una vez generado el modelo se realizó la evaluación.

3.5. Evaluación de la solución

La métrica utilizada para la evaluación fue la exactitud. Esta consiste en el porcentaje de instancias que se clasifican correctamente se define en términos de verdaderos positivos (*True positives*, por sus siglas en inglés TP), Falsos positivos (*False positives*, por sus siglas en inglés FP), Verdaderos negativos (*True Negatives*, por sus siglas en inglés TN) y falsos negativos (*False Negatives*, por sus siglas en inglés FN) como se muestra en la ecuación 2 [14]:

$$exactitud = \frac{TP + TN}{TP + TN + FP + FN}. \quad (1)$$

3.6. Experimentación y resultados

Para la comprobación del método propuesto se realizaron 4 experimentos, estos fueron validados mediante la técnica de validación cruzada a diez pliegues, utilizando una máquina de soporte vectorial para la clasificación, estos experimentos muestran la robustez del método, ya que se analiza en primera instancia el método con respecto a los del estado del arte con el *corpus* C10.

En el segundo experimento se observa la robustez del método con cincuenta autores con el *corpus* C50.

En el tercer experimento se analiza la exactitud para la clasificación con diferentes *corpus*, para este experimento se utilizaron los *corpus* C10 y PAN12, en el cuarto experimento se analizan las muestras balanceadas y desbalanceadas con el *corpus* C50.

El primer experimento se realizó con el *corpus* C10, el objetivo de este experimento fue observar la robustez del método analizando una muestra balanceada con diez autores y por cada autor 50 documentos de entrenamiento y 50 de validación.

El método propuesto se compara con uno de los más referenciados del *corpus* C10 el modelo de espacio de tensores [5], y algunos de los métodos más recientes como Doc2vec presentado en [14] y el método de aprendizaje profundo presentado en [15] como se muestra en la figura 1.

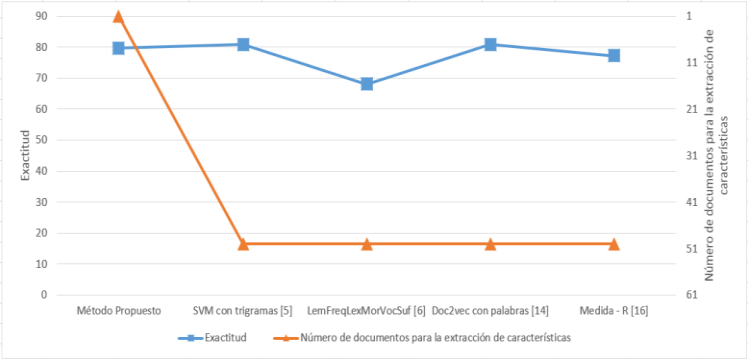


Fig. 1. Exactitud en relación al número de documentos para la extracción de características en el *corpus* C10.

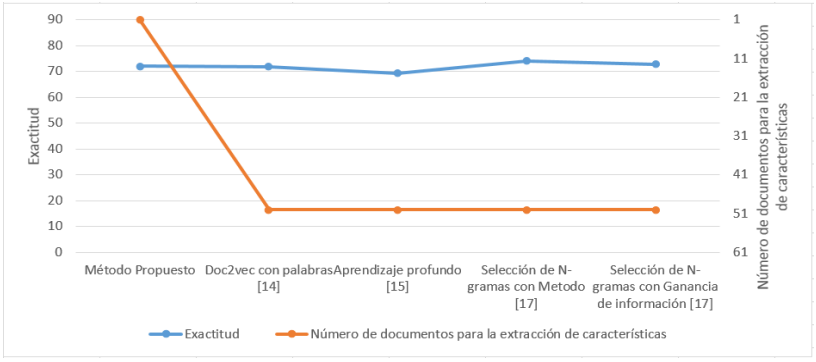


Fig. 2. Exactitud en relación al número de documentos para la extracción de características en el *corpus* C50.

El método propuesto alcanza una exactitud de 79.68 % con una diferencia de 1.12 % del modelo de espacio de tensores presentado en [5], obteniendo mejores resultados que algunas propuestas del estado del arte como se puede observar en la Fig. 1.

El segundo experimento se realiza de la misma manera que el primero, pero con el *corpus* C50, que contiene 50 autores cada uno de ellos con 50 documentos de entrenamiento y 50 de validación a diferencia del C10 que contiene 10 autores, el objetivo de este experimento es conocer si el método obtiene una buena exactitud en la clasificación con cincuenta autores, la exactitud obtenida es de 72.04, un resultado competitivo con respecto al estado del arte como se muestra en la Fig. 2.

En el tercer experimento se analizó el método bajo dos diferentes contextos: correos electrónicos y obras literarias. Se analizó el método propuesto considerando como características a los trigramas y bolsa de palabras. Utilizando el trabajo de [16] para comparar los resultados, en este trabajo reproducen los métodos más referenciados del estado del arte para atribución de autoría realizando la experimentación con los *corpus* C10 y PAN12.

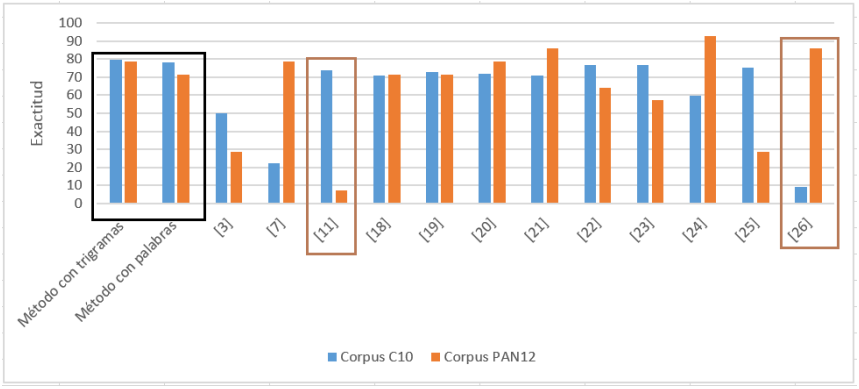


Fig. 3. Evaluación del método bajo diferentes contextos.

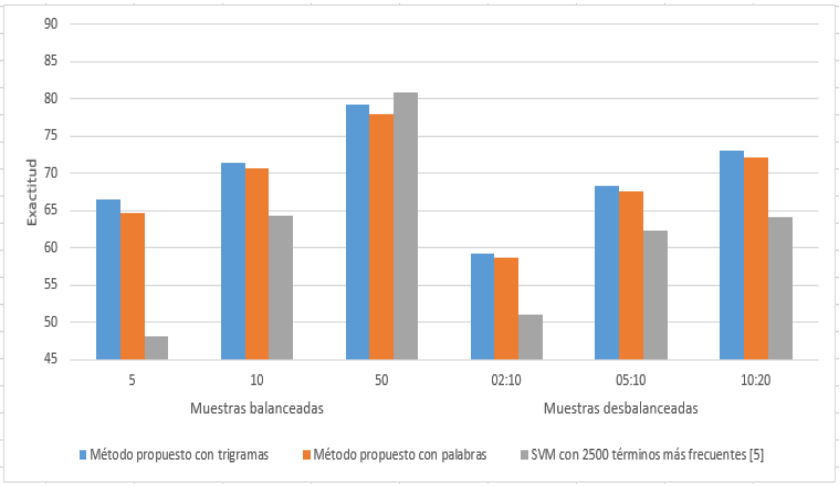


Fig. 4. Análisis de muestras balanceadas y desbalanceadas en el *Corpus* C10.

Se puede observar en la Fig. 3 que el método propuesto se mantiene constante en los dos *corpus* obteniendo resultados superiores al 70%, en comparación a los evaluados por [16] donde los métodos con un *corpus* obtienen buenos resultados y con el otro bajo resultado.

El método propuesto obtiene mejores resultados que algunos de los métodos evaluados por [16]. El análisis de muestras balanceadas y desbalanceadas se aborda principalmente en el estado del arte con el *corpus* C10. Se analiza el método propuesto considerando como características a los trigramas y bolsa de palabras.

Como se puede observar en la Fig. 4, los métodos propuestos obtienen mejores resultados en las muestras desbalanceadas que el propuesto por [5] y en las muestras balanceadas se obtienen resultados competitivos e incluso en las muestras balanceadas de tamaño cinco y diez es mejor, donde se puede determinar que el método propuesto es también robusto con muestras balanceadas y desbalanceadas.

4. Conclusiones y trabajo futuro

El número promedio de palabras por documento en el *corpus* C10 es de 507 con esto se obtiene una exactitud de 79.68%, para el C50 el promedio es de 500 de acuerdo a la experimentación, con estas se obtiene una exactitud de 72.004%, lo anterior puede observarse en la Fig. 1 y Fig. 2 donde el método presenta competitividad con los trabajos revisados en la literatura. A través de esta experimentación se concluye que en un solo documento se encuentra contenido el estilo de escritura de un autor.

La comparativa de resultados entre el *corpus* C10 y PAN12 en diferentes contextos muestra la robustez del método, esto se puede observar en la Fig. 3, donde el método se mantiene consistente en los resultados esto comparado con los métodos analizados en [16]. El análisis de métodos presentado por [16] muestra que algunos métodos presentan variabilidad en sus resultados cuando trabajan con *corpus* de diferentes contextos, un ejemplo de este es el método de [11] donde el resultado con el *corpus* C10 es de 78% y 7.1% de exactitud con PAN12 o el método [26] donde el resultado con el *corpus* C10 es de 9% y 85.7% de exactitud con PAN12.

El análisis de muestras desbalanceadas puede trasladarse a la vida real donde en algunos casos no existe el mismo número de documentos para cada autor, los resultados de la experimentación muestran la superación de los trabajos revisados en la literatura, donde el porcentaje de superación de la muestra 2:10 es de 8.3%, para la muestra 5:10 es de 6% y para 10:20 es de 8.92%, con este tipo de muestras permite asemejar la situaciones de la vida real donde no se cuenta con el mismo número de documentos por cada uno de los autores como se observa en la Fig. 4.

Los resultados presentados muestran que la presente investigación supero a los trabajos revisados en la literatura, esto se puede observar en la Fig. 4. Para las muestras balanceadas el método supera a dos de las tres muestras de tamaño cinco y diez con una diferencia de 18.32% y 7.04%. Sin embargo, en la muestra de tamaño 50 el método propuesto por [5] supera al método con 1.12% como se observa en la Fig. 4. Sin embargo, hay que considerar que utilizamos en promedio 1820 características y el método de [5] utiliza 2500.

Un área de oportunidad que surge a partir de esta investigación es el desarrollo de métodos que puedan determinar la autoría a partir de un solo documento mediante la implementación de métricas de similitud, con una exactitud superior a la de los trabajos de la literatura.

Referencias

1. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), pp. 538–556 (2009)
2. Anwar, W., Bajwa, I., Ramzan, S.: Design and implementation of a machine learning based authorship identification model. Scientific Programming, pp. 1–14 (2019)
3. De Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining e-mail content for author identification forensics. In: Association for Computing Machinery Special Interest Group on Management of Data (ACM SIGMOG) 30(4), 55 (2001)
4. Solorio, T., Montes, M., Pillay, S.: Authorship identification with modality specific meta features. In: Conference CLEF Labs and Workshop, Notebook Papers (2011)

5. Plakias, S., Stamatatos, E.: Tensor space models for authorship attribution. In: Proceedings of the Conference Hellenic Conference on Artificial Intelligence. Lecture Notes in computer Science, 5138, pp. 239–249, Springer, Heidelberg (2008)
6. Gómez-Adorno, H., Sidorov, G., Pinto, D., Vilariño, D., Gelbukh, A.: Automatic authorship detection using textual patterns extracted from integrated syntactic graphs. Sensors 16(9), 1374 (2016)
7. Koppel, M., Shler, J., Bonchek-Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. Journal of Machine Learning Research 8, pp. 1261–1276 (2007)
8. Pavlyshenko, B.: Genetic optimization of keywords subset in the classification analysis of texts authorship. Journal of Quantitative Linguistics 21, pp. 341–349 (2014)
9. Pastor, A., Montes-y-Gómez, M., Villaseñor-Pineda, L., Ariel, J., Martínez-Trinidad, J.: A new document author representation for authorship attribution. In: Mexican Conference on Pattern Recognition, pp. 283–292 (2012)
10. Mirończuk, M., Protasiewicz, J.: A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications, 106, pp. 36–54 (2018)
11. Stamatatos, E.: Authorship attribution based on feature set subspacing ensembles. International Journal on Artificial Intelligence Tools 15 (05), pp.823–838 (2006)
12. Ledeneva, Y., García, R.: Generación automática de resúmenes: retos, propuestas y experimentos (2017)
13. Zhang, Z., Li, X., Tian, X.: Research on feature weights of Liheci word sense disambiguation. In: 8th International Symposium on Computational Intelligence and Design (ISCID), 2, pp. 7–10 (2015)
14. Posadas-Durán, J., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., Chanona-Hernández, L.: Application of the distributed document representation in the authorship attribution task for small corpora. Soft Computing, 21, pp. 627–639 (2017)
15. Qian, C., He, T., Zhang, R.: Deep learning based authorship identification (2017)
16. Potthast, M., Braun, S., Buz, T., Duffhauss: Who wrote the web? Revisiting influential author identification research applicable to information retrieval. In: European Conference on Information Retrieval. Lecture Notes in Computer Science, 9626, pp. 393–407, Springer, Cham (2016)
17. Houvardas, J., Stamatatos, E.: N-Gram feature selection for authorship identification. In: Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, and Applications, Lecture Notes in Computer Science, 4183, pp. 77–86. Springer, Varna (2006)
18. Keselj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING, pp. 255–264 (2003)
19. Benedetto, D., Caglioti, E., Loreto, V.: Language trees and zipping. Physical Review Letters 88(4), 048702 (2002)
20. Koppel, M., Schler, J., Argamon, S.: Authorship attribution in the wild. Language resources and evaluation 45(1), pp. 83–94 (2010)
21. Stamatatos, E.: Author identification using imbalanced and limited training texts. In: Proceedings International Workshop on Database and Expert Systems, DEXA, pp. 237–241, Regensburg (2007)
22. Khmelev, D.V., Teahan, W.J.: A repetition-based measure for verification of text collections and for text categorization. In: Proceedings Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.104–110 (2003)
23. Peng, F., Schuurmans, D., Wang, S.: Augmenting naive Bayes classifiers with statistical language models. Information Retrieval, 7, pp. 317–345 (2004)

24. Burrows, J.D.: A measure of stylistic difference and a guide to likely authorship. Literary and Linguistic Computing 17(3), pp. 267–287 (2002)
25. Escalante, H.J., Solorio, T., Montes-y Gómez, M.: Local histograms of character n-grams for authorship attribution. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 288–298 (2011)
26. Arun, R., Suresh, V., Veni, C.: Stopword graphs and authorship attribution in text corpora. In: International Conference on Semantic Computing, pp. 192–196 (2009)

ISSN 1870-4069

Minería de texto para identificar las principales preocupaciones de los usuarios de Twitter durante COVID-19 en la Ciudad de México

Anabel Pineda-Briseño[1], Josimar Edinson Chire Saire[2]

[1] Tecnológico Nacional de México campus Matamoros,
Departamento de Sistemas y Computación,
México
[2] Institute of Mathematics and Computer Science,
University of São Paulo,
Brazil

anabel.pineda@itmatamoros.edu.mx, jecs89@usp.br

Resumen. COVID-19 es un nuevo coronavirus que originó un brote epidemiológico a principios de año en China. Esta enfermedad se ha diseminado rápidamente por el mundo dando origen a la primera pandemia en la era digital. En este sentido, una de los principales retos de la comunnidad científica internacional es aprovechar de manera efectiva la tecnología para monitorear a la población con el fin de obtener retroalimentación (en tiempo real) de su comportamiento durante cualquier contingencia de salud pública. Una alternativa viable para materializar lo antes planteado son las redes sociales. En una red social las personas generalmente actúan como sensores que identifican e informan rápidamente acontecimientos originados en cualquier parte del mundo, además de compartir datos personales, derivados de su comportamiento e inlcusive de salud. En este trabajo se presenta un estudio a través de minería de texto para identificar las principales preocupaciones de los usuarios de Twitter en la Ciudad de México durante el desarrollo del COVID-19. Después de extraer, visualizar y analizar datos extraídos de Twitter se concluye que las principales prepcupaciones de la población están centradas en temas que tiene que ver con la cantidad de casos confirmados y con las medidas de seguridad sanitarias implementadas por el Gobierno de México.

Palabras clave: COVID-19, minería de texto, principales preocupaciones, procesamiento de lenguaje natural, redes sociales, twitter.

Text Mining for Identifying the Main Concerns of Twitter Users during COVID-19 in Mexico City

Abstract. COVID-19 is a new coronavirus that caused an epidemiological outbreak earlier in the year in China. This disease has spread

rapidly throughout the world, giving rise to the first pandemic in the digital age. In this sense, one of the main challenges of the international scientific community is to effectively take advantage of technology to monitor the population in order to obtain feedback (in real time) on their behavior during any public health contingency. A viable alternative to materialize the aforementioned is social networks. In a social network, people generally act as sensors that quickly identify and report events originating anywhere in the world, in addition to sharing personal data derived from their behavior and even their health. This work presents a study through text mining to identify the main concerns of Twitter users in Mexico City during the development of COVID-19. After extracting, viewing and analyzing data extracted from Twitter, it is concluded that the main concerns of the population are focused on issues that have to do with the number of confirmed cases and with the sanitary security measures implemented by the Government of Mexico.

Keywords: COVID-19, text mining, main concerns, natural language processing, social media, twitter.

1. Introducción

A finales del 2019 en la ciudad de Wuhan, China, se indentificó una nueva enfermedad infecciosa llamada 2019-nCoV, mejor conocida como COVID-19, la cual dio origen a un brote de neumonía viral a principios del 2020. Debido al alto nivel de contagio de esta enfermedad respiratoria su propagación se ha extendido por todos los continentes, motivo por el cual la Organización Mundial de la Salud (OMS) declaró oficialmente al COVID-19 una pandemia [19][17] el día 11 de marzo del 2020. Por lo antes mencionado, COVID-19 se ha convertido en emergencia de salud pública internacional ya que representa un alto riesgo para la humanidad y un impacto negativo en otras áreas como la economía, educación, etc. A nivel mundial actualmente esta enfermedad ha infectado a millones de personas y ha ocasionado la muerte de miles de ellas. Hasta el 15 de mayo del 2020 se han reportado ante la OMS 4,347.935 casos positivos y 297,236 defunciones por COVID-19 [18]. En México, los primeros dos casos de COVID-19 reportados a la OMS fueron del 29 de febrero del 2020, y para el 15 de mayo se reportaban 40,186 casos positivos y 4220 defunciones [10]. En relación a la Ciudad de México, el reporte hasta el 15 de mayo fue de 12,456 casos positivos y de 997 personas fallecidas a causa de esta pandemia [13]. Por otro lado, el crecimiento explosivo del uso de las redes sociales en los últimos años indica que cada vez más personas emplean esta tecnología para identificar e informar acontecimientos, además de compartir información personal, ideas, intereses, sentimientos, experiencias e inclusive información relacionada con la salud. En este contexto, las redes sociales representan una fuente de datos atractiva que puede ser explotada para propósitos de monitoreo y vigilancia de salud pública [6][9]. Entre las redes sociales más usadas destaca Twitter, un servicio gratuito para compartir mensajes de texto limitados a 280 caracteres.

Twitter cuenta a nivel mundial con alrededor de 340 millones de usuarios activos [7], mientras que en México el número de usuarios llega a 9.45 millones, siendo la Ciudad de México la que cuenta con la mayor cantidad de usuarios activos [16]. La principal contribución de este trabajo es identificar a través de minería de texto las principales preocupaciones de la población durante la pandemia COVID-19 en la Ciudad de México, en un esfuerzo por entender cómo se manifiesta y desarrolla una contingencia de salud pública en las redes sociales, específicamente empleando Twitter como fuente de datos.

El resto del artículo se organiza como sigue. En la sección 2 se presenta el trabajo relacionado al uso de datos de redes sociales para fines de vigilancia y monitoreo de enfermedades infecciosas. En la sección 3 se describe la metodología empleada para la extracción, analisis y visualización de los datos de este trabajo. En la sección 4 se presentan los resultados obtenidos. Por último, en la sección 5, se exponen las conclusiones y el trabajo futuro del presente estudio.

2. Trabajo relacionado

Existen diversas propuestas relacionadas al uso de datos de redes sociales y su valiosa contribución en el campo de la salud pública. En esta sección se presentan algunas de las propuestas más recientes. Los proyectos [14] y [2] emplean técnicas de Procesamiento de Lenguaje Natural para generar nueva información a partir de datos recolectados de Twitter, los cuales resultaron ser sumamente útiles para la investigación en salud pública. Las propuestas [3] y [9] presentan un análisis que muestra la efectividad del uso de la información de las redes sociales como estrategia de investigación en salud pública. Ambas propuestas recomiendan utilizar la combinación de redes sociales con otras técnicas para estudiar la enfermedad y su propagación. En [15] se presenta un sistema de tiempo real que ayuda a la predicción y detección de una epidemia a través de la identificación de tuits de enfermades por localización geográfica. El trabajo propuesto en [8], combina datos de Twitter con datos de Google Trends para realizar un seguimiento de la propagación de enfermedades infecciosas. Otro estudio que se presentó en [1], analizó datos de Twitter que fueron recolectados durante algunos brotes de enfermedades infecciosas. Los resultados experimentales ayudaron a entender cómo actúa la gente con transtorno de pánico en medio de una contingencia de salud. Existen también otras propuestas relacionadas a aplicaciones de monitoreo y vigilancia de enfermedades infecciosas que han empleado como fuente de datos Twitter, entre las que destacan: Monitor de la Pandemia N1H1 [4], Monitor del Dengue en Brasil [12], y Monitor del COVID-19 en Colombia [11] y Sudamérica [5]. Ahora bien, debido a que COVID-19 es una enfermedad de la cual se conoce muy poco, y que ha ocasionado un impacto negativo en diversos ámbitos en todos los países, en este artículo se presenta un estudio exploratorio encaminado a identificar las principales inquietudes de los usuarios de Twitter en México, específicamente de la Ciudad de México, una de las ciudades más pobladas del mundo y de América Latina.

3. Metodología

Este trabajo realiza experimentos sobre texto proveniente de Twitter, y aplica técnicas de Procesamiento de Lenguaje Natural. Básicamente la metodología se resume en:

- Selección palabras para la búsqueda en red social.
- Delimitación de parámetros para la búsqueda.
- Preparación del texto para ser procesado.
- Visualización.

3.1. Selección de palabras relevantes

Se consideran las palabras utilizadas en las noticias, experimentos previos de búsqueda en la red social. Se seleccionaron las siguientas palabras: coronavirus y covid19. Durante las búsquedas se encontraron variaciones como: corona-virus, corona_virus, covid-19, #covid_19, por tanto se crearon e incluyeron las combinaciones con caracteres: @, #, –, _.

3.2. Delimitación de parámetros para la búsqueda

La recolección de datos se realiza a través de la Interfaz de Programación de Aplicaciones(API) de Twitter, los parámetros usados son:

- Fechas: 13-03-2020 a 20-03-2020 y 01-05-2020 a 09-05-2020.
- Palabras claves: Mencionados en la subsección anterior.
- Geolocalización: Latitud y Longitud: (19.4333,-99.1333), ver Fig.1 1.
- Idioma: Español.
- Radio: 50 kilómetros.

Fig. 1. Geolocalización de la Ciudad de México.

3.3. Preparación del texto para ser procesado

Este paso es importante para limpiar el texto de símbolos, palabras que no añaden valor al análisis del presente trabajo, etc. El proceso consiste en términos generales en:

- Convertir de mayúsculas a minúsculas.
- Eliminar signos de puntuacón como: comas, puntos, puntos y comas, signos de interrogación, signos de exclamación, entre otros.
- Excluir palabras con un tamaño menor a 3.
- Agregar excepciones, por ejemplo: https, url, etc.

3.4. Visualización

Los datos pre-procesados son empleados para responder preguntas durante la exploración de la información. Una adecuada visualización de los valores resultantes a través de gráficas ayuda de manera significativa en la realización de un buen análisis de la información. Este trabajo se enfoca en graficar información relacionada a:

- Mostrar el progreso de la pandemia graficando el número casos positivos por día y acumulados.
- Frecuencia de publicación de los usuarios de Twitter tanto del inicio del brote de la pandemia como durante el pico máximo de brotes de ésta.
- Histograma de los términos más publicados al inicio del brote de la pandemia y durante el pico máximo de brotes de ésta.
- Nubes de palabas (unigramas) más publicadas por día, tanto del inicio del brote de la pademia como del pico máximo de brotes de la misma.
- Nubes de pares de palabras (bigramas) más publicados por día, tanto del inicio del brote de la pademia como del pico máximo de brotes de la misma.

4. Resultados

En esta sección se presentan los resultados del estudio propuesto en este artículo. Para una mejor organización esta sección se ha dividido en cuatro subsecciones. La primera despliega a manera de introducción el progreso de la pandemia en la Ciudad de México. En la segunda subsección se muestran las gráficas de los datos extraídos correspondientes al inicio de brote de la pandemia. La tercera subsección despliega las gráficas de los datos extraídos correspondientes al pico máximo de brotes de la misma. Finalmente, la última subsección presenta un análisis general de los hallazgos del estudio.

4.1. Progreso de la pandemia COVID-19

En la Fig. 2 se presenta el progreso de los casos confirmados, por día y acumulados, del COVID-19 en la Ciudad de México. Los datos fueron extraídos del sitio oficial Coronavirus de la Secretaría de Salud del Gobierno de México [13]. De acuerdo a la información disponible en el portal, los primeros casos del COVID-19 en la ciudad se reportaron a finales del mes de febrero del 2020, mientras que para el periodo entre el 6 y el 10 de mayo, la Secretaría de Salud proyectaba el pico máximo de brotes de la pandemia. Hasta el 15 de mayo, fecha que se consultó el portal del Coronavirus de la Secretaría de Salud del Gobierno de México [13], se tenían registrados un total de 12,456 casos del COVID-19 confirmados, lo cual representa un promedio 95.8 contagios diarios.

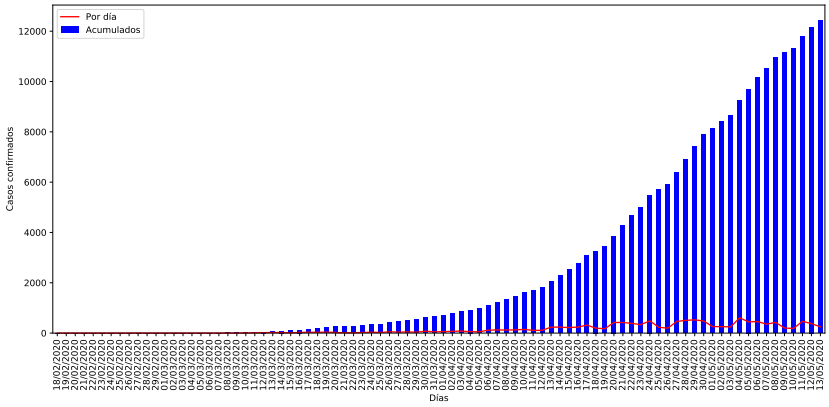


Fig. 2. Casos confirmados del COVID-19 en la Ciudad de México.

4.2. Con cuánta frecuencia y qué publicaron los usuarios de Twitter acerca del COVID-19 durante su brote en la Ciudad de México

Con el propósito de saber con qué frecuencia y qué publicaron los usuarios de Twitter acerca del COVID-19 durante el brote de casos en la Ciudad de México, se extrajeron datos de Twitter del periodo comprendido entre el 13-03-2020 y el 20-03-2020. Este periodo fue seleccionado ya que en este se identificó un incremento constante de casos positivos, tal como se puede constatar en la Fig.2. Para identificar con qué frecuencia se publicó acerca del COVID-19, en la Fig. 3 se despliega el número de tuits por día. Se puede observar en la gráfica que conforme se fueron confirmando casos, la cantidad de publicaciones incrementó, concentrándose la mayor parte en la segunda mitad del periodo. En total se contabilizaron un total de 22,755 mensajes de texto, que representan en promedio 2844 mensajes por día.

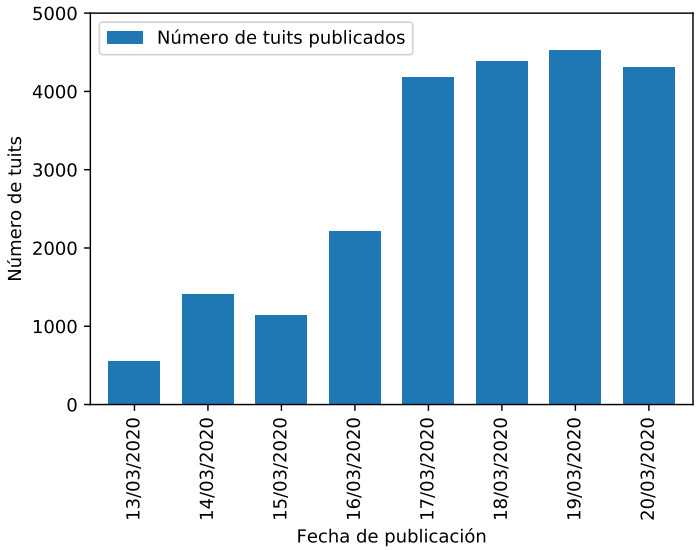


Fig. 3. Publicaciones diarias acerca del COVID-19 en la Ciudad de México al inicio del brote de la pandemia.

Por otro lado, con el fin de caracterizar el comportamiento o las preocupaciones de los usuarios de Twitter durante este periodo, fue interesante identificar y analizar las palabras más empleadas y su frecuencia de publicación. En la Fig.4 se despliegan las 30 palabras más publicadas durante el inicio del brote del COVID-19 en la Ciudad de México. Como se puede observar en la lista clasificada, las cinco palabras más populares fueron "casos", "presidente", "salud", "medidas" y "gobierno". Ahora bien, con el fin de poder tener un panorama general de lo que más se publicó en este periodo por día, se utilizaron nubes de palabras para representar gráficamente las palabras (unigramas) y pares de palabras (bigramas) más publicadas. Las Fig. 5 y 6 muestran este conjunto de nubes de palabras y pares de palabras construidas en base a los datos recolectados. En ambas representaciones, palabras y pares de palabras, se confirma la popularidad por día de la palabra "casos" y del par de palabras "casos confirmados", respectivamente.

4.3. Con cuánta frecuencia y qué publicaron los usuarios de Twitter acerca del COVID-19 durante el pico máximo de brotes en la Ciudad de México

Con el propósito de saber con qué frecuencia y qué publicaron los usuarios de Twitter acerca del COVID-19 durante el pico máximo de brotes en la Ciudad de México, se extrajeron datos de Twitter del periodo comprendido entre el 01-05-2020 y el 09-05-2020.

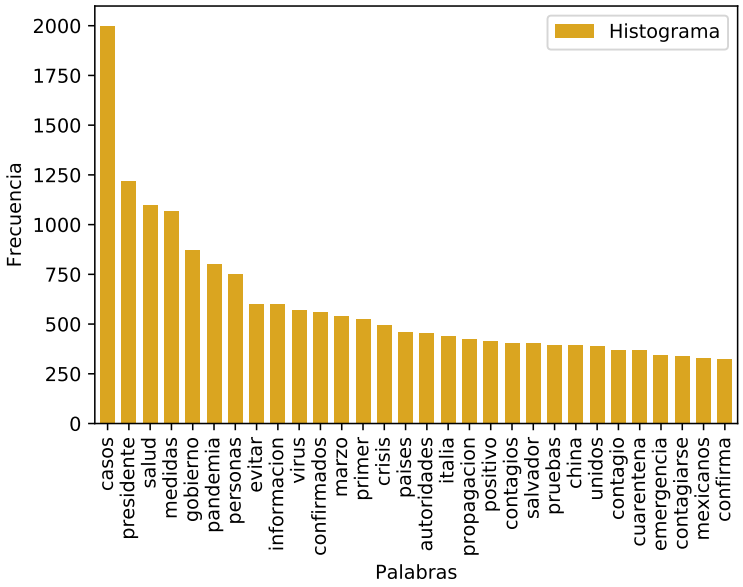


Fig. 4. Frecuencia de las palabras más publicadas acerca del COVID-19 al inicio del brote en la Ciudad de México.

Fig. 5. Nubes de palabras (unigramas) más publicados por día durante el inicio del brote del COVID-19 en la Ciudad de México.

Fig. 6. Nubes de pares de palabras (bigramas) más publicadas durante el inicio del brote del COVID-19 en la Ciudad de México.

De acuerdo a la Secretaría de Salud del Gobierno de México, dentro de este periodo de fechas se proyectó de manera inicial el pico máximo de brotes del COVID-19 en México. De manera similar a la subsección anterior, para identificar con qué frecuencia se publicó acerca del COVID-19 se graficó la cantidad de tuits diarios. Se puede observar en la Fig.7 que en términos generales la distribución de la frecuencia de publicación fue homogénea. Para este periodo fueron recuperados un total 20,703 tuits, que representan en promedio 2300 mensajes publicados por día.

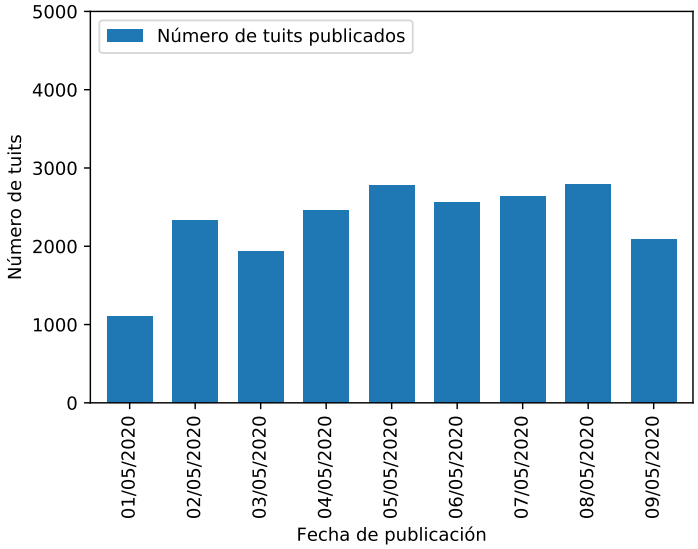


Fig. 7. Publicaciones diarias acerca de COVID-19 en la Ciudad de México durante el pico máximo de bronte de la pandemia.

En relación a las palabras más publicadas por la comunidad de Twitter en este periodo de exploración, en la Fig.8 se presenta un histograma de estas. Las palabras más populares de acuerdo a la lista clasificada son: “casos”, “gobierno”, “pandemia”, “pacientes” y “salud”. De igual manera son presentadas las gráficas de nubes de palabras más publicadas por día y nubes de pares de palabras más publicados por día. En la Fig. 9 se puede observar que la palabra más popular fue “casos”, mientras que en la Fig. 10 se identificó que en general el par de palabras más empleado por día fue “casos confirmados”.

4.4. Análisis comparativo de resultados

En esta sección se presenta un análisis comparativo de los resultados obtenidos en ambos periodos de estudio.

En primer lugar, se puede observar en la Fig.3 que durante el inicio del brote, como era de esperarse, las publicaciones diarias acerca de COVID-19 fueron

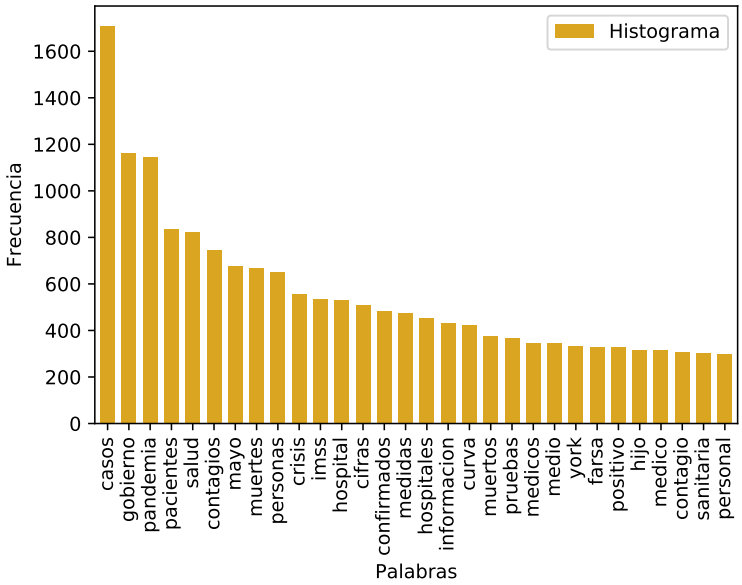


Fig. 8. Frecuencia de palabras más publicadas durante pico máximo de brotes de COVID-19 en la Ciudad de México.

Fig. 9. Nubes de palabras (unigramas) más publicados por día durante el pico máximo de brotes de COVID-19 en la Ciudad de México.

incrementando conforme se fueron reportando casos positivos. Sin embargo, de acuerdo con la Fig.7, durante el pico máximo de contagios esto no fue así, en este segundo periodo de estudio la frecuencia de publicación acerca del COVID-19 mostró en términos generales una distribución homogénea y un porcentaje menor (19.13 %) de publicaciones con respecto al inicio del brote de la pandemia.

Algo importante a destacar es que durante el pico máximo de contagios, la

Fig. 10. Nubes de pares de palabras (bigramas) más publicadas por día durante el pico máximo de brotes de COVID-19 en la Ciudad de México.

población de la Ciudad de México se encontraba en cuarentena, por lo que los datos sugieren que el exceso de información relacionada al COVID-19 pudo haber ocasionado la disminución de publicaciones debido al estrés, ansiedad y temor de la población por la emergencia sanitaria. Como información de referencia, el gobierno de México declaró cuarentena a partir del 20 de marzo del 2020, esto como parte de las estrategias para contener los contagios una vez que el país entró en una etapa de emergencia sanitaria.

Para distinguir de mejor manera cuáles fueron los temas que más han preocupado a la población de la Ciudad de México durante el inico de la pandemia y durante el pico máximo de brotes, se identificaron las 30 palabras más publicadas por periodo, destacando que coinciden un 43.33 % de las palabras más publicadas. Las tres palabras más publicadas que coinciden en el top 5 de ambas listas clasificadas fueron: “casos”, “gobierno” y“salud”, que sugieren temas relacionados con el número de casos confirmados y con las medidas de seguridad sanitarias implementadas por el Gobierno de México. En relación al resto de palabras que discreparon (56.66 %), y con el apoyo de las nubes de palabras (unigramas) y pares de palabras (bigramas), se logró identificcar otros temas de interés de la población. Por ejemplo, al inicio del brote de la pandemia la población además mostraba su preocupación publicando sobre las formas y la posibilidad de contagio; la carencia de equipos de protección médica; y la difusión de noticias falsas.

Por su parte, durante el pico máximo de brotes la población también manifestó preocupación comentando sobre la crisis sanitaria y económica; el apoyo de la Secretaría de la Defensa Nacional (Sedena) y la Secretaría de la Marina-Armada de México (Marina) con el “Plan de Auxilio a la población civil en casos de desastre” (Plan DN-III); las acusaciones lanzadas por cuatro diarios interna-

cionales a la Secretaría de Salud sobre el presunto ocultamiento de cifras de decesos y casos positivos COVID-19 en México; y por la compra de ventiladores para el COVID-19 a precios excesivos por parte del Gobierno Mexicano al hijo del director de la Comisión Federal de Electricidad (CFE) de México.

5. Conclusiones y trabajo futuro

En este artículo se presentó un estudio basado en minería de texto con el objetivo de identificar las principales inquietudes de los usuarios de Twitter durante la propagación del COVID-19 en la Ciudad de México. Los escenarios de exprimentación se enfocaron en extraer, visualizar y analizar datos durante el inicio del brote de la pandemia y durante el pico máximo de brotes de la misma. De acuerdo a los resultados se concluye que los temas centrales están fuertemente relacionados con el número de casos confirmados y con las medidas de seguridad sanitarias implementadas por el Gobierno de México, estas últimas a su vez generando el miedo por una posible crisis económica en la población de la Ciudad de México, una de las ciudades más pobladas del mundo y de América Latina. Como trabajo futuro derivado de esta investigación se pretende ampliar el campo de estudio a otras ciudades de México. Asimismo, medir el impacto mediático y el impacto que la pandemina está generando en otras áreas como la educación y la salud mental. Además de un estudio dedicado exclusivamente a monitorear el desarrollo del COVID-19 en la frontera México-USA.

Referencias

1. Ahmed, W., Bath, P.A., Sbaffi, L., Demartini, G.: Moral panic through the lens of Twitter: An analysis of infectious disease outbreaks. In: Proceedings of the 9th International Conference on Social Media and Society. pp. 217–221 (2018)
2. Breland, J.Y., Quintiliani, L.M., Schneider, K.L., May, C.N., Pagoto, S.: Social media as a tool to increase the impact of public health research. American journal of public health 107(12), 1890 (2017)
3. Charles-Smith, L.E., Reynolds, T.L., Cameron, M.A., Conway, M., Lau, E.H., Olsen, J.M., Pavlin, J.A., Shigematsu, M., Streichert, L.C., Suda, K.J., et al.: Using social media for actionable disease surveillance and outbreak management: a systematic literature review. PloS one 10(10) (2015)
4. Chew, C., Eysenbach, G.: Pandemics in the age of Twitter: content analysis of tweets during the 2009 H1N1 outbreak. PloS one 5(11) (2010)
5. Chire Saire, J.E.: Infoveillance based on social sensors to analyze the impact of Covid19 in South American population (2020)
6. Christaki, E.: New technologies in predicting, preventing and controlling emerging infectious diseases. Virulence 6(6), 558–565 (2015)
7. Clement, J.: Global social networks ranked by number of users 2020. Accedido en 16-04-2020 a url https://www.statista.com/ (2020)
8. Hong, Y., Sinnott, R.O.: A social media platform for infectious disease analytics. In: International Conference on Computational Science and Its Applications. pp. 526–540. Springer (2018)

9. Paul, M.J., Sarker, A., Brownstein, J.S., Nikfarjam, A., Scotch, M., Smith, K.L., Gonzalez, G.: Social media mining for public health monitoring and surveillance. In: Biocomputing 2016: Proceedings of the Pacific symposium. pp. 468–479. World Scientific (2016)
10. Rios Montanez, A.M.: Mexico: COVID-19 cases and deaths 2020. URL: https://www.statista.com/ [accessed 2020-05-15] (2020)
11. Saire, J.E.C., Navarro, R.C.: What is the people posting about symptoms related to coronavirus in Bogota, Colombia? arXiv preprint arXiv:2003.11159 (2020)
12. Saire, J.E.C.: Building intelligent indicators to detect dengue epidemics in Brazil using social networks. In: 2019 IEEE Colombian Conference on Applications in Computational Intelligence (ColCACI). pp. 1–5. IEEE (2019)
13. Secretaría de Salud, México: Sitio oficial del gobierno de México sobre el coronavirus. URL: https://coronavirus.gob.mx/ [accessed 2020-05-15] (2020)
14. Sinnenberg, L., Buttenheim, A.M., Padrez, K., Mancheno, C., Ungar, L., Merchant, R.M.: Twitter as a tool for health research: a systematic review. American journal of public health 107(1), e1–e8 (2017)
15. Sivasankari, S., Kavitha, M., Saranya, G.: Medical analysis and visualisation of diseases using tweet data. Research Journal of Pharmacy and Technology 10(12), 4306–4312 (2017)
16. Statista Research Department: Latin America: Twitter users 2020, by country. Accessed on 16-04-2020, url https://www.statista.com/ (2020)
17. Wang, C., Horby, P.W., Hayden, F.G., Gao, G.F.: A novel coronavirus outbreak of global health concern. The Lancet 395(10223), 470–473 (2020)
18. WHO: Coronavirus (COVID-19). URL: https://covid19.who.int/ [accessed 2020-05-15] (2020)
19. WHO: WHO statement regarding cluster of pneumonia cases in Wuhan, China. Beijing: WHO 9 (2020)

ISSN 1870-4069

Reconocimiento facial basado en Least-Squares Support Vector Machine y características LNDP

Pablo Augusto Campos-Zárate, Efrén Gorrostieta-Hurtado,
Jesús Carlos Pedraza-Ortega

Universidad Autónoma de Querétaro,
Facultad de Ingeniería,
México

{pacaza60, efrengorrostieta}@gmail.com,
caryoko@yahoo.com

Resumen. El reconocimiento facial en 3D se ha vuelto de gran interés para los investigadores en el campo de la visión artificial debido a las ventajas que posee contra el reconocimiento tradicional en 2D, en este trabajo se presenta un sistema de reconocimiento facial en 3D que utiliza características LNDP del rostro junto con un clasificador LSSVM que conserva las ventajas del SVM tradicional sin la necesidad de resolver un problema de programación cuadrático. Los experimentos realizados en la *Texas 3DFRD* muestran que este sistema es capaz de verificar e identificar rostros de la base de datos con valores de 99 %.

Palabras clave: Reconocimiento facial, clasificación multiclase, least-squares support vector machine, normales de superficie.

Facial Recognition Based on Least Squares Support Vector Machine and LNDP Features

Abstract. 3D face recognition has become very popular among computer vision reasearchers due to its advantajes against traditional 2D face recognition, this paper presents a 3D face recognition LNDP+LSSVM-based system wich keeps the benefits of a traditional SVM without having to solve a quadratic-programming problem but a linear-programming problem. Experiments conducted on the Texas 3DFR Database show that LNDP-LSSVM method can achieve verification and recognition rate over 99 %.

Keywords: Face recognition, multiclass classification, least-squares support vector machine, surface normals.

1. Introducción

Un sistema de reconocimiento de rostros puede trabajar de dos formas, para autenticación o para identificación facial [1], la tarea del reconocimiento, que es

el tema de interés en este trabajo, consiste en dada una imagen de entrada que se compara contra todas las imágenes en una base de datos para determinar la identidad de la imagen de entrada. El reconocimiento facial ha atraído la atención de los investigadores debido a sus numerosas aplicaciones como seguridad y vigilancia, el reconocimiento facial utiliza información biométrica sin la necesidad de un contacto directo con el individuo que se está trabajando. Aunque existe mucho trabajo relacionado con el reconocimiento tradicional (en 2 dimensiones), este aún presenta retos cuando se enfrenta a variaciones de iluminación, pose y expresiones. El reconocimiento en 3D surgió como una solución a las dificultades inherentes a las variaciones de pose e iluminación debido a que aprovecha la información geométrica completa del rostro, aunque de igual manera sigue enfrentando retos cuando se trata de expresiones, variaciones de pose extremas y oclusiones. De acuerdo con [2], los algoritmos de reconocimiento pueden dividirse en 3 categorías: Métodos holísticos, métodos locales y métodos híbridos.

El enfoque holístico utiliza toda la información del rostro, los métodos basados en *Principal Component Analysis* (PCA) en [3,4] entran en dicho enfoque, aunque muchos trabajos se han realizado dentro de esta categoría con buenos resultados, aún se enfrentan dificultades frente a oclusiones y datos faltantes debido a que la mayoría de estos métodos requieren un modelo completo del rostro, por lo que los métodos basados en características locales han atraído la atención de investigadores debido a que ofrecen mayor robustez frente a oclusiones, expresiones y valores faltantes así como variaciones de pose, una revisión de los métodos del estado del arte que utilizan características locales es presentado en [5].

En [6] se presentó un método para el reconocimiento de expresiones faciales haciendo una combinación de características extraídas por PCA y LBP, logrando un híbrido de características globales y locales alimentadas después a una *Support Vector Machine* (SVM). Más trabajo con información biométrica en 3D del rostro es realizado en [7], el proceso se divide en 4, a) adquisición de datos, b) extracción de la cabeza, c) identificación del plano de la cabeza y d) clasificación de la forma de la cabeza. El proceso de la clasificación es también realizado mediante una SVM. Dentro de los avances en reconocimiento facial haciendo uso de características locales y SVM, se encuentra el propuesto en [8] que propuso una manera de extraer información facial llamada *Angular Radial Signature* (ARS), mismas características que fueron alimentadas a una SVM con un Kernel PCA obteniendo un alto desempeño tanto en reconocimiento como en eficiencia. Para lo mejor de nuestro conocimiento, no se ha combinado el descriptor LNDP con una LS-SVM para obtener un sistema de reconocimiento facial, por lo que este trabajo se enfocó en conocer las capacidades de dicho sistema.

2. Métodos

2.1. Descriptor utilizado

El descriptor *Local Derivative Pattern* (*LDP*) [9] está inspirado en el descriptor *Local Binary Pattern* [10] (*LBP*) que fue propuesto para describir texturas en

imágenes. El LDP aprovecha información direccional de cada píxel de la imagen, obteniendo así mejores características a diferencia del descriptor no direccional LBP. Posteriormente [11] propuso una variación del LDP que aprovecha la información de las normales de una superficie para extraer las características de objetos 3D.

Normales de superficie Para la estimación de los vectores normales a la superficie, dado un conjunto de puntos $P = \{p_1, p_2, \ldots, p_n\}$ en una nube de puntos $p_i \in R^3$ donde $\mathbf{p}_i = [p_{ix}, p_{iy}, p_{iz}]^T$ representa las coordenadas 3D de cada punto y a cada punto le corresponde un vector normal $\mathbf{n}_i = [n_{ix}, n_{iy}, n_{iz}]^T$, se define un conjunto de k puntos vecinos para cada punto p_i, de manera que el conjunto de puntos vecinos a p_i es $Q_i = \{q_{i1}, q_{i2}, \ldots, q_{ik}\}$. De esta forma la matriz aumentada de vecinos es:

$$\mathbf{Q}_i^+ = [p_i, q_{i1}, q_{i2}, \ldots, q_{ik}]^T. \tag{1}$$

Para encontrar el vector n_i correspondiente a p_i, se ajusta un plano a la matriz (1) que minimice la distancia con cada punto, así el vector normal al plano, también será el vector normal al punto p_i [12]:

$$\min_{n_i} ||[\mathbf{Q}_i^+ - \bar{\mathbf{Q}}_i^+] \, \mathbf{n}_i||, \tag{2}$$

donde $\bar{\mathbf{Q}}_i^+ = \mathbf{1}_k + \frac{1}{k+1}(\mathbf{p}_i + \sum_{j=1}^{k} \mathbf{q}_{ij})$, esto es equivalente a calcular el *Principal Component Analysis* en $\mathbf{Q}_i^+$ y tomar el componente principal con menor covarianza.

En [12] también se hace mención a los métodos para la obtención de los puntos vecinos a p_i.

Local Normal Derivative Pattern La teoría detrás de este descriptor es muy similar a la propuesta para el descriptor LDP, para un componente normal $N(P)$ el LNDP de primer orden en 4 direcciones distintas $\alpha = 0°, 45°, 90°, 135°$ es [9,11] :

$$N'_{0°}(p_i) = N(p_i) - N(Q_{i4}), \tag{3}$$
$$N'_{45°}(p_i) = N(p_i) - N(Q_{i3}), \tag{4}$$
$$N'_{90°}(p_i) = N(p_i) - N(Q_{i2}), \tag{5}$$
$$N'_{135°}(p_i) = N(p_i) - N(Q_{i1}), \tag{6}$$

y el LNDP de orden n para un determinado α es como sigue:

$$LNDP_\alpha^n(p_i) = \{f(N_\alpha^{n-1}(p_i), N_\alpha^{n-1}(q_{i1})), f(N_\alpha^{n-1}(p_i), N_\alpha^{n-1}(q_{i2})), \ldots, f(N_\alpha^{n-1}(p_i), N_\alpha^{n-1}(q_{ik}))\}, \tag{7}$$

donde k es el número de vecinos de p_i:

$$f(N_\alpha^{n-1}(p_i), N_\alpha^{n-1}(q_{ij})) = \begin{cases} 0, & si \quad N_\alpha^{n-1}(q_{ij}).N_\alpha^{n-1}(p_i) > 0 \\ 1, & si \quad N_\alpha^{n-1}(q_{ij}).N_\alpha^{n-1}(p_i) \leq 0 \end{cases} \quad j = 1, \ldots, k. \tag{8}$$

Para cada píxel se concatenan los LNDP en las 4 direcciones y se forma un histograma espacial para cada componente normal:

$$LNDP^n(p_i) = \{LNDP_\alpha^n | alpha = 0°, 45°, 90°, 135°\}. \tag{9}$$

El descriptor final se obtiene concatenando los histogramas de las tres componentes normales.

2.2. Least Squares Support Vector Machines

Las LS-SVM son una modificación de las SVM tradicionales, que fue presentada por [15], que simplifica la formulación de las SVM de Vapnik [13,14] sin perder las ventajas que estas ofrecen. De esta forma el problema de optimización se convierte en un problema de programación lineal (LP) en lugar de un problema de programación cuadrático (QP). El planteamiento del problema propuesto por Suykens consiste en:

$$\underset{w,b,e}{min} J_p(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{l} e_k^2, \tag{10}$$

sujeto a

$$y_k[w^T \phi(x_k) + b] = 1 - e_k, \quad k = 1, \ldots, l.$$

El Lagrangiano del problema 10 es:

$$L(w, b, e; \alpha) = J_p(w, e) - \sum_{k=1}^{l} \alpha_k \{y_k[w^T \phi(x_k) + b] - 1 + e_k\}, \tag{11}$$

después de obtener las derivadas parciales de 11, el clasificador se puede encontrar resolviendo el sistema de ecuaciones:

$$\begin{bmatrix} 0 & y^T \\ y & \Omega + I/\gamma, \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix} \tag{12}$$

donde $Z^T = [\phi(x_1)^T y_1; \ldots; \phi(x_l)^T y_l]$, $y = [y_1; \ldots; y_l]$,$1_v = [1; \ldots; 1]$ y $\Omega = Z^T Z$. El *kernel trick* puede realizarse dentro de la matriz Ω. Así el clasificador es:

$$y(x) = sign \left[\sum_{k=1}^{l} \alpha_k y_k K(x, x_k) + b \right]. \tag{13}$$

Para el problema de la clasificación multiclase se utilizó en enfoque *OnevsAll* debido a que requiere menor trabajo computacional, en este se entrenan tantos

clasificadores binarios como clases distintas, si se tiene k individuos deben ser entrenados k clasificadores $y = \mathbf{w} \cdot \mathbf{x} + b$. Dada una imagen de entrada $\mathbf{x}$ se obtienen $y_q(\mathbf{x})(q = 1, \ldots, k)$, de esta forma para verificación se establece un umbral t y la muestra se acepta si $t - y_q(\mathbf{x}) \geq 0$ de lo contrario la muestra se rechaza. Para identificación la etiqueta de $\mathbf{x}$ se asigna como:

$$etiqueta(\mathbf{x}) = \underset{1 \leq q \leq k}{argmax}(y_q(\mathbf{x})). \tag{14}$$

2.3. Metodología y experimentación

El enfoque general propuesto para un sistema basado en una LSSVM consiste en lo siguiente, primero la base de datos es dividida en un conjunto de entrenamiento y un conjunto de prueba, para cada imagen en el conjunto de entrenamiento se hace un re-escalado al 10 %, las imágenes reducidas se utilizan como entrada para el algoritmo LNDP, mismo que retorna los vectores de características LNDP para cada imagen, estos vectores son utilizados para entrenar una LSSVM, las imágenes en el conjunto de prueba pasan por el mismo proceso de re-escalación y extracción de características antes de poder ser alimentadas a la LSSVM entrenada para la predicción de las nuevas etiquetas, cada uno de los pasos aquí mencionados será detallado más adelante en esta sección.

La base de datos utilizada es la Texas 3DFRD [16,17], cuenta con modelos 2D y 3D de 118 individuos en un rango de edad de 22 a 75 años aproximadamente de distintos grupos étnicos, hay modelos para rostros con expresión neutral y expresiones faciales, dado que la base de datos presenta los modelos normalizados, este trabajo se enfocó principalmente en el trabajo del método propuesto. Las imágenes de rango originales son de 751x501 pixeles donde el píxel con mayor intensidad representa el punto más cercano entre el rostro y el dispositivo de captura. Para la experimentación cada imagen en la base de datos se re-escaló a un tamaño de 75x50 pixeles por medio de una interpolación bilineal, esto es necesario debido a la cantidad de operaciones que se requieren para el cálculo de las normales de superficie, la Figura 1 muestra el comportamiento del tiempo cuando se varía el tamaño de la imagen con escalas entre el 10 % y el 100 %, como es claro el comportamiento se asemeja a un comportamiento de un algoritmo $O(n^3)$, que es precisamente lo que se espera de este algoritmo de acuerdo con [12], por lo que se consideró un tamaño de imagen al 10 % como adecuado para poder competir con otros algoritmos en términos de tiempo de cómputo. Para las normales de superficie, se utilizó una vecindad de los 25 puntos más cercanos utilizando las implementaciones de *k-Nearest Neighbor* y PCA del software Matlab, de acuerdo con lo experimentos de [12] utilizar más de 25 puntos en la vecindad para la estimación no mejora de manera significativa la calidad de las normales.

Para la extracción de las características LNDP corresponda a características locales, cada imagen se dividió en parches de 5x5 y a cada parche se aplicó el descriptor LNDP, los histogramas de todos los parches se concatenaron para formar el vector final de características LNDP, el grado del descriptor utilizado

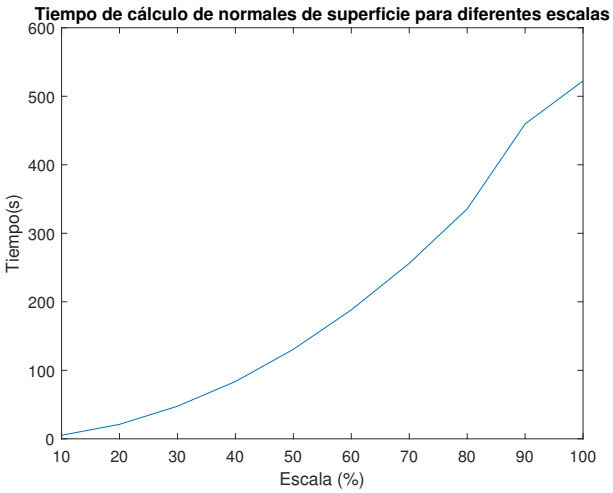


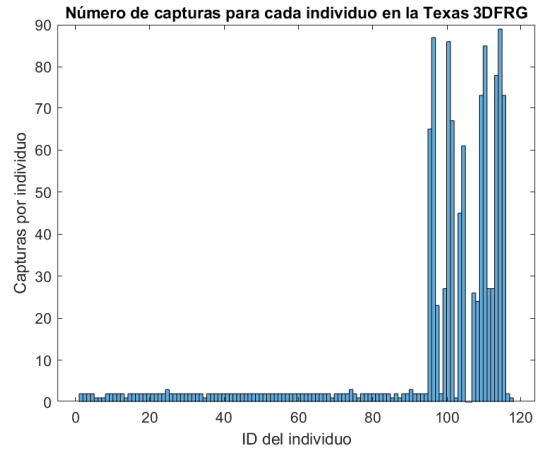
Fig. 1. Tiempo de cómputo de caracterísitcas LNDP para una imagen a diferentes escalas.

es $n = 2$. Por propósitos prácticos, se extrajeron las características de todas las imágenes en la base de datos y se almacenaron como en un archivo .csv mismo del que se tomaron los vectores para el entrenamiento y pruebas.

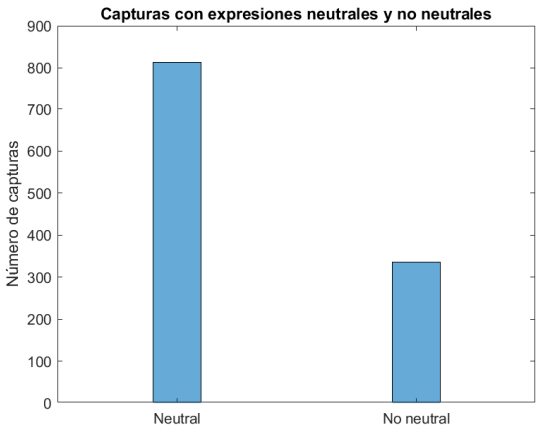
La Figura 2 muestra la cantidad de imágenes que hay por cada sujeto y la cantidad de imágenes que corresponden a expresiones faciales neutrales y no neutrales en la base de datos, como se observa en 2.a, los individuos con más de 5 capturas son pocos en comparación con los que tienen menos de 5, siendo que la mayoría tiene 2 capturas, también existe un desbalance entre las imágenes con expresiones neutrales y no neutrales, para que las comparaciones con otros algoritmos sean justas se utilizó la partición de datos propuesta en [16], el conjunto de entrenamiento consiste de 30 capturas aleatorias de 12 individuos distintos, el conjunto de prueba se dividió en 2: *gallery* y *probe*, la galería consiste de 105 imágenes neutrales de 103 individuos, la prueba contiene imágenes neutrales y no neutrales de 95 individuos distintos, el conjunto de entrenamiento se utilizó tal como está propuesto, para el conjunto de *testing* se utilizaron las imágenes de *probe* de personas que existen en el entrenamiento y por lo tanto son conocidas por el sistema. El entrenamiento de la LSSVM se realizó utilizando el paquete de LSSVMLab [18] con un kernel polinomial y una codificación 1vA.

2.4. Evaluación comparativa

Como base para comparar el desempeño del algoritmo se tomaron en cuenta los algoritmos [8] y [20], para una justa comparación, los tres algoritmos se ejecutaron en el mismo computador y con los mismos datos de entrenamiento y



(a)



(b)

Fig. 2. Histogramas de las capturas en la base de datos Texas 3DFRD. a) Número de capturas por cada individuo. b)Número de capturas que tienen expresiones neutrales y no neutrales

prueba, no se agregó ningún paso extra de normalización ni correcciones a las imágenes.

Lo primero que se verificó fue el tiempo que toma la extracción de características para cada algoritmo, la Tabla 1 muestra el tiempo que toma la extracción de características para los 3 métodos en un sola imagen, como era de esperarse el método de extracción LNDP toma el mayor tiempo dentro de los 3 debido a la cantidad de operaciones que deben realizarse para la extracción.

Tabla 1. Tiempo de extracción de características de los métodos evaluados.

Método	Tiempo (s)	Std (s)
LBP	0.0477	0.0017
ARS	0.0000536	0.00014
LNDP	5.92	0.1836

Los tiempos de entrenamiento para cada algoritmo también fueron computados, en la Tabla 2 se observa que el tiempo de entrenamiento para el algoritmo propuesto es ligeramente menor que el tiempo de [20], el tiempo para [8] aumenta debido a la transformación de los datos previa al entrenamiento.

Tabla 2. Tiempos de entrenamiento para los 3 algoritmos evaluados.

Método	Tiempo(s)
SVM+ARS [8]	1.62
SVM+LBP [20]	0.0897
Propuesto	0.0414

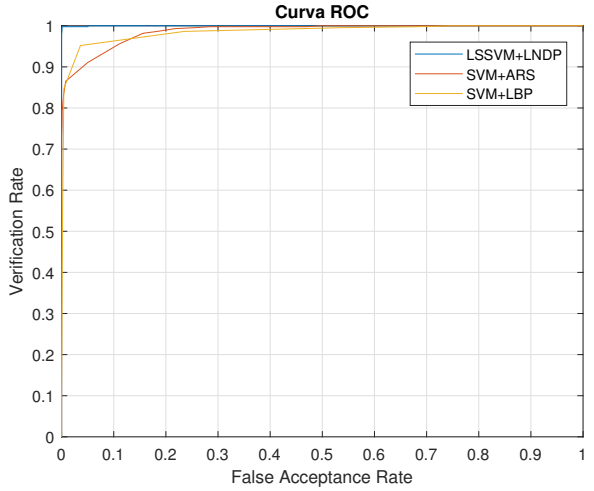
Finalmente el desempeño de los algoritmos se midió para las tareas de verificación e identificación utilizando las curvas ROC y CMC respectivamente, la Figura 3 presenta las curvas para los tres algoritmos evaluados, se observa que el método propuesto es superior, en lo que a *Verification rate* y *Recognition rate* respecta, la Tabla 3 compara los valores de VR y FAR para los 3 distintos métodos, en donde el método propuesto destaca en ambas métricas.

Tabla 3. Verification rate y Recognition rate para puntos fijos en 0.1 FAR y Rank-1

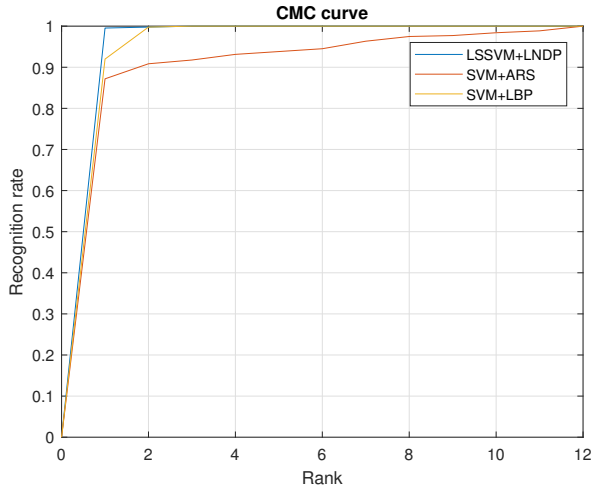
Método	VR en 0.1 FAR (%)	RR en Rank-1 (%)
SVM+ARS[8]	95.65	87.19
SVM+LBP[20]	95.8	91.99
Propuesto	**99.54**	**99.77**

3. Discusión

Con los datos utilizados para entrenamiento y pruebas con imágenes de la *Texas 3DFRD* [16,17], el método LSSVM+LNDP ofrece un alto desempeño en las tareas de identificación y verificación en comparación los otros métodos revisados, el apartado en el que este método es superado es en el tiempo de la extracción de características, donde los otros dos métodos resultaron ser altamente eficientes en comparación. El método propuesto tiene un alto desempeño cuando la base de datos contiene información con poca variabilidad en cuanto a



(a)



(b)

Fig. 3. Curvas ROC y CMC para los algoritmos evaluados. Los valores de Recognition rate en Rank-1 y Verification rate en 0.1 FAR son 99.54 % y 99.77 % respectivamente.

expresiones e información faltante respecta y con pocos individuos distintos a discriminar, más experimentos con diferentes datos deben realizarse para verificar el desempeño del método con datos con mayor variabilidad. En LSSVM+LNDP se pueden obtener mejores resultados en verificación e identificación a un costo computacional mayor aunque aceptable aún.

4. Conclusiones

Se ha presentado un sistema para reconocimiento facial que desempeña las tareas de verificación e identificación con desempeño del 99 % en la base de datos *Texas 3DFRD*, permitiendo discriminar entre individuos con imágenes con expresiones neutrales y no neutrales, el uso del descriptor LNDP permite obtener mejor información para discriminación que los descriptores ARS y LBP aunque a un mayor costo computacional, en los experimentos también se observó que el uso de una LSSVM requiere ligeramente menor trabajo computacional que una SVM tradicional. Se espera que este trabajo de pie a mayor investigación en algoritmos de reconocimiento facial basados en características locales y SVM's.

Acknowlegment. Este trabajo se realizó con el apoyo del Consejo Nacional de Ciencia y Tecnología.

Referencias

1. Li, S. Z., Jain, A. K.: Handbook of Face Recognition. 2nd edn. Springer-Verlag, London (2011)
2. Zhao, W., Chellappa, R., Phillips, P. J., Rosenfeld, A.: Face Recognition: A literature survey. ACM Computing Surveys, 35(4), 399–459 (2003)
3. Gao, Y., Leung, M. K. H.: Face recognition using line edge map. IEEE Trasaction on Pattern Analysis and Machine Intelligence, 24(6), 764–779 (2016)
4. Heseltine, T., Pears, N., Austin, J.: Three-dimensional face recognition: an eigensurface approach. In: 2004 International Conference on Image Processing, pp. 1421–1424, IEEE, Singapore (2004)
5. Soltanpour, S., Boufama, B, Jonathan-Wu, Q. M.: A survey of local feature methods for 3D face recognition. Pattern Recognition, 72, 391–406 (2017)
6. Luo, Y., Wu, C. M., Zhang, Y.: Facial expression recognition based on fusion feature of PCA and LBP with SVM. Optik-International Journal for Light and Electron Optics, 124(17), 2767–2770 (2013)
7. Sarakon, P., Charoenpong, T., Charoensiriwath, S. Face shape classification from 3D human data by using SVM. In: The 7th 2014 Biomedical Engineering International Conference, pp. 1–5, IEEE (2014)
8. Lei, Y., Bennamoun, M., Hayat, M., Guo, Y.: An efficient 3D face recognition approach using local geometrical signatures. Pattern Recognition, 47(2), 509–524 (2014)
9. Zhang, B., Gao, Y., Zhao, S., Liu, J.: Local Derivative Pattern Versus Local Binary Pattern: Face Recognition with High-Order Local Pattern Descriptor. IEEE Transactions on Image Processing, 19(2) (2010)
10. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and MAchine Intelligence, 24(7), pp. 971–987 (2002)
11. Soltanpour, S., Jonatha-Wu, Q.M.: High-order local normal derivative pattern (LNDP) for 3D face recognition. In: 2017 IEEE International Conference on Image Processing, pp. 2811–2815, IEEE, Beijing, China (2017)

12. Klasing, K., Althoff, D., Wollherr, D., Buss, M.: Comparison of Surface Normal Estimation Methods for Range Sensing Applications. In: IEEE International Conference on Robotics and Automation, pp. 3206–3211, Kobe, Japan (2009)
13. Vapnik, V.: Estimation of dependencies based on empirical data. Addendunm 1 (1982)
14. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning, 20, 273–297 (1995)
15. Suykens, J. A., Vandewalle, J.: Least squares support vector machine classifiers. Neural processing letters, 9(3), 293–300 (1999)
16. Gupta, S., Castleman, K. R., Markey, M. K., Bovik, A. C.: Texas 3D face recognition database. In: IEEE Southwest symposium on image analysis & interpretation, pp. 97–100 (2010)
17. Gupta, S., Markey, M. K., Bovik, A. C.: Anthropometric 3d face recognition. International journal of computer vision, 90(3), 331–349 (2010)
18. Brabanter, K., Karsmakers, P., Ojeda, F., Alzate, C., Brabanter, J., Pelkams, K., Moor, B., Vandewalle, J., Suykens, J. A. K.: LSSVMLab. URL: https://www.esat.kuleuven.be/sista/lssvmlab/ (2011)
19. Fawcett, T.: An introduction to ROC analysis. Pattern recognition letters, 27(8), 861–874 (2006)
20. Shi, L., Wang, X., Shen, Y.: Research on 3D Face Recognition Method Based on LBP and SVM. Optik, 165157 (2020)

ISSN 1870-4069

Clasificación de A/V en imágenes de retina usando arquitectura inspirada en U-Net con ConvLSTM bidireccionales

M.G. Hernández-Ramírez, Saul Tovar-Arriaga,
Jesús Carlos Pedraza-Ortega, Efrén Gorrostieta-Hurtado

Universidad Autónoma de Querétaro,
Facultad de ingeniería,
México

{mghernandez1705, efren.gorrostieta}@gmail.com,
saul.tovar@uaq.mx, caryoko@yahoo.com,

Resumen. La retina es la única forma no invasiva de tener acceso al sistema cardiovascular, permite estudiar cambios en las estructuras que se encuentran es esta zona, como microaneurismas, hemorragias y los cambios que presentan los vasos retinianos, todos estos cambios se pueden utilizar para diagnosticar una amplia gama de enfermedades. Una manera accesible de estudiar estos cambios es por medio de imágenes de fondo de ojo. El objetivo de este trabajo es clasificar los vasos retinianos en arteria y vena. Actualmente las redes basadas en aprendizaje profundo han logrado los mejores resultados en la segmentación y clasificación de imágenes médicas. Una de las redes más usadas por sus buenos resultados para esta tarea es U-Net. En este trabajo, proponemos una modificación de U-Net con bloques ConvLSTM bidireccionales para la clasificación de A/V en imágenes de fondo de ojo. En lugar de una concatenación en los bloques de U-Net, empleamos ConvLSTM bidireccionales para combinar los mapas de características extraídos, además de solo usar dos bloques de la U-Net, esto hace el entrenamiento menos tardado. El modelo propuesto se evalúa en la base de datos DRIVE logrando un rendimiento competente con los métodos existentes.

Palabras clave: Clasificación de vasos retinianos, redes neuronales convolucionales, U-Net, ConvLSTM.

A/V Classification in Retina Images Using U-Net Inspired Architecture with Bidirectional ConvLSTM

Abstract. The retina is the only non-invasive way to the cardiovascular system, it permits to study changes in the structures found in this area, such as microaneurysms, hemorrhages and changes in the retinal vessels. These alterations in the retina can be used to diagnose a wide range of

diseases. An accessible way to study these changes is through fundus images. This paper focuses on the classification of retinal vessels in arteries and veins. Networks based on deep learning have, in recent years, achieved the best results in segmentation and classification of medical images. One of the more commonly used networks, because of its high performance, is U-Net. In this work, a modified U-Net, with bidirectional ConvLSTM blocks is proposed for A / V classification in fundus images. Instead of concatenating across U-Net blocks, two-way ConvLSTM are used to combine the extracted feature maps. In addition, using only two U-Net blocks makes training less time consuming. The proposed model was evaluated using the DRIVE database, achieving competent performance with existing methods.

Keywords: Classification of retinal vessels, convolutional neural networks, U-Net, ConvLSTM.

1. Introducción

Algunas características del fondo del ojo proporcionan información sobre el estado de salud del ser humano, como problemas oculares y problemas relacionados a enfermedades crónicas [1], estas enfermedades provocan cambios en la estructura y función del ojo. Una de las enfermedades que está asociada a los cambios en el fondo de ojo es la hipertensión [8]. Uno de los cambios que provoca esta enfermedad es el estrechamiento de las arterias de menor tamaño, este cambio en la vasculatura puede evaluarse mediante las relaciones del diámetro de arteria y vena (AVR) [2], también se ven afectados los nervios retinianos, coroideo y óptico, estos experimentan una serie de cambios fisiopatológicos en respuesta a la presión arterial elevada, teniendo como resultado una serie de signos clínicos llamados retinopatía hipertensiva [3].

Una de las principales anomalías en la retinopatía hipertensiva es el estrechamiento de vasos retinianos y el cruce arteriovenoso patológico [4]. Estos cambios están relacionados no solo con los niveles actuales de presión arterial del paciente, sino también con los niveles medidos en el pasado, lo que sugiere que estos signos son marcadores persistentes de daño hipertensivo crónico [7]. El estrechamiento de vasos sanguíneos también se puede usar para predecir el desarrollo de la hipertensión en individuos normotensos. Por lo tanto, el estrechamiento podría ser un marcador temprano de hipertensión [9].

Algunas otras enfermedades también afectan la circulación sanguínea y como consecuencia pueden causar un AVR anormal como [5]. Así mismo, el AVR se puede utilizar para predecir el riesgo de muerte por enfermedad coronaria y cerebrovascular [10]. Estos cambios vasculares a menudo preceden al inicio de los síntomas y, si se detectan, pueden permitir tratamientos preventivos que pueden reducir el riesgo de pérdida de visión. Para abordar estos problemas se estudia la vasculatura de la retina mediante la segmentación, clasificación y medición de los vasos.

La retina es la única forma no invasiva para poder visualizar el sistema cardiovascular, es una ventana que permite estudiar los cambios de las carac-

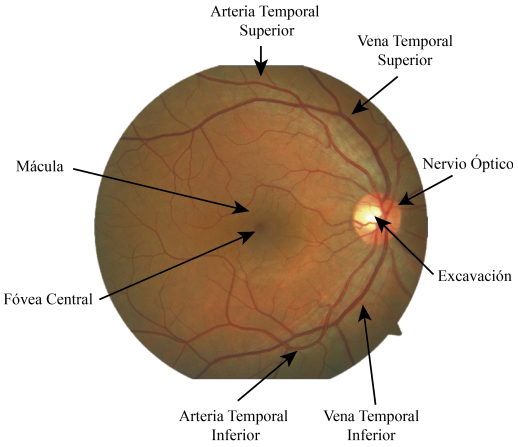


Fig. 1. Imagen sana de fondo de ojo, *Mexican Advanced Imaging Laboratory for Ocular Research* (MAILOR).

terísticas en esta estructura. Esto es posible utilizando una variedad de métodos de imágenes, que incluyen fotografía de fondo de ojo, angiografía con fluoresceína y tomografía de coherencia óptica. La fotografía de fondo de ojo es la manera preferida para obtener la información, tiene un menor costo que las anteriores y es fácil de usa. En la Fig. 1 se muestra una imagen de fondo de ojo en condición sana donde se pueden ver las partes más importantes. En la retina, todas las arterias y venas emanan del nervio óptico.

Por lo tanto, todas las arterias están conectadas entre sí, y también todas las venas, se identifican las venas por un color más obscuro al de las arterias porque transportan sangre desoxigenada al corazón [6]. Las venas normalmente tienen un calibre mayor al de las arterias, teniendo una relación AVR normal de 2:3. La clasificación de vena y arteria es una tarea realizada manualmente por especialista, es una tarea tardada y laboriosa. En una imagen de alta resolución, a menudo se pueden detectar más de cien vasos o segmentos de vasos, muchos de los cuales son ambiguos y requieren una cuidadosa visualización para clasificarlos. Este artículo se centra en la segmentación y clasificación automática de los vasos sanguíneos de imágenes de retina.

En los últimos años, el uso de redes neuronales ha mostrado avances significativos y resultados prometedores en la detección de patrones en imágenes, específicamente las redes neuronales convolucionales (CNN). Las CNN son particularmente apropiadas para la clasificación y segmentación de imágenes [11], [12]. En 2015 Ronneberger *et al.* prepusieron una red totalmente convolucional en forma de U [13] para la segmentación de imágenes médicas. La arquitectura U-Net propuesta ha demostrado una mejor capacidad para segmentar estructuras más finas, como la membrana celular, con muy pocas imágenes de entrenamiento, ha tenido éxito y se han propuesto diferentes variaciones.

En este estudio, se modifica la arquitectura de la U-Net y se propone una U-Net con 2 bloques y conectada con ConvLSTM bidireccionales inspirado por [21].

2. Antecedentes

Los métodos de clasificación de los vasos de la retina: arteria y vena, encontrados en la literatura se pueden clasificar en dos clases, en métodos basados en árboles y en características y los más populares en los últimos años, basados en aprendizaje profundo. En la Tabla 1 se muestra un resumen de algunos de los métodos usados en los últimos años.

Tabla 1. Descripción general de trabajos para la clasificación de vena y arteria de imágenes de retina.

Autor	Método	Métrica
Estrada *et al.* [14] (2015)	Estimación de topología, teoría de grafos	Exactitud = 0.935
Xu *et al.* [15] (2017)	Clasificación de píxeles, clasificador kNN.	Exactitud = 0.923
Welikala *et al.* [18] (2017)	CNN	Exactitud = 0.919
Xu *et al.* [16] (2018)	CNN U-Net	Exactitud = 0.832
Girard *et al.* [17] (2019)	CNN adaptación U-Net	Exactitud = 0.924

En la primera clase encontramos trabajos como el de Estrada *et al.* [14] donde utilizó un enfoque de estimación de topología basado en grafos, que estima la topología más probable para las etiquetas de arteria y de vena. Buscando posibles árboles etiquetados, usando algoritmos de búsqueda heurísticos.

El algoritmo fue probado con 4 bases de datos: WIDE, AV-INSPIRE, AV-DRIVE, CT-DRIVE. Otro trabajo es el propuesto por Xu *et al.* [15] donde propone una técnica de clasificación arteria y vena donde la imagen de retina es regularizada y normalizada para extraer múltiples características de los vasos sanguíneos. Se extraen diferentes características basadas en el color y textura para cada píxel y se clasifican en arteria y vena usando un clasificador kNN. Esta técnica fue probada en la base de datos DRIVE y logró una exactitud del 0.923.

Actualmente y gracias al poder de computo con el que hoy contamos y las bases de datos disponibles, los métodos más recurridos están basados en aprendizaje profundo. Un beneficio de estos modelos es su capacidad para realizar la extracción automática de características de los datos sin procesar, también llamada aprendizaje de características.

Un método de esta clasificación es el propuesto por Welikala *et al.* [18] donde propuso un método basado en el aprendizaje profundo para la clasificación de arterias y venas en toda la imagen de la retina.

La red propuesta esta formada por 6 capas, de las cuales 3 son capas convolucionales y 3 son capas completamente conectadas, la capa ReLU siguió a cada capa convolucional y las dos primeras capas completamente conectadas. Este método se evaluó en el conjunto de datos del Biobanco del Reino Unido y en la base de datos publica DRIVE.

Xu *et al.* [16] propuso una red neuronal convolucional inspirada en la arquitectura U-Net para la segmentación de venas y arterias en imágenes a color de la retina. La red fue entrenada y probada en el conjunto de datos publica DRIVE. La sensibilidad, especificidad y exactitud es de 0.870, 0.980, 0.832 respectivamente. Las tasas de clasificación errónea registradas fueron de 0.098 para venas y 0.237 para arterias. Girard *et al.* [17] Propone un método de clasificación A-V usando una CNN simple que clasificaba los píxeles en arteria y vena, la red está compuesta de cinco capas convolucionales de codificación y cinco capas convolucionales de decodificación junto con conexiones de derivación. La arquitectura propuesta es una derivación de la U-Net. Realiza un preprocesamiento a la imagen donde corrigen problemas de iluminación no uniforme, obteniendo 3 canales diferentes a los originales (RGB), concatenan los 3 canales nuevos a los 3 originales, teniendo una imagen de 6 canales. El método se probó en dos bases de datos DRIVE y MESSIDOR.

3. Metodología

En esta sección, se describe a detalle el conjunto de datos usados, el preprocesamineto realizado a las imágenes y la arquitectura usada.

Para la implementación de este trabajo se utilizó la base de datos DRIVE, la cual es una base de datos gratuita y la más usada para trabajos de segmentación y clasificación de vasos sanguíneos de retina, así como para la búsqueda y detección de diferentes patologías que pueden identificarse en esta zona.

3.1. Preprocesamiento

Las imágenes de fondo de ojo pueden tener una amplia variación en la pigmentación, debido a problemas con la iluminación no uniforme y a la mala exposición al momento de tomar la fotografía, lo que puede influir el rendimiento de la CNN. Realizar un preprocesamiento mejora la calidad de las imágenes y permitir que la CNN detecte con mayor facilidad las características buscadas.

El primer paso es la normalización de datos, la normalización es usada para escalar un atributo dentro de un rango especifico, facilitar y acelerar el entrenamiento de las redes. Las CNN no funcionan bien cuando los atributos numéricos de entrada tienen escalas muy diferentes. En este trabajo se utilizan dos técnicas de normalización, z-score y min-max.

La normalización z-score es útil cuando se tienen valores atípicos en un atributo:

$$x' = \frac{x - \mu_A}{\sigma_A}, \tag{1}$$

donde μ_A y σ_A son la media y desviación estándar.

En la normalización min-max se hace una transformada lineal de los datos, los valores se desplazan y reescalan para que terminen en un rango de 0 a 1:

$$x' = \frac{x - min(A)}{max(A) - min(A)}, \tag{2}$$

donde x' es el valor normalizado del valor original x relacionado con los valores mínimos y máximos de un atributo A.

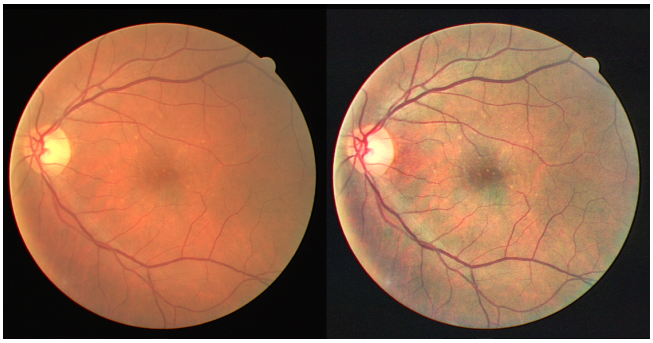


Fig. 2. Resultado de preprocesamineto de imagen. A la izquierda imagen original; a la derecha imagen preprocesada.

El segundo paso es la corrección de contraste y brillo. Para el primero se aplica una ecualización de histograma adaptativo limitada por contraste (CLAHE). Para mejorar de brillo se utiliza una corrección gamma. El resultado de este paso se puede ver en la Fig. 2.

3.2. Aumento de datos

Las bases de datos para la segmentación y clasificación de vasos retinianos suelen no ser muy grandes debido a su naturaleza, para abordar este problema se hacer un aumento de datos para mejorar el entrenamiento de la red neuronal.

Se implementa un aumento de datos mediante la extracción de parches aleatorios de cada imagen, haciendo giros vertical y horizontal aleatoriamente en cada parche. Con la intensión de aumentar artificialmente los datos y agregar más variabilidad al conjunto de datos de entrenamiento. En la Fig. 3 podemos ver algunos parches extraídos con sus respectivas etiquetas.

3.3. Arquitectura

En este trabajo se usan bloques ConvLSTM bidireccionales y parte reducida de la estructura U-net como se muestra en la Fig. 4. La estructura original, de

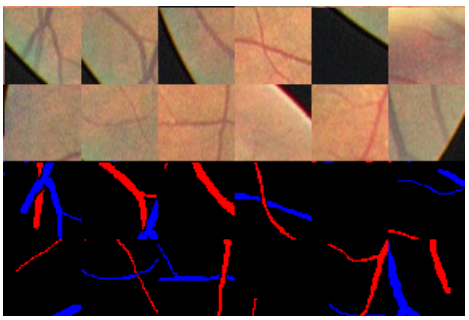


Fig. 3. Ejemplo de extracción de parches. Dos filas superiores imágenes; dos filas inferiores etiquetas.

U-net está compuesta por una parte de codificación, donde se captura tanto la información semántica como la contextual a medida que se duplica el número de filtros. La segunda parte está compuesta por decodificación, donde se restaura la información espacial y se fusionan las características obtenidas en los niveles más bajos.

La estructura propuesta consta de 2 niveles, en los que se aplican una doble convolución con un filtro de 3x3, cada una seguida de una unidad lineal rectificada (ReLU), una operación de agrupación máxima de 2x2 con paso de 2 para disminuir la resolución. La imagen de entrada es de 3 canales un tamaño de 64x64. En el primer nivel el número de canales utilizados fue de 64 y el tamaño de la imagen 32x32 tras dobles convoluciones. En el segundo nivel el número de canales utilizados fue de 128 y el tamaño de la imagen 16x16. La parte de decodificación sigue la estructura anterior, el número de canales fue el mismo, pero en sentido contrario. También se añaden las características que se encuentran en cada uno de los diferentes niveles, concatenando la salida del nivel de contracción con la entrada de expansión.

Los mapas de características correspondientes a la capa de codificación tienen una resolución más alta, mientras que los mapas de característica correspondientes a la decodificación contienen más información semántica. En lugar de una simple concatenación en esta propuesta se hace la combinación de estos dos mapas de características con ConvLSTM bidirecional inspirado en [21]. Después de cada capa de convolución se aplica una capa de normalización por bloque y la función de activación ReLU.

4. Materiales y experimentación

Para la presente investigación de segmentación de vena y arteria de imágenes de fondo de ojo, se trabaja con la base de datos publicada por Qureshi *et al.* [19] donde un oftalmólogo capacitado etiquetó manualmente los vasos del conjunto de pruebas como arteria, vena o superposición.

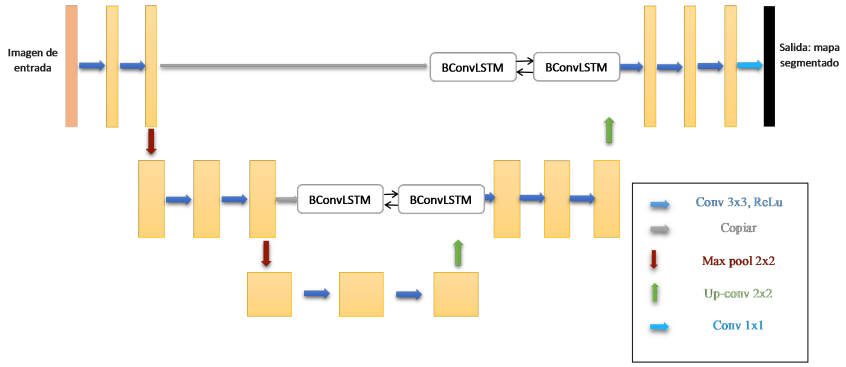


Fig. 4. Arquitectura propuesta.

Para esta clasificación se usó la base de datos publica DRIVE [20] que incluye un conjunto de cuarenta fotografías a color del fondo de ojo, obtenidas de un programa de detección de retinopatía diabética. En este conjunto de datos, siete de las cuarenta imágenes contienen patologías anormales como exudados, hemorragias y epitelio pigmentario. Las imágenes tienen un tamaño de 565x584 píxeles, se adquirieron utilizando una cámara Canon CR5 no midriática 3-CCD con un campo de visión (FOV) de 45 grados. La base de datos DRIVE se divide en dos conjuntos, el conjunto de entrenamiento y el conjunto de prueba, cada uno con veinte imágenes. El conjunto de prueba esta segmentado manualmente por dos observadores, el primer observador es aceptado como verdad fundamental.

Se utilizó optimización ADAM, con una tasa de aprendizaje de 0.001. En total, se realizaron 100 épocas, con tamaño de lote de 16, este es un hiperparámetro importante, es el número de muestras que se propagarán a través de la red. La métrica que se evalúa para decidir si la CNN está o no mejorando, es la función de perdida.

Para la validación se usó validación cruzada, con 5 particiones. De las 40 imágenes de la base de datos, 24 se usaron para entrenar, 6 para validar y 10 para prueba, estos resultados se muestran en la Tabla 2.

También se hicieron pruebas con la configuración predeterminada de la base de datos DRIVE, donde se usa 20 imágenes para entrenamiento, las cuales se dividen 16 para entrenar, 4 para validar y las 20 restantes de prueba.

Toda la implementación de este trabajo se desarrolló en TensorFlow y Keras, ambas librerías se especializan en aprendizaje profundo, y se usa Python como lenguaje de programación.

El equipo utilizado en las pruebas cuenta con un procesador Intel (R) Core (TM) i5-8400 CPU@2.80 GHz, con 16 GB de RAM y una tarjeta gráfica NVIDIA GeForce GTX 1070 con 8 GB de VRAM.

5. Resultados

Las métricas usadas para la evaluación del desempeño del modelo son sensibilidad, especificidad y exactitud. El cálculo de estas métricas se basa en cuatro posibles interpretaciones de los datos:

- Verdadero positivo (TP): la etiqueta de la muestra es positiva y se clasifica como tal.
- Verdadero negativo (TN): la etiqueta de la muestra es negativa y se clasifica como tal.
- Falso positivo (FP): la etiqueta de la muestra es negativa, pero se clasifica como positiva.
- Falso negativo (FN): la etiqueta de la muestra es positiva, pero se clasifica como negativa.

En función de estos valores, se pueden calcular las métricas mencionadas anteriormente de acuerdo a las ecuaciones 3, 4, 5:

- Sensibilidad: También llamada tasa de Verdaderos Positivos. Es la proporción de casos positivos que fueron correctamente identificados:

$$Sensibilidad = \frac{TP}{TP + FN}. \tag{3}$$

- Especificidad: También llamada Tasa de Verdaderos Negativos. Casos negativos que se clasificaron correctamente:

$$Especificidad = \frac{TN}{TN + FP}. \tag{4}$$

- Exactitud: Mide cuántas observaciones, tanto positivas como negativas, se clasificaron correctamente.Es la proporción de instancias negativas que se clasifican incorrectamente como positivas:

$$Exactitud = \frac{TP + TN}{TP + FP + FN + TN}. \tag{5}$$

En la Tabla 2 podemos ver una comparación de trabajos similares basados en el conjunto de datos DRIVE, donde se observa sensibilidad, especificidad y exactitud, métricas descritas anteriormente.

En la Fig. 5 se puede ver un ejemplo de un resultado obtenido del conjunto de datos DRIVE. Los falsos negativos son especialmente vasos retinianos pequeños que no se alcanzan a detectar.

En los vasos mayores a 5 píxeles, los falsos positivos se ubican en los bordes de los vasos. Existe una confusión entre artería y vena o viceversa en los vasos mas pequeños o en los puntos de intersección entre estos dos.

En la Fig. 6 se proporcionan las curvas *Receiver Operating Characteristic* (ROC) y *Precision-Recall* (PR) de la arquitectura propuesta para los resultados de rendimiento de la segmentación arteria y vena. La curva ROC muestra el

Tabla 2. Comparación de resultados de clasificación Vena/Arteria con estado del arte en base de datos DRIVE.

Método	Sensibilidad	Especificidad	Exactitud
Estrada *et al.* [14] (2015)	0.9170	0.9170	0.9100
Xu *et al.* [15] (2017)	-	-	0.9230
Welikala *et al.* [18] (2017)	0.7810	-	0.9190
Xu *et al.* [16] (2018)	0.8700	0.9800	0.8320
Girard *et al.* [17] (2019)	0.9370	0.9290	0.9493
Propuesto	0.8000	0.9780	0.9550

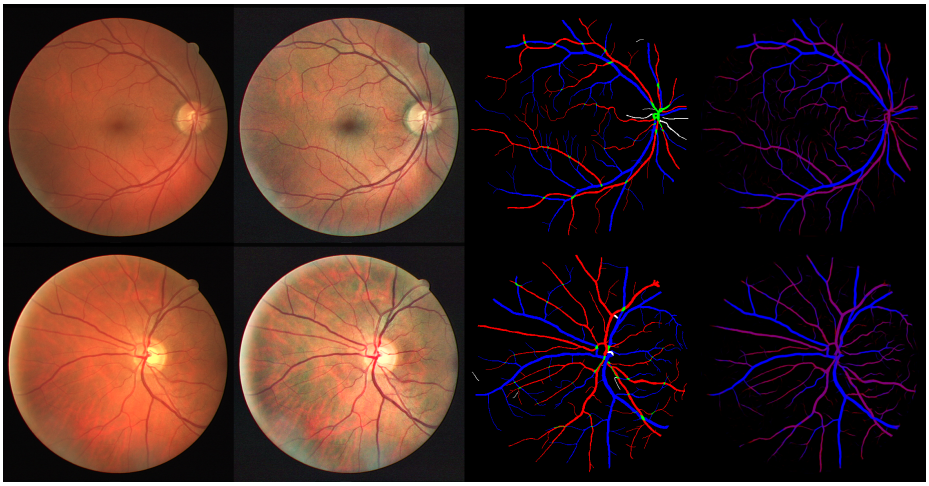


Fig. 5. Resultado de clasificación de arteria y vena de dos imágenes. Iniciando con la primera a la izquierda, imagen original; segunda imagen, preprocesamiento; tercera imagen verdad fundamental; ultima imagen salida clasificada.

rendimiento de un modelo de clasificación, el área bajo la curva es de 0.9722 para este modelo. La curva traza la sensibilidad contra especificidad, parámetros de tasa de verdaderos positivos y tasa de falsos positivos. La curva *Precision-Recall* tiene un área bajo la curva de 0.9003, esta gráfica permite ver a partir de qué *recall* se tiene una degradación de la precisión y viceversa, esta curva se centra principalmente en el rendimiento de la clase positiva.

6. Conclusiones

El estado patológico del fondo de ojo también puede afectar la clasificación de arteria y vena, especifícamele cuando existe una lesión obstruyendo los vasos, cuando las lesiones son pequeñas no se tiene problema y se clasifica correctamente pero cuando las lesiones son grandes y en un color contrastante al resto de la retina los vasos quedan ausentes o clasificados erróneamente. El preprocesamiento de las imágenes de entrada es de suma importancia ya que corrige algunos

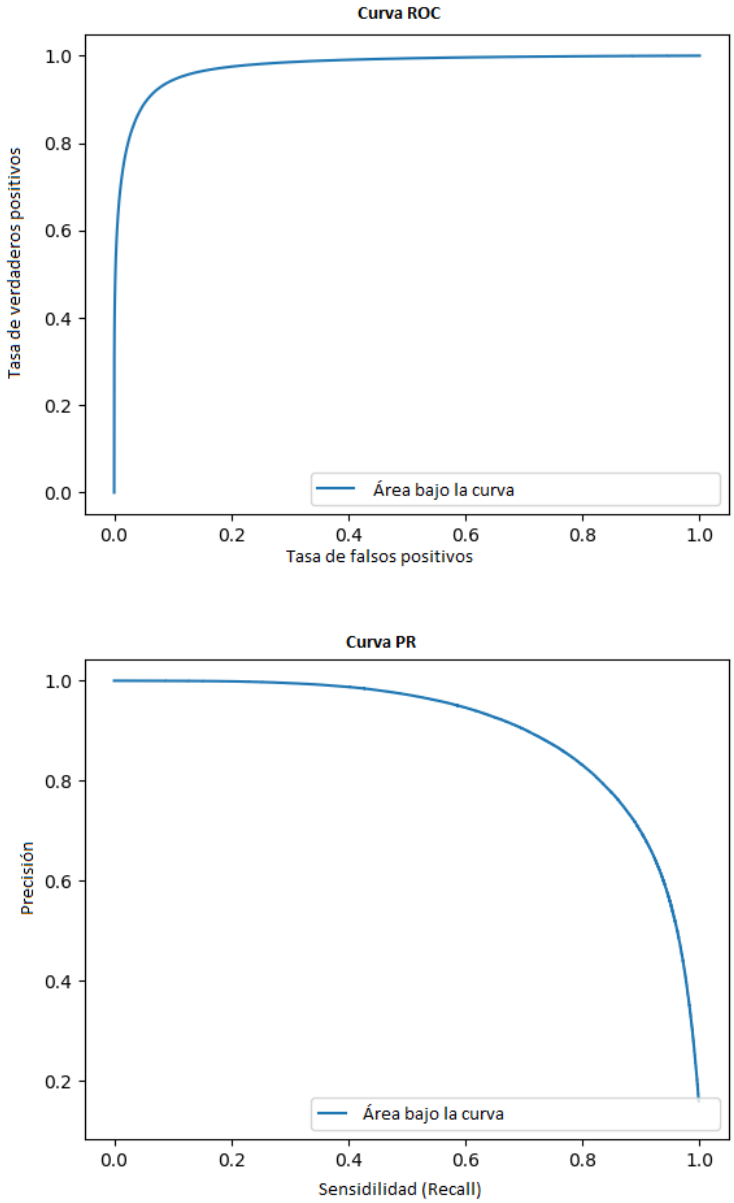


Fig. 6. Curvas de rendimiento de la arquitectura propuesta en la segmentación A/V en DRIVE.

errores mencionados anteriormente, resaltando las características deseadas en la imagen.

Acompañado del preprocesamiento, un aumento de datos es importante en este trabajo, la base de datos DRIVE cuenta con pocas imágenes para el entrenamiento de una CNN, lo cual implementar aumento de datos por extracción de parches y girarlos aleatoriamente ayuda a un mejor entrenamiento.

Tener una cantidad mayor de datos etiquetados mejoraría la clasificación, el modelo aprendería más información de la estructura provocando que el modelo sea robusto para diferentes patologías severas que compliquen la clasificación de los vasos.

El trabajo propuesto acelera el etiquetado manual de arterias y venas, proporcionando un etiquetado automático que requiere correcciones mínimas manualmente por parte de un experto. Los resultados en este estudio son muy similares o superiores a los resultados de los métodos de mayor rendimiento, se simplificó la complejidad de métodos similares con resultados parecidos. Al usar bloques ConvLSTM bidireccionales ayuda a combinar los mapas de características extraídos hace la arquitectura más susceptible.

El trabajo futuro se centrará en encontrar nuevos biomarcadores y mediciones clínicas utilizando la clasificación de arteria y vena para detectar los cambios en los vasos y con ellos poder detectar signos en temprana edad en diversas patologías.

Agradecimiento. Este trabajo ha sido apoyado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) Agradecemos a la Dra. Renata García Franco por el apoyo dado para la realización de esta investigación como al *Mexican Advanced Imaging Laboratory for Ocular Research* (MAILOR).

Referencias

1. Kanski, J. J., Bowling, B.: Clinical ophthalmology: a systematic approach. Elsevier Health Sciences, Location (2011)
2. Cheung, C. Y. L., Ikram, M. K., Sabanayagam, C., Wong, T. Y.: Retinal microvasculature as a model to study the manifestations of hypertension. Hypertension, 60(5), pp. 1094–1103 (2012)
3. Wong, T. Y., Ming, C., Cheung, G., Larsen, M., Sharma, S.: Diabetic retinopathy. Medicine, 77–85 (2016)
4. Sharma, K., Kanaujia, V., Mishra, P., Agarwal, R., and Tripathi, A. :. Hypertensive retinopathy. Clinical queries, nephrology, 2(3), pp. 136–139 (2013)
5. Kondermann, C., Kondermann, D., and Yan, M.: Blood vessel classification into arteries and veins in retinal images. In: Medical Imaging 2007: Image Processing, Vol. 6512, p. 651247 (2007)
6. Yu, H., Barriga, S., Agurto, C., Nemeth, S., Bauman, W., Soliz, P.: Automated retinal vessel type classification in color fundus images. In: Medical Imaging 2013: Computer-Aided Diagnosis, International Society for Optics and Photonics, Vol. 8670, p. 86700P (2013)
7. Sharrett, A. R., Hubbard, L. D., Cooper, L. S., Sorlie, P. D., Brothers, R. J., Nieto, F. J., ... y Klein, R.: Retinal arteriolar diameters and elevated blood pressure: the Atherosclerosis Risk in Communities Study. American journal of epidemiology, 150(3), pp. 263–270 (1999)

8. Wong, T. Y., Klein, R., Klein, B. E., Tielsch, J. M., Hubbard, L., Nieto, F. J.:Retinal microvascular abnormalities and their relationship with hypertension, cardiovascular disease, and mortality. Survey of ophthalmology, 46(1), pp. 59–80 (2001)
9. Leung, H., Wang, J. J., Rochtchina, E., Wong, T. Y., Klein, R., Mitchell, P.: Impact of current and past blood pressure on retinal arteriolar diameter in an older population. Journal of hypertension, 22(8), pp. 1543–1549 (2004)
10. Wang, J. J., Liew, G., Klein, R., Rochtchina, E., Knudtson, M. D., Klein, B. E., ... and Mitchell, P.: Retinal vessel diameter and cardiovascular mortality, pooled data analysis from two older populations. European heart journal, 28(16), pp. 1984–1992 (2007)
11. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
12. Badar, M., Haris, M., Fatima, A.: Application of deep learning for retinal image analysis: A review. Computer Science Review, 35, p. 1002035 (2020)
13. Ronneberger, O., Fischer, P., Brox, T.: U-net, Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer assisted intervention. Springer, Cham, pp. 234–241 (2015)
14. Estrada, R., Allingham, M. J., Mettu, P. S., Cousins, S. W., Tomasi, C., Farsiu, S.: Retinal artery-vein classification via topology estimation. IEEE transactions on medical imaging, 34(12), pp. 2518–2534 (2015)
15. Xu, X., Ding, W., Abràmoff, M. D., Cao, R.: An improved arteriovenous classification method for the early diagnostics of various diseases in retinal image. Computer methods and programs in biomedicine, 141, pp. 3–9 (2017)
16. Xu X., Tan T., Xu F.: An improved U-net architecture for simultaneous arteriole and venule segmentation in fundus image. Medical Image Understanding and Analysis, MIUA 2018, Communications in Computer and Information Science, Springer, vol. 894, pp. 333–340 (2018)
17. Girard, F., Kavalec, C., Cheriet, F.: Joint segmentation and classification of retinal arteries/veins from fundus images. Artificial intelligence in medicine, 94, pp. 96–109 (2019)
18. Welikala, R. A., Foster, P. J., Whincup, P. H., Rudnicka, A. R., Owen, C. G., Strachan, D. P., Barman, S. A.: Automated arteriole and venule classification using deep learning for retinal images from the UK Biobank cohort. Computers in biology and medicine, 90, pp. 23–32 (2017)
19. Qureshi, T. A., Habib, M., Hunter, A., Al-Diri, B.: A manually-labeled, artery/vein classified benchmark for the DRIVE dataset. In: Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, pp. 485–488 (2013)
20. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken: Ridge-based vessel segmentation in color images of the retina. IEEE Trans Med Imaging, (vol. 23, no. 4, pp. 501–509.) (2004)
21. Azad, R., Asadi-Aghbolaghi, M., Fathy, M., Escalera, S.: Bi-Directional ConvLSTM U-Net with Densley Connected Convolutions. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2019)

ISSN 1870-4069

Representación de información de una institución educativa a través de XML

Juan Manuel Cancino-Gordillo, Luis Yael Méndez-Sánchez, Francisco Valdez-Tovar, Mireya Tovar-Vidal

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación, Puebla,
México

{juanitotito.juan, luisyael1995, fravato}@gmail.com,
mtovar@cs.buap.mx

Resumen. El siguiente artículo presenta una aportación a la problemática que es el estructurar la información de forma adecuada, en este caso se están obteniendo datos de la Facultad de Ciencias de la Computación en relación con alumnos, profesores y materias de la carga académica del área de Posgrado se construye y se analiza un diagrama de árbol, propio para el diseño de archivos XML. En este caso se ha optado por manejar este tipo de archivos, ya que, su estructura es de tipo jerárquica, además de que se busca estructurar la información de tal manera que sea procesable por computadoras de forma automática. Con ayuda del complemento XPath propio para trabajar este tipo de archivos, se pretende acceder a la estructura del mismo, iniciando en un nodo raíz, seguido de los hijos y finalmente a las hojas del árbol, la principal función del archivo XML es realizar lo mismo que una base de datos tradicional, almacenamiento y consulta de los datos obtenidos previamente. Es por ello que se obtiene un sistema de software encargado del manejo de la información mediante archivos XML, con el cual se podrán llevar a cabo funcionalidades óptimas para la inserción, eliminación y modificación de los datos con los cuales se estén trabajando.

Keywords: Web semántica, XML, XPath, estructuración de información, recuperación de datos.

Representation of Information of an Educational Institution through XML

Abstract. The following article presents a contribution to the problem of structuring information appropriately, in this case data is being obtained from the Faculty of Computer Science in relation to students, teachers and subjects of the academic load of the area Postgraduate course, a tree diagram is constructed and analyzed, suitable for the design of XML files. In this case, we have made the decision to use this kind of files, since their structure is hierarchical, and we are able to create an information structure in such a way that it is automatically processible by computers. With the help of its own XPath plug-in to work with this type of files, it is intended to access its structure, starting at a root node,

followed by the children and finally the leaves of the tree, the main function of the XML file is to perform the same as a traditional data base, storage and consultation of previously obtained data. In this way, using XML files, we get a software system in charge of information management, with which optimal functionalities can be carried out for the insertion, deletion, and modification of any data they are working on.

Keywords: Semantic web, XML, XPath, information structuring, data retrieval.

1. Introducción

Hoy en la actualidad el tema de organizar la información resulta muy importante, ya que, de esta manera se tiene un mejor control de los datos que se están utilizando, además de que al mismo tiempo facilitan la comprensión del tema que se esté abordando.

El Consorcio de la World Wide Web, el cual se encarga de generar estándares para asegurar el crecimiento de la misma, es el encargado de desarrollar las recomendaciones y estructuras para la web semántica. El objetivo principal de dicha web es la publicación de datos en un formato legible e intercambiable por cualquier aplicación de consulta y procesamiento en línea, fundamentándose en agregar metadatos semánticos y ontologías para representar la información. Es imperante el establecer formalmente las relaciones y significados de los datos para que cualquier sistema pueda interpretarlos.

Así mismo, esta tecnología aporta un formato o estructura a los contenidos importantes dentro de los sitios de Internet, generando así un ambiente de recuperación en que los algoritmos buscadores se desplazan de sitio en sitio y pueden realizar procesos complejos para el usuario final. No es un componente externo, es una extensión del sitio al cual pertenece, en el que los datos tienen un objetivo explícito mejorando el trabajo colaborativo entre máquinas y personas.

En un tiempo muy cercano, estas estructuras darán la pauta para un nuevo procesamiento y entendimiento de lado de las computadoras para no solamente obtener los datos, sino llevando a cabo un entendimiento y comprensión de información más allá de lo que actualmente conocemos. Es de suma importancia para la implementación y funcionamiento de la web semántica, que las computadoras y algoritmos obtengan un formato de información con una estructura y reglas que permitan la automatización.

Por otro lado, hablando de la recuperación de datos, uno de los lenguajes más utilizados en la programación de datos del lado del servidor es precisamente XML, ya que aplica correctamente los estándares establecidos por la W3C y en base a esto, los profesionales en desarrollo web pueden establecer sus propias estructuras de etiquetas facilitando su búsqueda e interpretación a través de los algoritmos de búsqueda en Internet.

Otra tecnología muy común para la recuperación de información en los archivos XML es Xpath, la cual mediante una sintaxis establece la organización del documento, permitiendo su navegación y manipulación con tipos de datos comunes en lenguajes de programación además de que para acceder a los elementos del documento XML utiliza una notación parecida al acceso de archivos en los sistemas operativos [1].

Junto con ello, cabe mencionar que el uso de XML junto con Xpath nos brinda una gran funcionalidad en cuanto al ámbito del manejo de información, que es lo que principalmente se busca con el desarrollo de este trabajo, ya que, se tiene el conjunto de datos en un archivo XML, el cual utilizaremos para la estructuración y recuperación de la información que se requiera.

En este trabajo se están presentando alternativas no comunes para el manejo de la información, así como la inserción de datos o bien consultas sobre los mismos. Por esto, el uso de herramientas de la web semántica como lo es XML y XMLpath, nos resultan muy factibles para el desarrollo de este sistema, ya que, son una alternativa inicial para estructurar la información y a su vez lograr la recuperación de los datos que se requieran; con esto podemos mencionar que se podría lograr una aportación importante en este ámbito que es el manejo y recuperación de información.

Para el estado del arte actual, se han encontrado algunos artículos similares y relacionados con la problemática planteada y propuestas de solución. Bravo et al. [10] describen un modelo para representar los datos académicos e institucionales en un centro de investigación en la CDMX, el cual responde preguntas referentes a personas y los datos relevantes a las mismas dentro de su área de interacción; por otro lado Bravo Contreras et al. [11] presentan una metodología para el diseño y la construcción de ontologías incorporando los principios de diseño más destacados así como un proceso exhaustivo de evaluación y por último, Flores et al.[12] presentan el diseño de un modelo ontológico para búsqueda de información dentro de una institución de educación a nivel superior y busca dar respuesta a preguntas sobre los procedimientos de inscripción de los alumnos y su respectiva liberación de prácticas profesionales.

En el siguiente trabajo desarrollado, presentamos en la sección 2 el diseño de la propuesta, en donde se explicará a mayor detalle la metodología utilizada, así como también en la sección 3 se muestran algunos resultados en los que se da a conocer la funcionalidad de la solución propuesta anteriormente. Finalmente, las conclusiones y referencias son presentadas.

2. Diseño de la propuesta

El uso de las tecnologías de web semántica, en este caso como lo es XML, resulta muy importante en la propuesta que se presentará, ya que, mediante estas herramientas se busca lograr un mejor control de los datos que se estén utilizando. Junto con ello, sabemos que al tener bien organizada la información, resultará de una manera más sencilla obtener aquellos datos que se requieran mediante consultas específicas. Esto aplica tanto a datos físicos como digitales, mientras más organizada esté la información más fácil será recuperarla.

Para solventar la problemática anterior se utiliza un árbol para la organización de la información, es decir, con ayuda de esta técnica se puede tener un mejor control de la información permitiéndonos representar una jerarquía real, que se puede representar con archivos de Lenguaje de Marcado Extensible (XML). Ya que el XML es una alternativa eficaz al HTML para la adaptación a nuevos entornos, ya que busca disociar los datos de su representación [4].

El diagrama de árbol es una representación del archivo XML el cual nos permite relacionar los conceptos y sus elementos definidos, ayudando a la solución del

problema por medio de una herramienta visual que muestra el panorama completo de las relaciones.

La forma de implementación de un árbol es partir de un nodo principal o nodo raíz [4], posteriormente se añaden nodos relacionados y así sucesivamente, mientras más se profundice en las ramas, mayor detalle y precisión se obtiene al momento de alcanzar el nodo principal partiendo de cualquiera de las ramas.

Características de los diagramas de árbol:

- Se visualiza claramente las relaciones entre la generalidad y los detalles que conlleva.
- Se encuentra el nodo principal o raíz a partir del cual se plantea la solución.
- Al dar un panorama completo y gráfico de la solución, provee la capacidad de detectar nodos o ramas faltantes.
- Permite revisar detalladamente el proceso en su totalidad.
- Las relaciones se muestran de forma gráfica.
- Utiliza a menudo otras técnicas y herramientas para complementarse o inclusive, crearse desde el principio con mayor precisión.

Dicho lo anterior proponemos el árbol presentado en la Fig. 1, para llevar a cabo la organización de una Facultad, en este caso la Facultad de ciencias de la computación de la BUAP, usando un árbol no binario, no ordenado y no balanceado, podemos proponer una estructura jerárquica sobre los datos que se representan con los nodos padres e hijos [5].

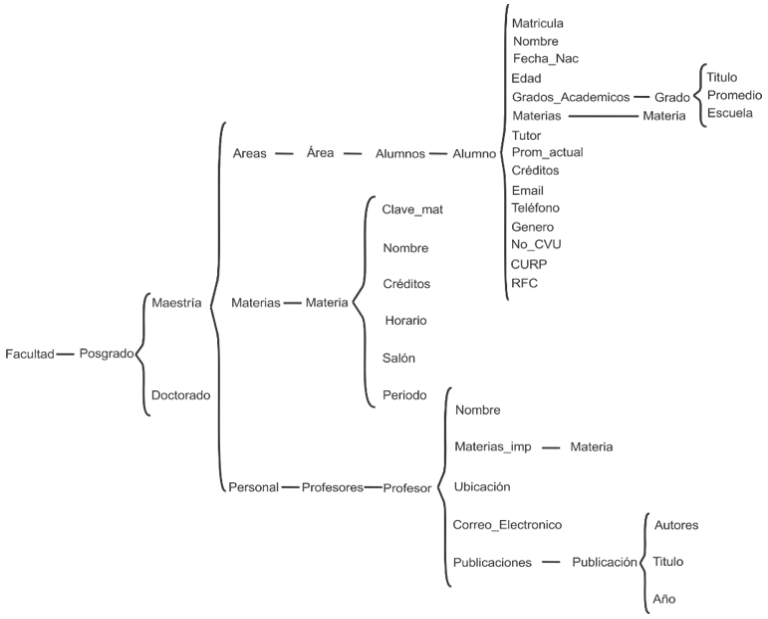


Fig. 1. Diagrama de árbol del archivo XML.

Como se hace mención no es un árbol binario o balanceado y se puede apreciar en la forma que se presentan los hijos de cada nodo. Sin embargo, esto no presenta problemas al explorar o descubrir información del árbol, ya que las búsquedas requieren de un nodo padre también conocido como *axis* [6] el cual en este caso es una ruta para acceder a la información. Como se ha mencionado anteriormente para acceder a un dato se asemeja a la escritura de directorios de archivos de una computadora, que puede ser interpretado por un humano o por una máquina.

La Fig. 1 consta como nodo central la facultad, por ejemplo, la Facultad de Ciencias de la Computación, despliega dos nodos hijos, Maestría y Doctorado, como tal sólo se está trabajando el área de posgrado. Por consiguiente, los siguientes nodos hijos serán los tres tipos de información que estamos tratando: alumnos, profesores y materias impartidas. En el caso de los alumnos se cuentan con los datos personales, siendo estos mismos los principales para brindar información acerca del alumno, cuenta con una sección de materias que están cursando, la cual está relacionada al nodo hijo llamado Materias, en el cual se está presentando la información de cada una que se ha registrado. El siguiente tipo de información es profesores, de la misma manera como el anterior, cuenta con algunos datos personales, esta vez añadiendo la ubicación donde se encuentra su cubículo, a su vez las materias que imparte tal profesor están relacionadas con la siguiente sección por medio de la clave de la materia, estos atributos están definidos en el XML correspondiente. Finalmente, la sección de materias contará con datos relevantes de estas mismas, así como su clave la cual fungirá como llave primaria para relacionarse con alumnos y profesores.

3. Resultados

El software cuenta con las cuatro funciones básicas de un sistema de gestión, que son: altas, bajas, modificación y búsqueda de registros. Las cuales funcionan con una combinación de las tecnologías de JavaScript [7] y PHP [8]. Para lograr el cometido de manipular la información y registros del sistema.

La primera funcionalidad es la búsqueda de registros, la cual está implementada completamente en JavaScript, utilizando las propiedades de una tabla HTML para buscar texto escrito dentro de las celdas de las tablas correspondientes.

Las operaciones de insertar y editar (Fig. 4 y Fig. 13-16**)** la información está implementado en PHP, simplemente utilizando la librería básica, sin usar *frameworks* especializados en XML para agregar, editar o eliminar información del archivo. La única desventaja de utilizar esta librería es que el archivo termina en una línea de texto, lo cual complica la lectura (humana) del archivo.

Una de las limitantes encontradas en el desarrollo de este trabajo es tratar las llaves primarias, como sólo conocemos una parte de la ruta en donde se guardará la información, sólo podemos verificar si dicha llave ya está ocupada, en el caso que esté ocupada notificar con un mensaje de error para que se corrija y en caso contrario simplemente insertar los datos del formulario que solicitó la inserción del elemento.

Se optó por la opción de utilizar un identificador que funja como llave primaria y que este a su vez no se repita, es decir, que se realizará la validación correspondiente para comprobar que los identificadores sean únicos; de esta manera estamos logrando que no se presenten otras complicaciones para acceder a la información de cada registro

que se tenga. La última funcionalidad implementada es la eliminación (Fig. 8-9) de registros, es una combinación de la consulta (Fig. 5-6) y edición de los registros (Fig. 10-12), ya que, JavaScript nos proporciona los parámetros requeridos para hacer la búsqueda del elemento con la librería de PHP, también conocido como XPath [9] y después simplemente eliminar el registro con funciones básicas utilizadas en el lenguaje antes mencionado.

4.1. Vista final del sistema

La página principal carga toda la información almacenada en el XML y la muestra en una presentación amigable para el usuario, donde se tienen las primeras separaciones de la jerarquía (estudiantes, profesores y materias) Fig. 2.

Fig. 2. Página principal.

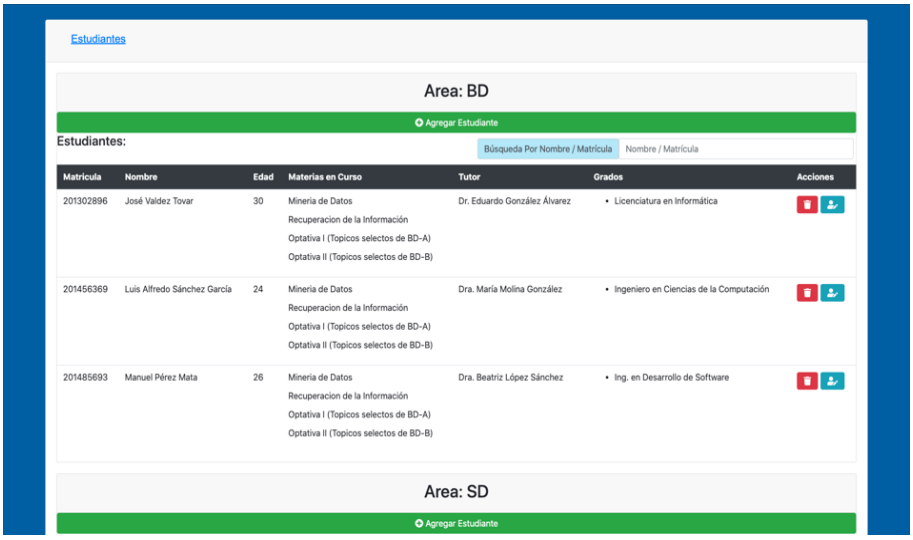


Fig. 3. Listado del alumnado.

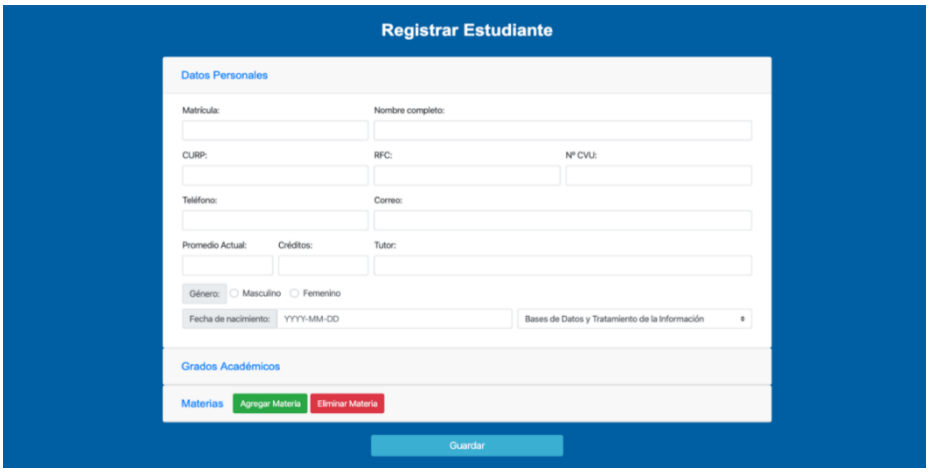


Fig. 4. Datos personales del estudiante.

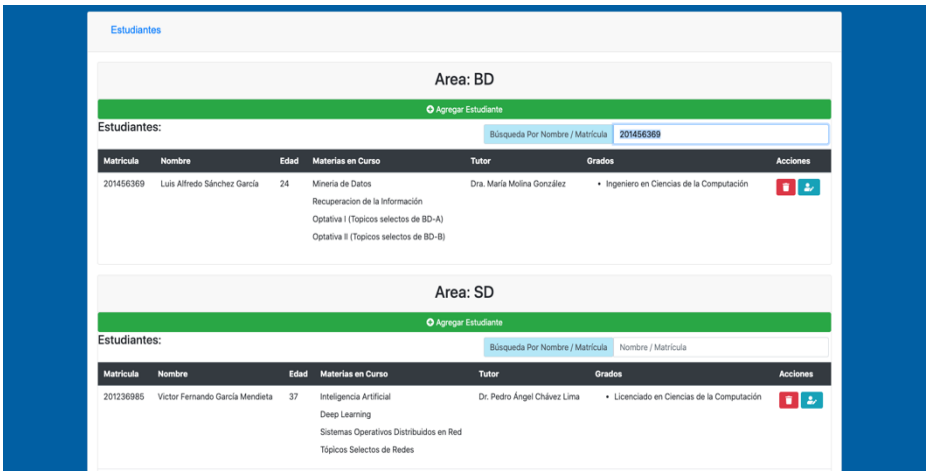


Fig. 5. Alumno con número de matrícula 201456369.

Una vez hecho clic en estudiantes (para este ejemplo), desplegará un listado del alumnado registrado en el XML Fig. 3.

Para agregar un registro damos clic en el botón “Agregar un estudiante” y tendremos que rellenar el formulario que nos arroja el sistema (ver Fig. 4).

Una vez llenados los datos de cada sección, damos clic en el botón “Guardar”, esto da la orden de tomar todos los datos del formulario y ubicarlos en la rama específica del archivo XML (ver fig.1), el cual empieza a hacer un recorrido de este fichero hasta llegar al apartado de estudiante, donde genera un nuevo nodo con la información escrita.

4.2. Consultas sobre los registros

Si lo que se pretende es filtrar a un alumno, por ejemplo, por medio de su matrícula, nos dirigimos a la sección de estudiantes, desplegamos la lista, a un costado de la parte derecha encontraremos el input destinado para el buscador, podemos filtrar los datos ya sea por nombre o matrícula, en este caso buscaremos al estudiante con una matrícula especificada por el usuario Fig. 5.

Para poder filtrar un profesor por medio de un nombre específico, nos dirigimos a la sección de profesores y de igual forma en el buscador que aparecerá del lado derecho; para ejemplificar esta consulta se buscará a la profesora de nombre Claudia Luna y podremos observar que imparte la materia de Sistemas Operativos Distribuidos en Red (ver Fig. 6).

Fig. 6. Datos de la Dra. Claudia Luna.

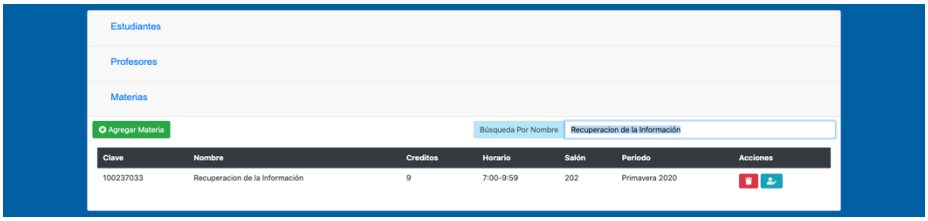


Fig. 7. Datos de la materia Recuperación de la información.

En el caso de querer conocer los horarios disponibles de la materia llamada "Recuperación de información", nos dirigimos a la sección de materias y en la parte derecha donde se encuentra el filtro de buscar, escribimos el nombre correspondiente a la materia que se requiera filtrar (ver Fig. 7).

De la misma manera, otra funcionalidad del sistema es eliminar un registro específico. En este caso podemos filtrar el nombre de un estudiante, y del lado derecho en la parte de acciones se encontrará un botón de color rojo con el ícono de un bote de basura, para eliminar dicho estudiante, únicamente damos clic sobre el botón y después nos mostrará un cuadro de diálogo como advertencia, ya que, una vez eliminado el registro no se puede recuperar, procedemos a dar clic en eliminar, nos direccionará a la página principal y podremos comprobar que efectivamente el registro ha sido eliminado, (ver Fig. 8-9).

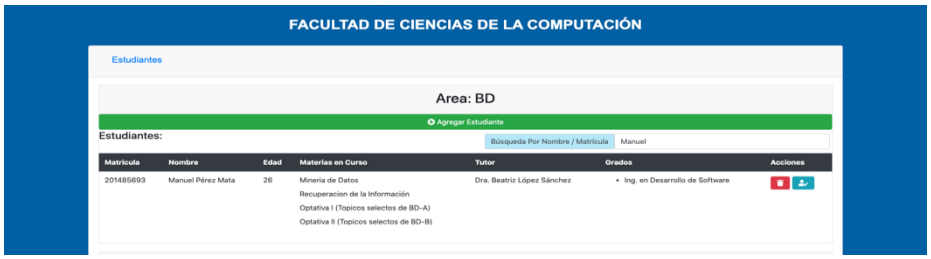


Fig. 8. Filtramos al estudiante llamado Manuel para eliminarlo por completo.

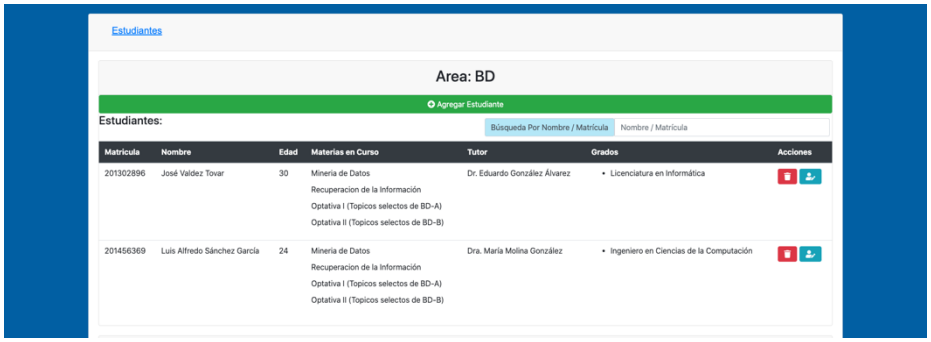


Fig. 9. Se observa que el estudiante llamado Manuel ha sido eliminado y ya no se encuentra entre los estudiantes del área de bases de datos.

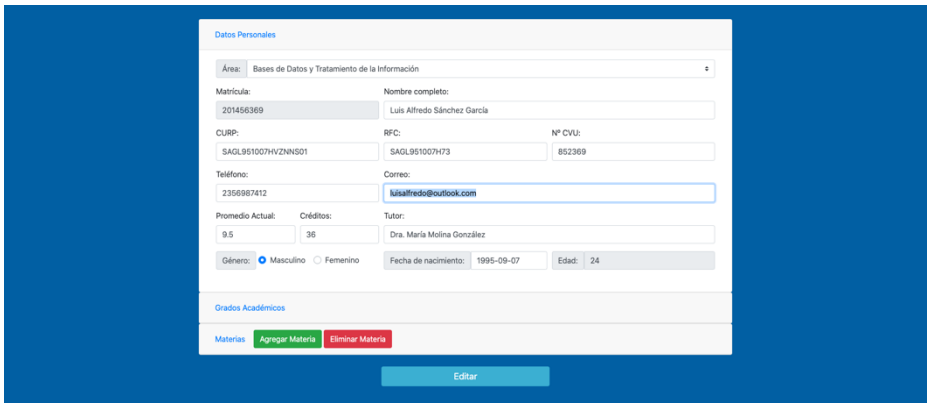


Fig. 11. Se ubica la parte donde se encuentra el correo y se escribe el nuevo para ser registrado.

Si lo que se pretende es actualizar un registro, de igual manera se puede realizar. Podemos filtrar un estudiante, por ejemplo, cambiando su correo electrónico registrado, para ello, buscamos a un alumno específico, del lado derecho en la parte de acciones, nos aparecerá un botón de color azul, damos clic y nos direccionará a una ventana con todos los datos del estudiante, localizamos la parte donde esté el correo y se procede a modificarlo, damos clic en el botón editar que se encuentra en la parte inferior, después

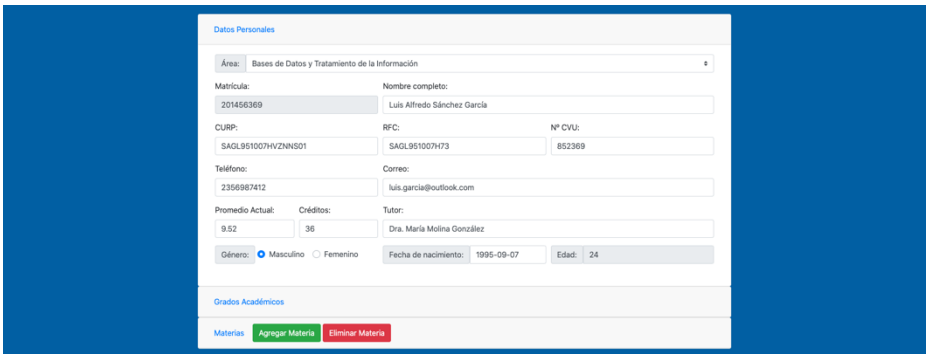


Fig. 12. Se comprueba que el correo ha sido editado correctamente ingresando a la información del estudiante llamado Luis.

de esto nos aparecerá un cuadro de diálogo indicándonos que la acción se ha completado (ver Fig. 10-12).

Los resultados obtenidos con la realización de este sistema para uso de tecnologías de la web semántica mediante XML han sido satisfactorios, ya que, se han logrado cumplir los objetivos planteados desde un inicio logrando que el procesamiento de la información sea más sencillo para la computadora mediante las herramientas XML y Xpath. De la misma manera, al hacer uso de una aplicación web para el tratamiento y recuperación de la información ha sido de gran ayuda, puesto que se logran realizar consultas específicas sobre los datos desde el mismo software. Además de que se tiene mejor organizada la información; por lo que es importante mencionar que con el desarrollo de este trabajo se lograron llevar a cabo funcionalidades importantes como lo son la inserción, eliminación y modificación de registros, así como llevar a cabo consultas sobre los mismos y gracias a ello se obtiene un software completo que brinda una alternativa al uso de herramientas de la web semántica para estructuración y recuperación de la información.

5. Conclusiones

El presente trabajo se llevó a cabo a través de la información obtenida directamente del personal de la Facultad de Ciencias de la Computación, como profesores, alumnos y con secretaría de posgrado para información relacionada a las materias que se imparten.

La utilización de un árbol jerárquico ha sido de gran ayuda para resolver la problemática que corresponde a la estructura, organización y manejo de la información mediante el uso de herramientas de la web semántica como son los archivos XML. La utilización de este lenguaje en conjunto con Xpath ha sido muy útil para el almacenamiento y consulta de los datos, ya que, se vuelve un método práctico y como una opción más en este ámbito, con la finalidad de tener una alternativa a las bases de datos tradicionales. Puesto que, en este caso brinda la misma funcionalidad para trabajar con la información que se está consultando, pues de esta manera logramos generar la inserción de nuevos registros al archivo fuente que es el XML, a su vez desde

el sistema desarrollado en web, podemos consultar por ejemplo, estudiantes del área de sistemas distribuidos, para después seleccionar el estudiante que se esté buscando y realizar la eliminación o bien actualización de ese registro según sea la necesidad del usuario.

En cuanto a los resultados obtenidos nos encontramos satisfechos porque se logró el objetivo de utilizar una metodología alternativa a los sistemas de bases de datos. Además de desarrollar un software simple, intuitivo y práctico para el usuario, el cual servirá para cumplir con la funcionalidad de inserción, eliminación, edición y consulta de la información, además que queda abierto para adaptar a otras aplicaciones o inclusive realizar las mejoras necesarias y tener nuevas aportaciones en el ámbito académico. Finalmente, como propuesta de trabajo a futuro mencionamos la creación de un modelo ontológico, basado instituciones de educación superior, donde se pueda manejar más información y a su vez más detallada, que tenga que ver con el personal académico como lo son docentes y estudiantes, así como personal administrativo, de la misma manera, incluir datos referentes a la carga académica que se ofrece, es decir, materias que se imparten o bien otros cursos que se brinden, abarcando todo a mayor detalle, ya que, lo que se busca con una ontología es ampliar el modelo que se esté desarrollando, para que con ello, el sistema web propuesto para la recuperación de información tenga una mejor funcionalidad y sea de mayor utilidad para el usuario.

Agradecimientos. Esta investigación es apoyada por el Fondo Sectorial de Investigación para la Educación, con el proyecto CONACyT CB/257357 y por el proyecto VIEP-BUAP 100409344-VIEP2019.

Referencias

1. Gardner, J.R., Rendon, Z.L.: XSLT and XPath: A guide to XML transformations. Upper Saddle River, New Jersey: Prentice Hall (2002)
2. Gheranghiyeh, M.S.: Geospatial processing in BaseX. Doctoral dissertation, Verlag nicht ermittelbar (2014)
3. Lazim, R.Y., Kamaludin, A.: Going Native: Indexing architecture of eXist-db_ an open source native XML database system. In: IEEE International Conference of IEEE Region 10 (TENCON), Xi'an (2013)
4. Thierry, B.: XML práctico Bases esenciales conceptos y casos prácticos. Editorial ENI (2015)
5. Martínez, G.F.A., Martín, Q.G.: Introducción a la programación estructurada en C. Valencia: Universitat de València (2003)
6. Holzner, S.: XPath: navigating XML with XPath 1.0 and 2.0: kick start. Indianapolis, Sams Pub. (2003)
7. Welling, L., Thomson, L.: Desarrollo web con PHP y MySQL: [PHP 5.3 y MySQL 5.1]. Anaya Multimedia (2009)
8. Gutierrez, E.: JavaScript: Conceptos básicos y avanzados: bibliotecas Prototype y Script.aculo.us. Ediciones Eni (2009)
9. Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., Seguy, D., Vrana, J.: Manual de PHP. https://www.php.net/manual/es/book.simplexml.php (2020)

10. Bravo, M., Martínez-Reyes, F., Rodríguez, J.: Representation of an academic and institutional context using ontologies. Research in Computing Science, 87 (2014)
11. Bravo-Contreras, M., Hoyos-Reyes, L., Reyes-Ortiz, J.: Methodology for ontology design and construction. Contaduría y Administración, 64(4) (2019)
12. Flores, J.C., Tovar, M., Cervantes, A.P.: Modelo ontológico para representar información sobre la práctica profesional en una institución educativa. Research in Computing Science, pp. 127–140 (2017)

ISSN 1870-4069

Control difuso mediante la estimación mejorada de la distancia inter-electrodos en un proceso de micro maquinado electroquímico por pulsos

Octavio Catarino Aguilar, Iván Gallegos Miranda,
Everardo Efrén Granda Gutiérrez, José Luis Tapia Fabela,
José Arturo Pérez Martínez

Universidad Autónoma del Estado de México,
México

octavio_ca@outlook.com, igallegosmiranda@outlook.es,
eegrandag@uaemex.mx, joseluis.fabela@gmail.com,
japerezm@uaemex.mx

Resumen. El Micro-Maquinado Electroquímico por Pulsos es un método de manufactura no convencional, en donde la herramienta y la pieza de trabajo no están en contacto, por lo que no se presenta desgaste, se evita la generación de microgrietas y rebabas metálicas, siendo esto una virtud para la fabricación de piezas de tamaño reducido. Para garantizar la precisión del proceso, es necesario mantener la distancia inter-electrodos constante, para una remoción homogénea de material. Esta distancia no se puede medir directamente, por lo que se debe estimar de manera indirecta mediante otras variables del proceso: voltaje y corriente. Durante el maquinado, el área expuesta del electrodo se modifica conforme este penetra en la pieza de trabajo, por lo que la corriente medida no indica de manera directa la dimensión de la distancia inter-electrodos. Para ello, se debe considerar la evolución del área de exposición del electrodo, ajustando de forma automática el valor de esta. Una vez que se determinó la distancia interelectrodos, se diseñó un controlador difuso cuyo objetivo es asegurar que esta distancia se mantenga en el orden de 240 µm, inclusive si cambia el área de exposición del electrodo, controlando el avance vertical del mismo. La implementación del control difuso con ajuste automático de la estimación de la distancia inter-electrodos demuestra ser una alternativa eficaz para alcanzar la profundidad deseada, manteniendo una remoción de material uniforme. Los resultados muestran que el esquema propuesto permite estimar el valor real de la distancia inter-electrodos logrando alcanzar la profundidad deseada a lo largo de la trayectoria horizontal, maquinando un canal de manera homogénea con una precisión limitada solamente por la resolución de movimiento del sistema mecánico, en el orden de 20 µm.

Palabras clave: Control por lógica difusa, micro-maquinado electroquímico por pulsos (PECMM), procesos de manufactura no convencional.

Fuzzy Control through Improved Estimation of Inter-Electrode Gap in a Pulsed Electrochemical Micro-Machining Process

Abstract. Pulsed Electrochemical Micro-Machining is a non-conventional manufacturing process where the tool electrode and the work piece are not in direct

contact, avoiding wear, microcracks as well as the production of metallic burrs; thus, making it virtuous for the machining of reduced-dimension pieces. To ensure the precision of the process, the inter-electrode distance must be kept constant, to allow a homogeneous removal of material. Such a distance cannot be directly measured; then, it must be indirectly estimated through another process variables: voltage and current. During the process, the exposed area of the electrode is modified due to the incursion of the tool electrode into the workpiece; therefore, the measured current does not necessarily indicate the dimension of the inter-electrode distance. Then, the evolution in the exposed electrode area must be considered, automatically adjusting its value. Once the inter-electrode distance was determined, a fuzzy logic controller was designed in order to ensure a constant inter-electrode distance in the range of 240 µm, even if variations on the exposed area of the electrode occur, by means of the vertical movement of the electrode. The implementation of the fuzzy controller with automatic adjustment of inter-electrode distance calculation, demonstrates to be an effective alternative to reach the desired depth, keeping a uniform material removal. Results show that the proposed scheme allows estimating the correct value of the inter-electrode distance, homogeneously machining a channel along a horizontal trajectory with a precision limited by the resolution of movement of the mechanical system, in the order of 20 µm.

Keywords: Fuzzy logic control, pulsed electrochemical micro-machining (PECMM), non-conventional manufacturing processes.

1. Introducción

La creciente demanda en la fabricación de componentes de tamaño reducido en áreas como robótica, biología, microelectrónica, material biomédico, entre otros; da auge a la investigación en el desarrollo de procesos de micro manufactura [1]. Como aplicaciones de este proceso se pueden mencionar: la fabricación de micro boquillas, micro engranes, instrumentos quirúrgicos (bisturíes), microtubos, micro inyectores, micro bujes entre otras aplicaciones [2].

El proceso de micro manufactura se realiza mediante los métodos convencional y no convencional. El proceso de manufactura convencional se genera mediante fricción entre la herramienta y la pieza de trabajo, produciendo micro fisuras y residuos metálicos; por otra parte, el proceso de manufactura no convencional realiza la remoción de material sin tener contacto entre la herramienta y la pieza de trabajo, de tal manera que no son generadas micro fisuras ni rebabas metálicas, lo cual ayuda a que la pieza de trabajo mantenga su integridad fisca, por ejemplo, en el fuselaje de un avión [3].

Uno de los procesos de manufactura no convencional es el Micro-Maquinado Electroquímico por Pulsos (PECMM, por sus siglas en inglés), en donde la herramienta y la pieza de trabajo no tienen contacto entre sí, existiendo una brecha entre ambas; a la que se denomina *gap* (distancia inter-electrodos) [4]. Dicho proceso aprovecha la corriente eléctrica y el fenómeno de disolución anódica para la remoción de material metálico [5].

El controlar PECMM presenta cierta complejidad, debido a que los resultados obtenidos al término del proceso tales como: tasa de remoción de material (MRR por sus siglas en inglés), diámetro de sobre corte (DOC por sus siglas en ingles) y

profundidad de maquinado, dependen de diversos parámetros de entrada. Dentro de estos parámetros se encuentran: frecuencia de la señal, ciclo de trabajo, distancia inter-electrodos, flujo del electrolito, corriente de maquinado y voltaje de polarización [6, 7].

El proceso de Maquinado Electroquímico (ECM por sus siglas en inglés), presenta la dificultad de medición de la distancia inter-electrodos, al estar en el orden de los micrómetros, y sin un rango visible, al estar inmersa en el fluido de trabajo, y dentro de la pieza, por lo que es muy complejo realizar la medición de dicha distancia implementando sensores tradicionales [8, 9]. Es por ello que, para saber la magnitud de la distancia inter-electrodos se estima indirectamente mediante la corriente de maquinado [10].

De acuerdo con [11], el utilizar técnicas de control convencionales resulta ineficiente para poder controlar el proceso ECM; por tal motivo, el implementar sistemas de control moderno resulta ser una alternativa viable, debido a que el proceso de maquinado electroquímico es no lineal; es decir, su comportamiento con respecto al tiempo tiene una variación que no es constante y sus variables necesitan ser reajustadas en tiempo real. Por tal motivo en [12] se reporta la experimentación del proceso ECM, en donde afirman que el controlar el proceso únicamente controlando los parámetros voltaje y flujo del electrolito resulta insuficiente debido a que nunca alcanza la estabilización.

En [13] se presenta un estudio para mantener constante la remoción de material, simulando el proceso utilizando la técnica de lógica difusa, la cual asemeja el control al actuar del ser humano, ya que se basa en comprender los cuantificadores de cualidad para nuestras inferencias y así determinar qué grado de pertenencia tienen el criterio a evaluar a los denominados conjuntos difusos. Resultó que, para mantener una remoción de material constante, es indispensable mantener la misma distancia inter-electrodos en todo el transcurso del proceso y con ello se garantiza una profundidad homogénea. Por tal motivo, en un trabajo previo mostrado en [14] se establece una propuesta para controlar la distancia inter-electrodos utilizando la técnica de lógica difusa, manteniendo constante la corriente de maquinado inclusive si la pieza de trabajo presenta perturbaciones; el control actúa ajustando y manteniendo constante la corriente de maquinado.

En [15], se implementa un control difuso para la estimación de la tasa de remoción de material en un proceso de maquinado por descarga; su planta, basada en el modelo Mamdani recibe 3 parámetros, los cuales son: corriente de descarga, tiempo del pulso en alto, y tiempo de pulso en bajo. Considera 6 reglas de pertenecía para así predecir la tasa de moción de material. El control difuso fue evaluado usando el software Matlab y comparando los resultados con pruebas de maquinado reales; bajo los mismos criterios, el modelo difuso probó tener una remoción de material en un 97.36% de similitud con las pruebas experimentales reales, lo cual permite considerar a la lógica difusa viable para ser usada como técnica de modelado.

Si bien en el trabajo no es implementada como técnica de control, comparando los resultados estimados y los obtenidos puede ser considera como prospecto para su implementación como técnica de control en un sistema de maquinado electroquímico por pulsos.

Técnicas basadas en redes neuronales también han sido puestas a prueba para la estimación de tasa de remoción de material y avance de la pieza de trabajo, tal como se explica en [16], donde se implementa una red neuronal que admite 4 criterios: la

corriente de maquinado, el tiempo de encendido y pagado de la fuente y la frecuencia; en su capa oculta cuenta con 3 neuronas y como salida se obtiene la tasa de remoción que permite determinar la velocidad de avance de la herramienta de trabajo. Para entrenar la red neuronal se usaron 19 muestras y 3 para su validación. Comparadas con las pruebas reales, se tuvo una diferencia del 1 al 5 %; sin embargo, la implementación de una red neuronal no considera que la pieza posiblemente puede presentar irregularidades en su superficie, lo cual en el peor de los casos podría llevar al sistema a un corto circuito.

En [14], se identifica que al finalizar el proceso la profundidad de avance de la herramienta de trabajo en comparación a la profundidad deseada presenta una diferencia. Por tal motivo, en este trabajo se presenta el desarrollo de un control basado en lógica difusa donde se mantiene constante la distancia inter-electrodos y la remoción de material, a través del monitoreo de la corriente de maquinado, generando el cálculo real de la distancia inter-electrodos al considerar dentro del sistema el incremento del área de exposición del electrodo, que se modifica cada vez que la herramienta de trabajo desciende y que, al estar en movimiento durante las trayectorias lineales, alteran la magnitud de la corriente de maquinado.

2. Metodología

En la Fig. 1. se muestra el esquema de los elementos del sistema de micro maquinado electroquímico por pulsos; se integra por un recipiente donde se coloca la pieza de trabajo, la cual está inmersa parcialmente en un electrolito de $H_2O + NaCl$ en una concentración del 16.6% (en masa), con una conductividad de 217.1 $\frac{mS}{cm}$ o 4.7 $\Omega * cm$. El electrolito fluye por un tubo de acero inoxidable de 700 μm de diámetro externo y 490 μm de diametro interno, con un flujo de 1 L/min.

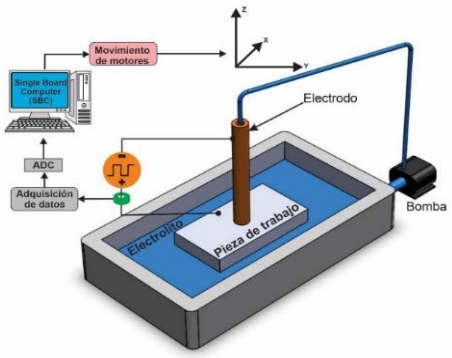


Fig. 1. Diagrama general del sistema.

Se realizaron pruebas de maquinado con la finalidad de determinar la causa del aumento en la profundidad final obtenida, mediante el análisis de los datos como se presenta en la Fig. 1. Se observa que el tiempo entre un ajuste y otro se prolonga; cada ajuste corresponde al descenso de un paso (20 µm) de la herramienta de trabajo que se actualiza una vez que la corriente vuelve a un valor mínimo debido al aumento en la

distancia inter-electrodos. Este descenso ocasiona un incremento en la corriente de maquinado tras cada ajuste, pues el electrodo se vuelve a acercar a la pieza. Esto da como resultado que la velocidad real de remoción de material disminuye, debido a que mientras se realiza el proceso de maquinado, la herramienta de trabajo penetra en la pieza de trabajo agregando área de exposición entre ambas, y reduciendo la densidad de corriente que se distribuye en dicha área. De esta forma, el valor real de la distancia inter-electrodos que estima el sistema no es el correcto y por ello el tiempo entre un ajuste y otro se prolonga; tras este aumento el tiempo de exposición es mayor y da como resultado una profundidad final mayor a la esperada, puesto que la distancia real a la que corresponde no ha sido determinada adecuadamente.

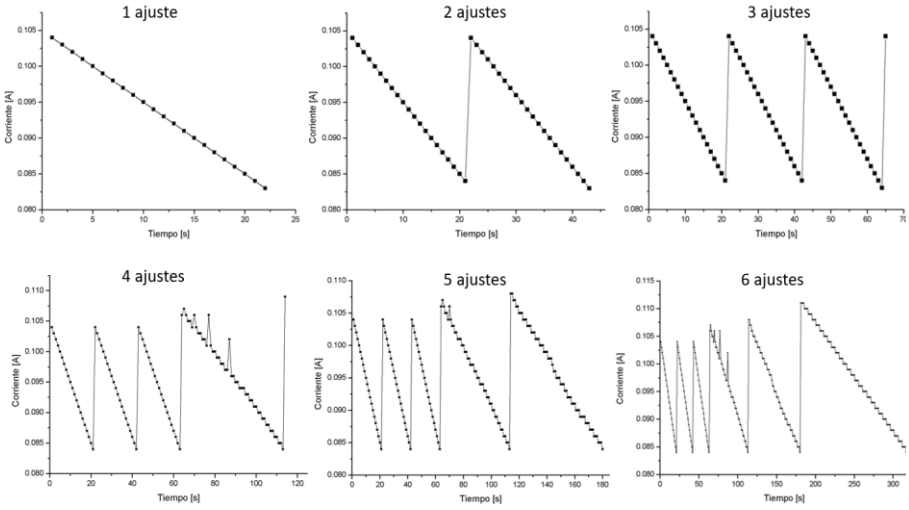


Fig. 2. Comportamiento de la corriente de maquinado: tras cada ajuste, el tiempo necesario para actualizar el valor de la corriente se incrementa debido al error en la estimación de la distancia inter-electrodos.

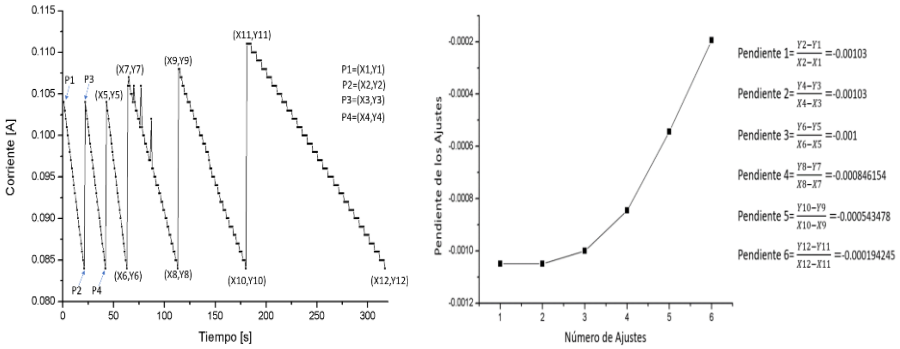
Tras visualizar el comportamiento de la corriente de maquinado después de cada ajuste, se analizó la pendiente de esta, implementando la ecuación (1):

$$Pendiente = \frac{I_2 - I_1}{S_2 - S_1}. \quad (1)$$

donde:
I_1=punto máximo en la corriente,
I_2=punto mínimo en la corriente,
S_1=punto máximo en el tiempo,
S_2=punto mínimo en el tiempo.

En la Fig. 3 (a) se muestra el patrón de cambio de la magnitud del valor de la pendiente de la corriente de maquinado; al graficar el valor de la misma con respecto a los ajustes realizados Fig. 3 (b), presenta un comportamiento creciente. Entonces, lo que se propone es generar un ajuste para mantener constante la magnitud de la pendiente de la corriente de maquinado; así, en cuanto el sistema identifique una variación en la

pendiente, este, elevará o hará descender la herramienta de trabajo y de esta forma el sistema difuso recibirá un valor más cercano al real de la distancia inter-electrodos, compensando de manera natural el incremento en el área de exposición debido a la penetración del electrodo en la pieza.



(a) (b)

Fig. 3. Patrón del comportamiento de la pendiente, previo a la incorporación del cálculo mejorado de la distancia inter-electrodos.

2.1. Ajuste de la distancia inter-electrodos con base en la pendiente

Para considerar el área expuesta adicional y entonces poder reajustar la corriente de maquinado de referencia que usa el control difuso para estimar y controlar la distancia inter-electrodos durante el proceso, se implementó el cálculo de la pendiente de la corriente de maquinado y su reajuste correspondiente como parte complementaria al estimar la distancia inter-electrodos por parte del sistema computacional.

En la Fig. 4 se presenta el diagrama de flujo de esta adecuación, donde se establecen los parámetros iniciales de la distancia inter-electrodos. Se crea un arreglo del tamaño de 5 elementos vacío, donde se almacenará la corriente de maquinado; una vez que el arreglo está lleno, se obtiene un promedio para después calcular la pendiente, que se compara con la pendiente inicial. Si presenta una variación significativa, se hace una solicitud de reajuste al sistema computacional que determinará si la herramienta de trabajo se eleva o desciende. Como respuesta habrá un cambio en el valor de la corriente de referencia que usa el sistema computacional para la estimación de la distancia inter-electrodos. Con este reajuste se pretende mantener una remoción de material controlada y constante que, como resultado, genere que la profundidad de maquinado final sea la deseada.

Se ha comprobado que, implementando técnicas de control difuso se puede controlar el proceso PECMM [14], manteniendo la distancia inter-electrodos lo más cercana a los 240 μm [19], por ello para alcanzar la profundidad deseada se incorporó el reajuste de la corriente de maquinado de referencia basada en el cálculo de su pendiente al modelo difuso ya existente, integrándose en la sección de retroalimentación de la planta, como se puede observar en la Fig. 5. El sistema de control se basa en el diseño Mamdani y el modelo SISO (single input- single output), la entrada al sistema de control es el error de la distancia inter-electrodos, que es estimada mediante el

monitoreo de la corriente de maquinado y el reajuste de la pendiente de la corriente de referencia, haciendo uso de la ecuación (2):

$$g = \frac{A}{I_{RMS} * (R * c)}, \tag{2}$$

donde:

g = distancia inter-electrodos,

A= área de exposición del electrodo,

I_{RMS}= corriente RMS de maquinado,

R= resistividad del electrolito,

c= (densidad^2*velocidad de remoción especifica) = (7.87^2) ($3.0*10^{-2}$).

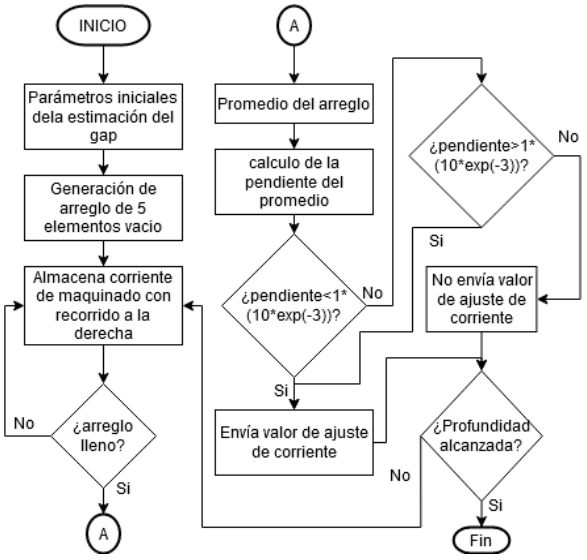


Fig. 4. Diagrama de flujo para ajuste de distancia inter-electrodos con base en el valor de la pendiente en la corriente de maquinado.

Por otro lado, la salida es el movimiento descendente o ascendente de la herramienta de trabajo, que se lleva a cabo mediante el sistema mecánico y los motores a pasos. La sección de reajuste de la corriente de referencia, se calcula y aplica fuera del regulador y antes del cálculo de distancia inter-electrodos; de esta manera, si el sistema detecta una variación en la magnitud de la pendiente se realizará un reajuste en la corriente de referencia. Este reajuste será contemplado al momento de estimar la distancia inter-electrodos, ya que se estará contemplando el área expuesta adicional para que de esta forma se pueda realizar una remoción de material controlada y constante a lo largo de la trayectoria de maquinado alcanzando la profundidad final deseada.

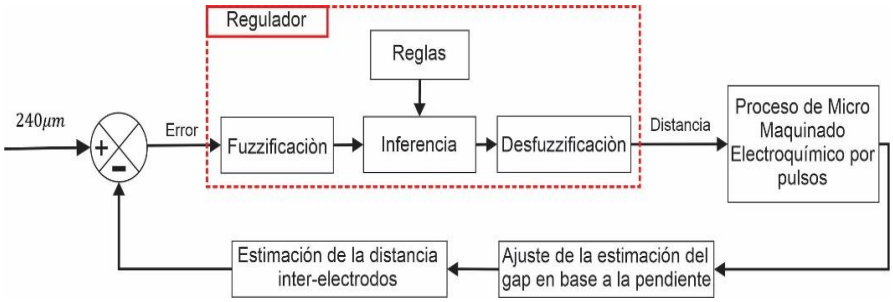


Fig. 5. Modelo SISO del sistema control implementando reajuste por pendiente.

Tabla 1. Conjuntos difusos del universo de discurso error.

Variable Lingüística	Función	Pertenencia a los conjuntos difusos
Negativo	μ_L	$(-60, 60)$
Nulo	μ_T	$(-60, 0, 60)$
Positivo	μ_L	$(-60, 60)$

Tabla 2. Conjuntos difusos del universo de discurso distancia.

Variable Lingüística	Función	Pertenencia a los conjuntos difusos
Desciende electrodo	μ_L	$(120, 480)$
Mantén electrodo	μ_T	$(120, 240, 480)$
Eleva electrodo	μ_L	$(120, 480)$

Los conjuntos difusos de la entrada o "error" de la distancia inter-electrodos que implementa el control se muestran en la **Tabla 1**, los parámetros fueron definidos con base en la caracterización y a la resolución del sistema mecánico.

En la **Tabla 2** se muestran los conjuntos difusos de la distancia requerida para desplazar el electrodo verticalmente y mantener el punto de referencia.

Los conjuntos difusos de entrada utilizados para el control, son representados en 3 intervalos, teniendo representación de saturación abierta como μ_L, y representaciones triangulares como μ_T, tal como se aprecia en la **Tabla 1**. Por otro lado, los conjuntos difusos de salida del universo de discurso "Distancia", se definen con 3 intervalos con formas μ_L y μ_T.

La relación de los conjuntos difusos de entrada y salida dan como resultado las reglas de inferencia mostradas en la Tabla 3, que permiten definir el comportamiento del sistema. De esta manera, si el control difuso a su entrada tiene un error negativo significa que la distancia inter-electrodos está en un valor mayor a los 240 μm; como respuesta, el sistema descenderá el electrodo en intervalos de 20 μm hasta que la distancia inter-electrodos se encuentre lo más cercana al punto de referencia. De forma contraria, si el error a la entrada del control difuso es positivo como respuesta el sistema elevará el electrodo en intervalos de 20 μm hasta alcanzar el punto de referencia,

Tabla 3. Reglas de inferencia.

Regla	Reglas de inferencia
1	Si (Error es negativo) entonces (Desciende electrodo)
2	Si (Error es nulo) entonces (Mantén electrodo)
3	Si (Error es positivo) entonces (Eleva electrodo)

Tabla 4. Parámetros iniciales y finales en pruebas de maquinado.

Prueba	Duración del proceso PECMM [min.]	Longitud prestablecida en el sistema [mm]	Longitud resultante del proceso PECMM [mm]	Profundidad del canal resultante [µm]
1	16	10	10.78	240
2	33	10	10.89	500

mientras el error a la entrada del control sea nulo, el electrodo se mantendrá sin movimiento vertical.

La técnica de defusificación empleada fue el centroide, la cual transforma la salida difusa en un número real, ubicando la coordenada $zeta$ (z) del centro de gravedad del conjunto difuso resultante a la salida y se explica en la ecuación (3) [17]:

$$centroide = \frac{\int z\mu_Y(z)dz}{\int \mu_Y(z)dz}, \tag{3}$$

donde:

z = universo de discurso,

μ_Y=función de pertenencia al conjunto de salida.

Para la validación del sistema de control se realizaron dos pruebas de maquinado lineal de 1 cm de longitud, a profundidades de $240\ \mu m\ y\ 500\ \mu m$, respectivamente. El voltaje de la fuente usada fue de 5 V con un ciclo de trabajo al 50 % y con pulsos a una frecuencia de 40 kHz. Como material de trabajo se usó acero estructural AISI-1018, y como electrolito una mezcla de $H_2O + NaCl$ en una concentración en masa del 16.6%. Las mediciones de profundidad se realizaron con un medidor de alturas digital de la marca FLOWLER modelo 48976.

3. Resultados

La **Tabla 4** presenta los resultados tras implementar el control difuso agregando el ajuste de la pendiente de la corriente de referencia al calcular la distancia inter-electrodos. La **prueba 1** tuvo una duración de 16 minutos, con una longitud lineal resultante de 10.78 mm, .78mm más de lo esperado y dentro del margen de error esperado, la profundidad homogénea alcanzada fue la deseada, de $240\mu m$. Por otra parte, la **prueba 2**, tuvo una duración de 33 minutos, su longitud lineal resultante fue de 10.89 mm, 0.89 mm más de lo esperado, debido a que el control está enfocado en el eje Z, mientras que en el desplazamiento horizontal del eje Y y X aún no se implementa

dicho control; la profundidad final fue de $500\ \mu m$, al igual que la prueba número 1 se alcanzó la profundidad deseada.

En la Fig. 6 y Fig. 7 se muestra el comportamiento de la corriente de maquinado a lo largo de todo el proceso de las pruebas realizadas. Es visible el continuo ajuste de la pendiente, donde cada ajuste corresponde a un movimiento de $20\ \mu m$ de la herramienta de trabajo. Se observa el crecimiento en el valor de referencia de la corriente de maquinado que toma el sistema de control difuso para realizar el avance vertical de la herramienta. El aumento de este valor en la corriente se debe a que, mientras el proceso se realiza, la herramienta de trabajo se introduce cada vez más en la pieza de trabajo aunado al movimiento horizontal de la misma, originando que el área expuesta entre pieza y herramienta aumente debido a que los bordes laterales de la herramienta de trabajo entran en contacto con el proceso de maquinado electroquímico. De este modo, la corriente requerida para realizar la misma remoción debe incrementarse si se desea mantenerla constante.

Para poder estimar la distancia inter-electrodos real, el sistema considera el valor de la pendiente del primer ajuste que realiza el control difuso, y lo toma como referencia durante todo el proceso, manteniendo el valor de esa pendiente lo más cercano al primer valor. En el momento que se detecta un cambio, se realiza un reajuste en el valor de la corriente de maquinado; a su vez, la etapa de estimación de distancia inter-electrodos toma ese valor y lo despeja en la ecuación (2) para obtener el área de exposición de la herramienta de trabajo.

Con ello, este parámetro es recalculado en cada ajuste y es posible estimar el valor real de la distancia inter-electrodos considerando el aumento del área de exposición conforme la herramienta de trabajo penetra en la pieza de trabajo. El control difuso, al tener el valor real de la distancia inter-electrodos puede realizar el descenso de la herramienta de trabajo en periodos más cortos, siempre manteniendo la distancia inter-electrodos cercana a los $240\ \mu m$ (dentro de los límites de resolución de movimiento del sistema mecánico, de 20 µm).

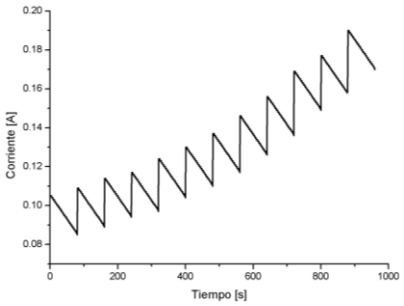


Fig. 6. Corriente de maquinado en prueba de 240 µm de profundidad.

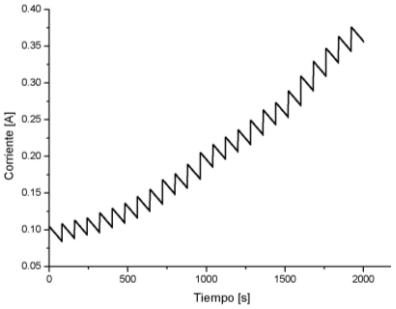
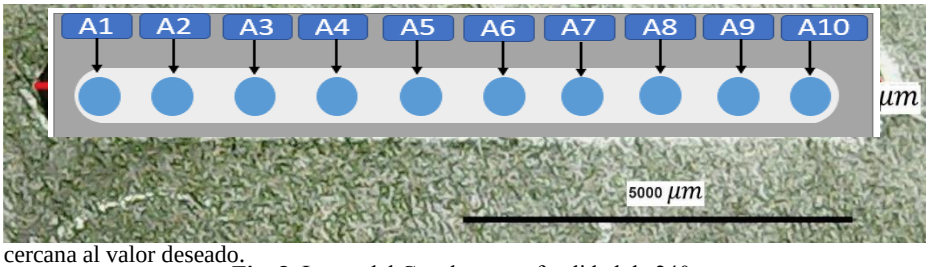


Fig. 7. Corriente de maquinado en prueba de 500 µm de profundidad.

El aumento en la corriente de maquinado que se puede observar tanto en la Fig. 6 y Fig. 7, es resultado de la compensación del aumento en el área de exposición entre la herramienta y pieza de trabajo y los ajustes escalonados se dan como resultado del ajuste de área de exposición en la ecuación (2), que es usada para la estimación de la distancia inter-electrodos.



cercana al valor deseado.

Fig. 8. Largo del Canal con profundidad de 240µm.

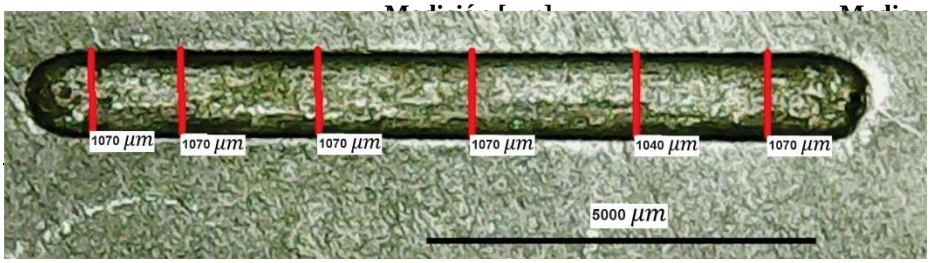


Fig. 9. Ancho del canal con profundidad de 240 µm.

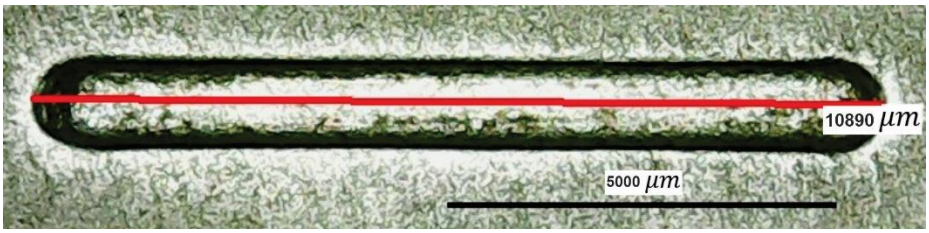


Fig. 10. Largo del Canal con profundidad de 500µm

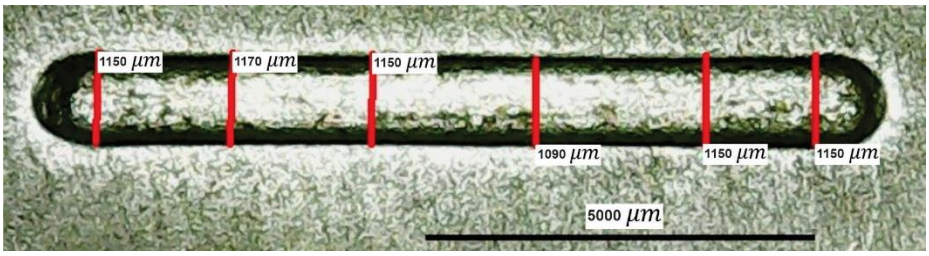


Fig. 11. Ancho del canal con profundidad de 500 µm.

La Fig. 8 muestra la longitud lineal de la prueba de maquinado número 1 cuya profundidad es de 240 μm , en la Fig. 9 se muestra el ancho del canal resultante de la prueba número 1, mismo que se aprecia ser homogéneo a lo largo de toda la trayectoria. La longitud lineal de la prueba numero 2 cuya profundidad fue de 500 μm se puede apreciar en la Fig. 10 así mismo en la Fig. 11 se puede observar el ancho resultante del canal el cual es lo mas homogeneo posible a lo largo de toda la trayectoria.

Para comprobar que se alcanzó la distancia deseada se realizaron 10 mediciones distribuidas equitativamente a lo largo los maquinados resultantes, tal y como se muestra en la Fig. 12.

Obteniendo el valor real de la distancia inter-electrodos mediante el reajuste de la corriente de maquinado de referencia por pendiente e integrándolo al sistema de control difuso, es posible obtener una remoción de material controlada y constante a lo largo de maquinados lineales, como se puede comprobar al analizar las mediciones de las dos pruebas realizadas en los 10 puntos establecidos y notar que el valor de la profundidad en todos ellos es similar a la deseada por el usuario tal y como se puede apreciar en la Tabla 5.

4. Conclusiones

PECMM es un proceso de maquinado no convencional, utilizado para el barrenado de orificios pequeños en materiales conductores. Este proceso es difícil de controlar debido a que en el intervienen fenómenos físicos, químicos, eléctricos, entre otros. Aunado a esto, aplicarlo a maquinados con movimientos en otros ejes, como en el caso del maquinado de canales, agrega un nivel de dificultad mayor, ya que la herramienta de trabajo ya no se encuentra estática como en la generación de barrenos ciegos, si no por el contrario se encuentra en movimiento constante mientras se adentra en la pieza de trabajo, lo cual genera que el área de exposición entre herramienta y pieza de trabajo se modifique constantemente y, por ello, el proceso no es candidato para la implementación de técnicas de control lineal o tradicional pero si para técnicas alternativas como lo es la lógica difusa.

El presente trabajo tiene como finalidad estimar la distancia inter-electrodos de manera precisa, mediante la incorporación de la pendiente de la corriente de maquinado, que es un indicador de la velocidad de remoción de material, así como con la implementación de un control difuso, manteniendo dicha distancia constante en un valor cercano a los $240\ \mu m$, y así poder alcanzar la profundidad homogénea cuando se realiza el maquinado de canales.

Tras analizar e identificar que la pendiente en la corriente de maquinado es un factor que también se debe considerar e integrar al control difuso, fue posible generar una remoción de material controlada y constante a lo largo del canal maquinado, generando reajustes en la corriente de maquinado en el lazo de retroalimentación que usa el control difuso para la estimación de la distancia inter-electrodos. Este ajuste considera la actualización del área expuesta entre herramienta y pieza de trabajo. Como resultado, la profundidad final alcanzada fue homogénea a lo largo de todo el maquinado del canal y del mismo valor al esperado; claro está que dicha distancia está determinada en un intervalo de $20\ \mu m$, siendo este valor el mínimo permisible por la resolución de movimiento del sistema mecánico implementado.

La diferencia entre la profundidad final y la deseada ya no se presenta, ya que con la implementación propuesta se obtuvieron canales maquinados con profundidad homogénea a lo largo de toda su longitud.

Como trabajo a futuro, se pretende integrar esta solución a maquinados con formas complejas integrando al control difuso criterios que permitan controlar la velocidad de remoción, mediante la manipulación de la fuente de voltaje. Asimismo, implementar el

control en los dos ejes restantes del sistema mecánico, manteniendo una remoción controlada en todos los ejes del sistema cartesiano en donde se reproduce el proceso.

Referencias

1. Grover, M.: Fundamentos de manufactura moderna. Mc Graw Hill (2010)
2. Kalpakjian, S., Schmid, S.R.: Manufactura, ingeniería y tecnología. Prentice Hall (2005)
3. Hassan, A., Gawad, E.H.: Advanced machining processes. McGraw-Hill (2005)
4. Katz, Z., Tibbles, C.: Analysis of micro-scale ECM Proccess. The International Journal of Advance Manufacturing Technology, 35 (154), pp. 923–928 (2012)
5. Sengottuvel, P., Satishkumar, S., Dinakaran, D.: Optimization of multiple characteristics of EDM parameters based on desirability approach and fuzzy modeling. Procedia Engineering 64, pp. 1069–1078 (2013)
6. Shabgard, M., Badamchizadeh, M.G., Ranjbary, K.: Fuzzy approach to select machining parameters in electrical discharge machining (EDM) and ultrasonic-assisted EDM processes. Journal of Manufacturing Systems, 32 (1), pp. 32–39 (2013)
7. Mohd, A., Sarkheyli, A., Mohd-Zain, A., Habibollah, H.: Fuzzy logic for modeling machining process: a review. Artificial Intelligence Review 43(3), pp. 345–349 (2013)
8. Katz, Z., Tibbles J.C.: Analysis of micro-scale ECM Process. The International Journal of Advance Manufacturing Technology, 35 (154), pp. 923–928 (2012)
9. Klocke, F.: Results of surface integrity and fatigue study of PECM and PEO processed γ-TiAl for turbine applications. Procedia CIRP19 (3), pp. 131–136 (2016)
10. Lilong-Baoji, M., Cheng, P., Yun, K.: Effect of power signal waveform on shape accuracy in electrochemical drilling. Materials Science and Engineering, (392), pp. 1–8 (2018)
11. Hung Lea, V., Khanh Tran, D.: Extending fuzzy logics with many hedges. Fuzzy Sets and Systems, 1(1), pp. 45–52 (2018)
12. Keasberry, J.V., Labib, W.A., Atkinson, J., Frost, W.H.: A fuzzy logic control approach to electrochemical machining (ECM) In: Proceedings of the 34th International, Formerly the International Machine Tool Design and Research Conferences, 34, pp. 153–160 (2008)
13. Morteza-Sadegh, A.: Controlling electrochemical machining by using a fuzzy logic approach. Journal of Modern Processes in Manufacturing and Production, 6(2), pp. 61–71 (2017)
14. Gallegos-Miranda, I., Pérez Martínez, J., Granda-Gutiérrez A.: Control de la distancia inter-electrodos en un proceso de micro maquinado electroquímico por pulsos (PECMM). Research in Computing Science 148(8), pp. 2015–2028 (2019)
15. Kohli, A., Wadhwa, A., Virmani, T., Jain, U.: Optimization of Material Removal Rate in Electrical Discharge Machining Using Fuzzy Logic. World Academy of Science. Engineering and Technology International Journal of Mechanical and Mechatronics Engineering, 12, pp. 2839–2844 (2012)
16. Trias, A., Azli, Y., Nor, H., Kamal, K., Ade, E.: Predicting material removal rate of electrical discharge machining (EDM) using artificial neural network for high gap current, In: International conference on electrical, control and computer engineering (2011)
17. Sanahuja, S.: Sistemas de control con lógica difusa: métodos de Mamdani y de Takagi-Sugeno-Kang (TSK). Grado en matemática computacional. pp. 55–60 (2017)
18. Martínez-Alvarado, R., Granda-Gutiérrez, E.E., Zuñiga, L.M., Guerrero, R.: Pulsed power supply for electrochemical machining. In: IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) 3(15), pp. 1–6 (2015)
19. Rajurkar, K.P., Wei, B., Kozak, J., McGeough, J.A.: Modelling and monitoring interelectrode gap in pulse electrochemical machining. In: CIRP Ann. - Manuf. Technol., 44(1), pp. 177–180 (1995)

ISSN 1870-4069

Sobre la automatización del razonamiento y el aprendizaje en lógicas modales

Yensen Limón[1], Everardo Bárcenas[2],
Guillermo Molero-Castillo[2], Rocío Aldeco-Pérez[2]

[1] Universidad Veracruzana,
México
[2] Universidad Nacional Autónoma de México,
México

zs16017367@estudiantes.uv.mx,
{barcenas,gmoleroca,raldeco}@fi-b.unam.mx

Resumen. Las lógicas modales forman una familia de lenguajes formales ampliamente utilizados en diversos dominios de las ciencias de la computación, tales como la verificación de sistemas, la web semántica, y la demostración automática de teoremas. El Aprendizaje Automatizado es una rama de la Inteligencia Artificial que versa sobre el estudio y desarrollo de algoritmos capaces de aprender la solución de problemas a partir de un conjunto de ejemplos. Recientemente, se han publicado estudios sobre la aplicación de estas técnicas en la optimización de los algoritmos de razonamiento en lógicas clásicas de alto orden. Motivados por estos resultados, se propone en este artículo el estudio de técnicas de aprendizaje automatizado en el contexto del razonamiento automatizado en lógicas modales. De manera particular, en este trabajo se presenta un algoritmo de satisfacibilidad para el cálculo μ y se describe como un algoritmo de aprendizaje por refuerzo puede ayudar en el desempeño del algoritmo de satisfacibilidad.

Palabras clave: Razonamiento automatizado, aprendizaje automatizado, lógicas modales.

On Automated Reasoning and Machine Learning in Modal Logics

Abstract. Modal logics are a family of formal languages widely used in many domains of computer science. Machine Learning is a branch of Artificial Intelligence that deals with the study and development of algorithms capable of learning problem solving from a set of examples. Recent studies have been published on the application of these techniques in the optimization of reasoning algorithms in classical higher-order logics. Motivated with these results, the study of machine learning techniques in the context of automated reasoning in modal logics is proposed in this

article. In particular, in this work we describe a satisfiability algorithm for the μ-calculus and describes how a reinforcement learning algorithm can help in the performance of the satisfiability algorithm.

Keywords: Automated reasoning, machine learning, modal logics.

1. Introducción

Las lógicas modales (ML), cuyo origen se remonta a la lógica aristotélica [4], forman una familia de lenguajes formales que incluyen la Lógica Temporal Lineal (LTL), la Lógica Temporal Ramificada (CTL), la Lógica Dinámica Proposicional (PDL), el Cálculo μ, las Lógicas Descriptivas, por mencionar algunos. Además del campo de la filosofía, las ML han encontrado un dominio de estudio muy amplio, en campos como las matemáticas, las humanidades y las ciencias computacionales. Dentro de este último ámbito, un tema largo y exitosamente estudiado es el desarrollo de métodos automáticos de razonamiento. El estudio de estos métodos, en particular algoritmos, es el principal tema de interés de este proyecto [13].

El Teorema de Correspondencia de van Benthem (1977) versa sobre la caracterización de la Lógica Modal K en términos del fragmento invariante bajo bisimulación de la Lógica Clásica de Predicados [3]. Caracterizaciones similares fueron posteriormente establecidas entre otras ML y contrapartes clásicas, por ejemplo, entre el Cálculo μ y la Lógica Monádica de Segundo Orden [9]. Estas caracterizaciones fundamentan lo que en la práctica ha resultado en el particular balance entre expresividad y complejidad computacional de los algoritmos de razonamiento asociados a las ML. Abundantes problemas en diversas áreas, tales como la Verificación Formal de Sistemas (*software* y *hardware*), Bioinformática, y Demostración Automática de Teoremas, por mencionar algunas, han sido modelados y solucionados de forma automática gracias al citado balance de las ML [3]. En este proyecto, estamos interesados en el estudio de técnicas de Aprendizaje Automatizado (*machine learning*) que puedan ser utilizadas en la optimización de algoritmos de razonamiento para ML.

El Aprendizaje Automatizado (AA) es una rama de la Inteligencia Artificial ampliamente estudiada en la actualidad y con muy variadas aplicaciones tales como en Visión Computacional, Robótica, Ciencia de Datos, etc [1]. El AA se basa, *grosso modo*, en el estudio y desarrollo de algoritmos capaces de aprender la solución de problemas a partir de un conjunto de ejemplos. Recientemente, se han publicado estudios sobre la aplicación de estas técnicas en la optimización de los algoritmos de razonamiento en lógicas clásicas de alto orden [1]. El conjunto de ejemplos estudiados en este caso han sido demostraciones matemáticas. Motivados por estos resultados, proponemos en este trabajo el estudio de técnicas de AA en el contexto del razonamiento automatizado en ML.

2. Motivación y antecedentes

En esta sección se describen los trabajos relacionados y se justifica la propuesta de este trabajo.

2.1. Demostración automática de teoremas y aprendizaje automatizado

En [1] se describe un demostrador automático de teoremas, basado en lógica de alto orden (HOL Light), optimizado con técnicas de aprendizaje profundo. Este demostrador se utilizó para demostrar la conjetura de Kepler. Existen diversos trabajos relacionados, tal como la herramienta TitacToe, basada en un algoritmo A* y lógica de alto orden (HOL4), pero la diferencia es que no emplea aprendizaje profundo [7]. Otro trabajo similar se reporta en [8], en el cual se describe la herramienta Gamepad. Esta herramienta también está basada en Hol Light, sin embargo, el conjunto de entrenamiento solo cuenta con 1602 teoremas y lemas. Por otra parte, en la herramienta reportada en [1], se consideran 2199 lemas y 29462 teoremas como conjunto de aprendizaje.

Adicionalmente, el entrenamiento puede considerar pruebas humanas. Estas pruebas se registran y son utilizadas como ejemplos de entrenamiento. Cuando es una demostración exitosa HOL Light registra el resultado de las aplicaciones tácticas utilizadas y el teorema para ser utilizado en un futuro. En este trabajo se propone aplicar técnicas de aprendizaje automático en algoritmos de razonamiento basados en lógicas modales. Esto implica una extensión en el dominio de aplicación de este tipo de algoritmos, tales como la verificación de sistemas.

2.2. Satisfacibilidad

Una expresión o fórmula bien formada en un sistema se dice satisfacible si y sólo si existe un modelo para dicha fórmula, es decir, si existe una interpretación que hace verdadera a la fórmula. Se dice que una fórmula es válida si y sólo es verdadera en cualquier interpretación. Nótese que el problema de validez es equivalente al de satisfacibilidad: una fórmula es válida, si y solo si, la negación de la fórmula no es satisfacible.

En el contexto de las lógicas modales un modelo puede entenderse como una estructura de Kripke [3]. Una estructura de Kripke se define como una tripleta $K = \langle D, R, L \rangle$ donde D es un conjunto no-vacío (usualmente de mundos posibles o estados), $R \subseteq D^2$ es una relación entre los miembros de D (usualmente conocida como relación de acceso) y L es una función que asigna valores a las variables del lenguaje base. Las familias particulares de relaciones en R están asociadas a ciertas lógicas, por ejemplo, en CTL las relaciones en R forman estructuras de árbol. Los algoritmos de satisfacibilidad tipo Fischer-Ladner están basado en la construcción explicita de modelos [6]. En los modelos construidos por estos algoritmos los elementos del dominio están definidos como conjuntos de subfórmulas (de la fórmula a la cual se le quiere determinar su satisfacibilidad o insatisfacibilidad).

Las variables proposicionales en estos conjuntos representan las asignaciones por la función L (de la estructura de Kripke), y las fórmulas modales representan la topología del modelo, es decir, si en un elemento del dominio está la fórmula $\langle m \rangle p$, entonces debe existir otro elemento del dominio, conectado a través de m, que contenga p.

A menudo, la complejidad computacional de los algoritmos Fischer-Ladner está determinada por la cantidad elementos del dominio candidatos a formar un modelo [14]. Por ejemplo, para el cálculo μ existe una cantidad exponencial, con respecto al tamaño de la fórmula inicial, de elementos del dominio candidatos a formar un modelo (recuerde que se definen como conjuntos de subfórmula inicial) [13]. Como la construcción de modelos implica una búsqueda en esta cantidad exponencial de elementos del dominio, la complejidad computacional del cálculo μ está en EXPTIME-completa [14,13]. Se propone en este trabajo que un algoritmo de satisfacifiblidad tipo Fischer-Ladner que elija los elementos del dominio a partir de un conjunto de modelos de aprendizaje. Esto con el objeto de optimizar el desempeño del algoritmos ya conocidos [14,13].

2.3. Verificación de sistemas

Un ejemplo sobre la aplicación en la automatización del razonamiento utilizando lógicas modales es la verificación de sistemas conscientes de contexto [11,12]. Este tipo de sistemas son capaces de sensar su contexto y reaccionar de acuerdo a la información obtenida. La gran cantidad de variables de contexto hacen que el desarrollo y mantenimiento de este tipo de sistemas sean costosos y proclives a errores.

En [10] se introduce una noción de consistencia en sistemas conscientes de contexto. La verificación de consistencia se basa en un algoritmo de satisfacibilidad tipo Fischer-Ladner del cálculo μ. Además, se describe en [11] la aplicación de este algoritmo en la verificación de un sistema de comunicación consciente de contexto. Este tipo de sistemas están compuestos por un conjunto de usuarios, ubicados en lugares, con diversas restricciones de comunicación (por ejemplo, sin señal de teléfono móvil), de acuerdo a un horario. Los canales de comunicación pueden ser síncronos (teléfono móvil) o asíncronos (correo electrónico). Este tipo de sistemas se definen consistentes en caso de que los usuarios se puedan comunicar siempre entre sí, sin importar las restricciones descritas.

Otro ejemplo de verificación de sistemas conscientes de contexto es el que se describe en [12]. En este artículo se presenta un ambiente de aprendizaje inteligente, el cual está compuesto por un conjunto de actividades de aprendizaje a realizarse en diversas ubicaciones equipadas con posiblemente distintos dispositivos (computadoras, proyectores, sensores, etc.). A través de diversos sensores el sistema es capaz de identificar a usuarios desatentos a las actividades de aprendizaje, y de acuerdo a las restricciones antes descritas, aunadas a las restricciones de aprendizaje de los estudiantes, el sistema interviene para atraer la atención del estudiante (videos, textos, avisos al instructor, etc.). El sistema se define consistente cuando es capaz de realizar una intervención sin importar las restricciones antes descritas.

La verificación de consistencia de estos sistemas también se basa en un algoritmo de satisfacibilidad tipo Fischer-Ladner del cálculo μ.

En ambos trabajos [11,12] también se describen diversos experimentos. Sin embargo, debido a la complejidad exponencial computacional del algoritmo de satisfacibilidad, los casos de experimentación fueron modestos. Se conjetura que una optimización del algoritmo de satisfacibilidad con técnicas de aprendizaje automatizado puede conllevar a experimentos más realistas.

2.4. Razonamiento sobre consultas

Dentro del análisis estático de lenguajes de programación, en particular, los lenguajes de consultas para bases de datos, existen diversos problemas de razonamiento. El problema de vacuidad de consultas consiste en determinar si el resultado de cierta consulta es siempre vacío, sin importar la base de datos. Determinar si el resultado de cierta consulta siempre se encuentra dentro del resultado de otra consulta, sin importar la base de datos, se le conoce como el problema de contención. La equivalencia de dos consultas se define en términos de contención: dos consultas son equivalentes, si y solo si, están contenidas entre sí. En el contexto de datos semi-estructurados, más concretamente, del lenguaje de consultas *XPath*, estos problemas de razonamiento son modelados en términos de satisfacibilidad en el cálculo μ en [14,13]. El algoritmo es tipo Fischer-Ladner. Diversos experimentos fueron también descritos en estos artículos. Motivados por experimentos más extensos, conjeturamos que optimizaciones, basadas en aprendizaje automatizado, al algoritmo permitirán su aplicación en escenarios más complejos.

3. Cálculo μ en árboles

En esta sección, se presenta el cálculo μ. Las fórmulas son interpretadas sobre árboles finitos. El alfabeto es considerado por dos conjuntos, $PROP$ y MOD, donde $PROP$ es un conjunto de letras proposicionales y $MOD = \{1, 2, 3, 4\}$ es un conjunto de modalidades.

Definición 1 (Sintaxis) *El conjunto de fórmulas del cálculo μ es definido por la siguiente gramática: $\varphi ::= p \mid X \mid \neg\varphi \mid \varphi \vee \psi \mid \langle m\rangle\varphi \mid \mu X.\varphi$. donde p es una proposición, m es una modalidad, y X es una variable.*

Se asume que las variables únicamente ocurren sobre el alcance de una modalidad. Las fórmulas son interpretadas como un subconjunto de nodos en árboles. Las proposiciones son utilizadas para etiquetar los nodos, la negación ($\neg$) es interpretada como el conjunto complemento, la disyunciones son interpretadas como la unión de conjuntos. En el caso de $\phi \wedge \varphi$ se escribe $\neg(\neg\phi \vee \neg\varphi)$. Las fórmulas modales $\langle m\rangle\phi$ se mantienen en nodos donde hay un nodo adyacente por m que soporta ϕ. El punto menos fijo es intuitivamente interpretado como un operador de recursión. Ahora se presenta la definición de las estructuras estilo Kripke.

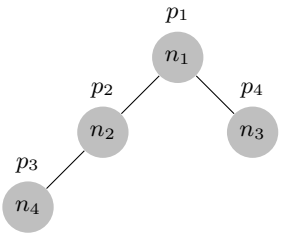


Fig. 1. Ejemplo de un modelo de árbol, donde $\phi = \mu X.(p_3 \vee \langle 1 \rangle X)$, entonces $n_1 = \{\phi, \langle 1 \rangle p_2, p_1\}$, $n_2 = \{\phi, p_2\}$, $n_3 = \{p_4\}$ y $n_4 = \{\phi, p_3\}$.

Definición 2 (Estructura de árbol) *Una estructura de árbol, o simple árbol, es una tupla* $\langle N, R^m, L \rangle$ *donde:*

- N *es el conjunto de nodos;*
- R^m *es una familia binaria de relaciones de nodos* $(N \times N)$ *formando una estructura de árbol; y*
- L *es una función de etiquetado* $L : N \to 2^{PROP}$.

Ahora se da una representación formal de la semántica de las fórmulas.

Definición 3 (Semántica) *Considere un árbol* K *y una valuación* $V : Var \to 2^N$, *donde* Var *es un conjunto de variables. La semántica de la fórmula es definida de la siguiente manera:*

$$
\begin{aligned}
[\![p]\!]_V^K &= \{n \mid p \in L(n)\} &,\\
[\![\neg\varphi]\!]_V^K &= N \backslash [\![\varphi]\!]_V^K &,\\
[\![\varphi \vee \psi]\!]_V^K &= [\![\varphi]\!]_V^K \cup [\![\psi]\!]_V^K &,\\
[\![\langle m \rangle \varphi]\!]_V^K &= \{n \mid R(n,m) \cap [\![\varphi]\!]_V^K \neq \emptyset\} &,\\
[\![X]\!]_V^K &= V(X) &,\\
[\![\mu X.\varphi]\!]_V^K &= \bigcap \{N' \mid [\![\varphi]\!]_{V[^{\mu X.\varphi}/_X]}^K \subseteq N'\} &.
\end{aligned}
$$

Una fórmula ϕ es satisfacible, si y solo si, existe un árbol (modelo), tal que la interpretación de ϕ sobre el árbol no sea vacía, esto es $[\![\phi]\!]_V^K \neq \emptyset$.

Ejemplo 1 *En la Figura 1, existe una representación gráfica del modelo de árbol. Considere por ejemplo la siguiente fórmula:* $\langle 1 \rangle p_2 \wedge p_1$. *Esta fórmula selecciona los nodos nombrados* p_1 *con un hijo llamado* p_2. *En la Figura 1, esta fórmula es soportada en* n_1. *La navegación recursiva puede ser expresado con el punto fijo. Esta fórmula* $\mu X.(p_3 \vee \langle 1 \rangle X) \wedge \langle 1 \rangle p_2 \wedge p_1$ *mantiene en un nodo a* p_1 *con un descendiente* p_3 *en el nodo* n_4 *y un hijo* p_2 *en el nodo* n_2. *En la Figura 1, la fórmula es soportada por el nodo* n_1.

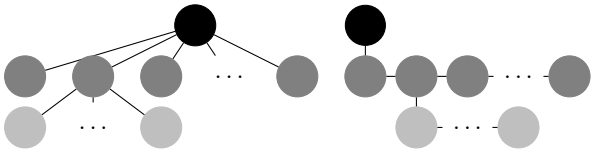


Fig. 2. Ejemplo de transformación de árbol de n-aridad a binario [2].

4. Satisfacibilidad y aprendizaje por refuerzo

El algoritmo de satisfacibilidad construye una versión sintáctica de una estructura de árbol, llamado árbol Fischer-Ladner [6], donde los nodos son conjuntos de subfórmulas.Ahora se define la noción de árboles. Es conocida la biyección entre árboles de n-aridad y binarios: Una relación es para el primer hijo(padre), y la otra para el siguiente hermano (previo), En la Figura 2, existe una representación gráfica de esta biyección. Por lo tanto, sin perder la generalidad, por conveniencia técnica, se trabaja con árboles binarios. Además, las fórmulas modales deben ser reinterpretadas:

- $\langle 1\rangle\varphi$ se mantiene en nodos donde su primer hijo es soportado por φ;
- $\langle 2\rangle\varphi$ es interpretada en nodos donde tiene en su siguiente hijo (derecha) a φ;
- $\langle 3\rangle\varphi$ se encuentra donde φ es el padre; y
- $\langle 4\rangle\varphi$ donde φ es el hermano previo (derecha).

Ahora consideramos fórmulas únicamente en forma normal negada. La negación solo ocurre en frente de proposiciones y fórmulas modalidades $\langle m\rangle\top$, donde para $\top$ se presenta por $p \vee \neg p$. La función $nnf(\phi)$ se define a continuación:

$$\begin{array}{rr} nnf(X) = \neg X & nnf(\varphi \vee \psi) = nnf(\varphi) \wedge nnf(\psi), \\ nnf(p) = \neg p & nnf(\mu X.\varphi) = \mu x.nnf(\varphi)\left[{}^{X}/_{\neg X}\right], \\ & nnf(\langle m\rangle\varphi) = \langle m\rangle nnf(\varphi) \vee \neg\langle m\rangle\top. \end{array}$$

La fórmula en forma normal negada se define como la fórmula resultante de remplazar las negaciones $\neg\varphi$ con $nnf(\varphi)$. Algunas nociones son requeridas antes de introducir los árboles Fischer-Ladner trees.

Definición 4 *Se define una relación binaria R^{FL} en fórmulas con $i = 1, 2$:*

$$\begin{array}{lll} R^{FL}(\varphi, nnf(\varphi)) & R^{FL}(\varphi_1 \wedge \varphi_2, \varphi_i) & R^{FL}(\varphi_1 \vee \varphi_2, \varphi_i), \\ R^{FL}(\langle m\rangle\varphi, \varphi) & R^{FL}(\mu X.\varphi, \varphi[{}^{\mu x.\varphi}/_{X}]). & \end{array}$$

Definición 5 (Fischer-Ladner Closure) *Dada una fórmula φ, Fischer-Ladner closure de φ es definida como $FL(\varphi) = FL(\varphi)_k$, Donde k es el más pequeño entero positivo que satisface $FL(\varphi)_k = FL(\varphi)_{k+1}$, donde:*

$$\begin{array}{l} FL(\varphi)_0 = \{\varphi\}, \\ FL(\varphi)_{i+1} = FL(\varphi)_i \cup \{\psi' \mid R^{FL}(\psi, \psi'), \psi \in FL(\varphi)_i\}. \end{array}$$

A continuación, se define el conjunto lean para los nodos sintácticos. Este conjunto está compuesto intuitivamente por las proposiciones y subfórmulas modales obtenidas de la fórmula de entrada. Las proposiciones son utilizadas para nombrar los nodos y las subfórmulas modales dan la topología de la información sobre los árboles candidatos.

Definición 6 (Lean) *Dada una fórmula φ y una proposición p' no ocurrente en φ, el lean de φ es definido de la siguiente manera para todas $m \in MOD$:*

$$lean(\varphi) = \{p, \langle m\rangle\varphi \in FL(\varphi)\} \cup \{p', \langle m\rangle\top\}$$

.

Ejemplo 2 *Considere la siguiente fórmula: $\varphi = \langle 1\rangle\langle 1\rangle\neg p_1 \wedge \langle 1\rangle\neg p_1 \wedge \neg p_1 \wedge \mu X.(p_1 \vee \langle 1\rangle X) \wedge \langle 2\rangle p_2 \wedge p_3$. El lean de φ es definido como:*

$$lean(\varphi) = \{p_1, p_2, p_3, \langle 1\rangle\neg p_1, \langle 1\rangle\langle 1\rangle\neg p_1, \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X), p', \langle 2\rangle p_2, \langle 1\rangle\top, \langle 2\rangle\top, \langle 3\rangle\top, \langle 4\rangle\top\}.$$

Ahora se definen los nodos en árboles Fischer-Ladner.

Definición 7 (Nodos) *Dada una fórmula φ, un nodo en un árbol Fischer-Ladner es definido como un subconjunto de $lean(\varphi)$, es decir;*

- *contiene al menos una proposición;*
- *si $\langle m\rangle\psi$ ocurre en el nodo, también $\langle m\rangle\top$; y*
- *$\langle 3\rangle\top$ y $\langle 4\rangle\top$ ambas no pueden ocurrir.*

Ejemplo 3 *Considerando la misma fórmula y el correspondiente lean presentado en el Ejemplo 2. Entonces se determinan los siguientes nodos:*

- $n_1 : \{p_3, \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X), \langle 1\rangle\neg p_1, \langle 1\rangle\langle 1\rangle\neg p_1, \langle 2\rangle p_2\}$;
- $n_2 : \{p', \langle 1\rangle\neg p_1, \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X)\}$;
- $n_3 : \{p', \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X)\}$;
- $n_4 : \{p_1\}$;
- $n_5 : \{p_2\}$.

Finalmente definimos los árboles Fishcer-Ladner.

Definición 8 (Árbol Fischer-Ladner) *Intuitivamente se definen los árboles Fischer-Lader como:*

- *$\emptyset$ representa un árbol Fischer-Ladner vacío ; y*
- *(n, X_1, X_2) es un árbol Fischer-Ladner, dado que n es un nodo (nodo raíz), , X_1 y X_2 son árboles Fischer-Ladner.*

Para aclarar el contexto, nos referiremos como simplemente árbol en lugar de árbol Fischer-Ladner.

Ejemplo 4 *Considere los nodos en el Ejemplo 3. Entonces definimos el árbol como:* $T = (n_1, (n_2, (n_3, (n_4, \emptyset, \emptyset), \emptyset), \emptyset), (n_5, \emptyset, \emptyset))$. *En la Figura 3 se presenta el árbol.*

De esta forma se describe el algoritmo de satisfacibilidad del cálculo μ basado en una búsqueda primero en profundidad. El Algoritmo 1 decide si una fórmula es satisfacible o insatisfacible. Dentro del algoritmo se cuenta con una función *main* la cual es encargada de crear todos los nodos que se utilizarán. Después, los nodos son iterados y busca el nodo que satisfaga la fórmula. Como consecuencia inicia una construcción del árbol para satisfacer la fórmula. El nodo seleccionado es utilizado como candidato y es a partir de él donde comienza la búsqueda. Posteriormente, el siguiente paso del algoritmo es entrar a la función *search*. En el Algoritmo 1 la función *search* presenta diversos casos, cuando la fórmula tiene la forma de $p, \top, \neg p$ o $\neg\langle m\rangle\top$, entonces, el algoritmo evalúa la satisfacibilidad del nodo retornando *true*. Si esta contiene puntos fijos, entonces, el algoritmo realiza la expansión de la fórmula llamando a la función *search* nuevamente. Los casos para la disyunción y conjunción el algoritmo invoca la función *search* para ambos. Cuando las fórmulas tienen la forma de modalidad, este busca el siguiente nodo. La fórmula puede contener modalidades $\langle 1\rangle$, $\langle 2\rangle$, $\langle 3\rangle$ o $\langle 4\rangle$. Entonces el nodo candidato es eliminado de la lista de nodos, el algoritmo obtiene la lista de los posibles nodos siguientes con la función *next*, posteriormente la lista de nodos es iterada y el árbol es actualizado. Después es llamada la función *search*, si este retorna $false$, entonces, el nodo siguiente es removido del árbol. Cada vez que un nodo es agregado, este pasa a un estatus como utilizado y no puede ser reutilizado en niveles subsecuentes del árbol, esto es con el fin de no construir árboles infinitos. En el caso que los nodos se terminen en la búsqueda, entonces el algoritmo retorna $false$. El algoritmo termina cuando ha viajado por todos los nodos o el árbol es construido.

La satisfacibilidad en fórmulas se define a continuación.

Definición 9 *Dada una fórmula* ϕ *y un nodo* n*, se define la satisfacibilidad* $n \vdash \phi$ *como:*

$$\frac{}{n \vdash \top} \qquad \frac{\varphi \in n}{n \vdash \varphi} \qquad \frac{\varphi \notin n}{n \vdash \neg\varphi} \qquad \frac{n \vdash \psi}{n \vdash \varphi \vee \psi} \qquad \frac{n \vdash \varphi}{n \vdash \varphi \vee \psi}$$

$$\frac{n \vdash \varphi \quad n \vdash \psi}{n \vdash \varphi \wedge \psi} \qquad\qquad \frac{n \vdash \varphi[{}^{\mu x.\varphi}/_{x}]}{n \vdash \mu x.\varphi}$$

La función Δ_m es responsable de verificar la unión de un nodo respecto a otro nodo a través de una modalidad m. Cuando n' es el nodo posible a unir, n_1 es el nodo actual, m son las modalidades $\langle 1\rangle$ o $\langle 2\rangle$, $\bar{m}$ son las modalidades inversas $\langle 3\rangle$ o $\langle 4\rangle$ y $\langle m\rangle\varphi$ una fórmula modal.

Definición 10 *Dado dos nodos* n*,* n' *y una fórmula* φ*, esta es determinada por aquellos nodos que son consistentes con respecto a la fórmula* $\Delta_m(n, n')$ *donde* $m \in \{1, 2, 3, 4\}$*, si y solo si, las fórmulas modales* $\langle m\rangle\varphi_1, \langle\bar{m}\rangle\varphi_2 \in lean(\varphi)$*, esto se determina de la siguiente manera:*

Algoritmo 1 Algoritmo de satisfacibilidad del cálculo μ con una búsqueda primero en profundidad sobre árboles.

```
function MAIN(φ)
    Y ← N^φ
    T ← ∅
    for all n_i of Y do
        if n_i ⊢ φ then
            T ← (n_i, ∅, ∅)
            if search(n_i, n_i, φ) then
                return true
            else
                T ← ∅
            end if
        end if
    end for
    return false
end function
```

- $\forall\langle m\rangle\varphi \in lean : \langle m\rangle\varphi \in n \Leftrightarrow n' \vdash \varphi$
- $\forall\langle \bar{m}\rangle\varphi \in lean : \langle \bar{m}\rangle\varphi \in n' \Leftrightarrow n \vdash \varphi$

Ejemplo 5 *Dada la fórmula* $\varphi = \langle 1\rangle\langle 1\rangle\neg p_1 \wedge \langle 1\rangle\neg p_1 \wedge\neg p_1\wedge \mu X.(p_1 \vee \langle 1\rangle X) \wedge \langle 2\rangle p_2\wedge p_3$ *reutilizando el Ejemplo 2 la unión es la siguiente:* $\Delta_m(n_1, n_2)$, $\Delta_m(n_1, n_5)$, $\Delta_m(n_2, n_3)$ *y* $\Delta_m(n_3, n_4)$. *Donde* $n_1 = \{p', \langle 1\rangle\mu X.(p_1\vee\langle 1\rangle X), \langle 1\rangle\neg p_1, \langle 1\rangle\langle 1\rangle\neg p_1, \langle 2\rangle p_2, p_3\}$, $n_2 = \{p', \langle 1\rangle\neg p_1, \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X)\}$, $n_3 = \{p', \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X)\}$, $n_4 = \{p_1\}$ *y* $n_5 = \{p_2\}$. *Esos nodos son consistentes debido a que los nodos hijos contiene a los testigos de las fórmulas modales.*

Definición 11 (nodes) *Dado un árbol* T, *la función nodes es definida como:*

- $nodes(\emptyset) = \{\}$.
- $nodes((n, T_1, T_2)) = \{n\} \cup nodes(T_1) \cup nodes(T_2)$.

Un nodo T_n es denotado como:

- T_n es un $n \in nodes(T)$.

Definición 12 (Subárbol) *El subárbol* T' *de un árbol dado* T ($T' \subseteq T$) *es definido como:*

- $T \subseteq T$
- $T \subseteq (n, T_1, T_2)$ *si* $T \subseteq T_1$ *o* $T \subseteq T_2$

Definición 13 (root) *La función* $root(T)$ *toma el árbol como entrada y retorna el nodo raíz del árbol.*

- $root(\emptyset) = \emptyset$.

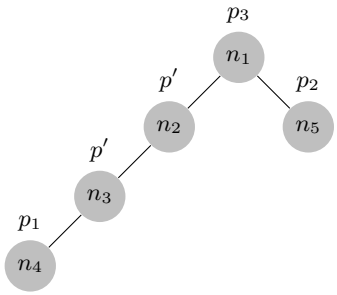


Fig. 3. Modelo de Árbol Fischer-Ladner para $\varphi = \langle 1\rangle\langle 1\rangle\neg p_1 \wedge \mu X.(p_1 \vee \langle 1\rangle X) \wedge \langle 1\rangle\neg p_1 \wedge \neg p_1 \wedge \langle 2\rangle p_2 \wedge p_3$, donde $n_1 = \{p_3, \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X), \langle 1\rangle\neg p_1, \langle 1\rangle\langle 1\rangle\neg p_1, \langle 2\rangle p_2\}$, $n_2 = \{p', \langle 1\rangle\neg p_1, \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X)\}$, $n_3 = \{p', \langle 1\rangle\mu X.(p_1 \vee \langle 1\rangle X)\}$, $n_4 = \{p_1\}$ y $n_5 = \{p_2\}$

- $root((n, T_1, T_2)) = n$.

Definición 14 (neighborhood) *Dado un árbol T y un nodo n, la función neighborhood obtiene los nodos adyacentes. Considerando que existen los árboles T', T_1 y T_2.*

- $parent(n, T) = n'$ *tal que,* $(n', T_n, T') \subseteq T$.
- $child(n, T) = n'$ *tal que,* $(n, (n', T_1, T_2), T') \subseteq T$.
- $ps(n, T) = n'$ *tal que,* $(n', T', T_n) \subseteq T$.
- $fs(n, T) = n'$ *tal que,* $(n, T', (n', T_1, T_2)) \subseteq T$.
- $neighborhood(n, T) = \{parent(n, T), child(n, T), ps(n, T), fs(n, T)\}$.

Definición 15 (delete) *Dado un árbol T y un nodo n que es eliminado del árbol, la función delete se define como:*

- $delete(\emptyset, n) = \emptyset$.
- $delete((n, T_1, T_2), n) = (\emptyset, T_1, T_2)$.
- $delete((n', T_1, T_2), n) = (n', delete(T_1, n), delete(T_2, n))$.

donde $n \neq n'$.

Definición 16 (update) *Dado un árbol T, dos nodos diferentes n y n' y una modalidad m, la fórmula update es definida como:*

- *cuando* $n' \notin nodes(T)$.
  - $update((n, T_1, T_2), n, n', 3) = (n', (n, T_1, T_2), \emptyset)$.
  - $update((n, T_1, T_2), n, n', 4) = (n', \emptyset, (n, T_1, T_2))$.
  - $update((n, \emptyset, T_2), n, n', 1) = (n, (n', \emptyset, \emptyset), T_2)$.
  - $update((n, T_1, \emptyset), n, n', 2) = (n, T_1, (n', \emptyset, \emptyset))$.
  - $update((n'', T_1, T_2), n, n', m) = (n''$, $update(T_1, n, n', m)$, $update(T_2, n, n', m))$.
  - $update(\emptyset, n, n', m) = \emptyset$.
- *cuando* $n' \in nodes(T)$.

– $update(T, n, n', m) = T$.

La función *next* obtiene el conjunto de nodos subsecuentes para identificar un nodo candidato. Esta función es definida por una tripleta de elementos $(n, \langle m \rangle \varphi, Y)$. Donde n es el nodo actual, $\langle m \rangle \varphi$ es una fórmula modal, Y es el conjunto de nodos no utilizados y T es el árbol.

Definición 17 (next) *Dado un nodo n, una fórmula modal $\langle m \rangle \phi$, un conjunto de nodos no utilizados Y, y un árbol T se define la función next para determinar la unión de los nodos que sean considerados modalmente consistente con respecto a la fórmula. Esta declaración es denotada por $\Delta_m(n, n')$, si y solo si, para todas las fórmula $\langle m \rangle \phi$ en n y $\langle \bar{m} \rangle \phi$ en n', determinadas por la dirección m. La función next es definida como:*

– $next(n, \langle m \rangle \varphi, Y, T) = \{n' \in Y \cup neighborhood(n, T) \mid \Delta_m(n, n')\}$, *donde* $m = 1, 2, 3, 4$.

Como puede observarse del algoritmo. Los nodos no son generados previamente a la construcción del modelo, si no que se generan al vuelo a través de la función *next*. Sin embargo esta función aún tiene que buscar en el conjunto completo de posibles nodos. A continuación describimos como a través de aprendizaje por refuerzo podemos determinar cuáles son los nodos *siguientes*. Para esto primero definimos que es un proceso de decisión de Markov.

Definición 18 (Proceso de Decisión de Markov) *Un Proceso de Decisión de Markov se define como un tupla (S, A, P, R), donde*

– *S es un conjunto de estados;*
– *A es un conjunto de acciones;*
– *P es una función de transición probabilistica entre los estados; y*
– *R es una función de recompensas de transición.*

Un proceso de decisión de Markov es construido a partir de un conjunto de entrenamiento. Las transiciones entre los estados se definen en función de la probabilidad de transición aprendida [5]. En el contexto del algoritmo de satisfacibilidad, el conjunto de aprendizaje está compuesto por modelos ya construidos, como los resultantes de la experimentación en [12,13,14].

5. Conclusiones

Las lógicas modales representan una familia de lenguajes formales con aplicación en un amplio dominio de campos en las ciencias de la computación, por ejemplo, en la verificación de sistemas. En este trabajo se describe un algoritmo de satisfacibilidad para una de las lógicas modales decidibles con mayor expresividad, el cálculo μ. Se propone además el uso de métodos de aprendizaje automatizado para optimizar el desempeño del algoritmo. De manera más específica, se propone la aplicación de aprendizaje por refuerzo en el proceso de

Algoritmo 1 Función Search del algoritmo de satisfacibilidad

```
function SEARCH(n_0, n_1, φ)
    if φ = p then
        if p ∈ n_1 then
            return true
        end if
    end if
    if φ = ¬p then
        if p ∉ n_1 then
            return true
        end if
    end if
    if φ = ¬⟨m⟩⊤ then
        if ⟨m⟩⊤ ∉ n_1 then
            return true
        end if
    end if
    if φ = ⊤ then
        return true
    end if
    if φ = φ_1 ∧ φ_2 then
        if search(n_0, n_1, φ_1) and
            search(n_0, n_1, φ_2) then
            return true
        end if
    end if
    if φ = φ_1 ∨ φ_2 then
        if search(n_0, n_1, φ_1) or
            search(n_0, n_1, φ_2) then
            return true
        end if
    end if
    if φ = ⟨m⟩φ then
        Y ← Y\{n_1}
        X ← next(n_1, ⟨m⟩φ, Y ∪ {n_0}, T)
        for all n_j of X do
            T ← update(T, n_1, n_j, m)
            if search(n_1, n_j, φ) then
                return true
            else
                T ← delete(T, n_j)
            end if
        end for
    end if
    if φ = μx.φ then
        return search(n_0, n_1, φ[μx.φ/x])
    end if
    return false
end function
```

búsqueda para la construcciones de modelos del cálculo μ. La implementación de esta propuesta es nuestro principal interés como trabajo a futuro.

Es también de nuestro interés la aplicación de esta propuesta en la verificación de sistemas [11].

Agradecimientos. Esta obra se financió con recursos del proyecto IA105420 del Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM.

Referencias

1. Bansal, K., Loos, S.M., Rabe, M.N., Szegedy, C., Wilcox, S.: Holist: An environment for machine learning of higher-order theorem proving (extended version). CoRR (2019)
2. Bárcenas, E., Lavalle, J.: Global numerical constraints on trees. Logical Methods in Computer Science 10(2), 1–28 (2014)
3. Blackburn, P., van Benthem, J.F., Wolter, F.: Handbook of modal logic, vol. 3. Elsevier (2006)
4. Castro Manzano, J.M., Medina Delgadillo, J.: Pons asinorum: una versión terminística. In: XVII Congreso latinoamericano de filosofía medieval, La edad media desde América latina: Aportes a la tradición. pp. 163–179 (2019)
5. Doberkat, E.: Stochastic relations. Foundations for Markov transition systems (01 2007)
6. Fischer, M.J., Ladner, R.E.: Propositional modal logic of programs (extended abstract). In: Hopcroft, J.E., Friedman, E.P., Harrison, M.A. (eds.) ACM Symposium on Theory of Computing. pp. 286–294. ACM (1977)
7. Gauthier, T., Kaliszyk, C., Urban, J.: TacticToe: Learning to reason with HOL4 tactics. arXiv:1804.00595 (2018)
8. Huang, D., Dhariwal, P., Song, D., Sutskever, I.: Gamepad: A learning environment for theorem proving (2018)
9. Janin, D., Walukiewicz, I.: On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In: Montanari, U., Sassone, V. (eds.) CONCUR. Lecture Notes in Computer Science, vol. 1119, pp. 263–277. Springer (1996)
10. Limón, Y., Bárcenas, E., Benítez-Guerrero, E.: Reasoning in context-aware systems with modal logics. Research in Computing Science 133, 51–61 (2017)
11. Limón, Y., Bárcenas, E., Benítez-Guerrero, E., Molero, G.: On the consistency of context-aware systems. Journal of Intelligent and Fuzzy Systems 34(5), 3373–3383 (2018)
12. Limón, Y., Bárcenas, E., Benítez-Guerrero, E., Molero-Castillo, G., Velázquez-Mena, A.: A satisfiability algorithm for the mu-calculus for trees with presburger constraints. In: 2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT) (2019)
13. Limón, Y., Bárcenas, E., Benítez-Guerrero, E., Molero-Castillo, G., Velázquez-Mena, A.: Mu-calculus satisfiability with arithmetic constraints (in press). Programming and Computer Software, PCS (2020)
14. Limón, Y., Bárcenas, E., Benítez-Guerrero, E., Nieto, M.A.M.: Depth-first reasoning on trees. Computación y Sistemas 22(1), 189–201 (2018)

ISSN 1870-4069

Detección de noticias falsas en sitios web y redes sociales: Una investigación del estado del arte

Sergio Damián, Alexander Gelbukh, Hiram Calvo

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

b190394@sagitario.cic.ipn.mx, gelbukh@gelbukh.com,
hcalvo@cic.ipn.mx.

Resumen. La divulgación de noticias falsas en redes sociales ha provocado grandes impactos en la sociedad durante los últimos años, por lo que es importante evitar su propagación. En la actualidad, existen diferentes formas de atacar este problema, las cuales parten por la detección de estas noticias e incluso perfilar a los usuarios que tienden a divulgar este tipo de noticias, ya sea de forma intencional o no. El presente artículo discute las características que posee una noticia para ser catalogada como "falsa", algunas propuestas de solución en el estado del arte, para finalmente concluir sobre los resultados obtenidos y sugerir posible trabajo a futuro en el área. Todo esto con un enfoque en las redes sociales y sitios de internet dedicados a la divulgación de noticias.

Palabras clave: Noticias falsas, aprendizaje automático, desinformación, detección, clasificación

Fake News Detection in Web Sites and Social Media: A Survey of the State of the Art

Abstract. Fake news spreading in social media has caused a big impact in society during the last years. Nowadays, there are different ways to attack this problem, some of them start by detecting fake news and other ones by profiling the users who tend to spread this kind of news, either intentionally or not. The present paper discusses the features and characteristics that can represent what fake news are, some solution proposals in the state of the art, and finally give some conclusions about the obtained results and suggest possible future work in the area. All of this focusing in social media and web sites that are dedicated to spread news.

Keywords: Fake news, automatic learning, misinformation, detection, classification.

1. Introducción

De acuerdo con la investigación en [1] el impacto de la divulgación de noticias por parte de reporteros y periodistas ha sido un factor muy importante en la sociedad, ya

que tiende a ser la principal fuente de consumo de información fiable por parte de los ciudadanos. No obstante, hoy en día se manejan las redes sociales como un medio de difusión más eficiente que los medios convencionales como la radio y el periódico, por lo que se permite que cualquier usuario sea capaz de compartir información ya sea verídica o engañosa. Es por este motivo que la detección y la prevención del consumo de noticias en redes sociales, presenta un problema a resolver con el fin de proteger el consumo de información fiable y verídica.

Podemos notar con el paso del tiempo, diferentes acontecimientos en donde se comenta que la gente tiende a razonar incorrectamente o a cambiar de opinión debido a información falsa o engañosa. Un ejemplo claro de esto es el rumor descrito en [2] en donde un ciudadano arremetió en un restaurante con un arma de fuego, argumentando que leyó información de que en ese lugar existía abuso infantil, liderado por la excandidata a la presidencia en Estados Unidos en 2016, Hillary Clinton. Por supuesto que esto no se limita únicamente al ámbito político, pero principalmente el auge y la comprensión de la gravedad del problema comenzó con este tipo de noticias durante esas elecciones presidenciales.

Definir el concepto de noticias falsas, (del inglés *fake news*), tiene un amplio panorama debido a las diferentes características que éstas pudieran poseer, desde el motivo o la intención por la que se está difundiendo ese tipo de información, hasta su estilo de escritura, e incluso la desinformación que el usuario tiene al momento de escribir y/o compartir algún hecho en las redes sociales. La definición que nos ofrece el sitio Web Wikipedia en español [3] es: "Las noticias falsas son un tipo de bulo que consiste en un contenido pseudo periodístico difundido a través de portales de noticias, prensa escrita, radio, televisión y redes sociales y cuyo objetivo es la desinformación". En el trabajo realizado en [4], se muestra una investigación donde concluyen que normalmente el generalizar el concepto tiende a ser de mucha dificultad, debido a la amplia gama de características que pudiesen definirlo. No obstante, nos ofrece la siguiente definición: "Las noticias falsas se refieren a todo tipo de historia falsa o noticia que principalmente está publicada y distribuida en Internet, con el fin de engañar a propósito o atraer lectores con objetivos financieros, políticos, entre otros".

Finalmente, podemos encontrar que las noticias falsas no son más que una categoría de publicación de información falsa, dentro de las cuales se encuentran: Rumores, teorías de conspiración, sátira, desinformación, propaganda, entre otras. [5].

Con ayuda de estas definiciones, en este trabajo trataremos la definición como todas aquellas publicaciones de contenido engañoso que pueda afectar la opinión o el punto de vista respecto a algún tema en el lector.

El presente trabajo presenta un análisis de diez proyectos desarrollados en los últimos dos años [9-11, 14-20], los cuales están enfocados en la detección de noticias falsas (de acuerdo con su propia definición) en redes sociales y sitios web dedicados a la recolección de noticias utilizando diferentes metodologías, algoritmos y/o técnicas que se conocen en la actualidad y que tienden a presentar buenos resultados. Es así como el presente trabajo se organiza de la siguiente manera: en la sección 2 se discute sobre cómo se recolecta la información de las publicaciones y cómo se preprocesa; en la sección tres se describen las diferentes características del texto que el estado del arte considera para su extracción y evaluación; en la sección cuatro se describen las técnicas y algoritmos utilizados para la solución del problema; y finalmente, en la sección cinco

se discute sobre cómo se ha abordado el problema, la dirección a donde se considera que la investigación se dirigirá en el futuro y conclusiones adicionales.

2. Recolección del conjunto de datos y preprocesamiento

Categorizar una publicación como noticia falsa, es un reto que conlleva a cuestionarnos en primer lugar sobre qué se considera como noticia falsa y qué es una noticia verídica. En [6] se menciona que diversos autores consideran a las noticias falsas como una subcategoría de lo que se considera como "información engañosa", la cual también puede ser catalogada como rumores, spam, etc., por lo que es importante que en cada trabajo se determine qué es lo que se considerará para la creación de su conjunto de datos y su detección. No obstante, existen algunos ejemplos de conjuntos de datos previamente catalogados, tal es el caso de la red social de Twitter.

Dentro de las propuestas de solución desarrolladas para la detección de noticias falsas, se ha generalizado dos grupos de soluciones, unas basadas en el contenido y otras basadas en la red que se está investigando [6-8]. No obstante, independientemente de qué tipo de propuesta de solución es realizada, se determina que se requiere un preprocesamiento del conjunto de datos, una representación y una selección de algoritmos, ya sean de aprendizaje automático clásicos o de aprendizaje profundo para el desarrollo de una solución. Por supuesto existen algunas otras propuestas un tanto alejadas de un desarrollo con algoritmos de aprendizaje automático, como pueden ser detección de anomalías, patrones difusos, etcétera [6]. Sin embargo, el enfoque de este trabajo consiste en analizar las propuestas que se han desarrollado para la detección y clasificación de noticias especialmente mediante el uso de algoritmos de aprendizaje automático y de aprendizaje profundo.

La tarea de recolectar información y catalogar si una noticia es falsa o no, conlleva una complejidad alta, debido a que se debe de tener conocimiento de la información presentada en el conjunto de datos, por lo cual, la gran mayoría de las soluciones analizadas tienden a trabajar con datos previamente categorizados de fuentes de bases de datos en internet. Sin embargo, el preprocesamiento de éstos es la parte en la que cada solución empieza a diferir una de otra. La tabla 1 muestra ejemplos de conjuntos de datos utilizados por algunas soluciones de detección de noticias falsas [9, 10, 15, 17]. En adición, se ha encontrado en el estado del arte que también es posible crear un conjunto de datos propio mediante APIs que son procedentes de los sitios web o redes sociales, por ejemplo, el uso de Twitter API para la extracción de publicaciones y sus correspondientes metadatos [16].

Tabla 1. Ejemplos de conjuntos de datos utilizados.

Nombre del conjunto de datos	**Proveedor**
BuzzFeed Political News dataset	BuzzFeed
LIAR dataset	Polifact
real_or_fake news dataset	Kaggle
Random Political News dataset	Data.world

Durante el preprocesamiento, es común realizar una limpieza a los datos, donde se utilizan técnicas de NLP (Por sus siglas en inglés, procesamiento de lenguaje natural)

como convertir los textos a letras minúsculas, eliminar puntuación o remover stop words o palabras vacías, con el fin de estandarizar la información para un análisis más eficiente. En [11,15] se utiliza este tipo de preprocesamiento, donde además se extraen características adicionales que se encuentran de manera implícita en los textos, las cuales se discutirán en la sección 4. Además, la representación final del conjunto de datos puede ser implementada utilizando alguna técnica como *Bag of words* [10], *Count Vectorizer* [10,16] o TF-IDF [10,14–16].

3. Extracción de características en las noticias falsas

Durante el desarrollo de una propuesta de solución que deriva en utilizar algoritmos de aprendizaje automático o aprendizaje profundo, se extraen algunas características de los textos, (las cuales son mencionadas en las secciones 3.1 y 3.2), como una fuente de información adicional para mejorar el desempeño del algoritmo. A continuación, se describen en qué consisten las características mencionadas en las publicaciones [6–8] las cuales fueron clasificadas en dos tipos: basadas en el contenido y basadas en el contexto.

3.1. Características basadas en el contenido

Este tipo de características consisten principalmente en analizar a fondo la redacción que tiene cada publicación, para determinar si existen características adicionales implícitas que pudieran permitir una mejor detección de noticias falsas. Por lo tanto, éstas son extraídas cuando se realiza un análisis léxico, sintáctico o semántico. En la tabla 2 se muestran algunos ejemplos de características basadas en el contenido y de qué tipo son, utilizadas en [11, 15, 19].

Tabla 2. Lista con ejemplos de características basadas en el contenido.

Característica	Tipo de característica
Uso de N-gramas para palabras y caracteres	Sintáctico
Conteo de signos de puntuación	Léxico
Uso de letras mayúsculas	Léxico
Uso de palabras altisonantes	Semántico
Hashtags o emoticones (sentimientos)	Semántico

3.2. Características basadas en el contexto

Este tipo de características consisten principalmente en analizar información referente al tipo de usuarios que están difundiendo, leyendo y/o compartiendo las noticias en una cierta plataforma social en internet. También incluye metadatos del sitio web como el analizar la propagación de una cierta publicación en el medio. En [13] es mencionada la importancia de obtener información de los usuarios, como su edad, género, procedencia, etc.

Tabla 3. Ejemplos de características basadas en el contexto.

Característica	Tipo de característica
Origen geográfico del usuario	Usuario
Propagación de una publicación	Red
Número de páginas visitadas	Usuario
Número de “me gusta”, “retweets”, etc.	Usuario
Número de publicaciones compartidas	Usuario
Red de amigos	Red
Comentarios en una publicación	Red

Tabla 4. Algoritmos de aprendizaje máquina.

Algoritmo	Porcentaje de trabajos analizados donde se utiliza
Decision Tree	20%
Logistic Regression	30%
SVM	40%
Naive Bayes	20%
Random Forest	40%
Gradient Boosting	30%

En la tabla 3 se muestran algunos ejemplos de características basadas en el contexto, definidas en [13, 20].

En la tabla 4 se presentan los algoritmos utilizados en los trabajos que se orientaron en el aprendizaje automático [9-11,14-18].

4. Métodos y algoritmos utilizados para la solución del problema

Como se menciona en la sección dos, en esta sección se analizan las soluciones propuestas en el estado del arte basadas en el aprendizaje máquina y en el aprendizaje profundo exclusivamente. Cabe mencionar que en el estado del arte, existen trabajos que no se limitan a trabajar con un solo tipo de algoritmo, si no que parten de desarrollos de aprendizaje automático clásicos, para posteriormente realizar una comparación de resultados con algoritmos de aprendizaje profundo u otro tipo de modelos. Así también, el desarrollo no se limita a trabajar con el texto de las publicaciones, sino también algunos trabajos consideran trabajar con las imágenes que acompañan a los textos.

4.1. Soluciones basadas en aprendizaje máquina

Dado que podemos catalogar este problema como una clasificación, existen diversos algoritmos que permiten identificar las características más importantes y orientar el

Tabla 5. Técnicas de aprendizaje profundo.

Algoritmo	Porcentaje de trabajos analizados donde se utiliza
MLP (Multilayer Perceptron)	10%
CNN (Convolutional Neural Net)	20%
LSTM (Long Short-Term Memory)	10%
Bi-LSTM (Bidirectional LSTM)	10%
GCN (Graph Convolutional Net)	20%

modelado de la solución a partir de éstas. Por lo tanto, es común observar que existen diversas soluciones basadas en algoritmos de aprendizaje automático.

Como se puede observar, los algoritmos de SVM y Random Forest son los más populares para utilizarlos en la detección de noticias falsas de acuerdo con los trabajos analizados en el estado del arte (en ambos casos, el 40% de los trabajos empleó estos algoritmos), los cuales, a su vez, también han obtenido buenos resultados en comparación con los obtenidos en trabajos enfocados en aprendizaje profundo.

4.2. Soluciones basadas en aprendizaje profundo

Estas soluciones tienden a estar acompañadas de técnicas como *Word Embeddings*, ya sea que se utilice Doc2Vec [16], GloVe [14,18,19] o algún *embedding* personalizado. La tabla 5 presenta las técnicas de aprendizaje profundo utilizadas en el estado del arte.

Se puede observar que las redes neuronales convolucionales son las preferidas para la implementación de la solución, de acuerdo con el estado del arte analizado.

5. Conclusiones y trabajo a futuro

La detección de noticias falsas en sitios web y redes sociales ha sido un problema importante a resolver en la actualidad, por lo cual, el presente trabajo ofrece un panorama actual de las diferentes tecnologías de aprendizaje automático y aprendizaje profundo para la implementación de posibles soluciones. Una vez que se han analizado las diferentes metodologías y propuestas de solución en el estado del arte, es importante mencionar que la selección de una característica, técnica o modelo, depende mucho del tipo de información que se analizará, así como también de factores como los recursos computacionales con los que se cuenta.

Se analizó que existen diferentes fuentes de datos de donde podemos partir para la elaboración de una propuesta de solución, algunos de extracción de características y los diferentes algoritmos de aprendizaje automático y aprendizaje profundo, esto con la finalidad de presentar algunas opciones que han resultado adecuadas para el problema y que se pueda partir desde este tipo de trabajos hacia propuestas de solución más elaboradas y con mejor adaptación al detectar si una publicación es engañosa o no.

Puede ser interesante considerar otro tipo de técnicas y también si existen autores que hayan desarrollado su propio clasificador para atacar este problema con mayor eficacia, de acuerdo a las características propias que se presentan.

Referencias

1. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., Procter, R.: Detection and resolution of rumours in social media: a survey. ACM Comput. Surv., 51(2), pp. 32:1–32:36 (2018)
2. Kang, C., Goldman, A.: Washington pizzeria attack. Fake News Brought Real Guns, 5 (2016)
3. Wikipedia: Fake news. https://es.wikipedia.org/wiki/Fake_news (2020)
4. Zhang, X., Ghorbani, A.A.: An overview of online fake news: Characterization, detection, and discussion. Information Processing and Management (2020)
5. Meel, P., Vishwakarma, D.K.: Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. Expert Systems with Applications (2020)
6. Bondielli, A., Marcelloni, F.A.: Survey on fake news and rumour detection techniques. Information Sciences (2019)
7. Conroy, N.J., Rubin, V.L., Chen, Y.: Automatic deception detection: methods for finding fake news. In: Proceedings of the Association for Information Science and Technology (2015)
8. Zhou, X., Zafarani, R.; Network-based fake news detection. ACM SIGKDD Explorations Newsletter (2019)
9. Ozbay, F.A., Alatas, B.: Fake news detection within online social media using supervised artificial intelligence algorithms. Physica A: Statistical Mechanics and Its Applications (2020)
10. Vasuagarwal, H., Sultana, P., Malhotra, S., Sarkar, A.: Analysis of classifiers for fake news detection. In: International Conference on Recent Trends in Advanced Computing (2019)
11. Aldwairi, M., Alwahedi, A.: Detecting fake news in social media networks. Procedia Computer Science (2018)
12. Reis, J.C.S., Correia, A., Murai, F., Veloso, A., Benevenuto, F.: Explainable machine learning for fake news detection. In: Proceedings of the 11th ACM Conference on Web Science (2019)
13. Shu, K., Zhou, X., Wang, S., Zafarani, R., Liu, H.: The role of user profiles for fake news detection. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (2019)
14. Katsaros, D., Stavropoulos, G., Papakostas, D.: Which machine learning paradigm for fake news detection? In: Proceedings - IEEE/WIC/ACM International Conference on Web Intelligence, WI (2019)
15. Hnin-Ei, W., ZarZar, W.: Content based fake news detection using n-gram models. In: Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS) ACM, Munich, Germany, 5 (2019)
16. Helmstetter, S., Paulheim, H.: Weakly supervised learning for fake news detection on Twitter. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (2018)
17. Rasool, T., Butt, W.H., Shaukat, A., Akram, M.U.: Multi-label fake news detection using multi-layered supervised learning. In: ACM International Conference Proceeding Series (2019)
18. Benamira, A., Devillers, B., Lesot, E., Ray, A.K., Saadi, M., Malliaros, F.D.: Semi-supervised learning and graph neural networks for fake news detection. In: Proceedings of

the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) Association for Computing Machinery, pp. 568–569 (2019)
19. Abedalla, A., Al-Sadi, A., & Abdullah, M. A closer look at fake news detection: A deep learning perspective. In: ACM International Conference Proceeding Series. https://doi.org/10.1145/3369114.3369149 (2019)
20. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. Explainable machine learning for fake news detection. In: WebSci 2019, Proceedings of the 11th ACM Conference on Web Science. https://doi.org/10.1145/3292522.3326027 (2019)

ISSN 1870-4069

Combinación de rankings como método para la identificación de biomarcadores de vaginosis bacteriana

Jesús Francisco Pérez-Gómez[1], Juana Canul-Reich[1], Erick De-La-Cruz-Hernandez[2]

[1] Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencias y Tecnologías de la Información,
México
[2] Universidad Juárez Autónoma de Tabasco,
División Académica Multidisciplinaria de Comalcalco,
México

juana.canul@ujat.mx

Resumen. La Vaginosis Bacteriana (VB) es una condición patológica que se detecta en mujeres sexualmente activas asintomáticas y sintomáticas que causa secreción vaginal inusual. Esta condición no es potencialmente peligrosa, pero aumenta el riesgo de contraer Enfermedades de Transmisión Sexual (ETS) y nacimientos prematuros. Sus causas no están del todo claras, pero pruebas moleculares [1, 2] sugieren que ciertos microorganismos se expresan durante esta condición. Para hallarlos, se requieren estudios que identifiquen biomarcadores. En este trabajo se utilizaron Algoritmos de Aprendizaje Automático (AAA) para encontrar los mejores atributos predictivos, es decir, biomarcadores de la VB y estudiar su relación con la enfermedad. Para ello, se utilizaron métodos de selección de atributos como *Relief*, *Fisher Score*, *Decision Trees*, *Boruta* y *Random Forests*. Los experimentos de cada método bajo un esquema de Validación Cruzada (VC) generaron cinco "rankings individuales". Con base en ellos se obtuvieron dos rankings combinados. Un primer ranking combinado se obtuvo al escalar y promediar los rankings individuales. Un segundo ranking se basa de un análisis de frecuencias a partir de los rankings individuales. Una comparativa de los rankings obtenidos identifican atributos en común como biomarcadores de la VB. El conjunto de datos analizado es una prueba microbiológica vaginal de 201 mujeres de Tabasco, México. Este es un primer esfuerzo para explorar los atributos relevantes en un conjunto de datos real de vaginosis bacteriana y es la base para planificar diversos experimentos con algoritmos clasificadores. Otros métodos de clasificación y selección de atributos para la detección de la VB se encuentran en investigación.

Palabras clave: Selección de atributos, decision tree, relief, fisher score, Boruta, Random Forest.

Combination of Rankings as a Method for Biomarker Identification of Bacterial Vaginosis

Abstract. Bacterial Vaginosis (BV) is a pathological condition detected in asymptomatic and symptomatic sexually active women that causes an unusual

vaginal discharge. This condition is not potentially dangerous, but it increases the risk of contracting Sexually Transmitted Diseases (STDs) and premature births. Its causes are not entirely clear, but molecular evidence [1, 2] suggests that certain microorganisms are expressed during this condition. To find them, studies of biomarkers identifying are required. In this work, Machine Learning Algorithms (AAA) were used to find the best predictive features and to study its relationship with the disease. For this, feature selection (FS) methods such as Relief, Fisher Score, Decision Trees, Boruta, and Random Forests were used. The experiments of each method under a Cross-Validation (VC) scheme generated five "individual rankings." Based on them, two combined rankings were obtained. A first combined ranking was obtained by scaling and averaging the individual rankings. A second combined ranking was obtained based on an analysis of frequencies from the individual rankings. A comparison of the combined rankings identifies features in common as biomarkers of BV. The dataset analyzed is a vaginal microbiological test of 201 women from Tabasco, Mexico. This is a first effort to explore the most relevant features in a real bacterial vaginosis dataset and it is the basis for the implementation of experiments with classifying algorithms. Divers classifiers algorithms for the detection of BV and other feature selection methods are under further investigation.

Keywords: Feature selection, decision trees, relief, fisher score, Boruta, Random Forest.

1. Introducción

La Vaginosis Bacteriana es una alteración del microbiota vaginal. Es una disbiosis que comúnmente se detecta en mujeres sexualmente activas asintomáticas y sintomáticas dónde se observan hallazgos clínicos como flujo vaginal anormal en color (gris o verde) y olor a pescado [1]. Las mujeres con esta infección tienen un 60% más riesgos de contraer Virus de Inmunodeficiencia Humana (VIH) y aumenta en un 30% las probabilidades de transmitir el VIH a una pareja no infectada [3]. La epidemiología de la vaginosis bacteriana como un estado comunitario de microbiota vaginal sigue siendo poco conocida, y muchas veces controvertida [2].

Hasta ahora, algunos estudios moleculares mencionan organismos anaerobios como *Lactobacillus*, *Atopobium vaginae*, *Gardnerella vaginalis* y *Megasphaera* tipo 1 y 2 implicados en la expresión de esta afección [1, 2]. Un biomarcador o marcador biológico se refiere a cualquier sustancia, estructura o proceso que pueda medirse en el cuerpo o sus productos, que permita influir o predecir la enfermedad [4]. Esta respuesta medible puede ser funcional, fisiológica, bioquímica a nivel celular o una interacción molecular [5]. Para identificarlos, se requiere determinar el nivel de información clínicamente relevante de todos los datos obtenidos en torno a la enfermedad [6].

A partir de este enfoque los mejores predictores o atributos relevantes se pueden determinar al utilizar métodos de aprendizaje automático. Por tanto, esta investigación se aborda como un problema de selección de atributos.

El objetivo principal es explorar los atributos y determinar los más relevantes en el conjunto de datos de VB utilizando cinco Algoritmos de Selección de Atributos (ASA). Específicamente se utilizaron los métodos *Relief*, *Fisher Score*, *Decision Tree*, *Boruta* y *Random Forests*. Con cada algoritmo se obtuvo el nivel de relevancia de cada

atributo, lo que conforman los "rankings individuales". Con base en éstos se obtuvieron dos rankings combinados de atributos, que muestran los predictores más relevantes de la BV. Un primer ranking combinado se basa en el promedio de los 5 valores de importancia obtenidos de los rankings individuales después de ser escalados al rango entre 0 y 1. El segundo ranking combinado se basa en un análisis de frecuencia y la obtención de la moda estadística a partir de las posiciones de los atributos en los rankings individuales.

Los dos rankings combinados de atributos con los mejores predictores de BV que se derivan de los experimentos son provistos. La base de datos utilizada en los experimentos [1] son datos de diagnóstico molecular de Vaginosis Bacteriana conformado de 201 instancias y 57 atributos. Las muestras corresponden a mujeres de Comalcalco, Tabasco y fueron obtenidas y analizadas en el Laboratorio de Investigación en Enfermedades Infecciosas y Metabólicas de la DAMC-UJAT.
Finalmente, los objetivos que sigue esta investigación se resumen en los siguientes:

1. La obtención de un primer ranking combinado de atributos con los predictores más relevantes de la VB a partir de la combinación de cinco algoritmos de selección de atributos.
2. La obtención de un segundo ranking combinado de atributos con los predictores más relevantes de la VB basado en un análisis de frecuencias a partir de los rankings individuales.

La motivación se fundamenta en analizar los datos proporcionados desde la perspectiva del Aprendizaje Automático. Experimentos adicionales, métodos y técnicas del área de la Inteligencia Artificial están siendo propuestos para la ampliación de la investigación a partir de los resultados obtenidos.

Este documento se organiza de la siguiente manera. La Sección 2 describe algunas investigaciones relacionadas a los métodos y desarrollos experimentales enfocados a la selección de atributos. La Sección 3 detalla el conjunto de datos utilizado y los métodos de aprendizaje automático implementados en esta investigación. La sección 4 explica a detalle las fases experimentales de la investigación. La Sección 5 muestra los resultados obtenidos en todos los experimentos desarrollados, y finalmente, en la Sección 6 se proporcionan las conclusiones generales del proyecto.

2. Trabajos relacionados

En esta sección se describen algunos proyectos de investigación relacionados a la selección de atributos o reducción de dimensiones propios del área de aprendizaje automático, y que han motivado la utilización de los algoritmos propuestos en el desarrollo experimental.

El propósito del trabajo de Yolanda Baker y demás [7] fue descubrir los atributos relevantes de la VB y aplicar algunos algoritmos de clasificación para su diagnóstico. Los autores aplicaron veinte algoritmos de selección de atributos en combinación con nueve algoritmos de clasificación utilizando la herramienta WEKA. La precisión -proporción de instancias correctamente clasificadas-, recall -proporción de verdaderos positivos-, y el número de atributos reducidos fueron algunas de las métricas de rendimiento que se consideraron en este estudio. Los autores encontraron que los

Tabla 1. Lista de atributos que conforman el conjunto de datos de Vaginosis Bacteriana [1].

Atributos	Valores
VBPCR	Etiqueta clase: 1=positivo, 2=Negativo, 3=Indeterminado
EDADENA, EDAD30	Edad del paciente
Citolog, CitologiaOrd, CitologiaBICAT	Citologia normal, ordinaria o anormal
Crispatus, L. Gasseri, L. Iners, L. Jensenii, CripatusCq, GasseriCq, JenseniiCq, InersCq, Megasphaera Phylotipo1, Atopobium, Gardnerella V., CT, NG, MH, UP, UU	Microorganismos obtenidos mediante qPCR.
BVNumero	Numero de patógenos
BVCombination	Combinación de patógenos
HSV12	Herpes tipo 1
RMY0911ELSY	Relacionado con MYDE0911-ELSY
ELSY, HPV, HPVgenotypes, SingleHPVComplete, MultipleHPVComplete, LRIHPVComplete, PHRHPVComplete, HRHPVComplete	Relacionados con VPH
@6, @11, @42, @44, @84, @E626	Relacionados con VPH
E653, E666, E616, E618, E631, E633, E635, E639, E645, E651, E652, E656, E658, E659, E673	Relacionados con VPH

algoritmos *Functional Trees (FT)* y *WrapperSubSetEval* resultaron ser la mejor combinación de acuerdo con las métricas utilizadas.

Beck y Foster [8] utilizaron los algoritmos *Random Forests (RF)* y *Logistic Regression (LR*) para diagnosticar la VB. Para rankear los atributos consideraron *purity increase in the node* como medida de rendimiento en *RF*. Para *LR*, los atri-butos fueron rankeados al considerar el cociente medio y la desviación estándar en todos los experimentos de la validación cruzada. *Relief* se implementó como una alternativa para calcular un tercer ranking. Como resultado se obtuvo una tabla con los atributos más relevantes. Los autores categorizaron atributos como *Aeroccocus*, *Atopobium*, *Dialister, Eggerthella* y *Gardnerella* como los más relevantes.

Muhammad y otros [9] presentaron un sistema de diagnóstico clínico para enfermedades del corazón en el cual propusieron un método de selección de atributo basado en tres algoritmos selectores: *Fisher Score-based feature selection, Forward Feature Selection (FFS)* y *Reverse Feature Selection (RFS*). En la fase de clasificación, los autores utilizaron *SVM* con el kernel de base radial. Para evaluar el rendimiento de las técnicas propuestas se consideraron medidas como *Mathews's correlation*, precisión, especificidad y sensibilidad. Las bases de datos para realizar las pruebas fueron obtenidas del repositorio *UCI (University of California Irvine*). Finalmente, los autores obtuvieron mejores resultados con la metodología propuesta en comparación con los métodos de selección de atributos de manera individual, tanto en rendimiento como en costos computacionales.

La investigación de Kumar y Shaikh [10] demuestra que el algoritmo *random forests* utilizado en los experimentos mejoró su rendimiento al utilizar los atributos seleccionados con *Boruta*. Para llegar a esta conclusión, implementaron algoritmos de selección de atributos como *Relief, Recursive Feature Elimination (RFE) y Boruta*. El conjunto de datos para los experimentos fue obtenido del repositorio de aprendizaje automático UCI que incluye 303 observaciones y 13 atributos. Para los experimentos, primero particionaron los datos en conjuntos de entrenamiento (70%) y conjuntos de prueba (30%).

El método de validación cruzada de 10 pliegues se implementó para evitar el sobreajuste del modelo. Los autores concluyeron que todos los métodos selectores obtuvieron resultados similares, la diferencia se centró en el orden de importancia de los atributos. Sin embargo, la combinación de *random forests* como clasificador y la utilización del subconjunto de atributos confirmados por *Boruta* obtuvieron medidas de rendimiento superiores a las demás pruebas.

3. Materiales y métodos

En esta sección se detalla el conjunto de datos utilizado para la implementación de los experimentos propuestos, así como los métodos de selección de atributos utilizados en el desarrollo experimental. Los métodos descritos a continuación se implementaron en el lenguaje R, y para ello se utilizó el ambiente de trabajo R-Studio en la versión 1.2.5001.

3.1. Conjunto de datos

La base de datos que se utilizó para los experimentos de esta investigación se basa en un estudio de diagnóstico molecular de Vaginosis Bacteriana [1]. Está integrado por muestras cervicales de 201 exámenes ginecológicos. Los microorganismos implicados en esta condición fueron determinados por la técnica *Quantitative Polymerase Chain Reaction* (*qPCR*). El conjunto de datos se conforma de 201 instancias y 57 atributos. Las muestras y el análisis microbiológico se realizaron en los Laboratorios de Investigación en Enfermedades Infecciosas y Metabólicas de la División Académica Multidisciplinaria de Comalcalco, Tabasco. Un resumen de los atributos que conforman la base de datos se muestra en la Tabla 1.

3.2. Preprocesamiento de datos

El conjunto de datos proporcionado contiene algunos datos faltantes, por lo que antes de ser utilizado se sometió a un preprocesamiento. Para esto, las instancias y atributos con valores nulos se eliminaron. De acuerdo con los proveedores de los datos [1], este proceso no representa la reducción de información relevante para la detección de la VB. Los atributos eliminados corresponden a datos relacionados con el VPH, por lo que su eliminación no representa una pérdida de información para el estudio de la VB.

Originalmente los datos proporcionados contienen tres etiquetas: positivo, negativo e indeterminado. Para los propósitos particulares de esta investigación, la clase

indeterminada se eliminó, ya que en la fase de entrenamiento de los métodos utilizados el interés es identificar entre pacientes enfermos y no enfermos.

Finalmente, la base de datos preprocesada contiene 1 etiqueta clase, 34 atributos y 173 instancias.

3.3. Algoritmos de selección de atributos

Los ASA, por sus siglas, son técnicas para el descubrimiento del conocimiento que proporcionan el entendimiento del problema a través del análisis de los atributos más relevantes [11]. En tareas de clasificación, estos métodos permiten mejorar el rendimiento de un algoritmo clasificador al reducir los costos y sobrecargas operacionales. Muchos ASA incluyen un ranking de atributos como un mecanismo de selección principal, lo que denota los atributos con mayor y menor relevancia [12]. Un ranking de atributos evalúa la relevancia individual de los atributos, sin considerar las posibles interacciones entre ellos [13]. Los procedimientos de cada ASA para calcular un ranking de atributos se describen a continuación.

Árboles de Decisión (*Decision Tree*)

Un árbol de decisión (*DT*, por sus siglas en inglés) es un algoritmo de clasificación. Sin embargo, en sus procedimientos evalúa la importancia de los atributos. En el proceso de creación de un árbol, la relevancia de los atributos se obtiene de acuerdo a la entropía. Max Bramer [14] describe la entropía como una medida teórica de "incertidumbre" de la información que contiene un conjunto de datos de entrenamiento. El paquete en R *caret* [15] proporciona una implementación de árboles de decisión en el algoritmo *J48*.

Bosques Aleatorios (*Random Forests*)

Este método de ensamble es una combinación de árboles de decisión relacionados entre ellos [16]. Su enfoque genera una gran cantidad de árboles, donde cada árbol depende de los valores de un vector de instancias aleatorias, mide su desempeño y selecciona los mejores del conjunto [14]. El paquete en R *FSelector* [17] proporciona una implementación del algoritmo *Random Forests*.

Boruta

Este es un método de selección de atributos tipo envoltura que se construye en torno al algoritmo de clasificación *Random Forest*. La idea general es que en cada iteración se genere una serie de atributos sombra a partir de los predictores, copia cada uno de ellos y permuta entre sí los elementos de cada nueva columna. Un modelo con *Random Forest* se ajusta, y las importancias relativas de cada atributo se calculan. Si una variable queda por debajo de las sintéticas (ruido), será indicativo de que su aportación al modelo es dudosa, y por tanto se elimina. El proceso continua hasta que todas las variables son aceptadas, rechazadas o finalizan las iteraciones [18]. El paquete en R *Boruta* [19] proporciona una implementación del algoritmo *Boruta.*

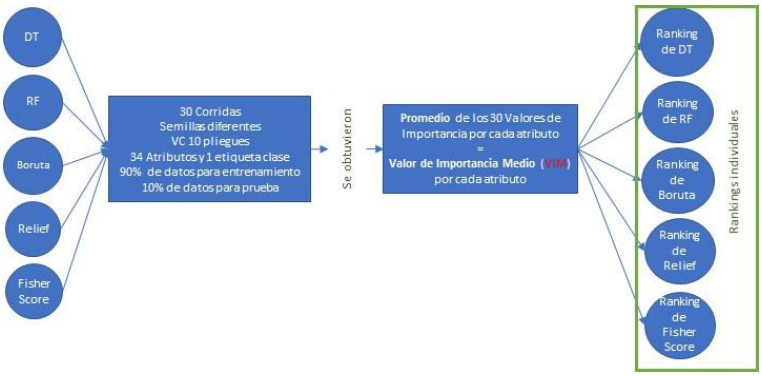


Fig. 1. Proceso de creación de los rankings individuales. En letra roja se especifica el criterio que se utilizó para el ordenamiento de mayor a menor relevancia. DT: *decision tree*; RF: *random forests*; VC: validación cruzada.

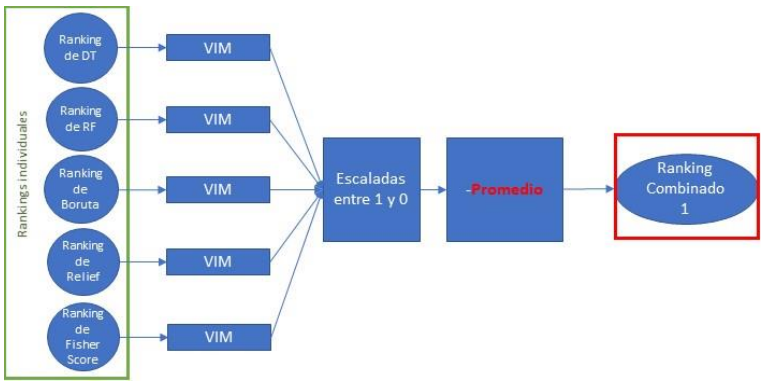


Fig. 2. Proceso de creación del primer ranking combinado de atributos. En color rojo se realza la medida considerada para ordenar los atributos del ranking combinado. DT: *decision tree*, RF: *random forests*; VIM: valor de importancia medio.

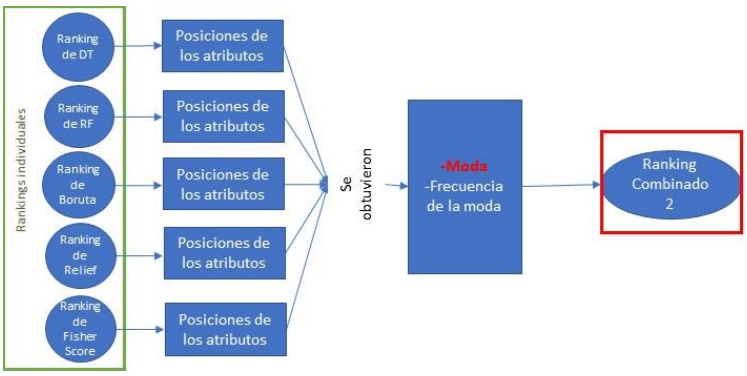


Fig. 3. Proceso de creación del segundo ranking combinado de atributos. En color rojo se realza

Relief

Relief es un método multivariado tipo filtro para la selección de atributos. Selecciona atributos relevantes basándose en la diferencia de valores de atributos entre pares de instancias utilizando el algoritmo *nearest-neighbor* y proporciona una puntuación para cada atributo [20]. El algoritmo *nearest-neighbor* se basa en la distancia Manhattan. El paquete en R *FSelector* [17] proporciona una implementación del algoritmo *Relief.*

Fisher Score

Fisher score es un método supervisado de selección de atributos que se utiliza para la reducción de dimensionalidad. Éste crea una puntuación para cada atributo de manera independiente bajo el criterio de Fisher, que conduce a un subconjunto de características subóptimo [21]. La idea general de este método es encontrar los atributos donde la distancia entre los puntos de clases diferentes sea tan grande como sea posible, mientras que la distancia entre los puntos de datos de la misma clase sea tan pequeña como sea posible [22]. Matemáticamente, el *F-score* -otro nombre con el que se conoce- proporciona una medida de qué tan bien un atributo a la vez puede discriminar entre diferentes clases [23]. Cuanto más alto sea el *F-score*, mayor el poder de discriminación. El paquete en R *PredPsych* [23] proporciona una implementación del algoritmo *Fisher Score*.

4. Diseño Experimental

La fase experimental para explorar los predictores más relevantes, es decir, los biomarcadores de la Vaginosis Bacteriana consisten de 30 corridas de cada ASA bajo un esquema de validación cruzada de 10 pliegues. A través de las 30 corridas se utilizaron semillas diferentes para asegurar la aleatoriedad de los datos. Esto significa que en cada corrida el conjunto de datos completo es fraccionado en 10 partes iguales, y cada fracción considera el 90% de las instancias para datos de entrenamiento y el 10% restante para datos de prueba. Al final, una corrida de validación cruzada promedia las 10 medidas de rendimiento obtenidas. Los ASA se aplicaron al subconjunto de entrenamiento de cada validación cruzada, el cual varía en cada iteración. Esto permitió calcular 30 Valores de Importancia (VI) para cada atributo de acuerdo con las métricas de cada ASA. Posteriormente, todos los VI se promediaron y se determinó así un Valor de Importancia Medio (VIM) para cada atributo. De esta manera se obtuvieron los rankings de atributos individuales. Este proceso se muestra gráficamente en la Fig. 1.

Para obtener un primer ranking combinado de atributos, los VIM de cada atributo -uno por cada ASA- se escalaron al rango entre 0 y 1. El escalado consistió en la obtención del cociente al dividir el VIM entre el número 100. Una vez escalados, se obtuvo el promedio de los VIM. El promedio obtenido es la base para la creación del primer ranking combinado. Este proceso se muestra gráficamente en la Fig 2.

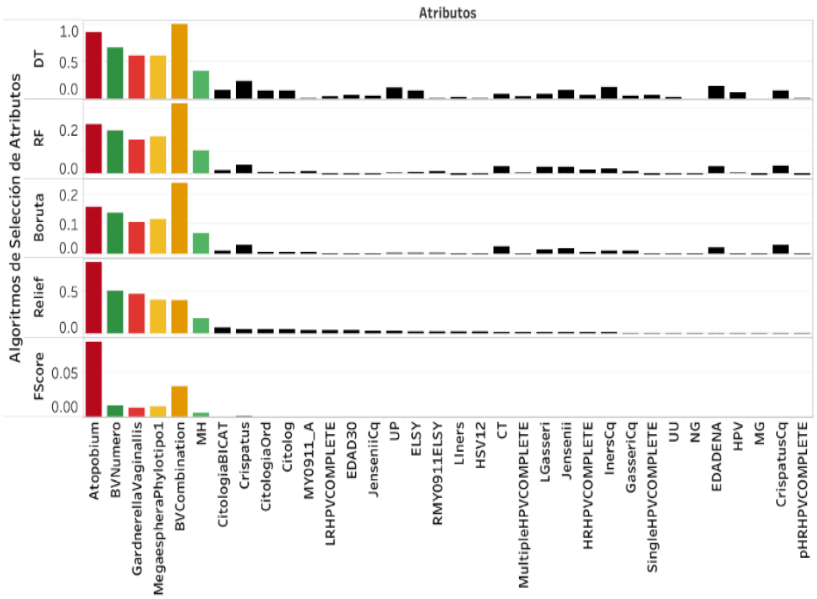


Fig. 4. Representación gráfica de los rankings individuales de atributos obtenidos mediante los algoritmos de selección de atributos (ASA). El cálculo de los valores de importancia medio (VIM) se representa por las barras verticales. Cada algoritmo emplea su propia escala de valores. DT: *decision tree*; RF: *random forests*.

Tabla 2. Primer ranking combinado de atributos. Se basa en el escalado y promedio de los cinco valores de importancia medios (VIM) obtenido por cada atributo en los rankings individuales. El VIM es el criterio de ordenamiento.

Atributos	**Promedio de VIM**
Atopobium	0.641467676
BVCombination	0.547947691
BVNumero	0.443284461
MegaespheraPhylotipo1	0.343827918
GardnerellaVaginallis	0.335289495
MH	0.21830713
Crispatus	0.064334192
EDADENA	0.040210846
CitologiaBICAT	0.037516153
InersCq	0.035282781
UP	0.033222477
CrispatusCq	0.030599123
Jensenii	0.030567158
Citolog	0.028379047
CitologiaOrd	0.028356376

Al tomar como base los rankings individuales, se realizó un análisis de frecuencias. Éste se basa en las posiciones conseguidas por los atributos dentro de cada ranking individual. Para esto, se estimaron la moda estadística y la frecuencia de la moda. El segundo ranking combinado de atributos se calculó a partir de este análisis. La moda estadística se considera como criterio de ordenamiento, ya que calcula el valor que ocurre con más frecuencia en un conjunto de observaciones. Este proceso se muestra gráficamente en la Fig. 3.

5. Resultados

Los resultados de los experimentos con Algoritmos de Selección de atributos se muestran a continuación. En la Fig. 4 se muestran los rankings individuales obtenidos. El eje X representa los atributos del conjunto de datos. El eje Y representa los ASA implementados. Los VIM obtenidos por cada atributo, es decir, el promedio de todos los valores de importancia a través de las 30 corridas de validación cruzada, se representa por las barras verticales en escala de cada método. Los seis atributos con mayor VIM se identifican con barras de color rojo, verde y amarillo en diferentes tonalidades.

Con el fin de crear el primer ranking combinado de atributos se consideraron los VIM obtenidos a partir de los rankings individuales. Los algoritmos emplean diferentes escalas de valores para mostrar sus resultados, por lo que se adaptaron a una misma escala. Para esto, los cinco VIM se escalaron y posteriormente se promediaron. El resultado se muestra en la Tabla 2. Los atributos se ordenan con base en el valor del VIM obtenido.

Por razones de espacio, se muestran solo los 15 atributos con los valores más altos. Posteriormente se realizó un análisis de frecuencia. Para crearlo, se calcularon las posiciones obtenidas por los atributos en cada uno de los rankings individuales. A partir de estas posiciones, se calculó la moda estadística y la frecuencia de la moda. Los resultados se muestran en la Tabla 3. Por razones de espacio, solo se muestran los primeros 15 atributos. Finalmente, los dos rankings combinados de atributos resultantes de los experimentos se comparan en la Tabla 4.

6. Conclusión

En este trabajo se identificaron los atributos más relevantes en un conjunto de datos microbiológico acerca de la vaginosis bacteriana. Para determinarlos, se implementaron cinco métodos de selección de atributos propios del área de aprendizaje automático en diversos experimentos. Con base en el análisis de los resultados obtenidos se crearon dos rankings combinados de atributos.

Es de importancia el considerar que entre ellos identifican al menos 10 atributos en común como los de mayor relevancia, sobre todo en las primeras seis posiciones.

Tabla 3. Segundo ranking combinado de atributos. Se basa en un análisis de frecuencias a partir de los rankings individuales. La moda se considera como criterio de ordenamiento. R: Relief, FS: fisher score, J48: decision tree, B: *boruta*, RF: *random forests*.

Atributos	Algoritmos de Selección de Atributos					Moda	Frecuencia de la moda
	R	FS	J48	B	RF		
BVNumero	2	2	1	1	1	1	3
Atopobium	1	1	2	2	2	2	3
GardnerellaVaginallis	3	3	3	3	3	3	5
MegaespheraPhylotipo1	4	4	4	4	4	4	5
BVCombination	5	5	5	5	5	5	5
MH	6	6	6	6	6	6	5
CitologiaBICAT	7	7	7	7	7	7	5
Crispatus	8	9	8	10	10	8	2
LRHPVCOMPLETE	12	20	13	8	8	8	2
ELSY	16	10	18	9	9	9	2
EDAD30	13	8	12	11	11	11	2
LIners	18	11	19	12	12	12	2
CitologiaOrd	10	13	9	13	13	13	3
RMY0911ELSY	17	14	14	20	19	14	2
Citolog	9	15	11	14	15	15	2

Tabla 4. Comparativa de los dos rankings combinados de atributos obtenidos. Por límites de espacio, sólo los primeros 15 atributos son mostrados. La etiqueta "a" denota que el atributo aparece en ambos rankings.

Ranking combinado 1	Ranking combinado 2
Atopobium[a]	BVNumero[a]
BVCombination[a]	Atopobium[a]
BVNumero[a]	GardnerellaVaginallis[a]
MegaespheraPhylotipo1[a]	MegaespheraPhylotipo1[a]
GardnerellaVaginallis[a]	BVCombination[a]
MH[a]	MH[a]
Crispatus[a]	CitologiaBICAT[a]
EDADENA	Crispatus[a]
CitologiaBICAT[a]	LRHPVCOMPLETE[b]
InersCq	ELSY[b]
UP	EDAD30[b]
CrispatusCq	Liners
Jensenii	CitologiaOrd[a]
Citolog[a]	RMY0911ELSY
CitologiaOrd[a]	Citolog[a]

Dichos atributos -considerados como biomarcadores de la vaginosis bacteriana- se analizarán para determinar su significancia biológica, ya que están potencialmente relacionados con la presencia o expresión de la infección.

Los rankings obtenidos son la base para planificar diversos experimentos con algoritmos clasificadores con el fin de definir los atributos o subconjunto de atributos óptimos en la detección de la VB. Esta es una investigación en curso que forma parte de un análisis exploratorio extenso. Más experimentos con otros métodos de selección de atributos y técnicas de clasificación están siendo investigadas.

Referencias

1. Sanchez-Garcia, E.K., Contreras-Paredes, A., Martinez-Abundis, E., Garcia-Chan, D., De La Cruz-Hernandez, E.: Molecular epidemiology of bacterial vaginosis and its association with sexually transmitted pathogens in healthy women. J. Med. Microbiol. Mol., 68 (2019)
2. Verstraelen, H., Swidsinski, A.: The biofilm in bacterial vaginosis: Implications for epidemiology, diagnosis and treatment: 2018 update. Curr. Opin. Infect. Dis. 32, pp. 38–42 (2019)
3. Hoang, T., Toler, E., DeLong, K., Mafunda, N.A., Bloom, S.M., Zierden, H.C., Moench, T.R., Coleman, J.S., Hanes, J., Kwon, D.S., Lai, S.K., Cone, R.A., Ensign, L.M.: The cervicovaginal mucus barrier to HIV-1 is diminished in bacterial vaginosis. PLoS Pathog (2020)
4. Anonymous: Biomarkers in risk assessment: Validity and Validation (2001)
5. World Health Orgnization: Biomarkers and risk assessment: concepts and principles. Environmental Health Criteria, 155, pp. 82 (1993)
6. Strimbu, K., Tavel, J.A.: What are biomarkers? Curr. Opin. HIV AIDS. 5, pp. 463–466 (2010)
7. Baker, Y.S., Agrawal, R., Foster, J.A., Beck, D., Dozier, G.: Detecting bacterial vaginosis using machine learning. In: Proceedings of the ACM Southeast Regional Conference on - ACM SE '14. pp. 1–4 (2014)
8. Beck, D., Foster, J.A.: Machine learning classifiers provide insight into the relationship between microbial communities and bacterial vaginosis. BioData Min. 8, pp. 1–9 (2015)
9. Saqlain, S.M., Sher, M., Shah, F.A., Khan, I., Ashraf, M.U., Awais, M., Ghani, A.: Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines. Knowl. Inf. Syst. 58, pp. 139–167 (2019)
10. Kumar, S.S., Shaikh, T.: Empirical evaluation of the performance of feature selection approaches on random forest. In: Int. Conf. Comput. Appl. ICCA. Pp. 227–231 (2017)
11. Khaire, U.M., Dhanalakshmi, R.: Stability of feature selection algorithm: A review. J. King Saud Univ. Comput. Inf. Sci. (2019)
12. Iguyon, I., Elisseeff, A.: An introduction to variable and feature selection (2003)
13. Duch, W., Grabczewski, K., Winiarski, T., Biesiada, J., Kachel, A.: Feature selection based on information theory, consistency and separability indices. In: ICONIP, Proc. 9th Int. Conf. Neural Inf. Process. Comput. Intell. E-Age. 4, pp. 1951–1955 (2002)
14. Bramer, M.: Principles of data mining. Springer, United Kingdom (2016)
15. Kuhn, M.: Building predictive models in R using the caret package. J. Stat. Softw. 28, pp. 1–26 (2008)
16. Medina-Merino, F., Ñique Chacón, I.: Bosques aleatorios como extensión de los árboles de clasificación con los programas R y Python. pp. 165–190 (2017)
17. Romanski, P.: Package FSelector. http://cran.r-project.org/web/packages/FSelector/FSelector.pdf (2013)
18. Guerrero, J.A.: El problema de la dimensionalidad. Rev. Estadística y Soc. 1, pp. 22–24 (2016)

19. Kursa, M.B., Rudnicki, W.R.: Feature selection with the Boruta package. J. Stat. Softw. 36, pp. 1–13 (2010)
20. Robnik-Sikonja, M., Kononenko, F.: Theoretical and empirical analysis of relieff and rreliefF. Mach. Learn, 53, pp. 23–69 (2003)
21. Gu, Q., Li, Z., Han, J.: Generalized fisher score for feature selection. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence, UAI, pp. 266–273 (2011)
22. Valizade-Hasanloei, M.A., Sheikhpour, R., Sarram, M.A., Sheikhpour, E., Sharifi, H.: A combined fisher and laplacian score for feature selection in QSAR based drug design using compounds with known and unknown activities. J. Comput. Aided. Mol. Des., 32, pp. 375–384 (2018)
23. Koul, A., Becchio, C., Cavallo, A.: PredPsych: A toolbox for predictive machine learning-based approach in experimental psychology research. Behav. Res. Methods, 50, pp. 1657–1672 (2018)

ISSN 1870-4069

Ontología de estilos de aprendizaje para asistir en el aprendizaje significativo

Yuridiana Alemán, María-Josefa Somodevilla, Darnes Vilariño

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México

{yuridiana.aleman, mariajsomodevilla, dvilarinoayala}@gmail.com

Resumen. El presente trabajo muestra los avances de una investigación enfocada en una metodología para el aprendizaje ontológico en un subdominio pedagógico. La propuesta general se centra en el uso de técnicas de Procesamiento de Lenguaje Natural y Recuperación de Información en todas las fases del aprendizaje ontológico. Se trabaja con el idioma español, con recursos relacionados con el aprendizaje significativo y la personalización de la enseñanza. Particularmente, se analizan los elementos teóricos de los estilos de aprendizaje, generando una lista de conceptos importantes creada con la ayuda de un experto en el dominio (*gold standard*), junto con una ontología creada a partir de esa lista. La lista creada, y por lo tanto, la ontología, contiene conceptos que están integrados en el corpus utilizado en la investigación general, por lo que se convierten en recursos importantes para el desarrollo del sistema de ontologías propuesto.

Palabras clave: Aprendizaje ontológico, ontología, corpus pedagógico, estilos de aprendizaje, Honey-Alonso.

Learning Styles Ontology to Assist in Meaningful Learning

Abstract. This work shows advances of a research focused on an ontological learning methodology for pedagogical subdomain. The main proposal focuses on the use of Natural Language Processing and Information Retrieval techniques in ontological learning process. It works with Spanish language, resources related to meaningful learning and teaching personalization. Particularly, learning styles theoretical elements are analyzed, and a list of important concepts is created with the help of a domain expert (*gold standard*). With these resources, a learning styles ontology is created. The list and ontology contain concepts for the corpus

used in the main research, therefore, these resources are important for the proposed ontology system development.

Keywords: Ontology learning, ontology, pedagogic corpus, learning styles, Honey-Alonso.

1. Introducción

Una ontología es una formalización explícita, en donde un texto se representa por medio de relaciones semánticas y palabras clave. Existen varias clasificaciones de esta estructura, siendo la más común las ontologías de dominio, las cuales son generalizaciones sobre tareas específicas en un área concreta del conocimiento. Estas representaciones se han convertido en uno de los recursos más utilizados al momento de estructurar información. Su uso va desde la formalización de un dominio específico hasta la creación sistemas de recuperación de información para consultas semánticas.

Analizando el dominio pedagógico, el aprendizaje se define como un cambio en la conducta debido a la experiencia [4]. Otras definiciones integran elementos relacionados con la didáctica, en las que se describe como el proceso de adquisición de una disposición, relativamente duradera, para cambiar la percepción o la conducta como resultado de una experiencia [1]. El aprendizaje significativo se define como un proceso que se genera en la mente humana cuando subsume nuevas informaciones de manera no arbitraria y sustantiva [2]. Para abordar esta temática, se analiza la personalización del aprendizaje, donde un estudiante aprende mejor con ciertos esquemas y técnicas de acuerdo a sus características intrínsecas [3].

El objetivo general de la investigación consiste en la construcción de ontologías de manera semiautomática a partir de textos pedagógicos no estructurados para expresar con claridad y precisión el conocimiento. Para el dominio de estudio, se utilizan tópicos relacionados con la implementación de técnicas de enseñanza dentro del salón de clases. Este artículo se enfoca en la fase de la creación de una ontología manual de estilos de aprendizaje que sea auxiliar en la evaluación de los procesos semiautomáticos implementados.

La estructura del artículo se muestra a continuación, en la sección 2 se exponen los elementos teóricos relacionados las ontologías y el dominio, especialmente los estilos de aprendizaje. La sección 3 integra las investigaciones más recientes en cuanto a la creación de ontologías en el dominio pedagógico. La sección 4 explica la metodología aplicada y la sección 5 analiza los componentes de la ontología diseñada. Finalmente, en la sección 6 se presentan las conclusiones obtenidas hasta el momento.

2. Marco teórico

La palabra *Ontología* se deriva del griego *ontos* (estudio del ser) y *logos* (palabra). En las ciencias computacionales, una ontología se define como "una

base de datos que describe los conceptos en el mundo o algún dominio, algunas de sus propiedades y como los conceptos se relacionan entre sí"[16]. Esta base de datos se define a partir de un corpus base, del cual se extraen los elementos principales o palabras clave. Posteriormente, del mismo texto se infieren las relaciones entre palabras clave, de esta manera, se crea una estructura de grafo donde los nodos son las palabras clave y las aristas representan la relación existente entre ellas. Entre las aplicaciones más representativas de las ontologías se encuentran la representación formal del conocimiento, lo que facilita el manejo e integración de datos con estructuras diferentes. Una ontología se crea mediante un proceso denominado aprendizaje ontológico, el cual se basa em texto no estructurado y se define como el proceso de identificar términos, conceptos, relaciones y, opcionalmente, axiomas a partir de información textual y usarlos para construir y mantener una ontología [17].

En una ontología, los conceptos representan la base para la descripción de la información. Esta descripción se realiza mediante tres componentes: Términos, atributos y relaciones. Los términos son nombres utilizados para referirse a un concepto específico que puede incluir un conjunto de sinónimos que especifican los mismos conceptos. Los atributos describen el concepto a detalle utilizando características, y las relaciones se utilizan para representar correspondencias entre diferentes conceptos y proveer una estructura general de la ontología [14].

Para la implementación de la metodología, se analiza el dominio pedagógico, y esta investigación se centra en la estructuración de los estilos de aprendizaje. El aprendizaje se asume como un proceso en el que el sujeto que aprende lo hace de manera dinámica, de acuerdo con unas disposiciones y características particulares y en el que están involucrados un orden y un procedimiento lógico. De este análisis surge el estudio de los estilos de aprendizaje, los cuales reflejan la forma en que el individuo aprende, describen las condiciones bajo las que un discente se encuentra en la mejor situación para aprender, o qué estructura necesita para mejorar el proceso de aprendizaje. Existen variaciones en cuanto a la manera en que los seres humanos captan y procesan información.

Se han propuesto varias teorías para describir los distintos tipos de aprendizaje ([1,8,10]), los cuales están basados en los estudios de David Kolb [9]). Para esta investigación se tomó el modelo de Honey Alonso [1], el cual propone un método para describir cómo los estudiantes resuelven sus problemas y aplican conocimientos nuevos a partir de la experiencia personal dentro de su entorno de aprendizaje, considerando los procesos psicológicos de percepción y procesamiento. De este enfoque, se distinguen 4 estilos de aprendizaje, los cuales se explican a continuación.

- *Activo:* Personas que se involucran con experiencias nuevas, tienden a actuar primero y luego piensan en las consecuencias. Muestra como principales conductas al momento de aprender la animosidad, la improvisación, la búsqueda y el descubrimiento de novedad, el riesgo y la espontaneidad.
- *Reflexivo:* Personas que son observadores y analizan sus experiencias desde diferentes perspectivas. Recopilan y analizan datos en detalle antes de tomar una conclusión. Este perfil tiene conductas de receptividad, ponderación,

análisis, exhaustividad y toma de conciencia, y entre las otras menos centrales pero presentes en él, la observación, la identificación de pequeños detalles, la elaboración de argumentos, la previsión, la habilidad para redactar informes y la prudencia.

- *Teórico:* Muestra dentro de las principales características la lógica, la metódica, la objetividad, la criticidad y la estructuración en las acciones. Las personas con este estilo de aprendizaje adaptan e integran sus observaciones en teorías complejas y lógicamente fundadas. Su sistema de valores prioriza la lógica y la racionalidad antes del análisis y la síntesis.
- *Pragmático:* Incluye personas que prueban sus ideas, teorías y técnicas nuevas y tratan de ver si funcionan en la práctica. No les gustan las discusiones largas sobre el mismo tema. Son prácticos y se adhieren a la realidad. Las cinco principales características de este estilo de aprendizaje se hallan la experimentación, la practicidad, el dirigirse a situaciones y a personas de manera directa, la eficacia y el realismo.

Para determinar el estilo de aprendizaje predominante, se utiliza el Cuestionario Honey Alonso de Estilos de Aprendizaje (CHAEA), el cual contiene 80 reactivos, 20 relacionadas con cada estilo.

3. Trabajos relacionados

La Figura 1 muestra las investigaciones realizadas en el dominio pedagógico de acuerdo al enfoque utilizado, si la ontología se creo para clases presenciales u *online*.

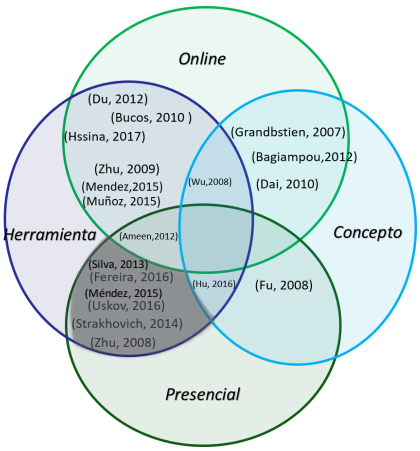


Fig. 1. Investigaciones sobre ontologías en el dominio pedagógico.

Dentro de las investigaciones de ontologías de una materia o tópico en particulas, en [7] se propone el proyecto OURAL (*Ontologies for the Use of digital learning Resources and semantic Annotations on Line*) el cual integra las disciplinas de ciencias de la educación, informática y psicología cognitiva con el fin de crear servicios para *E-learning*. En [6] también se analizan el dominio educativo, sin embargo, al ser aplicado al idioma Chino, utilizan un preprocesamiento para analizar las características de dicho idioma: acoplamiento, relevancia y consenso.

En [18] se desarrolla un sistema de educación en Internet basado en ontologías, que implementa el intercambio y la reutilización del material de aprendizaje en diferentes sistemas. Es una investigación cualitativa, donde se aborda un ejemplo con un curso básico de computación en línea que describe los módulos del sistema: aprendizaje, interfaz y recursos.

[5] presentan ENGOnto; la ontología integra múltiples ontologías relevantes para que los agentes personalizados aborden los cambios dinámicos del proceso de aprendizaje de los alumnos, la interacción entre el instructor y los recursos de aprendizaje en el entorno de la enseñanza del idioma inglés. La ontología se construyó manualmente, pero los autores describen el proceso de generación de dependencia de puntos de conocimiento (integración de clases).

En [19] se presenta el diseño de secuencias de aprendizaje utilizando la formalización por medio de ontologías (lenguaje OWL); dicho diseño se enfoca en el aprendizaje individual con el propósito de establecer una secuencia personalizada de acuerdo con el nivel y características de cada alumno en la educación superior en ambientes virtuales.

El trabajo de investigación propuesto se centra en el área sombreada, es decir, la construcción de la ontología será como una herramienta para clases presenciales. Dentro de esta área, [11] propone utilizar un modelo ontológico para la personalización del aprendizaje que involucre el perfil de los estudiantes de acuerdo con la teoría de inteligencia múltiple de Howard Gardner, así como usar una ontología de dominio que ayude a representar el conocimiento en plataformas de aprendizaje virtuales. Este modelo es utilizado en la educación online, donde se infiere el tipo de inteligencia del estudiantes y de acuerdo a esto se recomiendan contenidos pertinentes.

En [13] presenta una reingeniería de una ontología de estilos de aprendizaje en varios enfoques, la cual tiene el objetivo de apoyar la creación, adaptabilidad y uso de objetos de aprendizaje. Otras investigaciones como la de [5] se propone una ontología para la interacción entre el estudiante y el maestro en la enseñanza del idioma inglés; mientras que en la segunda investigación la ontología involucra el uso del Internet para la mejora del aprendizaje.

En [15] se centran en el aprendizaje autónomo en línea, proponiendo una ontología en base al Internet de las cosas (IoT) aplicado al aprendizaje dentro del aula con la ayuda de la tecnología, tomando como referencia los tipos de inteligencias de los estudiantes.

4. Metodología

La Figura 2 muestra la metodología general para la investigación, la cual consta de tres fases principales. El primer paso consiste en la creación de un corpus compuesto de artículos publicados en el idioma español, dentro de esta fase se realizó un proceso de validación de clases y expansión de un corpus inicial a fin de incrementar la riqueza del vocabulario. En la segunda fase se obtienen los conceptos principales y las relaciones taxonómicas y no taxonómicas para la obtención de la ontología. Finalmente, se anexa un módulo de evaluación para determinar la eficacia de los experimentos, dentro del cual se necesitarán las herramientas analizadas en el presente artículo.

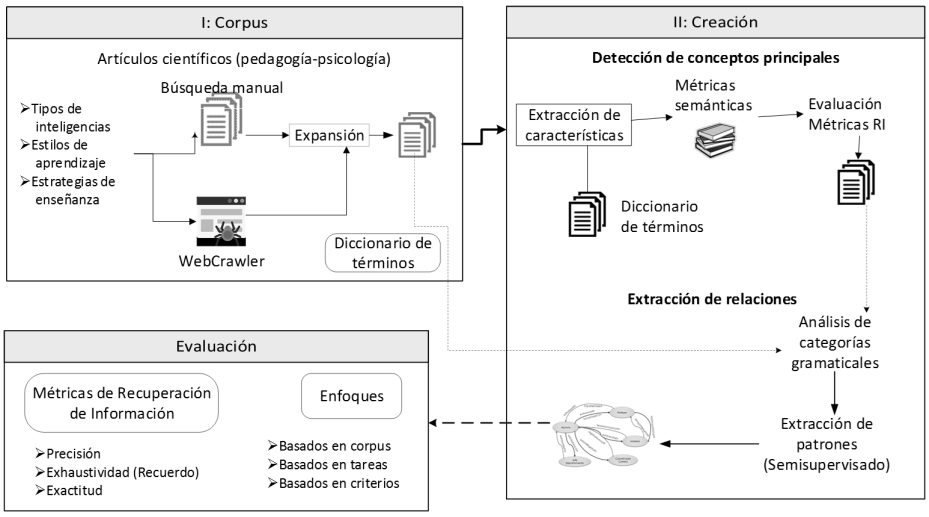


Fig. 2. Metodología general propuesta.

El corpus final está conformado por artículos relacionados con los tópicos principales seleccionados anteriormente (tipos de inteligencias, estrategias de enseñanza y estilos de aprendizaje), de tal manera que se crean tres clases iniciales. La Figura 3 muestra la relación que existe entre los tópicos seleccionados para su estudio.

Para la detección de conceptos cada una de las clases se procesa de manera independiente. Los avances previos de esta investigación integran la propuesta de un método para la detección de conceptos principales basado en métricas de similitud textual, un algoritmo de filtrado de documentos para la expansión del corpus inicial y el análisis de patrones de categorías gramaticales para la detección de relaciones entre conceptos.

A fin de tener un referente de comparación para los experimentos, se realizó una ontología de manera manual.

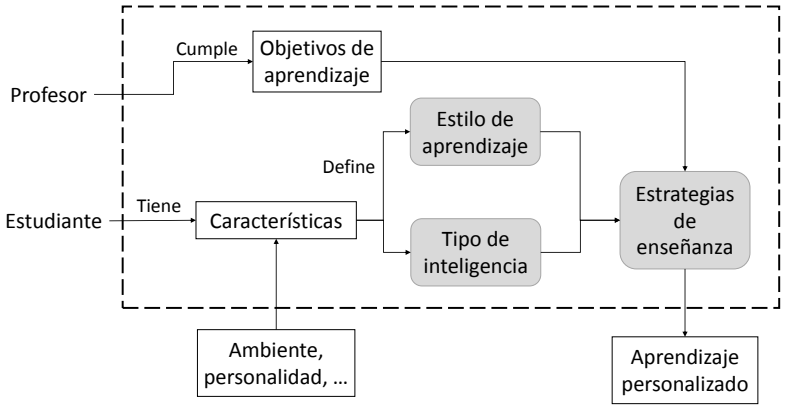


Fig. 3. Papel de los elementos seleccionados en el aprendizaje significativo.

En el presente artículo se trabajó con la clase de estilos de aprendizaje, ya que presenta mejor comportamiento en los experimentos realizados. Para su construcción, se utiliza la herramienta *Protégé*, la cual, facilita la estructuración de los componentes necesarios. En el proceso de creación, se sigue la metodología de [12], la cual consta de los siguientes pasos:

1. Definir el dominio y el alcance de la ontología, la aplicación final y a qué tipo de preguntas responderá dicha ontología.
2. Considerar el rehúso de ontologías. En caso de poder aplicarse, se recolectan términos y características de ontologías comunes ya existentes.
3. Enumerar términos importantes en la ontología. Definir una lista de los términos que se quieren almacenar en la ontología, así como de sus propiedades y las relaciones que existen entre ellos.
4. Definir las clases y la jerarquía de las clases (estructura conceptual del dominio). Primero se crean las clases para los conceptos generales en el dominio.
5. Definir las propiedades de las clases, esto es, la estructura interna de los conceptos.
6. Definir los diferentes tipos de valores que describan a los conceptos como valor asociado, cardinalidad, valores permitidos, entre otros.
7. Crear instancias de clases individuales en la jerarquía.

4.1. Lista de conceptos para validación

El *gold standard* se generó manualmente con la ayuda del experto en el dominio y contiene una serie de conceptos por clase considerados como importantes. Las listas contienen 130 elementos para los tipos de inteligencias, 87 para las

estrategias de enseñanza y 184 para los estilos de aprendizaje, sin embargo, estas listas tienen elementos en común.

La Figura 4 muestra el diagrama de Venn con el número de elementos que se comparten por clase.

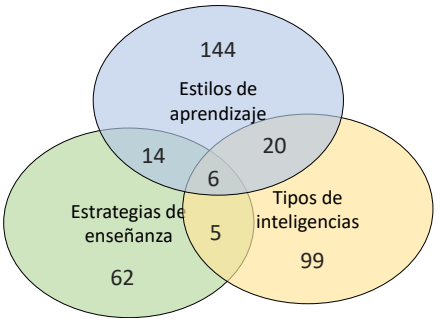


Fig. 4. Número de elementos del conjunto de evaluación por clase.

Las tres clases comparten 6 conceptos, mientras que la clase de estilos de aprendizaje es la que comparte un mayor número de elementos con otras clases (14 con estrategias de enseñanza y 20 con tipos de inteligencias). La clase estrategias de enseñanza, a pesar de tener más texto que las demás, tiene menos conceptos importantes y comparte pocos con otras clases. La Tabla 1 contiene ejemplos de algunos conceptos del conjunto de evaluación por clase y los que están presentes en los tres conjuntos.

Tabla 1. Elementos del conjunto de evaluación por clase (extracto).

Estilos de Aprendizaje	Estrategias de Enseñanza	Tipos de Inteligencia	Común a 3 clases
TIC	Aprendizaje	Perrsona	Docente
Reflexivo	Estrategia	Inteligencia múltiple	Estrategia
Estudio	Conocimiento	Desarrollo	Estudiante
Pragmático	Metacognitiva	Teoría	Habilidad
Académico	Análisis	Gardner	Individuo

5. Descripción de la ontología

1: Determinar el dominio y alcance de la ontología.

- Nombre de la ontología: Ontología sobre estilos de aprendizaje según el modelo de Honey-Alonso.
- Propósito: Representar y formalizar el conocimiento del dominio.

- Alcance: Apoyo para lograr el aprendizaje significativo por medio de la personalización del mismo.
- Usuarios finales: Docentes de clases presenciales.
- Fuentes de conocimiento: Expertos en el dominio, artículos especializados sobre el tema.

2: Considerar el rehúso de ontologías. Como tal no se tomó ninguna ontología creada, pero se tomaron en cuenta las investigaciones de Silva y Ponce [13].

3: Enumerar términos importantes en la ontología. Se elaboró una lista de conceptos importantes para el tema. La tabla 2 muestra algunos de ellos.

Tabla 2. Glosario de términos (extracto).

Término	Sinónimo	Significado
Instrumento	Cuestionario	Aquel que plantea una serie de preguntas para extraer determinada información de un grupo de personas.
Teoría		Conjunto organizado de ideas que explican un fenómeno, deducidas a partir de la observación, la experiencia o el razonamiento lógico.
CHAEA	Cuestionario Honey Alonso	Cuestionario sobre Estilos de Aprendizaje que consta de ochenta preguntas (veinte items referentes a cada uno de los cuatro Estilos) a las que hay que responder manifestando acuerdo o desacuerdo.
EstiloAprendizaje	Teoría de estilos de aprendizaje	Los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo los discentes perciben, interrelacionan y responden a sus ambientes de aprendizaje
Item	Pregunta, reactivo	Distintas preguntas que se plantean en una evaluación.
Alumno	PersonaAlumno, discente, estudiante	Que recibe enseñanza.

En la tabla se muestran los conceptos más cercanos a tema principal (estilos de aprendizaje) de la ontología, la cual analiza a los estudiantes a través de un instrumento de evaluación. Los sinónimos integrados corresponden a los diferentes términos utilizados en los artículos de referencia.

4: Definir las clases y la jerarquía de las clases. La Figura 5 muestra algunos conceptos previamente definidos, especificando el tipo de elemento que serán dentro de la ontología (izquierda), además se muestran algunas jerarquías predefinidas.

En la Tabla 3 se muestran los nombres de la relaciones, sus inversas, dominio y rango de cada una de ellas. Estas fueron validadas igualmente por el experto en el dominio. En la tabla sólo se presenta un extracto de 8 relaciones, las cuales integran los conceptos más relevantes, como los tipos de estilos, los autores de la teoría y el instrumento que sirve para detectarlos.

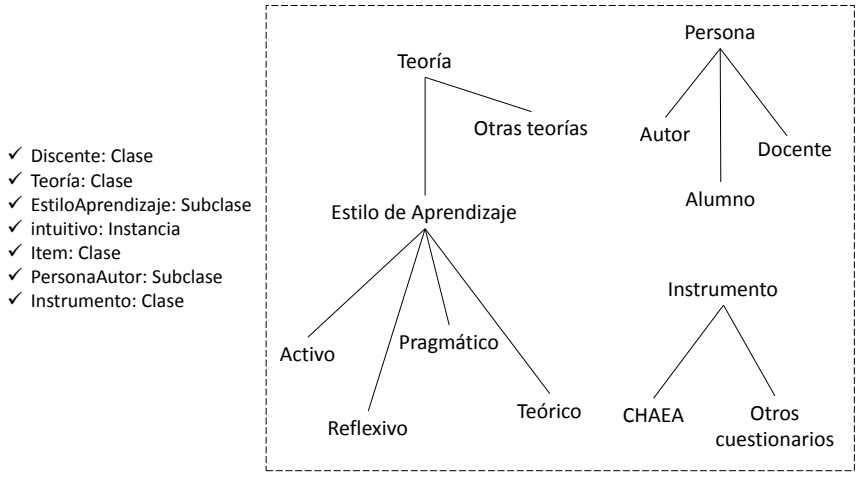


Fig. 5. Taxonomía y especificación de conceptos (extracto)

Tabla 3. Relaciones en ontología de Estilos de Aprendizaje (extracto).

Nombre	Dominio	Rango	Inversa
EsAutorDe	PersonaAutor	Teoria	TieneAutor
EsCoautorDe	PersonaAutor	Teoría	TieneCoautor
EsDescritoPor	EstiloAprendizaje	Característica	DescribeAEstilo
AnalizaA	Instrumento	Teoría	EsAnalizadaPor
TieneCaracterística	PersonaAlumno	Característica	EsCaracterística
TieneEstilo	PersonaAlumno	EstiloAprendizaje	EsEstilo
EsItemDeInstrumento	Item	Instrumento	EstaFormadoPorItem
EsItemDeEstilo	Item	EstiloAprendizaje	TieneItemRelacionado

En el caso de las relaciones inversas, el rango y el dominio cambian, por ejemplo para la relación inversa *DescribeAEstilo*, el rango es *EstiloAprendizaje* y el dominio *Característica*. Otro aspecto importante es que para esta teoría en particular, se tienen dos autores, por lo que es necesaria la relación *EsCoautorDe*.

5 y 6: Definir las propiedades de las clases y valores permitidos. Las propiedades de los datos de muestran en la Tabla 4.

7: Crear instancias de clases individuales en la jerarquía. La Tabla 5 muestra ejemplos de instancias para las clases *autor*, *característica* y *EstiloAprendizaje*.

8: Implementación. La Figura 6 muestra algunos de los conceptos principales con las relaciones entre ellos, omitiendo las relaciones inversas. En esta figura se presentan de manera gráfica los conceptos a fin de tener una idea general de la ontología.

La Figura 7 se presenta un ejemplo de las propiedades de la subclase *EstiloAprendizaje*, específicamente la instancia *Estilo reflexivo*.

Tabla 4. Clases y propiedades de datos de la ontología estilos de aprendizaje (extracto).

Clase	Subclase	Propiedades de datos	Tipo
Característica		TieneNombreCaracterística	String
Instrumento		TieneNombreInstrumento	String
Persona		TieneNombre	String
		TieneGénero	String
	PersonaAlumno	TieneMatrícula	String
		TieneNivelAcadémico	String
		TieneEdad	Integer
	PersonaAutor		
Teoría		TieneNombreTeoría	String
	EstiloAprendizaje	TieneNombreEstilo	String
Item		TieneNumItem	Integer
		TieneTextoItem	String

Tabla 5. Extracto de ejemplos de instancias.

Clase	Atributo	Instancias
Característica	NombreCaracterística	Descubridor Animador Crítico
Autor	NombreAutor	Honey Alonso
EstiloAprendizaje	NombreEstilo	Pragmático Teórico Reflexivo

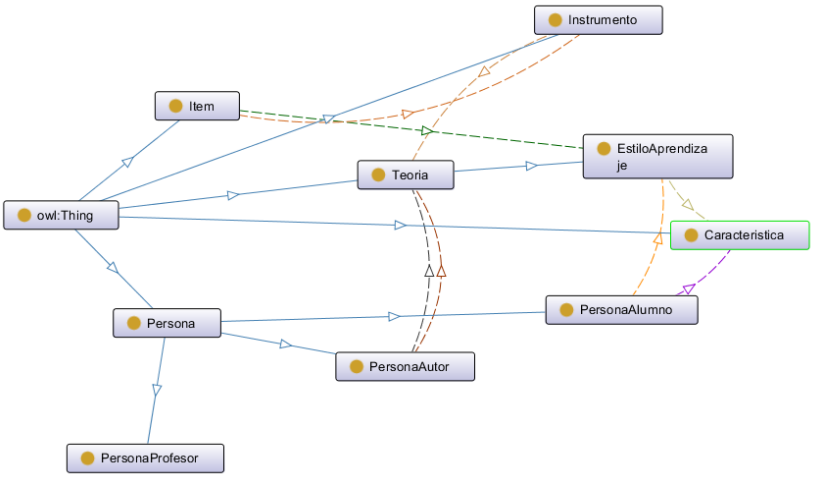


Fig. 6. Grafo representando la ontología de estilos de aprendizaje (extracto).

En el ejemplo se observan las otras instancias de la subclase (los otros tres estilos), así como otras instancias relacionadas de la clase *Característica*. En la parte derecha se observan todas las características que presentan los estudiantes con este tipo de aprendizaje predominante.

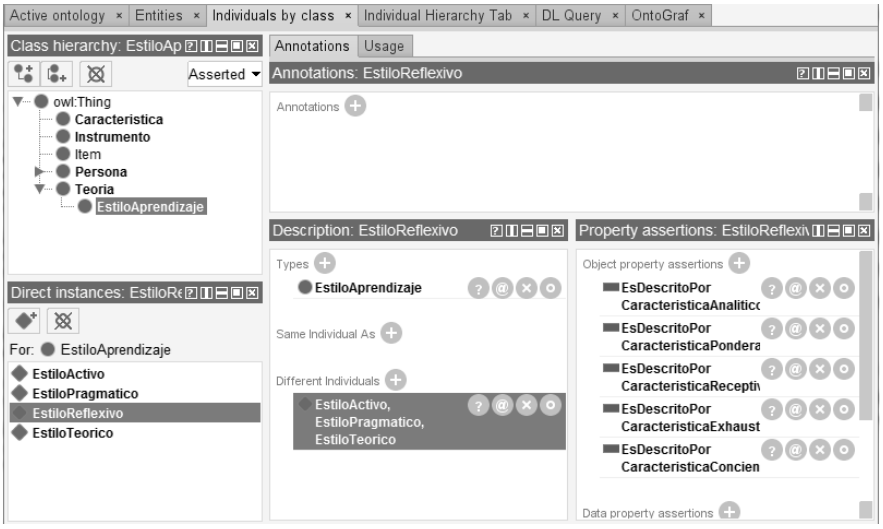


Fig. 7. Ejemplo del uso del razonador en el estilo de aprendizaje Activo.

Algunos otros ejemplos de relaciones entre instancias se listan a continuación:

- $EsAutorDe(Honey, TeoriaEstilosAprendizaje)$
- $EsCoautorDe(Alonso, TeoriaEstilosAprendizaje)$
- $AnalizaA(InstrumentoCuestionarioCHAEA, TeoriaEstilosAprendizaje)$
- $EsDescritoPor(EstiloTeorico, CaracteristicaEficaz)$
- $TieneEstilo(AlumnoEjemplo, EstiloActivo)$
- $EsItemDeInstrumento(Item2, InstrumentoCuestionarioCHAEA)$
- $EsItemDeEstilo(Item10, EstiloReflexivo)$

6. Conclusiones y trabajo en proceso

En este trabajo se presentaron dos herramientas que serán de utilidad para la evaluación de las metodologías y procedimientos realizados enfocados en la detección de clases y relaciones. El conjunto de validación cuenta con elementos importantes para la realización semiautomática de ontologías, mientras que la creación manual de dichas ontologías, permite evaluar los métodos para la extracción de relaciones entre conceptos. La ontología descrita de estilos de aprendizaje contiene los conceptos proporcionados junto con el experto en el dominio. Además, los conceptos integrados en dicha lista se encuentran en el conjunto de texto utilizado en los experimentos realizados con procesos semiautomáticos.

Como trabajo actual se está trabajando en la realización de consultas en la presente ontología, además del diseño de las ontologías de los otros dos tópicos de la metodología: estrategias de enseñanza e inteligencias múltiples. Por último, con estas herramientas se evaluarán exhaustivamente los experimentos realizados en la investigación general.

Referencias

1. Alonso, C., Gallego, D., Honey, P.: Los Estilos de Aprendizaje: Procedimientos de diagnóstico y mejora (2007)
2. Ausubel, D., Novak, J.: Psicología educativa, un punto de vista cognoscitivo. Trillas, México (1983)
3. Cakula, S., Sedleniece, M.: Development of a personalized e-learning model using methods of ontology. Procedia Computer Science 26, 113 – 120 (2013), http://www.sciencedirect.com/science/article/pii/S1877050913012830, iCTE in Regional Development, December 2013, Valmiera, Latvia
4. Chance, P.: Aprendizaje y conducta. Manual moderno (2001)
5. Fan, Z., Apple, F., Horace, I., Jiaheng, C.: Engonto: Integrated Multiple English Learning Ontology for Personalized Education. In: 2008 International Conference on Computer Science and Software Engineering. vol. 5, pp. 210–213. IEEE (2008)
6. Fu, J., Jia, K., Xu, J.: Domain ontology learning for question answering system in network education. In: 2008 The 9th International Conference for Young Computer Scientists. pp. 2647–2652 (2008)
7. Grandbastien, M., Azouaou, F., Desmoulins, C., Faerber, R., Leclet, D., Quenu-Joiron, C.: Sharing an ontology in education: Lessons learn from the OURAL project. In: Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007). pp. 694–698 (2007)
8. Herrmann, N.: The creative Brain.NASSP Bulletin, vol. 66 (1982)
9. Kolb, D.: Learning style inventory. MA: Hay Group, Hay Resources Direct, Boston USA (1976)
10. McCarthy, B.: The 4MAT System: Teaching to Learning Styles with Right/left Mode Techniques. EXCEL (1987), https://books.google.com.mx/books?id=WWXVSgAACAAJ
11. Méndez, N.D.D., Carranza, D.A.O., Ocampo, M.G.: Representación ontológica de perfiles de estudiantes para la personalización del aprendizaje. Revista Educación en Ingeniería 10(19), 105–115 (2015)
12. Noy, N.F., Mcguinness, D.L.: Ontology development 101: A guide to creating your first ontology. Tech. rep., Stanford Knowledge Systems Laboratory (2001)
13. Silva Sprock, A., Ponce, J.: Reingeniería de una ontología de estilos de aprendizaje para la creación de objetos de aprendizaje. Eduweb 7, 49–64 (2013)
14. Sánchez López, S.E.: Modelo de indexación de formas en sistemas VIR basado en ontologías. Ph.D. thesis, Escuela de ingeniería y Ciencias. Universidad de las Américas Puebla (2007)
15. Uskov, V., Pandey, A., Bakken, J.P., Margapuri, V.S.: Smart engineering education: The ontology of internet-of-things applications. In: 2016 IEEE Global Engineering Education Conference (EDUCON). pp. 476–481 (April 2016)
16. Weigand, H.: A multilingual ontology-based lexicon for news filtering-the TREVI project. In: Proceedings of the IJCAI Workshop on Multilingual Ontologies-Nagoya (1997)
17. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: A look back and into the future 44, 1–36 (2011)
18. Wu, H.: Research of internet education system based on ontology. In: 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery. vol. 4, pp. 602–605 (2008)
19. Zhu, F., Yao, N.: Ontology-based learning activity sequencing in personalized education system. In: 2009 International Conference on Information Technology and Computer Science. vol. 1, pp. 285–288. IEEE (2009)

ISSN 1870-4069

Propuesta para determinar la ruta más corta utilizando la arquitectura M-ANFIS

Eduardo Chandomí-Castellanos, Elías N. Escobar-Gómez,
Sabino Velázquez-Trujillo, J. Alexander López-Rosas,
Carlos Venturino De-Coss-Pérez, Yordi López-Escalante,
Daniel Domínguez-Romero, Alejandro Flores-Méndez

Tecnológico Nacional de México,
I.T. Tuxtla Gutiérrez,
México

eduardo.chandomi@ittuxtlagutierrez.edu.mx,
{elias.eg, sabino.vt, carlos.dp }@tuxtla.tecnm.mx,
{Alexander.lr, yordi_lopez}@outlook.com,
daniel_dr98.hotmail.com, Alejandrofm21@live.com

Resumen. Este trabajo propone un método para resolver el problema de la ruta más corta, considerando variables imprecisas. El método se basa en un esquema de lógica difusa y una arquitectura adaptable con redes neuronales. Para considerar la incertidumbre, en el modelo se toma en cuenta el efecto de tres variables: Estado de las calles, Zonas de tráfico y Precipitación pluvial (intensidad de lluvia) junto a una arquitectura adaptable con redes neuronales. Las funciones de pertenencia de estas variables se calculan de manera experimental, obteniendo tiempos de recorrido más cercanos a los reales. El modelo evalúa la incertidumbre para cada una de las aristas de la red (calles), con el objetivo de ajustar el tiempo de recorrido de las rutas establecidas. Para hacer adaptativo el modelo se emplea un sistema de inferencia difuso ANFIS (Adaptive network based Fuzzy Inference System) Mandani, que aproxima el resultado a un valor deseado, finalmente el modelo actualiza y modifica las funciones de pertenencia haciendo adaptable a nuevos escenarios. Para la validación del modelo M-ANFIS se comparan los tiempos de recorrido obtenidos con el modelo propuesto y los obtenidos con lógica difusa con los tiempos de recorrido obtenidos en un muestreo.

Palabras clave: Neurodifuso, lógica difusa, redes neuronales artificiales, ruta más corta, ajuste de tiempo.

Proposal for Determining Shortest Route using M-ANFIS Architecture

Abstract. This work proposes a method to solve the shortest path problem with imprecise variables. The method is based on a fuzzy logic

scheme and an adaptive architecture with neural networks. For the uncertainty ,the effect of three variables for each street is considerate: state of the streets, traffic zones and rainfall (intensity of rain) with an adaptable neural networks architecture.The membership functions are experimentally calculated, getting times closer to the real. the model evaluates the uncertainty for each of the edges of the network (streets), with the objective to adjust set route travel time. The model use Mandani Adaptive network based fuzzy inference system, which approximates the result to a expected value, finally the model update and modify the membership functions making it adaptable to new scenarios. For the validation is compared the travel time M-ANFIS model with fuzzy logic travel time and desired travel time.

Keywords: Neuro fuzzy, fuzzy logic, artificial neural network, shortest path, time adjust.

1. Introducción

Los costos de un producto generalmente varían dependiendo del tipo de producto que se va a comercializar, el proceso de manufactura y las condiciones del proceso de distribución, siendo el proceso de distribución quien genere costos del 30 % del costo de producción. En la actualidad, las empresas invierten en máquinas, procesos y métodos que aseguren un producto de calidad, que sea entregado al cliente en tiempo y forma. Esto genera una necesidad en las empresas por buscar alternativas que mejoren la entrega de dichos productos, siendo el primer objetivo que lleguen al lugar adecuado en el tiempo determinado [17], en buenas condiciones y a un costo competitivo [12]. Las mejores soluciones de optimización de ruta ayudan a las empresas de reparto a minimizar el tiempo de conducción y reducir el consumo de combustible.

La inteligencia artificial es una herramienta que nos ayuda a resolver probleperte difíciles de modelar. Métodos heurísticos como la lógica difusa o redes neuronales son alternativas con más aplicaciones en el área de ruta más corta [14,1,9,18,5]. En la literatura existen trabajos que presentan redes neuronales para resolver probleperte de trasporte [7] y planeación de ruta de robots [19,15], entre otros. La unión de estas dos técnicas dan como resultados métodos que adaptan sus parámetros, llamados Neurodifusos [2,11,16,10]. El objetivo principal de este trabajo es aplicar una red Neurodifusa donde las funciones de pertenencia se adaptan y se ajustan a los nuevos escenarios dando como resultado la optimización de la ruta de reparto. Generando para ello nuevas alternativas de recorrido asegurando las entregas oportunas, es decir entregar la mercancía en el horario que se establece. Adicionalmente obtener beneficios como la reducción de gasto de combustible, disminución del mantenimiento del vehículo ocasionado por el desgaste y el recorrido de rutas en mal estado, en igual manera como empresa manejar la política y garantizar la entrega oportuna.

2. Definiciones

2.1. Ruta más corta

El problema de la ruta más corta es quizás el más simple de todos los problemas de flujo de red. Para este problema, deseamos encontrar una ruta de costo mínimo (o longitud) desde un nodo fuente específico "s" a otro nodo receptor especificado "t", asumiendo que cada arco $(i, j) \in A$ tiene un costo asociado (o longitud) c_{ij}. Algunas de las aplicaciones del problema de la ruta más corta son determinar una ruta entre dos nodos especificados de una red que tenga una longitud mínima, o una ruta que tarde menos en atravesar, o una ruta que tenga la máxima confiabilidad.

2.2. Neruodifuso

El Sistema de Inferencia Neurodifuso Adaptativo ANFIS (por sus siglas en inglés, $Artificial\ Neural\ Fuzzy\ Inference\ System$), es una combinación de dos métodos de computación blanda: las Redes Neuronales Artificiales (RNA) y Lógica Difusa (LD) [8]. La Lógica Difusa tiene la capacidad de cambiar los aspectos cualitativos del conocimiento humano y las ideas en el proceso de análisis cuantitativo. Por otro lado, las RNA's, tiene una mayor capacidad en el proceso de aprendizaje, para adaptarse a su entorno.

Las RNA's se utilizan para ajustar automáticamente las funciones de pertenencia partir de datos o para enriquecerlo mediante aprendizaje a partir de ejemplos. Esto reduce la tasa de errores en la determinación de reglas en lógica difusa. El término de "sistemas Neurodifusos"se aplica a sistemas que tengan las siguientes propiedades:

- Se entrenan con un algoritmo de aprendizaje que modifica información del mismo sistema difuso. El proceso de aprendizaje no se basa en conocimiento, sino en un conjunto de datos dado.
- El sistema Neurodifuso toma la forma de una red neuronal multicapa, donde las unidades en sus nodos utilizan operaciones T-norma o T-conorma sustituyendo las funciones de activación que normalmente emplea las redes neuronales. La primera capa representa las variables de entrada. La segunda contiene las funciones de pertenencia y las reglas difusas que representan la experiencia del experto mediante conexiones y capas ocultas. La última capa representa las variables de salida.
- Aproxima una función de n dimensiones, definida parcialmente de datos de entrenamiento. Las reglas difusas codificadas dentro del sistema representan muestras difusas y se interpretan como patrones datos de entrenamiento.

Los sistemas difusos y de red neuronal trabajan juntos en una arquitectura homogénea híbrida, interpretada como una red neuronal con parámetros difusos o como un sistema difuso con parámetros o funcionamiento distribuidos.

3. Método propuesto

En esta investigación se aplica un sistema Neurodifuso que combina las redes neuronales con las técnicas de inferencia difusa. La metodología de los sistemas ANFIS proporcionan alternativas de solución con métodos cualitativos y cuantitativos en la construcción de modelos.

3.1. Variables de entrada

Para la identificación de factores que reducen el tiempo de reparto se realiza un estudio a operarios de vehículos de comida rápida, el estudio detecta 52 factores. En dicho el análisis se determinan aquellos cuyo comportamiento es incierto, con mayor frecuencia y con una relación importante en el proceso de reparto. El modelo propuesto consta de tres variables, que modifican el tiempo de viaje de cada arista de la red, descritos a continuación.

- Estado de la calle (E), es definida por la cantidad y tamaño de obstáculos que varía la velocidad del vehículo, es considerada difusa interpretando un nivel para cada estado de la calle.
- Zona con tráfico (Z), se consideran ciertas zonas que generan más tráfico al momento de transitar, debido a la proximidad de zonas escolares, centros comerciales, entre otros.
- Precipitación pluvial (P), la intensidad de lluvia es la cantidad de agua que se precipita en el área donde se conduce, la que provoca una disminución de visibilidad y de velocidad del vehículo; esta variable está medida en ml/m^2.

3.2. Modelo difuso

El sistema difuso y las funciones de pertenencia calculadas de manera experimental, fueron presentados en el Congreso Mexicano de Inteligencia Artificial (COMIA) 2019 y publicados como número especial de la revista Research in Computing Science, ISSN: 1870-4069, indexada en Latinindex y DBLP [3].

3.3. Selección del tipo de arquitectura multicapa

Los modelos ANFIS integran información incluida dentro de un conjunto de datos con el conocimiento de expertos, expresados en forma lingüística. Se utiliza una arquitectura basada en el mecanismo de inferencia tipo mamdani (M-ANFIS) [6], Ecuación 1:

$$Regla\ k = \textbf{si}\ P\ \textbf{es}\ M_{i,j},\ E\ \textbf{es}\ M_{i,j}\ \textbf{y}\ Z\ \textbf{es}\ M_{i,j}\ \textbf{entonces}\ D_k = M_{ok}(\mu; AD_{ok}). \tag{1}$$

En la Figura 1 se ilustra la arquitectura FIS de Mamdani[13]. La base de reglas en este trabajo se escribe en la Ecuación 2.

$$
\begin{array}{l}
1.\ \textbf{si}\ (P\ es\ M_{11}),\ (E\ es\ M_{21}),\ (Z\ es\ M_{31}),\ \textbf{entonces}\ D_1 = M_{o1}(\mu; AD_{o1}),\\
2.\ \textbf{si}\ (P\ es\ M_{11}),\ (E\ es\ M_{21}),\ (Z\ es\ M_{32}),\ \textbf{entonces}\ D_2 = M_{o2}(\mu; AD_{o2}),\\
3.\ \textbf{si}\ (P\ es\ M_{11}),\ (E\ es\ M_{21}),\ (Z\ es\ M_{32}),\ \textbf{entonces}\ D_3 = M_{o3}(\mu; AD_{o3}),\\
\vdots,\\
53.\ \textbf{si}\ (P\ es\ M_{16}),\ (E\ es\ M_{22}),\ (Z\ es\ M_{33}),\ \textbf{entonces}\ D_{53} = M_{o53}(\mu; AD_{o53}),\\
54.\ \textbf{si}\ (P\ es\ M_{16}),\ (E\ es\ M_{23}),\ (Z\ es\ M_{31}),\ \textbf{entonces}\ D_{54} = M_{o54}(\mu; AD_{o54}),
\end{array}
\tag{2}
$$

donde: P, E, Z, corresponde a las variables de entrada, Precipitación pluvial, Estado de la calle y Zona con tráfico respectivamente. M_{ij} representan la $j-esima$ función de pertenencia de la $i-esima$ entrada, D_k y M_{ok} la salida de la $k-esima$ regla, y la $k-esima$ FM de salida. AD_{ok} son los parámetros consecuentes que caracterizan las formas de las funciones de pertenencia de salida, llamada Ajuste Difuso.

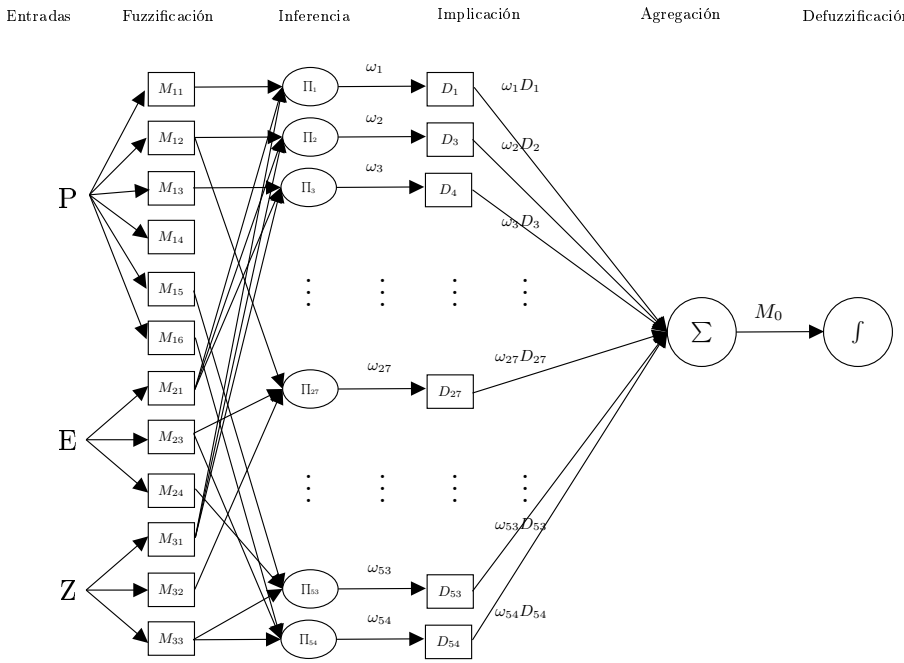


Fig. 1. propuesta de la arquitectura del sistema de inferencia difuso de Mamdani

3.4. Desarrollo de la arquitectura tipo Mamdani

El modelo ANFIS Mamdani consta de 5 capas (Layers), cada salida de la capa se describe a continuación:

Capa1 (Fuzzificación): Las entradas de esta capa P, E, Z representan las entradas reales del sistema. La salida representa el grado de pertenencia para el cual la variable de entrada satisface el término lingüístico. En este trabajo se utilizan funciones de pertenencia calculadas de manera experimental, descritos en las Ecuaciones 3-7 [3].

No realizan ningún tipo de procesamiento sobre los valores de entrada de las variables, simplemente se encargan de distribuir estos valores a la capa siguiente:

i) Gaussianas para las entradas $M_{i,j}$ para $(i = 1),(j = 2,3,4,5)$; $(i = 2),(j = 2)$; $(i = 3),(j = 2)$:

$$\mu_{\{M_{i,j}\}}(x_i) = \begin{cases} e^{-k_j(x_i - m_j)^2}, & 0 \leq x_i \leq b_j, \\ 0, & b_j > x_i. \end{cases} \tag{3}$$

ii) Pseudo-exponencial para la entrada M_{ij} $(i = 2)$, $(j = 3)$; $(i = 3),(j = 1,3)$:

$$\mu_{\{M_{i,j}\}}(x_i) = \begin{cases} \frac{1}{1+k_j(x_i - n_j)^2}, & a \leq x_i \leq b_j, \\ 0, & x_i < a_j. \end{cases} \tag{4}$$

iii) Función de pertenencia Singleton para la entrada $M_{1,1}$:

$$\mu_{\{M_{11}\}}(x_i; a_j) = \begin{cases} 0, & x_i = a_j, \\ 1, & x_i \neq a_j. \end{cases} \tag{5}$$

iv) Gamma para la función de pertenencia, $M_{1,6}$:

$$\mu_{\{M_{1,6}\}}(x_i) = \begin{cases} 1 - e^{-k_j(x_i - p_j)^2}, & d_j \leq x_i \leq e_j, \\ 0, & 0 < d_j - . \end{cases} \tag{6}$$

v) La función Sigmoidal corresponde únicamente a la entrada $M_{2,3}$:

$$\mu_{\{M_{2,3}\}}(x_i) = \begin{cases} \frac{1}{1+e^{-k_j(x_i - q_j)}}, & a_j \leq x_i \leq b_j, \\ 0, & x_i < b_j, \end{cases} \tag{7}$$

donde a_j, b_j, d_j, e_j, k_j, m_j, n_j, p_j, q_j son los parámetros que caracterizan a las funciones de pertenencia de cada variable de entrada.

Capa 2 (Capa de Inferencia o Capa de Reglas): En esta capa cada nodo le corresponde una etiqueta lingüística (bueno, regular, malo entre otros), para cada una de las variables de entrada de cada regla se calcula ω_k, considerado como el factor de ponderación o fuerza de disparo, se determina evaluando las expresiones de pertenencia de la regla antecedente.

Los valores de entrada o valores reales se transforman en valores difusos usando las funciones de cada variable de la capa 1, Ecuación 8.

$$\begin{aligned}
\omega_1 &= \mu_{M11}(P)\,\mu_{M21}(E)\,\mu_{M31}(Z),\\
\omega_2 &= \mu_{M12}(P)\,\mu_{M21}(E)\,\mu_{M31}(Z),\\
\omega_3 &= \mu_{M13}(P)\,\mu_{M21}(E)\,\mu_{M31}(Z),\\
&\vdots\\
\omega_{53} &= \mu_{M15}(P)\,\mu_{M23}(E)\,\mu_{M32}(Z),\\
\omega_{54} &= \mu_{M16}(P)\,\mu_{M23}(E)\,\mu_{M33}(Z).
\end{aligned} \tag{8}$$

Capa 3 (Implicación o Antecedente): Un consecuente es un conjunto difuso representado por una función de pertenencia $Z = D_i$. Existen métodos conocidos, dos de los más usado son: el método "**y**"(mínimo), que trunca el conjunto difuso de salida, y el PROD (producto), el cual escala al conjunto difuso de salida.

El grado de verdad de la premisa de cada regla es computada y aplicada al consecuente de cada regla. Este resultado es un subconjunto difuso que será asignado a cada variable de salida correspondiente. Se aplica el método de Mamdani o método de inferencia MIN-MAX. Para cada regla se hace el MIN entre el nivel de disparo y la función de pertenencia de la variable de salida indicada en el consecuente. En esta capa, la implicación de cada salida FM se calcula con la Ecuación 9:

$$M_{imp,k} = \omega_k D_k, \qquad k = 1, 2, 3, \ldots, 53, 54, \tag{9}$$

donde, $M_{imp,k}$ representa la implicación de la función de pertenencia de salida.

Capa 4 (Agregación o Consecuente): En esta capa, la agregación se realiza para producir una salida general, $M_o(u)$, utilizando el operador de unión o máximo, Ecuación 10:

$$M_o(u) = \sum_{k=1}^{54} M_{imp,k}. \tag{10}$$

Sustituyendo la Ecuación 9 en 10, se obtiene 11:

$$M_o(u) = \sum_{k=1}^{54} \omega_k D_k, \tag{11}$$

donde, D_k representa la salida "**entonces**", el cual describe una función de pertenencia, remplazando la función de pertenencia pseudo-exponencial en la Ecuación 11 se obtiene 12:

$$M_o(u) = \sum_{k=1}^{54} \omega_k e^{-k(x-m)^2}. \tag{12}$$

Capa 5 (Combinación o Defuzificación): La salida es calculada utilizando el método de centroide o centro de gravedad (COG) que retorna el centro del área bajo la curva, descrita en la Ecuación 13:

$$o = \frac{\int M_o(u)\, u du}{\int M_o(u)\, du}. \tag{13}$$

3.5. Actualización de pesos

La determinación de los parámetros del sistema de inferencia difusa es de vital importancia. Se utiliza un algoritmo de optimización para determinar estos parámetros, se realiza de tal forma que se minimiza la medida de error entre el destino y la salida real.

Durante el proceso de aprendizaje del FIS, los parámetros de la premisa y consecuentes se sintonizan hasta que se alcanza la respuesta deseada del FIS.

Los ANFIS utilizan sistemas de aprendizaje híbridos, dichos métodos son del gradiente descendiente y el estimador de mínimos cuadrados.

Más específicamente, la regla realiza la minimización del error al cuadrado, entre la entrada u y la salida d con el objetivo de ajustar el vector de peso $w = [\theta, w_1, w_2, ...w_n]^T$ de la red. En resumen, el objetivo es obtener un peso w^* óptimo para que el error al cuadrado $\{E(w^*)\}$ de todo el conjunto de muestras sea lo más bajo posible. En [4] se afirma la notación matemática, considerando una configuración de peso óptima, Ecuación 14:

$$E(w^*) \leq E(w), \qquad \forall w \in \Re^{n+1}. \tag{14}$$

La función del error al cuadrado relacionado con las p muestras de entrenamiento se define en la Ecuación 15:

$$E(w) = \frac{1}{2} \sum_{k=1}^{P} \left(d^k - u\right)^2. \tag{15}$$

Sustituyendo el resultado de la Ecuación 15 requerida para obtener la salida de una neurona ADALINE, Ecuaciones 16 y 17:

$$E(w) = \frac{1}{2} \sum_{k=1}^{P} \left(d^k - \left(\sum_{k=1}^{n} w_i x_i^k - \theta\right)\right)^2, \tag{16}$$

$$E(w) = \frac{1}{2} \sum_{k=1}^{P} \left(d^k - \left(w^k x_i^k - \theta\right)\right)^2. \tag{17}$$

Así, la expresión de la Ecuación 17 calcula el error cuadrático medio para las p muestras de entrenamiento proporcionadas para el proceso de entrenamiento del ADALINE. El siguiente paso consiste en la aplicación del operador de gradiente en el error cuadrático medio con respecto al vector w, con el fin de buscar un valor óptimo para la función de error cuadrático, es decir, Ecuación 18:

$$\nabla E\left(w\right) = \frac{\partial E\left(w\right)}{\partial w}. \tag{18}$$

Sustituyendo 17 en 18 se obtienen 19 y 20.

$$\nabla E\left(w\right) = \sum_{k=1}^{P} \left(d^k - \left(w^T x_i^k - \theta\right)\right)\left(-x^k\right), \tag{19}$$

$$E\left(w\right) = \frac{1}{2}\sum_{k=1}^{P}\left(d^k - \left(\sum_{k=1}^{n} w_i x_i^k - \theta\right)\right)^2. \tag{20}$$

Finalmente, los pasos para adaptar el vector de peso se deben ejecutar en la dirección opuesta del gradiente, el objetivo de optimización es minimizar el error cuadrático. La variación D_w para actualizar el vector de peso ADALINE viene dada por la Ecuación 21:

$$\nabla E\left(w\right) = -\sum_{k=1}^{P}\left(d^k - u\right)x^k. \tag{21}$$

Insertando (20) en (21) se obtiene (22):

$$\nabla w = \eta \nabla E\left(w\right). \tag{22}$$

De manera complementaria, se expresa (22) como (23):

$$w^{actual} = w^{anterior} + \eta\sum_{k=1}^{P}\left(d^k - u\right)x^k. \tag{23}$$

Simplificando la expresión, w se actualiza después de presentar cada $k-esima$ muestra de entrenamiento, Ecuación 24:

$$w^{actual} = w^{anterior} + \eta\left(d^k - u\right)x^k, \tag{24}$$

donde:
$w = [\theta, w_1, w_2, \ldots w_n]^T$, es el vector que contiene el umbral y los pesos. $x^k = [-1, x_1^k, x_2^k, \ldots x_n^k]^T$, es la $k - esima$ muestra de entrenamiento. d^k, es la la $k - esima$ muestra de entrenamiento. u, es el valor de salida del elemento lineal. η, es una constante que define la tasa de aprendizaje.

Se procede a indicar los pasos a seguir para la implementación de esta regla.

1. Se inicializan los parámetros de cada función de pertenencia con valores pequeños al azar.
2. Se selecciona el par de entrenamiento y se aplica como vector de entrada a la red.
3. Se calcula la salida de la red.
4. Se calcula el error entre la salida de la red y la salida deseada, y se acumula el valor en el error total.
5. Se repiten los pasos del 1 al 4 por cada vector en el conjunto de entrenamiento hasta que el error del conjunto sea suficientemente bajo.

4. Análisis de resultados

Una red adaptable es una estructura de red con un comportamiento general de entrada-salida y está determinado por una colección de parámetros modificables. El modelo se adapta cuando los parámetros se modifican y se ajustan hasta obtener un valor lo bastante aproximado a la salida real.

Este mapeo no lineal está regulado por un conjunto de datos de entrada-salida deseados de un sistema. Este conjunto de datos se denomina conjunto de datos de entrenamiento así mismo los procedimientos que ajustan los parámetros para mejorar el rendimiento de la red se denominan reglas de aprendizaje.

Una regla de aprendizaje explica cómo se deben actualizar estos parámetros (o pesos) para minimizar una medida de error predefinida. En la Figura 2 se observa la comparación entre la salida estimada con el método M-ANFIS y la salida muestreada de una ruta seleccionada de 15 calles de la cuidad de Tuxtla Guitérrez, Chiapas; se observa puntos donde la diferencia mínima es de 0.007 (nodo 6) y la diferencia máxima es de 0.19 (nodo 14), siendo la mayoría menor que 0.1. El modelo ajusta estos parámetros hasta obtener resultados aproximados a los deseados, minimizando el error entre ambos conjuntos, esta medida de error calcula la discrepancia entre la salida real de la red y una salida deseada (Figura 3), entre más cercano esté a cero mejor será el resultado.

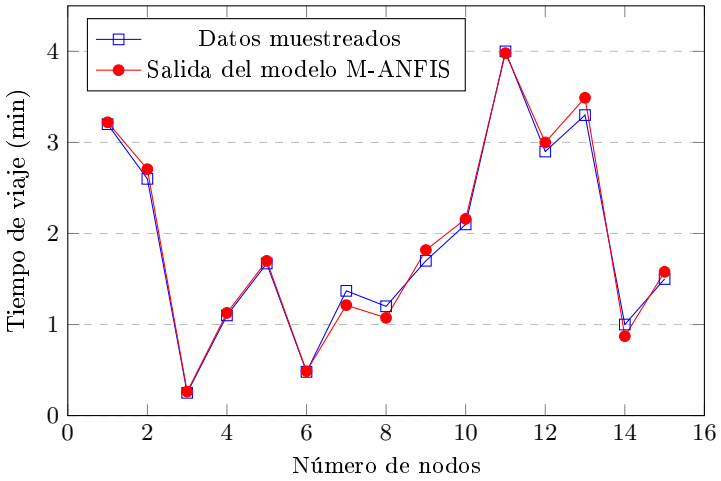


Fig. 2. Datos muestreados vs salida adaptativa

El modelo propuesto aproxima el valor de salida difusa con el valor muestreado del tiempo recorrido. El modelo aprende a partir de un vector de entrada con datos de recorrido, un vector de salida con tiempos estimados por el modelo, 1000 épocas con un mínimo de error de 0.01 y una tasa de aprendizaje de 0.001.

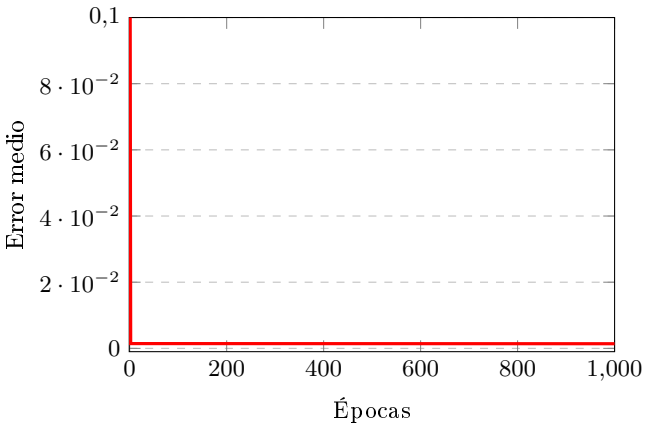


Fig. 3. Convergencia a cero del error cuadrático medio

En la Tabla 1 se muestran los tiempos estimados con lógica difusa comparadas con los tiempos estimados aplicando el modelo M-ANFIS. Los resultados de ambos modelos se comparan con los tiempos obtenidos en el muestreo. Esto con el objetivo de comparar el error porcentual absoluto medio de los modelos con Lógica Difusa y el error porcentual absoluto medio del modelo M-ANFIS. Se observa en la Tabla 2 las medias de los errores porcentuales de LD con 25.06 % y 5.14 % de M-ANFIS, las varianzas de 11 % y 0.14 % respectivamente, comprobando así que existe menos variabilidad con el modelo M-ANFIS.

Una vez calculado el error se ajustan los pesos, que son parámetros que modifican las funciones de pertenencia del modelo, a continuación se muestra la diferencia de las funciones de pertenencia antes de aprender y después, ver Figura 4(a) y 4(b). Como en el caso de estudio no hubo lluvias, no se activaron reglas de dicha variable y no se vieron modificadas en la investigación.

5. Conclusiones y recomendaciones

En conclusión la combinación de sistemas adaptables y su cualidad de ajuste de parámetros con el conocimiento de expertos en forma de reglas del tipo "**si... entonces**", dado un conjunto de pares de datos de entrada y salida, se adapta y aproxima los valores a valores cercanos a los reales. Estos sistemas son convenientes para repetir el comportamiento de un sistema no lineal cuyo modelo matemático no es obtenido fácilmente.

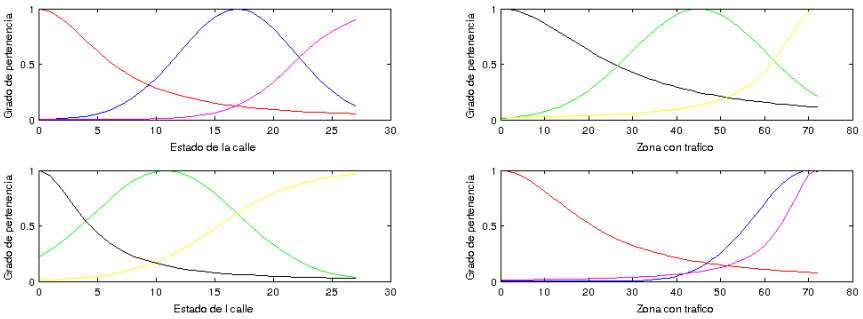
Las variables difusas utilizadas en este trabajo fueron elegidas por su frecuencia y su ocurrencia de aparición, en el estudio se encontraron más variables que afectan ocasionalmente el tráfico y que contienen un impacto en el tiempo de recorrido, para futuros trabajos se recomienda evaluar esos factores e integrarlos en un sistema que evalúe esas y otras variables consideradas en entornos diferentes.

Tabla 1. Comparación de la salida difusa con la salida adaptativa.

	Variables Difusas			Valor Real (min)	Lógica Difusa		M-ANFIS	
	P	E	Z		Salida	Diferencia	Salida	Diferencia
1	0	1	1	3.3	2.9	0.4	3.2213	0.0787
2	0	7	1	2.6	2.552	0.048	2.706	0.106
3	0	1	21	0.25	0.2625	0.0125	0.263	0.013
4	0	1	21	1.1	0.57	0.53	1.128	0.028
5	0	1	1	1.67	1.695	0.025	1.699	0.029
6	0	1	1	0.48	1.1078	0.6278	0.487	0.007
7	0	1	1	1.37	0.87	0.5	1.213	0.157
8	0	1	1	1.2	1.088	0.112	1.073	0.127
9	0	1	1	1.7	1.856	0.156	1.818	0.118
10	0	1	1	2.1	1.5296	0.5704	2.16	0.06
11	0	1	42	4	3.977	0.023	3.98	0.02
12	0	23	21	2.9	3.2038	0.3038	3	0.1
13	0	14	1	3.3	4.8361	1.5361	3.49	0.19
14	0	1	21	1	1.1625	0.1625	0.8715	0.1285
15	0	1	21	1.5	1.8075	0.3075	1.579	0.079
Total				28.47	29.4178		28.6888	

Tabla 2. Datos para comparar el error porcentual absoluto medio.

	Tiempo de recorrido muestreado	Tiempo con LD	Porcentaje de error con LD	Tiempo con M-ANFIS	Porcentaje de error con M-ANFIS
1	3.3	2.9	12 %	3.2213	2 %
2	2.6	2.552	2 %	2.706	4 %
3	0.25	0.2625	5 %	0.263	5 %
4	1.1	0.57	48 %	1.128	3 %
5	1.67	1.695	1 %	1.699	2 %
6	0.48	1.1078	131 %	0.487	1 %
7	1.37	0.87	36 %	1.213	11 %
8	1.2	1.088	9 %	1.073	11 %
9	1.7	1.856	9 %	1.818	7 %
10	2.1	1.5296	27 %	2.16	3 %
11	4	3.977	1 %	3.98	1 %
12	2.9	3.2038	10 %	3	3 %
13	3.3	4.8361	47 %	3.49	6 %
14	1	1.1625	16 %	0.8715	13 %
15	1.5	1.8075	21 %	1.579	5 %
	Media		25.06 %		5.14 %
	Varianza		11.0 %		0.14 %



(a) Estado de la calle (b) Zona con tráfico

Fig. 4. Funciones de pertenencia antes y después del aprendizaje

Opcionalmente se recomienda comparar los resultados del sistema M-ANFIS con otros sistemas de adaptabilidad como lo son algoritmos genéticos y evaluar la eficiencia que alguno de los dos tendría en un sistema informático, a diferencia de las RNA's los Algoritmos Genéticos son utilizados en la resolución de problemas de búsqueda y optimización reproduciendo genéticamente una población a lo largo de una serie de generaciones.

Referencias

1. Arnold, W., Hellendoorn, H., Seising, R., Thomas, C., Weitzel, A.: Fuzzy routing. Fuzzy Sets and Systems 85(2), 131–153 (1997)
2. Chai, Y., Jia, L., Zhang, Z.: Mamdani model based adaptive neural fuzzy inference system and its application. International Journal of Computational Intelligence 5(1), 22–29 (2009)
3. Chandomí-Castellanos, E., Escobar-Gómez, E.N., Velázquez-Trujillo, S., De León, H.R.H., Pérez-Patricio, M., Pérez, C.V.D.C., Gutiérrez, T.: Modelo para la determinación de la ruta más corta con funciones experimentales para arcos difusos. Research in Computing Science 148, 317–330 (2019)
4. Da Silva, I.N., Spatti, D.H., Flauzino, R.A., Liboni, L.H.B., dos Reis Alves, S.F.: Artificial Neural Networks: A Practical Course. Springer (2016)
5. Escobar-Gómez, E., Camas-Anzueto, J., Velázquez-Trujillo, S., Hernández-de León, H., Grajales-Coutiño, R., Chandomí-Castellanos, E., Guerra-Crespo, H.: A linear programming model with fuzzy arc for route optimization in the urban road network. Sustainability 11(23), 6665 (2019)
6. Guney, K., Sarikaya, N.: Comparison of mamdani and sugeno fuzzy inference system models for resonant frequency calculation of rectangular microstrip antennas. Progress In Electromagnetics Research 12, 81–104 (2009)
7. Ida, K., Gen, M., Li, Y.: Neural networks for solving multicriteria solid transportation problem. Computers & industrial engineering 31(3-4), 873–877 (1996)
8. Jang, J.S.: Anfis: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics 23(3), 665–685 (1993)

9. Ji, X., Iwamura, K., Shao, Z.: New models for shortest path problem with fuzzy arc lengths. Applied Mathematical Modelling 31(2), 259–269 (2007)
10. Kim, S.W., Park, S.Y., Park, C.: Spacecraft attitude control using neuro-fuzzy approximation of the optimal controllers. Advances in Space Research 57(1), 137–152 (2016)
11. Kiran, T.R., Rajput, S.: An effectiveness model for an indirect evaporative cooling (iec) system: Comparison of artificial neural networks (ann), adaptive neuro-fuzzy inference system (anfis) and fuzzy inference system (fis) approach. Applied Soft Computing 11(4), 3525–3533 (2011)
12. Kotler, P., Armstrong, G., Brown, L.G., McEnally, M.R., Paczkowski, T.J.: Principles of Marketing: Company Case & Video Commentaries. Prentice Hall (1991)
13. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International journal of man-machine studies 7(1), 1–13 (1975)
14. Okada, S., Gen, M.: Fuzzy shortest path problem. Computers & industrial engineering 27(1-4), 465–468 (1994)
15. Orozco-Soto, S.M., Fernández, J.C.R.: Control par calculado difuso basado en pasividad para seguimiento de trayectorias de robots manipuladores. Research in Computing Science 91, 131–141 (2015)
16. Suparta, W., Alhasa, K.M.: Modeling of tropospheric delays using anfis. SpringerBriefs in Meteorology (2016)
17. Visser, J., Nemoto, T., Browne, M.: Home delivery and the impacts on urban freight transport: A review. Procedia-social and behavioral sciences 125, 15–27 (2014)
18. Wei, D.c.: Implementation of route selection function based on improved floyd algorithm. In: 2010 WASE International Conference on Information Engineering. vol. 2, pp. 223–227. IEEE (2010)
19. Yang, S.X., Luo, C.: A neural network approach to complete coverage path planning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34(1), 718–724 (2004)

ISSN 1870-4069

Clasificación del género de peatones y conductores basada en su comportamiento en la vía pública

Francisco Fernando Torres Rosas, Maricela Quintana López,
Héctor Rafael Orozco Aguirre, Saul Lazcano Salas

Universidad Autónoma del Estado de México,
Centro Universitario UAEM Valle de México,
México

fer04.torres92@gmail.com,
{mquintanal,hrorozcoa,slazcanos}@uaemex.mx

Resumen. Los accidentes vehiculares representan un problema de salud pública a nivel mundial, los traumatismos por accidentes de tránsito son la primera causa de muerte en niños y adultos jóvenes. En México, a través de la estadística de accidentes elaborada por el Instituto Nacional de Estadística y Geografía se logra tener un panorama de los accidentes viales en el ámbito nacional, así como las consecuencias humanas y materiales que conlleva. Sin embargo, no se tiene a detalle cuáles fueron los factores o comportamientos de los peatones o conductores, que contribuyeron a que el accidente ocurriera. En este artículo, se presenta la clasificación del género basado en el comportamiento que los actores viales, peatón y conductor, exhiben dentro de la vía pública. Para ello, se usa el algoritmo J48 que proporciona árboles de decisión que permiten examinar el conocimiento adquirido. El mejor clasificador de género, para los peatones obtuvo un 71.61% de acierto usando validación cruzada de 5 pliegues. De los 13 atributos, solo se utilizan 6 para la clasificación en el árbol de decisión, dando lugar a 7 reglas, 3 para la clase masculina y 4 para la femenina. En el caso de los conductores, el mejor clasificador obtuvo un 73.39% de acierto, pero lo logra a costa de la clase femenino. En ambos actores viales, los atributos que proporcionan más información para la separación de las clases son cruzar o conducir bajo los efectos del alcohol y conducir cansado.

Palabras clave: Clasificación, árboles de decisión, minería de datos.

Gender Classification of Pedestrians and Drivers based on their Behavior on Public Road

Abstract. Car accidents represent a public health problem worldwide, road traffic injuries are the leading cause of death in children and young adults. In Mexico, through the statistics of accidents prepared by the National Institute of Statistics and Geography, it is possible to have an overview of road accidents at the national level, as well as the human and material consequences that they entail. However, the factors or pedestrians and drivers' behavior that contributed to the accident occurred are not detailed. In this article, the gender classification

based on the road actors' behavior, pedestrians and drivers, exhibit on public roads is presented. For this, the J48 algorithm is used, which provides decision trees that allow examining the acquired knowledge. The best gender classifier, for pedestrians obtained a 71.61% success rate using 5-fold cross validation. Of the 13 attributes, only 6 are used for classification in the decision tree, giving rise to 7 rules, 3 for the male class and 4 for the female one. In the case of drivers, the best classifier obtained a 73.39% success rate, but it does so at the expense of the female class. In both road actors, the attributes that provides more information for separating classes are crossing or driving under the influence of alcohol and driving tired.

Keywords: Classification, decision trees, data mining.

1. Introducción

Los accidentes viales representan un problema de salud pública a nivel mundial, por ello la Organización de las Naciones Unidas (ONU) aprobó el "Decenio de Acción para la Seguridad Vial 2011-2020" y lo asignó a la Organización Mundial de la Salud (OMS) para su implementación [1]. Dentro de sus objetivos se encuentra el de compilar el estimado de la cantidad de víctimas mortales de tránsito en todo el mundo, para posteriormente, implementar estrategias y acciones sobre seguridad vial, en el ámbito regional, nacional y mundial, que disminuyan dicha estimación en un 50%; es decir, que se busca reducir la cantidad de víctimas mortales y traumatismos para el año 2020, la cual se aproxima anualmente a 1.3 millones, y de 20 a 50 millones respectivamente.

De acuerdo con el Informe de Estado Global sobre Seguridad Vial [2], los traumatismos por accidentes de tránsito son la octava causa de muerte para todos los grupos de edades y la primera en niños y adultos jóvenes de 5 a 29 años. Estos decesos tienen repercusiones sociales y económicas, ya que el costo aproximado que tienen los accidentes viales para un país está entre el 1% y el 3% de su producto nacional bruto [1].

México tomó acción sumándose al grupo de países que desean lograr un cambio significativo, por ello los gobiernos federales, estatales y regionales están implementando cada vez más medidas para prevenir estos percances, mejorando la infraestructura vial, limitando la velocidad en lugares como escuelas, integrando la información existente sobre accidentes viales y regularizando los estándares de seguridad mínimos con los que un vehículo debe contar [3].

Por otro lado, en México, a través de la estadística de Accidentes de Tránsito Terrestre en Zonas Urbanas y Suburbanas (ATUS) elaborada por el Instituto Nacional de Estadística y Geografía (INEGI) [4], se logra tener un panorama acerca de los accidentes viales en el ámbito nacional, así como las consecuencias humanas y materiales que conlleva [5]. Sin embargo, no se tiene detalle de cuáles fueron los factores o comportamientos de los peatones o conductores, que contribuyeron a que el accidente ocurriera.

Se llama actor vial tanto al peatón que es la persona que va a pie por la vía pública, como al conductor, que en este caso es aquel que conduce un vehículo particular automotor dentro de la vía pública. El tener el comportamiento de un actor vial, permite modelarlo usando un sistema multiagente [6] para poder realizar posteriormente una

simulación en la que, cuando ocurra un accidente, se puedan examinar los comportamientos de los involucrados. De esta manera, será posible tener información puntual que permita crear estrategias o campañas enfocadas a reducir los comportamientos viales que producen un accidente y, en consecuencia, si las personas acatan las recomendaciones, reducir el número de accidentes.

En este artículo, se presenta la forma en que se recolectó la información para permitir obtener los comportamientos de peatones y conductores y la generación de un clasificador para distinguir el comportamiento por género. El resto del artículo se organiza como sigue: en la sección 2 se muestran los trabajos relacionados, mientras que la metodología utilizada en esta investigación se presenta en la sección 3, y en la 4 el desarrollo del clasificador. En la sección 5, se presentan los experimentos realizados y los resultados obtenidos. Finalmente, en la sección 6 se presentan las conclusiones y trabajo futuro.

2. Trabajos relacionados

Comprender las circunstancias en las que los actores viales, conductores y peatones, tienen más probabilidades de morir o sufrir lesiones graves en un accidente automovilístico, o bien, estar inmersos en una situación de riesgo, es de particular preocupación para que los gobiernos a distinto nivel implementen programas encaminados a la mejora de la seguridad vial. Existen trabajos que han utilizado algoritmos de clasificación, como es el caso de Romero [7], donde usan C4.5 para organizar y clasificar la información de accidentes de tránsito, con base en las causas que los generaban como el clima, la infraestructura e infracciones de tránsito entre otras.

En [8] hacen uso de árboles de decisión y regresión CART, para determinar qué zonas son de mayor accidentalidad y qué variables se presentan en esos accidentes. Zhang [9] utilizó los algoritmos ID3 y C4.5 para el análisis de colisiones de tráfico en la zona de Saskatchewan, Canadá, donde identifican los factores que contribuyen a las colisiones y su gravedad. Encontraron que la bebida y no respetar las reglas de tránsito son los factores que más contribuyen para los accidentes. Por último, en cuanto al género, para hombres el factor es la bebida, los errores al conducir y las reglas de tránsito y para mujeres fueron reglas de tránsito, ambiente y bebida. En Jain [10], usan los algoritmos J48, Naive Bayes y Bayes Net para analizar accidentes de tránsito y determinar qué áreas son más propensas a accidentes de tránsito en la India. Uno de los factores que más influyen es la velocidad con la se transita.

En [11] se analizan las causas de los accidentes mediante los algoritmos Naive Bayes, Random Forest, MLP y AdaBoost, añadiendo algunas características que pueden ser medidas en tiempo real, como el clima y condiciones de la vialidad, de este modo, generan una predicción sobre la posible fatalidad de los accidentes. Dogru & Subasi [12], realizan una propuesta basada en el intercambio de información entre vehículos (velocidades, ubicación, trayectorias, tipo de vialidad, etc.) en un entorno de IoT y usando Random Forest, realizan una predicción de accidentes en vías de alta velocidad.

En [13], se utilizó una red bayesiana, árboles de decisión y redes neuronales artificiales, para detectar factores con la mayor influencia en accidentes

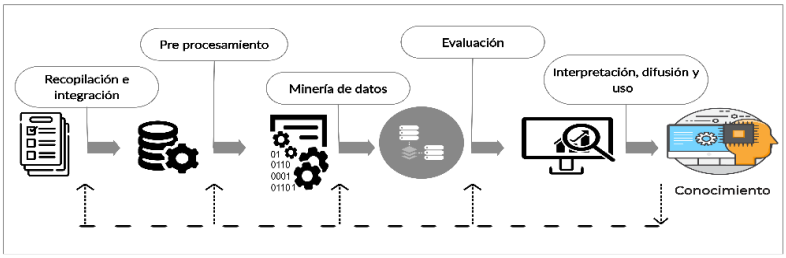


Fig. 1. Etapas del proceso de extracción del conocimiento

automovilísticos. Para ello, se llevó a cabo un experimento con datos de accidentes de tráfico en Reino Unido. De este experimento, se dedujo que los tres factores más frecuentes en un accidente vial son las condiciones de luz, la maniobra del vehículo y el tipo de carretera. Por su parte, en [14] para comprender los factores que conducen a la gravedad de los accidentes automovilísticos en la ciudad de Riad, Arabia Saudí, se utilizan tres técnicas de clasificación: CHAID, J48 y Naive Bayes. De este trabajo, se resaltan el peligro de distracción mientras se conduce y que los accidentes con autos más viejos tienen más probabilidades de provocar lesiones o muertes.

Si bien se sabe que algunos de los factores de riesgo, como el consumo de alcohol y drogas, afectan la gravedad, un modelo exacto de sus influencias sigue siendo un tema de investigación abierto. Sin embargo, no hay una precisión lo suficientemente confiable como para conocer todas las causas involucradas en un accidente vial, lo que sugiere que estas son complejas y se requiere más investigación. Por ende, hay varios factores de riesgo presentes esperando ser descubiertos o analizados.

3. Metodología

Para poder llevar a cabo esta investigación, se utilizó la metodología para el descubrimiento del conocimiento y minería de datos [15] propuesta en 1996 por Fayyad y que otros autores como Hernández han complementado o ampliado para adaptarla a sus necesidades particulares, aunque la esencia es la misma [16]. Una diferencia principal, es que no en todos los problemas se parte de una base de datos, por lo que se agrega la etapa de adquisición o recolección de datos, como es el caso de esta investigación. Considerando lo anterior, en la Fig. 1, se muestran las etapas de la metodología para la extracción de conocimiento, mismas que se explican a continuación.

3.1. Etapas del proceso de extracción del conocimiento

El proceso de extracción del conocimiento es tanto iterativo como interactivo; iterativo porque en cualquier etapa es posible regresar a una etapa anterior y repetir el proceso, mientras que la interacción se refiere a que un experto o usuario de interés ayude a la validación del conocimiento extraído [16]. Las etapas son las siguientes:

- **Recopilación e integración:** los datos necesarios para un análisis pueden estar almacenados en diferentes bases de datos por lo que la diversidad o dispersión

en una organización, puede ser un gran problema, además existen casos en los que los datos no han sido recolectados o son insuficientes para analizarlos y es necesario contar con un mecanismo para adquirirlos.

- **Preprocesamiento:** para poder utilizar los datos es necesario realizar una selección basada en la calidad de los datos que se tienen con el fin de utilizar los más relevantes para el problema. De igual forma, una vez que se han seleccionado los datos, es necesario eliminar el ruido que estos puedan tener o considerar situaciones como el manejo de valores faltantes o perdidos ya que puede conducir a resultados imprecisos. Otra tarea que también se debe realizar es la transformación de atributos, que consiste en realizar operaciones o funciones a los atributos originales.
- **Minería de datos:** el objetivo de esta fase es producir conocimiento, esto se hace construyendo un modelo para observar patrones y relaciones entre los datos que pueden usarse para hacer predicciones que ayuden a entender situaciones. Aquí se determina qué tipo de tarea se va a realizar, la representación final del conocimiento adquirido y en consecuencia el algoritmo a utilizar.
- **Evaluación:** el resultado de la etapa de minería de datos idealmente debe cumplir con al menos tres características: ser comprensibles, interesantes y precisos [16]. Las primeras dos características son evaluadas por el experto y son parte de la interpretación. Para determinar la precisión existen varias formas de evaluación, entre estas la validación simple y la validación cruzada.
- **Interpretación y uso:** etapa donde un analista o experto hace una recomendación con base en el modelo y sus resultados, o bien, aplica tal modelo a otro conjunto de datos.

4. Desarrollo del clasificador

A continuación, se presenta el desarrollo del clasificador de género basado en el comportamiento vial. Se muestran cada una de las etapas de la metodología aplicada al problema de clasificar a hombres y mujeres basándose en su comportamiento vial.

4.1. Recopilación e integración

Se examinó la información de la página del INEGI [4] respecto a los accidentes de tránsito en México, la cual consta de 46 atributos referentes al accidente, pero poco se habla del comportamiento de los involucrados. Sólo se maneja si el conductor se encontraba en estado de ebriedad y si usaba el cinturón de seguridad [17]. Por esta razón, para la adquisición de los datos, en esta investigación, se entrevistó a agentes de tránsito, y con base en sus respuestas y consideraciones, se diseñó una encuesta para los actores viales, que se aplicó a la población estudiantil del Centro Universitario UAEM Valle de México de la Universidad Autónoma del Estado de México (UAEM).

Las entrevistas se realizaron a 41 agentes de tránsito tanto estatal como municipal con el fin de que dijeran qué aspectos de comportamiento, tanto del peatón como del conductor, están presentes cuando un accidente ocurre. En la Tabla 1, se presentan algunos de los factores y el porcentaje en que estos agentes consideran que influyen en

Tabla 1. Factores que influyen en un accidente vial según los agentes de tránsito.

Factor	Porcentaje
Exceso de velocidad	46.34 %
Alcohol, medicamento y/o estupefacientes por parte del conductor	63.41 %
Alcohol, medicamento y/o estupefacientes por parte del peatón	29.26 %
Conducción distraída	65.85 %
Conducción con fatiga o sueño	65.85 %
Tratar de rebasar por la derecha por parte de los conductores	48.78 %
Cruzar por el medio de la calle o avenida por parte del peatón	58.53 %
Intención de ganarle al semáforo o vehículo por parte del peatón	63.41 %
Cruce distraído por parte del peatón	53.65 %

un accidente vial. Aproximadamente el 63% de los agentes, consideran que la intoxicación por parte del conductor, así como la conducción distraída o con sueño o fatiga influyen de manera importante en que ocurra un accidente.

Por otro lado, el que un peatón intente ganarle al semáforo, cruzar la calle o avenida por en medio de esta o cruce distraído es considerado como un factor de riesgo aproximadamente por el 60% de los agentes. En cuanto al alcohol, los agentes están de acuerdo que es un factor más relevante para conductores que para peatones en un accidente vial.

Con base en la información proporcionada por los agentes de tránsito, se diseñó una encuesta, que de acuerdo con Akerkar [18] es una técnica de adquisición del conocimiento relativamente fácil de realizar y recomendada para automatizar el procesamiento eficiente de los conocimientos recopilados y analizar los resultados.

La encuesta consta de 32 preguntas que se responden en una escala de Likert, que va del 1 al 5, donde 1 es nunca, y 5 es siempre, y que es apropiada para medir actitudes o comportamientos [19].

El público para recolectar los datos fueron estudiantes de entre 18 y 30 años. Considerando la matrícula en el ciclo escolar 2019B, que fue de 3811 alumnos, y considerando un intervalo de confianza del 95%, la cantidad de encuestados debió ser al menos de 350, esto de acuerdo con la ecuación de muestreo aleatorio simple (ver ecuación 1):

$$Tamaño\ de\ la\ muestra = \frac{\frac{Z^2 * p(1-p)}{e^2}}{1 + \frac{z^2 * p(1-p)}{e^2 N}}, \quad (1)$$

donde N= tamaño de la muestra, e = margen de error (porcentaje expresado con decimales) y z= puntuación z, número de desviaciones estándar por encima o por debajo de la media de población.

La encuesta se aplicó a un total de 587 estudiantes, 384 peatones (238 hombres y 146 mujeres) y 203 conductores (149 hombres y 54 mujeres). En la Tabla 2, se presentan las preguntas, así como la media y desviación estándar de las respuestas dadas.

Tabla 2. Resultados de la encuesta de comportamiento vial de peatones y conductores.

Comportamiento de peatones		**Comportamiento de conductores**	
Mirar a ambos lados $\bar{x} = 4.8, \sigma = 0.54$	Uso de teléfono $\bar{x} = 1.6, \sigma = 0.87$	Respeto por la luz roja $\bar{x} = 4.7, \sigma = 0.62$	Conducción con sueño o fatiga $\bar{x} = 4.6, \sigma = 1.14$
Calle de un solo sentido mira a ambos lados $\bar{x} = 4.3, \sigma = 0.99$	Uso de audífonos $\bar{x} = 2.4, \sigma = 1.26$	Respeto por el límite de velocidad $\bar{x} = 4.2, \sigma = 0.92$	Conducción bajo los efectos del alcohol $\bar{x} = 2.3, \sigma = 0.99$
Respeto semáforo peatonal $\bar{x} = 4.5, \sigma = 0.75$	Cruzar bajo efectos del alcohol $\bar{x} = 1.9, \sigma = 1.06$	Respeto por el paso peatonal $\bar{x} = 4.7, \sigma = 0.69$	Conducción bajo efectos de una droga $\bar{x} = 1.7, \sigma = 0.88$
Semáforo vehicular en verde $\bar{x} = 2.3, \sigma = 1.12$	Cruzar corriendo $\bar{x} = 3.1, \sigma = 0.91$	Uso del teléfono $\bar{x} = 1.8, \sigma = 0.95$	Conducción distraída $\bar{x} = 1.4, \sigma = 1.05$
Uso paso cebra $\bar{x} = 3.9, \sigma = 1.03$	Caminar por la banqueta $\bar{x} = 4.2, \sigma = 0.83$	Uso de los manos libres $\bar{x} = 2.6, \sigma = 1.47$	Uso del cinturón de seguridad $\bar{x} = 2.0, \sigma = 0.78$
Cruzar por esquinas $\bar{x} = 3.8, \sigma = 0.88$	Caminar distraídos $\bar{x} = 2.7, \sigma = 1.10$	Respeto por la señalización $\bar{x} = 4.5, \sigma = 0.77$	-
Uso puente peatonal $\bar{x} = 4.4, \sigma = 0.84$	-	-	-

4.2. Preprocesamiento de datos

La encuesta aplicada constó de un total de 32 campos o atributos, ya que además de las preguntas mostradas en la Tabla 2, también se le preguntó al actor vial, si se ha visto involucrado en un accidente vial y se le pidió que describiera algunos aspectos de este. Para esta etapa se seleccionaron únicamente 24 preguntas, 13 de peatones y 11 de conductores, y se separaron en dos archivos. Fueron pocas las encuestas que presentaron valores faltantes, sólo 3 y lo que se realizó para corregir, fue utilizar el valor entero más cercano a la media aritmética de la pregunta en cuestión.

4.3. Minería de datos

La minería de datos consiste en la extracción de información sensible, previamente desconocida, que reside de manera implícita en los datos. La minería de datos prepara y explora los datos para obtener información oculta en ellos [8]. Para esta etapa se eligió el algoritmo J48 que es la implementación del algoritmo C4.5, creado por Ross Quinlan como una extensión de su antecesor el ID3, y uno de los más utilizados en la minería de datos [20]. Este algoritmo utiliza el enfoque de divide y vencerás para generar el clasificador, calcula la ganancia de cada atributo y elige el de mayor ganancia para que sea la raíz del árbol, los datos son divididos con base en el dominio del atributo, de manera iterativa, se repite el proceso con cada partición [16].

El factor de confianza, en la construcción del árbol de decisión, es el parámetro que influye en el tamaño y capacidad de predicción del árbol construido, ya que está asociado a las operaciones de poda del árbol. Para cada operación, se compara la cantidad de error que sufriría el árbol de decisión antes y después. A menor probabilidad, se exigirá que la diferencia en los errores de predicción sea más

significativa para no podar. El valor por defecto es 25%. Según disminuya este valor, se aprueba una poda mayor, el tamaño del árbol disminuye y en consecuencia el error puede crecer [8], [21]. Los experimentos realizados y resultados en esta etapa se presentan en la sección 5 debido a su extensión e importancia.

4.4. Evaluación

El método que se usó para la evaluación fue la validación cruzada con n pliegues, para este caso n se manejó con los valores 10 y 5 pliegues para todos los experimentos realizados. La validación cruzada consiste en dividir aleatoriamente los datos en n grupos. Un grupo es para el conjunto de prueba y con el otro conjunto n-1 restante se usa para construir un modelo y se utiliza para pronosticar el resultado de los datos del grupo reservado. Por ejemplo, usar 5 pliegues equivale a utilizar el 80% de los datos para construir el clasificador y 20% para probarlo. El proceso se repite n veces, utilizando un conjunto de datos diferente para cada prueba. El error se calcula como la media aritmética de los errores en cada repetición [16, 22].

4.5. Interpretación y uso

Para esta etapa se contempla utilizar el modelo que entrega el algoritmo J48 para elegir el comportamiento que presentarán los agentes dentro de una simulación con el fin de poder registrar las conductas que estos exhibirían al ocurrir un accidente vial, y posteriormente, analizar los datos para determinar las conductas que más influyen en los diferentes niveles de riesgo de sufrir un accidente vial.

5. Experimentos y resultados

Para esta etapa se realizaron experimentos con el algoritmo J48 en el software WEKA en su versión 3.8.3 [23], con dos objetivos principalmente: el primero determinar si se podía generar un clasificador con un resultado aceptable que determine el género del actor vial basado en su comportamiento y el segundo determinar de qué forma son diferentes, en su comportamiento vial, los masculinos de los femeninos, y esto se logra a través de examinar los árboles construidos. Se presentan los resultados y el análisis de los experimentos realizados con los datos de peatones y conductores.

5.1. Peatones

En el caso de los peatones, se tiene un total de 384 encuestas, 238 hombres y 146 mujeres. Debido a que las clases no están balanceadas, un resultado adecuado con el cual poder discernir si el clasificador es aceptable, es que su porcentaje de acierto sea de un 62% o más, ya que ese es el resultado obtenido por omisión cuando se considera la clase mayoritaria y todas las instancias de género femenino tendrían una clasificación incorrecta dando un error del 38%. Se realizaron varias ejecuciones del algoritmo J48, en el que se modifican tanto el factor de confidencia como el número de pliegues utilizados para la validación cruzada. Los resultados se presentan en la Tabla 3. Se

Tabla 3. Experimentos con el algoritmo J48 para peatones.

N.º	Pliegues	Confidencia	Acierto	Error
1	10	0.25	66.92% (257)	33.07% (127)
2	10	0.20	68.22% (262)	31.77% (122)
3	10	0.15	68.75% (264)	31.25% (120)
4	10	0.10	68.75% (264)	31.25% (120)
5	10	0.05	69.79% (268)	30.20% (116)
6	5	0.25	68.48% (263)	31.51% (121)
7	5	0.20	70.83% (272)	29.16% (112)
8	5	0.15	71.35% (274)	28.64% (110)
9	5	0.10	71.61% (275)	28.38% (109)
10	5	0.05	70.31% (270)	29.68% (114)

Tabla 4. Experimento 9 con un 71.61% de acierto.

Masculino	Femenino	% de acierto por clase
202	36	84.87%
73	73	50%

puede observar que los mejores resultados se obtienen al usar la validación cruzada de 5 pliegues, y que el mejor, basado en porcentaje de acierto, es el experimento 9 seguido del experimento 8. Para poder determinar si el experimento sirve para el segundo objetivo es necesario ver más a detalle los resultados de dichos experimentos. Para ello, se muestra en la tabla 4 la matriz de confusión, y en la Fig. 2 el árbol de decisión, el cual tiene 13 nodos, de los cuales 7 son hojas.

La matriz de confusión, mostrada en la Tabla 4, indica que, de las 238 instancias masculinas, 202 se clasifican correctamente y 36 de forma incorrecta dando un porcentaje de acierto para la clase masculina del 84.87%. Por otro lado, de las 146 mujeres, solo se logra clasificar de manera correcta a 73 dando un acierto por clase del 50%.

Analizando el árbol de la Fig. 2, se muestra que el nodo raíz "caminar bajo los efectos del alcohol", divide a las 384 instancias en 2 conjuntos, 199 que nunca cruzan bajo los efectos del alcohol (rama izquierda) y 185 que cruzan bajo los efectos del alcohol, de estos últimos, 155 cruzan casi nunca o a veces bajo los efectos del alcohol, y los 30 restantes casi siempre o siempre, siendo 27 de ellos masculinos.

En las hojas del árbol, se observa la clase y la cantidad de instancias que llegaron a la misma, seguido del número de instancias mal clasificadas. Así, en la rama derecha la conclusión es que el género es masculino, cayendo en esta rama 185 instancias, de entre las cuales hay 41 sujetos femeninos mal clasificados.

De los 13 atributos que se utilizan en la clasificación, sólo 6 aparecen en el árbol final, el cuál al ser recorrido proporciona las siguientes reglas:

Reglas de comportamiento para peatones masculinos

1. Si a veces cruza bajo los efectos del alcohol entonces es masculino

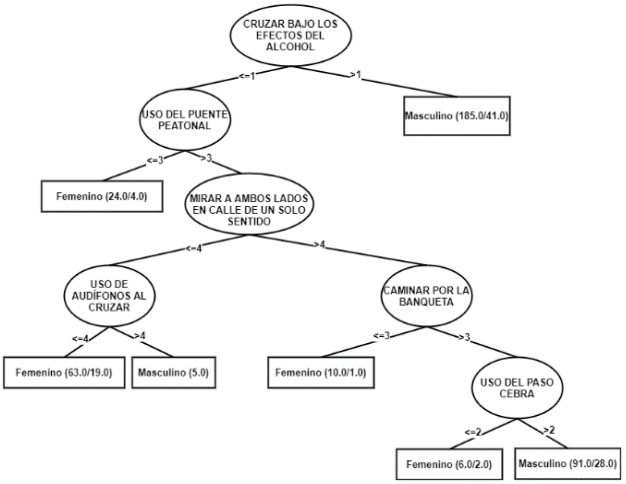


Fig. 2. Árbol de decisión del comportamiento vial de peatones.

2. Si nunca cruza bajo los efectos del alcohol, casi siempre o siempre usa un puente peatonal, a veces o casi siempre al cruzar mira a ambos lados de la calle, aun cuando es de un solo sentido, y siempre usa audífonos entonces es masculino.
3. Si nunca cruza bajo los efectos del alcohol, casi siempre o siempre usa un puente peatonal, siempre al cruzar mira a ambos lados de la calle, aun cuando es de un solo sentido, casi siempre o siempre usa la banqueta y a veces, casi siempre o siempre usa el paso cebra entonces es masculino.

Reglas de comportamiento para peatones femeninos

1. Si nunca cruza bajo los efectos del alcohol y a veces, o casi nunca o nunca usa un puente peatonal entonces es femenino.
2. Si nunca cruza bajo los efectos del alcohol, casi siempre o siempre usa un puente peatonal, a veces o casi siempre al cruzar mira a ambos lados de la calle, aun cuando es de un solo sentido, y a veces usa audífonos entonces es femenino.
3. Si nunca cruza bajo los efectos del alcohol, casi siempre o siempre usa un puente peatonal, siempre al cruzar mira a ambos lados de la calle, aun cuando es de un solo sentido, y a veces usa la banqueta entonces es femenino.
4. Si nunca cruza bajo los efectos del alcohol, casi siempre o siempre usa un puente peatonal, siempre al cruzar mira a ambos lados de la calle, aun cuando es de un solo sentido, siempre o casi siempre usa la banqueta y nunca o casi nunca usa el paso cebra entonces es femenino.

5.2. Conductores

En el caso de los conductores, se encuestó a un total de 203 personas, 149 hombres y 54 mujeres. Las clases no están balanceadas, por lo que, en este caso un resultado adecuado con el cual poder discernir si el clasificador es aceptable, es que su porcentaje de acierto sea igual o esté por arriba del 73.39%, ya que ese es el resultado obtenido

Tabla 5. Experimentos con el algoritmo J48 para conductores.

N.°	Pliegues	Confidencia	Acierto	Error
1	10	0.25	70.44% (143)	29.55% (60)
2	10	0.20	71.42% (145)	28.57% (58)
3	10	0.15	72.90% (148)	27.09% (55)
4	10	0.10	73.39% (149)	26.60% (54)
5	10	0.05	73.39% (149)	26.60% (54)
6	5	0.25	70.44% (143)	29.55% (60)
7	5	0.20	72.41% (147)	27.58% (56)
8	5	0.15	73.39% (149)	26.60% (54)
9	5	0.10	73.39% (149)	26.60% (54)
10	5	0.05	73.39% (149)	26.60% (54)

Tabla 6. Matrices de confusión en los experimentos para conductores

N.°	Matrices de confusión		% acierto por clase	% acierto
	Masculino	Femenino		
1	131	18	87.9	70.44
	42	12	22.2	
2	142	7	95.3	71.42
	51	3	5.5	
3	148	1	99.3	72.9
	54	0	0	
6	136	13	91.2	70.44
	47	7	12.9	
7	147	2	98.6	72.41
	54	0	0	

por omisión cuando se considera la clase mayoritaria, es decir, en el que se clasifica para masculino todas las instancias y las instancias de género femenino tendrían una clasificación incorrecta dando un error del 26.61%. En la Tabla 5, se presentan los resultados de las ejecuciones del algoritmo J48. Se observa que los experimentos con mejor porcentaje de acierto son el 4, 5, 8, 9 y 10, ya que tuvieron el número mayor de instancias correctamente clasificadas, pero se observa que absorben todas las instancias de género femenino como error, por lo que, si bien cumplen con el primer objetivo, no cumplen con el segundo objetivo de diferenciar entre los dos géneros.

A fin, de poder rescatar las diferencias de comportamiento por género entre los conductores, se procede a examinar las matrices de confusión de los experimentos 1, 2, 3 6, y 7, mismas que se muestran en la Tabla 6. Las columnas 2 y 3 presentan la matriz de confusión de cada experimento, el primer renglón corresponde a la clase masculino mientras que a la clase femenino corresponde el segundo renglón. Por ejemplo, en el experimento 1, de los 149 conductores masculinos, sólo 131 se clasificaron correctamente y 18 se clasificaron erróneamente como femeninos. Por otro lado, de las 54 conductoras femeninas, solo 12 se clasifican correctamente y 42 de forma incorrecta.

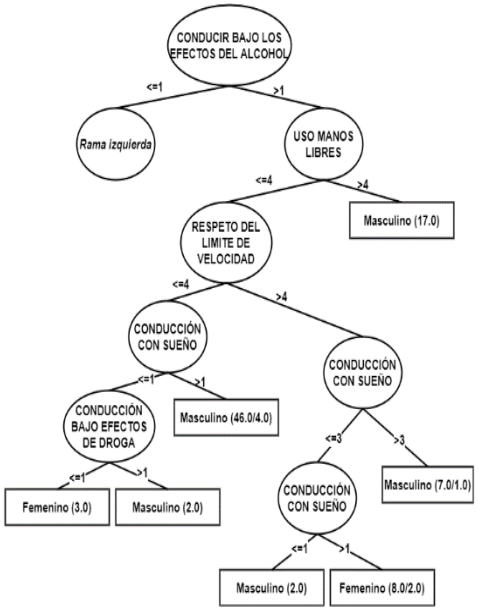


Fig. 3. Árbol de decisión del comportamiento vial de conductores.

El porcentaje de acierto de la clase masculino es de 88% mientras que de la clase femenina es de 29%. Se observa en los experimentos del 1 al 3, que fueron generados con validación cruzada de 10 pliegues, que conforme va aumentando el nivel de acierto del clasificador, se va disminuyendo el porcentaje de acierto de la clase femenino.

El árbol del experimento 1, provee más información para diferenciar las clases, tiene 20 hojas, 7 de la clase femenino y 13 de la clase masculino. En la Fig. 3, se observa que la raíz del árbol nuevamente está relacionada con el consumo de alcohol. Dividiéndose las 203 instancias en 118 que nunca conducen bajo los efectos del alcohol (ver Fig. 4) y 85, la rama derecha, que si lo hacen. También de las 85 instancias que si conduce bajo los efectos del alcohol sólo 9 son de género femenino, y de las 76 de género masculino, 42 manejan cansados. En la rama izquierda dada en la Fig. 4, se encuentran los 118 que nunca conducen bajo los efectos del alcohol. Se puede observar que hay mayor profundidad en el árbol por tratar de definir las hojas que pertenecen al género femenino. En los dos primeros niveles del subárbol hay dos hojas con 59 instancias, de las 118 que inmediatamente clasifican a masculino, teniendo 11 instancias mal clasificadas.

6. Conclusiones y trabajo futuro

El objetivo de este trabajo fue clasificar a los actores viales por su género, basándose en su comportamiento vial. Los actores que se incluyeron fueron el peatón y el conductor de un vehículo particular. Con los experimentos conducidos utilizando el algoritmo C4.5 (J48) en el software WEKA, se encuentra que el mejor clasificador de

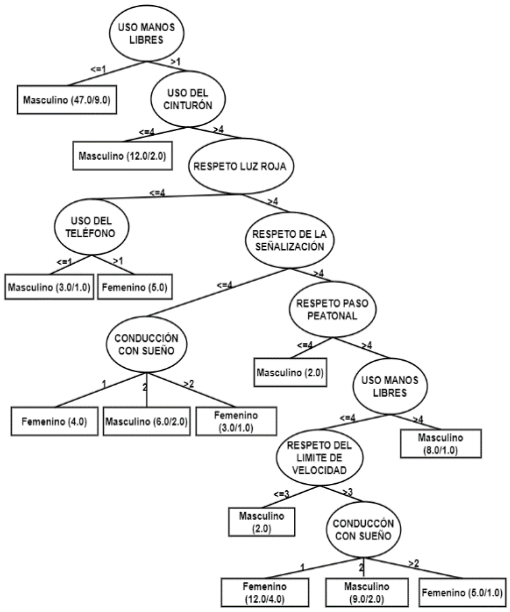


Fig. 4. Rama izquierda del árbol de decisión para conductores.

género, para los peatones obtuvo un 71.61% de acierto usando validación cruzada de 5 pliegues. De los 13 atributos, sólo se usan 6 para el árbol de decisión, dando lugar a, 3 reglas para la clase masculino y 4 para la femenina. En el caso de los conductores, el mejor clasificador que tiene el 73.39% de acierto, lo logra a costa de la clase femenino. Esto es porque la clase masculina es mayoritaria, sólo el 26.61% son mujeres. Al parecer, los datos que se tienen son insuficientes para diferenciar a los conductores en hombres y mujeres basados en su comportamiento.

Vale la pena mencionar que, en ambos actores viales, el atributo que proporciona más información para la separación de las clases es la de cruza o conduce bajo los efectos del alcohol, de los 185 peatones que lo hacen, 144 son hombres, mientras que, en los conductores de 85 instancias, 76 son hombres. Otro atributo que aparece varias veces en el árbol de los conductores es el de la conducción con sueño. Este atributo junto con el de conducción bajo los efectos del alcohol, fueron evaluados por los agentes de tránsito como los que más influyen en un accidente vial.

En este artículo, se considera que los resultados son alentadores en el caso de los peatones, y que para los conductores es necesario un mayor número de encuestados y repetir el experimento para discernir si la razón por la que no fue posible obtener un buen clasificador es porque ambos géneros, masculino y femenino, se comportan igual, o bien, esto es provocado por el desbalanceo de las clases obtenidas en los experimentos.

Como trabajo futuro se considera realizar un análisis comparativo empleando otros métodos basados en árboles, como Random Forest, así como algoritmos basados en reglas como PART, para mejorar la precisión de la clasificación.

Además, se pueden generar reglas de asociación para descubrir otras regularidades en los conjuntos por género.

Referencias

1. Sminkey, L.: Plan mundial para el decenio de acción para la seguridad Vial 2011-2020. World Health Organization (2011)
2. World Organization Health: Global status report on road safety 2018: Summary (2018)
3. OPS México: Organización panamericana de la salud (2018)
4. INEGI: Instituto Nacional de Estadística y Geografía (2016)
5. Secretaría de Salud: Informe sobre la situación de la seguridad vial (2017)
6. Osorio, L.A.: Modelo de simulación vial basado en agentes de software (2013)
7. Romero, A.G.: Minería de datos en el análisis de accidentes de tránsito en el Ecuador (2019)
8. Calderón, D.H, Sora, D.F.: Análisis de accidentalidad vehicular usando técnicas de minería de datos (2019)
9. Zhang, X.F., Fan, L.: A decision tree approach for traffic accident analysis of Saskatchewan highways. In: Canadian Conference on Electrical and Computer Engineering (2013)
10. Jain, A., Ahuja, G., Mehrotra, A.: Data mining approach to analyse the road accident in India. IEEE, pp. 175–179 (2016)
11. Xia, X.L., Nan, B., Xu, C.: Real-time traffic accident severity prediction using data mining technologies. In: Proceedings of the International Conference on Network and Information Systems for Computers, ICNISC, pp. 242–245, Institute of Electrical and Electronics Engineers Inc. (2017)
12. Dogru, N., Subasi, A.: Traffic accident detection using random forest classifier. In: 15th Learning and Technology Conference, pp. 40–45, Institute of Electrical and Electronics Engineers Inc. (2018)
13. Castro, Y., Kim, Y.J.: Data mining on road safety: factor assessment on vehicle accidents using classification models. International Journal of Crashworthiness, 21, pp. 104–111 (2016)
14. Al-Turaiki, I., Aloumi, M., Aloumi, N., Alghamdi, K.: Modeling traffic accidents in Saudi Arabia using classification techniques. In: 4th Saudi International Conference on Information Technology (Big Data Analysis), KACSTIT, Institute of Electrical and Electronics Engineers Inc. (2016)
15. Fayyad, U., Piatetsky-Shapiro, G., Smith, P.: From data mining to knowledge discovery and data mining. AI Magazine, 17(1), pp. 37–54 (1996)
16. Hernández, J., Ramírez, M.J., Ferri, C.: Introducción a la minería de datos (2004)
17. INEGI: https://www.inegi.org.mx/programas/accidentes/default.html#Documentacion (2018)
18. Akerkar, R., Srinivas, P.: Knowledge-based system (2010)
19. Vallejo, P.: Medición de actitudes en psicología y educación: construcción de escalas y problemas metodológicos (2006)
20. Quinlan, J., Kumar, V., Wu, X.: The top 10 algorithms in data mining. Knowledge and Information Systems 14(1), pp. 1–37 (2008)
21. Drazin, S..: Decision tree analysis using Weka. Machine Learning-Project II (2012)
22. Hoyos, G., Sleyther, Y.: Evaluación del riesgo crediticio en entidades bancarias en el área de microfinanzas utilizando árboles de decisión. pp. 1–67 (2017)
23. Weka: https://www.cs.waikato.ac.nz/ml/weka/ (2018)

ISSN 1870-4069

Procesamiento de Lenguaje Natural: una solución para detectar noticias falsas sobre la 4T en México

M. Beatriz Bernábe Loranca, Mayra Vanessa Montiel Estrada, Rogelio González Velázquez, Dylan Salomón Lozada Mendoza

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{beatriz.bernabe, mayra.montielestrada, rogelio.gzzvzz, ing.dylan21}@gmail.com

Resumen. Para diversos propósitos, las redes sociales se han convertido en un escaparate de importancia relativa. Uno de los éxitos de las redes es la facilidad de comunicación y el impacto en distintas actividades que obedecen a los intereses individuales y grupales. Estos intereses pueden identificarse como el deseo de compartir noticias, eventos, sucesos o bien, pronunciarse en la red a través de opiniones descuidando muchas veces la veracidad del contenido. En particular, Twitter es un medio que permite escribir y compartir información sin filtros, por tanto, es casi inmediata la publicación del tweet. El usuario en su entusiasmo por escribir en Twitter, de manera incluso irreflexiva, se expresa rápidamente, con descuidos frecuentes sobre la fuente de la información que comparte y en consecuencia, el usuario asume que la información que publica es verdadera, otorgándole al tweet una ingenua credibilidad. Es en este punto que se describe el presente trabajo, bajo la creación de diccionarios y análisis de sentimientos, en combinación con Naive Bayes y Arboles de Decisión, se ha extraído información de Twitter por un periodo de 3 meses con el fin de contabilizar el porcentaje de tweets falsos y reales alrededor de la 4T (4 Transformación) en México.

Palabras clave: Redes sociales, twitter, PLN, noticias falsas.

Natural Language Processing: A solution to Detect Fake News about 4T in Mexico

Abstract. For various purposes, social media has become a showcase of relative importance. One of the successes of the networks is the ease of communication and the impact on different activities that respond to individual and group interests. These interests can be identified as the desire to share news, events, and happenings or, to pronounce on the network through opinions, often neglecting the veracity of the content. Twitter is a medium that allows you to write and share information without filters; therefore, the publication of the tweet is almost immediate. The user in his enthusiasm for writing on Twitter, even thoughtlessly, expresses himself quickly, with frequent oversights about the source of the

information he shares and consequently, the user assumes that the information he publishes is true, giving the tweet a naive credibility. It is at this point that the present work is described, under the creation of dictionaries and sentiment analysis, in combination with Naive Bayes and Decision Trees, information has been extracted from Twitter for a period of 3 months in order to account for the percentage of false and real tweets around the 4T (4 Transformation) in Mexico.

Keywords: Social networks, twitter, NLP, fake news.

1. Introducción

La mayor parte de trabajos para Procesamiento de Lenguaje Natural (PLN) se han desarrollado en inglés dado que las librerías de los lenguajes de programación que procesan textos leen la información en ese idioma. Considerando que el objetivo de este trabajo es analizar las noticias verdaderas/falsas de la 4T en Twitter, es razonable que la mayor parte de opiniones y noticias sean escritas en idioma español. A pesar de tal restricción, el uso de Python en este trabajo se justifica por las bondades de sus librerías en el procesamiento de lenguaje natural.

Considerando que Twitter es una red social con alta difusión y versátil para concentrar discusiones en el ambiente político, económico, etcétera, esta red representa un atractivo repositorio de datos para extraer y analizar información. Por otro lado, Phyton es un lenguaje accesible que ofrece muchas ventajas de implementación, entre ellas la utilidad para descargar información de Twitter y recursos de análisis de sentimientos entre otros.

A partir del uso dinámico de distintos tipos de usuarios sumado a la multiplicidad de noticias compartidas, las noticias falsas se hacen presentes de manera constante. Estas noticias tienen objetivos diversos, sin embargo, se asume que uno de los propósitos principales es desinformar y manipular la opinión del público. Cada publicación en Twitter se caracteriza por tener el contenido asociado y el usuario activo responsable del Tweet.

Debido a que la difusión de información falsa impacta en la economía, educación, en decisiones de carácter ambiental e incluso personales, en este artículo se ha puesto énfasis en separar noticias falsas y reales usando las características de los tweets extraídos en el proceso de descargas, la cual es la primera fase de este artículo. La ventaja de esta fase consiste en que los tweets al heredar información de los usuarios, ha sido posible organizarlos y etiquetarlos para uso posterior. En este punto, una pregunta se ha formulado ¿Es posible clasificar los tweets como noticias reales o falsas basándonos en lo que publica la gente? La respuesta no es inmediata ni fácil, no obstante, hemos centrado nuestra atención en aprovechar la clasificación jerárquica del lenguaje R para dividir los tweets.

El tema bajo estudio es “La cuarta transformación de México 4T”, la cual refiere a la visión que el presidente Andrés Manuel López Obrador tiene para el futuro de México.

Para efectos de organización, el presente trabajo se estructura como sigue: Esta introducción como sección 1. La sección 2 aborda conceptos generales de los tópicos sobre este trabajo y un breve estado del arte.

En la sección 3 se describe el proceso de clasificación y en la sección 4 se discuten los resultados. Finalmente se presentan las conclusiones.

2. Generalidades y estado del arte

En esta sección se resumen los conceptos básicos y un breve estado del arte.

2.1 Procesamiento de lenguaje natural

El Procesamiento de Lenguaje Natural (PLN) es un área de investigación que explora el manejo de las computadoras para comprender y manipular textos orales o escritos del lenguaje natural con el propósito de hacer cosas útiles. PLN se entiende pragmáticamente como el estudio y análisis de los aspectos lingüísticos de un texto a través de programas informáticos. Entre las aplicaciones que utilizan PLN, destacan los correctores ortográficos y clasificación de documentos entre otros [1]. En términos amplios, los fundamentos del PLN pueden verse distintas disciplinas como la informática, ciencias de la información, lingüística, matemáticas, ingeniería eléctrica y electrónica, inteligencia artificial, robótica, psicología, etc.

Dentro de las aplicaciones del PLN, el procesamiento de textos es el más popular, pero ello no excluye otros usos de importancia mayúscula como la traducción automática, interfaces de usuario, recuperación de información en varios idiomas, reconocimiento de voz, sistemas expertos, etcétera [2].

2.2 Noticias falsas y su impacto

Las noticias turbias son una preocupación constante por la gran variabilidad de implicaciones. Las Noticias Falsas (NF) han sido preocupación de investigadores para el su control. Internet y nuevas tecnologías de comunicación e información, han contribuido a la extrapolación de NF considerando que distintos medios como las redes sociales son de fácil acceso para cualquier usuario.

Para un público considerable, se entiende como noticia falsa aquella que es divulgada de tal modo que se considere un peligro de desinformación para distintos grupos [3]. Las NF tienen 70% más de probabilidades de ser replicadas y se identifican por títulos atractivos que buscan capturar a los usuarios y consecuentemente son distribuidas de manera irreflexiva (la mayoría de las personas solo ve el titulo seductor y no se preocupa por la fuente ni lee la noticia completa, generalmente con gran cantidad palabras emotivas) [4].

Para poder detectar NF, los siguientes puntos son importantes: a) fabricaciones serias (descubiertas en los medios de comunicación principales o participantes, prensa amarilla o tabloides); b) engaños a gran escala; c) falsificaciones humorísticas (sátira de noticias, parodia, programas de juegos) [5].

2.3 Estado del arte

El estado del arte en esta área es extenso, de tal manera que reunir trabajos relacionados merece un artículo aparte. En este escenario, después de revisar distintos documentos, se eligieron los que a continuación se resumen.

1) **FAGON:** Modelo de detección de noticias falsas que utiliza transformación gramatical en una red neuronal**.** FAGON describe un modelo de detección de noticias falsas que determina eficientemente si la proposición es verdadera o no, lo cual es posible cuando del artículo dado aprende al realizar una transformación gramatical a través de la red neuronal. Especialmente el modelo se enfoca en el idioma coreano y se compone de dos módulos: oración generador y clasificación. Este trabajo a diferencia de otros utiliza la incorporación de texto rápido para reflejar las post-posiciones en la gramática del idioma coreano para proceder a dividir las palabras en morfemas. Los autores disponen de una capa CNN (modelo de aprendizaje profundo/red neuronal convolucional), independiente para cada oración para mejorar la precisión del modelo extrayendo más características de palabras [6].

2) **Modelado colaborativo en lenguaje natural a través de redes sociales.** Los autores describen el diseño de un "chat-bot" que funciona sobre redes sociales para realizar tareas de modelado colaborativo. Esta herramienta proporciona soporte para modelado interpretando diversos requisitos expresados en Lenguaje Natural (LN), acercando de esta manera el modelado a participantes con poca experiencia en esa área. Al funcionar sobre redes sociales, el "chat-bot" es accesible desde múltiples dispositivos y facilita la colaboración en movilidad y en tiempo real. En la tarea de modelado se pueden involucrar un gran número de personas usando LN y no es necesario que los usuarios tengan mucha experiencia en las tareas de modelado [7].

3. Extracción de información y proceso de clasificación usando Python y R

Para alcanzar una aproximación de clasificación sobre noticias verdaderas o falsas de la información de Tweets, se eligió agrupamiento jerárquico del lenguaje R. A pesar de que R ofrece k-medias como alternativa de clasificación que es fácil de comprender, este tipo de clasificación solo proporciona óptimos locales de los criterios que optimiza y su uso se encuentra fuera de los propósitos de este trabajo dado que pertenece a la clasificación por particiones. En este punto, la clasificación jerárquica se adapta correctamente para el tipo de datos del problema de este artículo.

Por otra parte, se definió como corpus los tweets del usuario en una ventana de tiempo de dos meses utilizando representaciones de TF-IDT (frecuencia de término frecuencia inversa de documento). Posteriormente ha sido posible identificar tendencias y características, además de un análisis descriptivo para establecer índices de impacto [8]. Para efectos de orden, primero ha sido necesario descubrir información del tema a tratar, para ello y previo a la clasificación, se consiguieron los tweets relacionados a frases sobre la 4T.

3.1. Obtención del conjunto de datos: descargas de Twitter

Se realizaron descargas para construir el conjunto de datos sobre tweets relacionados con la 4T, Tren Maya, AMLO, La Mañanera, y Reforma, los cuales son medios y expresiones populares sobre AMLO y la 4T.

Tabla 1. Lista de variables existentes en los tweets.

No.	Variable	No.	Variable
1	Id	17	Texto de Biografía
2	Texto Tweet	18	Ubicación
3	No. Retweets	19	URL
4	No. Favoritos	20	Verificado
5	Largo del Tweet	21	No. Respuestas
6	Valor del Sentimiento	22	Porcentaje Tweets con Usuario
7	Sentimiento	23	Porcentaje Tweets con Hashtag
8	Id Usuario	24	Porcentaje De Retweets
9	Nombre	25	Porcentaje Tweets con links
10	Usuario	26	Porcentaje Tweets con Media
11	Fecha Creación cuenta	27	Tweets/Followers
12	Followers	28	Following/Followers
13	Following	29	Tiene URL usuario
14	No. Favoritos del Usuario	30	Que Usa el Usuario
15	No. Tweets	31	Real o Fake
16	No. Listas		

Distintas y múltiples son las críticas recibidas para el gobierno en turno. Se presume que las redes sociales han sido el mayor espacio de expresión para elogiar o censurar a la 4T. En este sentido, es evidente que muchas noticias son falsas, tanto para desacreditar como para aplaudir al presidente AMLO y su equipo. Esto ha dado lugar a proponer estrategias para evaluar el porcentaje de noticias falsas. Si la mayoría de las opiniones negativas fueran falsas, entonces la reputación del presidente ha sido ensuciada injustamente, caso contrario, el gobierno de AMLO debe enderezar su praxis política.

Por tanto, se eligió a Twitter como caso de estudio. El primer paso fue extraer información descargando tweets desde el 15 de marzo hasta el 9 de mayo de 2020 usando *tweepy*, una biblioteca de Python para acceder fácilmente a la API de Twitter, con la cual se alcanzó un total de 23,780 descargas. Las descargas se depuraron con un proceso de limpieza que consistió de eliminar re-tweets y borrando información excluyente del problema, consiguiendo así un total de 221 tweets. De esta información reducida, se obtuvieron 31 características (variables), que pueden verse en la tabla 1, las cuales representan el resultado de las noticias que han sido identificadas y definidas por los autores de acuerdo a la metodología de la sección 3.2.

Para propósitos de una clasificación consistente de noticias falsas o verdaderas, se eligió un subconjunto de 12 características extraídas de la tabla 1, las cuales pueden verse en la tabla 2. Estas características se interpretan como sigue:

1. Los tweets con mayor número de hashtag tienen más “valor” como opiniones de los usuarios que las noticias replicadas.

2.- El análisis del total de tweets por medio de sentimientos proporciona una bifurcación de tweets negativos y positivos.

3.- Es importante la fuente de los tweets y la popularidad cuando son re-tweets.

4.- Ocupar los hashtags en el contenido de los tweets son necesarios para la clasificación de dichos tweets.

Tabla 2. Variables y valores para la clasificación.

Tabla de Características	Valor de Referencia
No. Retweets	≥ 50
No. Favoritos	≥ 25
Largo del Tweet	≥ 100
Sentimiento	Negativo, Positivo
No. Favoritos Usuario	≥ 3000
No. Tweets Usuario	≥ 3000
Porcentaje de Tweets con hashtag	≥ 25%
Porcentaje De Retweets de Usuario	≥ 50%
Tweets con links	≥ 25%
Tweets/Followers	≥ 40
Following/Followers	≥ 1.5
Tiene URL usuario	True

5.- La cuenta del usuario requiere vigencia para comprobar el contenido.

6.- El porcentaje de tweets con hashtag se mantiene para conocer el impacto de los hashtags dentro del algoritmo y definir un peso de acuerdo con el sentimiento asignado.

En la siguiente tabla 2 se indican las variables con su valor.

3.2. Metodología

Para el análisis de los datos se determinaron características divididas en características de usuario y del tweet. Se utilizó la librería *NLKT* que proporciona más de 50 recursos corporales y léxicos y ofrece un conjunto de bibliotecas de procesamiento de texto para clasificación, *tokenización*, derivación, etiquetado, análisis y razonamiento semántico para bibliotecas PLN [9]. Por otro lado, también se utilizó *TexBlob*, una API simple para investigar tareas comunes de PLN como el etiquetado de parte del discurso, extracción de frases nominales, análisis de sentimientos, clasificación y traducción [10]. Estas librerías ayudaron a extraer las características que presentaron en las tablas anteriores (ver tablas 1 y 2).

Una vez que el texto se *tokenizó* junto con la limpieza de datos, el texto se analizó con el paquete *Text Blob* para determinar la polaridad negativa o positiva de los tweets. Este procedimiento facilitó la tarea del análisis de sentimientos dado que es más practico comparar los tweets del corpus que contiene, de otro modo, se debe homogeneizar cada palabra a un diccionario de palabras positivas y negativas. Por otra parte, se utilizó el diccionario de palabras para la enumeración de palabras y determinar que tweets utilizar en este problema.

Las otras características derivadas del tweet fueron la proporción de número de tweets/seguidores, proporción de número de following/seguidores y la fuente que usan los usuarios para publicar. Estos tweets se extrajeron en forma CSV con su información asociada que representa una característica del tweet y al final el predictor. Se divido el conjunto de datos en entrenamientos y prueba de datos, separando el conjunto de datos en la proporción 60:40 para aplicar los clasificadores de *machine learning* del software R. En este caso se eligió la clasificación jerárquica [11].

3.3. Pseudocódigo

Módulo 1: Clasificación de los Tweets

Algoritmo: Clasificación de Noticias Reales-Falsas
Variables: mod,t.ids,Datos
Inicio:
Importar las librerías e1071, naivebayes, caret
Procedimiento:

1. Leer: Datos (Archivo.csv)
2. Generamos la partición de datos en el conjunto de datos
3. Se define la semilla en 215
4. Generar el conjunto de entrenamiento t.ids con créateDataPartition(ConjuntoDeDtaos$VariableAPredecir, Probabilidad,Lista)
5. Definir el modelo NaiveBayes a mod, tratando de predecir Real_Fake en función de Retweets + Favoritos + Largo_Twet + Favoritos_Usuario + No_Twets + Tweets_con_hashtag + Porcentaje_De_Retweets + Tweets_con_links + Tweets_Followers + Following_Followers + URL_usuario + Sentimiento, del Dataframe (Datos)
6. Cargados dataframe y las predicciones, se prueba en validación (pred <- predict(mod, Datos[-t.ids,])
7. Se genera la tabla con asignaciones Falso y Verdadero
8. Se genera la matriz de confusión pasando la tabla anterior

Módulo 2: Sentimientos

Algoritmo: Analizador de Sentimientos de Tweets
Variables: Tweets, Valor
Inicio:
Importar las librerias csv, TextBlob
Procedimiento:

1. Leer Archivo(“Datos.csv”)
2. Modificar el Nombre del archivo que se desea examinar para los sentimientos(“Sentimientos.csv”)
3. Seleccionar la columna a evaluar, considerando que comienza en 0
4. Asignar Tweet a Valor
5. Evaluar tweet con sentiment.polarity
6. Si Valor.sentiment.polarity >0
7. Escribir (Arreglo Tweets,“Positivo”, Valor.sentiment.polarity)
8. Si Valor.sentiment.polarity < 0
9. Escribir (Arreglo Tweets,“Negativo”, Valor.sentiment.polarity)
10. Si Valor.sentiment.polarity = 0
11. Escribir (Arreglo Tweets,“Neutral”, Valor.sentiment.polarity)
12. Guardar Archivo Datos en (“Sentimientos.csv”)

3.4. Discusión de resultados

Antes de entrenar los algoritmos, una etiqueta falsa fue asignada a noticias cargadas de sentimientos o emociones y a las noticias con redacción ausente de júbilo se le etiquetó como verdaderas, lo que puede verse en la columna ASIGNADO de la tabla 3.

En este escenario, el algoritmo de Arboles de Decisión (AD), al calificar noticias ASIGNADAS como falsas aseguraba que esas noticias son efectivamente falsas, que en esta combinación se contabilizaron 22 noticias falsas (falso, falso).

Tabla 3. Matriz de algoritmo – asignación.

	Asignada	**Arboles de Decisión**	**Naive Bayes**
Falsa	96	120	129
Verdadera	125	101	92

Tabla 4. Matriz de resultados por dos técnicas.

Algoritmo / **Asignadas**	**Arboles de Decisión**		**Naive Bayes**	
	Tweet Falso	Tweet Verdadero	Tweet Falso	Tweet Verdadero
Falsa	22	37	33	18
Verdadera	11	19	15	22

Tabla 5. Matriz de resultados (noticias reales).

Tweet	**Sentimiento**	**Noticia**	**Cantidad**
#Informe Construcción del #TrenMaya generara 80,000 empleos en 2020 y una derrama de 35,000 millones de pesos: #AMLO Dijo que el 30 de abril se firmara el contrato del primer tramo (Palenque-Escarcega) para iniciar los trabajos al siguiente día	**Positivo**	**REAL**	**24**
AMLO reitera continuidad del Complejo Cultural Chapultepec INNECESARIO	**Neutral**	**REAL**	**12**
APROBACION 6 de abril 2020 477% (-02) Sin registrar aun el posible efecto del mensaje de @lopezobrador_ ayer domingo, su aprobación sigue en una continua disminución con 15 puntos en lo que va del mes llegando a 477% @el economista	**Positivo**	**REAL**	**7**

Tabla 6. Matriz de resultados (noticias falsas).

Tweet	**Sentimiento**	**Noticia**	**Cantidad**
QUEDATEENCASA es una estrategia de manipulación para que el gobierno se ahorre millones en retrovirales ivermectina y cloroquina para esta pandemia dicen los de derecha, AMLO no quiso suspender millones de Tren Maya ni SantaLucia ni Refineria esa la realidad de #quedatetucasa	**Negativo**	**FALSA**	**40**
Hacen el ridiculo los que promueven noticias falsas y cada vez se desgastan más porque son exhibidos, tanto politicos como medios de comunicacion: AMLO	**Negativo**	**FALSA**	**15**
@JorgeGCastaneda @OEA_oficial Y como por que es derrota eso para la 4T ridiculo Acaso enviaron ellos un candidato y pensar que un dia pense que podrias ser candidato a presidente de Mexico Te duele que AmloMasFuerteQueNunca	**Neutral**	**FALSA**	**12**

Caso contrario, si la noticia fue previamente ASIGNADA como falsa, el AD la califica verdadera, entonces tal noticia es verdadera entendiéndose que la etiqueta ASIGNADA fue incorrecta, lo que implica que 37 noticias son reales de acuerdo a AD (falso, verdadero).

Del mismo modo para una etiqueta ASIGNADA como verdadera, si AD evalúa como falsa la noticia, significa que se cometió un error al ASIGNAR previamente la noticia como verdadera. Para esta situación, AD enumero 11 noticias falsas (verdadero, falso). Bajo el mismo análisis, una noticia ASIGNADA verdadera que es evaluada como verdadera por AD, significa que es una noticia real y en este espacio el número de noticias verdaderas es 19 (verdadero verdadero).

Para el algoritmo Naive Bayes (NB) la interpretación es similar. Si NB califica falsas aquellas noticias ASIGNADAS falsas, se comprueba que esas noticias efectivamente son falsas. Para este caso, se cuentan con 33 noticias falsas por NB (falso, falso). De otro modo, la noticia que se le ASIGNÓ la etiqueta de falso y NB la evalúa verdadera, se concluye que la noticia es verdadera. Esto se interpreta como un error en la ASIGNACIÓN dando como resultado 18 noticias reales de acuerdo a NB (falso, verdadero). Finalmente, para una ASIGNACIÓN verdadera, si NB devuelve el valor falso a tal noticia, quiere decir que se cometió un error al ASIGNAR esa noticia como verdadera, por tanto, NB traduce para este caso una notica verdadera a falsa. En este caso son 15 noticias falsas (verdadero, falso). Cuando una noticia ASIGNADA verdadera NB comprueba que es verdadera porque la ha calificado así, significa que es una noticia real. En este caso son 22 noticias reales identificadas por NB (verdadero verdadero).

En la tabla 4 se observa el número total de noticias falsas y reales con las predicciones que realizaron los algoritmos de Arboles de Decisión y Naive Bayes. A diferencia de la tabla 3, se registraron todas las noticias, sin selección o etiqueta. Observe que en la tabla 3 son menos noticias debido a la escogencia de las noticias por criterio de lectura.

En las tablas 5 y 6 se observan 3 noticias reales y falsas respectivamente. Estas noticias se eligieron porque a criterio de los autores son representativas del problema. Las 3 noticias reales de las tablas 4 y 5 tienen asociada la frecuencia y valor que el algoritmo Naive Bayes encontró para cada noticia

En la gráfica 1 se presenta el porcentaje de las noticias falsas y reales que se alcanzaron como resultado de los tweets con Naive Bayes., arrojando un 42% reales y 58% falsas (Fakes) con una certidumbre del 65.2 %

3.5. Sentimientos

A continuación, en las tablas 7 y 8 se muestran los resultados obtenidos por los diccionarios de sentimientos, las cuales nos indican las palabras que se tomaron en cuenta después del proceso de limpieza de datos, así como la frecuencia (Frec.) para cada palabra.

La grafica representa el porcentaje de sentimientos de la evaluación de cada tweet, donde se observa que es mayor el número de sentimientos neutrales dado que los tweets clasificados en este sentimiento pueden ser noticias reales o falsas.

En la tabla 9 se observa el total de noticias reales y falsas en relación a los sentimientos de los tweets considerando que el sentimiento es el que califica.

Fig. 1. Porcentaje de noticias reales y falsas.

Tabla 7. Lista de palabras positivas.

Nº.	Palabras	Frec,	Nº.	Palabras	Frec.
1	Trabajo	194	**15**	Honesto	63
2	Vida	150	**16**	Bueno	61
3	Admiro	134	**17**	Cambio	60
4	El Mejor	125	**18**	Héroe	60
5	Gracias	123	**19**	Favor	58
6	Información	111	**20**	Respeto	54
7	Empleo	95	**21**	Verdad	51
8	Verdad	92	**22**	Amigo	51
9	Fuerte	85	**23**	Responder	50
10	Seguridad	85	**24**	Legal	49
11	Aprobación	82	**25**	Responsabilidad	46
12	Apoyamos	75	**26**	Simpático	45
13	Abrazos	75	**27**	Amor	44
14	Besos	70	**28**	Respeto	42

Tabla 8. Lista de palabras negativas.

Nº.	Palabras	Frec.	Nº.	Palabras	Frec.
1	Muertos	250	**15**	Tonto	90
2	Delincuencia	225	**16**	Basura	85
3	Guerra	200	**17**	Mediocre	85
4	Miedo	200	**18**	Incumplido	75
5	Estupidez	189	**19**	Pobreza	75
6	Crisis	167	**20**	Enojo	75
7	Culpable	150	**21**	Cabrón	70
8	Muerte	135	**22**	Chingar	65
9	Malgasto	125	**23**	Cacas	60
10	Púdrete	125	**24**	Frustración	60
11	Miente	100	**25**	Vergüenza	55
12	Pánico	100	**26**	Mierda	50
13	Corrupto	100	**27**	Pendejo(s)	50
14	Puto	95	**28**	Difamar	45

Para interpretar la tabla 9 se determina que una noticia falsa tiene signo negativo, caso contrario, una noticia real tiene signo positivo.

Obsérvese que cuando el análisis de sentimientos evaluó la noticia como positiva y previamente esa noticia es falsa o real, entonces se considera que se integra esa noticia

Fig. 2. Porcentaje de las palabras por categoría.

Tabla 9. Tabla de relación sentimiento – tipo de noticia.

Sentimiento	Noticias Reales	Noticias Falsas
Positivo	41	36
Neutral	29	50
Negativo	25	40

Tabla 10. Definición de adulaciones, imparcialidad y descalificaciones.

Medida	Sentimiento	Noticias Reales (+)	Noticias Falsas (-)
Valor absoluto	Positivo	Adulaciones	Adulaciones
	Neutral	Imparcial / Incierto	Imparcial / Incierto
Regla de los signos	Negativo	Descalifica	Adulaciones

en una lista de *sentimientos positivos en valor absoluto*, lo cual se traduce en una aprobación hacia la 4T con un puntaje de 77.

El otro caso sucede cuando el análisis de sentimientos otorga el peso negativo. Se ajusta a *la regla de los signos* y se debe sumar de la tabla 5, el valor de 43 sobre noticias reales de aprobación a los 77 puntos que obtuvieron en *sentimientos positivos en valor absoluto*. Por tanto, la 4T obtiene 120 puntos de aprobación ("adulaciones").

Del mismo modo para puntos en contra sobre descalificación solo tiene 25 puntos y el valor de 40 de sentimiento negativo con noticia falsa se traduce a una noticia de aprobación.

Manteniendo el supuesto de que sentimientos positivos son puntos a favor de la 4T interpretándolo como consentimiento, y considerando sea válido sumar de forma lineal, en la tabla 9 obtendríamos un puntaje de 77 para la 4T y de desaprobación 65 puntos. En distintas combinaciones, la 4T sigue siendo aprobada en la red social Twitter.

Los puntajes de neutrales suman 79 puntos, lo que implica un número significativo que merece ser revisado en trabajo futuro.

En la tabla podemos observar cómo los sentimientos y las noticias influyen para poder determinar si el nivel de aprobación de la 4T y AMLO es mayor o menor.

4. Discusión y trabajo futuro

La información que es repartida dentro de las redes sociales como un medio de expresión, parecen ser un punto crítico para que las acciones del mandatario sean vistas de manera correcta e incorrecta

Algunas llegan a ser un escándalo, pero independientemente del impacto, el presidente toma algunas noticias y las comenta en su rueda de prensa diaria denominada "La mañanera". Generalmente cuando las noticias no son gratas para el presidente de México, están son desmentidas por él mismo sin importar que los medios tengan pruebas o coincidan en el contenido.

Las noticias falsas se han convertido en un recurso estratégico debido al poder de influencia social, política y económica. Ello puede ser objeto de enfrentamientos informativo, por ejemplo, para contaminar al público del rival, captar su atención y adquirir influencia sobre su comportamiento o pensamiento. Por otra parte, las noticias falsas han permitido manipular resultados electorales, provocar alteraciones en los mercados y cotizaciones bursátiles, ahondar crisis financieras e hipotéticamente, modificar la conducta y pensamiento de la opinión pública para facilitar su control y aceptación de los órdenes sociales establecidos.

Muchos son los esfuerzos por distintos autores de lingüística computacional para identificar noticias falsas y consecuentemente, existen una decena de algoritmos para atender este problema. Sin embargo, en este trabajo nos hemos ocupado de presentar una manera de resolver la inspección de noticias falsas a través de una estrategia común y pragmática dentro del análisis de sentimientos, la cual es muy conocida en PLN.

De distintas combinaciones, como puede verse en la sección de resultados, la 4T mantiene seguidores, por ejemplo, 77 a favor contra 65 puntos. En distintas combinaciones, la 4T sigue siendo aprobada en la red social Twitter

El reto de este trabajo se concentra en revisar los sentimientos neutrales y dividirlos en aprobación o descalificación, además de construir un algoritmo de clasificación para noticias falsas sobre textos extraídos de las redes sociales.

Finalmente, la ruta para identificar noticias falsas de manera exactas es larga, sin embargo, de la variabilidad de avances de diversos problemas es posible alcanzar a eliminar esas noticias.

Referencias

1. Gil, I., Rodríguez J.V.: El procesamiento del lenguaje natural aplicado al análisis del contenido de los documentos. Revista General de información y documento, pp. 205–218 (1996)
2. Chowdhury, G.: Natural language processing. In: Annual review of information science and technology, pp. 51–89 (2003)
3. Federación Internacional de Periodistas: ¿Qué son las Fake News?: guía para combatir la desinformación en la era de la posverdad. pp. 1–17 (2018)
4. Sarah Romero: La ciencia confirma que las 'fake news' se extienden más rápido que la verdad. Muyinteresante (2018)
5. Rubin, V.L, Chen, Y., Conroy, N.: Deception detection for news: three types of fakes. In: Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact:

Research in and for the Community, 83. American Society for Information Science, pp. 1–4 (2015)
6. Youngkyung, S., Chang-Sung, J.: FaGoN: Fake news detection model using grammatic transformation on neural network. In: Thirteenth International Conference on Knowledge, Information and Creativity Support Systems (KICSS), pp. 1–5 (2018)
7. Pérez, S.: Modelado colaborativo en lenguaje natural a través de redes sociales. pp. 1–13 (2018)
8. Sanabria, V.: Aplicación de técnicas de agrupamiento (clustering) para el análisis estadístico de tendencias en twitter basado en el lenguaje de programación R. pp. 15–16 (2017)
9. NLTK: Kit de Herramientas de lenguaje natural. https://www.nltk.org/ (2017)
10. TextBlob: Procesamiento de texto simplificado. https://textblob.readthedocs.io/en/dev/ (2018)
11. RStudio. https://rstudio.com/ (2017)

ISSN 1870-4069

Máquinas de soporte vectorial para pronóstico en la osteosíntesis de la fractura transtrocantérica de cadera

María Dolores Torres Soto[1], Aurora Torres Soto[2], Julio César Aguilar Gómez[3], María de la Luz Torres Soto[3], Jesús Adrián Lozano Martínez[2], Eunice Esther Ponce de León Sentí[2]

[1] Benemérita Universidad Autónoma de Aguascalientes, Departamento de Sistemas de Información, México
[2] Benemérita Universidad Autónoma de Aguascalientes, Departamento de Ciencias de la Computación, México
[3] Centenario Hospital Miguel Hidalgo, México

{mdtorres, atorres}@correo.uaa.mx, giulio-8421@live.com.mx, tishats@hotmail.com, jmajoras.link.l@gmail.com, eponce@correo.uaa.mx

Resumen. En este trabajo se presenta un análisis comparativo de 4 índices para el pronóstico de la osteosíntesis de fractura de cadera en pacientes del Centenario Hospital Miguel Hidalgo de la ciudad de Aguascalientes, México, atendidos entre Enero de 2014 y Abril de 2019. Además, se presenta una propuesta novedosa de un clasificador basado en máquinas de soporte vectorial (SVM por sus siglas en inglés) para mejorar el porcentaje de precisión de los índices reportados en la literatura con resultados muy alentadores para la población de Aguascalientes, México. El algoritmo SVM fue ejecutado con los parámetros arrojados por un experimento factorial en donde se ponen en juego 2 kernel (Rbf y poly) y tres valores para el parámetro C, así como un conjunto de 8 paquetes de datos, uno por cada índice, uno con 11 variables captadas para este estudio (incluidos los índices) y otro con una combinación de variables que produjo los mejores resultados para pronóstico de falla en la osteosíntesis de fractura transtrocantérica de cadera para la población de Aguascalientes.

Palabras clave: Máquinas de soporte vectorial, fractura transtrocantérica, índices de predicción de falla en osteosíntesis de fractura transtrocantérica de cadera.

Support Vector Machines for Prognosis in Osteosynthesis of Transtrochanteric Hip Fracture

Abstract. This work presents a comparative analysis of 4 indices for the prognosis of osteosynthesis of hip fracture in patients from the Centennial

Hospital Miguel Hidalgo in the city of Aguascalientes, Mexico, treated between January 2014 and April 2019. In addition, it is presented a novel proposal for a classifier based on support vector machines (SVM) to improve the percentage precision of the indices reported in the literature with very encouraging results for the population of Aguascalientes, Mexico. The SVM algorithm was executed with the parameters provided by a factorial experiment where 2 kernels (Rbf and poly) and three values for the parameter C are put into play, as well as a set of 8 data packages, one for each index, one with 11 variables captured for this study (including indices) and another with a combination of variables that produced the best results for prognosis of failure in osteosynthesis of transtrochanteric hip fracture for the population of Aguascalientes.

Keywords: Support vector machines, transtrochanteric fracture, failure prediction indices in transtrochanteric hip fracture osteosynthesis.

1. Introducción

En este artículo, se presentan los resultados de un análisis comparativo de índices de pronóstico de falla mediante Máquinas de Soporte Vectorial (SVM por sus siglas en inglés) para la predicción de la osteosíntesis de fractura de cadera transtrocantérica.

Para la realización de este trabajo, se contrastaron 4 índices reconocidos por la literatura en el área de traumatología: el índice Punta-Ápex (IPA), el índice Punta Ápex Modificado (IPAM), el índice Fognagnolo y el índice Parker.

Se obtuvieron los registros históricos de fractura transtrocantérica del Centenario Hospital Miguel Hidalgo del Estado de Aguascalientes, México de enero de 2014 a abril de 2019 para verificar el poder de predicción de los 4 índices anteriores, provenientes de literatura, así como indagar la posibilidad de encontrar una propuesta de pronóstico con mayor índice de precisión que los 4 índices de la literatura con respecto del pronóstico de la osteosíntesis de este tipo de fractura.

En la actualidad, la precisión del pronóstico de éxito o falla oscila mucho. De hecho, no hay clasificación de las fracturas subtrocantéreas que se relacione bien con el pronóstico [1]. Esta patología puede producir complicaciones clínicas potenciales derivadas del ingreso, la intervención quirúrgica y la pluripatología característica de la población de adultos mayores. De hecho, a esta patología, se le puede asociar un índice de mortalidad a los 6-12meses que oscila entre el 20-35% [2,3,4,5].

Las consecuencias de la falla son tan graves como la muerte del paciente, por lo que el objetivo de este trabajo es diseñar un clasificador basado en SVM para alcanzar un poder predictivo de la evolución de la osteosíntesis de factura transtrocantérica por encima del 85%. Contando con un clasificador confiable, es factible implementar medidas correctivas tempranas a pacientes en caso de falla.

En este sentido, se lograron resultados muy interesantes con poder de predicción de hasta 90.5%, cuando los índices de literatura se encuentran entre 75 % y 84% para la población de Aguascalientes, según los datos históricos del Centenario Hospital Miguel Hidalgo, que es el único Hospital de tercer nivel del Estado de Aguascalientes.

2. Conceptos importantes

2.1. Osteosíntesis de fractura de cadera

La osteosíntesis es el término que se emplea para describir la intervención mediante la cual se vuelven a alinear los fragmentos de hueso fracturado y se mantienen en posición con elementos externos metálicos. Estos pueden ser tornillos, placas o clavos. A diferencia de una prótesis de cadera, la osteosíntesis preserva la cabeza femoral original en las fracturas de cadera.

La osteosíntesis estará indicada cuando los fragmentos óseos sean considerados vitales, particularmente la cabeza femoral. Los vasos sanguíneos que nutren la cabeza femoral son muy delicados y pueden dañarse al desplazarse los fragmentos óseos, si esto ocurre, a pesar de que una osteosíntesis la vuelva a poner en su lugar, la cabeza femoral sufrirá necrosis (muerte celular), y eventualmente, colapsará siendo necesario colocar una prótesis [9].

2.2. Fractura transtrocantérica

Este tipo de fractura se somete a una intervención quirúrgica, y existen varios índices que buscan pronosticar o predecir el fallo de dicha osteosíntesis. Lamentablemente, dichos índices no son totalmente confiables y las comorbilidades asociadas a fracturas transtrocantéreas llevan a la muerte del paciente en gran parte de los casos [10].

A nivel mundial, se estimaron alrededor de 1,700,000 casos de fractura de cadera en 2007; de los cuales el 70% sucedieron en mujeres [11].

En las últimas décadas, el incremento en la esperanza de vida después de los 60 años, ha propiciado el crecimiento exponencial de fracturas en cadera, debido a numerosos factores como reducción en densidad mineral ósea y aumento en el riesgo de caídas que se producen con la edad [12]. Para el año 2050, se prevé que ocurrirán alrededor de 6,300,000 casos [11].

En México, la pirámide poblacional, muestra un incremento en la esperanza de vida; por consiguiente, la cantidad de fracturas transtrocantéricas, representará un problema de salud que requiere atención mediata por los subsistemas de salud. En el Instituto Mexicano del Seguro Social (IMSS), se registraron en un hospital de tercer nivel un total de 1,365 fracturas para el año 2007 [13]. De las cuales, 912 correspondieron a fracturas transtrocantéricas de fémur. Éstas causaron el 90% de la ocupación del servicio, con un promedio de estancia hospitalaria de 7.5 días.

Con respecto a la mortalidad que causa esta fractura; se reporta alrededor del 20% en los pacientes durante el primer año [13].

Las fracturas transtrocantéricas, son un importante problema de salud pública en México y en Aguascalientes, y debido a las complicaciones que presentan de manera secundaria a un manejo tardío e inadecuado, podrían considerarse como una urgencia quirúrgica [10].

En el Centenario Hospital Miguel Hidalgo, se atiende un número considerable de estos casos, siendo manejados con tratamiento quirúrgico la mayor parte de ellos. Dicho tratamiento es la reducción abierta y fijación interna con sistema DHS 135° para fracturas estables y no estables [10].

2.3. Índices de predicción de falla de osteosíntesis de fractura transtrocantérica

Existen índices de predicción de falla de la osteosíntesis de fracturas transtrocantéricas conocidos y validados en la literatura se enlistan a continuación.

2.3.1. Índice Punta-Ápex (IPA), (TAD-Baumgaertner)

Baumgaertner describe una escala en cuanto a la calidad de la reducción, categorizando la reducción de la fractura en buena, aceptable y pobre. Dentro de su clasificación describe como reducción buena, a aquellas fracturas en las cuales la alineación (ángulo cervicodiafisario) es normal o con tendencia al valgo en la proyección anteroposterior, así como una alineación con no más de 20 grados de angulación en la proyección lateral, y un desplazamiento no mayor de 4 milímetros de los fragmentos.

Para ser considerada como una reducción aceptable, debe contar con al menos un criterio de buena reducción, ya sea con respecto a la alineación o el desplazamiento. Por último, Baumgaertner, clasificó como una reducción pobre, a aquellas que no cumplen con ninguno de los dos criterios en términos de la alineación o desplazamiento.

2.3.2. Índice Punta-Ápex modificado al calcar (PAM)

El PAM se obtiene del mismo modo que el TAD original en la visión lateral, pero varía en la anteroposterior, obteniéndose la distancia entre la punta del tornillo y una línea adyacente a la cortical inferior del cuello femoral. En una serie de 77 pacientes los autores no observaron ninguna separación en aquellas fracturas con un PAM postquirúrgico inferior a 20.98 mm, favoreciendo la posición inferior del tornillo cefálico en la radiografía AP.

2.3.3. Índice de Parker (Cleveland)

El índice de Parker utiliza la ubicación del tornillo, esta métrica se basa en las zonas de Cleveland, la cabeza femoral se divide en tercios superior, central e inferior como se muestra en la figura 1. Se encontró que la región 1 y 9 son las más inseguras debido a su alto porcentaje de migración, y las regiones 2 y 5 se encontraron como las más seguras [6,10,14].

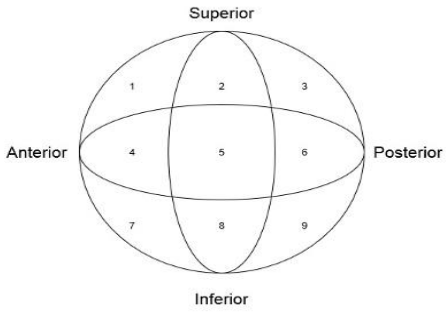


Fig. 1. Zonas de Cleveland [15].

2.3.4. Reducción postquirúrgica en varo (Fognagnolo)

Fognagnolo modifica la clasificación de Baumgaertner, agregando como dato relevante para clasificar la calidad de la reducción a un sobre-posicionamiento o contacto mayor del 80 % en ambas proyecciones [15].

Cada autor menciona que su índice es el más confiable para predecir el fallo de fracturas transtrocantéricas. Sin embargo, existe evidencia clínica de que hay pacientes sin factores positivos de estos índices de predicción de falla y que, aun así, presentan falla de la osteosíntesis durante los siguientes seis meses del postquirúrgico [10].

2.4. Máquinas de soporte vectorial

Las máquinas de soporte vectorial, máquinas de vectores de soporte o máquinas de vector soporte (Support Vector Machines, SVMs) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo de trabajo en los laboratorios AT&T.

Estos algoritmos, están estrechamente relacionados con problemas de clasificación y regresión [16]. Dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases en dos espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas como pertenecientes a una o la otra clase.

Actualmente, se han aplicado a problemas de regresión, agrupamiento y clasificación en múltiples clases presentando un muy buen desempeño [18].

La SVM encuentra el hiperplano óptimo de separación (HSO) entre clases, que se distingue de otros, por ser el que maximiza los márgenes entre las clases [20] como se muestra en la Fig. 2. Los vectores de soporte, son los ejemplos de datos que se encuentran más cercanos al HSO e influyen en su trazo [19].

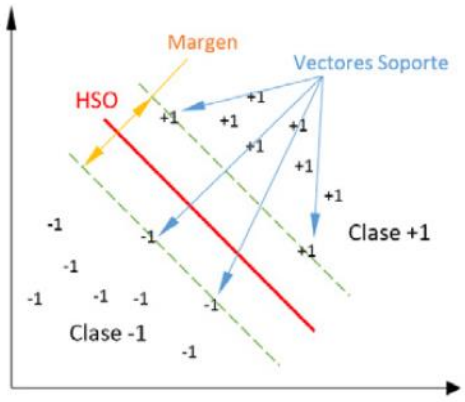


Fig. 2. Funcionamiento de una SVM [19]

Muchos problemas reales, no son problemas linealmente separables, por los que las SVM no logran encontrar un HSO. Por lo tanto, se hacen transformaciones mediante el uso de métodos de kernel que aumentan la dimensionalidad en el mapeo de las características conocido como espacio de características. De manera que si un problema originalmente se encuentra en R2 puede cambiar a R3. De esta forma, se puede encontrar una solución lineal en el espacio de características que, a su vez, es una solución no lineal en el espacio original de entrada [21].

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alto (o incluso infinito) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una correcta clasificación.

3. Material y método

En la fig. 3. Puede verse la metodología general seguida en esta investigación. A continuación, se detalla cada uno de los elementos que componen esta metodología.

En primer lugar, se trabajó con una base de datos histórica de pacientes tratados por fractura transtrocantérea de cadera entre enero de 2014 y abril de 2019 en el Centenario Hospital Miguel Hidalgo de la ciudad de Aguascalientes, México. Esta base de datos se trabajó completa, pues el archivo histórico cuenta con poco menos de 80 expedientes, a los cuáles les fueron aplicados los criterios de exclusión e inclusión correspondientes, conservando un total de 51 pacientes sometidos a tratamiento quirúrgico tras sufrir una fractura transtrocantérica de cadera. De los 51 pacientes, se contó con 13 pacientes en los que se presentó falla y 38 que no presentaron falla.

Fig. 3. Metodología.

A continuación, se realizó un preprocesamiento que consistió en una limpieza, selección y discretización de los datos obtenidos de los historiales clínicos de los pacientes con este tipo de afección que fueron sometidos a proceso quirúrgico con elementos fijadores.

Como segunda acción, se realizó la selección de los pacientes que serían considerados en la investigación. Por tratarse de una afección claramente incremental en pacientes geriátricos, se consideraron los siguientes criterios:

Criterios de inclusión. Personas de ambos sexos, mayores de 60 años, operadas en el Centro Hospital Miguel Hidalgo para el tratamiento de fracturas transtrocantéricas con tornillo de compresión dinámica con un seguimiento mínimo de 6 meses con expediente clínico y radiográfico completo.

Criterios de exclusión. Pacientes menores de 59 años, atendidos en sector privado, pacientes que recibieron tratamiento conservador o que contaron con expediente clínico y radiográfico incompleto. Sujetos con fracturas en terreno previamente patológico o con deformidades angulares preexistentes, así como pacientes con re-intervenciones quirúrgicas. Las variables utilizadas después del preprocesamiento con las categorías para cada una, se muestran en la Tabla 1.

Tabla 1. Variables utilizadas.

Variable	**Definición Conceptual**	**Categorías**
1. Género	Género biológico del paciente	1. Femenino. 2. Masculino
2. Edad	Edad Cronológica del paciente	1. 60-65 años 2. 66-79 años 3. Mayores de 80 años
3. IMC	Índice de masa corporal. Dada por el peso en kilogramos dividido por el cuadrado de la talla en metros (kg/m2).	1. Normal 2. Sobrepeso 3. Obesidad
4. Tabaquismo	Enfermedad adictiva crónica que evoluciona con recaídas. La nicotina es la sustancia responsable de la adicción, actuando a nivel del sistema nervioso central.	1. Positivo 2. Negativo
5. Enfermedades Crónico-Degenerativas	Son enfermedades de larga duración y por lo general de progresión lenta [6].	1. Diabetes 2. Hipertiroidismo 3. Hipotiroidismo 4. Insuficiencia Renal 5. Hipertensión 6.Cáncer
6. Tiempo	Tiempo transcurrido entre la lesión y el evento quirúrgico en horas.	1. Menos de 48 horas 2. Más de 48 horas
7. Estabilidad	Se refiere a la estabilidad de la fracturas.	1. Estable 2. Inestable
6. IPA	Índice Punta Ápex (Baumgaertner) (mayor a 25 = Mal pronóstico)	1. Bueno 2. Malo
7. IPAM	Índice Punta Ápex modificado al calcar (mayor a 20 mm = Mal pronóstico)	1. Bueno 2. Malo

8. IPARKER	Índice de Parker (mayor a 40 en proyección radiográfica AP = Mal pronóstico).	1. Bueno 2. Malo
9. IFOGNA	Reducción en varo (Fognagnolo) Reducción<135° de ángulo cervicodiafisariopostquirúrgico inmediato	1. Bueno 2. Aceptable 3. Malo
10. Muerte	Muerte del paciente antes de los 6 meses del postquirúrgico	1. No 2. Sí
11. Falla	Variable de Clasificación	1. No 2. Sí

Algunas de las variables que requieren un mayor nivel de descripción, por lo que para éstas, se presenta el siguiente detalle:

Con respecto del IMC, la Organización Mundial de la Salud (OMS) define el sobrepeso como un IMC >= 25, y la obesidad como IMC >= 30 [6].

Las principales enfermedades crónico-degenerativas incorporadas en el estudio son:

1. Diabetes mellitus tipo 2.
   Incapacidad del cuerpo para utilizar eficazmente la insulina, lo que a menudo es consecuencia del exceso de peso o la inactividad física.
2. Hipotiroidismo.
   Afección en la cual la glándula tiroides no produce suficiente hormona tiroidea. Esta afección a menudo se llama tiroides hipoactiva.
3. Hipertiroidismo.
   Condición en la cual existe demasiada hormona tiroidea en el organismo. En otras palabras, la glándula tiroides está hiperactiva.
4. Insuficiencia renal etapa terminal
   Incapacidad de filtrado glomerular, etapa terminal menos de 15 %.
5. Hipertensión arterial
   Trastorno en el que los vasos sanguíneos tienen una tensión persistentemente alta, lo que puede dañarlos a ellos y a otros órganos en el cuerpo.
6. Enfermedad Oncológica Concomitante
   Cáncer, con tratamiento concomitante, es decir, con medicamento más radioterapia.

En cuanto a la clasificación de las fracturas transtrocantéricas existen diversas descripciones, sin embargo, en la actualidad las clasificaciones mayormente utilizadas, son la clasificación de Evans, y la clasificación de la Arbeitsgemeinschaft für Osteosynthesefragen (AO) a estabilidad de una fractura se propone como: estables AO A1.1 a la A2.1 (Evans I) e inestables A2.2 a la A2.3 (Evans II) Índices de falla de osteosíntesis de fracturas transtrocantéricas. Las anteriores, son métricas propuestas en la literatura (Evans, 1949).

Contando con las variables para introducir al algoritmo, se procedió al diseño de la SVM haciendo uso de python 3.7 y de las librerías de sklearn para aprendizaje automático. Para todos los experimentos, se trabajó con un 80% de los datos en entrenamiento y con 20% para validación. Con el programa desarrollado, se realizó un pequeño diseño factorial para ver que el algoritmo trabajara con los mejores parámetros, de los cuáles se variaron: el parámetro *C*, que controla las violaciones al

Tabla 2. Niveles de parámetros para experimentación**.**

Parámetro	**Valor**
Kernel	1. *Rbf* 2. *Poly*
C	1. 1 2. 10,000 3. 100,000

Tabla 3. Niveles de parámetros para experimentación**.**

Combinación	**Valor**
A	Kernel *Rbf* *C=1*
B	Kernel *Rbf* *C=10,000*
C	Kernel *Rbf* *C=100,000*
D	Kernel *Poly* *C=1*
E	Kernel *Poly* *C=10,000*
F	Kernel *Poly* *C=100,000*

margen toleradas durante el ajuste del modelo [22], el kernel, que permite crear una nueva dimensión para encontrar el hiperplano de separación óptimo [23]; se seleccionaron los kernel *rbf* y *poly* y se trabajó con el valor de *gamma,* (que establece la anchura del kernel), de manera tal que si su valor es grande se presentará sobreajuste y de ser pequeño, la frontera de decisión sería muy rígida [24]. Por default gamma es *1*/*n* establecido por scikit-sklearn [25], donde *n* es el número de característica que se utilizan para la clasificación.

El establecimiento de los parámetros de experimentación se muestra en la Tabla 2. Logrando un conjunto de combinaciones que va desde la *A* hasta la *E* que se pueden observar en la Tabla 3.

Como última etapa de esta metodología, se realizaron las pruebas estadísticas con ayuda del software SPSS 20. Las variables de desempeño del algoritmo fueron: Precisión del modelo en validación, Precisión positiva, Falsos Positivos, Falsos y Precisión negativos. Con estos criterios de desempeño, se establecieron los valores de los parámetros con los que el algoritmo se ejecuta mejor y posteriormente, se obtuvieron los resultados y conclusiones de la investigación.

4. Resultados

Con respecto de la mejor combinación de parámetros para nuestra Máquina de Soporte Vectorial, se realizó un diseño factorial y se obtuvieron los cálculos estadísticos con SPSS ver. 20.

Se hicieron 30 réplicas con cada combinación de parámetros y se encontró que las variables de medición del desempeño del algoritmo no siguen una distribución normal (ninguna de ellas), por lo anterior, se realizó una pruena de kruskall Wallis para

Tabla 4. Mejores combinaciones de parámetros.

Combinación	Valor
C	Kernel *Rbf* *C*=100,000
D	Kernel *Poly* *C*=1

Tabla 5. Mejores combinaciones de parámetros.

Insumo	Poder Predictivo
Índice PA	78.18%
Índice PAM	77.79%
Índice Parker	75.0%
Índice Fognagnolo	84.0%
Usando Todas las Variables predictoras	81.8%
Propuesta Final	90.5%

establecer si las distintas combinaciones de los parámetros originaban alguna diferencia estadística en el desempeño del algoritmo. Los resultados estadísticos mostraron que, en efecto, existe diferencia entre los grupos de combinaciones variando el kernel y el parámetro C. Identificando los mejores resultados que fueron:

Se realizó una prueba U de Mann Whitney para ver si estas dos combinaciones presentaban diferencias estadísticamente significativas y se encontró que en efecto, la combinación C es mejor pues arroja menos falsos negativos.

En cuanto al poder predictivo de los diferentes índices y el poder predictivo de las variables precedentes (variables 1,2,3,4,5,6,7) presentadas en el apartado de la metodología, se puede revisar la tabla 5.

Como puede observarse, la propuesta de SVM haciendo uso de todas las variables predictoras, (esto descalifica a la variable dependiente Falla) se logró un desempeño del 81.8% (Nótese que el resultado es menos bueno que el del Ìndice Fognagnolo), sin embargo, al estudiar otras posibilidades con la SVM en sus mejores condiciones, se identificó que si se une el índice Fognagnolo con las variables 1,4,6,7 y 9 como predictoras (en la figura 4, se denomina a este experimento: Propuesta Final), el desempeño del clasificador es definitivamente mejor que el uso de los índices de manera independiente. Como puede observarse, se alcanza una mejoría de 6.5 puntos.

Encontramos que la principal potencialidad del uso de las variables 1,4,6,7, y 9 radica en que el clasificador resultante es especialmente bueno en evitar falsos negativos además de ser muy bueno en la identificación de los casos positivos y negativos. Ésta es una característica especialmente deseable, ya que los datos históricos contienen menor proporción de casos de falla que de casos de éxito. Podemos atribuir los excelentes resultados a la identificación de los mejores parámetros para cada máquina de soporte y a la selección del mejor conjunto de variables predictoras, lo que resulta en una propuesta novedosa de índice nuevo basado en el índice fognagnolo con nuevas variables a ser consideradas por la literatura.

5. Conclusiones y discusión

En relación a la problemática de un buen clasificador para apoyar al sector médico con una herramienta que proporcione por lo menos un 85% de certeza en el pronóstico de la evolución de la osteosíntesis de la fractura transtrocantérica, pudimos establecer que los índices reportados en la literatura son menos confiables de lo que se creía, pues se consideraba que se encontraban en un 80% de precisión. Cabe mencionar que con respecto al análisis comparativo de índices, podemos establecer que el mejor de ellos es el Fognagnolo, sin embargo, si agregamos información como el género, la categoría evans, la condición de tabaquismo del paciente, el tiempo transcurrido entre la fractura y su atención quirúrgica, los resultados del clasificador son mejores.

Es relevante comentar que los índices de la literatura cuentan con una precisión engañosa, pues son buenos para predecir no falla, pero no lo son para predecir falla.

Es importante mencionar, que los resultados de esta investigación superan el poder predictivo de los índices reportados en la literatura del área de traumatología en 6.5 puntos porcentuales, que implican un gran número de pacientes que pueden ser beneficiados con una atención temprana previa a la falla normalmente encontrada después de 6 meses de la intervención quirúrgica. Con los resultados presentados en este estudio, se logra proponer un nuevo índice con mejor precisión que los existentes hasta ahora en la literatura para la predicción de falla en la osteosíntesis de fractura transtrocantérica.

Referencias

1. López, E. de O.: Factores que influyen en la evolución y el pronóstico funcional de pacientes con fractura de cadera en la unidad de recuperación funcional de un hospital de media estancia (2017)
2. Sosa, M., Segarra, M.C., Hernandez, D., Gonzalez, A., Liminana, J.M., Betancor. P.: Epidemiology of proximal femoral fracture in Gran Canaria (Canary Islands). Age Ageing 22(4), pp. 285–288 (1993)
3. Pages, E., Cuxart, A,. Iborra, J., Olona, M., Bermejo, B.: Factors associated with mortality and gait impairment in elderly patients with hip fractures. Med Clin (Barc) 110(18), pp. 687–691 (1998)
4. Parker, M.J., Palmer, C.R.: Prediction of rehabilitation after hip fracture. Age Ageing 24(2), pp. 96–98 (1995)
5. Rosell, P.A., Parker, M.J.: Functional outcome after hip fracture. A 1-year prospective outcome study of 275 patients. Injury 34(7), pp. 529–532 (2003)
6. OMS: Datos interesantes acerca del envejecimiento, Envejecimiento y Ciclo de Vida. https://www.who.int/ageing/about/facts/es/ (2019)
7. INEGI: Encuesta Intercensal (2015)
8. Mendoza-Lemus, O.F., Abrego, A., Ramos, T., Alvarez, E.: Curso de Fracturas de Cuello Femoral e intertrocantéreas (2013)
9. Aguirre-Rodríguez V.H., Inzunza-Enríquez, G., Bibiano-Escalante, O., Martínez-Ruiz, F.J., Valero-González, F.S.: Osteosíntesis de cadera con clavo centromedular: posición en decúbito lateral, sin mesa de fracturas. Acta ortopédica mexicana 30(6), pp. 279–283 (2016)
10. Aguilar, J.C.: Análisis comparatico de cuatro diferentes índices de fallo en la osteosíntesis de fracturas transtrocantéricas, además de características propias de los pacientes atendidos quirúrgicamente en el centenario Hospital Miguel Hidalgo. Universidad Autónoma de Aguascalientes (2020)

11. Avellana, Z.J.A., Fernández, P.L.: Guía de Buena práctica clínica en geriatría. Anciano afecto de fractura de cadera. Elsevier (2007)
12. Moreno, J.A., García, I., Serra, J.A., Nuñez, C., Bellón, J.M.: Álvarez A. Estudio comparativo de los modelos de rehabilitación en las fracturas de cadera. Rehabilitación 3, pp. 123–131 (2006)
13. New Zealand guidelines group: Acute management and immediate rehabilitation after hip fracture, amongst people aged 65 years and over (2003)
14. Aguilar-Alcalá, A., Torres-Gómez, L.: Factores asociados a fallo en la osteosíntesis de fracturas transtrocantéricas. Acta ortopédica mexicana 31(4), pp. 189–195 (2017)
15. Cleveland, M., Bosworth, D.M., Thompson, F.R., Wilson, H.J., Ishizuka, T.: A ten-year analysis of intertrochanteric fractures of the femur. JBJS, 41(8), pp. 1399–1408 (1959)
16. Vander Plas, J.: Python dada science handbook. O'Reilly (2017)
17. Fogagnolo, F., Kfuri, M., Paccola, C.A.J.: Intramedullary fixation of pertrochanteric hip fractures with the short AO-ASIF proximal femoral nail. Archives of Orthopaedic and Trauma Surgery, 124(1), pp. 31–37 (2004)
18. Carmona, E.: Tutorial sobre Máquinas de Vectores Soporte (SVM). Madrid (2016)
19. González, R., Barrientos, A., Toapanta, M., Del Cerro, J.: Aplicación de las máquinas de soporte vectorial (SVM) al diagnóstico clínico de la enfermedad de párkinson y el temblor esencial, RIAI - Rev. Iberoam. Autom. e Inform. Ind., 14(4), pp. 394–405 (2017)
20. Betancourt, G.: Las Máquinas de Soporte Vectorial (SVMs). Sci. Tech., 27 (2005)
21. Carmona, E.: Tutorial sobre Máquinas de Vectores Soporte (SVM). Madrid (2016)
22. Amat-Rodrigo, J.: Máquinas de vector soporte (Support Vector Machines, SVM) (2020)
23. Martínez-Heras, J.: Máquinas de Vectores de Soporte (SVM). IArtificial.net (2020)
24. Salamanca-Polo, L.: Generación de conjuntos de máquinas de soporte vectorial mediante técnicas de remuestreo e inyección de ruido en las etiquetas de clase. Universidad Autónoma de Madrid (2017)
25. Scikit-learn developers: Sklearn.svm.SVC. Scikit learn (2019)

ISSN 1870-4069

Modelo de inferencia difuso en la recomendación de tallas de prendas de vestir en el comercio electrónico

Eliana Cornejo Cortez

Instituto Tecnológico de Apizaco,
México

eliana.cornejocortez.313@my.csun.edu

Resumen. El comercio electrónico de prendas de vestir enfrenta un 40 % de devoluciones de productos debido a problemas de talla. La falta de una estandarización de tallas en el mundo genera confusión durante la elección de la talla correcta. Un modelo de inferencia difuso basado en el método max-min se propone para recomendar la mejor talla en diferentes tipos de prendas y marcas utilizando las medidas corporales del usuario y las guías de tallas de las prendas. A partir de medidas como la circunferencia de la cintura, cadera y busto, es posible recomendar la mejor talla. Las recomendaciones realizadas fueron aceptables para los usuarios.

Palabras clave: Prendas de vestir, tallas, lógica difusa.

Fuzzy Inference Model in Garment Size Recommendation in E-commerce

Abstract. Apparel e-commerce faces 40% product returns due to size issues. The lack of a standardization of sizes in the world generates confusion when choosing the correct size. A fuzzy inference model based on the max-min method is proposed to recommend the best size for different types of garments and brands, using the body measurements of the user and garment size guides. From measurements such as waist, hip and bust circumference, it is possible to recommend the best size. The recommendations made were acceptable to users.

Keywords: Garment, sizes, fuzzy logic.

1. Introducción

La moda ha cambiado de la era de las prendas hechas a medida a las prendas producidas en serie. Actualmente, las tallas de las prendas están basadas en las

medidas de un modelo promedio que representa a un mercado objetivo. A partir del modelo representativo, se crean diferentes tamaños de la prenda a través de una guía de tallas, para satisfacer a una gama específica de clientes [15]. La guía de tallas se compone de medidas clave del cuerpo. Se utiliza para definir las tallas de prendas dentro de una línea de moda, y es crucial para garantizar la consistencia del ajuste en los incrementos entre cada talla [18].

Se han desarrollado varios sistemas para estandarizar los tamaños de las prendas alrededor del mundo, y al mismo tiempo, para generar confianza en los clientes al momento de realizar compras en línea. Sin embargo, el tipo de tejido, y la asimetría corporal, complican la estandarización de medidas.

En todo el mundo, la tecnología 3D se ha utilizado ampliamente para recopilar datos sobre el tamaño del cuerpo, y ha sido fundamental para los avances en el análisis de la forma del cuerpo, ropa hecha a medida, recomendaciones de estilo y talla de ropa [17]. Sin embargo, el problema de las prendas mal ajustadas aún persiste y esto es una preocupación importante para los minoristas y consumidores.

Se propone un modelo de inferencia difuso basado en el método max-min, el cual devuelve una recomendación de talla, de acuerdo a las medidas ingresadas por el usuario y la guía de tallas proporcionada. A través del uso de las curvas gaussianas, estableciendo como parámetros los valores óptimos de cada talla de una prenda, y la variación que existe entre las tallas, se determina la talla que mejor se ajusta.

El principal objetivo es disminuir el número de devoluciones en el comercio electrónico de prendas de vestir, a través de una recomendación de talla, tomando como entrada las medidas del usuario y el tipo de prenda seleccionada.

El documento se divide en cinco secciones, distribuidas de la siguiente forma: La sección 2 describe propuestas de solución, las aplicaciones disponibles en el mercado actual y un análisis de las herramientas de recomendación de tallas en México y en las principales marcas a nivel mundial. La sección 3 describe los conceptos y características de una guía de tallas. La sección 4 muestra el proceso de desarrollo del modelo de recomendación de tallas propuesto, experimentos y resultados. La sección 5 muestra las conclusiones.

2. Trabajo relacionado

Existen diversas propuestas de solución para el problema de la elección de tallas en el comercio electrónico. En [16] se describe el uso de un modelo tridimensional para representar al cliente, generado a partir de parámetros dados por el usuario. El sistema sugiere la mejor talla con base en la prenda seleccionada.

En [19] se propone un prototipo de un probador virtual basado en realidad aumentada. El usuario puede ver un modelo 3D personalizado vistiendo prendas previamente almacenadas. La selección de la prenda se realiza a través del escaneo de la etiqueta con un dispositivo móvil. El prototipo está diseñado específicamente para hombres. En [12] se describe un modelo similar en el que se selecciona un modelo genérico de hombre o mujer. Posteriormente, se modifica

conforme las medidas proporcionadas por el usuario. En [11] se muestra un modelo para probar camisetas, utilizando un Kinect V2. Plantea un mecanismo de fragmentación de las prendas en cuatro partes, para que la superposición de las camisetas sobre el cuerpo virtual sea más precisa. Las entradas son los puntos que proporciona el Kinect. En [13] se plantea un método para vestir un modelo tridimensional utilizando un módulo de detección de colisiones. Se ajusta automáticamente al maniquí en cuanto la prenda se coloca cerca de él, detectando los vértices de las prendas, e iterando hasta que se adapten completamente al cuerpo. En [14] se propone un sistema de recomendación de tallas basado en un enfoque difuso y funciones triangulares.

Sin embargo, gran parte de las propuestas de solución anteriores involucran el uso de sensores o cámaras, a las cuales no todos los usuarios de comercio electrónico tienen un fácil acceso.

Actualmente, hay compañías que ofrecen nuevas interfaces en línea que buscan ayudar a los consumidores a elegir prendas que se ajusten adecuadamente a ellos durante el proceso de compra, las cuales se describen a continuación.

2.1. Aplicaciones en el mercado actual

Metail. Es una herramienta que ayuda a diseñadores de moda a tener una idea rápida de cómo se verán sus diseños cuando los usen sus consumidores. A través de fotografías del modelo, se realiza el montaje en un avatar para simular diferentes prendas [4].

Fit Finder. Es una aplicación que se instala en tiendas en línea asociadas. El usuario responde entre cuatro y nueve preguntas. La aplicación devuelve una recomendación de talla, así como la probabilidad de que la talla s.ea la adecuada. Fit finder considera las respuestas del usuario y compara con usuarios que poseen rasgos similares [3].

Model My Outfit Permite la personalización de un avatar para probar diferentes prendas. La implementación para otros sitios web no está habilitada, y el número de prendas para probarse es limitado. Su principal ventaja es la facilidad y la cantidad de elementos para editar el avatar conforme los gustos y preferencias del usuario. [5].

NaizFit. Es una aplicación que se integra en tiendas en línea. Permite la recomendación de talla utilizando cinco parámetros que son pedidos al consumidor al momento de solicitar ayuda para encontrar su talla. Devuelve una o dos tallas sugeridas, así como la probabilidad de que la talla sea la adecuada [6].

Virtusize. Es una aplicación que se integra a las tiendas en línea, con el fin de permitir la comparación de dos prendas: una prenda favorita comprada anteriormente y una prenda que se desea comprar. El objetivo de la comparación es la

recomendación de la talla. Las medidas requeridas dependen del tipo de prenda a comparar. Como resultado devuelve la talla sugerida, y permite visualizar la diferencia de tamaño entre ambas prendas, el porcentaje de similitud, y comparar con otras tallas [9].

True Confidence. Utliza algoritmos de aprendizaje automático, y genera recomendaciones de ropa y calzado. Analiza los datos del comprador con personas que tengan medidas similares para ofrecer recomendaciones de talla [8]. Actualmente, ninguna de estas aplicaciones tiene una evolucionada interfaz para el ajuste virtual, pero muestran las posibilidades de utilizar la tecnología para mejorar la experiencia virtual del producto y brindar al consumidor una experiencia inmersiva antes de la compra. El objetivo de estas aplicaciones de software se centra en reducir las devoluciones de los clientes, a través de una experiencia virtual mejorada.

2.2. Panorama actual: El uso de herramientas de recomendación de talla en el comercio electrónico de prendas de vestir en México y en el mundo

De acuerdo a las principales marcas de prendas de vestir en el mundo, se revisaron sitios web con diferentes mercados objetivo para identificar cómo el usuario busca la talla en cada uno de ellos. Se identificaron cuatro tipos, los cuales se enumeran a continuación:

1. Utilizan una guía de tallas para que el usuario pueda identificar su talla conforme sus medidas.
2. Utilizan TrueFit o FitFinder, o alguna de las aplicaciones mencionadas en la sección anterior para encontrar la talla.
3. Utilizan una lista de tallas disponibles. No incluye una guía de tallas o referencias para identificar las dimensiones de las prendas.
4. No venden en línea. Por lo tanto, no incluye guía de tallas, ni lista de tallas disponibles, únicamente fotografías de sus artículos.

De los sitios analizados, el 52 % utiliza guía de tallas, entre las cuales se encuentran Nike, Puma, Tommy Hilfiger y Bershka. El 16 % no utiliza guía de tallas como Calvin Klein o American Eagle, y únicamente colocan la lista de las tallas disponibles por cada prenda. El 12 % no efectúa ventas en línea, y el 20 % utilizan aplicaciones como TrueFit o FitFinder, tales como Zara y Aeropostale.

Conforme los datos anteriores, se observa que el 80 % de los sitios no utilizan una aplicación como TrueFit o FitFinder para encontrar la talla de sus clientes.

En México, de un total de 87 sitios evaluados, únicamente 18 presentaron una guía de tallas para que el usuario pueda elegir una talla. El resto de los sitios únicamente proporciona su lista de tallas disponibles, sin dar ningún indicio al consumidor de qué medidas utilizan en las prendas, causando que el comprador desista en efectuar una compra.

3. Guías de tallas

En [17] se define una guía de tallas como una "tabla de datos que muestra las medidas del cuerpo o de la prenda representada con una talla".

Los componentes generales que constituyen una guía de tallas se describen a continuación:

a) *Encabezado.* Indica el tipo de prenda.
b) *Etiquetas.* Representan un rango de medidas corporales que encajan en la talla.
c) *Punto de medición individual.* Medida de cintura, cadera, busto, hombros, tomada en centímetros o en pulgadas.

La Tabla 1 especifica las partes de una guía de tallas para pantalones de mujer.

Tabla 1. Ejemplo de guía de tallas de pantalones para mujer. Se especifican las partes que componen la guía de tallas: a) Encabezado b) Etiquetas de talla c) Puntos de medición individual.

	PANTALONES (a)	
TALLA (b)	Circunferencia de cintura (c)	Altura de la entrepierna (c)
10	74 cm	73 cm
12	80 cm	74 cm
14	86 cm	73 cm

La creación de una guía de tallas se puede resumir mediante los siguientes pasos:

1. Establecer dimensiones clave,
2. Segmentar la población en grupos,
3. Definir intervalos entre dimensiones clave,
4. Diseñar el sistema de tallas para la cobertura de una población.

En la creación de guías de tallas, es necesario establecer intervalos entre las diferentes tallas. Estas son las diferencias de números enteros entre las dimensiones clave. Es un número difícil de decidir, debido a que pequeñas variaciones, incluso de hasta 0,5 cm en algunas dimensiones, pueden ser notables y suficientes para impactar en la satisfacción del ajuste. Por tal motivo, los fabricantes utilizan diferentes intervalos.

Se han empleado datos de escaneos 3D de poblaciones para determinar las medidas requeridas y adaptarse mejor a una población variada.

Cuantas más mediciones se usen para estandarizar un sistema de tallas, menor será la proporción de la población que cubrirá cada talla.

4. Modelo difuso para la recomendación de tallas: metodología

El primer paso de la metodología, es la elección de la prenda, la cual debe tener una guía de tallas definida por el fabricante o minorista. El detalle de los pasos se describen a continuación:

1. El fabricante define la guía de tallas.
2. Procesamiento de la guía de tallas.
3. Aplicación de la función de pertenencia.
4. Aplicación del método max-min para encontrar la talla del usuario.

4.1. El fabricante define la guía de tallas

El tamaño de la guía de tallas es variable respecto al número de tallas que presente cada fabricante. Ejemplo de guías de tallas de diferentes marcas se muestran en las Tablas 2, 3 y 4.

Tabla 2. Guía de tallas para blusas y sudaderas de la tienda en línea de Adidas.

TALLA	BUSTO	CINTURA	CADERA
XXS	74.5 cm	58.5 cm	83.5 cm
XS	79.5 cm	63.5 cm	88.5 cm
S	85.5 cm	69.5 cm	94.5 cm
M	91.5 cm	75.5 cm	100.5 cm
L	98 cm	82 cm	107 cm
XL	105.5 cm	90 cm	114 cm
XXL	108 cm	93.5 cm	115.5 cm
1X	114 cm	99 cm	121.5 cm
2X	118.5 cm	103.5 cm	125.5 cm
3X	128.5 cm	113.5 cm	135.5 cm
4X	138.5 cm	123.5 cm	145.5 cm

Tabla 3. Guía de tallas de blusas de mujer de la marca Mango.

TALLA	BUSTO	CINTURA	CADERA
S	83-88 cm	67-72 cm	92-97 cm
M	89-94 cm	73-78 cm	98-103 cm
L	95-101 cm	79-85 cm	104-110 cm
XL	102-109 cm	86-94 cm	111-117 cm

Tabla 4. Guía de tallas de blusas, sacos y abrigos de mujer de la marca Michael Kors.

TALLA	BUSTO	CINTURA	CADERA
00	79 cm	61 cm	85 cm
0	81 cm	64 cm	88 cm
2	84 cm	66 cm	90 cm
4	86 cm	69 cm	93 cm
6	89 cm	71 cm	95 cm
8	91 cm	74 cm	98 cm
10	95 cm	77 cm	102 cm
12	99 cm	81 cm	105 cm
14	103 cm	85 cm	109 cm
16	107 cm	89 cm	113 cm
18	110 cm	93 cm	117 cm

4.2. Procesamiento de la guía de tallas

Los datos que se obtienen con el análisis de la guía son los siguientes:

1. Número de tallas de la prenda.
2. Número de medidas requeridas para encontrar la talla.
3. Tipo de cuerpo para el cual la prenda esta confeccionada.
4. Intervalos entre cada talla. Distancia de una talla a la siguiente.

Con el fin de ejemplificar cada uno de los elementos anteriores, se utilizan las guías de tallas de las Tablas 2, 3 y 4 y para demostrar las diferencias entre tallas de cada marca. Los datos obtenidos se muestran en la Tabla 5.

Tabla 5. Tabla comparativa de las variaciones de tallas entre tres diferentes marcas.

	Adidas	**Michael Kors**	**Mango**
Número de tallas	11	11	4
Número de medidas	3	3	3
Desviación estándar busto	20.307	10.5589	8.5963
Desviación estándar cintura	20.759	10.5296	8.7987
Desviación estándar cadera	19.471	10.5486	8.3964
Proporcionalidad: Busto-cadera	0.92	0.9345	0.9141
Proporcionalidad: Cintura-cadera	0.78	0.75 56	0.7606

Las desviaciones estándar indican que los incrementos entre cada talla no son proporcionales. Para Adidas y Mango, los incrementos en la medida de la cintura entre cada talla es ligeramente mayor que las otras dos medidas (busto y cadera).

Las constantes de proporcionalidad determinan la diferencia entre cada medida (busto, cintura y cadera). En este caso, en qué proporción la medida de la cintura y la medida del busto aumentan o disminuyen con base en la medida de la cadera.

4.3. Aplicación de la función de pertenencia Gaussiana

La función de pertenencia de un conjunto nos indica el grado en que cada elemento pertenece a dicho conjunto [21]. Se determinó utilizar la función Gaussiana, para obtener un resultado consistente y diferenciable en todos los puntos. El número de funciones utilizadas es igual al número de tallas multiplicado por el número de medidas obtenidas de la guía de tallas.

Para crear la matriz de relación difusa, se consideran los siguientes elementos:

1. Las medidas de la prenda (valores óptimos por cada talla).
2. Las medidas del usuario.
3. La desviación estándar de cada una de las medidas de la prenda.
4. El coeficiente de proporcionalidad entre las medidas de la prenda.

Considerando la ecuación de la función de pertenencia Gaussiana, la cual se define como:

$$e^{(\frac{-(x-c)^2}{2\sigma^2})} \tag{1}$$

donde x es la medida del cliente, en este caso, la medida de la circunferencia de la cintura, cadera o busto. c es el valor de ajuste perfecto para la talla seleccionada, y σ la desviación estándar de la medida de la prenda.

Las medidas deben ser tomadas como se indica en la norma [10].

4.4. Aplicación del método max-min

Una vez obtenida la matriz de relación difusa, el método se determina por:

1. Cálculo de las intersecciones de las tres medidas para cada una de las tallas.
2. Cálculo del máximo de las intersecciones por talla.
3. Sugerencia de talla basado en el mayor grado de pertenencia. Ver Figura 1.

La interfaz del modelo se puede apreciar en la Figura 2, en donde en primera instancia debe seleccionarse la marca, y posteriormente la inserción de las medidas solicitadas.

4.5. Experimentos y resultados

Recomendación de talla en tres diferentes marcas Para demostrar la eficiencia de la recomendación de talla en diferentes marcas y con diferentes tipos de tallas, para el primer experimento se tomaron como base las medidas de un mismo usuario, y se utilizaron las guías de tallas de las marcas de Adidas,

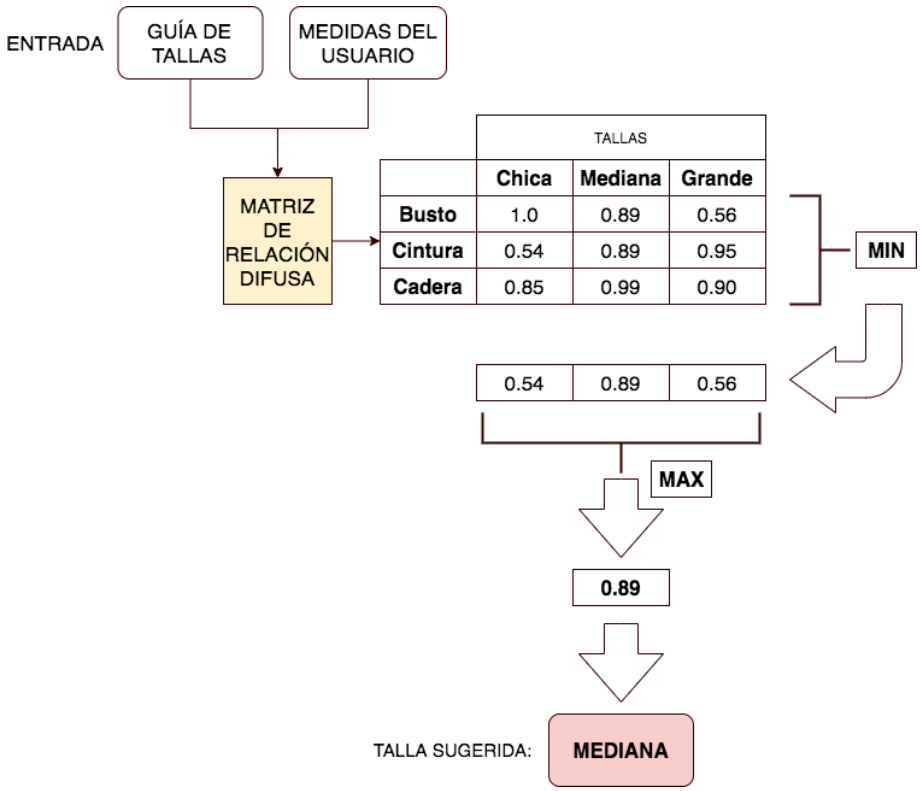


Fig. 1. Proceso general. A partir de la guía de tallas de la prenda y las medidas del usuario, se obtiene una matriz de relación difusa, utilizando la función Gaussiana. Al aplicar el método max-min, se obtiene la recomendación la talla.

Mango y Michael Kors (Ver Tablas 2, 3 y 4). Cabe mencionar que cada marca maneja distintos nombres y números de talla, así como los intervalos entre cada una.

Las medidas utilizadas fueron las siguientes:

- Busto: 88 cm,
- Cintura: 73 cm,
- Cadera: 99 cm.

La Figura 3 muestra las gráficas para las cuatro tallas de la marca Mango, sombreando el grado de pertenencia que tiene el usuario a cada una de las tallas.

De acuerdo a la Figura 3, se observan los siguientes puntos:

1. Se muestran tres campanas de Gauss por cada talla. La primera representa la medida óptima de la cintura, la segunda y tercera, del busto y cadera respectivamente.

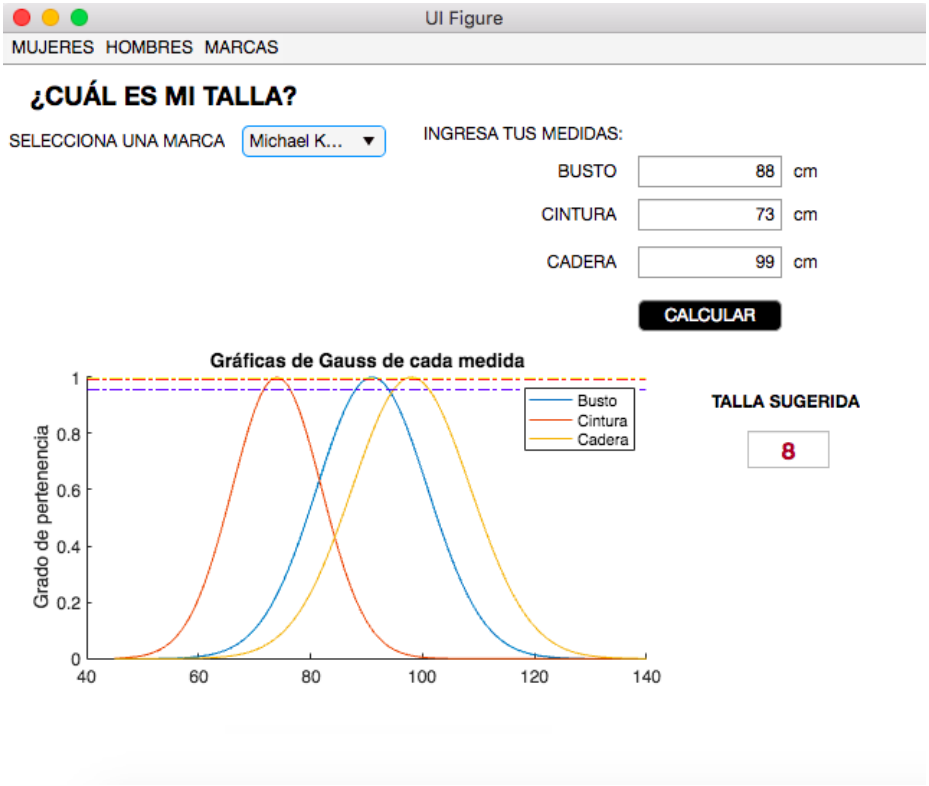


Fig. 2. Interfaz de la aplicación en la que se demuestra con los datos ingresados, el nivel de pertenencia a la talla 8. Las líneas punteadas indican el grado de pertenencia a cada medida.

2. Cada una indica el grado de pertenencia de las medidas del usuario respecto al de la prenda. Por lo tanto, las partes sombreadas en color negro indican el grado total de pertenencia a una talla.

Se sugiere la talla cuya unión sea mayor en las tres medidas. La talla sugerida para cada marca fueron las siguientes:

– Marca Adidas: Talla M,
– Marca Mango: Talla M,
– Marca Michael Kors: Talla 8.

Conforme a los resultados anteriores, se observa que el mismo nombre de talla no determina que las medidas de la prenda sean las mismas en diferentes marcas.

El modelo siempre sugerirá la mejor talla disponible de acuerdo a las medidas proporcionadas, por lo cual fue necesario implementar una condición en la que

si el grado de pertenencia obtenido era menor a 0.85 en más de una medida, indicar una talla no disponible para el usuario.

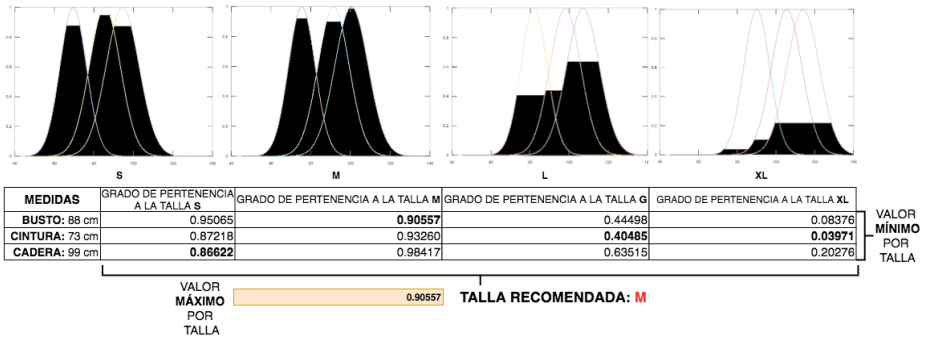


Fig. 3. Grados de pertenencia para cada talla en la marca Mango. Después de aplicar el método max-min, la talla M es recomendada.

Recomendación de talla en una misma marca y creación de una guía de talla. Para demostrar la estabilidad del modelo, se probó una camisa de mujer de la marca *Love Tree* en sus tres diferentes tallas: *S*, *M* y *L*, a 22 mujeres de un rango de edades de 18 a 55 años. Se utilizaron tres medidas de la prenda y de los participantes para realizar el experimento. Para la obtención de la guía de tallas de la prenda, se utilizó la aplicación en línea Sizely [7], y una cinta métrica para determinar los valores óptimos de cada talla. La guía de tallas completa se muestra en la Tabla 6.

Tabla 6. Guía de tallas de camisas de mujer de la marca Love Tree.

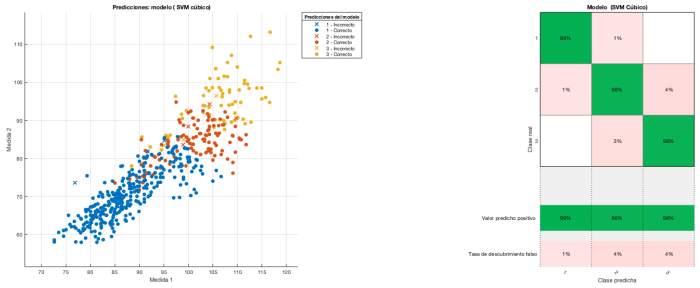
TALLA	BUSTO	CINTURA	CADERA
S	84-94 cm	70-99 cm	80-86 cm
M	95-101 cm	100-106 cm	87-94 cm
L	102-109 cm	107-110 cm	95-108 cm

La Tabla 7 muestra un fragmento de los datos recabados de las medidas de las participantes, la talla sugerida, la talla real, y si estaban conformes con el ajuste de la prenda, es decir, si el ajuste es adecuado, holgado o ajustado.

De las 22 participantes, el número de tallas sugeridas correctas fue del 100 %; por otra parte, la satisfacción de ajuste (holgado/ajustado/adecuado) fue del 86 %.

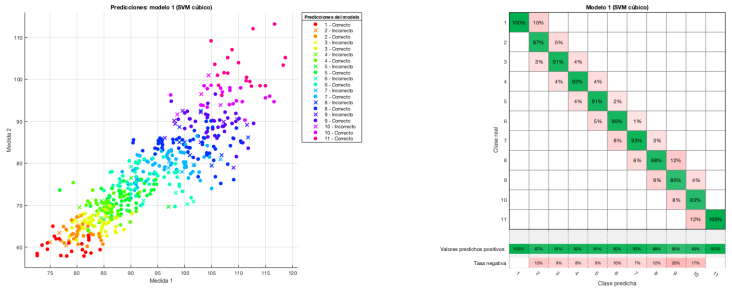
Tabla 7. Fragmento de las medidas tomadas a las 22 participantes. Se especifica la talla recomendada, la talla aprobada por el usuario, y la satisfacción de ajuste de la prenda.

Busto	Cintura	Cadera	Talla sugerida	Talla real	Ajuste
88 cm	73 cm	84 cm	S	S	Holgado
99.5 cm	92.5 cm	84 cm	M	M	Adecuado
101 cm	90 cm	94 cm	M	M	Adecuado



(a) Gráfico de dispersión (b) Matriz de confusión

Fig. 4. Gráficas obtenidas del modelo utilizando la base de datos de [1] y la guía de tallas de la Tabla 6. Precisión: 97.8 %. Total de datos no clasificados: 11 de 508.



(a) Gráfico de dispersión (b) Matriz de confusión

Fig. 5. Gráficas obtenidas del modelo utilizando la base de datos de [1] y la guía de tallas de la Tabla 4. Precisión: 90.1 %. Total de datos no clasificados: 51 de 508.

Validación del modelo Del mismo modo, el algoritmo fue sometido a la predicción de tallas de las medidas corporales de tres bases de datos. Las bases

de datos *body measurements* y *girls*, obtenidas de [1] y [2], las cuales contienen medidas de busto, cintura y cadera, con 508 y 50 registros, respectivamente. La tercera base de datos fue creada y diseñada para generar medidas aleatorias para nueve diferentes tipos de cuerpos, descritos en [20], con 527 registros.

Las guías de tallas utilizadas para este experimento fueron las de la marca Michael Kors y Love Tree, las cuales contiene 11 y 3 tallas respectivamente. Se utilizó una validación cruzada de cinco particiones para cada base de datos. Posteriormente, se probaron cinco clasificadores (SVM, KNN, árboles de decisión, Naive Bayes y análisis discriminante linear y cuadrático) con distintas variaciones para identificar el mejor. Los resultados arrojaron resultados entre 90 % y 98 % de precisión utilizando el clasificador SVM cúbico. Los gráficos de dispersión y las matrices de confusión se muestran en la Figura 4 y en la Figura 5.

5. Conclusión

Actualmente, el comercio electrónico de prendas de vestir representa un problema en el momento de seleccionar la talla adecuada. El costo de devoluciones y la insatisfacción de los clientes es el principal problema. Se planteó un modelo de inferencia difuso, para sugerir la mejor talla de acuerdo a las medidas del cliente y las medidas de la prenda. Los resultados demuestran que un porcentaje significativo de sugerencias de talla son aceptables para las medidas dadas. Las prendas de vestir están diseñadas para un tipo específico de cuerpo, lo que influye en que no sea posible sugerir en todos los casos un tamaño adecuado debido a las variaciones en los tipos de cuerpos y la asimetría corporal. Por otra parte, considerando que el 100 % de los e-commerce de prendas de vestir en el país no cuentan con ningún tipo de recomendador de talla, la implementación del modelo no genera una inversión significativa para ser incluida en cualquier tienda en línea, convirtiéndose en una opción factible para pequeñas y medianas empresas en México.

Referencias

1. Body Measurements (2018). https://doi.org/10.1001/jama.1970.03180130069023, https://www.kaggle.com/vikrant4/bdims
2. Girls dataset (2019), https://www.kaggle.com/kyoyasuhisa/girls-dataset
3. Fit finder (2020), https://www.fitanalytics.com/fit-finder
4. Metail (2020), https://metail.com/
5. Model my outfit (2020), http://modelmyoutfit.com/
6. Naizfit (2020), https://www.naiz.fit/fashion-ecommerce/
7. Sizely (2020), https://www.size.ly/
8. True Confidence (2020), https://www.truefit.com/Products/Confidence-Engine
9. Virtusize (2020), http://www.virtusize.com/site/how-it-works
10. ISO: ISO 8559-1:2017 Size designation of clothes — Part 1: Anthropometric definitions for body measurement. 1 edn. (2017), https://www.iso.org/standard/61686.html

11. Kanduri, Y., Prasad, S.D.: A modular approach for cloth modelling in virtual fitting room. In: 11th International Conference on Industrial and Information Systems, ICIIS 2016 - Conference Proceedings (2016). https://doi.org/10.1109/ICIINFS.2016.8262916
12. Kunkhet, A.: Designing an application of 3D character modelling in trialling size of clothes on online clothing shopping. In: 2nd Joint International Conference on Digital Arts, Media and Technology 2017: Digital Economy for Sustainable Growth, ICDAMT 2017 (2017). https://doi.org/10.1109/ICDAMT.2017.7904975
13. Li, H., Liu, Z., Wang, W.: A more practical automatic dressing method for clothes animation. Proceedings - 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016 **2**, 253–256 (2016). https://doi.org/10.1109/IHMSC.2016.220
14. Nasibov, E., Vahaplar, A., Demir, M., Okur, B.: A fuzzy logic Approach to predict the best fitted apparel size in online marketing. In: Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings (2017). https://doi.org/10.1109/ICAICT.2016.7991773
15. Nayak, R., Padhye, R.: Garment Manufacturing Technology (2015). https://doi.org/10.1016/C2013-0-16494-X
16. Noordin, S., Ashaari, N.S., Wook, T.S.M.T.: Virtual fitting room: The needs for usability and profound emotional elements. In: Proceedings of the 2017 6th International Conference on Electrical Engineering and Informatics: Sustainable Society Through Digital Innovation, ICEEI 2017 (2018). https://doi.org/10.1109/ICEEI.2017.8312427
17. Pandarum, R., Yu, W.: Garment sizing and fit. In: Garment Manufacturing Technology (2015). https://doi.org/10.1016/B978-1-78242-232-7.00007-2
18. Piller, F.T., Tseng, M.M.: Handbook of research in mass customization and personalization (2009). https://doi.org/10.1142/P7378
19. Sekhavat, Y.A.: Privacy preserving cloth try-on using mobile augmented reality. IEEE Transactions on Multimedia (2017). https://doi.org/10.1109/TMM.2016.2639380
20. Simmons, K., Istook, C.L., Devarajan, P.: Female Figure Identification Technique (FFIT) for apparel part I: Describing female shapes. Journal of Textile and Apparel, Technology and Management **4**(1) (2004)
21. Veit, D.: Fuzzy logic and its application to textile technology. In: Simulation in Textile Technology: Theory and Applications (2012). https://doi.org/10.1533/9780857097088.112

ISSN 1870-4069

Reconocimiento de trayectorias del brazo usando sensores de aceleración y giroscopio

Pedro López-Rodríguez, José Ruiz-Pinales, Juan Gabriel Aviña-Cervantes, José Luis López-Ramírez, José Guadalupe Beltrán-Hernández

Universidad de Guanajuato,
División de Ingenierías, Campus Irapuato Salamanca,
Departamento de Ingeniería Electrónica,
México

{p.lopez.rodriguez, pinales, avina,
jl.lopezramirez, jg.beltranhernandez}@ugto.mx

Resumen. Debido a la gran variedad de dispositivos inteligentes existentes en la actualidad los cuales están dotados con un gran número de sensores como: giroscopios, acelerómetros, magnetómetros, termómetros, entre otros. Dichos sensores son utilizados en el desarrollo y diseño de interfaces hombre-máquina cuya finalidad puede ser el monitoreo de la actividad del usuario, manipulación o manejo de una máquina sin interacción físicamente con la misma. Este trabajo presenta el reconocimiento de 12 trayectorias distintas y continuas del brazo utilizando datos precedentes de los tres ejes del sensor de aceleración y giroscopio del brazalete Myo™. Para realizar el reconocimiento de la trayectoria, se ha implementado y entrenado una red neuronal convolucional para la clasificación de las trayectorias. Finalmente, se realizó una validación cruzada para la validar la funcionalidad del sistema propuesto y cuyos resultados obtenidos han sido del 99.97 % de exactitud durante la clasificación.

Palabras clave: Acelerómetro, giroscopio, inteligencia artificial, redes neuronales convolucionales.

Arm Trajectory Recognition using Acceleration and Gyroscope Sensors

Abstract. Due to the great variety of intelligent devices existing today which are equipped with many sensors such as: gyroscopes, accelerometers, magnetometers, thermometers, among others. These sensors are used in the development and design of human-machine interfaces, whose aim could be to monitor user activity, manipulate or operate a machine without physically interacting with it. This work presents the recognition of 12 different and continuous trajectories of the arm using the data from the three axes of the acceleration sensor and gyroscope of the Myo™ armband. To perform trajectories recognition,

a convolutional neural network has been implemented and trained. Finally, a cross validation was performed to validate the functionality of the proposed system and whose results have been 99.97 % accurate during classification.

Keywords: Accelerometer, gyroscope, artificial intelligence, convolutional neural networks.

1. Introducción

Al hablar del reconocimiento de gestos se hace referencia al proceso mediante cual se realiza la comprensión y clasificación de movimientos significativos de las manos, brazos, la cara y en algunas ocasiones movimientos con la cabeza. El reconocimiento de gestos se ha convertido en un campo de investigación de gran importancia, ya que se utiliza en el desarrollo de interfaces hombre-máquina.

El desarrollo acelerado de dispositivos inteligentes (teléfonos, pulseras, relojes, etc.) y a su vez dotados de una gran cantidad de sensores (aceleración, giroscopio, presión, etc.) ha promovido una constante evolución en el desarrollo de interfaces hombre-máquina [10, 18, 23, 36]. Incluso se pueden encontrar patentes donde se utilizan dispositivos inteligentes para el reconocimiento de escritura utilizando los sensores de aceleración que se encuentran en dichos dispositivos [26, 33]. Otra área que ha sido de interés para los investigadores o desarrolladores de aplicaciones (App) inteligentes es la utilización de datos de los sensores de dichos dispositivos para realizar inferencias en diferentes aspectos de la vida [34]. Comúnmente se desarrollan aplicaciones para el monitoreo de la salud, registro del estado físico y de seguridad [17, 28, 29].

Para llevar a cabo la tarea de reconocimiento de la actividad humana mediante el uso de sensores o por el uso de otros medios de medición de la actividad, se deben tener en cuenta dos acciones principalmente. La primera acción que se tiene que atender para llevar a cabo el reconocimiento, es realizar la segmentación de las actividades durante el monitoreo. Para el caso de señales eléctricas como es el caso del presente trabajo, es llevar a cabo una segmentación en pasos de tiempo de la señal, dicha tarea se puede realizar mediante el uso de ventanas deslizantes de longitud fija y dividir cada señal en segmentos iguales. Sin embargo, al realizar la acción de segmentación por ventaneo surge la cuestión sobre la longitud de la ventana para tener una buena precisión del reconocimiento o clasificación de la señal. Para llevar a cabo esta tarea, se han planteado diferentes soluciones [18, 19, 22].

La segunda acción para realizar es una extracción de características de la señal segmentada para tener una buena clasificación o reconocimiento de la señal. Esta acción puede ser mediante el uso de metodologías tradicionales como la FFT (*Fast Fourier Transform*) [15] , transformada wavelet [15], modelos autorregresivos [30], entropía espectral [25], etc. Sin embargo, con el desarrollo de la inteligencia artificial (AI por sus siglas en inglés), se tiene un enfoque diferente en la extracción de características basada en el aprendizaje profundo o *Deep Learning*, cuya idea principal radica en el aprendizaje automático de las

características directamente de los datos en bruto [18,23]. Sin embargo, se puede hacer uso de algunas técnicas de filtrado o preprocesamiento de la señal como parte de un suavizado de los datos, cuya finalidad es tener una señal lo más limpia posible para el uso de técnicas de AI [36].

En este trabajo, se propone una arquitectura de red neuronal convolucional (CNN por sus siglas en inglés) para el reconocimiento de trayectorias del brazo utilizando datos procedentes de un acelerómetro o giroscopio. La razón de usar CNNs es que éstas pueden llevar a cabo de manera óptima la tarea de extracción de características directamente sobre los datos en bruto sin necesidad de usar métodos heurísticos de extracción de características, lo cual podría ayudar a resolver algunos problemas debido a oscilaciones transitorias como es descrito en [15]. Además, las CNNs constituyen una de las más poderosas herramientas para la clasificación de patrones, señales, imágenes, etc.

2. Trabajos relacionados

Como ya se ha mencionado, existen trabajos previos relacionados al reconocimiento de gestos, escritura o trayectorias de la mano mediante el uso de dispositivos como relojes o celulares inteligentes, pulseras, guantes o algún otro equipo especializado dotado de un gran número de sensores, dentro de los cuales se encuentran el acelerómetro y giroscopio [3, 8, 10]. Estos últimos, fueron los utilizados para el desarrollo del presente trabajo, cuya finalidad es medir el cambio de aceleración y orientación en los movimientos de la mano o brazo respectivamente. Para poder realizar el reconocimiento de los movimientos, se tiene como primer paso, realizar la segmentación (comienzo y final) de cada gesto o trazo de forma adecuada, esta acción puede realizarse de forma automática como en [12] o mediante una inspección visual de la señal y realizarlo de forma manual [27]. En nuestro estudio, la segmentación se realizó de forma automática haciendo uso de la pantalla táctil de una laptop, la cual se utilizó como plantilla de trabajo para la realización de los diferentes trazos y así recopilar la información únicamente durante la realización del trazo mediante la técnica *PenUp-PenDown*, tal y como se plantea en [3].

En [12] utilizaron un teléfono inteligente para adquirir la señal del acelerómetro y giroscopio, realizando un preprocesamiento de los datos antes de utilizar el método DTW (*Dynamic Time-Warping*) como sistema de reconocimiento [1]. En [27], presentan un método capaz de reconocer la flexión de los dedos de la mano utilizando un guante dotado por acelerómetros y de un giroscopio Para la etapa de reconocimiento en [27], los autores utilizaron cinco clasificadores: Red Neuronal Estática (FNN) [13], Support Vector Machines (SVM) [5], k-Nearest Neighbors (kNN) [7], Long Short-Term Memory (STM) [16] y Gated Recurrent Unit (GRU) [6].

En [8], se presenta un algoritmo llamado 2-DifViz, que convierte los movimientos inerciales de la mano realizados en el aire (capturados con la pulsera Myo™) en secuencias de coordenadas (x, y) a un plano Cartesiano 2-D, para después realizar el reconocimiento de los dígitos comprendidos en Z={0,1, ...,

9}, dibujados en el aire mediante la implementación de modelos de aprendizaje profundo como lo son las Gated Recurrent Unit (GRU) [6] y las Convolutional Neural Networks (CNN) [24] mediante las cuales obtuvieron un reconocimiento promedio del 96.7 % usando la métrica de exactitud (*Accuracy*). En [23], se utilizaron como sistema de adquisición de señales un reloj inteligente dotado con un acelerómetro, diseñaron un sistema capaz de clasificar 10 gestos o trazos continuos mediante la implementación de un modelo de red neuronal convolucional, obteniendo un reconocimiento del 97.3 %.

Como se puede observar en [8, 18, 21, 23, 36], la implementación de modelos de aprendizaje profundo se obtienen resultados de reconocimiento muy prometedores para los sistemas de reconocimiento de gestos, trazos, escritura a mano o simplemente los movimientos de la mano, haciendo uso de las señales de un acelerómetro o giroscopio.

Para el presente trabajo, se optó por realizar 12 gestos continuos utilizando el brazo, mismos que se implementaron en [12], con la diferencia de utilizar el brazalete Myo™ como sistema de captura de información sobre los movimientos de la mano. Para el procesamiento y reconocimiento de los gestos, se implementó y diseñó una CNN. Los resultados obtenidos en este trabajo de investigación son presentados en la sección 4.

3. Redes neuronales convolucionales

Las Redes Neuronales Convolucionales (CNN por sus siglas en ingles) son redes neuronales jerárquicas cuya estructura está inspirada en el sistema visual biológico [24]. La principal diferencia entre las redes neuronales clásicas o estándar y las CNN es que además de utilizar capas completamente conectadas utilizan capas convolucionales, donde se utilizan filtros de aprendizaje que se deslizan a lo largo de los datos de entrada. De forma general una arquitectura CNN puede ser descrita de la siguiente manera:

Capa Convolucional: para un sistema unidimensional, la convolución entre dos vectores $x \in \mathbb{R}^n$ y $f \in \mathbb{R}^m$ es un vector $c \in \mathbb{R}^{n-m+1}$, donde el vector f también conocido como filtro de convolución, el cual se desliza a lo largo del vector x, obteniéndose el producto escalar en cada paso y cuyos valores obtenidos forman la salida de la capa convolucional. De tal forma que $c_i = f^T x_{i:i+m-1}$, donde cada elemento de c es calculado como el producto escalar entre el vector f y un subsegmento de x.

No-linealidad: para aprender la no-linealidad de los sistemas, la capa convolucional suele estar acompañada de una función de activación no-lineal aplicada puntualmente a la salida. Comúnmente se utilizan tres funciones de activación: sigmoidal, tangente hiperbólica y ReLU (Rectified Linear Unit).

Capa de agrupamiento (*Pooling*): esta capa comúnmente sigue a una capa convolucional, cuyo objetivo es reducir la dimensión de la representación obtenida después de la convolución. Para realizar esta tarea existen dos maneras, obtener un máximo o un promedio de pequeños bloques rectangulares de los datos.

Capa completamente conectada (*Fully connected*): después de realizar el proceso en las capas convolucionales, es necesario que la salida de estas capas sea convertida en un vector el cual será usado en la clasificación. Para que una arquitectura CNN pueda aprender dependencias no-lineales, se pueden utilizar una o más capas completamente conectadas para realizar la clasificación.
Capa *Soft-Max*: Finalmente, la salida obtenida en la última capa se pasa a través de una capa *Soft-Max* que obtiene la estimación de la probabilidad de las clases. De forma general estas son las capas presentes en una arquitectura CNN, las cuales son acopladas para formar una red CNN, que puede ser entrenada como un todo.

3.1. Arquitectura implementada

En este trabajo, se ha propuesto la arquitectura CNN mostrada en la Fig. 1, donde se puede observar que las señales provienen del brazalete Myo™ el cual cuenta con una Unidad de Medición Inercial (UMI) dotada con un acelerómetro, un giroscopio y un magnetómetro. Una vez realizada la captura de las señales, estas son enviadas a la red neuronal convolucional para su posterior procesamiento y clasificación. La arquitectura de la CNN propuesta en la Fig. 1, es descrita a continuación.

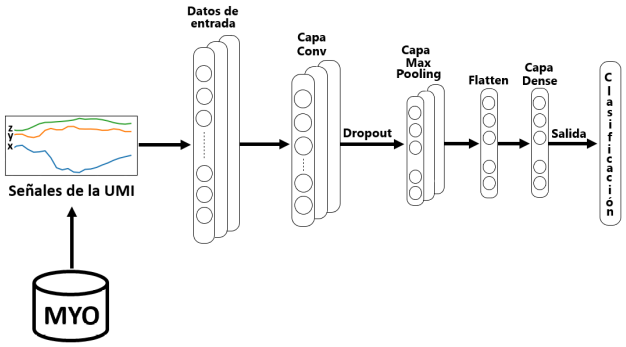


Fig. 1. Arquitectura propuesta de la Red Neuronal Convolucional (CNN).

La arquitectura mostrada en la Fig. 1 consta de una capa convolucional con 16 filtros convolucionales de tamaño 1×3 usando una función de activación ReLU para la extracción de características. Después de la capa convolucional se ha utilizado un *dropout* con una tasa de 0.9 para evitar el sobre-aprendizaje. Finalmente se ha implementado una capa completamente conectada (Dense) con función de activación *Soft-Max* para realizar la clasificación en 12 clases.

Esta arquitectura fue entrenada durante 50 épocas (*epochs*) usando el optimizador Adam y un tamaño de lote (*batch size*) de 32 y una tasa de aprendizaje de 10^{-5}.

La arquitectura presentada en este trabajo se implementó partiendo de los trabajos [18, 21, 23], sin embargo, la arquitectura final es diferente a los trabajos mencionados anteriormente. Para llevar a cabo la selección del número de capas, la función de activación, tasa de dropout, tipo de pooling, etc., así como otros hiper-parámetros se partió de los estudios [4, 9, 11, 20]. Sin embargo, el diseño final de la arquitectura se obtuvo básicamente de forma empírica, es decir, a prueba y error hasta lograr encontrar el diseño de la arquitectura que presentó el más alto rendimiento.

4. Experimentación y resultados

Para poder desarrollar y probar el rendimiento del sistema propuesto en este estudio, se ha desarrollado una base de datos lo suficientemente amplia para tener un grupo datos para entrenamiento y otro para prueba. La adquisición de las señales fue mediante el uso de una interfaz comercial llamada Myo™, dispositivo portátil desarrollado por Thalmic Labs Inc., la cual es un brazalete de un material expandible que le permite estirarse y contraerse, permitiendo así adaptarse a la fisionomía de cada usuario. Este brazalete está dotado con una Unidad de Medición Inercial (UMI) y 8 sensores para la medición de señales de electromiograma (EMG). La UMI del brazalete Myo™ consiste en un acelerómetro, un giroscopio y un magnetómetro, todos entregando señales tridimensionales. Este dispositivo puede ser conectado a un dispositivo (computadora, Tablet, celular, etc.) y cuya frecuencia de muestreo para cada sensor de la UMI es de 50Hz y de 200Hz par los sensores EMG. Este brazalete ha sido utilizado en diversos trabajos de investigación [2, 8, 14, 32, 36] obteniéndose buenos resultados, por tal motivo se optó por utilizar el brazalete Myo™.

Para el presente trabajo, se optó por realizar el reconocimiento de 12 gestos (t_1-t_{12}), los cuales se presenta en la Fig. 2. Para el análisis experimental de

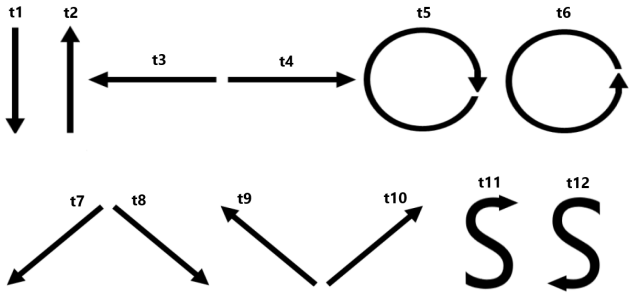


Fig. 2. Gestos planificados y capturados para formar la base de datos.

las señales, es común realizar un preprocesamiento a las mismas, que permita extraer ciertas propiedades o características propias de un conjunto de datos.

Un análisis en el espacio frecuencial es una forma común en el procesamiento de señales, ya que de esta manera se puede determinar la respuesta de un sistema para diferentes frecuencias, aunque también se puede realizar un estudio en el tiempo [15]. Algunas de las técnicas usadas comúnmente a la hora de realizar un preprocesamiento de señales son mediante el filtrado de las mismas a fin de obtener las frecuencias características del sistema [14, 31], otra técnica de preprocesamiento es usar el error cuadrático medio [36], la Transformada Rápida de Fourier, entre otras.

Sin embargo, con los avances en algoritmos de Inteligencia Artificial es posible tomar los datos en bruto, es decir, sin realizar una tarea de extracción de características previa al procesamiento de la información. En el presente trabajo se tomaron los datos entregados por el brazalete Myo™ para después ser procesados directamente en una arquitectura CNN descrita anteriormente, Fig. 1.

La base de datos cuenta con 400 trazos por cada clase (gesto), no obstante, se ha realizado un análisis tomando diferentes números de muestras para poner a prueba la arquitectura CNN propuesta y observar cuál es la susceptibilidad que tiene la red neuronal propuesta al número de muestras. Para analizar el rendimiento de la arquitectura propuesta se ha hecho una validación cruzada k-fold, la cual es una técnica estadística muy usada para evaluar el desempeño de clasificadores. Para llevar a cabo la validación cruzada (VC), se debe dividir la totalidad de los datos en k grupos del mismo tamaño. Una vez dividida la base de datos, se utiliza $k-1$ grupos para el entrenamiento del sistema y el restante para realizar el test. Este proceso se repite k veces utilizando para cada iteración un grupo distinto de test, generando en cada iteración un $Error_n$ cuyo promedio es utilizado como estimación final:

$$estimadorVC = \frac{1}{K}\sum_{n=1}^{K}(Err_n). \tag{1}$$

Para nuestro trabajo se utilizó un valor de $K = 10$, lo cual significa que, de la base de datos se tomó el 90 % de las muestras para entrenamiento y el 10 % restante es utilizado para prueba. Este valor de k es el más recomendado ya que valores más grandes pueden hacer que la varianza de la estimación aumente. La Fig. 3 muestra la curva ROC promediada para todas las clases. El área bajo la curva (AUC) es 1.0. Por lo tanto se deduce que el clasificador es capaz de separar perfectamente las clases.

Los resultados obtenidos del análisis de reconocimiento realizado se presentan en la Fig. 4, donde se puede observar el rendimiento de la arquitectura CNN propuesta para la clasificación de los 12 trazos propuestos. De este análisis se constata que a menor número de muestras se tiene un mayor rendimiento de clasificación de las señales del giroscopio con respecto a la misma cantidad de muestras usadas del acelerómetro. Sin embargo, conforme se va aumentado el número de señales el rendimiento tiende a ser el mismo.

Los resultados mostrados son referentes al promedio de la validación cruzada para los diferentes números de muestras utilizados en cada prueba realizada.

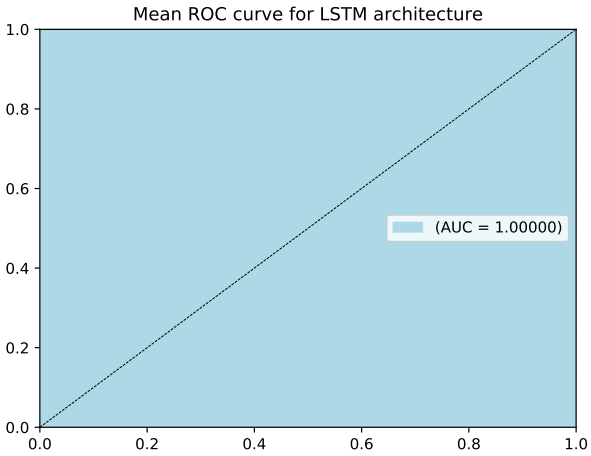


Fig. 3. Curva ROC promedio.

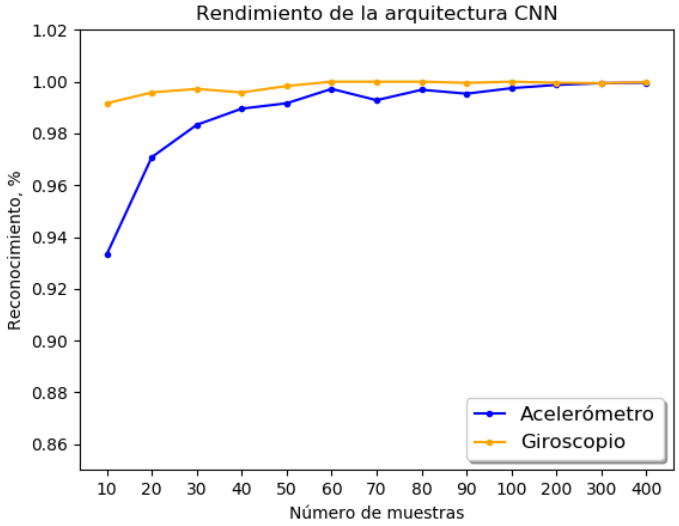


Fig. 4. Resultados de reconocimiento obtenidos para diferente número de muestras de entrenamiento.

Además, los resultados con la arquitectura propuesta han sido comparados con algunos trabajos semejantes del estado del arte, los cuales son mostrados en la Tabla 1.

Como se puede observar, los resultados obtenidos por la arquitectura CNN propuesta han logrado superar a los obtenidos en el estado del arte analizados.

Tabla 1. Análisis comparativo del porcentaje de reconocimiento en exactitud de trazos realizados con el brazo del usuario (⋆Método Propuesto).

Método	**Accuracy (%)**
CHMM [35]	99.80
CNN [23]	97.30
DTW [12]	94.00
CNN-GRU [8]	96.70
CNN⋆	99.97

Sin embargo, los resultados obtenidos en [8, 23, 35] se han obtenido únicamente para el reconocimiento de 10 trazos. Con respecto al presente trabajo se han realizado los mismos trazos en [12] obteniendo resultados superiores con el sistema propuesto.

5. Conclusiones

En la presente investigación, se ha propuesto un método para realizar el reconocimiento de trazos o gestos basándose en señales del acelerómetro y giroscopio. Los resultados obtenidos han demostrado que el uso de algoritmos de inteligencia artificial es posible obtener con gran precisión el reconocimiento de gestos. La arquitectura CNN propuesta demostró tener la capacidad suficiente para realizar el reconocimiento de los 12 trazos propuesto con un alto porcentaje de exactitud, por lo cual, se podría aseverar que las CNN pueden ser utilizadas en el procesamiento de señales obtenidas de acelerómetros y giroscopios. Sin embargo, ha pesar de que se tienen altos porcentajes de clasificación para ambos tipos de señales, el utilizar una cantidad pequeña de muestras del giroscopio basto para obtener un alto índice de clasificación, en comparación, al utilizar esa misma cantidad de muestras provenientes del acelerómetro.

Algunas ventajas que podemos resaltar del estudio realizado es la utilización de datos de sensores que se encuentran en una gran cantidad de dispositivos de uso común como los relojes o celulares inteligentes, así como la implementación de técnicas de AI las cuales pueden ser implementadas a través de múltiples plataformas de programación o de desarrollo de software, como lo es la librería de Keras en Python, la cual fue utilizada para nuestra implementación de la arquitectura CNN propuesta en este trabajo. Sin embargo, nuestro estudio está limitado a un dispositivo comercial (brazalete Myo™) pero que no es de uso común como lo son los relojes y teléfonos inteligentes.

En estos momentos, el estudio muestra resultados prometedores para el reconocimiento de gestos, sin embargo, se tiene la incertidumbre si el sistema implementado en este trabajo de investigación sea lo suficientemente robusto para ponerlo a prueba en un sistema de reconocimiento de gestos utilizado en la vida diaria de las personas, como lo son los sistemas de escritura o lenguajes de señas. Pues bien, el trabajo a futuro que se tiene planeado es llevar la

implementación de nuestro sistema actual a un sistema de reconocimiento de gestos de la vida diaria.

Referencias

1. Akl, A., Valaee, S.: Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, compressive sensing. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 2270–2273 (March 2010)
2. Ali, S., Samad, M., Mehmood, F., Ayaz, Y., Qazi, W.M., Khan, M.J., Asgher, U.: Hand gesture based control of NAO robot using myo armband, vol. 953. Springer International Publishing (020)
3. Amma, C., Georgi, M., Schultz, T.: Airwriting: A wearable handwriting recognition system. Personal and Ubiquitous Computing 18(1), 191–203 (2014)
4. Baldi, P., Sadowski, P.: Understanding Dropout. Tech. rep.
5. CHRISTOPHER J.C. BURGES: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling pp. 1–9 (2014), http://arxiv.org/abs/1412.3555
7. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
8. Dash, A., Sahu, A., Shringi, R., Gamboa, J., Afzal, M.Z., Malik, M.I., Dengel, A., Ahmed, S.: AirScript - Creating Documents in Air. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 908–913. IEEE (nov 2017)
9. Du, S.S., Wang, Y., Zhai, X., Balakrishnan, S., Salakhutdinov, R., Singh, A.: How Many Samples are Needed to Estimate a Convolutional or Recurrent Neural Network? (may 2018), http://arxiv.org/abs/1805.07883
10. Gkournelos, C., Karagiannis, P., Kousi, N., Michalos, G., Koukas, S., Makris, S.: Application of wearable devices for supporting operators in human-robot cooperative assembly tasks. Procedia CIRP 76, 177–182 (2018)
11. Gotmare, A., Keskar, N.S., Xiong, C., Socher, R.: A Closer Look at Deep Learning Heuristics: Learning rate restarts, Warmup and Distillation (oct 2018), http://arxiv.org/abs/1810.13243
12. Gupta, H.P., Chudgar, H.S., Mukherjee, S., Dutta, T., Sharma, K.: A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors. IEEE Sensors Journal 16(16), 6425–6432 (2016)
13. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, USA, 1st edn. (1994)
14. He, S., Yang, C., Wang, M., Cheng, L., Hu, Z.: Hand gesture recognition using MYO armband. Proceedings - 2017 Chinese Automation Congress, CAC 2017 2017-January, 4850–4855 (2017)
15. Herff, C., Krusienski, D.J.: Extracting Features from Time Series, pp. 85–100. Springer International Publishing, Cham (2019)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)

17. Hong, Y.J., Kim, I.J., Ahn, S.C., Kim, H.G.: Mobile health monitoring system based on activity recognition using accelerometer. Simulation Modelling Practice and Theory 18(4), 446–455 (2010)
18. Ignatov, A.: Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. Applied Soft Computing Journal 62, 915–922 (2018)
19. Ignatov, A.D., Strijov, V.V.: Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. Multimedia Tools and Applications 75(12), 7257–7270 (2016)
20. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging Weights Leads to Wider Optima and Better Generalization (mar 2018), http://arxiv.org/abs/1803.05407
21. Jiang, W., Yin, Z.: Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. Proceedings of the 23rd ACM international conference on Multimedia - MM '15 pp. 1307–1310 (2015)
22. Kai Kuspa, T.P.: Classification of Mobile Device Accelerometer Data for Unique Activity Identification. In: Semantic Scholar, Corpus ID: 1404951. pp. 1–4 (2013)
23. Kwon, M.C., Park, G., Choi, S.: Smartwatch user interface implementation using CNN-based gesture pattern recognition. Sensors (Switzerland) 18(9), 1–12 (2018)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. In: Proceedings of the IEEE. vol. 86, pp. 2278–2324 (1998)
25. Li, M., Liu, X., Liu, X.: Infrasound signal classification based on spectral entropy and support vector machine. Applied Acoustics 113, 116 – 120 (2016)
26. Li, R., View, M., Gu, A., Lin, S.h.: (12) Patent Application Publication (10) Pub. No.: US 2016/0018902 A1 (2016)
27. Ma, W., Hu, J., Liao, J., Fan, Z., Wu, J., Liu, L.: Finger gesture recognition based on 3d-accelerometer and 3d-gyroscope. In: Douligeris, C., Karagiannis, D., Apostolou, D. (eds.) Knowledge Science, Engineering and Management. pp. 406–413. Springer International Publishing, Cham (2019)
28. Mathie, M.J., Basilakis, J., Celler, B.G.: A system for monitoring posture and physical activity using accelerometers. In: 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society. vol. 4, pp. 3654–3657 vol.4 (2001)
29. Mehta, D.D., Zañartu, M., Feng, S.W., Cheyne II, H.A., Hillman, R.E.: Mobile voice health monitoring using a wearable accelerometer sensor and a smartphone platform. IEEE Transactions on Biomedical Engineering 59(11), 3090–3096 (2012)
30. Mendez, M.O., Ruini, D.D., Villantieri, O.P., Matteucci, M., Penzel, T., Cerutti, S., Bianchi, A.M.: Detection of sleep apnea from surface ecg based on features extracted by an autoregressive model. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 6105–6108 (Aug 2007)
31. Patil, S., Kim, D., Park, S., Chai, Y.: Handwriting Recognition in Free Space Using WIMU-Based Hand Motion Analysis. Journal of Sensors 2016, 1–10 (2016)
32. Rawat, S., Vats, S., Kumar, P.: Evaluating and exploring the MYO ARMBAND. Proceedings of the 5th International Conference on System Modeling and Advancement in Research Trends, SMART 2016 pp. 115–120 (2017)
33. Song, Y., Data, R.U.S.A.: (12) Patent Application Publication (10) Pub . No .: US 2014 / 0272493 A1 (2014)

34. Vu, T.H., Misra, A., Roy, Q., Wei, K.C.T., Lee, Y.: Smartwatch-based Early Gesture Detection 8 Trajectory Tracking for Interactive Gesture-Driven Applications. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2(1), 1–27 (2018)
35. Xu, S., Xue, Y.: Air-writing characters modelling and recognition on modified CHMM. 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings pp. 1510–1513 (2017)
36. Zhang, Q., Wang, D., Zhao, R., Yu, Y.: MyoSign. In: Proceedings of the 24th International Conference on Intelligent User Interfaces - IUI '19. pp. 650–660. ACM Press, New York, New York, USA (2019)

ISSN 1870-4069

Uso de *n*-gramas sintácticos en la atribución de traductor

Christian Caballero, Hiram Calvo, Ildar Batyrshin

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

{cdch10, batyr1}@gmail.com, hcalvo@cic.ipn.mx

Resumen. La atribución de traductor es la tarea de atribuir un texto traducido a su traductor. La tarea es similar a la más conocida tarea de atribución de autoría en procesamiento de lenguaje natural, pero, a diferencia de esta tarea, la atribución de traductor puede ser más difícil ya que el tema está determinado ya por el autor del texto original, lo que restringe el vocabulario a usar por el traductor. En este trabajo se presentan los resultados de representar los textos usando *n*-gramas sintácticos para atribuir los textos traducidos a su traductor usando algoritmos de aprendizaje automático.

Palabras clave: Atribución de traductor, *n*-gramas sintácticos, aprendizaje automático, estilometría.

Use of Syntactic *n*-grams in Translator Attribution

Abstract. Translator attribution is the task of attributing a text to its translator. This task is related to the similar and well-known task in natural language processing of Authorship Attribution. However, unlike the latter, translator attribution might be a more difficult task since the topic is predetermined by the original author, which limits the vocabulary to be used by the translator. This work presents the results of representing the texts using syntactic *n*-grams to attribute translated texts to their translator using machine learning algorithms.

Keywords: Translator attribution, syntactic *n*-grams, machine learning, stylometry.

1. Introducción

Atribuir un fragmento de texto a un autor (es decir, atribución de autoría) es una tarea bien establecida en el procesamiento del lenguaje natural [8,9,10,17]. Sin embargo, atribuir un documento traducido a su *traductor* (se podría llamar

a esto atribución de traductoría) es una tarea poco estudiada. Hay tan solo unos pocos artículos abordando este problema y solo un pequeño subconjunto de ellos reportan resultados exitosos.

En [1], Mona Baker propuso hace dos décadas un esquema del marco de trabajo para estudiar el estilo de un traductor o de un grupo de traductores. Primeramente, cabe mencionar que la palabra *estilo* es una palabra "cargada" en el sentido de que su significado varía con respecto a quién la usa y en qué contexto [3]. Para Baker en ese estudio (y en el presente trabajo), el *estilo* "es una cuestión de patrones: involucra el describir patrones preferidos o recurrentes de comportamiento lingüístico en lugar de instancias individuales o extraordinarias de intervención" [1]. Esta definición incluiría, por supuesto, elecciones tanto conscientes como inconscientes por parte del escritor.

Mona Baker concluye su artículo sugiriendo que, en lugar de analizar varias traducciones del mismo traductor (como ella hizo para dos traductores), tal vez se debería comparar el mismo texto fuente traducido por diferentes traductores con la finalidad de encontrar el estilo del traductor, de esa forma se controla por las características del trabajo original. Este enfoque es usado en [5] y [11] y es el seguido en este trabajo para poder comparar los resultados.

Baker también menciona en [1] que entre las opciones para buscar el estilo individual del traductor están los patrones sintácticos, dispositivos de cohesión e incluso la puntuación. Sin embargo, ella solo utiliza características más tradicionales (en el campo de la atribución de autoría) como son razón de tipo a *token* y longitud promedio de oración, y frecuencia de uso de la palabra *say*, que es el verbo para reportar más común en inglés.

A continuación se menciona el trabajo que se ha realizado en este problema. Después, se da detalle sobre el corpus usado para realizar los experimentos y los métodos empleados. En seguida, se presentan los resultados, se discuten y presentan las conclusiones y la bibliografía referida en este trabajo.

2. Trabajo relacionado

2.1. Trabajos con textos traducidos

Como se mencionó en la introducción, Mona Baker llamó la atención al problema de identificar el estilo personal de un traductor. Su equipo de investigación produjo un corpus de textos traducidos, llamado el Translational English Corpus (TEC), que contiene alrededor de diez millones de palabras de texto traducido al inglés de distintas lenguas fuente. De este corpus, ella tomó textos de dos traductores para su artículo [1]: seis textos (cinco del español y uno del portugués) de un traductor y tres del árabe del segundo traductor.

Con esos textos, comparó la variación de la proporción de tipo a *token* con un promedio móvil de 1 000 palabras, longitud promedio de oración y estructuras de diálogo indirecto (realizaciones del verbo *say*). Ella encontró diferencias entre los traductores y consistencias a lo largo de sus producciones individuales.

La conclusión que presenta en ese artículo apunta a usar corpus paralelos de traducciones para mantener fijas las variables del autor y lengua originales.

Alrededor de esa época, Mikhailov y Villika publicaron un artículo [14] en el que buscaban responder si los traductores tienen una huella estilística usando algunos de los métodos más usados en la atribución de autoría, tales como riqueza léxica, palabras más frecuentes y palabras favoritas en un corpus de textos rusos traducidos al finés. Desafortunadamente, sus resultados mostraron que los textos eran más similares a los originales que entre traducciones del mismo traductor. Es decir, traducciones de distintas obras del mismo autor son más similares entre sí que traducciones de diferentes textos por el mismo traductor. Sin embargo, Mikhailov y Villika mencionan que aún piensan que el estilo del traductor se puede encontrar pero usando otras características como palabras modales, conjunciones o formas gramaticales.

Hay más trabajos con traducciones en la última década. Hedegaard y Simonsen [7] consideraron la influencia del traductor como "ruido" en el problema de hacer la atribución de autoría en textos traducidos. Ellos utilizaron semántica de marcos, palabras más frecuentes y n-gramas de caracteres con una máquina de vectores de soporte. Concluyen que si se quisiera hacer la atribución al traductor se deberían usar los n-gramas de caracteres y las palabras más frecuentes ya que los traductores dejan una huella rastreable.

Otros trabajos, como [15,6] utilizan técnicas de agrupamiento en textos traducidos y observan que los textos se agrupan por los autores originales en lugar de hacerlo por su traductor. Ambos trabajos utilizan las palabras más frecuentes y métricas comunes en atribución de autoría como la Delta de Burrows [2].

2.2. Trabajos en estilometría del traductor

Más recientemente, en 2018 y 2019, se publicaron un par de trabajos que lidian con el problema tal como lo planteó Baker en el 2000, usando corpus de traducciones paralelas. Ambos trabajos usan técnicas de aprendizaje automático, pero difieren en las características que utilizan para representar los documentos.

Primero en [11], Lynch y Vogel abordan este problema usando n-gramas de palabras y etiquetas de categoría gramatical (*POS tags*) y también características más clásicas como longitud promedio de oración o proporción de verbos a palabras totales. Utilizan cuatro algoritmos de clasificación (máquina de vectores de soporte, Bayes ingenuo, árbol de decisión y regresión logística) en distintas combinaciones de experimentos ya que cuentan con dos corpus. Sus resultados son buenos con bigramas de palabras y también con longitud promedio de oración, aunque no estaban tan interesados en la clasificación por sí misma sino en qué características de cada traductor eran las que ayudaban más a diferenciar entre traductores.

El segundo trabajo es de El-Fiqi *et al.* [5]. Ellos usan las mismas características que Mikhailov y Villika [14], además de usar motivos de redes para entrenar una máquina de vectores de soporte y un árbol de decisión. Los motivos de redes son subgrafos recurrentes en un grafo de adyacencia de las palabras de un documento. Ellos utilizan dos corpus de traducciones paralelas y encuentran buenos resultados después de hacer selección de características y normalización a las características de los motivos de redes.

3. Métodos

En esta sección se detalla el corpus en el que se realizaron los experimentos así como los pasos para realizarlos (ver figura 1). Se desarrollaron programas en el lenguaje de programación Python[1] para los experimentos. Los programas se apoyaron en el uso de bibliotecas especializadas para manejo de lenguaje natural (spaCy[2]), algoritmos de aprendizaje automático (scikit-learn[3]) y análisis de datos (pandas[4]) para el preprocesamiento, experimentación, evaluación y reporte de resultados.

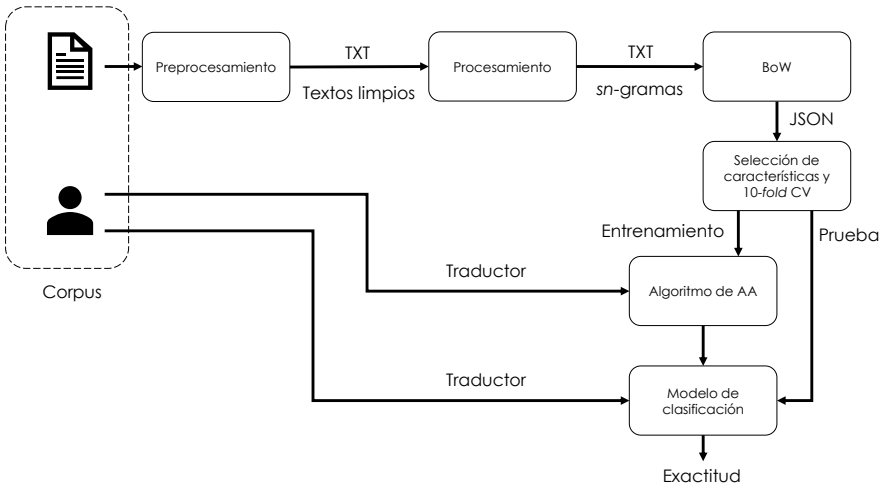


Fig. 1. Proceso para obtener la exactitud del modelo de clasificación usando validación cruzada.

3.1. Corpus

Los autores de [5] utilizaron dos corpus para sus experimentos y los hicieron disponibles en Internet. Se tomó la decisión de usar su corpus más extenso: el de *Don Quijote*. Este corpus está compuesto por tres traducciones al inglés de las dos partes de la novela de *Don Quijote*, 52 capítulos de la primera parte y 74 de la segunda. La tabla 1 muestra los detalles de los nombres de los traductores y el número de palabras mínimo y máximo por capítulo para cada traductor.

[1] https://www.python.org
[2] https://spacy.io
[3] https://scikit-learn.org
[4] https://pandas.pydata.org

Table 1. Número de palabras por capítulo del corpus de *Don Quijote*.

Traductor	Parte I	Parte II
Charles Jarvis	1 700 - 8 202	797 - 5 530
John Ormsby	1 648 - 8 304	823 - 5 680
Thomas Shelton	1 820 - 8 921	759 - 5 158

3.2. Preprocesamiento

En la etapa de preprocesamiento, se reemplazaron los caracteres especiales por sus contrapartes (por ejemplo, *ë* por *e*); se removieron los números que se encontraban entre corchetes, usados para indicar notas al pie, y se colapsaron dos o más espacios en blanco (incluyendo saltos de línea) a un solo espacio simple.

3.3. Procesamiento

El corpus provisto por [5] cuenta con los textos ya separados por capítulo y por traductor. Así que después de preprocesarlos, se analizaron usando la biblioteca especializada de procesamiento de lenguaje natural spaCy. Esta biblioteca cuenta con modelos de lenguaje basados en redes neuronales convolucionales de distintos tamaños para una variedad de lenguas que proporcionan la capacidad de hacer tokenizado, etiquetado de categoría gramatical (*POS-tagging* en inglés), análisis de dependencias sintácticas y etiquetado de entidades nombradas.

Ya analizados los textos, se generaron archivos de dependencias usando el formato de analizador de Stanford enmascarando todos los nombres propios con su etiqueta de categoría gramatical "PROPN", pasando todo a minúsculas y omitiendo las relaciones de dependencias de los signos de puntuación, para después usar un programa provisto por [16] para extraer n-gramas sintácticos. Los n-gramas sintácticos de palabras son n palabras adyacentes en el árbol sintáctico de las oraciones del texto.

Después de tener los n-gramas sintácticos con $n \in \{2, 3\}$ para cada oración, se contabilizaron las ocurrencias de cada bigrama y trigrama sintáctico para cada documento para tener dos representaciones de bolsa de palabras de los documentos. Esta información se guardó a disco a archivos JSON[5] para su fácil conversión a matrices dispersas usando scikit-learn[6].

3.4. Entrenamiento de los clasificadores

Se seleccionaron los cuatro algoritmos de clasificación usados en los trabajos del estado del arte [5,11]: regresión logística, máquina de vectores de soporte, Bayes ingenuo y árbol de decisión; así como la misma metodología de evaluación cruzada con diez iteraciones (*10-fold cross-validation*).

[5] https://www.json.org
[6] https://scikit-learn.org

Esta metodología de evaluación divide el conjunto de datos en diez subconjuntos y utiliza nueve partes para el entrenamiento de los clasificadores y la restante para su validación [13]. Este proceso se repite diez veces de tal forma que, al final, se habrán usado todos los datos tanto para entrenamiento como para validación. El resultado de la validación es el promedio aritmético (media) de alguna medida de evaluación. La métrica de evaluación seleccionada fue la exactitud, definida como el número de instancias correctamente clasificadas entre el total de las instancias [4], ya que el número de instancias entre clases está balanceado.

Para reducir el número de dimensiones (número de n-gramas sintácticos distintos) con los que se representan los documentos, se seleccionaron las 45 características más relevantes para la clasificación usando el estadístico χ^2, un método común de selección de características en el campo de la recuperación de información [12].

Se utilizaron los valores por defecto de los hiperparámetros de los cuatro clasificadores. Para la máquina de vectores de soporte se utilizó un *kernel* lineal y un término de regularización en ℓ_2 y una estrategia *uno contra todos* para k clasificadores en un problema de clasificación con k clases. Para el clasificador de Bayes ingenuo multinomial se utilizó un término de suavizado de Laplace $\alpha = 1$. El árbol de decisión utiliza la función de impureza "Gini". Por último, el clasificador de regresión logística tiene un término de regularización en ℓ_2 y una función de pérdida multinomial cuando se le pasa un problema multiclase.

4. Resultados experimentales

Los resultados de los experimentos se muestran en el cuadro 2. Los cuatro clasificadores son máquina de vectores de soporte (SVC), Bayes ingenuo multinomial (NB), árbol de decisión (DT) y regresión logística (LR). Los valores de exactitud para cada conjunto de características están en negritas.

Table 2. Resultados de la clasificación

Características	SVC	NB	DT	LR
bigramas sintácticos	96,29 %	95,77 %	85,72 %	**97,35** %
trigramas sintácticos	88,89 %	**91,01** %	76,17 %	89,68 %

En [5], los autores, que utilizaron este mismo corpus, reportan un valor de exactitud promedio de 95,10 % usando motivos de red de tamaño tres como características. Obtienen este valor después de haber hecho selección de características y empleando un árbol de decisión como algoritmo de clasificación.

5. Discusión de resultados

Los autores de [5] argumentan que esas características de motivos de redes que utilizan logran capturar información sintáctica ya que las extraen de redes de adyacencia de lemas de los textos. Nosotros usamos verdadera información sintáctica ya que los n-gramas sintácticos se obtienen a partir de dependencias sintácticas de las oraciones. Por lo tanto, nuestros resultados son comparables con los suyos.

Una ventaja de su método sobre el nuestro es que no necesitan un modelo de lenguaje para obtener las relaciones sintácticas. Sin embargo, si este tipo de método se aplica a una lengua que cuente con un modelo de lenguaje (típicamente, una lengua natural con grandes recursos) no hay mayor complicación en realizar el análisis sintáctico. La biblioteca empleada en este trabajo es bastante rápida: analiza el corpus entero–más de 400 000 palabras por traductor– en menos de 15 minutos en una computadora personal y cuenta con modelos para varias lenguas (inglés, alemán, francés, español, portugués, italiano, holandés, griego, noruego, lituano).

Por otro lado, como ya se mencionó, el usar n-gramas sintácticos captura *verdadera* información sintáctica. Además, se pueden usar variedades de estos n-gramas: con pares palabra-categoría gramatical o pura categoría gramatical, por ejemplo. El análisis de los textos solo se realiza una vez y toda la información del modelo se mantiene disponible.

6. Conclusiones y trabajo a futuro

En este trabajo se presentó el poco estudiado problema de atribución de traductor, que consiste en hacer la atribución de un texto traducido a su traductor en lugar de a su autor original. El problema está relacionado con el más conocido problema en procesamiento de lenguaje natural de atribución de autoría.

La metodología seguida es la propuesta en [1] por la profesora de Estudios de Traducción Mona Baker en el año 2000 de usar corpus paralelos de traducciones (traducciones de un mismo texto por varios traductores a la misma lengua meta).

En [5], El-Fiqi *et al.* usan esta misma metodología y prueban con técnicas de aprendizaje automático en un par de corpus de traducciones paralelas utilizando motivos de redes. Los motivos de redes son patrones recurrentes que aparecen en redes (o grafos) de adyacencia de palabras. Por la naturaleza de las características que emplearon, argumentan que sus características recogen información sintáctica.

En el presente trabajo se presentaron los resultados de usar verdadera información sintáctica por medio de n-gramas sintácticos y se probó en uno de los corpus usado en [5]. Los resultados que obtenemos con bigramas sintácticos usando selección de características con el estadístico χ^2 es mejor que el obtenido en el trabajo citado con el valor añadido de usar información sintáctica.

Como trabajo adicional y líneas a explorar en un futuro quedan probar con otro tipo de características en otros corpus paralelos y semiparalelos (traducciones de diferentes textos por los mismos traductores). Así como realizar

experimentos con otros pares de lenguas y textos de otros géneros para evaluar la sensibilidad de los métodos al género o incluso a la tipología textual.

Otra línea adicional sería probar estas técnicas para discernir entre traductores humanos y traductores automáticos o incluso entre distintos traductores automáticos.

References

1. Baker, M.: Towards a methodology for investigating the style of a literary translator. Target. International Journal of Translation Studies 12(2), 241–266 (2000)
2. Burrows, J.: 'Delta': a measure of stylistic difference and a guide to likely authorship. Literary and linguistic computing 17(3), 267–287 (2002)
3. Crystal, D.: Style: the varieties of english. A history of literature in the English language 10 (1987)
4. Eisenstein, J.: Introduction to Natural Language Processing. The MIT Press (2019)
5. El-Fiqi, H., Petraki, E., Abbass, H.A.: Network motifs for translator stylometry identification. PLOS ONE 14(2) (2019)
6. Forsyth, R.S., Lam, P.W.: Found in translation: To what extent is authorial discriminability preserved by translators? Literary and Linguistic Computing 29(2), 199–217 (2014)
7. Hedegaard, S., Simonsen, J.G.: Lost in translation: authorship attribution using frame semantics. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2. pp. 65–70. Association for Computational Linguistics (2011), https://www.aclweb.org/anthology/P11-2012
8. Holmes, D.I.: Authorship attribution. Computers and the Humanities 28(2), 87–106 (1994)
9. Juola, P.: Authorship attribution. Foundations and Trends in Information Retrieval 1(3), 233–334 (2008)
10. Koppel, M., Schler, J., Argamon, S.: Computational methods in authorship attribution. Journal of the American Society for information Science and Technology 60(1), 9–26 (2009)
11. Lynch, G., Vogel, C.: The translator's visibility: Detecting translatorial fingerprints in contemporaneous parallel translations. Computer Speech & Language 52, 79–104 (2018)
12. Manning, C.D., Raghavan, P., Schütze, H.: An introduction to information retrieval. Cambridge University Press (2008)
13. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. Massachusetts Institute of Technology (1999)
14. Mikhailov, M., Villikka, M.: Is there such a thing as a translator's style? In: Proceedings of the Corpus Linguistics. pp. 378–385. Lancaster, UK (2001)
15. Rybicki, J.: The great mystery of the (almost) invisible translator. Quantitative Methods in Corpus-Based Translation Studies: A practical guide to descriptive translation research 231 (2012)
16. Sidorov, G.: Syntactic n-grams in computational linguistics. Springer (2019)
17. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)

ISSN 1870-4069

Detección de ciberataques a través del análisis de mensajes de redes sociales: revisión del estado del arte

Jorge Enrique Coyac-Torres, Grigori Sidorov, Eleazar Aguirre-Anaya

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

B160512@sagitario.cic.ipn.mx, sidorov@cic.ipn.mx,
eaguirrea@ipn.mx

Resumen. Este artículo muestra que de la misma forma como se ha utilizado la información de redes sociales para extraer ciertos eventos específicos; por ejemplo, la aceptación de un producto en el mercado, preferencias políticas del público, eventos de salud, entre otros. Del mismo modo, existen investigaciones, que utilizan estos datos, para detectar diversos eventos o actos maliciosos que podrían poner en riesgo al ciberespacio. Por ende, este trabajo busca identificar las áreas de oportunidad que puedan existir para desarrollar futuras investigaciones de este tipo, a partir de la comparación de resultados obtenidos de cada uno de estos trabajos. Igualmente, demuestra como las herramientas de procesamiento del lenguaje natural, y los algoritmos de aprendizaje automático y profundo, han contribuido con estas tareas al ser implementadas, por cada autor, dentro de sus metodologías.

Palabras clave: Ciberataque, redes sociales, procesamiento del lenguaje natural, ciberseguridad, aprendizaje automático.

Cyberattacks Detection by Social Network Media Messages Analysis: State of the Art Review

Abstract. This paper shows that in the same way that information from social networks has been used to extract certain specific events; for example, the acceptance of a product in the market, political preferences, health events, among others. Similarly, there are investigations, which use this data to detect various malicious events or acts that could put cyberspace at risk. Therefore, this work identify the areas of opportunity that may exist for the development of future research of this type, from the comparison of results obtained by each of these works. Likewise, it demonstrates how natural language processing tools, and deep and

machine learning algorithms, have contributed to this type of task by being implemented, by each author, within their methodologies.

Keywords: Cyberattack, social networks, natural language processing , cybersecurity, machine learning.

1. Introducción

El poder de difusión de la información que tienen las redes sociales, principalmente para llegar a muchos lugares, se debe al impulso que le han otorgado las herramientas derivadas de las Tecnologías de la Información(TI). Pero si bien, cuando la tecnología evoluciona a pasos muy acelerados para brindar más facilidades y comodidades, de igual manera los riesgos y amenazas crecen con ella.

Este artículo muestra un panorama actualizado sobre aquellas investigaciones orientadas al análisis de la información contenida en mensajes de redes sociales, teniendo estas como objetivo en común, el detectar los eventos y/o actos que reflejen comportamientos maliciosos que amenacen el ciberespacio. Los resultados obtenidos, en cada una de las investigaciones, permitieron a sus autores la creación de metodologías y mecanismos que ayudaron a detectar y prevenir el riesgo de futuros ciberataques. Por lo tanto, este trabajo demuestra que las siguientes investigaciones justifican la idea de que, entre la diversidad de los temas discutidos en las redes sociales también se puede encontrar información, que después de aplicarle diversas técnicas de aprendizaje automático y de procesamiento del lenguaje natural, sirva como una herramienta de apoyo enfocada a exponer diversos eventos de ciberseguridad.

Tales investigaciones tomaron un enfoque de análisis similar a los que se han realizado en otras áreas; por ejemplo, la aceptación de algún nuevo producto en el mercado [4], el seguimiento de los eventos políticos [23], de salud [10], económicos [22], entre otros que precisamente hacen uso de la información contenida en los mensajes de las redes sociales. Pero con la diferencia de que los siguientes trabajos están orientados a eventos de ciberseguridad, como es la detección de los tipos de ciberataques, predicción de los eventos relacionados a la ejecución de un ciberataque, e incluso las posibilidades de usar una falla en software para poder ejecutar un ataque tomando ventaja de la misma.

Cabe resaltar que la mayoría de estos trabajos, no se hubieran logrado sin una metodología basada en el uso de herramientas del área de procesamiento del lenguaje natural, y apoyada en algunos casos por el uso de algoritmos de aprendizaje automático o aprendizaje profundo. Este artículo está organizado de la siguiente manera. La sección 2, presenta un análisis sobre las redes sociales y los eventos de ciberseguridad. En la sección 3, se encuentra una revisión de trabajos dirigidos a la detección de ciberataques empleando la información que es publicada en las redes sociales. Finalmente, la sección 4 es dedicada a nuestra conclusión.

2. Las redes sociales y eventos de ciberseguridad

Las redes sociales se han convertido en un gran medio masivo de difusión de la información, y actualmente han llegado a lugares donde la humanidad difícilmente se habría imaginado explorar. Diversos tópicos como política, entretenimiento, tecnología y muchos otros más son discutidos en ellas.

Toda esta diversidad de temas, ha permitido desarrollar varios trabajos dedicados al análisis de información generada por los mismos usuarios, teniendo a estos miembros participando en una forma similar a la de sensores de eventos sociales. Por lo tanto, la cantidad y calidad de la información contenida en las redes sociales depende, en gran medida, de la interacción que existe entre los integrantes de las mismas. Siendo los mensajes que se publican, referentes a los diversos sucesos que ahí se discuten, la forma de retro-alimentar esta fuente de información.

Han existido numerosos trabajos que se han basado en la extracción de información que existe en las redes sociales referentes a algún evento en específico; por ejemplo, el análisis del impacto en el mercado sobre algún nuevo producto [4], el seguimiento de alguna enfermedad dentro de una comunidad [10], el seguimiento de preferencias durante elecciones políticas [23], y varios más que le prestan atención a la plétora de eventos que puedan seguir surgiendo en una sociedad tan globalizada al día de hoy. Por ende, entre todos esos trabajos, existen otros cuya atención está dirigida al hallazgo de información del área de ciberseguridad, y posteriormente, utilizan todos esos datos para crear herramientas o mecanismos de prevención sobre futuros eventos y riesgos que atenten en contra del ciberespacio.

La preocupación ante el riesgo de ser víctimas de distintos tipos de ciberataques, y que estos puedan generar daños a infraestructuras y servicios de cómputo de usuarios, negocios, empresas e incluso instituciones de gobierno, ha generado un gran interés por parte de muchos investigadores el poder anticiparse a estos eventos. Estudios como el de Saidi [19] han expuesto la importancia de un análisis de las comunidades e información contenida de las redes sociales, concluyendo que un modelado y análisis semántico pueden ayudar a extraer información para proporcionar una visión profunda de las operaciones de grupos clandestinos de terroristas cibernéticos.

La siguiente sección muestra aquellos trabajos que se han concentrado en la detección de diferentes tipos de ciberataques. Todos ellos utilizando la información contenida en las redes sociales, y apoyándose del uso de técnicas y algoritmos del área de procesamiento del lenguaje natural, aprendizaje automático o aprendizaje profundo.

3. Detección de ciberataques en las redes sociales

Antes que nada, es importante aclarar que el análisis de la información contenida en mensajes de redes sociales, para la detección de ciberataques, no puede ser comparado con las herramientas y mecanismos comúnmente usados

en el área de ciberseguridad; por ejemplo, los antivirus, *firewalls*, Sistemas de Detección de Intrusos (o por sus siglas en inglés *IDS*), Sistemas de Prevención de Intrusos (o por sus siglas en inglés *IPS*), entre otros.

Los siguientes trabajos tienen como meta descubrir amenazas que puedan existir en el ciberespacio, pero de una forma alternativa. Por ende, estos no pretenden de alguna manera ser considerados como herramientas para mitigar los diferentes tipos de ciberataques. Más bien, sólo cumplen con la función de avisar o alertar del hallazgo de diversas formas de este tipo de amenazas.

Este artículo expone cuatro enfoques diferentes sobre la detección de ciberataques, los cuales son: **detección de eventos relacionados a ciberataques**, **detección de tipos de ciberataques**, **interés en vulnerabilidades de software para ejecutar un ciberataque**, y finalmente **ciberataques en las redes sociales**. Las siguientes subsecciones muestran trabajos relevantes que son parte de esta revisión del estado del arte.

3.1. Detección de eventos relacionados a ciberataques

Este apartado considera los trabajos enfocados en la predicción y extracción de posibles eventos que están relacionados a un tipo de ciberataque, como son el secuestro de cuentas (Hijacking), la denegación de servicio distribuido (o de sus siglas en inglés DDoS), entre otros.

Ritter [16] demostró que un gran número de eventos relacionados a fallos en ciberseguridad son mencionados en Twitter, logrando crear un extractor de esos eventos a través de un método semi-supervisado aplicado al flujo de publicaciones en esta red social. Mientras que Khandpur [11] desarrolló un método que de forma dinámica podía extraer aquellos eventos relacionados a ciberataques reportados y discutidos en Twitter, detectando un rango de tres diferentes tipos de amenazas.

Los diferentes trabajos de Hernández [9] [8] plantean que se puede predecir la respuesta de grupos específicos involucrados en Hacktivismo (acrónimo de *hacker* y activismo) cuando los sentimientos son lo suficientemente negativos hacia un usuario, ambos métodos reúnen un *corpus* de tweets, emplean análisis de sentimientos, y finalmente usan diferentes herramientas estadísticas en cada trabajo para predecir la posible aparición del ciberataque.

Pero no sólo Twitter ha demostrado ser una red social recurrente para la obtención de información. También se han analizado mensajes de otras redes sociales provenientes de foros *Hacker* que residen en la *Deep/Dark* web. Tal es el trabajo de Goyal [6], donde empleó herramientas de aprendizaje profundo (Redes neuronales profundas) y series de tiempo para predecir ciberataques con información de diversos foros Hacker.

Deliu [5] concluyó con un análisis comparativo entre Máquinas de Vectores de Soporte (o por sus siglas en inglés SVM) y Redes Neuronales Convolucionales (o por sus siglas en inglés CNN) para la detección de estos eventos dentro de foros similares a los previos. Encontrando resultados significativamente parecidos entre ambas herramientas de aprendizaje.

Finalmente, la tabla 1 muestra una comparación de los resultados obtenidos por los trabajos previamente mencionados. Aquí se observa que los algoritmos más recientes, del área de aprendizaje automático y profundo, han contribuido en la mejora de resultados, según la medida F1 en cada uno de los trabajos. Del mismo modo, se puede visualizar que entre más específico es el ciberataque a detectar, la medida F1 tiende a incrementar su valor.

Tabla 1. Comparación de resultados entre las investigaciones enfocadas a la detección de eventos relacionados a ciberataques.

Investigación	**Objetivo**	**Algoritmos**	**Medida F1**
Hernández [8]	Eventos de ciberseguridad	Regresión lineal	0.442
Goyal [6]	Malware, e-mail malicioso, URL malicioso	Redes neuronales profundas	0.653
Ritter [16]	DDoS, Hijacking, Fuga de datos	SVM + regularización L2	0.676
Khandpur [11]	DDoS, Hijacking, Fuga de datos	Convolución de kernels	0.71
Deliu [5]	Malware, Spam, DDoS	CNN , SVM	0.976

3.2. Tipos de ciberataques

Los trabajos aquí presentados están orientados a detectar y clasificar distintas formas de ciberataques que son distribuidos en las redes sociales, y que pueden engañar al usuario para que interactúe con el contenido que existe dentro de los mensajes de estas redes; por ejemplo, acceder a sitios con malware a través de una URL, ataques del tipo ingeniería social, Spam, descarga de contenido malicioso, entre otros. De tal manera que estas amenazas tienen como objetivo afectar y desestabilizar el sistema o recurso que le pertenece al usuario.

En Liao [12] presentan un framework llamado *iACE* cuya función es extraer Indicadores de Compromiso (o por sus siglas en inglés IoC [15]) a partir de textos no estructurados. El concepto IoC se define como una descripción, sobre alguna actividad y/o artefacto malicioso, que es relevante a un incidente de ciberseguridad, y que este puede ser identificado a través del análisis de sus patrones de comportamiento. Con base en lo anterior, el autor propone el uso de técnicas de procesamiento del lenguaje natural(NER/ER) para la recolección de información proveniente de fuentes públicas como blogs, artículos y otros medios escritos de acceso público, y crear una serie de términos que permitan definir características únicas de diferentes ciberataques.

Un trabajo dedicado a la detección de *Phishing* fue el de Wooi [20], donde desarrollaron un mecanismo de alerta de seguridad en tiempo real que se activa al encontrar este tipo de amenazas en los mensajes de Twitter. Dicha investigación utilizó un modelo de clasificación derivado del aprendizaje automático "Random Forest".

Shu [21] usó un modelo de regresión logística para predecir el comportamiento de un ciberataque, ante amenazas como correo malicioso, URL maliciosa y distribución de malware. Previamente, aplicaron técnicas de análisis de sentimientos, al flujo de mensajes de Twitter, para determinar la probabilidad de encontrar un ciberataque.

Madisetty [13] enunció un enfoque basado en conjuntos de redes neuronales para la detección de Spam en Twitter. El modelo empleó, a nivel de tweet, características de contenido del usuario y de n-gramas. Después, diferentes arquitecturas de CNN fueron agregadas a una red neuronal, para que finalmente pudieran obtener una detección de Spam. Consiguiendo una medida F de 0.894.

Igualmente, se han realizado investigaciones para identificar información que esté relacionada a diferentes tipos de ciberataques, tomando como fuente los foros de discusión y blogs que residen en la *Deep/Dark* web.

En Grisham [7] crearon un método para identificar mensajes que contienen información o características de distribución de aplicaciones *malware*, y que expone a los usuarios que están esparciendo estas amenazas. En su investigación hacen uso de arquitecturas de Redes Neuronales Recurrentes (o por sus siglas en inglés RNN) y análisis de las conexiones entre usuarios usando grafos. Sus resultados muestran que frecuentemente los archivos esparcidos están en formato .zip, y dirigidos a aplicaciones Android. Demostrando que, la mayoría de usuarios responsables de difundir este contenido, son aquellos que poseen alguna posición administrativa en este tipo de foros.

Finalmente, la tabla 2 muestra una comparación de los resultados obtenidos en cada uno de los trabajos anteriores. Como se puede observar, la medida F1 tiende a subir cuando los ciberataques a detectar son específicos a un tipo. Por otra parte, esta medida incrementó aún más cuando los algoritmos se apoyaron de herramientas del área del procesamiento del lenguaje natural, tal y como fue demostrado en el trabajo de Liao [12]. Pero aún está pendiente, el realizar investigaciones que consideren otros tipos de amenazas como detección de diferentes tipos de ataques de ingeniería social, detección de cuentas dedicas a la difusión de estas amenazas, entre otras.

Tabla 2. Comparación de resultados entre las investigaciones enfocadas a la detección de ciberataques que se difunden en mensajes de redes sociales.

Investigación	**Objetivo**	**Algoritmos**	**Medida F1**
Liao [12]	Indicadores de compromiso	NER/ER, SVM	0.95
Wooi [20]	Phishing	Random Forest	0.95
Madissety [13]	Spam	CNN	0.893
Grisham [7]	Distribución de malware	RNN	0.87
Shu [21]	Malware, URL malicioso, e-mail malicioso	Regresión logística + PCA	0.644

3.3. Interés en vulnerabilidades para uso de *exploits*

Los **exploits** [17] son un tipo de *malware*. Se trata de programas maliciosos que contienen datos o códigos ejecutables que se aprovechan de las vulnerabilidades del software instalado en un equipo local o remoto. Estos han llamado la atención de usuarios maliciosos para comprometer, extraer información e incluso tomar control de los equipos atacados. Dichas fallas de seguridad en el software de infraestructura de cómputo y comunicaciones, hacen que los activos de los usuarios se conviertan en víctimas de estos ciberataques.

Para este tipo de ciberataques existen un par de conceptos que se mencionan a lo largo de esta subsección. El primero es **exploit de Prueba de Concepto** (o del inglés **PoC exploit**), el cual es desarrollado como parte de un proceso para exponer e informar sobre una vulnerabilidad en un software, teniendo como objetivo el demostrar la existencia de este fallo de seguridad en el código del mismo.

El segundo concepto es **exploit de Mundo Real** (o del inglés **real-world exploit**), estos exploits son los que se utilizan para generar ciberataques a equipos de cómputo, redes e infraestructuras de comunicación de las víctimas, aprovechando las fallas de seguridad en software de los equipos para lograr un acto perjudicial al usuario.

La diferencia entre los exploits de *PoC* y de *real-world* es que, los primeros no tienen la intención de generar un daño, más bien son utilizados para alertar al proveedor de su fallo de seguridad en software, y que se tiene que crear una solución. Sin embargo, los de *real-world*, si genera un daño y muchas veces son desarrollados a partir de la información de los exploits *PoC*.

Los siguientes trabajos están enfocados a identificar información que existe en las redes sociales para la detección de exploits que pudieran representar un futuro ciberataque.

En Sabottke [18] crearon una técnica que permite detectar la existencia de posibles exploits del tipo *real-world*, a partir de información disponible en la red social Twitter. El número de características obtenidas para aplicar en un modelo de aprendizaje automático (Maquina de Vectores de Soporte) fueron 38. Entre esas características se consideró el texto y estadísticas de los tweets, un marcador llamado CVSS (de sus siglas en inglés Common Vulnerability Score System) e información de una base de datos de vulnerabilidades NVD (de sus siglas en inglés National Vulnerability Database).

En Mittal [14] crearon un *framework* llamado CyberTwitter, que tiene la función de analizar información contenida en mensajes de Twitter, para hallar las vulnerabilidades que representen una amenaza, y poder alertar al usuario de las mismas. Donde emplearon técnicas del procesamiento del lenguaje natural como NER, manejo de sentencias RDF (de sus siglas en inglés Resource Description Framework) para la representación de conceptos de ciberseguridad a través de ontologías. Creando al final su sistema de alerta con la utilización de reglas SWRL (de sus siglas en inglés Semantic Web Rule Language).

Por otro lado, Chen [3] desarrolló un método para analizar información de los mensajes de Twitter, y poder predecir cuándo las vulnerabilidades serán usadas

en un ciberataque del tipo exploit *real-world* o *PoC*. Empleó un análisis basado en un grafo llamado CVE-autor-Tweet, para descubrir el conjunto de características a utilizar en su modelo de predicción. Al final propone dos modelos, uno para la clasificación llamado FEEU, con una medida F1 promedio de 0.514. Y el otro para la regresión, llamado FRET, siendo este último el que indica cuando será ejecutado el exploit. Según sus resultados, puede predecir el uso de exploits desde que aparece la vulnerabilidad, 37.5 días antes para el exploit *PoC*, y 11.9 días para exploits *real-world*.

Un estudio que consideró información de redes sociales y foros de la *Deep/Dark* web fue realizado por Almukaynizi [2], donde abordaron la probabilidad de usar un exploit como un problema de clasificación binario; positivo si tiende a usarse o negativo en caso contrario. Emplearon un modelo de clasificación *Random Forest*, logrando resultados promedio de 0.57 para *precision*, 0.93 para *recall*, y 0.67 para una medida F1.

Por último, la tabla 3 muestra una comparación de resultados obtenidos por cada uno de los trabajos previos. Como se puede observar, el algoritmo de aprendizaje automático, *Random Forest*, mostró la mejor medida F1. Aunque Mittal [14] no presentó un valor de esta medida, y sólo reportó la cantidad de mensajes correctamente detectados dentro de un corpus de tweets. En ese trabajo, se argumenta que sus resultados fueron mejorando conforme implementaba herramientas del procesamiento del lenguaje natural.

Tabla 3. Comparación de resultados entre las investigaciones enfocadas al interés en las vulnerabilidades para su uso en exploits.

Investigación	Objetivo	Algoritmos	Medida F1
Sabottke [18]	exploits PoC, exploits real-world	SVM	0.45
Chen [3]	exploits PoC, exploits real-world	FEEU, FRET	0.514
Almukaynizi [2]	Vulnerabilidades discutidas	Random Forest	0.67
Mittal [14]	Vulnerabilidades discutidas	NER, SWRL	—

3.4. Ciberataques en las redes sociales

Este apartado hace referencia a trabajos, cuyo objetivo es la detección de ciberataques dirigidos de una cuenta a otra dentro de la misma red social. Por ejemplo, el robo o falsificación de identidad, aumento o pérdida de reputación, control de cuentas para uso en ataques tipo *Botnet*, entre otras nuevas formas de ciberataques adaptadas a estos sitios de comunicación.

En Al-Qurishi [1] crearon un sistema de predicción de ejecución de un ciberataque tipo *Sybil*, analizando información contenida en Twitter. En esta amenaza el atacante puede crear múltiples cuentas falsas o incluso instalar *malware* en el equipo de la víctima, para que posteriormente ese equipo realice *likes* o *reviews*, y el usuario víctima no tenga conocimiento de esto.

Siendo el objetivo del ataque en redes sociales, el aumentar o reducir la reputación de una cuenta de usuario. El trabajo está integrado por tres módulos, que son, la recolección de datos, mecanismo de extracción de características, y un modelo de regresión profunda haciendo uso de Tensorflow para esta tarea. Logrando un resultado del 86 %, tal y como muestra la tabla 4.

Tabla 4. Resultado de la investigación enfocada a ciberataques entre cuentas de la misma red social.

Investigación	**Objetivo**	**Algoritmos**	**Medida F1**
Al-Qurish [1]	Sybil	Modelo de regresión profunda	0.86

Como se puede observar, sólo se ha identificado un trabajo que está relacionado a la detección de un ciberataque, y cuyo objetivo es controlar o afectar la cuenta de un usuario específico desde otra cuenta de la misma red social.

4. Conclusión

Aunque existen diversas amenazas en las redes sociales como el robo de identidad, esparcimiento de noticias falsas, mensajes de ingeniería social para engañar al usuario (fraudes, extorsiones, etc.), entre otras. Este artículo presentó aquellas investigaciones que están orientadas a la detección y predicción de ciberataques que tienen la intención de alterar, controlar, manipular, dañar o afectar los servicios digitales, equipos de cómputo y comunicaciones de las víctimas. Por otra parte, los resultados obtenidos en cada uno de los trabajos mostraron que el uso de algoritmos de aprendizaje automático, aprendizaje profundo y herramientas del procesamiento del lenguaje natural, contribuyen a una mejor detección de este tipo de amenazas. Por último, entre los enfoques mostrados en secciones anteriores, se considera que hay oportunidades de investigación, principalmente en dos de ellos. El primero es para mejorar el valor F1 con relación a la predicción del uso de vulnerabilidades en exploits. Y para el segundo enfoque, se tiene que trabajar en investigaciones con relación a la detección de ciberataques entre cuentas de la misma red social. Igualmente, para futuras investigaciones se tiene que seguir considerando, dentro de las metodologías, el implementar las herramientas más recientes de aprendizaje automático y profundo y del procesamiento del lenguaje natural, con la intención de mejorar la efectividad de resultados obtenidos de las futuras soluciones.

Agradecimientos. Los autores agradecemos a el Instituto Politécnico Nacional, pero en especial a el Centro de Investigación en Computación y a CONACYT, por el apoyo recibido durante la realización de este artículo. Agradecemos a apoyo de proyectos SIP 20200859 y 20200797 y Conacyt A1-S-47854.

Referencias

1. Al-Qurishi, M., Alrubaian, M., Rahman, S.M.M., Alamri, A., Hassan, M.M.: A prediction system of sybil attack in social network using deep-regression model. Future Generation Computer Systems 87, 743–753 (2018), http://www.sciencedirect.com/science/article/pii/S0167739X17300821
2. Almukaynizi, M., Grimm, A., Nunes, E., Shakarian, J., Shakarian, P.: Predicting cyber threats through hacker social networks in darkweb and deepweb forums. In: Proceedings of the 2017 International Conference of The Computational Social Science Society of the Americas. CSS 2017, Association for Computing Machinery, New York, NY, USA (2017), https://doi.org/10.1145/3145574.3145590
3. Chen, H., Liu, R., Park, N., Subrahmanian, V.: Using twitter to predict when vulnerabilities will be exploited. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3143–3152. KDD '19, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3292500.3330742
4. Das, S., Behera, R.K., kumar, M., Rath, S.K.: Real-time sentiment analysis of twitter streaming data for stock prediction. Procedia Computer Science 132, 956–964 (2018), http://www.sciencedirect.com/science/article/pii/S1877050918308433, international Conference on Computational Intelligence and Data Science
5. Deliu, I., Leichter, C., Franke, K.: Extracting cyber threat intelligence from hacker forums: Support vector machines versus convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 3648–3656 (2017)
6. Goyal, P., Hossain, K.T., Deb, A., Tavabi, N., Bartley, N., Abeliuk, A., Ferrara, E., Lerman, K.: Discovering signals from web sources to predict cyber attacks (2018)
7. Grisham, J., Samtani, S., Patton, M., Chen, H.: Identifying mobile malware and key threat actors in online hacker forums for proactive cyber threat intelligence. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 13–18 (2017)
8. Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, K., Martinez-Hernandez, V., Perez-Meana, H., Olivares Mercado, J., Sanchez, V.: Social sentiment sensor in twitter for predicting cyber-attacks using l1 regularization. Sensors 18, 1380 (04 2018)
9. Hernández, A., Sanchez, V., Sánchez, G., Pérez, H., Olivares, J., Toscano, K., Nakano, M., Martinez, V.: Security attack prediction based on user sentiment analysis of twitter data. In: 2016 IEEE International Conference on Industrial Technology (ICIT). pp. 610–617 (2016)
10. Khan, M.A.H., Iwai, M., Sezaki, K.: A robust and scalable framework for detecting self-reported illness from twitter. In: 2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom). pp. 303–308 (2012)
11. Khandpur, R.P., Ji, T., Jan, S., Wang, G., Lu, C.T., Ramakrishnan, N.: Crowdsourcing cybersecurity: Cyber attack detection using social media. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1049–1057. CIKM '17, Association for Computing Machinery, New York, NY, USA (2017), https://doi.org/10.1145/3132847.3132866
12. Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., Beyah, R.: Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 755–766. CCS '16, Association for Computing Machinery, New York, NY, USA (2016), https://doi.org/10.1145/2976749.2978315

13. Madisetty, S., Desarkar, M.S.: A neural network-based ensemble approach for spam detection in twitter. IEEE Transactions on Computational Social Systems 5(4), 973–984 (2018)
14. Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 860–867 (2016)
15. Peréz, D.: Iocs, una palabra de moda, un tema caliente. pero, ¿realmente conocemos sus capacidades? https://www.pandasecurity.com/spain/mediacenter/seguridad/iocs-y-sus-capacidades/ (2016)
16. Ritter, A., Wright, E., Casey, W., Mitchell, T.: Weakly supervised extraction of computer security events from twitter. In: Proceedings of the 24th International Conference on World Wide Web. pp. 896–905. WWW '15, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2015), https://doi.org/10.1145/2736277.2741083
17. the Robot, M.: ¿qué son los exploits y por qué son una amenaza? https://www.kaspersky.es/blog/exploits-problem-explanation/6520/ (2015)
18. Sabottke, C., Suciu, O., Dumitraş, T.: Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In: Proceedings of the 24th USENIX Conference on Security Symposium. pp. 1041–1056. SEC'15, USENIX Association, USA (2015)
19. Saidi, F., Trabelsi, Z., Salah, K., Ghezala, H.B.: Approaches to analyze cyber terrorist communities: Survey and challenges. Computers and Security 66, 66–80 (2017), http://www.sciencedirect.com/science/article/pii/S0167404817300068
20. Seow Wooi, L., Sani, N.F.M., Abdullah, M.T., Yaakob, R., Sharum, M.: An effective security alert mechanism for real-time phishing tweet detection on twitter. Computers & Security 83 (02 2019)
21. Shu, K., Sliva, A., Sampson, J., Liu, H.: Understanding Cyber Attack Behaviors with Sentiment Information on Social Media, pp. 377–388 (06 2018)
22. Urolagin, S.: Text mining of tweet for sentiment classification and association with stock prices. In: 2017 International Conference on Computer and Applications (ICCA). pp. 384–388 (2017)
23. Younus, A., Qureshi, M., Saeed, M., Touheed, N., O'Riordan, C., Pasi, G.: Election trolling: Analyzing sentiment in tweets during Pakistan elections 2013. pp. 411–412 (04 2014)

ISSN 1870-4069

Detección de aleta caudal de ballena jorobada por medio de mapas sobresalientes basado en mediana

Rosa I. Ramos-Arredondo[1], Blanca E. Carvajal-Gámez[2,4], Francisco J. Gallegos-Funes[3], Miguel Ángel Castillo-Martinez[1]

[1] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
México
[2] Instituto Politécnico Nacional,
SEPI Escuela Superior de Cómputo,
México
[3] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
México
[4] Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas,
México

{rramosa1300@alumno.ipn.mx, becarvajal@ipn.mx,
fgallegosf@ipn.mx, mcastillom1503@alumno.ipn.mx}

Resumen. El procesamiento de imágenes es una herramienta en diferentes áreas de estudio como la visión computacional que en conjunto aporta en otras áreas como en la biología. En este trabajo proponemos la mejora de mapa sobresaliente basado en mediana para detectar la aleta caudal de la ballena jorobada en imágenes de avistamientos que fueron tomadas en altamar. En el trabajo hacemos un estudio comparativo con los métodos de Saliency Object Segmentation (SOS) y el método de mapa de atención sobresaliente de regiones escasamente salientes en una muestra de 40 imágenes. En donde el método propuesto obtiene una F-Mensure del 81%, siendo 45% mayor en compasión con los otros métodos. La métrica de intersection over union (IoU), intersección sobre la unión que sirve para medir la detección de objetos en problemas de visión computación se logra un promedio de 0.6169, obteniendo un 0.235 menos del SOS y 0.2981 del método de mapa de atención sobresaliente de regiones escasamente salientes.

Palabras clave: Procesamiento de imágenes, extracción de características, detección, ballena jorobada, identificación de animales.

Humpback Whale Tail Fin Detection Using Saliency Maps Based on Median

Abstract. Image processing is a tool in different areas of study such as computational vision that together provides other areas such as biology. In this work, we propose the improvement of the median-based saliency maps to detect

the tail fin of the humpback whale in images of sightings that were taken at sea. In the work, we do a comparative study with the saliency object segmentation (SOS) methods and the outstanding care map method of scarcely salient regions in a sample of 40 images. Where the proposed method obtains an F-Mensure of 81%, being 45% higher in compassion with the other methods. The intersection over union (IoU) metric used to measure the detection of objects in computer vision problems is achieved an average of 0.6169, obtaining 0.235 less than the SOS and 0.2981 of the outstanding care map method of scarcely salient regions.

Keywords: Image processing, feature extraction, detection, humpback whale, animal identification.

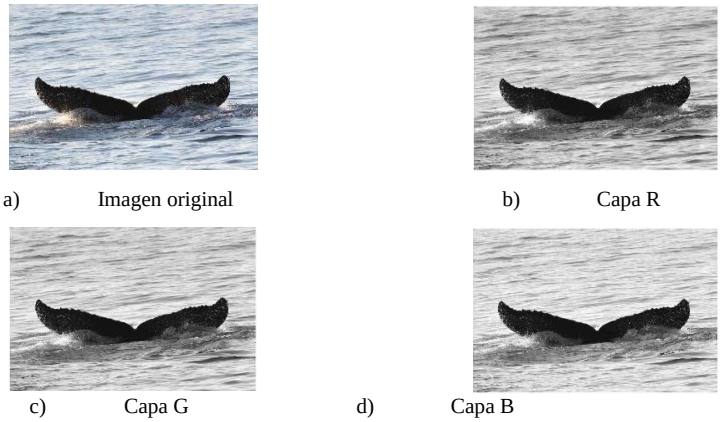
1. Introducción

El área de visión de computacional se enfoca en tareas que van en la detección de objetos, seguimiento de objetos, extracción de características, clasificación de elementos de forma automáticas, y para ellos hace uso de herramientas que en el procesamiento de imágenes se especializa en su análisis puntual [1-3]. Por ejemplo, el monitoreo de animales en su habita permite a especialista estudiar factores de comportamiento, población por mencionar y para los cetáceos que son animales de gran tamaño y su habita es inaccesibles controlar; la visión computacional y el procesamiento de imágenes apoya a su estudio [4-6]. Las imágenes que recolecta los biólogos o las personas que realizan observaciones en el mar permite obtenerlas; como es el caso de donde obtuvimos las muestras de imágenes de la ballena jorobada en el sitio de web que cuenta con una interfaz de usuarios para su recolección, análisis, compresión y divulgación de su estudio [7]. El algoritmo que proponemos hace uso de mapa sobresaliente; herramienta que se basa de la manera que la visión humana hace uso de tres detectores de sensibilidad en la retina, que da respuesta a la longitud de onda del campo visible, permitiendo al humano interpretar su entorno. El detector sobresaliente se basa en la habilidad automática de la vista humana de enfocar a elementos sobresalientes. Estos elementos pueden ser desde cambios de color, formas, texturas entre otros para obtener de una imagen (medio de entrada) la detección de elementos sobresalientes [8-11].

El presente trabajo se encuentra dividido en métodos y materiales donde se exponen la obtención de los mapas sobresalientes de los métodos a estudiar, así como el método propuesto en la detección de la aleta de la ballena jorobada. En la sección de resultados y discusión se muestra el estudio con las muestras de imágenes de la ballena jorobada en la detección de la aleta caudal por medio de mapa sobresalientes basada en mediana propuesto y la implementación de los otros métodos.

2. Métodos y materiales

En el estudio de cetáceos, el identificarlas es uno de los primeros pasos para determinar su estado de salud, comportamiento individual o en grupo, permitiendo identificar anomalías y esto a su vez un mayor conocimiento en las especies [5-6].



a) Imagen original b) Capa R

c) Capa G d) Capa B

Fig. 1. Representación de una imagen digital RGB.

La ballena jorobada es un cetáceo que entre sus características representativas se encuentra las larga aletas pectorales, las protuberancias llamados folículos pilosos en la cabeza y mandíbula y su aleta caudal. Es un animal marino acrobático por la frecuencia que se impulsa a la superficie del agua deja ver su aleta caudal y eso hace que se una de las características para su identificación [4-5]. Las imágenes para el estudio se tomaron de happywhale, cuenta con una interfaz web para que personas e investigados suban las fotografías de sus viajes de avistamientos, posteriormente un equipo de especialistas las identifique, permitiendo con ello hacer una red de conocimiento de avistamientos de animales marino [7]. Se tomaron 40 imágenes digitales RGB en plano general y/o primer plano de la ballena jorobada para implementar y realizar las pruebas del algoritmo propuesto, como con los otros métodos de mapa sobresalientes a analizar.

2.1. Imagen digital

Las imágenes obtenidas de la cámara digital son del tipo RGB o true-color; están compuesta por una matriz de pixeles y cada píxel está representado por tres capas y cada una tiene un valor escalar. Adicional a la composición de los pixeles de una imagen digital RGB se tiene la representación espacial por medio de filas y columna.

La representación de una imagen monocromática I (x, y), es un arreglo de valores escalares de dos dimensiones, x, y; son las coordenadas espaciales en donde contienen la intensidad (niveles de grises). El valor escalar que puede tomar la intensidad de una imagen es de 8bit y está dado entre 0 y 255(28). La dimensión de la imagen está representada por M, número de filas, N número de columnas y el valor de la intensidad es la función I (x, y), (1):

$$I(x,y) = \begin{Bmatrix} I(1,1) & I(1,2) & \dots & I(1,N) \\ f(2,1) & I(2,2) & \dots & I(2,N) \\ \vdots & \vdots & & \vdots \\ I(M,1) & I(M,2) & \dots & I(M,N) \end{Bmatrix}. \quad (1)$$

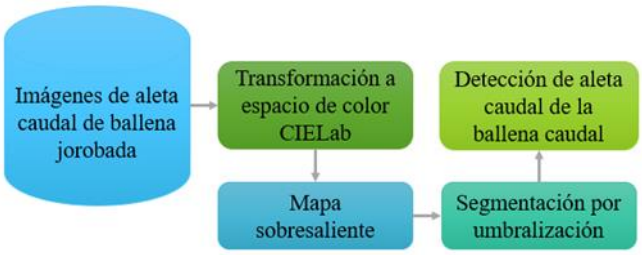


Fig. 2. Diagrama de la metodología del estudio.

Cada capa es un plano 2D que puede ser representado como una imagen monocromática; así C_i puede conformarse como $C_i = [R_i, G_i, B_i]$, en donde $0 \leq R_i, G_i, B_i \leq 255$.

2.2. Metodología de estudio

La metodología que proponemos es realizarle una transformación de espacio de color de RGB al espacio de color CIELab con un estándar de iluminación D65 (espacio de color que contempla como interpreta la visión humana). Después de la transformación del espacio de color a CIELab se realiza el estudio para la obtención de mapa sobresalientes. Estos mapas son la salida de los métodos de análisis para obtener las regiones sobresalientes y después se realiza la tarea de segmentación por umbralización. Por último, se realiza el cálculo de las métricas comparando los resultados obtenidos después de la segmentación y la segmentación manual de la base de datos.

2.3. Transformación de espacio de color CIE Lab

La dimensión de este espacio de color está compuesta por L que es la luminosidad, la componente a representa el tono del color y la componente b describen la saturación a lo largo de los ejes verde- rojo y azul-amarillo. La conversión entre CIE XYZ y CIELab está dado por los tres componentes de valores relativos y referenciado por el punto blanco D65:

$$\begin{aligned} L &= 116 . \quad Y' - 16 , \\ a &= 50 . \quad (X' - Y') , \\ b &= 0.0193R + \quad (Y' - Z') , \end{aligned}$$

$$\text{donde, } X' = f_1\left(\frac{X}{X_{ref}}\right) , \; Y' = f_1\left(\frac{Y}{Y_{ref}}\right) , Z' = f_1\left(\frac{Z}{Z_{ref}}\right), \tag{2}$$

$$f_1(T) = \begin{cases} T^{\frac{1}{3}} & para\ T > \left({}^{24}/_{116}\right)^3, \\ 7.787 . c + \dfrac{16}{116} & para\ T \leq \left({}^{24}/_{116}\right)^3. \end{cases}$$

2.4. Método para la obtención de mapas de sobresaliente de objetos - Saliency Object Segmentation (SOS)

El método para la obtención de mapas de sobresalientes de objetos en combinación con el modelo de campo aleatorio condicional (CRF). Se basa en el deslizamiento de una ventana a través de la imagen, comparando en cada ventana el contraste entre la distribución de ciertas características en la ventana interna, con la distribución en el contorno de la ventana. La ventana deslizante W está dividida por dos partes; una ventana borde B y una ventana interior K. El mapa F contiene cada punto de característica que son representadas por x. La hipótesis del detector es que los puntos sobresalientes están en K y los puntos de B son partes del fondo. Siguiendo con la hipótesis x ∈ K; W toma valores de una variable aleatoria. Los elementos sobresalientes toman valores entre 0 y 1; la medición se define por una probabilidad condicional [9]:

$$S_o(x) = P\left(Z \in K \middle| F(Z) \in Q_{F(x)}\right), \tag{3}$$

$$S_o(x) = \frac{P\left(F(x)\middle|H_0\right)P(H_0)}{P\left(F(x)\middle|H_0\right)P(H_0)+P\left(F(x)\middle|H_1\right)P(H_1)}. \tag{4}$$

La función característica F toma los valores de la imagen en el espacio de color CIELab en el punto x, L(x), a(x), b (x); se mapearía en F(x)=(L(x), a(x), b(x)). Para crear el mapa de sobresaliente se construyen tomando a diferentes escalas la ventana

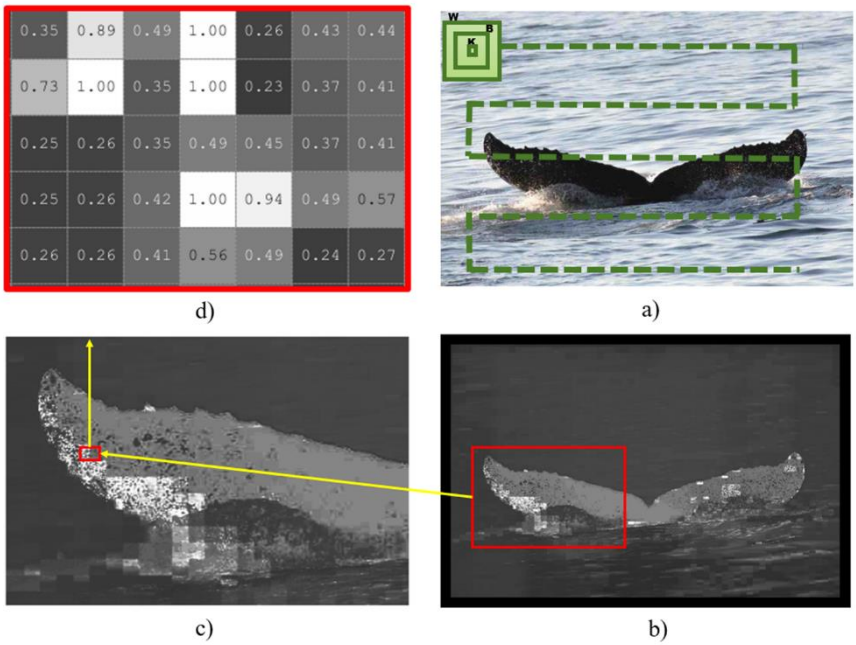


Fig. 3. Mapas sobresaliente de Saliency Object Segmentation (SOS). a) Imagen original b) Mapas sobresalientes c) Acercamiento de mapas sobresalientes d) Valores del acercamiento de mapas sobresalientes.

deslizante de W, por lo tanto, se desplaza en diferentes posiciones dentro de la imagen [8].

2.5. De mapas sobresalientes de regiones escasamente salientes

Este método considera una imagen digital como una firma compuesta por una mezcla de señales espaciales dispersas en un primer plano. El algoritmo analiza la composición de las capas y busca encontrar ciertos puntos y/o regiones de atención de la vista humana. Se propone la reducción de la imagen de estudio para obtener un acercamiento de la distancia perceptual del humano en combinación de la identificación de las regiones y/o puntos de mayor atención. Se define la función de signo de la transformada de coseno discreta (DTC) como un descriptor binario, que es la firma de la imagen en un primer plano:

$$x = f + b\,, \quad x, f, b \in \mathbb{R}^N, \tag{5}$$

donde, f es la representación del primer plano de la imagen o señal de la firma y b es la representación del fondo de la imagen:

$$\textit{ImagenSignature}(\mathrm{x}) = \mathrm{sign}\ (\mathrm{DTCx}))\ . \tag{6}$$

Para obtener el mapa de sobresaliente se suprime el ruido con el suavizado de la imagen por medio de la kernel Gaussiano la cual permite proporcionar el tamaño del objeto de interés, donde, x^i es la representación de cada capa del espacio de color de la imagen y g es el kernel de suavizado Gaussiano:

$$m = g * \sum_i \left(\bar{x}^i \circ \bar{x}^i\right). \tag{7}$$

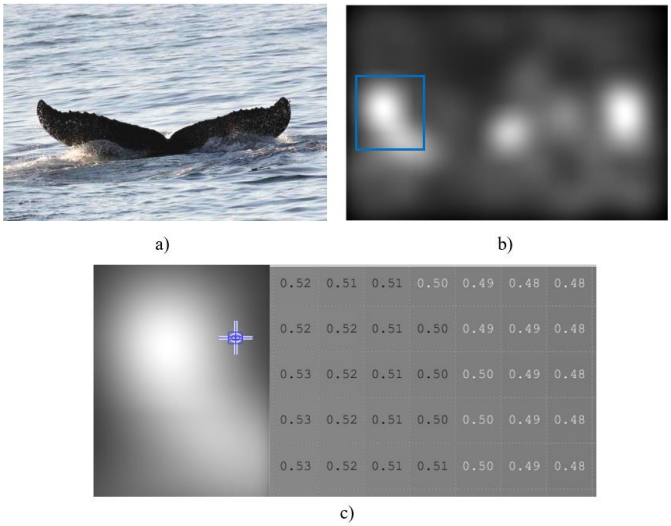


Fig. 4. Representación del método de mapas sobresalientes de regiones escasamente salientes a) Imagen digital original, b) Mapas sobresalientes, c) Acercamiento de mapas sobresalientes.

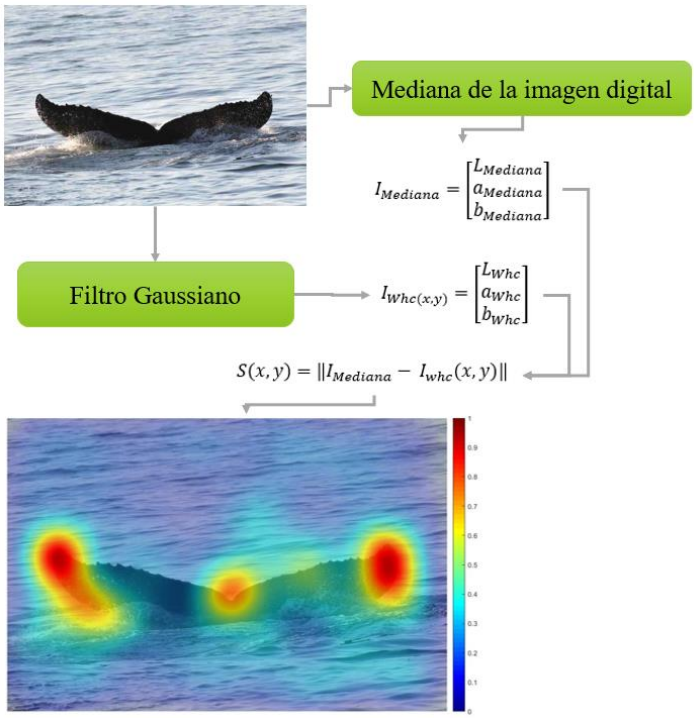


Fig. 5. Diagrama del método de mapas sobresalientes mediante mediana propuesto en una imagen digital.

2.6. Método de mapas sobresalientes basados en mediana

El método propuesto basado en mediana toma de base el método de detección de la región saliente ajustada a la frecuencia; utilizado para la detección de regiones sobresalientes que generen un mapa con limites definidos de los objetos sobresalientes en una imagen. El ajuste a la frecuencia se extiende al análisis de las características como el color y la iluminación. Permitiendo obtener la detección automática de las regiones útiles en aplicaciones y la segmentación de las imágenes [11]. El método de suavizado reduce a la mitad del espectro de frecuencia normalizado de la imagen. Además, de reducir sucesivamente las frecuencias espaciales de la imagen de entrada. Para obtener mapas sobresalientes se ocupa la transformación inversa de la transformada rápida de Fourier, usando un filtro promedio 5X5 esto es un equivalente R(k):

$$R(k) = Ln|X(k)| - g_n * In|X(k)|\,, \tag{8}$$

donde, g_n es un kernel promedio de 5x5. Reduciendo (9) en una dimensión espectral:

$$R(k) = \frac{1}{25} ln\left[\frac{|X(k)|^2}{|X(k-1)||X(k+1)|}\right]. \tag{9}$$

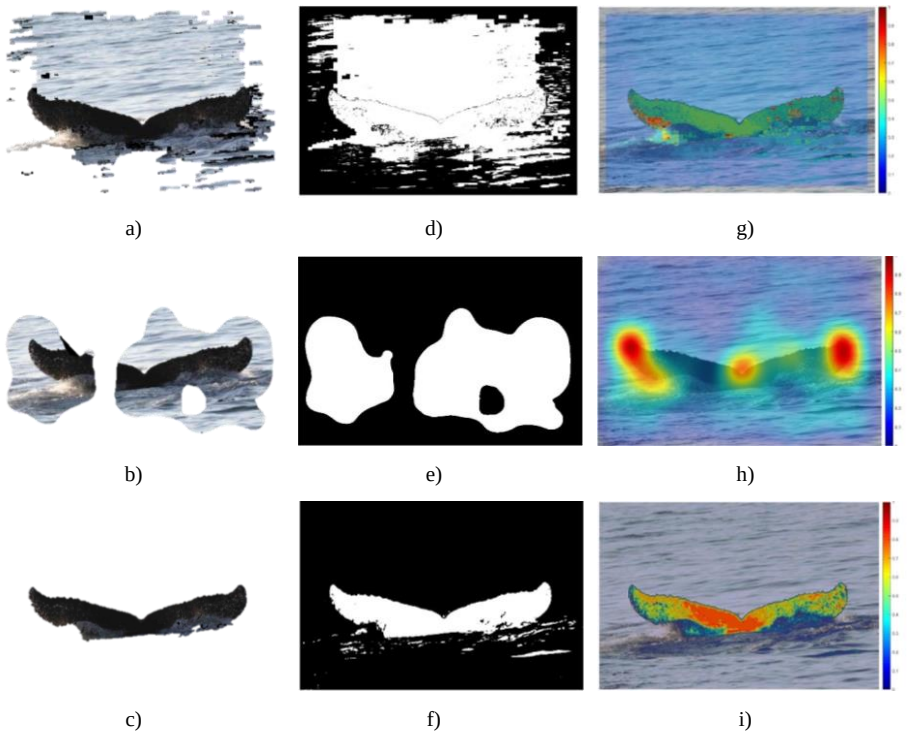


Fig. 6. Resultados de los métodos implementados en el estudio y el método propuesto a) Segmentación con el método SOS, b) Segmentación con los mapas sobresalientes de regiones escasamente salientes. c) Segmentación con mapas sobresalientes basados en mediana, d), e), f) Mascara binaria después de aplicarles la umbralización promedio global a los métodos estudiados y el propuesto en ese orden g), h) i) Mapas sobresaliente de los métodos estudiados y el propuesto en ese orden.

La diferencia de Gaussiana permite la sustracción de una versión borrosa de una imagen original de otra versión menos borrosa de la original. Es implementada para la detección de cambios de iluminación cuando la desviación estándar del Gaussiano se encuentra entre 1:1.6:

$$\sum_{n=0}^{N-1} G(x,y,\rho^{n+1}\sigma) - G(x,y,\rho^{n}\sigma) = G(x,y,\sigma\rho^{N}) - G(x,y,\sigma)\,. \tag{10}$$

Cuando σ_1 y σ_2 varia para mantener constante ρ en 1.6. Proporcionando que las regiones sobresalientes estarán cubiertas y no solo los bordes de estas mismas. Para obtener los mapas sobresalientes $S(x,y)$ de una I, imagen digital, se obtiene, $I_{Mediana}$ la mediana de los pixeles de la imagen y I_{whc} es la versión borrosa Gaussiana de la imagen original usando un kernel binomial de 5x5. Permitiendo la eliminación de detalles en la textura [12]. Al considerar las características de color e iluminación se puedes obtener el mapa sobresaliente por medio de (11):

$$S(x,y) = \|I_{Mediana} - I_{whc}(x,y)\|\,. \tag{11}$$

2.7. Segmentación por umbralización promedio global

Los mapas de sobresalientes de cada método estudiado contienen valores acotados inferior y superiormente; que representa su nivel de sobresaliente de la imagen. Para obtener una máscara que permita seleccionar la región y/u objeto de interés; es necesario definir un umbral que pueda segmentar las regiones de mayor valor de los mapas (11):

$$Umbral = \mu_{F(x)}$$
$$Mascara = \begin{cases} F(x) = 1 & x > Umbral \\ F(x) = 0 & x \leq Umbral \end{cases}, \tag{12}$$

donde, F(x) es la máscara binaria que permite separar la aleta caudal de la ballena jorobada de la imagen digital.

3. Resultados

Por medio de los métodos de sobresalientes [8-9, 12] y el método propuesto basado en la mediana se realiza un estudio para obtener una región sobresaliente la cual es comparada con la obtenida del proceso de segmentación manual de las muestras de la aleta caudal.

4. Discusión

La muestra de imágenes se le realizó una segmentación manual con el propósito de analizar el método propuesto con los otros métodos a estudiar y con ello medir la respuesta de precisión, de sensibilidad, F-Measure e intersección sobre la unión (intersection over union - IoU) que nos indique cual puede aportar comportamiento para la detección de la aleta caudal de la ballena jorobada.

La precisión se calculó con la cardinalidad de los conjuntos de la máscara de mapas sobresalientes (SF) y la máscara de la segmentación manual (GF), entre la máscara de mapas sobresalientes.

$$Precisión = \frac{|SF \cap GF|}{|SF|}. \tag{13}$$

La sensibilidad se calculó con la cardinalidad de los conjuntos de la máscara del mapa de atención prominente (SF) y la máscara binaria de la segmentación manual (GF), entre GF:

$$sensibilidad = \frac{|SF \cap GF|}{|GF|}. \tag{14}$$

La F-Measure nos permite medir la relación entre precisión y sensibilidad, indicando que tan acertado es el método segmentado.

$$F_\beta = \frac{(1+\beta^2) Precisión \ x \ Recall}{\beta^2 \ x \ Precisión + Recall}. \tag{15}$$

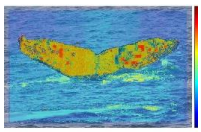
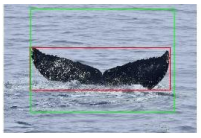
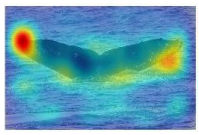
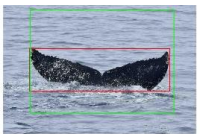
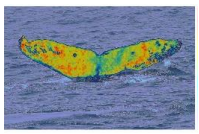
			
a) Método para la obtención de mapas de sobresaliente de objetos - Saliency Object Segmentation (SOS).			d) Imagen de prueba del documento
Precisión=0.33 Sensibilidad=0.82 F-Measure=0.42		IOU=0.31	Precisión=0.20 Sensibilidad=**0.94** F-Measure=0.27 IOU=0.20
			
b) Método de mapas sobresalientes de regiones escasamente salientes			e) Imagen de prueba del documento
Precisión=0.34 Sensibilidad=0.85 F-Measure=0.42		IOU=0.32	Precisión=0.28 Sensibilidad=0.87 F-Measure=0.36 IOU=0.27
			
c) Método de mapas sobresalientes basados en mediana(propuesto)			f) Imagen de prueba del documento
Precisión=**0.81** Sensibilidad=**0.86** F-Measure=**0.82**		IOU=**0.72**	Precisión=**0.89** Sensibilidad=0.92 F-Measure=**0.90** IOU=**0.84**

Fig. 7. Estudio de rendimiento entre dos imágenes del estudio. a) d) Método para la obtención de mapas de sobresaliente de objetos - Saliency Object Segmentation (SOS). b), e) Método de mapas sobresalientes de regiones escasamente salientes c) f) Método de mapas sobresalientes basados en mediana.

La intersección sobre la unión nos permite medir la detección de la aleta caudal de la ballena jorobada [13]:

$$IOU = \frac{|SF \cap GF|}{|SF \cup GF|}. \tag{16}$$

En la Fig. 7. se muestra las métricas comparativas de los métodos de mapas sobresalientes después de realizar la umbralización promedio global a dos imágenes. Las del lado derecho muestra de color rojo la detección de la aleta caudal por medio de la segmentación manual y en color verde la detección implementado los métodos del estudio. La columna extrema derecha es el resultado de la imagen con la que se ha

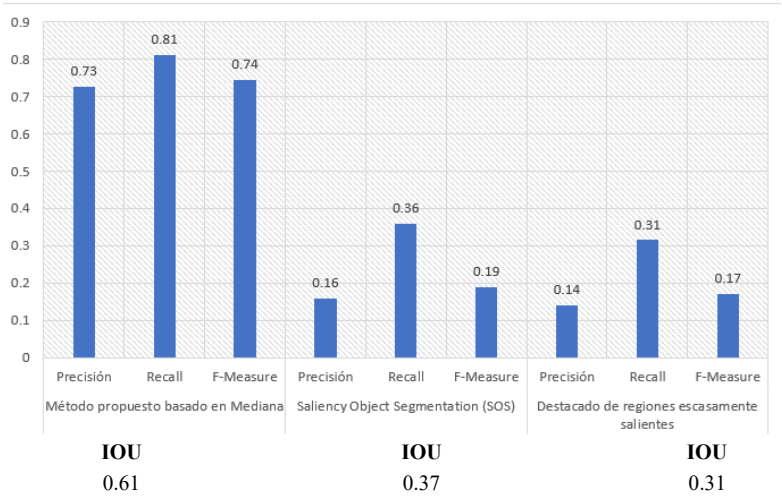


Fig. 8. Resultados de las muestras con relación de cada uno de los métodos.

desarrollado las pruebas visuales del documento por eso ya no fue necesario repetir sus implementaciones más que la detección. Por medio de las métricas los verdaderos positivos son donde SF se relaciona con GF, en caso contrario los falsos positivos del SF esta ubicado en donde no se encuentra la ballena jorobada. En la Fig. 8. se puede observar las comparativas entre los métodos y el propuesto donde el IOU es de 0.6, un valor que permite localizar a la ballena jorobada dentro de las imágenes del estudio, esto puede ser utilizado en el etiquetado de muestras para el entrenamiento de aprendizaje supervisado de redes neuronales convolucionales consume gran tiempo de personal capacitado, así como la calidad de la imagen son primordiales en el desarrollo de modelos de aprendizajes [14-15]. Los métodos comparativos del propuesto muestran un IOU bajo, esto deja que estos métodos se planteen con otro enfoque en donde obtuvieran un mayor rendimiento.

5. Conclusión

Los métodos con lo que se compara el método propuesto de mapas sobresalientes basados en mediana obtuvieron resultados menores en el estudio de 40 imágenes de la aleta caudal de la jorobada. Presentando este comportamiento en imágenes donde el mar se encuentra con una mayor uniformidad en el color y en los casos donde se tiene un mayor contraste entre los valores de las capas de color estos métodos tienen una mejor respuesta en la información que contiene los mapas de sobresaliente. En la detección de la aleta caudal se consigue obtener entre los métodos, pero el único que alcanza a pasar el 0.5 que es considerado una correcta detección es con el método propuesto. Para trabajos futuro se propone etapas en el preprocesamiento antes de obtener los mapas de sobresalientes porque la captura de las imágenes en los avistamientos no puedes ser homologadas y este factor beneficia en los métodos de mapas sobresalientes. Se queda abierto en la utilización como una herramienta en el

etiquetado de muestras que suelen ser necesario en el aprendizaje supervisado de redes neuronales.

Agradecimientos. Los autores agradecen al Instituto Politécnico Nacional de México (IPN) por la ayuda y el apoyo al proyecto de investigación con número 20200736.

Referencias

1. Petrou, M., Petrou, C.: Image processing: the fundamentals. John Wiley & Sons (2010)
2. Acharya, T., Ray, A.K.: Image processing: principles and applications. John Wiley & Sons (2005)
3. Burger, W., Burge, M.J., Burge, M.J., Burge, M.J.: Principles of digital image processing. pp. 221, London: Springer (2009)
4. Ranguelova, E., Huiskes, M., Pauwels, E.J.: Towards computer-assisted photo-identification of humpback whales. In: Image Processing, (ICIP'04), International Conference on, 3, pp. 1727–1730 (2004)
5. Urian, K., Gorgone, A., Read, A., Balmer, B., Wells, R.S., Berggren, P., Hammond, P.S.: Recommendations for photo-identification methods used in capture-recapture models with cetaceans. Marine Mammal Science, 31(1), pp. 298–321 (2015)
6. Hoyt, E., Iñiguez, M.: Estado del avistamiento de cetáceos en América Latina. In: WDCS, Chippenham, 60 (2008)
7. Cheeseman, T., Southerland, K., Burns, D., Calambokidis, J., Friedlaender, A., Gabriele, C., Harrison, P., Jones, L., Neilsen, J., Schwarz, L., Straley, J., Clapham, P.: Tired eyes no more: a fully automated, high-accuracy photo-identification matching system for humpback whales. https://happywhale.com (2020)
8. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11), pp. 1254–1259 (1998)
9. Rahtu, E., Kannala, J., Salo, M., Heikkilä, J.: Segmenting salient objects from images and videos. Computer Vision–ECCV, pp. 366–379 (2010)
10. Li, J., Gao, W.: Visual saliency computation: A machine learning perspective. LNCS 8408, Springer (2014)
11. Ko, B.C., Nam, J.Y.: Object-of-interest image segmentation based on human attention and semantic region clustering. Journal of Optical Society of America A, 23(10), pp. 2462–2470 (2006)
12. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: Computer Vision and Pattern Recognition, IEEE conference on, pp. 1597–1604) (2009)
13. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)
14. Ochoa-Ruiz, G., Angulo-Murillo, A.A., Ochoa-Zezzatti, A., Aguilar-Lobo, L.M., Vega-Fernández, J.A., Natraj, S.: An asphalt damage dataset and detection system based on retinanet for road conditions assessment. Applied Sciences 10(11), pp. 3974 (2020)
15. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: Eighth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2016)

ISSN 1870-4069

Generación de un corpus en español con expresiones agresivas

César Jesús Núñez-Prado[1], Liliana Chanona-Hernández[2], Grigori Sidorov[1]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México
[2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
México

{cesar.jnprado, lchanona}@gmail.com,
sidorov@cic.ipn.mx

Resumen. El bullying es un concepto que se refiere al acoso físico, verbal o psicológico entre una o más personas hacia un individuo o un grupo de individuos. Este tipo de agresión evolucionó al mismo paso que los avances tecnológicos y es conocido como cyberbullying. Ahora los agresores hacen uso de las redes sociales para continuar con tales ataques hacia sus víctimas. Por otra parte, las redes sociales no cuentan con filtros efectivos que sean capaces de identificar este tipo de conductas entre sus usuarios y ello complica la detección a tiempo y las consecuencias psicológicas que sufren las víctimas. Se busca crear un corpus en español que contenga las expresiones más utilizadas entre un grupo específico de personas para identificar la violencia en los mensajes publicados en Twitter.

Palabras clave: Bullying, cyberbullying, corpus, redes sociales, Twitter.

Generation of a Corpus in Spanish with Aggressive Expressions

Abstract. Bullying is a concept that refers to physical, verbal or psychological harassment between one or more people towards an individual or a group of individuals. This type of aggression evolved at the same time as technological advances and is known as cyberbullying. Now the attackers use social networks to continue such attacks on their victims. On the other hand, social networks do not have effective filters that are capable of identifying this type of behavior among their users and this complicates early detection and the psychological consequences suffered by victims. The aim is to create a corpus in Spanish that contains the most widely used expressions among a specific group of people to identify violence in messages published on Twitter.

Keywords: Bullying, corpus, social networks, twitter.

1. Introducción

La violencia no es algo nuevo, ha existido desde siempre, y en el presente las formas de materialización han ido evolucionando. Se han ideado nuevas maneras de llevarla a cabo e inclusive se han servido de los avances tecnológicos para ello. La violencia que se vive día a día en nuestro país y en el mundo entero va en incremento, ya que en todos los medios de transmisión (radio, televisión, periódicos, internet, entre otros) se ve reflejada como tema de morbosidad para atraer a un número mayor de audiencia y son los niños la parte de la población más afectada y vulnerable ya que buscan repetir las acciones que les rodean o que observan, sin detenerse a pensar si sus acciones pueden repercutir en alguien.

Con el creciente uso y desarrollo de redes sociales, una nueva modalidad de esa violencia ha aparecido y es conocida como *cyberbullying* o ciber-acoso. En ésta nueva forma, las víctimas son exhibidas de muchas maneras, tales como intimidación pública, publicación de información falsa, insultos hacia su persona, insultos hacia sus preferencias, publicación de fotos privadas, etc.

De acuerdo con [1] "El *cyberbullying* es un daño intencional y repetido a través del uso de computadoras, teléfonos celulares y otros dispositivos electrónicos" y posee ciertas características tales como: el comportamiento debe ser deliberado, la intimidación refleja un patrón de comportamiento, no solo un incidente aislado y que el objetivo debe percibir que el daño fue infligido.

Tabla 1. Diferentes tipos de acoso cibernético.

Tipos de acoso	Método
Grooming	Engatusamiento de menores por parte de pedófilos
Cebo	Uso de artimañas para atraer a niños a encuentros fuera de la red
Sexting	Enviar y pedir fotografías sexualmente explícitas
Ciber-acecho	Seguimiento de los pasos *online* de la víctima
Cyberbullying	Ofender, burlarse o agredir utilizando la lengua escrita o por medio de imágenes

Las principales consecuencias mostradas en las víctimas de este tipo de violencia son: baja autoestima, actitudes pasivas, trastornos emocionales, depresión, ansiedad, pensamientos suicidas, deseos de dejar de asistir a clases, aislamiento y en el peor de los casos el intento de suicidio u homicidio.

2. Trabajos relacionados

La aplicación en [2] se desarrolló por una joven norteamericana motivada por la noticia del suicidio de una niña menor a ella, la cual había sido acosada por sus compañeros de clase. La idea principal en este trabajo es detectar y detener el acoso en la fuente, es decir; antes de que ocurra el acoso y el daño sea irreversible. La aplicación

Tabla 2. Resultados de encuestas a estudiantes.

Porcentaje de encuestados	Tipo de interacción
99 %	Utilizan redes sociales
41%	No tienen agregados a sus padres
57 %	Sus padres no usan ningún tipo de red social
30 %	Ha sido víctima de violencia cibernética
15 %	Ha realizado *bullying* o *cyberbullying* por lo menos en una ocasión

está diseñada para el idioma inglés y al ejecutarse muestra una pantalla emergente cada vez que detecta palabras que puedan ser ofensivas y le pide al emisor que se tome un momento para pensar si realmente quiere enviar un mensaje el cual pueda herir al receptor.

En [3] se analiza el contenido de los mensajes publicados en *Twitter* y se determina si el mensaje contiene acoso o no, mediante el uso de algoritmos de aprendizaje automático, n-gramas, errores gramaticales y está diseñado para el idioma español.

3. Sujeto de estudio

Gracias a las autorizaciones por parte del Coordinador de Área de la Secretaría de Educación Pública (SEP) y del director de una secundaria técnica en la Ciudad de México, se pudo trabajar con una población mixta de 900 alumnos. El objetivo principal de involucrar a alumnos en ese rango de edad (entre 12 y 15 años) es la de conocer las interacciones de los estudiantes con las redes sociales, la manera en que se expresan y las redes sociales que utilizan.

El 41 % que afirmó, que en sus cuentas en redes sociales no tienen agregados a sus padres ni a ningún familiar, lo hacen debido a que se sienten hostigados y vigilados. Esta encuesta revela que un gran porcentaje de los padres de familia no tienen conocimiento de las interacciones de sus hijos dentro de las redes sociales, lo cual conlleva como consecuencia, al desconocimiento de si su hijo es posible víctima o victimario de acoso cibernético.

Dentro de ésta población, un 30 % aceptó haber sido víctima de *cyberbullying* por lo menos en una ocasión y un 15 % confirmó haber realizado por lo menos una vez *cyberbullying*.

Al ser menores de edad los alumnos de la secundaria con los cuales se trabajó, se pidió autorización a los padres de familia. En estas pláticas los padres comentaron que se han presentado algunos casos de acoso dentro de la comunidad estudiantil pero que no han podido terminar con el hostigamiento hacia las víctimas.

4. Generación de diccionario con malas palabras

A partir de las encuestas realizadas a los estudiantes, se pudo crear un diccionario con un total de 115 malas palabras consideradas a partir de las expresiones que utilizan

Tabla 3. Muestra del diccionario.

alv	estupido	putito
culo	fuck	homosexual
gorda	trans	pendejo
puta	culero	huevos
joto	mamon	wey
verga	naco	guey

con mayor frecuencia en sus interacciones en las redes sociales. Las palabras que se incluyeron aparecieron con al menos un 50 % en las encuestas y se agregaron inclusive cuando las palabras están mal escritas (con faltas de ortografía), ya que los encuestados refirieron que escribir correctamente les quita mucho tiempo.

Estas palabras se agregaron al diccionario tanto en femenino como en masculino, en singular y en plural, con abreviaturas e incluso con faltas ortográficas ya que los encuestados refirieron que escribir correctamente les quita demasiado tiempo.

5. Sistema de extracción de *tweets*

Para la descarga de mensajes de la plataforma *Twitter* se aperturó una aplicación de desarrollo interna (disponible para programadores), la cual proporciona los permisos y las claves necesarias para realizar una correcta recuperación de información dentro de la misma red social. El sistema se desarrolló en el lenguaje de programación *Python* y para realizar la conexión con *Twitter* se utilizó la librería *Tweepy*. Para la búsqueda y descarga de los *tweets* se utilizaron las palabras del diccionario creado a partir de las encuestas realizadas. Se extrajeron cerca de 2000 mensajes a los cuales se les realizó un filtrado para eliminar los mensajes repetidos y los mensajes escritos en un idioma diferente al español. Una vez que se contó con mensajes únicos (1300 mensajes no repetidos y en español) se realizó una fase de procesamiento de lenguaje natural utilizando librerías de *Natural Language ToolKit* (NLTK) y librerías internas de *Python*. Esta fase incluyó:

- Conversión del texto a minúsculas,
- Eliminación de enlaces en internet,
- *Tokenización*,
- Eliminación de caracteres especiales y signos de puntuación,
- Eliminación de *stickers* y
- Eliminación de palabras de parada (*stop words*).

6. Evaluación y generación del corpus

El archivo con los mensajes descargados de *Twitter* y ya procesados se envió a un jurado conformado por 5 personas. El jurado es mixto, es decir; con hombres y mujeres

con un rango de edad entre los 20 y los 45 años y que tienen contacto diario con redes sociales. El jurado se encargó de calificar los mensajes con solo dos opciones: "con acoso" o "sin acoso". La medida que se utilizó para considerar si el mensaje se agregaba al corpus, fue que al menos 3 miembros del jurado calificaran de acoso al mismo mensaje. El corpus final quedó compuesto por 600 mensajes en español con acoso cibernético.

7. Conclusiones y trabajo a futuro

El manejo de las redes sociales en la actualidad es casi inevitable y no importa si el usuario es adulto mayor, estudiante de primaria, alto funcionario, hombre o mujer. En algunas ocasiones, las vacantes para empleos requieren que se dominen las redes sociales y el manejo de dichas redes puede traer consigo ventajas y desventajas.

Fue a partir de las desventajas, que se pensó en contribuir con la creación de un corpus que sirva como herramienta para la detección de este tipo de violencia cibernética y con ello ayudar a las personas más vulnerables.

De las encuestas realizadas a los alumnos se concluye que uno de los principales factores que propician el ser víctimas de acoso cibernético, radica en que no se cuenta con la información necesaria para realizar una denuncia y con ello limitar en cierta proporción el ciber-acoso sufrido en las redes sociales.

El corpus creado refleja de manera real, el uso de la lengua empleado por estudiantes de nivel secundaria.

Como trabajo a futuro se busca que el corpus creado pueda crecer en dimensión para que pueda contener más expresiones agresivas utilizadas y el error al detectar el ciber-acoso se pueda reducir.

También se piensa en realizar un sistema que haga uso de inteligencia artificial para lanzar una alerta al usuario que sea una posible víctima de violencia en redes sociales.

Agradecimientos. Agradecemos a CONACYT, SNI, IPN (SIP, COFAA), apoyo de proyectos SIP 20200859 y 20200797 y Conacyt A1-S-47854.

Referencias

1. Ruiz-Palacios, F.: Corea capacita a SSP contra ciberdelitos. El Universal, http://www.eluniversal.com.mx/articulo/metropoli/cdmx/2016/11/2/corea-capacita-ssp-contra-ciberdelitos (2016)
2. Rethink Words: ReThink before the damage is done. Rethinkwords, http://www.rethinkwords.com (2016)
3. Ramos-Márquez, J..: Detección de Acoso en mensajes de Twitter. Centro de Investigación en Computación, Tesis de Maestría (2017)
4. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional, n-gramas sintácticos, filtrados y generalizados (2013)
5. Sanz, E.: ¿En qué consiste el "sexting"? Muy interesante. http://www.muyinteresante.es/curiosidades/preguntas-respuestas/ien-que-consiste-el-sexting (2016).
6. Russell, M.A.: Mining the Social Web, Data Mining Facebook, Twitter, Linkedin, Google+, Github, and More. http://www.webpages.uidaho.edu/~stevel/504/Mining-the-Social-Web-2nd-Edition.pdf (2013)

7. Bird, S., Klein, E., Loper, E.: Natural language processing with python, analyzing text with the natural language toolkit. O'Relly Media (2010.

ISSN 1870-4069

Evaluación de cuatro clasificadores para el reconocimiento de síndrome de muerte súbita del lactante utilizando ácidos grasos de cadena corta como fuente de información

Karen E. Villagrana-Bañuelos[1], Laura A. Zanella-Calzada[2], Carlos E. Galván-Tejada[1], Hamurabi Gamboa-Rosales[1], Jorge I. Galván-Tejada[1]

[1] Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, México
[2] LORIA (INRIA, CNRS, Université de Lorraine), France

{kvillagrana;ericgalvan;hamurabigr;gatejo}@uaz.edu.mx, laura.zanella-calzada@univ-lorraine.fr

Resumen. La inteligencia artificial y su aplicación en el área médica, ha venido a coadyuvar al diagnóstico de diversas enfermedades. La manipulación, procesamiento, análisis, e interpretación de grandes y complejos datos, ha permitido, identificar relaciones, que en muchas ocasiones, permanecen imperceptibles para la mente humana. El desarrollo del aprendizaje automático, permite, diseñar modelos, que ayuden a predecir enfermedades; algunas de ellas, se desconoce aún con los avances científicos, la razón que las genera, por lo que encontrar patrones, permite inferir sobre dichas enfermedades, para prevención y diagnóstico. El síndrome de muerte súbita del lactante (SMSL), es la muerte de un menor de un año de edad, sin causa aparente. Una de la posibles explicaciones al SMSL son los errores innatos del metabolismo (EIM), especialmente los defectos en la beta-oxidación de los ácidos grasos, por lo que este estudio se centra en analizar la relación de los ácidos grasos de cadena corta (AGCC) con casos identificados de muerte por SMSL contra controles, evaluando y comparando el rendimiento de los clasificadores k Nearest Neighbors, Random Forest, Naïve Bayes y Regresión Logística, para incrementar las pruebas de que existe una relación de los procesos metabólicos con este síndrome. Los modelos de Random Forest y de Naïves Bayes obtienen un desempeño igual para esta muestra de sujetos de SMSL, con un área bajo la curva de 0.75 y una exactitud de 0.8889 para ambos clasificadores.

Palabras clave: Síndrome de muerte súbita del lactante, ácidos grasos de cadena corta, k vecinos más cercanos, bosque aleatorio, Naïve Bayes, regresión logística.

Evaluation of Four Classifiers for the Recognition of Sudden Infant Death Syndrome Using Short Chain Fatty Acids as a Source of Information

Abstract. Artificial intelligence and its application in the medical area, has come to contribute to the diagnosis of various diseases. The manipulation, processing, analysis, and interpretation of large and complex data has allowed us to identify relationships that, on many occasions, remain imperceptible to the human mind. The development of machine learning allows to design models that help to predict diseases; some of them, even with scientific advances, the reason that generates them is unknown, so finding patterns allows us to infer about these diseases, for prevention and diagnosis. Sudden infant death syndrome (SIDS) is the death of a child under one year of age, with no apparent cause. One of the possible explanations for SIDS, is the inborn errors of metabolism (IEM), especially defects in fatty acid beta-oxidation, so this study focuses on analyzing the relationship of short chain fatty acids (SCFAs) in cases identified from death by SIDS versus control cases, evaluating and comparing the performance of k Nearest Neighbors (kNN), Random Forest (RF), Naïve Bayes and Logical Regression to increase the evidence that there is a relationship of metabolic processes with this syndrome. The Random Forest and Naïves Bayes models obtained the same performance for this sample of SIDS subjects, with an area under the curve of 0.75 and an accuracy of 0.8889 for both classifiers.

Keywords: Sudden infant death syndrome, short chain fatty acids, k nearest neighbors, random forest, Naïve Bayes, logical regression.

1. Introducción

Por décadas la inteligencia artificial, ha existido, pero su relación con la medicina se vio retrasada debido a cuestiones éticas y culturales [18]. A pesar de ello el aprendizaje automático, por medio de los modelos predictivos, ha cobrado gran relevancia en las ciencias médicas, estudiando diferentes enfermedades multifactoriales [1, 10, 23], especialmente en el área pediátrica [16], donde ha permitido incursionar en la asistencia diagnóstica tal como el trabajo de Homer Warnerm, que en 1961 desarrolló un sistema de diagnóstico automatizado para diagnosticar correctamente a pacientes con cardiopatías congénitas a partir de cincuenta variables clínicas, obtuvo como resultado, que el sistema fue más certero, comparado con los diagnósticos de los cardiólogos [18], otro ejemplo más reciente es el trabajo de Halabi et al [13], quienes trabajaron con técnicas de aprendizaje automático en la identificación de la edad ósea pediátrica; y así múltiples investigaciones han consolidado la pertenencia del uso de técnicas de aprendizaje automático en el área médica [24], buscando proporcionar un instrumento de apoyo.

Existen múltiples enfermedades que en nuestro siglo, siguen siendo un reto para la comunidad científica, ya que no se ha logrado esclarecer consistentemente

el origen de las mismas. Resultan de interés, ya que la inteligencia artificial permite manipular grandes y complejas cantidades de datos, encontrando relaciones que ha simple vista, parecen inexistentes. Así patologías, de las cuales aún no se ha descubierto la causa certera, podrían beneficiarse, de la inteligencia artificial, para lograr identificar patrones, que no se tenían contemplados, y así inferir con el objetivo de prevenirlas o tratarlas.

El síndrome de muerte súbita del lactante (SMSL) es definido como la muerte de un menor de un año de edad, que permanece sin explicación después de una autopsia y de una investigación exhaustiva de la escena de muerte [14]. A pesar de los múltiples esfuerzos de investigaciones en área médica, las explicaciones posibles abarcan una amplia rama de teorías sobre el origen de éste síndrome, dentro de las cuales destacan los defectos en el corazón [7, 19, 26, 27], teorías del triple riesgo [11, 15], pérdida del automatismo cardiorrespiratorio [2, 3], los errores innatos del metabolismo [17], entre otros.

Atendiendo, a la teoría patogénica que los errores innatos del metabolismo (EIM) pudieran explicar las muertes por SMSL; entenderemos por EIM a un grupo de enfermedades causadas por una mutación genética que tiene como efecto la producción de una proteína anómala que conlleva la alteración del funcionamiento fisiológico de la célula; según sea la función alterada, los efectos fisiopatológicos del acumulo de sustancias no metabolizadas dependen del grado de acumulación y de su posible toxicidad [6], de ahí la gran variedad de presentaciones clínicas, y la dificultad en el diagnóstico si no se busca de manera intencionada. Los trastornos del ciclo de beta-oxidación de ácidos grasos, forman parte de los EIM. La enzima acetil coenzima A, es generada a partir de ciclos repetidos de oxidación, necesitando de grupos de cuatro enzimas específicas para cada una de las longitudes de cadena de los ácidos de cadena muy larga, larga, media y corta. Se han estudiado los defectos de oxidación de los ácidos de cadena media, sin embargo, Opdal et al [20] destaca la necesidad de investigar otros tipos de defectos como los EIM, en especial los defectos del metabolismo de los ácidos grasos de cadena corta (AGCC) [5]. Siendo de especial interés identificar, si existe relación entre el SMSL y los niveles de ácidos grasos de cadena corta, lo cual permitiría tener marcadores para pronosticar riesgo de presentar el síndrome.

Debido a lo anterior, el objetivo de este estudio es evaluar y comparar el rendimiento de los clasificadores k Nearest Neighbors (kNN), Random Forest(RF), Naïve Bayes(NB) y Regresión Lógistica(RL) en el reconocimiento de SMSL utilizando AGCC.

2. Materiales y métodos

En la presente sección se describen los datos utilizados para el desarrollo de los modelos de clasificación, al igual que una breve descripción de los mismos. También, las métricas utilizadas para la validación y la comparación de los modelos son mencionadas.

2.1. Descripción de la base de datos

Para el desarrollo de este trabajo fue utilizada la base de datos, '$SCFA\ profile\ in\ babies\ dying\ from\ SIDS$', la cual se encuentra públicamente disponible en `https://www.metabolomicsworkbench.org` proporcionada por el $NIH\ Common\ Fund's\ National\ Metabolomics\ Data\ Repository$, con el número de proyecto asignado PR000512.

La información contenida en esta base de datos se refiere a 18 sujetos, cinco controles y 13 pacientes diagnosticados con SMSL. Esta información incluye los valores obtenidos para las características presentadas en la Tabla 1, los cuales fueron calculados post-mortem a través de la extracción fría medida con ionización de electrones - cromatografía de gases espectrometría de masas sin derivación.

Tabla 1. Características contenidas en la base de datos de perfil de AGCC, en lactantes fallecidos por SMSL.

	Características
1	Horas de intervalo post-mortem
2	Semanas de edad gestacional
3	Semanas de edad post-natal
4	Ácido isovalérico
5	Ácido octanoico
6	Ácido propiónico
7	Ácido isobutírico
8	Ácido butírico
9	Ácido hexanoico
10	Ácido valérico
11	Ácido acético

2.2. Modelos de clasificación

En esta sección se describen brevemente los modelos de clasificación que han sido utilizados para la identificación de lactantes que presentaron síndrome de muerte súbita. Estos clasificadores fueron seleccionados debido a sus diferentes características, así como por ser utilizados en el campo de la biomédica para diferentes enfermedades, ya sea para dar pronósticos o diagnóstico de las mismas [9, 24, 28].

Random Forest

Random forest (RF) o bosques aleatorios, se refiere a un conjunto de clasificadores que aprenden de manera aleatoria múltiples árboles de decisión. Este método consiste en dos etapas principales, entrenamiento y prueba. En la etapa de entrenamiento se construyen los múltiples árboles de decisión, mientras que

en la etapa de prueba se lleva a cabo la clasificación o la predicción de la variable de salida con base en las variables de entrada.

La distribución que se obtiene de los árboles de decisión de cada bosque es inicialmente promediada para después llevar a cabo la clasificación. Los predictores de cada muestra son combinados usando la media para el caso donde las variables son continuas y el voto mayoritario para el caso donde las variables son categóricas.

Una característica de esta técnica es que no presenta problemas significativos de sobreajuste, y no se ve afectada por el ruido o los valores atípicos que pueden estar presentes en los datos. Además, puede reducir los errores de generalización, permitiendo lograr un valor alto de precisión [4].

Regresión Logística

Los modelos de regresión logística (RL) presentan como requisito que la variable de salida sea categórica, y calculan generalmente la probabilidad, en un rango de 0 a 1, de que una clase pertenezca a alguna de las categorías del conjunto de datos.

La probabilidad estimada que se obtiene como salida es usada para inferir qué tan confiable puede ser el valor que se predijo en comparación con el valor real, dado un valor de entrada.

Existen diferentes tipos de RL, tales como regresión logística binaria, multinominal y ordinal, las cuales varían dependiendo del número de categorías que presente la variable de salida [8].

***K*-Nearest Neighbor**

K-NN o K-vecinos cercanos, es una técnica regularmente utilizada cuando se tiene poco o nulo conocimiento sobre la distribución de los datos.

K-NN consiste en minimizar la distancia Euclidiana de la muestras de entrada con respecto a las características, buscando de esa manera que cada muestra se categorice con sus vecino más cercanos.

Por lo tanto, esta técnica se basa en una regla de clasificación en donde se asigna a cada muestra la etiqueta de la categoría mayoritaria de sus K muestras más cercanas. Regularmente K es asignado con un número impar, para evitar de esta manera lazos [21].

Naïve Bayes

Naïve Bayes (NB) se refiere a una técnica de clasificación basada en probabilidad, en donde se calcula el conjunto de probabilidades a través de contar la frecuencia y las combinaciones de valores en un conjunto de datos.

Esta técnica asume que todas las variables son independientes considerando el valor de la variable dependiente o de salida. Esta suposición de independencia condicional es raramente aceptada en aplicaciones reales, por lo que se le categoriza como 'ingenua', pero este algoritmo tiende a aprender rápidamente en problemas de clasificación controlados.

Por lo tanto, el teorema Bayes es usado para determinar la probabilidad condicional, en donde se calcula la probabilidad de la ocurrencia de un evento A, de un evento B, de un evento A cuando un evento B ocurre, y de un evento B cuando un evento A ocurre. Con base en esto, se obtiene la probabilidad de cada muestra de corresponder a alguna de las dos categorías posibles [25].

2.3. Validación

Con el fin de evaluar el comportamiento de los modelos desarrollados se llevó a cabo una etapa de validación, con base en tres métricas principales, la curva ROC (acrónimo de receiver operating characteristic), al área bajo la curva (AUC, por sus siglas en inglés) y la exactitud.

Para el cálculo de estas métricas es necesario conocer el número de verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN). Con base en esto, es posible obtener la sensibilidad y la especificidad. La sensibilidad calcula con la Ecuación 1, la cual permite conocer la probabilidad de que un caso de positivo SMSL sea clasificado correctamente:

$$Sensibilidad \ = \ \frac{VP}{VP + FN}. \tag{1}$$

La especificidad se calcula con la Ecuación 2, la cual permite conocer la probabilidad de que un caso negativo de SMSL sea clasificado correctamente:

$$Especificidad \ = \ \frac{VN}{VN + FP}. \tag{2}$$

A través de estas dos métricas es posible obtener la curva ROC, la cual permite visualizar el comportamiento de los clasificadores, en conjunto con el AUC. El AUC representa la probabilidad que una muestra positiva aleatoria sea identificada de manera correcta [12].

Por otro lado, la medición de la exactitud se refiere a la tasa del número de predicciones correctas con respecto al total de predicciones hechas, y se puede calcular con la Ecuación 3:

$$Exactitud \ = \ \frac{predicciones\ correctas}{total\ de\ predicciones}. \tag{3}$$

3. Experimentos y resultados

En esta sección se describen los experimentos y los resultados obtenidos a través de la metodología seguida.

Para este trabajo fue utilizado un conjunto de datos contenido por once características que describen los valores presentados por pacientes con SMSL positivo y SMSL negativo. Estos datos fueron modelados a través de cuatro clasificadores: RF, RL, K-NN y NB, con el fin de identificar automáticamente la clase a la que pertenecía cada paciente. Para el entrenamiento de los modelos se

tomó el 70 % de los datos y el 30 % restante se utilizó para prueba. Una vez que los modelos fueron desarrollados, se sometieron a una evaluación a través de una validación estadística, para poder conocer el grado de generalización alcanzado.

En la Tabla 2 se presentan los resultados obtenidos de la validación de los modelos, en donde se pueden observar los valores de exactitud y AUC calculados para cada clasificador.

Tabla 2. Resultados de la validación de los modelos obtenidos.

Clasificador	**Exactitud**	**AUC**
RF	0.8889	0.7500
RL	0.6667	0.6071
K-NN	0.7777	0.6786
NB	0.8889	0.7500

Por otro lado, en la Figura 1 se muestran las curvas ROC obtenidas para cada modelo de clasificación, dónde se puede observar además su comportamiento en términos de sensibilidad y especificidad, permitiendo conocer también el desempeño de cada clasificador para reconocer los verdaderos positivos y verdaderos negativos, siendo dos métricas comúnmente utilizadas para evaluar pruebas médicas.

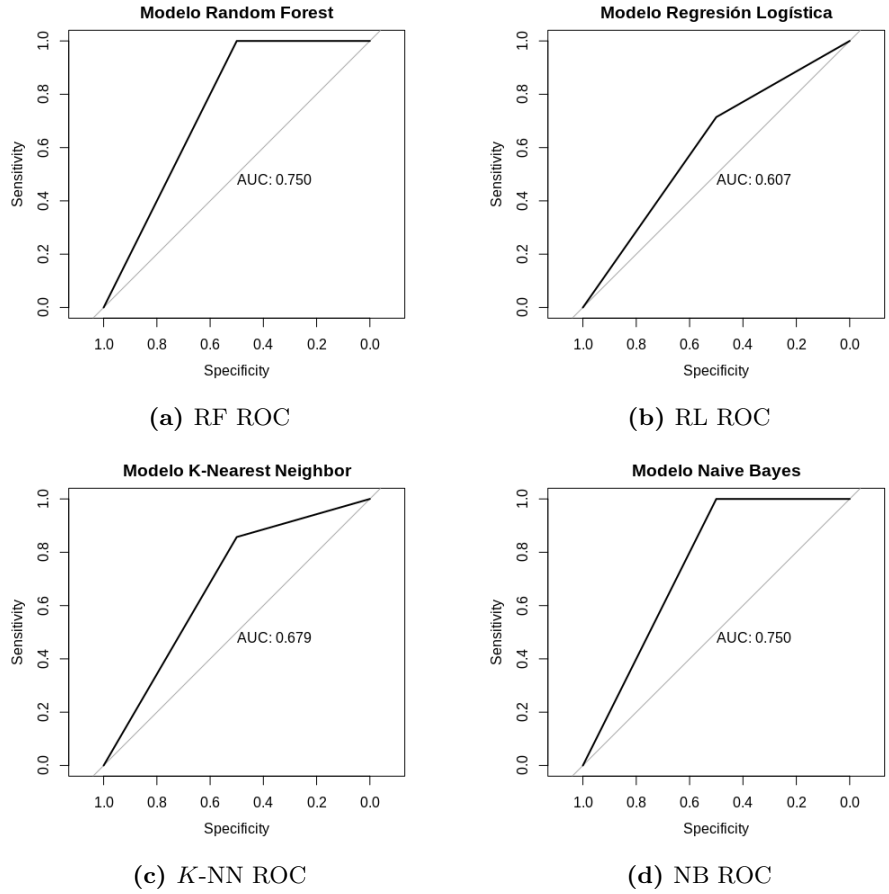
Toda la experimentación fue desarrollada en R [22], un entorno de software libre para gráficos y computación estadística.

4. Discusión y conclusiones

Se realizó un estudio comparativo sobre distintos clasificadores reportados en la literatura para la clasificación de diferentes problemas en el área biomédica, tomando en cuenta el perfil de AGCC para clasificar los casos de muerte por SMSL e incrementar las pruebas de que existe una relación de los procesos metabólicos con este síndrome.

Los clasificadores que se tomaron en cuenta fueron: Regresión Logística (RL), K-Nearest Neighbor (kNN), Random Forest (RF) y Naïve Bayes (NB). En la Tabla 2, se muestran los diferentes clasificadores, con su valor respectivo de exactitud y de área bajo la curva. De esta comparación se puede concluir los siguientes puntos:

- Independientemente del modelo de clasificación seleccionado, todos tiene la capacidad de tener un área bajo la curva mayor a 0.5, lo cual es un factor a tomar en cuenta en un tamizaje de AGCC.
- El desempeño del modelo de regresión logística es menor que el resto en ambas métricas, sin embargo es un modelo de bajo costo computacional que permite evaluar el comportamiento de los algoritmos de clasificación para este problema, así como ser un algoritmo que de manera clásica se utiliza en el área de medicina.



(a) RF ROC **(b)** RL ROC

(c) K-NN ROC **(d)** NB ROC

Fig. 1. Curvas ROC obtenidas para los distintos modelos de clasificación.

– Los modelos de Random Forest y de Naïves Bayes obtienen un desempeño igual para esta muestra de sujetos de SMSL, con un área bajo la curva de 0.75 y una exactitud de 0.8889 para ambos clasificadores. Ambos con una sensibilidad de 1, pudiendo así clasificar correctamente a todos los positivos.

Es importante mencionar que la base de datos utilizada para realizar este análisis es limitada a 18 sujetos, y es un conjunto desbalanceado ya que 5 son sujetos de control y 13 casos diagnosticados con SMSL, lo cual dificulta la generalización del comportamiento de los clasificadores, así como el hecho de que RF y NB tengan el mejor desempeño, ya que debido a la naturaleza de ambos clasificadores pueden estar sobre ajustados a la clase de casos y por ello tener una sensibilidad de 1.

5. Trabajo futuro

Debido a que el SMSL es un proceso multifactorial el cual incluye procesos metabólicos de los ácidos grasos, se propone como trabajo a futuro, realizar pruebas utilizando técnicas de reducción de características para identificar los ácidos grasos de cadena corta que tienen mayor influencia en este proceso, así como la revisión del desempeño de algoritmos de mayor costo computacional. De manera específica quedando los siguientes puntos:

- Selección de características utilizando métodos stepwise.
- Análisis univariado de los ácidos grasos de cadena corta para observar su comportamiento relacionado con el SMSL.
- Implementación de algoritmo genético para observar la interacción de diferentes combinaciones de ácidos grasos así como la importancia de los mismos.
- Implementar técnicas complejas de aprendizaje por computadora, con la limitante del número de sujetos, que es pequeña.

Referencias

1. Alcalá-Rmz, V., Zanella-Calzada, L.A., Galván-Tejada, C.E., García-Hernández, A., Cruz, M., Valladares-Salgado, A., Galván-Tejada, J.I., Gamboa-Rosales, H.: Identification of diabetic patients through clinical and para-clinical features in mexico: an approach using deep neural networks. International journal of environmental research and public health 16(3), 381 (2019)
2. Bright, F., Vink, R., Byard, R., Duncan, J., Krous, H., Paterson, D.: Abnormalities in substance p neurokinin-1 receptor binding in key brainstem nuclei in sudden infant death syndrome related to prematurity and sex. PLoS ONE 12(9), 1–14 (2017)
3. Buchanan, G.: Impaired co2-induced arousal in sids and sudep. Trends in Neurosciences 42(4), 242–250 (2019)
4. Byeon, H.: Is the random forest algorithm suitable for predicting parkinson's disease with mild cognitive impairment out of parkinson's disease with normal cognition? International Journal of Environmental Research and Public Health 17(7), 2594 (2020)
5. Chavez-Ocaña, S.; Bravata-Alcantara, J., Sierra-Martinez, M.: Errores innatos del metabolismo, una mirada a un tópico poco valorado. Rev Hosp Jua Mex 85(3), 159–167 (2018), [ ]
6. Chávez-Ocaña, S., Bravata-Alcántara, J., Sierra-Martínez, M.: Errores innatos del metabolismo, una mirada a un tópico poco valorado. Rev Hosp Jua Mex 85(3), 159–167 (2018)
7. Courts, C., Madea, B.: Genetics of the sudden infant death syndrome. Forensic Science International 203(1), 25–33 (2010)
8. Dreiseitl, S., Ohno-Machado, L.: Logistic regression and artificial neural network classification models: a methodology review. Journal of biomedical informatics 35(5-6), 352–359 (2002)
9. Galván-Tejada, C.E., Zanella-Calzada, L.A., Galván-Tejada, J.I., Celaya-Padilla, J.M., Gamboa-Rosales, H., Garza-Veloz, I., Martinez-Fierro, M.L.: Multivariate feature selection of image descriptors data for breast cancer with computer-assisted diagnosis. Diagnostics 7(1), 9 (2017)

10. Galván Tejada, J.I., Galván Tejada, C.E., López Monteagudo, F.E., Alonso González, O., Moreno Báez, A., Celaya Padilla, J.M., Zanella Calzada, L.A., et al.: Image registration measures and chronic osteoarthritis knee pain prediction: Data from the osteoarthritis initiative. Revista mexicana de ingeniería biomédica (2019)
11. Guntheroth, W., Spiers, P.: The triple risk hypotheses in sudden infant death syndrome. Pediatrics 110(5), 1–6 (2002)
12. Hajian-Tilaki, K.: Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. Caspian journal of internal medicine 4(2), 627 (2013)
13. Halabi, S.S., Prevedello, L.M., Kalpathy-Cramer, J., Mamonov, A.B., Bilbily, A., Cicero, M., Pan, I., Araújo Pereira, L., Teixeira Sousa, R.: The rsna pediatric bone age machine learning challenge. Radiology 290(2), 498–503 (2018)
14. Haynes, R.: Biomarkers of sudden infant death syndrome (sids) risk and sids death. In: Duncan, J., Byard, R. (eds.) SIDS Sudden Infant and Early Childhood Death: The Past, the Present and the Future. Adelaide (AU), University of Adelaide Press (2018)
15. Kinney, H., Thach, B.: The sudden infant death syndrome. N Engl J Med 361(8), 795–805 (2009)
16. Kokol, P.; Završnik, J., Blažun Vošner, H.: Artificial intelligence and pediatrics: A synthetic mini review. Pediatr Dimensions 2(4), 1–5 (2017)
17. Labayru Echeverría, M.: Libro blanco de la muerte súbita infantil, chap. Errores innatos del metabolismo. Patología neurológica y muerte súbita del lactante, pp. 90–102. No. 2003 in NA, Ediciones Ergon, Madrid, España. Grupo de Trabajo de Muerte Súbita Infantil-AEP, 2 edn. (2013)
18. Lugo-Reyes, S., Maldonado-Colín, G., C., M.: Inteligencia artificial para asistir el diagnóstico clínico en medicina. Revista Alergia México 61, 110–120 (2014)
19. Nikolaos, S., Ioakeimidis, T., Meditskou, S., Iakovidou-Kritsi, Z.: Sudden infant death syndrome due to long qt syndrome: a brief review of the genetic substrate and prevalence. Journal of Biological Research-Thessaloniki 24(6), 1–6 (2017)
20. Opdal, S., Rognum, T.: The sudden infant death syndrome gene: does it exist? Pediatrics 114(4), e506–12 (2004)
21. Peterson, L.E.: K-nearest neighbor. Scholarpedia 4(2), 1883 (2009)
22. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2020), `https://www.R-project.org/`
23. Rodríguez-Ruiz, J.G., Galván-Tejada, C.E., Zanella-Calzada, L.A., Celaya-Padilla, J.M., Galván-Tejada, J.I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Soto-Murillo, M.A.: Comparison of night, day and 24 h motor activity data for the classification of depressive episodes. Diagnostics 10(3), 162 (2020)
24. Romero-Rosales, B.L., Tamez-Pena, J.G., Nicolini, H., Moreno-Treviño, M.G., Trevino, V.: Improving predictive models for alzheimer's disease using gwas data by incorporating misclassified samples modeling. Plos one 15(4), e0232103 (2020)
25. Saritas, M.M., Yasar, A.: Performance analysis of ann and naive bayes classification algorithm for data classification. International Journal of Intelligent Systems and Applications in Engineering 7(2), 88–91 (2019)
26. Sarquella-Brugada, G., Campuzano, O., Cesar, S.: Sudden infant death syndrome caused by cardiac arrhythmias: only a matter of genes encoding ion channels? Int J Legal Med 130, 415–420 (2016)
27. Sweeting, J., Semsarian, C.: Cardiac abnormalities and sudden infant death syndrome. Paediatr Respir Rev 15(4), 301–6 (2014)

28. Tamez-Pena, J., Orozco, J., Sosa, P., Valdes, A., Nezhadmoghadam, F.: Ensemble of svm, random-forest and the bswims method to predict and describe structural associations with fluid intelligence scores from t1-weighed mri. In: Challenge in Adolescent Brain Cognitive Development Neurocognitive Prediction. pp. 47–56. Springer (2019)

ISSN 1870-4069

Detección de trayectorias anómalas en flotillas de vehículo

Alexis Jair López-Gutiérrez, Perfecto Malaquias Quintero-Flores, Ana Karen Hernandez-Vazquez

Tecnológico Nacional de México,
Instituto Tecnológico de Apizaco,
Departamento de Estudios de Posgrado e Investigación,
México

{alexistegno,anakarenhv26,parfait.phd}@gmail.com

Resumen. En la actualidad, las prácticas logísticas para la transportación y distribución de mercancías constituyen un elemento clave para elevar la competitividad de empresas, regiones y países. No obstante, la elección de rutas alternas ajenas a la planificación de rutas provoca que las empresas logísticas terminen proporcionando servicios de mala calidad con unidades que ponen en riesgo la entrega oportuna de las mercancías contribuyendo negativamente en el funcionamiento de la cadena de suministro. El análisis de datos espacio-temporales pueden ayudar a obtener información más precisa sobre las opciones de ruta de las flotillas de vehículos. En este sentido, el presente estudio tiene como objetivo desarrollar e implementar un módulo que permita el procesamiento, análisis y despliegue de información satelital orientada al análisis de patrones para encontrar anomalías en las trayectorias de los operadores. La implementación del algoritmo denominado TODS - Detección y segmentación de valores atípicos de trayectoria, agrupa y segmenta trayectorias en rutas normales y rutas anómalas en diferentes períodos del día, para ser de ayuda en la toma de decisiones y minimizar costos de operaciones. Para evaluar los resultados se realizó pruebas utilizando tres conjuntos de datos de trayectorias recopilados por los operadores en Tlaxcala, México. La evaluación de resultados indica que TODS encontró entre un 30 % y un 40 % de rutas anómalas. Además, es posible observar el rendimiento del algoritmo, que en promedio se gasta más del 90 % en tiempo de ejecución en la recuperación de información.

Palabras clave: Datos espacio-temporales, segmentación de trayectoria, agrupación de trayectoria, algoritmo basado en densidad, algoritmo basado en distancia.

Trajectory Outlier Detection on Vehicle Fleet

Abstract. Nowadays, logistics for transportation and distribution of merchandise are a key element to increase the competitiveness of companies, regions, and countries. However, the election of alternative routes

outside the panned routes causes the logistic companies to provide a poor-quality service, using units that endanger the appropriate deliver of merchandise and impacting negatively the way in which the supply chain works. The analysis of the spatial-temporal data can help get more precise information about the route options for vehicle fleets. With this in mind, this paper aims to develop and implement a module that allows the processing, analysis and deployment of satellite information oriented to the pattern analysis, to find anomalies in the paths of the operators by implementing the algorithm TODS – Trajectory Outlier Detection and Segmentation, to group and segment car road trajectories in normal routes and routes abnormal during different times of the day to be able to help in the decision making, focusing on minimizing operational costs. To evaluate the results, tests were carried out using three sets of trajectory data collected by the operators in Tlaxcala, Mexico. The evaluation of results indicates that TODS found between 30 % and 40 % of failed routes. In addition, it is possible to observe the performance of the algorithm, which on average spends more than 90 % at runtime in information retrieval.

Keywords: Spatial-temporal data, trajectory segment, trajectory clustering, density-based algorithm, distance-based algorithm.

1. Introducción

La logística de transporte y distribución de mercancías a nivel global se encuentra en una fase de inflexión respecto a lo que se venía haciendo en años anteriores, saliendo a flote una serie de problemas de distribución que se han vuelto comunes en el sector. Una mala planificación de rutas logísticas de entrega, no tener los vehículos en tiempo y forma, o tener retrasos en la entrega, son parte de los problemas de distribución a los que mayormente se ven enfrentadas las empresas mexicanas. De acuerdo con Kotler et al. [5], los principales elementos de los costos de la distribución física son el transporte (37 %), el control de existencias (22 %), el almacenamiento (21 %) y otros como la recepción de órdenes, el servicio al cliente, la distribución y la administración (20 %). Por tanto, el transporte de distribución logística es la actividad económica más importante entre los componentes de las estepas de la logística empresarial.

Al entender la distribución física de productos como la actividad de trasladar los bienes desde el centro de producción hasta el consumidor final, es evidente pensar que el elemento clave de una cadena de suministro es el sistema de transporte logístico, sin embargo existen malas prácticas en las empresas causando deficiencias en la productividad para la entrega de productos con atrasos de tiempo en la gestión de la cadena de suministros, contribuyendo negativamente con más de la mitad de las perdidas en las empresas siendo factor de merma y declive de las empresas Mexicanas de acuerdo con el estudio [1]. Por tanto, existe la necesidad de identificar y controlar todos aquellos factores del costo para garantizar la competitividad empresarial.

Los operadores encargados de repartir insumos en muchas ocasiones definen la planeación de rutas de manera empírica. La asignación empírica consiste en la repartición de los insumos de acuerdo con la experiencia del operador. En muchas ocasiones, la elección de rutas alternas ajenas a la planificación de rutas provoca que las empresas logísticas terminen proporcionando servicios de mala calidad con unidades que ponen en riesgo la entrega oportuna de las mercancías y el funcionamiento de la cadena de suministro, padeciendo esta situación principalmente las pequeñas empresas arrastrando consigo a las medianas y a las grandes empresas del país.

Con fundamento en lo anterior, la mejor manera para reaccionar ante posibles hechos anómalos que afecten a la optimización de rutas es la detección de las trayectorias fuera del patrón. Tener información precisa sobre las rutas del operador durante un período de tiempo permitirá aprender, estimar y detectar anomalías en la planeación de rutas, facilitando la toma de decisiones en los costos de operaciones. En éste trabajo se implementa un algoritmo de agrupación y segmentación de trayectorias de flotillas de los vehículos aplicando el enfoque basado en agrupación y un enfoque basado en distancia. Después, los resultados de la agrupación se utilizan para determinar rutas anómalas.

El resto de éste documento está organizado de la siguiente manera. En la sección 2 presenta los antecedentes referentes a esta investigación. La implementación del algoritmo para la detección de trayectorias anómalas se describe en la sección 3. La sección 4 presenta los resultados obtenidos y por último en la sección 5 concluye éste trabajo.

2. Antecedentes

Existe varios trabajos que abordan la detección de trayectorias anómalas desde un enfoque basado en distancia como se demuestra en el trabajo [9] donde proponen un algoritmo llamado Center of Sliding Window (CSW) para detectar ubicaciones de conducción anormales causadas por condiciones de tráfico particulares como semáforos, cruces de calles o rotondas. Lee et al. [6] propusieron el algoritmo TRAjectory Outlier Detection (TRAOD) utilizando una medida de similaridad denominada distancia angular, en el cual la dirección de las sub-trayectorias anómalas difiere de las de las sub-trayectorias vecinas. Por otro lado, algunos trabajos se han centrado en la detección de trayectorias atípicas basados en densidad, estudios como [2] proponen un nuevo concepto de valor atípico local basado en densidad con datos inciertos para detectar rápidamente valores atípicos.

Los métodos de detección atípicos basados en la densidad relativa pueden procesar conjuntos de datos de densidad múltiple de manera efectiva. La complejidad temporal de éste tipo de métodos puede llegar a $O(m^2)$, donde m es el número de puntos en la base de datos, lo cual es comparable a los enfoques basados en la distancia. En la investigación [8] propusieron un algoritmo de detección de valores atípicos basados en la trayectoria de la densidad para compensar las desventajas del algoritmo TRAOD que no puede detectar anomalías

cuando existe densidad entre los puntos dentro de una única región de interés. Sin embargo, Cao et al. [2] propuso un nuevo concepto de valor atípico local basado en la densidad en datos inciertos para detectar rápidamente valores atípicos. En contraste con el enfoque basado en densidad y el enfoque basado en distancia, existen varios algoritmos de agrupación para encontrar valores atípicos en el conjunto de datos, como K-means, DBSCAN, STING y BRICH Yuan et al. [11]. Liao [7] señaló que los valores atípicos son el subproducto de la agrupación. Domínguez et al. [3] aplicaron un algoritmo de agrupación basado en la densidad, obteniendo el pulso de la ciudad permitiendo la detección de anomalías (comportamientos inesperados) entre los usuarios en Nueva York.

3. Desarrollo de la propuesta de solución

Para abordar éste problema, se utilizo un enfoque híbrido basado en agrupación y basado en distancia utilizando la medida de similaridad denominada distancia euclidiana y distancia angular con el fin de segmentar rutas normales con un alto número de conducción por esa vía y trayectorias anómalas con menor número de conducción en ciertos intervalos de tiempo, con la implementación del algoritmo TODS [10] para dar solución a la detección de anomalías en trayectorias utilizando datos históricos. La selección del algoritmo se debió a una exhaustiva revisión de trabajos referentes al área, además porque satisface la solución para encontrar trayectorias qué salgan del patrón entre los operadores logísticos.

Como se describe en la Figura 1, los datos de las trayectorias de los dispositivos son recopilados y almacenados en una base de datos. Después se procesan aplicando una serie de filtros para finalmente agruparlos y segmentarlos en dos conjuntos, anómalos y normales respectivamente. El algoritmo principal esta dividido en 3 fases; Extracción y Preprocesamiento de Datos, Agrupación y Segmentación. Se observa en el algoritmo 1 y sus entradas se definen de la siguiente manera:

- Región de inicio (SR) y región de finalización (ER): representan las regiones rectangulares donde se deben cruzar las trayectorias seleccionadas;
- Hora de inicio (SH) y hora de finalización (EH): representan el intervalo de tiempo. en el cual las trayectorias deben cruzar SR y ER, estos deben respetar la desigualdad $SH < EH$;
- Tamaño de interpolación (I), desviación estándar (SD) y sigma (σ) : son parámetros que se utilizan para preprocesamiento y limpieza de las trayectorias;
- Las medidas de similaridad utilizadas para éste trabajo son la distancia euclidiana (D) y la distancia angular (θ): se utilizan para identificar el par de segmentos; anómalos y normales.
- El número de grupos que hacen referencias a grupos con un mismo patrón (KS): Es el numero mínimo de agrupaciones normales a encontrar.

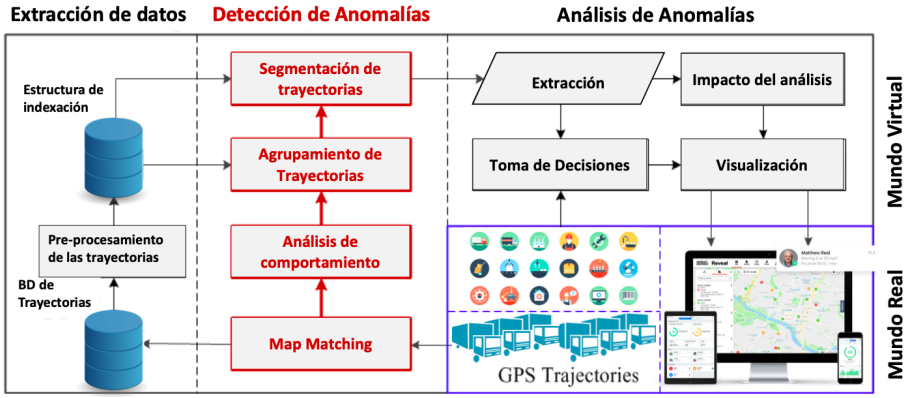


Fig. 1. Proceso de Extracción y análisis de datos para la detección de trayectorias anómalas.

La salida del algoritmo son dos conjuntos ordenados de trayectorias normales y anómalas, además con sus respectivos segmentos clasificados en secuencias.

Algoritmo 1 Algoritmo principal.

Entrada: $SH, EH, SR, ER, I, SD, \sigma, D, \theta, KS$

1: t ← $FindTrajectories(SR, ER)$
2: **Para** i ← 0 `to` Length(t) **Hacer**
3: FilterNoisePoints(t[i], SD,σ)
4: InterpolatePoints(t[i], I)
5: **Fin Para**
6: C ← GetCandidates(t, SH, EH, SR, ER)
7: idx ← CreateClusteringGrid(C, D)
8: GT ← GetTrajectoriesGroups(C, idx)
9: SGT ← GetStandardTrajectories(GT, KS)
10: R ← GetTrajectoriesRoutes(GT, SGT, D, θ)
11: **return** SGT,R

3.1. Fase 1 - Extracción y preprocesamiento de trayectorias

Se comienza a obtener los datos de las trayectorias sin procesar desde la base de datos con el método FindTrajectories (línea 1). El sistema de base de datos utilizado fue PostgreSQL, debido a su capacidad para ejecutar consultas geométricas para recuperar todas las trayectorias almacenadas que se cruzan con dos geometrías rectangulares distintas (SR y ER). Una vez recuperado los datos, se procede al procesamiento de los datos en el método FilterNoisePoints (línea 3) eliminando puntos inválidos en dos pasos. El primer paso calcula una distribución normal usando la distancia entre todos los puntos adyacentes. Se eliminan los puntos que extrapolan el intervalo de confianza definido por σ con

el método InterpolatePoints (línea 4). El parámetro I define la distancia límite superior entre dos puntos adyacentes.

Por lo tanto, si la distancia entre dos puntos adyacentes es mayor que el valor de I, se ejecuta una interpolación para insertar tantos puntos como sea necesario para ajustar la distancia del punto más corto que I. La interpolación aumenta la densidad de puntos de las trayectorias para mejorar el agrupamiento realizada por el algoritmo de agrupación. El segundo paso calcula la desviación estándar de los puntos de trayectoria para eliminar por completo las trayectorias que superan el parámetro SD definido. El siguiente paso es eliminar las trayectorias que no cumplen con el intervalo de tiempo dado por SH y EH, filtrando segmentos que respetan la dirección de la trayectoria con el método GetCandidates (línea 6). Para la recuperación y optimización de consultas espacio - temporales, en éste trabajo se utilizo un sistema de estructura de índice basada en cuadricula para facilitar la resolución de consultas por el algoritmo de agrupación en una forma eficiente mapeando los puntos de las trayectorias como se puede observar en el algoritmo 2. El algoritmo de agrupación utilizado en éste trabajo se denomina Density-based spatial clustering of applications with noise (DBSCAN.

3.2. Fase 2 - Agrupación

La fase de agrupación usa las medidas de similitud basada en distancia; la distancia euclidiana y la distancia angular. El Algoritmo 2 detalla la función GetTrajectoriesGroups (línea 8) del Algoritmo 1. Las entradas del algoritmo son: el conjunto de candidatos C y el sistema de indexación idx. Ambas variables son producidas por la fase de Extracción y Preprocesamiento de Trayectorias. La salida del algoritmo es un conjunto de grupos (GT), donde cada grupo contiene trayectorias con una ruta similar entre SR y ER, como se puede ilustrar en la figura 2.

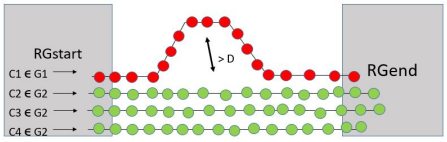


Fig. 2. Agrupación de candidatos.

El algoritmo 2 comienza a ordenar la lista de candidatos en orden ascendente por el total de puntos (línea 1). Para cada candidato, el algoritmo comienza a limpiar la variable currentGroup (línea 4). Después de eso, realiza una comparación con el candidato actual c con los elementos del grupo en GT (línea 6). Si el candidato coincide en un grupo existente, la variable currentGroup almacena ese grupo respectivo y da por finalizado la búsqueda (líneas 7–8). La función

FitInGroup utiliza la estructura de índice para verificar la distancia de todos los puntos pertenecientes a c en la agrupación de trayectorias actuales G.

Si no se ha encontrado el grupo G (línea 11), se crea un nuevo grupo para asignar el candidato (línea 12) y éste se agrega al grupo GT. Finalmente, el candidato c se asigna al grupo actual (línea 15) y el siguiente candidato repite el proceso (línea 3).

Algoritmo 2 Algoritmo de agrupación de trayectorias.

Entrada: C, idx
1: $SortTrajectories(C)$
2: GT $\leftarrow$ $CreateSetStructure()$
3: **Para todo** c $\in$ C **Hacer**
4: $currentGroup \leftarrow empty$
5: **Para todo** c $\in$ C **Hacer**
6: **Si** $FitInGroup(c, G, idx)$ **entonces**
7: $currentGroup \leftarrow$ G
8: $break$
9: **Fin Si**
10: **Fin Para**
11: **Si** $currentGroup = empty$ **entonces**
12: $currentGroup \leftarrow CreateGroup()$
13: $AddGroup(GT, currentGroup)$
14: **Fin Si**
15: $AddCandidate(currentGroup, c)$
16: **Fin Para**
17: **return** GT

3.3. Fase 3 - Segmentación

Los parámetros de entrada del algoritmo 3 son: conjunto de grupos (GT), conjunto de grupos normales (STG), conjunto de grupos anómalos ($NSTG$), distancia de agrupamiento (D), y el umbral de la diferencia angular (θ) El proceso de segmentación consta de dos pasos, en el primer paso en la figura 3 realiza la segmentación agrupando las trayectorias usando la distancia euclidiana. Identifica todos los puntos de trayectoria como normales o anómalas utilizando la función de agrupación HasNear, clasificando un punto dado como normal debido a su proximidad a puntos de trayectorias cercanos.

En el segundo paso realiza la corrección de segmentación evaluando la diferencia angular entre segmentos de trayectorias, reclasificando los puntos del principio al final de los segmentos anómalos calculando la diferencia angular entre ellos y los segmentos normales utilizando las funciones BackExtension y FrontExtension, respectivamente. En la figura 4 se observa la segmentación por diferencia angular. Finalmente se crea una ruta compuesta por segmentos normales y anómalos etiquetados por DS y NDS, respectivamente. Finalmente, el algoritmo regresa los segmentos de rutas anómalos separados en conjuntos

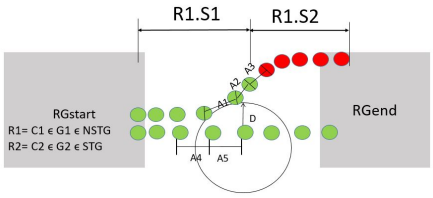


Fig. 3. Segmentación de candidatos por distancia.

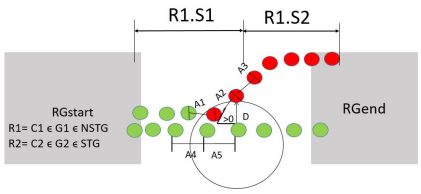


Fig. 4. Segmentación de candidatos por diferencia angular.

normales y anómalos, para la previa visualización utilizando el Api de Google MAPS.

Algoritmo 3 Algoritmo de segmentación de trayectorias.

Entrada: GT, STG, NSTG, D, θ

$res \leftarrow CreateList()$
Para todo C $\in$ NSTG **Hacer**
  Para todo p $\in$ C.points **Hacer**
    $p.std \leftarrow HasNear(STG, p, D)$
  Fin Para
  Para todo p $\in$ C.points **Hacer**
    Si p.std = FALSE **entonces**
      $BackExtension(p, C, STG, D, \theta)$
      $FrontExtension(p, C, STG, D, \theta)$
    Fin Si
  Fin Para
  $R \leftarrow CreateRoute()$
  $R \leftarrow CreateRoute()$
  $Split(R.DS, R.NDS, C)$
  $AddRoute(res, R)$
Fin Para
return res

4. Resultados

Para el desarrollo del módulo, se utilizo Java (JDK versión 1.8.0_121) como lenguaje de programación, por sus bondades a la hora de la ejecución y rendimiento. Para el manejo de consultas y recuperación de la información, se utilizo el gestor de base de datos PostgreSQL, porque cuenta con una extensión para manejar datos geoespaciales, y tiene como principales características, los tipos de datos espaciales, índices espaciales y funciones que operan sobre ellos. Con respecto a la especificación de hardware, se utilizó una macBook Air con procesador Intel Core I5 y 8GB de RAM para ejecutar el módulo.

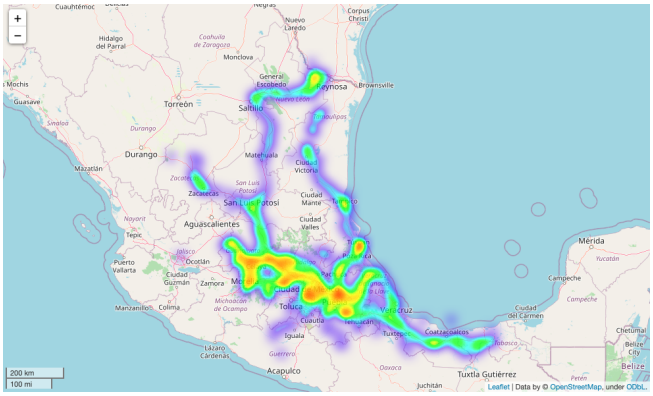


Fig. 5. Representación visual de la densidad espacial.

Para éste trabajo se utilizo el conjunto de datos del proyecto RutaGPS, teniendo información espacio - temporal de más de 50 usuarios en un período de 6 meses, es decir, marzo del 2019 hasta septiembre del 2019. Contiene 17,621 trayectorias y 6,798,282 puntos, incluidos todos los usuarios, en la tabla 1 se observa un extracto de los datos.

En la figura 5 se muestra la representación visual de la densidad espacial. Todo el análisis se realizó utilizando tres conjuntos de datos heterogéneos.

El primer conjunto de datos contiene aproximadamente 5024 trayectorias recopiladas durante 2019 por 24 semanas, por un operador que hace su repartición de productos entre la ciudad de Tlaxco, Tlaxcala a Apizaco Tlaxcala. El segundo conjunto tiene aproximadamente 14074 trayectorias recopiladas durante 2019 por 50 semanas, por un operador que se desplaza desde la zona conurbada de San Pablo Apetatitlán, Santa Ana Chiahutempan y Santa Cruz Tetela, en el estado de Tlaxcala. Y el tercer conjunto cuenta con 5880 trayectorias durante 2019 por 22 semanas, por un operador logístico que se desplaza de la ciudad de Apizaco, Tlaxcala a la ciudad de Santa Ana Chiahutempan. La Tabla 2 presenta el resumen de los 3 conjuntos de datos.

Los valores para los parámetros definidos para el primer conjunto son; $SH = 8h, EH = 20h, I = 35m, SD = 100m, \sigma = 99{,}7\,\%, D = 50m, \theta = 30$ y $KS = 20$. Para el segundo conjunto los valores seleccionados son: $SH = 10h, EH = 16h, I = 30m, SD = 120m, \sigma = 99{,}7\,\%, D = 50m, \theta = 30$ y $KS = 3$. Y para el tercer conjunto los parámetros seleccionados son: $SH = 9h, EH = 20h, I = 35m, SD = 100m, \sigma = 99{,}7\,\%, D = 50m, \theta = 30$ y $KS = 2$. Algunos parámetros como SD, σ y I pueden definirse automáticamente en futuros estudios. Estos valores de parámetros se definieron empíricamente en base a la densidad de puntos que cuenta por naturaleza cada conjunto de datos seleccionados, así como por las sugerencias relacionadas por la literatura.

Para el primer escenario, el operador tiene un patrón de movilidad mediante la carretera federal Tlaxco-Apizaco, observada en color verde en la figura 6. Por otra parte, se observa una anomalía con una trayectoria atípica pintada en

Tabla 1. Formato de datos del conjunto RutaGPS.

ID	Deviceid	Fixtime	Latitude	Longitude	Speed (kn)
1	1	2019-01-31 15:13:13	9.5607245	-98.130251	0.12
2	1	2019-01-31 15:13:13	19.560675	-98.130277	0.13
3	1	2019-01-31 15:13:14	19.560788	-98.130509	0.02
.	.	.	.	.	.
5601	2	2019-02-02 08:04:04	19.47644	-98.12148	2.94
5602	2	2019-02-02 08:05:25	19.47924	-98.12568	7.86
.	.	.	.	.	.
11304	3	2019-02-04 14:04:04	19.452497	-98.1027815	6.76
11305	3	2019-02-04 14:06:06	19.45248	-98.1027885	6.80

color rojo en la figura 6 tomando una ruta distinta a la frecuente desviando su trayectoria por segmentos que conducen a rutas fuera del patrón de movilidad del operador, llegando a la ciudad de Apizaco por segmentos distintos a la carretera federal Tlaxco-Apizaco.

Para el segundo escenario, el operador tiene una trayectoria cotidiana mediante la carretera federal Vía corta Sta. Ana-Puebla/México, observada en color verde en la figura 6 tomando una ruta distinta a la frecuente, desviando su trayectoria por segmentos que conducen a rutas fuera del patrón de movilidad del operador, llegando a la población de Magdalena Tlatelulco por segmentos distintos a la carretera federal Vía corta. La ruta anómala se desvía antes de tomar la carretera federal Apizaco - Chiahutempan, tomando una trayectoria por la carretera federal Apizaco - Tlaxcala, con destino al entronque hacia el libramiento de Tlaxcala. Esta anomalía se puede explicar, ya que las trayectorias tomadas durante éste proceso son trayectorias que aluden al periodo de mantenimiento y trabajos de ampliación en la carretera Apizaco - Chiahutempan. Razón probable que en éste periodo existía un embotellamiento de autos durante la hora pico. Por ultimo, para el escenario tres, el operador logístico tiene una ruta desde la ciudad de Apizaco, Tlaxcala a la ciudad de Santa Ana Chiahutempan, Tlaxcala, tal y como se observa en la Figura 6. Para el escenario 3 se observan distintos patrones en el mapa de carretera, estos patrones de igual manera se vieron afectados por los trabajos de mantenimiento y de ampliación en la carretera federal Apizaco-Chiahutempan. Existe una trayectoria atípica resaltada en color rojo observada en la Figura 6, tomando una ruta distinta a la frecuente desviando su trayectoria por segmentos que rodean a la carretera federal. Cabe mencionar que la ruta seleccionada cómo anómala, es una ruta común para el transporte publico durante los trabajos de mantenimiento.

El consumo de combustible es mayor en las constantes paradas entre cruces que en segmentos donde la velocidad promedio puede alcanzar hasta los 100 km/h. Si bien la trayectoria observada en color rojo representa un evento anómalo, se puede concluir que el consumo de combustible es mayor en las constantes paradas entre cruces de camino cada 200 metros que en segmentos donde la velocidad promedio puede alcanzar hasta los 100 km/h. Además, el tiempo que se demora es significativamente menor en segmentos de carretera

Tabla 2. Resumen estadístico de conjuntos de datos sobre trayectorias y puntos recopilados (coordenadas).

Estatísticas	Escenario 1	Escenario 2	Escenario 3
Cantidad de trayectorias	5024	14074	5880
Cantidad de puntos de trayectoria (Media)	252.9	45.1	270.3
Cantidad de puntos de trayectoria (SD)	494.9	32.9	501.8
Cantidad de puntos de trayectoria (Max)	4810	1127	5120
Cantidad de puntos de trayectoria (Min)	1	1	1
Cantidad de puntos	20354	90450	15223
Distancia de puntos adyacentes (Media)	6.4(m)	48.9(m)	6.8(m)
Distancia de puntos adyacentes (SD)	7.9(m)	908.1(m)	8.1(m)
Distancia de puntos adyacentes (Max)	234.9(m)	669323.1(m)	238.9(m)
Distancia de puntos adyacentes (Min)	0.0(m)	0.0(m)	0.0(m)

que en segmentos donde existen relativamente cruces por cada intersección de calles.

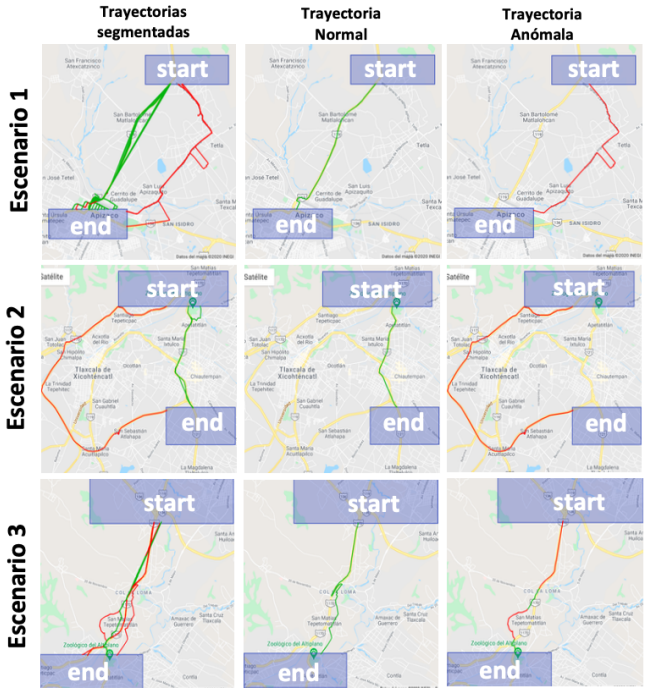


Fig. 6. Resultados de cada escenario.

Analizando los resultados de las trayectorias en la tabla 3 es posible observar que el algoritmo TODS encontró más rutas normales que anómalas.

Tabla 3. Estadísticas en la detección de Trayectorias anómalas por escenario.

Variables	Conjunto 1	Conjunto 2	Conjunto 3
Tiempo (ms) Consulta BD	2697.41	402.20	1214.29
Tiempo (ms) Obtención de candidatos	492.01	25.40	31.14
Tiempo (ms) Agrupación de candidatos	9.84	0.06	0.24
Tiempo (ms) Obtención de trayectorias normales	0.03	0	0.02
Tiempo (ms) Segmentación de trayectorias	21.13	0.23	0.61
Trayectorias analizadas	37	5	6
Grupos trayectorias normales	25	3	4
Grupos trayectorias anómalas	12	2	2
Trayectorias analizadas %	100 %	100 %	10 %
Trayectorias normales %	67 %	60 %	66.6 %
Trayectorias anómalas %	33 %	40 %	33.3 %

Se explica debido al hecho de que el algoritmo agrupa las trayectorias similares en dos fases; 1 en la primera fase agrupa las trayectorias usando la distancia euclidiana. En la segunda fase agrupa las trayectorias con la diferencia angular. Además, se observa el tiempo por cada fase medido en milisegundos, esto por la cantidad mínima de trayectorias analizadas. La selección de las dos regiones denominadas "startz .endd", delimita el espacio de búsqueda en el conjunto de datos, al evaluar un área de interés. El resultado de analizar áreas de interés, las observamos en la Tabla 3, con 37 trayectorias analizadas, 25 grupos de trayectorias normales y 12 grupos de trayectorias anómalas para el escenario 1. En el escenario 2 se analizaron 5 trayectorias en el espacio de búsqueda, se encontraron 3 grupos de trayectorias normales y 2 grupos de trayectorias anómalas. Y para el escenario 3, se analizaron 6 trayectorias, con 4 trayectorias normales y 2 anómalas.

4.1. Evaluación

Para evaluar el algoritmo TODS, se realizó un análisis cualitativo y cuantitativo. El análisis cualitativo evalúa el proceso del algoritmo para identificar si es adecuado para detectar anomalías.

Además, el análisis cuantitativo evalúa la eficiencia en tiempo de ejecución del algoritmo con respecto a uno de los algoritmos referentes para la detección de anomalías en trayectorias de acuerdo en la literatura denominado TRA-SOD [4].

Para los valores definidos para el algoritmo TRA-SOD son los sugeridos en [4], que son: $maxDist = 50 myminSup = 4$. Los algoritmos TODS y TRA-SOD se compararon bajo los mismos escenarios. Los resultados obtenidos se pueden ver en la Figura 7, observando la clasificación completa de las trayectorias con respecto a las rutas normales y las trayectorias pertenecientes a rutas anómalas. Analizando la clasificación de trayectorias es posible observar que TRA-SOD encontró las mismas rutas anómalas como lo hizo el algoritmo TODS para el escenario uno y tres.

Fig. 7. Comparativa entre los algoritmos TODS y TRA-SOD.

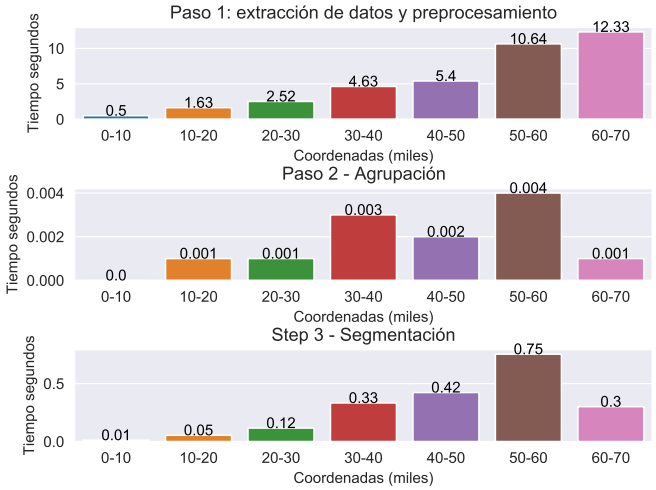


Fig. 8. Tiempo de ejecución por fases.

Sin embargo, para el escenario dos el algoritmo TRA-SOD detecta todas las trayectorias como anómalas. Esto sucede debido a la dificultad para definir el valor apropiado para el parámetro de vecindad k más cercano.

Para evaluar el rendimiento del tiempo de ejecución del algoritmo TODS, se seleccionaron 7000 muestras de pares de regiones generadas aleatoriamente que tienen, al menos, una ruta anómala. Al analizar el rendimiento en tiempo de ejecución del algoritmo TODS considerando cada una de sus fases, en la Figura 8 es posible ver que en promedio se gasta más del 90 % del tiempo de ejecución

en el paso 1 (Recuperación). Esto se explica porque el proceso de generar los grupos y crear la estructura de indexación lleva mucho tiempo. Por el contrario, las fases restantes han reducido el tiempo de ejecución debido a los beneficios aportados por la estructura de índice que aceleran la búsqueda de trayectorias vecinas.

5. Conclusión

La detección de valores atípicos de trayectoria es de gran importancia para reducir y optimizar los costos de operaciones y descubrir eventos particulares en la transportación y distribución en la cadena de suministros. En éste estudio, implementamos el algoritmo TODS para detectar valores atípicos de trayectoria, utilizando una técnica de agrupamiento para agrupar y segmentar trayectorias de los operadores como segmentos normales y anómalos mediante el análisis de los datos históricos. Los resultados experimentales mostraron la efectividad del algoritmo en la detección eventos anómalos. Además, se realizó una comparación con el algoritmo TRA-SOD, mostrando que el algoritmo TODS es mas efectivo en la detección de rutas anómalas. En cuanto a investigaciones futuras, se puede probar la detección en tiempo real. Además, la auto-parametrización para reducir la configuración de los parámetros empíricos.

Referencias

1. 6 problemas de distribución logística de productos [última milla], https://www.beetrack.com/es/blog/logistica-de-distribucion/, library Catalog: www.beetrack.com
2. Cao, K., Shi, L., Wang, G., Han, D., Bai, M.: Density-Based Local Outlier Detection on Uncertain Data. In: Li, F., Li, G., Hwang, S.w., Yao, B., Zhang, Z. (eds.) Web-Age Information Management. pp. 67–71. Springer International Publishing, Cham (2014)
3. Domínguez, D.R., Redondo, R.P.D., Vilas, A.F., Khalifa, M.B.: Sensing the city with Instagram: Clustering geolocated data for outlier detection. Expert Systems with Applications 78, 319–333 (2017)
4. Fontes, V.C., de Alencar, L.A., Renso, C., Bogorny, V.: Discovering trajectory outliers between regions of interest. In: Proceedings of XIV GEOINFO, November 24-27, 2013, Campos do Jordão, Brazil. vol. 1, pp. 49–60 (2013)
5. Kotler, P., Roberto, E.L.: Marketing social: estrategias para cambiar la conducta pública. Ediciones Díaz de Santos (Dec 1991), google-Books-ID: vMRDpBfowEMC
6. Lee, J.G., Han, J., Li, X.: Trajectory Outlier Detection: A Partition-and-Detect Framework. In: 2008 IEEE 24th International Conference on Data Engineering. pp. 140–149 (Apr 2008), iSSN: 2375-026X
7. Liao, T.W.: Clustering of time series data—a survey. Pattern Recognition 38(11), 1857–1874 (2005)
8. Liu, Z., Pi, D., Jiang, J.: Density-based trajectory outlier detection algorithm. Journal of Systems Engineering and Electronics 24(2), 335–340 (Apr 2013)

9. Munoz-Organero, M., Ruiz-Blaquez, R., Sánchez-Fernández, L.: Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. Computers, Environment and Urban Systems 68, 1–8 (2018)
10. Schmitt, J.P., Baldo, F.: A Method to Suggest Alternative Routes Based on Analysis of Automobiles' Trajectories. In: 2018 XLIV Latin American Computer Conference (CLEI). pp. 436–444 (2018)
11. Yuan, G., Sun, P., Zhao, J., Li, D., Wang, C.: A review of moving object trajectory clustering algorithms. Artificial Intelligence Review 47(1), 123–144 (2017)

ISSN 1870-4069

Desarrollo e implementación de un simulador para el seguimiento de objetos en redes de sensores

Ana Karen Hernández-Vázquez, Alexis Jair López-Gutiérrez, Perfecto Malaquías Quintero-Flores

Instituto Tecnológico de México,
Instituto Tecnológico de Apizaco,
Departamento de Estudios de Posgrado,
México

{anakarenhv26,alexistegno,parfait.phd}@gmail.com

Resumen. El seguimiento de objetos tiene varias aplicaciones en la vida real, como la vigilancia en redes de vehículos, monitoreo de vida silvestre, propagación de incendios, observación de pacientes, localización, etc. Las redes de sensores consisten en un conjunto de cientos o miles de dispositivos autónomos, los cuales son llamados nodos sensores, estos nodos están distribuidos físicamente en áreas geográficas para poder monitorear condiciones físicas o ambientales, con una capacidad de almacenamiento y comunicación de datos en una red de forma inalámbrica a corta distancia. Estas aplicaciones deben de recopilar periódicamente los datos sensoriales y usarlos para reconstruir el estado general del área supervisada mediante la agregación de datos. Este trabajo presenta el seguimiento de objetos mediante: 1.- Estructura Jerárquica de Agrupación de sensores para red de sensores, 2.- Seguimiento de ubicación basado en agrupación para redes de sensores de seguimiento de objetos. Los resultados de la simulación muestran la disminución de sobrecarga de comunicación sin disminuir la precisión del seguimiento de objetos y se lo logra la extensión de la vida útil de la red.

Palabras clave: Redes de sensores, seguimiento de objetos, eficiencia energética, vida útil de la red.

Development and Implementation of a Simulator for Tracking Objects in Sensor Networks

Abstract. Object tracking has many applications in real life, such as vehicle network surveillance, wildlife monitoring, fire spread, patient observation, location, etc. Sensor networks consist of a set of hundreds or thousands of autonomous devices, which are called sensor nodes, these nodes are physically distributed in geographic areas to be able to monitor physical or environmental conditions, with a capacity for data storage

and communication in a network wirelessly at close range. These applications should periodically collect sensory data and use it to reconstruct the overall condition of the monitored area through data aggregation. This work presents the tracking of objects through: 1.- Hierarchical Structure of Sensor grouping for sensor network, 2.- Location tracking based on grouping for sensor networks of object tracking. The simulation results show the decrease in communication overhead without decreasing the accuracy of the object tracking and the extension of the useful life of the network is achieved.

Keywords: Sensor networks, object tracking, energy efficiency, network lifetime.

1. Introducción

La visión futurista que Mark Weiser describió en su artículo *The Computer for the 21st Century* [13] es una realidad, se describen entornos saturados de elementos con capacidades de computo y comunicación, totalmente integrados en la vida de las personas y que proporcionan información asociada a las necesidades y al entorno en el que se encuentren en cada momento.

Las redes de sensores presentan una amplia variedad de implementaciones que son utilizadas como aplicaciones, en donde el nodo sensor o varios sensores pueden ser capaces de modelar el estado de un entorno en particular. Entre las múltiples aplicaciones de las redes de sensores se pueden encontrar procesos como monitoreo, vigilancia, seguridad y tácticas de vigilancia, control de inventarios, seguimiento de objetos o personas, atención médica, etc. El desarrollo de este tipo de redes plantea nuevos problemas, pero a su vez acerca más a la realidad escenarios que antes se pensaron en el plano de ciencia ficción.

El resto de este artículo está organizado de la siguiente manera, en la sección 2 se presentan los antecedentes. El trabajo relacionado es presentado en la sección 3. Desarrollo metodológico forma parte de la sección 4. El desarrollo de trabajo experimental e interpretación de resultados obtenidos se presenta en la sección 5. La sección 6 concluye este trabajo.

2. Trabajo relacionado

El seguimiento de objetos de redes de sensores ha llamado la atención de los investigadores debido a su amplia aplicación, la mayoría de estos estudios no pueden abordar el equilibrio entre la eficiencia energética y la precisión del seguimiento. Como parte de la implementación de algoritmos que mejoren el seguimiento de objetos en ámbitos internos y externos, se han publicado un número significativo de propuestas con el enfoque de seguimiento de objetos en redes de sensores. Existe una gran demanda de seguimiento de ubicación de objetos en movimiento desde diversos servicios basados en ubicación [15]. Una de las aplicaciones mas desafiantes de tal tecnología es el seguimiento de objetos, objetivos y/o eventos en movimiento.

En este tipo de redes de sensores, la energía es un problema clave, ya que afecta directamente la vida útil de la red. La energía de las redes se puede conservar cambiando el nodo de un sensor a modo de suspensión cuando no hay ningún objeto en el rango de detección del nodo y activando un nodo del sensor cuando un objeto debe ingresar al rango de detección. Es probable que la ubicación del objeto, la hora de llegada y la ruta oculten algunas reglas de asociación útiles, que se pueden extraer mediante la aplicación del algoritmo de minería de datos adecuado. Es por lo que los autores en [10] han propuesto un esquema de seguimiento de objetos para redes de sensores de seguimiento de objetos (OTSN por sus siglas en inglés) utilizando el enfoque de minería de datos.

En este esquema se ha mejorado el algoritmo Apriori para las reglas de asociación minera y se ha aplicado a *OTSN*, este algoritmo se aplica a la información de movimiento anterior del objeto y se excavan reglas de asociación útiles, que luego se utilizan para predecir la siguiente ubicación del objeto con mayor precisión y aumentar la vida útil de la red. El filtro de Kalman [12], el filtro de partículas y la minería de datos, son los modelos más utilizados para predecir la siguiente ubicación del objeto, sin embargo, estos modelos no toman en consideración la durabilidad de la vida de la red de sensores, es por ellos que este trabajo ofrece un enfoque diferente, donde la vida útil de la red y la eficiencia energética son dos de los mayores puntos a atacar.

Hawbani et al [6] proponen un nuevo modelo de seguimiento, denominado *Seguimiento de ubicación basado en agrupación en redes de sensores de seguimiento de objetos.* (*GLT* por sus siglas en inglés) esta basado en la estructura jerárquica de agrupación (*GHS* por sus siglas en inglés) [7], en *GHS* los nodos se dividen en grupos de acuerdo con su región de cobertura máxima, tal que cada grupo contiene un número de nodos y un número de líderes. *GLT* esta conformado por dos niveles, el primer nivel denominado árbol de notificación (*NT* por sus siglas en inglés), mejora el mecanismo de activación, el mecanismo de limpieza de datos y el mecanismo de balance de energía. En *GLT* no hay necesidad de un mecanismo de recuperación, ya que *NT* no se basa en predicciones. Por otro lado, el segundo nivel, árbol de expansión jerárquica, soporta el mecanismo de reporte de datos y el mecanismo de prolongación de vida útil. El árbol de notificación y el árbol de expansión jerárquica están diseñados para trabajar juntos y así poder lograr su mejor desempeño en términos de vida útil de la red, precisión en el seguimiento, reducción de tráfico y reducción de retardo de transmisión. *GLT* garantiza un alto rendimiento con el soporte del árbol (*NT*) y el árbol (*HST*, por sus siglas en inglés) para agregar e informar. Los autores en [7] proporcionan una estructura jerárquica de agrupación (*GHS*, por sus siglas en inglés) para dividir los nodos en la red de sensores inalámbricos en grupos para ayudar a la computación y comunicación de la colaboración dinámica y distribuida del sistema.

Para obtener un trabajo de colaboración estable entre los sensores, los nodos deben poder organizarse en estructuras (estructura jerárquica) de modo que los objetivos de la red se alcancen utilizando el costo mínimo de comunicación.

La idea es dividir los nodos según sus regiones geográficas máximas cubiertas de manera que cada grupo contenga una serie de nodos y una serie de líderes. El impacto de este modelo se ve reflejado en (i) enrutamiento de datos y ahorro de energía, (ii) seguimiento de objetos y balance de energía y (iii) evitar la redundancia de datos. Para evaluar el rendimiento de la estructura los autores implementaron un enrutamiento basado en grupos y un seguimiento de objetos basado en grupos. La estructura de agrupación muestra un buen rendimiento en el consumo de energía y la disipación de energía durante el enrutamiento de los objetos y que se generan pocos datos redundantes durante el seguimiento de objetos. En el trabajo [3] los autores presentan un análisis y comparación de diferentes algoritmos para la localización de sensores, para la evaluación de los algoritmos, se simuló una red inalámbrica de sensores en MATLAB con un área delimitada de 100 m x 100 m en donde se distribuyeron aleatoriamente 100 sensores con rango parcial para su localización.

Una vez que cada sensor en la red estimó su posición inicial, se utilizo dicho conjunto de posiciones iniciales como punto de inicio para los diferentes algoritmos analizados. Los resultados obtenidos muestran que los algoritmos iterativos optimizados son más eficientes en precisión y número de iteraciones. Entre los algoritmos de localización analizados se encuentran: LeastSquare, Min-Max, Spatially-Constrained Local Problems, Push-Pull Estimator y Levenberg-Marquardt. TS Chen et al. [1] proponen un método de seguimiento de objetos distribuidos utilizando la estructura de la red de polígonos convexos, llamados caras, los nodos en las caras cooperan para encontrar las trayectorias de un objeto y luego estas trayectorias se utilizan para predecir el movimiento de los objetos. El método propuesto basado en la construcción del árbol de trayectoria puede reducir tanto el espacio de almacenamiento de las trayectorias recolectadas como el tiempo dedicado al análisis de predicción de trayectorias. Las simulaciones muestran que el método propuesto puede reducir el consumo de energía de los nodos y hacer una predicción de la dirección de movimiento de los nodos con precisión.

3. Desarrollo metodológico

En la literatura se han propuesto muchos protocolos para reducir el consumo de energía en las redes de sensores para el seguimiento de objetos. Además, en las redes de sensores, los modelos de agrupación, modelo de agregación de datos, modelo de activación de nodos y modelo de informe de datos juegan un papel esencial que podría maximizar la vida útil de la red [4, 11]. Con estas motivaciones, la idea que se presenta en este trabajo es de implementar GLT [6] y GHS [7] para el seguimiento de objetos, objetivos y/o eventos.

3.1. Modelo estructural

***GHS* y *GLT*:** La idea principal es dividir los nodos de la red en grupos según sus regiones geográficas máximas cubiertas de tal manera que cada grupo contenga un número de nodos y un número de líderes.

Por otro lado, *GLT* es un modelo para detectar movimientos de un objetivo utilizando la Estructura Jerárquica de agrupación [7]. Este modelo de agrupación tiene varias ventajas que hace que las redes de sensores estén libres de estrés en muchos aspectos, como el enrutamiento de datos y evitar la redundancia de datos. A continuación, se proporciona un algoritmo de agrupación distribuido (algoritmo 1, que se ejecuta dentro de la unidad de procesamiento del nodo sensor).

Se compone de dos niveles. El primer nivel, la zona de activación (Árbol de notificaciones/NT), mejora el mecanismo de activación, el mecanismo de limpieza de datos y el mecanismo de equilibrio de energía. El segundo nivel llamado Árbol de expansión jerárquica, es compatible con el mecanismo de informe de datos y el mecanismo de prolongación de la vida útil.

Definición 1 *(lista de vecinos). Para cada nodo $s_a \in S$ su lista de vecinos se define como un conjunto de nodos* $nl(S_a) = \{S_b | S_b \in S \; y \sqrt{(X_{Sa} - X_{Sb})^2 + (Y_{Sa} - Y_{Sb})^2} < r_a + r_b\}$ *donde r_a y r_b son los rangos de detección para S_a y S_b respectivamente; S es el conjunto para todos los nodos sensores en la red. nl(S)=nl(S_i — $S_i \in S$) es una matriz de listas de vecinos.*

Definición 2 *(grupo). Un conjunto de nodos, denotado como $G^k = \{s_0 \cdots, s_j\}$- $\forall s_i \in S$, se dice que es un grupo de tamaño* ***k*** *(es decir, contiene k nodos) si y solo si $G^k = \{s_0 \cdots s_k\}$ ocurre exactamente k veces como un subconjunto en la matriz nl(S). La matriz de todos los grupos en la red se denota por G^*.*

Definición 3 *(grupos asociados). Los grupos asociados para s_j son definidos como un conjunto de grupos. $G * (S_j) = \{G^k | G^k \in G^* \; y \; S_j \in G^k\}$.*

Modelo de seguimiento de objetos: Se definen tres estados para cada nodo: activo, inactivo y en reposo. En estado activo, el nodo recopila los datos sensoriales y los reporta al nodo de agregación, mientras que en estado inactivo, el nodo se prepara para rastrear le objeto y espera una activación el nodo proxy seleccionado. En el estado de reposo no se pueden recopilar datos sensoriales. Los nodos soportan dos tipos de mensajes:

- Insertar paquetes de control.
- Borrar paquetes de control.

Tabla 1. Símbolos usados en el algoritmo de agrupación.

Simbolo	Significado
S	Un conjunto de sensores.
D	Distancia euclidiana

V_i	La lista de nodos vecinos de s_i.
	Aqui llamamos a V el vector del sensor.
	$\mid V_i \mid$ Es el número de sensores dentro de V_i.
	$D = \sqrt{(x_{si} - x_{sj})^2 + (y_{si} - y_{sj})^2}$
	$\forall_{sj} \in S\, hacer:$
	{
	$D = \sqrt{(x_{si} - x_{sj})^2 + (y_{si} - y_{sj})^2}$
	$\forall_{sj} \in S\, hacer:$
	{
	$si(r_i + r_j < D)\, hacer:$
	$V_i.add(s_j)$;
	}

N_i	La lista de vectores vecinos para s_i $\forall s_j \in V_i\, hacer$ Encuentre V_i y luego agrega V_j a N_i llamando a: $N_i.add(V_j)$. Esto es encontrar el vector para cada nodo sensor s_j tal que $s_j \in V_i\, i \neq j$. El número de vectores en N_i es denotado por $\mid N_i \mid$

F_i	La lista de vectores vecinos filtrados
	para s_i. El proceso de filtrado
	se ejecuta como:
	$\forall V_a \in N_i\, hacer:$
	{
	$V_x \leftarrow null$;
	$\forall s_c \in V_a\, hacer:$
	{
	$si(s_c \in V_i) hacer:$
	{
	$V_x.add(s_c)$;
	}
	}
	$F_i.add(V_x)$
	}
	Es decir, el sensor en N_i se agregará a F_i
	si y solo si pertenecen a V_i de lo contrario
	serán ignorados. $\mid F_i \mid$ es el número
	de vectores dentro de F_i.

$G^*(s_j)$	Los grupos asociados de s_i, de modo que $G^*(s_j) = \{G^x \mid G^x \in G^*$ y $s_j \in G^x\}$.
$F_i[n]$	Obtiene el vector filtrado con el índice n.

Algoritmo 1 Algoritmo de agrupación.

Entrada: s_i, S
Salida: $G^*(s_i)$ los grupos asociados de s_i

```
1: V_i ← s_i. Vector
2: Para (int k ← 0 ; k¡|F_i|; k++) Hacer
3:    V_k ← F_i[k];
4:    s_k ← V_i[k];
5:    Si (|V_k|=2) entonces
6:       G^k ← V_k
7:       G*(s_i)+ = G^k
8:    Si no
9:       Para (int m ← k + 1; m < |F_i|; m + +) Hacer
10:         V_m ← F_i[m];
11:         Si S_k ∈ V_m entonces
12:            G^{|V_k ∩ V_m|} ← V_k ∩ V_m
13:            G*(s_i)+ = G^{|V_k ∩ V_m|}
14:         Fin Si
15:      Fin Para
16:   Fin Si
17: Fin Para
```

Los dos niveles:

- Primer nivel (Árbol de notificaciones/NT): Se debe de tener en cuenta que cada vértice es un grupo de nodos, según la Definición 2. La tarea principal de NT es evitar la redundancia y duplicación de datos. Ver algoritmo 2.
- Segundo nivel (Árbol de expansión jerárquica/HST): El mecanismo de informe de datos y el mecanismo de prolongación de vida útil son los dos objetivos principales de este nivel. El primer nivel logra el proceso de recopilación de datos.

Algoritmo 2 Construyendo el árbol de notificaciones.

Entrada: s_j
Salida: NT (s_j)

```
1: Root ← s_j;
2: G_j ← (V_j, E_j)// V_j = {s_x | s_x ∈ G*(s_j)} y E_j ⊆ [V_j]^2
3:   G_{j,i} ← G_j − s_j//{G_{j,0}, G_{j,1}, ··· , G_{j,i}} subgráfos.
4: Para (int i ← 0; i <| G_{j,i} |; i + +) Hacer
5:    T_{j,i} ← BuildSpanningTree(G_{j,i}) //construye el árbol de
   expansión para cada subgráfo
6:          Root.children.add(T_{j,i})
7: Fin Para
```

Proceso de construcción de hst: El proceso de construcción de HST es basado en NT. Por conveniencia, sea HST=$\{L_0, L_1, L_2, \cdots, L_x\}$ los niveles del

árbol, $1 \leq x \leq n/2$, donde $n = |S|$ es el número de nodos en la red. Cada nivel contiene un conjunto de nodos tal que $L_i = \{s_0, s_1, \cdots\}$. Se supone que el nodo raíz se encuentra en el primer nivel del árbol. El proceso de construcción del *HST* se extiende desde el nodo raíz hasta las hojas (de arriba hacia abajo) de la siguiente manera:

1. Para el nodo raíz, se agrega a $L_0 = \{s_r\}$, y luego para cada nodo en el árbol de notificaciones $NT(s_r) = \{s_r \rightarrow s_{r+0} \rightarrow s_{r+1} \rightarrow \cdots\}$; si s_{r+i} no es marcado, entonces se agrega s_{r+i} a L_1 y s_{r+i} será un hijo de s_r.
2. Para otros nodos en otros niveles el método de procesamiento es el mismo que el del primer nivel. La única diferencia es que el primer nivel contiene un solo nodo, es decir, la raíz.

Mecanismo de reporte de datos usando hst: El árbol de notificaciones agrega los datos y el nodo proxy comenzará el proceso de reenvío. Cualquier nodo en la ruta del proxy al nodo raíz reenvía los datos recopilados a su padre en *HST* o a su padre en *NT*. El padre en *HST* es el predeterminado. Si la energía del padre en *HST* es mayor que el valor del umbral, entonces se seleccionada el padre en *HST*. De lo contrario, se selecciona el padre en *NT* (ver algoritmo 3).

Algoritmo 3 Reporte de datos.

Entrada: Nodo informador.
Salida: Los datos del nodo informador se envían a su padre.

```
DataReport(ReporterNode, pkg)
 ParentInHST ← ReporterNode.ParentSensor;
Si (ParentInHST.BatteryPercentage > ParentInHST.Energygthreshold) entonces
   ReporterNode.SendData(ParentInHST,pkg);
   DataReport(ParentInHST, pkg); // llamada recursiva
Si no // Seleccione un nodo del NT del nodo informador
   ParentInNT ← SwitchOperation(ReporterNode,ParentInHST); // ver algoritmo
   7
   ReporterNode.SendData(ParentInNT,pkg);
   ParentInNT.ReceiveData(ReporterNode,pkg);
   DataReport(ParentInNT,pkg) // llamada recursiva
Fin Si
```

Enrutamiento de datos: Enrutamiento de datos simple basado en el modelo de agrupación. Cuando un nodo fuente tiene datos que enviar, primero comienza buscando el grupo de distancia mínima a la estación base (algoritmo 4) y luego desde el grupo seleccionado elige el siguiente salto para reenviar los datos. El siguiente salto se puede seleccionar de acuerdo con la siguiente distancia mínima al siguiente nodo objetivo o la energía máxima del nodo objetivo. Después de mantener el siguiente salto, el líder del grupo seleccionado transmitirá los datos directamente al nodo objetivo.

Algoritmo 4 Enrutamiento de datos.

Entrada: s_i es el nodo fuente
Salida: envía datos a EB directamente o reenvía datos a través de una ruta de múltiples saltos.

```
1: DataRouting(s_i, Message msg)
2: TargetGroup ← GetMinDistanceGroupFor(s_i);
3: s_t ← MinDistanceSensor(TargetGroup);//s_t es el nodo destino.
4: Si (s_t.ID ≠ SinkNode.ID) entonces
5:    s_i.SendData(s_t);
6:    s_t.ReceiveData(s_i);
7:    DataRouting(s_t, msg);
8: Si no
9:    s_i.SendData(SinkNode);
10:   SinkNode.ReceiveData(s_i);
11: Fin Si
```

Tabla de objetos detectados (SOT por sus siglas en inglés) y tabla de mensajes de notificación (NMT por sus siglas en inglés) : Para $s_j \in S$ se asume que tiene una Tabla de Objetos Detectados $SOT(s_j)$ para guardar la información del objeto detectado (es decir, posición actual), y tiene una Tabla de Mensajes de Notificación $NMT(s_j)$ para almacenar la notificación enviada/recibida-hacia/desde otros nodos. Cuando un objeto $o_i \in M$ es detectado por s_j, s_j puede crear una nueva entrada en SOT(s_j) para o_i si y solo si s_j no tiene ningún registro de notificación recibido de otro $s_x \in G^*(s_j)$. Cuando s_j crea una nueva entrada en SOT(s_j) enviará un mensaje de notificación de insertar a $G^*(s_j)$ a través del árbol de notificaciones NT(s_j) para notificar a todos los nodos en $G^*(s_j)$ y pedirles que agreguen este mensaje de notificación a sus tablas NMT; por lo tanto los otros nodos en $G^*(s_j)$ no detectarán o_i hasta que s_j envíe un mensaje de notificación de eliminación que indique que o_i salió de su rango.

En este momento, todos los nodos en $G^*(s_j)$ eliminarán los mensajes de notificación de o_i y entrarán en cualquier otro nodo en $G^*(s_j)$. La selección del nodo proxy se puede determinar en función de los registros de NMT; por ejemplo, cuando hay más de un registro para o_i en NMT(s_j) = $\{N_{i,j}, N_{i,0}, N_{i,1}, ..., N_{i,x}\}$ $x \neq j$, $N_{i,x}$ es una notificación de mensajes para el objeto o_i que es detectado por el nodo s_x. En este caso s_j será el nodo proxy si cumple los requisitos, es decir, ID máximo, energía residual, el estado actual, etc., si s_j pierde esta ronda o no fue elegido como nodo proxy, eliminara el registro de SOT(s_j) y elimina el mensaje de notificación $N_{i,j}$ de NMT (s_j). Algo que se debe de destacar es que todos los grupos $G^*(s_j)$ tendrán la misma NMT (Tabla de mensajes de notificación).

4. Desarrollo de trabajo experimental

Para la evaluación del rendimiento, se ha desarrollado un simulador en Java. El desarrollo en Java se decidió ya que es un lenguaje multiplataforma y hay

innumerables bibliotecas y/o utilidades creadas para extender su funcionalidad, además de la experiencia previa con este lenguaje. El entorno de simulación cuenta con un área de trabajo, un panel de control, un apartado para el indicador de los estados de los nodos sensores y en la parte superior un menú para crear la configuración de la simulación, como se observa en la figura 1.

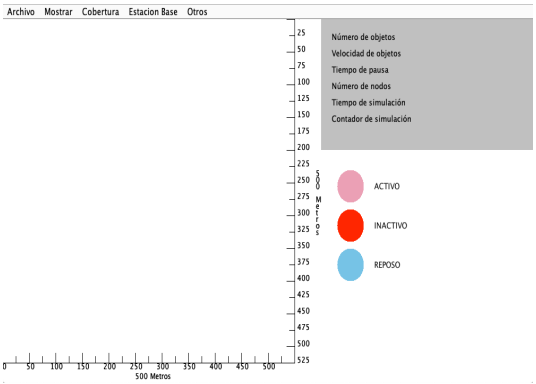


Fig. 1. Interfaz gráfica del simulador.

Las pruebas se desarrollaron con una base de datos sintética (contiene las tablas de sensores y topologías). La tabla sensores tiene los atributos de id, idTopologias, nodoID, pox, poy, poz y r (radio). La tabla de topologías tiene como atributos idTopologias y nombre (nombre de la topología).

Tabla 2. Parámetros de simulación (E_{elec}=Disipación de energía, ε_{mp}=Modelo multitrayecto del amplificador transmisor, EB=Estación base).

Parámetro	Valor
Tamaño de la red	450 x 450 m^2
Nodos	600 nodos
Radio	12 metros
Tamaño de datos	256 bits
Energía inicial	0.5 julios
E_{elec}	50 nJ/bit
ε_{mp}	0.001 pJ/bit/m^2
Localización de EB	Estática

Cobertura de Topología: Cada sensor puede recopilar datos mediante el monitoreo de un área generalmente pequeña que se encuentra en su rango de detección. Decimos que el sensor proporciona *cobertura* a esta área. Uno de los problemas fundamentales de las redes de sensores inalámbricas es la cobertura

de los objetos en conjunción con las limitaciones de eficiencia de energía [8]. El problema de cobertura se describe como un problema de calidad de servicio, donde el objetivo es encontrar qué tan bien, en términos de calidad de datos de monitoreo, el campo es monitoreado por los sensores. Por otro lado, el problema se formula como la maximización de la vida útil de la red bajo el área o la restricción de cobertura. En la primera formulación (ver figura 2 a la izquierda), los sensores deben de monitorear toda el área (por ejemplo, una gran región cuadrada), mientras que en la segunda los sensores deben cubrir un conjunto de puntos (objetivos) que yacen en el campo (ver figura 2 de la derecha).

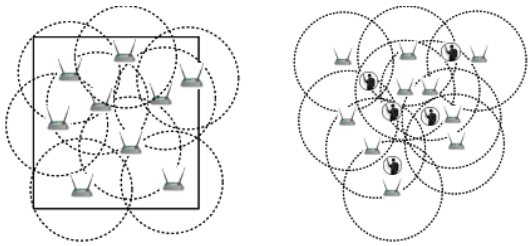


Fig. 2. Dos tipos principales de cobertura: cobertura de área y objetivo (el cuadro grande denota el área cubierta, los soldados a la derecha denotan los objetivos cubiertos.

Cobertura basada en cuadricula: Para los esquemas de cobertura basados en cuadrática, se seleccionaron dos algoritmos proporcionados en [8], Versión 1 de cobertura cuadrática y la versión 2 de cobertura cuadrática.

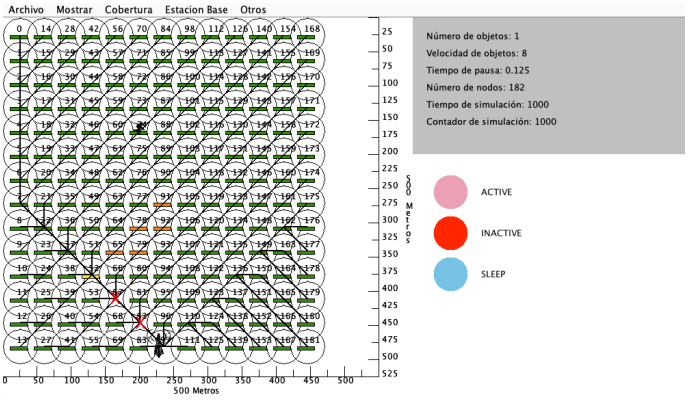


Fig. 3. Árbol de expansión jerárquica de 182 nodos desplegados utilizando el esquema de cobertura cuadrática 1. El rango de detección es de 25 metros y el rango de comunicación es de 50 metros. La barra de proceso indica la capacidad de la batería. El nodo de estación base es 97.

Cobertura basada en Zigzag: En el esquema de cobertura zigzag [5], el área de interés se divide en múltiples patrones de zigzag con múltiples esquinas y segmentos de líneas, cada nodo se despliega en una esquina del patrón de zigzag. Ver figura 4.

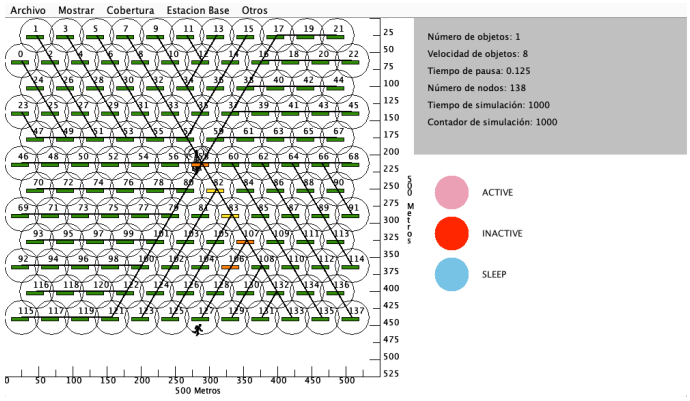


Fig. 4. Árbol de expansión jerárquica de 138 nodos desplegados utilizando el esquema en zigzag, el rango de detección es de 25 metros, mientras que el rango de comunicación es de 50 metros. La barra de proceso indica la capacidad de la batería. El nodo de estación base es 58.

Consumo de Energía: Se utiliza un modelo simple llamado "Modelo de radio de primer orden" [9]. En GLT, *HST* consume la mayor parte de la energía durante el enrutamiento de datos. *NT* consume una menor cantidad de energía.

4.1. Comparación de resultados obtenidos

Para la evaluación, se realizó un análisis cuantitativo. Esta evaluación se llevó a cabo con los parámetros descritos en la tabla 2. Además, el análisis evalúa la eficiencia haciendo énfasis a cuatro rubros en general; 1) número paquetes transmitidos, 2) consumo promedio de energía de la recolección de datos versus el número de nodos, 3) vida útil de la red y 4) consumo de energía con respecto al tiempo. La evaluación se realizó haciendo comparación con los protocolos de agrupación en clústeres existentes (CAICS por sus siglas en inglés [2]), (DLC por sus siglas en inglés [11]) y (MHC por sus siglas en inglés [14]) de acuerdo en la literatura. Los parámetros predeterminados utilizados en esta sección son los mismos que [2]. La estación base está ubicada en el centro del campo (200, 200). Las unidades de medida son el tiempo de simulación (segundos), el retardo de transmisión (milisegundos), el consumo de energía (julios) y la velocidad del objeto.

La eficiencia del envió de paquetes en el algoritmo GLT con respecto a CAIC, MHC y DLC se hace notorio, debido a que el algoritmo GLT ubica la estación

base en el centro de la detección. Cuando la estación base está ubicada en el centro del campo de detección, la vida útil de la red es más duradera y el número de paquetes de operación de conmutación se reduce.

En contraste, cuando la ubicación del receptor se encuentra a los extremos del campo de detección, la vida útil de la red se reduce considerablemente y como resultado, el número de paquetes de operación de conmutación aumentará para prolongar la vida útil de la red el mayor tiempo posible.

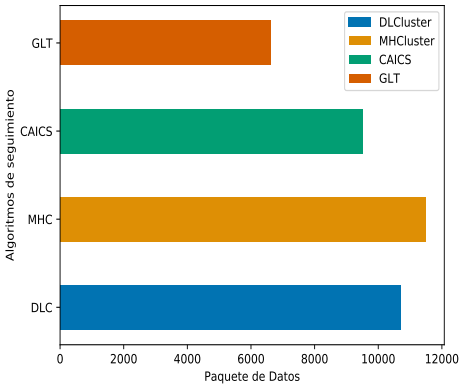


Fig. 5. Número de paquetes de datos transmitidos.

El tamaño de la red (es decir, el número de nodos en la red) tiene un impacto en el retraso de la transmisión y el consumo de energía en los cuatro enfoques (es decir, GLT, CAICS, DLC y MHC). En el caso de GLT, cuanto mayor sea el tamaño de la red, mayor será el número de grupos.

Por lo tanto, el tamaño de NT y HST será mayor. Por otro lado, en las redes basadas en clúster, el número de miembros del clúster aumentara en cada grupo, lo que conduce a más nodos involucrados en la detección y el seguimiento. El consumo medio de energía y el retraso medio de transmisión de la recopilación de datos frente al número de nodos se muestran en la figura 6.

5. Conclusiones

Las empresas en México optan por nuevas tecnologías para optimizar e innovar sus procesos y estar a la altura de los mercados internacionales. Las redes de sensores es una de las aplicaciones clave que está ampliamente adaptada debido a su gran cantidad de implementaciones y usos. Dado que el principal obstáculo que enfrenta la mayoría de las técnicas de seguimiento en redes de sensores es el hecho de que la red de sensores sufre de una fuente de alimentación muy limitada que a su vez restringe el tipo de tareas que debe realizar la red de sensores. El objetivo principal de este trabajo se cumple, ya que la durabilidad de la red y la conservación de energía son atacados con la implementación de

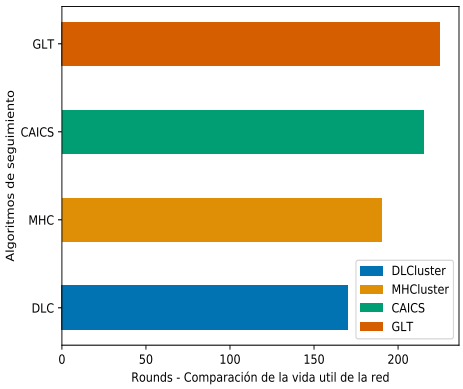


Fig. 6. Consumo promedio de energía de la recolección de datos versus el número de nodo.

los algoritmos planteados, siendo así que este trabajo se puede tomar como referencia para poner en marcha en ambientes productivos manufactureros.

Se crearon topologías para tener el enfoque de redes de sensores, los resultados de la simulación reflejan una disminución de sobrecarga de comunicación sin disminuir la precisión del seguimiento del objeto y se logra una buena extensión de la vida útil de la red. Los nodos en el árbol de notificaciones se activarán para garantizar la precisión del seguimiento, pero sus datos recopilados se transmitirán al nodo proxy en un solo salto. Para trabajos futuros se propone el estudio del movimiento rápido de los objetos.

Referencias

1. Chen, T.S., Chen, J.J., Wu, C.H.: Distributed object tracking using moving trajectories in wireless sensor networks. Wireless Networks 22(7), 2415–2437 (Oct 2016)
2. Chen, W.P., Hou, J., Sha, L.: Dynamic clustering for acoustic target tracking in wireless sensor networks. IEEE Transactions on Mobile Computing 3(3), 258–271 (Jul 2004)
3. Faudo, I.E., Cota-Ruiz, J., Aguilera, F.J.E., Perea, P.R.: Análisis y comparación de algoritmos de localización en redes inálambricas de sensores. Pistas Educativas, Instituto Tecnológico de Celaya 108 (2014)
4. Feng, J., Lian, B., Zhao, H.: Coordinated and Adaptive Information Collecting in Target Tracking Wireless Sensor Networks. IEEE Sensors Journal 15(6), 3436–3445 (Jun 2015)
5. Hawbani, A., Wang, X.: Zigzag Coverage Scheme Algorithm & Analysis for Wireless Sensor Networks. Network Protocols and Algorithms p. 19 (Dec 2013), http://www.macrothink.org/journal/index.php/npa/article/view/4688
6. Hawbani, A., Wang, X., Karmoshi, S., Kuhlani, H., Ghannami, A., Abudukelimu, A., Ghoul, R.: GLT: Grouping Based Location Tracking for Object Tracking Sensor Networks. Wireless Communications and Mobile Computing (2017)

7. Hawbani, A., Wang, X., Karmoshi, S., Wang, L., Husaini, N.: Sensors Grouping Hierarchy Structure (GHS) for Wireless Sensor Network. International Journal of Distributed Sensor Networks 11(8), 650519 (2015)
8. Hawbani, A., Wang, X., Naji, A., Karmoshi, S.: Grid Coverage Algorithm & Analyzing for wireless sensor networks. Network Protocols And Algorithms 3, 1 (Mar 2014)
9. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. pp. 10 pp. vol.2– (Jan 2000)
10. Liao, W., Chang, K., Kedia, S.P.: An object tracking scheme for wireless sensor networks using data mining mechanism. In: 2012 IEEE Network Operations and Management Symposium. pp. 526–529 (April 2012)
11. Mamun, Q., Ramakrishnan, S., Srinivasan, B.: Selecting Member Nodes in a Chain Oriented WSN. In: 2010 IEEE Wireless Communication and Networking Conference. pp. 1–6 (Apr 2010)
12. Raga, M.C.M.: Filtro de Kalman y sus aplicaciones (2018)
13. Weiser, M.: The computer for the 21 st century. Scientific American (1991)
14. Xue, W., Jun-Jie, M., Sheng, W., Dao-Wei, B.: Cluster-based Dynamic Energy Management for Collaborative Target Tracking in Wireless Sensor Networks. Sensors 7 (Jul 2007)
15. Zheng, Y., Zhou, X. (eds.): Computing with spatial trajectories. Springer, New York (2011)

ISSN 1870-4069

Revisión del estado del arte en técnicas de procesamiento de lenguaje natural para análisis de malware

Horacio Rodriguez-Bazan, Grigori Sidorov,
Ponciano Jorge Escamilla-Ambrosio

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

b160524@sagitario.cic.ipn.mx,
sidorov, pescamilla}@cic.ipn.mx

Resumen. Hoy en día, casi todas las personas en el mundo tienen al menos un dispositivo móvil, la compañía Google Inc. creó el sistema operativo Android en 2008 que se convertiría en el sistema operativo más utilizado en los dispositivos móviles de todo el mundo, sin embargo, ya que cubre más del 80 % del dispositivo móviles en el mundo se convierte también en el objetivo principal para los ciberdelincuentes, aprovechando el riesgo que existe en las aplicaciones que instalamos en nuestros dispositivos mismas que podría contener malware (del inglés, software malicioso), analizar todas las aplicaciones almacenadas en una tienda de aplicaciones antes de ponerlas a disposición de las personas sería una tarea difícil o prácticamente imposible ya que día a día salen nuevas aplicaciones al mercado. Existen investigaciones que han trabajado con este tipo de aplicaciones maliciosas para analizarlas, aplicando técnicas como Procesamiento de Lenguaje Natural PNL (del inglés, Natural Language Processing, abreviado NLP), en este artículo se presentan aquellos trabajos que aplicaron PNL, para la detención de código malicioso en las aplicaciones, analizando dichos trabajos y las técnicas utilizadas, los resultados y el trabajo futuro.

Palabras clave: Análisis de malware, PLN, Word2Vec, BoW, TF-IDF.

State of the Art Review of Natural Language Processing Techniques for Malware Analysis

Abstract. Today, almost everyone in the world has at least one mobile device, the company Google Inc. created the Android operating system in 2008 that would become the most widely used operating system on mobile devices worldwide, however, since it covers more than 80% of the mobile devices in the world, it also becomes the main target for cyber

criminals, taking advantage of the risk that exists in the applications that we install on our devices that could contain malware, analyzing all the applications stored in an application store before making them available to people would be a difficult or practically impossible task, since new applications come onto the market every day. There are investigations that have worked with this type of malicious applications to analyze them, applying techniques such as Natural Language Processing NLP. This article presents those works that applied NLP, to stop code malicious in applications, analyzing those jobs and the techniques used, the results and future work.

Keywords: Malware analysis, PLN, Word2Vec, BoW, TF-IDF.

1. Introducción

El creciente desarrollo en las comunicaciones ha hecho que la humanidad busque nuevas formas para comunicarse más rápida y efectivamente, siendo la telefonía móvil la más usada alrededor del mundo para dicho propósito, esta necesidad ha llevado a tener la mayoría de los recursos en un solo dispositivo móvil. Siendo así que la compañía Google Inc. en 2008 creó la primera versión del sistema operativo móvil, el cual se convertiría en el sistema operativo móvil más utilizado hoy en día cubriendo más del 80% del mercado de los dispositivos móviles alrededor del mundo.

Estas nuevas formas de comunicación vía Internet han traído consigo riesgos en todos los sistemas conocidos, mismos los ciberdelincuentes han aprovechado para comprometer la seguridad de los usuarios, Android no ha sido la excepción, por el contrario, al dominar la gran mayoría del mercado de dispositivos móviles se ha convertido en el objetivo preferido de los ciberdelincuentes. Por esta razón es que día a día salen el mercado una cantidad importante de aplicaciones que contienen malware (del ingles, ***MAL****icious soft****WARE***) y sería una tarea complicada analizar cada una de las nuevas aplicaciones por una analista de malware.

Hoy en día las técnicas de Inteligencia Artificial (del ingles, Artificial Intelligence, AI) han sido aplicadas a diversas áreas del conocimiento para resolver problemas que no habían sido posible resolver mediante los métodos tradicionales, gracias a los avances y desarrollos de algoritmos supervisados, así como no supervisados (Machine Learning) ha sido posible emplear estás técnicas a diversos problemas.

Dichas técnicas han sido aplicadas al área de ciberseguridad, en especial para el análisis de malware, como es el caso de estudio de este artículo, la aplicación de las técnicas de Procesamiento de Lenguaje Natural, PLN (del inglés Natural Language Processing, NLP) para el análisis de malware.

El artículo está organizado con la siguiente estructura, en la Sección 1 se mencionan los aspectos relevantes de las aplicaciones Android, Sección 2 muestra la revisión del estado del arte analizando los trabajos en los que se han empleado técnicas del Procesamiento del Lenguaje Natural, en la sección 3 se presenta un

revisión de los resultados obtenidos con las diferentes técnicas aplicadas, en la última sección 4 una recopilación de las áreas no exploradas por los investigadores y que sirven como trabajo a futuro en el análisis de malware aplicando técnicas de Procesamiento de Lenguaje Natural.

2. Aplicaciones Android - APK

Un archivo *APK* es una aplicación para el sistema operativo Android, el sistema operativo móvil de Google Inc. Algunas aplicaciones vienen preinstaladas en los dispositivos, mientras que otras aplicaciones se pueden descargar desde Google Play sitio oficial de Google o inclusive de otras tiendas de aplicaciones. Las aplicaciones descargadas de Google Play se instalan automáticamente en su dispositivo, mientras que las descargadas de otras fuentes deben instalarse manualmente.

Por lo general, para los usuarios es transparente la instalción ya que nunca ven los archivos *APK* porque Android maneja la instalación de aplicaciones en segundo plano a través de Google Play u otra plataforma de distribución de aplicaciones. Sin embargo, hay muchos sitios web que ofrecen descarga directa de archivos *APK* para usuarios de Android que desean instalar aplicaciones manualmente. En este caso, debe tener cuidado de confiar en la fuente del archivo *APK*, ya que el malware se puede distribuir en archivos *APK*.

Los paquetes o aplicaciones de Android contienen todos los archivos necesarios para una sola aplicación. A continuación, se muestra una lista de los archivos y carpetas más destacados:

- ***META-INF /***: contiene el archivo manifiesto, la firma y una lista de recursos en el archivo.
- ***lib /***: bibliotecas nativas que se ejecutan en arquitecturas de dispositivos específicos (armeabi-v7a, x86, etc.)
- ***res /***: recursos, como imágenes, que no se compilaron en resources.arsc
- ***assets /***: archivos de recursos sin formato que los desarrolladores agrupan con la aplicación
- ***AndroidManifest.xml***: describe el nombre, la versión y el contenido del archivo APK.
- ***classes.dex***: las clases compiladas en lenguaje Java que se ejecutarán en el dispositivo (archivo classes.dex)
- ***resources.arsc***: los recursos compilados, como cadenas, utilizados por la aplicación (archivo .arsc)

Los archivos *APK* se guardan en formato comprimido .zip y se pueden abrir con cualquier herramienta de descompresión. Por lo tanto, si desea explorar el contenido de un archivo *APK*, puede cambiar el nombre de la extensión del archivo a ".zip" y abrir el archivo, o puede abrir el archivo directamente a través del cuadro de diálogo abierto de una aplicación .zip [3].

3. Técnicas de procesamiento de lenguaje natural para el análisis de malware

En esta sección se presenta el análisis de los trabajos relacionados al análisis de malware aplicando las técnicas de Procesamiento de Lenguaje Natural.

Algunas técnicas para el análisis de malware proponen el uso de Sandbox (ambiente de pruebas controlado) para analizar las muestras, sin embargo, existen muestras que detectan si están siendo ejecutadas en un ambiente controlado pueden cambiar su comportamiento durante la ejecución. Es así que los investigadores proponen otro método para el análisis porque utilizar sandbox requiere de tiempo para cada una de las muestras y recolectar los resultados sin perde de vista el punto anterior que si la muestra detecta que esta siendo analizada podría cambiar su comportamiento por lo que los datos recopilados pueden no ser los correctos. Las cadenas ASCII extraídas de archivos ejecutables son útiles para analizar malware a este método se le conoce como análisis estático. Con el reciente desarrollo de técnicas de procesamiento de lenguaje natural (PLN), es posible usar estas cadenas como método de detección de malware.

Investigadores proponen un método de detección de malware que utiliza cadenas ASCII con técnicas de PLN. Dicho método divide estas cadenas en palabras y distingue la diferencia de las palabras entre archivos ejecutables benignos y maliciosos. La construcción de un modelo de lenguaje, construye un modelo de lenguaje usando técnicas de PLN. La tecnología de PLN utilizada en este tipo de métodos es LSI (del inglés, Latent Semantic Analysis) y Doc2Vec como lo proponen en esta investigación por Mohammadinodooshan et al. [4].

Doc2Vec, realiza un modelo de lenguaje para convertir las palabras en vectores de características utilizando el modelo PV-DM (del inglés, Paragraph Vector Distributed Memory) del párrafo vectorial, el entrenamiento se lleva mediante Maquinas de Soporte Vectorial (del inglés, Support Vector Machine ,SVM) con los datos de entrenamiento. Este enfoque muestra buenos resultados siempre y cuando las muestras no estén ofuscadas en ese caso este método de detección puede no funcionar como se espera, ya que depende de las cadenas para generar el modelo de espacio vectorial, al ser cadenas ofuscadas estas puede que no se detecten, en las condiciones ideales de trabajo este método tiene una efectividad Doc2Vec de 89% y LSI de 78% [4].

Las llamadas a API como entradas para clasificadores han sido exploradas por varios investigadores que emplean estos datos para sus modelos propuestos. Además, aprovechan el desarrollo en el Procesamiento del Lenguaje Natural (PLN), algunos métodos como n-gramas, doc2vec (Vectores de párrafo), TF-IDF (del inglés, Term Frequency – Inverse Document Frequency, abreviado TF-IDF) para convertir esas secuencias de API a modelo de espacio vectorial antes de alimentar a los clasificadores.

La selección de características generalmente se emplea n-gramas, para generar el vector de características para este fin se pueden emplear TF-IDF, vectores de párrafos (PV-DM y PV-DBOW). Dado el vector de características se emplean cuatro clasificadores comunes: Máquina de vectores de soporte (del inglés, Support Vector Machine, abreviado SVM), vecino K más cercano (del inglés, k-Nearest

Neighbors, abreviado K-NN), bosque aleatorio (del inglés, Random Forest, abreviado RF) y perceptrón multicapa (del inglés, Multi-Layer Perceptron, abreviado MLP).

Los resultados que se obtienen cuando se aplican estos clasificadores muestra la precisión de tres métodos usando cuatro diferentes clasificadores en caso de clasificar dos grupos de malware SVM adquiere la precisión más alta, 99.06% utilizando el método TF-IDF para vectorizar las características de malware. En caso de usando KNN (k = 3) como clasificador, el modelo de bolsa distribuida de palabras nos da la mayor precisión (98.70%) en comparación con Memoria distribuida y TF-IDF con 97.55% y 98.51%, respectivamente como muesta Tran et al. en está investigación [7].

Otro enfoque que se ha aplicado al análisis de malware es el uso de los n-gramas, con este enfoque los investigadores en lugar de utilizar secuencias de código de operación (del inglés, Operation Code, abreviado opcode), con esta técnica solo se analizan las llamadas a API extraídas de secuencias de operandos. Sin embargo, el número de llamadas API únicas es excesivo y las llamadas a API varían a medida que evoluciona el nivel API de Android. Una solución es utilizar el paquete n-gramas que se extraen desde secuencias de paquetes para representar el comportamiento de la aplicación y combinarlos con algoritmos de aprendizaje automático para construir modelos de predicción de malware.

A través de los datas decompilados se genera el código *Smali* que es usado para formar los n-gramas, calculando la frecuencia de paquetes únicos de n-gramas en todos los bloques divididos de cada aplicación, y la firma de n-gramas de una secuencia es un conjunto de todas sus subsecuencias con una longitud n. Dado que el número de n-gramas de paquetes únicos aumenta exponencialmente a medida que n aumenta, se recomendable experimentar variando n de 1 a 5.

Para comparar el rendimiento del paquete Zhang et al. emplean n-gramas con opcode n-gramas en diferentes clasificadores, se probaron cuatro diferentes algoritmos supervisados de aprendizaje automático, por ejemplo, Naive Bayes (NB), Árbol de decisión (DT), vecino k-más cercano (KNN) y aleatorio Bosque (RF), para construir modelos de clasificación.

Los resultados presentados con dichos clasificadores muestran que con n-gramas usando n entre 1 y 5, con cada uno de los clasificadores se tiene una precisión entre 95% y 98%, lo cual hace indicar que para estos datos no hay una diferencia marcada haciendo variaciones de los n-gramas [9].

Uno de los trabajos desarrollados por ElMouatez et al. es *MalDy* (mal die), en el cual se utilizan técnicas supervisadas de aprendizaje automático (Machine Learning), la idea principal de MalDy es el modelado de los informes de comportamiento en una secuencia de palabras, junto con el procesamiento avanzado del lenguaje natural (PLN) y las técnicas de aprendizaje automático (ML) para la ingeniería automática de características de seguridad relevantes para detectar y atribuir malware sin la supervisión de un analista de malware.

MalDy realiza una representación de características (palabras) haciendo ingeniería en características automáticamente sin la intervención de un experto en seguridad, MalDy propone emplear el modelo de PLN bolsa de palabras (del

inglés, Bag of Word, BoW), usando la Frecuencia de Documento de Frecuencia Inversa (TF-IDF) o Función Hashing tricking (FH) MalDy tiene dos variantes basadas en la técnica BoW elegida, ya sea TF-IDF o FH. Estas técnicas generan vectores de longitud fija a partir de informes de comportamiento utilizando frecuencias de palabras.

La evaluación de MalDy muestra la efectividad y la portabilidad de MalDy en todo el espectro de los análisis y de la configuración, las técnicas de PNL y ML para construir conjuntos de aprendizaje automático discriminativos. MalDy logra más del 94% de puntaje f1 en la tarea de detección de Android en los conjuntos de datos Malgenome, Drebin y MalDozer y más del 90% en la tarea de atribución [5].

Otro trabajo realizado en el que se propone un método resistente a la ofuscación es el propuesto por Aghamohammadi et al., llamado ORDroid, que puede detectar malware mutados y transformados, empleando redes neuronales (RNN) y procesamiento de lenguaje natural (PLN) para lograr este propósito. Adicionalmente proponen un método ligero de detección de malware, llamado LightDroid. La idea principal de este método es seleccionar un número mínimo de características del archivo Android Manifest, junto con una serie de características basadas en imágenes del archivo ejecutable Dalvik de manera que la precisión del modelo resultante esté cerca del estado del arte, mientras que su complejidad computacional sea lo más baja posible.

Los resultados muestran que LightDroid es el más ligero, con un 97,49% de precisión en los datos de prueba, mientras que ORDroid muestra que, teniendo en cuenta la precisión general de ambas pruebas y los datos transformados, dicho modelo es el mejor en comparación con los métodos más relacionados con una precisión del 98.07% en lo normal y del 93.00% en los datos transformados[1].

DroidVecDeep, un método de detección de malware de Android que utiliza una técnica de aprendizaje profundo.

Primero extraen características y las clasifican usando la impureza de disminución media. Después, transforman las características en vectores compactos basados en word2vec. Finalmente, entrenan el clasificador basado en el modelo de aprendizaje profundo.

Los datos de las aplicaciones se obtienen a través de la ingeniería inversa del APK y se extrae el archivo AndroidManifest.xml y la carpeta que contiene el código fuente smali usando Apktool. Los “Permission” e “Intent Action” se obtiene del archivo AndroidManifest.xml. Los permisos describen los permisos requeridos por la aplicación, aunque algunos de ellos serían potencialmente riesgosos, del código “smali” se obtienen algunas cadenas sensitivas que se utilizan como características.

Las características extraídas las representan como un vector de características basado en la codificación “one-hot”, con base en las investigaciones en Procesamiento de Lenguaje Natural (PLN), la codificación “one-hot” no contiene ninguna información de corpus, y la distancia entre todas las palabras es la misma. Word2vec define el vector de palabras de acuerdo con el contexto, y las palabras con alta relevancia tienen distancias más cercanas.

Word2vec utiliza dos modelos para producir una representación distribuida de palabras: el modelo continuo de bolsa de palabras (CBOW) y el modelo Skip-Gram. En este trabajo se trata la falta de extracción y caracterización de algunas características en trabajos del estado del arte, para ello evaluaron 3,000 aplicaciones benignas y 12,000 malware. Los resultados muestran que DroidVecDeep funciona bien en precisión y eficiencia de ejecución y es superior a algunas herramientas de detección maliciosas [2].

Una novedosa combinación de arquitecturas de visualización y aprendizaje profundo para un análisis híbrido entre el análisis estático y dinámico basado en el procesamiento de imágenes aplicado en un entorno de Big-Data. Es el primero de su tipo para lograr una detección inteligente de malware de día cero.

Para esta investigación, los autores utilizaron el conjunto de datos Ember con un subconjunto que contiene 70,140 archivos benignos y 69,860 maliciosos. El conjunto de datos se dividió aleatoriamente en un 60% de entrenamiento y un 40% de pruebas usando Scikit-Learn.

Los resultados se presentaron en tres enfoques para el análisis estático, dinámico y para procesamiento de imágenes. Los autores probaron el modelo de tres maneras diferentes para el análisis estático, logrando una precisión máxima del 98.9% usando el algoritmo DNN, para el análisis dinámico la precisión máxima fue del 93.6% usando el algoritmo CNN y para el procesamiento de imágenes fue hasta 96.3% usando "CNN 2 + LSTM".

En dicha investigación se tuvieron cuatro contribuciones, la primera es una nueva propuesta de un marco escalable e híbrido, ScaleMalNet facilita la recolección de muestras de malware de diferentes fuentes de forma distribuida y la aplicación de preprocesamiento de manera distribuida.

El segundo es una nueva técnica de procesamiento de imágenes para la clasificación de malware. El tercero ScaleMalNet sigue un enfoque de dos etapas, en la primera etapa el archivo ejecutable se clasifica en malware o legítimo utilizando análisis estático y dinámico y en la segunda etapa el archivo ejecutable de malware se clasifica en la familia de malware correspondiente, y la contribución de cuatro es independiente evaluación del desempeño de las arquitecturas clásicas de MLA y aprendizaje profundo, comparando varios modelos de análisis de malware [8].

Otra técnica implementando un enfoque secuencial, que incluye agrupamiento, clasificación y Blockchain. El aprendizaje automático (ML) extrae la información automáticamente del malware utilizando técnicas de agrupamiento y clasificación y almacena la información en la cadena de bloques.

La investigación tuvo dos contribuciones, la primera es que la técnica propuesta consiste tanto en tecnología Blockchain como en técnicas de aprendizaje automático para la detección de malware de dispositivos IoT (del inglés, Internet of Things), y la segunda es que el aprendizaje automático extrae automáticamente la información del malware mediante la técnica de agrupamiento y clasificación, y almacena la información en la Blockchain. Los autores propusieron el algoritmo Naive Bayes de funciones múltiples para la clasificación basada en árboles de

decisión para extraer características más importantes para proporcionar clasificación y regresión para lograr una alta precisión y robustez.

Para esta investigación, los autores utilizaron un conjunto de datos que incluye 6,192 aplicaciones de malware benignas y 5,560 recopiladas de Google Play Store y Chinese App store con este conjunto de datos, logran una precisión del 98.0% utilizando el algoritmo Naive Bayes [6].

4. Mejores resultados

Investigadores han tratado de resolver el problema del análisis de malware, como se vio en la sección anterior este malware puede ser para PC (con sistema operativo Microsoft Windows) o para Móviles (con sistema operativo Android), estos dos tipos arquitecturas son las más atacadas por malware, lo que ha representado una tarea difícil de resolver cuando se presentan muestras nuevas o de día cero (aquellas de las que no se conoce nada).

La aplicación de algoritmos supervisados y no supervisados traen consigo que se puedan clasificar en familias y conocer de qué tipo de muestra se está tratando, los resultados mostrados en esta investigación sugieren el uso se técnicas de Procesamiento de Lenguaje Natural (PLN), por ejemplo el uso de n-gramas para generar la selección de características, sin embargo, no existe una guía que indique que características son las que se deben tomar en cuenta o de qué modo, es decir, si las características se toman con base en los resultados que se obtienen al ejecutar la muestra (análisis dinámico) o mediante el código (análisis estático).

Sabemos que hoy en día con los controles de seguridad que existen los atacantes están invocando la forma de pasar estos controles de seguridad a fin de hacer llegar al usuario estas aplicaciones con malware, los dos enfoques tanto el análisis dinámico como el estático han dado buenos resultados al introducirlos en los clasificadores más comunes Naive Bayes, KNN, RF, DT y RNN por citar algunos, en la mayoría de ellos se alcanza una precisión por arriba del 95% en la mayoría de los casos. Lo cual hace pensar que se tiene el problema resuelto para el análisis de malware, sin embargo, estos algoritmos se han probado en corpus conocidos.

Para nuevos desarrollos de métodos de análisis de malware al menos se deben considerar estos algoritmos como base para el desarrollo de alguna nueva forma de detección, ya que se ha observado que son los más usados por los investigadores gracias a los resultados que dan mediante la combinación de técnicas de Procesamiento de Lenguaje Natural con algoritmos de Machine Learning.

Table 1 muestra en resumen los trabajos que se revisaron como estado del arte, donde ellos muestran los algoritmos y técnicas de Procesamiento de Lenguaje Natural aplicados, como se observa, en todos los casos no se aplican las mismas técnicas, sin embargo obtienen resultados alentadores en cada uno de ellos, los resultados dependen de las características que están empleando en los modelos, lo que hace alusión es que no hay una forma única de elegir las características, pero sí que existen métodos para elegir las caracterizas más optimas.

Table 1. Tabla comparativa de algorítmos y resultados.

Herramienta	Algoritmos PLN	Clasificadores	Plataforma	Resultados
A [4]	LSI	SVM	PC	LSI 78.0%
	Doc2Vec			Doc2Vec 89.0%
B [7]	n-gramas	SVM	PC	SVM - TF-IDF 99.06%
	Doc2Vec	KNN		KNN - PV-DBOW 98.70%
	TF-IDF	RF		KNN - PV-DM 97.55 %
	TF-IDF	MLP		KNN - TF-IDF 98.51%
C [9]	n-gramas	NB	PC	n-gramas[1-5] 95% a 98%
		DT		
		KNN		
		RF		
D [5]	BoW	KNN	Android	KNN 59.2%
	TF-IDF	RF	PC	RF 87.00%
	FH	SVM		SVM 87.71%
E [1]	Word2Vec	RNN	Android	98.07%
F [2]	Word2Vec	DBN	Android	97.29%
	CBoW	SVM		96.40%
	Skip-Gram	NB		92.30%
G [8]	Scikit-Learn	DNN	Android	98.9%
		CNN		93.6%
		CNN2 + LSTM		96.3%
H [6]	Word2Vec	KNN	Android	92.0%
		BN		98.0%
		DBN		87.0%

5. Trabajo a futuro

Durante la revisión del estado del arte, se identificaron algunas áreas que no han sido del todo exploradas, aplicando las técnicas de Procesamiento de Lenguaje Natural para el análisis de malware en Android.

– Análisis estático, aplicando procesamiento de lenguaje natural en los archivos clave de las aplicaciones como lo son *AndroidManifest.xml*, *classes.dex*, *resources.arsc*, ya ellos pueden ser procesados como texto para generar un modelo de espacio vectorial que permita conocer que tan similares son unas aplicaciones de otras o que caracteristicas se comparten, se debe tener en cuenta los métodos de ofuscación de Android dado que el texto para analizar no sería el mismo.

– Análisis dinámico, con base en los datos generados mediante la ejecución de la aplicacion en un ambiente controlado recolectar dichos datos y aplicar técnicas de procesamiento de lenguaje natural a fin de tener un modelo de espacio vectorial que incluya otras características adicionales a las llamadas a sistema API.
– Análisis hibrido, con base en los datos generados en los dos puntos anteriores (análisis estático y dinámico), generar un nuevo modelo de espacio vectorial que permita incluir ambos resultados a fin de tener las dos formas del análisis y que se puedan aplicar los algoritmos de Machine Learning con base en los dos criterios.
– Probar dichos modelos con otros corpus a fin de medir que tan eficiente son estos modelos cuando se enfrentan a datos desconocidos.
– Desarrollo de un método basado en aplicación de técnicas de lenguaje natural y aprendizaje automático para detectar autoría de códigos maliciosos ofuscados en aplicaciones Android, a fin de tener otra característica que se pueda incluir en un modelo de análisis de malware para robustecer los modelos de detección.
– Evaluación del método (experimentos con aplicación de este método con aplicaciones Android de autoría diversa).

En resumen, este articulo mostro las técnicas de procesamiento de leguaje natural aplicadas al análisis de malware, como se mostró en esta revisión aún existen técnicas que se pueden explorar para analizar dichas muestras para general modelos de detección que permitan trabajar con diferentes corpus, que puedan ser entrenados y probados con cualquier corpus, es decir un modelo que pueda detectar cualquier malware y que además obtenga una alta precisión.

Agradecimientos. Agradecemos a CONACYT, SNI, IPN (SIP, COFAA), apoyo de proyectos SIP 20200859 y 20200797 y Conacyt A1-S-47854.

References

1. Aghamohammadi, A., Faghih, F.: Lightweight versus obfuscation-resilient malware detection in android applications. Journal of Computer Virology and Hacking Techniques pp. 125–139 (2019)
2. Chen, T., Mao, Q., Lv, M., Cheng, H., Li, Y.: Droidvecdeep: Android malware detection based on word2vec and deep belief network. KSII Transactions on Internet and Information Systems 13(4), 2180–2197 (2019)
3. Hoog, A.: Android Forensics: Investigation, Analysis and Mobile Security for Google Android. Syngress Publishing, 1st edn. (2011)
4. Ito, R., Mimura, M.: Detecting unknown malware from ascii strings with natural language processing techniques. In: 2019 14th Asia Joint Conference on Information Security (AsiaJCIS). pp. 1–8 (2019)
5. Karbab, E.B., Debbabi, M.: Maldy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports. Digital Investigation 28, S77–S87 (2019)

6. Kumar, R., Zhang, X., Wang, W., Khan, R.U., Kumar, J., Sharif, A.: A multimodal malware detection technique for android iot devices using various features. IEEE Access 7, 64411–64430 (2019)
7. Tran, T.K., Sato, H.: Nlp-based approaches for malware classification from api sequences. In: 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES). pp. 101–105 (2017)
8. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Venkatraman, S.: Robust intelligent malware detection using deep learning. IEEE Access 7, 46717–46738 (2019)
9. Zhang, P., Cheng, S., Lou, S., Jiang, F.: A novel android malware detection approach using operand sequences. In: 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC). pp. 1–5 (2018)

ISSN 1870-4069

Enfoque semántico de políticas para gestionar la conciencia de grupo en Groupware

Mario Anzures-García, Luz A. Sánchez-Gálvez

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{mario.anzures, sanchez.galvez}@correo.buap.mx

Resumen. En el dominio de *Groupware,* la interacción es esencial entre los actores (usuarios) y los recursos compartidos, para alcanzar la meta común propuesta. La conciencia de grupo debe estar presente en este contexto, ya que permite a cada actor ser consciente de lo que hacen otros en el espacio compartido. Consecuentemente, esta conciencia debe ser dinámica, adaptarse a las necesidades cambiantes del grupo, detallarse de manera formal y explícita. Por ello, se requiere un enfoque basado en políticas para ajustarse continuamente a las condiciones internas y externas del sistema. Por tanto, este artículo propone un enfoque semántico de políticas para gestionar la conciencia de grupo; que está apoyado en una ontología, para describir, representar y adecuar las políticas de una manera explícita y formal. Este enfoque plantea políticas de Persona, Labor y Lugar para, gestionar adecuadamente la conciencia de grupo en *Groupware*. Finalmente, un caso de estudio se presenta para aplicar el enfoque propuesto.

Palabras clave: Conciencia de grupo, groupware, ontología, políticas.

Policies Semantic Approach for Managing the Group Awareness on Groupware

Abstract. In the Groupware domain, interaction is essential between actors (users) and shared resources, to achieve the proposed common goal. Group awareness must be present in this context, since it allows each actor to be aware of what other's users are doing, and so, they know what happens in the shared workspace. Consequently, this awareness must be dynamic, adapt to the changing needs of the group, and be detailed formally and explicitly. Thus, a policy-based approach is required to continually adjust the system to the constraints and conditions internal and external. Consequently, this paper proposes a policy ontological approach for managing group awareness, since ontology establishes a vocabulary expressive enough to describe, represent, and modify policies in an explicit, and formal way. This approach defines policies of Person, Labor y Place to manage the group awareness. Finally, a case study to apply proposed approach is presented.

Keywords: Group awareness, groupware, ontology, policy-based approach.

1. Introducción

El uso de políticas es adecuado para desarrollar sistemas complejos y dinámicos, porque separan la funcionalidad de los mismos de las reglas que los gobiernan, así, se pueden cambiar las políticas para adaptar los sistemas a las necesidades y dinámicas del contexto. Sin embargo, el modelado de sistema basados en políticas, necesita un enfoque ontológico, para satisfacer la simplicidad, la expresividad, el análisis, la escalabilidad y la exigibilidad requeridos para especificarlas.

La conciencia de grupo proporciona un mejor conocimiento sobre las actividades individuales y estado del grupo. Además, la conciencia de las tareas y actividades influye en la coordinación y el desempeño de las tareas de manera positiva [1]. Por ello, la conciencia de grupo es un requisito cuando se desarrollan sistemas que deben soportar la colaboración de manera efectiva, de acuerdo a [2]. En consecuencia, existen diversos trabajos relacionados con la conciencia de grupo: el *framework* de la Teoría de la Conciencia de Grupo [3], los cuatro tipos de conciencia [4], la extensión de Teoría de la Conciencia de Grupo [5], GAKS (*Group Awareness Knowledge-based System*) [6], el enfoque formal basado en roles [7], el uso de ontologías para la representación del contexto y la conciencia de grupo en sistemas colaborativos [8, 9], así como propuestas de mecanismos para dicha conciencia en el área de la salud [10], dispositivos móviles [1, 11], videojuegos multijugador [12], interfaces de usuario [2] y widgets [13].

Las propuestas sobre conciencia de grupo se fundamentan en teorías, enfoques formales e informales, y se han aplicado en distintas áreas. Sin embargo, se echa en falta el uso de políticas, sustentadas en ontologías, para gestionar la conciencia de grupo. Por tanto, este trabajo propone un enfoque ontológico de políticas para gestionar la conciencia de grupo en Groupware, simplificando su adaptación y dinamismo a las necesidades cambiantes del grupo. La ontología se crea con Protégé y se hace uso del razonador, Pellet. Dos aportaciones, esencialmente, son realizadas en este trabajo:

1. **El modelo para soportar el enfoque.** Se presenta un esquema del enfoque que soporta las políticas que regulan la conciencia de grupo, suministrando una perspectiva general de la estructura y de las relaciones de sus componentes.

2. **El enfoque semántico de políticas**. Se propone un enfoque de políticas de la conciencia de grupo, basado en una ontología, Así como se muestra su aplicación en un groupware, denominado Espacio Virtual Académico (EVA).

El documento tiene la siguiente estructura: Sección 2 explica el marco teórico que fundamenta nuestra propuesta; describiendo el *Groupware*, la conciencia de grupo, los sistemas basados en políticas y las ontologías. Sección 3 presenta el enfoque semántico de políticas para gestionar la conciencia de grupo. Sección 4 detalla las conclusiones y el trabajo futuro.

2. Marco teórico

En esta sección, se presentan los fundamentos teóricos que sustentan el enfoque ontológico de políticas para gestionar la conciencia de grupo en Groupware.

2.1. Groupware

Es una aplicación basada en computadora que permite a un grupo de personas alcanzar una meta común, a través de una interfaz de entorno compartido [14]; tanto para escenarios profesionales (reuniones planeadas o de negocios) como de ocio (reuniones espontáneas para compartir o intercambiar experiencias). El desarrollo de *Groupware* considera cinco aspectos principales [14, 15]:

- **Entorno.** Espacio compartido donde trabaja el grupo que refuerza el aspecto de cooperación entre sus miembros y promueve la conciencia de grupo, permitiendo observar lo que realiza el resto de participantes. El entorno es una interfaz de usuario compartida, que contiene [14]: *La Vista de información*, que presenta los objetos colaborativos y las operaciones sobre éstos; *la Vista de participantes*, que facilita la comunicación persona-a-persona, suministrando widgets, audio, vídeo, etc, para simplificar la ***conciencia de grupo*** (conocer qué personas están participando y qué están haciendo); y la *Vista de Contexto*, denominada memoria o historial de grupo, que proporciona el material histórico del trabajo de grupo, útil para alcanzar de manera efectiva el objetivo común.

- **Organización.** Establece la división de trabajo, mediante la distribución de responsabilidades, para simplificar la interacción y alcanzar el objetivo común. Usando: *protocolos* (reglas de comportamiento), *estrategias* (plan de acción); y *dinámica de grupo* (que condiciona y afecta los objetivos del grupo).

- **Comunicación.** Los elementos que la caracterizan son los actores, la información compartida y el medio utilizado para tal efecto. La comunicación en su dimensión temporal, se clasifica en síncrona (mismo tiempo) y asíncrona (diferente tiempo).

- **Colaboración.** Denota un grado de participación mayor de los actores; mediante su interacción expresa en actividades conjuntas, compartiendo información y utilizando los recursos para tal efecto. En la colaboración intervienen [14]: *Actores*, usuarios del grupo, que desempeñan diferentes roles; *Roles*, que determinan un patrón de comportamiento asignado al actor, condicionando su actividad en el sistema con permisos sobre qué tareas pueden realizar; *Tareas*, conjunto de actividades encaminadas a la consecución de un objetivo; y *Recursos*, datos u objetos que se comparten para alcanzar la meta común.

- **Coordinación.** La coordinación requiere de una planificación y sincronización de las actividades efectuadas por los miembros del grupo, por ello, se deben definir dependencias temporales entre las actividades [14] y utilizar mecanismos de concurrencia [16], atenuando las condiciones de competencia y garantizando el uso mutuamente exclusivo de los recursos compartidos.

Tabla 1. Elementos de la Conciencia de Grupo Actual [3].

Categoría	Elementos de la conciencia de grupo	Preguntas específicas
¿Quién?	Presencia	¿Hay alguien en el espacio de trabajo?
	Identidad	¿Quién está participando? ¿Quién es ese?
	Autoría	¿Quién está haciendo eso?
¿Qué?	Acción	¿Qué están haciendo?
	Intención	¿Cuál es el objetivo de esa acción?
	Artefacto	¿En qué objeto están trabajando?
¿Dónde?	Ubicación	¿Dónde están trabajando?
	Mirada	¿Dónde están mirando?
	Vista	¿Dónde pueden ver?
	Alcance	¿Dónde pueden llegar?

Tabla 2. Elementos de la Conciencia de Grupo Pasada [3].

Categoría	Elementos de la conciencia de grupo	Preguntas específicas
¿Cómo?	Historial de Acción	¿Cómo ocurrió esa operación?
	Historial de Artefacto	¿Cómo ese artefacto tiene ese estado?
¿Cuándo?	Historial de Eventos	¿Cuándo sucedió ese evento?
¿Quién? (pasado)	Historial de Presencia	¿Quién estuvo ahí y cuándo?
¿Dónde? (pasado)	Historial de Ubicación	¿Dónde ha estado una persona?
¿Qué? (pasado)	Historial de Acción	¿Qué hizo una persona?

2.2. Conciencia de grupo

Es la percepción sobre lo que cada actor desarrolla, proporcionando el conocimiento que se tiene sobre lo que está sucediendo en el entorno. Por tanto, dicha conciencia es esencial para reducir las pérdidas de trabajo, así como proveer la comunicación, coordinación, y colaboración entre sus miembros.

Como se mencionó en la sección 1, varios modelos se han propuesto, los cuales han aplicado distintos mecanismos para suministrar conciencia de grupo, buscando el mismo objetivo: poner a disposición de los miembros del grupo información de este, sus actividades (actuales, pasadas o futuras) y su estado, para ayudarlos a coordinar mejor sus propias acciones teniendo en cuenta metas grupales y situacionales [17].

El enfoque semántico se fundamenta en los trabajos [3, 4]. El primero define diferentes elementos, validándolos mediante un conjunto de preguntas (véase la Tabla 1) y estableciendo elementos referentes a acciones pasadas (véase la Tabla 2). El segundo propone cuatro tipos de conciencia [4]: informal, quién está y qué hace; social, información que un actor tiene de los demás en un contexto social; estructural,

conocimiento de roles y responsabilidades de actores; y conocimiento del espacio de trabajo, interacciones de otros y sus recursos. Además, nuestra propuesta toma en consideración los trabajos relacionados con enfoque formales [7], en particular, con ontologías [8, 9] y aquellos que muestran la aplicación de mecanismos de conciencia de grupo en diferentes áreas [10-13].

2.3. Desarrollo de sistemas basados en políticas

La gestión de sistemas basada en políticas separa la política de la implementación del sistema; proporcionando la flexibilidad y autonomía necesarias para regular sistemas complejos, determinando su comportamiento y modificándolo a fin de ajustar sus componentes o sistema, sin hacer cambios sustanciales en su estructura o entorno [22]. Este enfoque ha incluido áreas tradicionales, como seguridad, manejo de recursos, intercambio y difusión de información [22]. Ya que proporciona muchos beneficios: comportamiento autónomo, reusabilidad, eficacia, extensibilidad, sensibilidad al contexto, verificabilidad, soporte a componentes, protección de un mal diseño, errores, o componentes dañinos, y razonamiento del comportamiento del sistema. *Groupware* se ha centrado en las políticas de coordinación como *control de acceso* [16], *control de concurrencia* [16, 20] y *floor control* —asignación de permisos [21]; y de seguridad (necesarias para definir acciones a realizar cuando ocurre algún problema).

En la literatura, múltiples enfoques de especificación de políticas se han propuesto: uso de lenguajes formales, notación basada en reglas, y entradas de una tabla con múltiples atributos. Estos enfoques son restrictivos en muchos aspectos, un enfoque se debe elegir por las características del dominio de aplicación y los requisitos de especificación [22]: *Simplicidad*, *Expresividad*, *Análisis*, *Escalabilidad* y *Exigibilidad*. Por tanto, una ontología es ideal para especificar políticas, ya que gestiona sistemas; simplificando análisis y reutilización, reduciendo incoherencias y conflictos [22, 23]:

- Simplifica la tarea de gobernar el comportamiento de entornos complejos.
- Utiliza conceptos para expresar el dominio y entidades; simplificando su descripción, facilitando un profundo análisis y cuidadoso razonamiento.
- Facilita el acceso a la información de la política, para calcular dinámicamente las relaciones entre las políticas y el entorno.

2.4. Ontología

Una ontología es "la especificación explícita de una conceptualización" [23]. Una *conceptualización* es un modelo abstracto de cómo la gente piensa sobre algo del mundo real, determinada por objetos, conceptos, otras entidades existentes en un área y sus relaciones. *Especificación explícita* significa que los conceptos y las relaciones de un modelo abstracto reciben nombres y definiciones explícitas. Una ontología describe formalmente e inequívocamente las nociones relevantes de un dominio; facilitando el acceso basado en el contenido, comunicación, e integración a través de diferentes sistemas; asignando un significado preciso a los datos almacenados. Cuatro elementos se identifican para una ontología: *Clases, Relaciones, Axiomas e Instancias*.

En este trabajo, la ontología se específica con OWL 2.0 [23, 24], una extensión de OWL, que aborda sus limitaciones de expresividad y requisitos de desempeño; utilizando lógica descriptiva con propiedades computacionales deseables [24]. OWL 2.0 se basa en la lógica descriptiva SROIQ(D) [25], extensión de SHOIN, con un conjunto de constructos expresivos que lo hacen más útil en la práctica. También existen herramientas que automatizan la construcción de ontologías, como: Ontolingua server [26], WebOnto [27], OilEd [28], OntoSaurus [29], Protégé [30], TopBraid Compositor [31], WebODE [32], OntoEdit [33]. Este artículo, se centra en Protégé, que se usa ampliamente para crear ontologías, debido a su escalabilidad y extensibilidad; simplificando la inferencia de conocimiento a través de razonadores (en este trabajo se utiliza Pellet), lenguajes de consulta y reglas.

3. Enfoque semántico de políticas para la conciencia de grupo

En la Fig. 1, se presenta el esquema general del enfoque semántico de políticas para gestionar la conciencia de grupo. Este esquema parte de las Tablas 1 y 2, proponiendo agrupar los elementos de la conciencia de grupo en **Persona** (correspondientes a la pregunta ¿Quién?), **Labor** (relacionados con la pregunta ¿Qué?) y **Lugar** (referidos a la pregunta ¿Dónde?). Esta agrupación permite el establecimiento de tres políticas, que sustentan el enfoque, aquí propuesto:

- **PoP.** La **Po**lítica de **P**ersona está integrada de los componentes Actor y Rol, derivados de la literatura de Groupware y descritos en la Sección 1, denominada introducción. La conciencia relacionada con la persona se refiere a la presencia e identidad de la misma en el entorno, así como a que tiene la autoría de alguna labor que se está llevando a cabo.

- **PoL.** La **Po**lítica de **L**abor está integrada de los componentes Tarea, Recurso y Operación, derivados de la literatura de Groupware, los dos primeros descritos, también, en la introducción. El tercero será explicado en la siguiente sección. La conciencia respecto a la labor tiene que ver con la Tarea u Operación que lleva a cabo una persona, utilizando un Recurso existente en el Groupware.

- **PoG.** La **Po**lítica de lu**G**ar está integrada de los componentes Entorno, Grupo, Fase e Interface de usuario, derivados de la literatura de Groupware y, descritos, también, en la introducción; con excepción de Fase, que se toma de acuerdo a [34]. La consciencia de lugar se corresponde con la labor que hace una persona en el Entorno, Grupo, Fase e Interface de Usuario.

Además, cada uno de estos componentes presentan un widget, es decir, un elemento de conciencia de grupo (Véase la Fig. 1), que facilitará tener una perspectiva pasada, presente y futura de lo que ocurre en el Groupware. Los componentes mencionados para cada política PoP, PoL y PoG, constituyen la ontología de las políticas para gestionar la conciencia de grupo, desarrollada en Prótegé y utilizando el razonador Pellet, para verificar la consistencia de la misma.

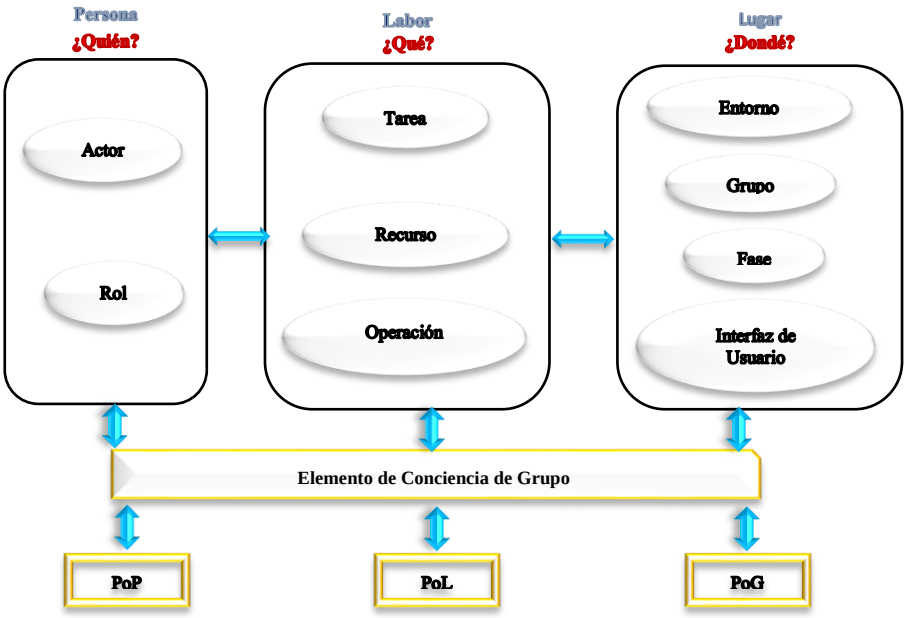


Fig. 1. Enfoque semántico de políticas para gestionar la conciencia de grupo.

3.1. Enfoque formal de políticas

En la Fig. 1, se muestra el enfoque semántico de políticas y las relaciones entre sus elementos; dicho enfoque gestiona dinámicamente las políticas de la conciencia de grupo y establece dos elementos:

- **Simples:** A, R, T, Rec, Op, E, G, F, IU, ECG, P; para Actor, Rol, Tarea, Recurso, Operación, Entorno, Grupo, Fase, Interfaz de Usuario, Elemento de Conciencia de Grupo, Política; respectivamente.

- **Compuestos:** AsigR, AsigT, AsigRec, AsigOp, AsigE, AsigG, AsigF, AsigIU, EleA, EleR, EleT, EleRec, EleOp, EleE, EleG, EleF, EleIU; para Asignación de Rol, Asignación de Tarea, Asignación de Recurso, Asignación de Operación, Asignación de Entorno, Asignación de Grupo, Asignación de Fase, Asignación de Interfaz de Usuario, Establecer ECG de Actor, Establecer ECG de Rol, Establecer ECG de Tarea, Establecer ECG de Recurso, Establecer ECG de Operación, Establecer ECG de Entorno, Establecer ECG de Grupo, Establecer ECG de Fase y Establecer ECG de Interfaz de Usuario; respectivamente.

Los elementos simples y compuestos se definen formalmente (por razones de espacio, se presentan sólo algunas definiciones de éstos), de acuerdo a [35].

- **Actor (A).** A representa un conjunto de actores. Un actor es un participante en el trabajo en grupo, que puede ser una persona (usuario de un sistema), un conjunto de éstas o un elemento del sistema, que realiza una función dentro del mismo:

$$A = \{a_1, a_2, \dots, a_n\}. \tag{1}$$

- **Rol (R).** R representa un conjunto de roles. Un rol es la función (ocupación o puesto) que un actor desempeña para alcanzar un objetivo común:

$$R = \{r_1, r_2, \dots, r_k\}. \tag{2}$$

- **Tarea (T).** T representa un conjunto de tareas. Una tarea es la función (ocupación o puesto) que un actor desempeña para alcanzar un objetivo común:

$$T = \{t_1, t_2, \dots, t_l\}. \tag{3}$$

- **Recurso (Rec).** Rec representa un conjunto de recursos. Un recurso es un dato u objeto utilizado en las tareas realizadas por el rol para alcanzar el objetivo:

$$Rec = \{rec_1, rec_2, \dots, rec_m\}. \tag{4}$$

- **Operación (Op).** Op representa un conjunto de operaciones. Una operación es la acción que se realiza sobre un recurso:

$$Op = \{op_1, op_2, \dots, op_o\}. \tag{5}$$

- **Entorno (E).** E representa un conjunto de entornos. Un entorno es el espacio de trabajo compartido proporcionado por el groupware:

$$E = \{e_1, e_2, \dots, e_p\}. \tag{6}$$

- **Grupo (G).** G representa un conjunto de grupos. Un grupo es un conjunto de actores que realizan una tarea común:

$$G = \{g_1, g_2, \dots, g_q\}. \tag{7}$$

- **Fase (F).** F representa un conjunto de fases. Que es un momento de colaboración. Una Fase se constituye de una o más Tareas y se presenta en un Grupo:

$$F = \{f_1, f_2, \dots, f_w\}. \tag{8}$$

- **Interfaz de Usuario (IU).** IU representa un conjunto de interfaces de usuario. Una interfaz de usuario es la función (ocupación o puesto) que un actor desempeña para alcanzar un objetivo común:

$$IU = \{iu_1, iu_2, \dots, iu_s\}. \tag{9}$$

– **Elemento de Conciencia de Grupo (ECG).** ECG representa un conjunto de elementos de la conciencia de grupo. Un ECG es una etiqueta, fotografía, fecha, etc., que identifica a cada componente que participa en el trabajo en grupo, proporcionando la conciencia de grupo:

$$ECG = \{ecg_1, ecg_2, \dots, ecg_t\}. \tag{10}$$

– **Política de Persona (PoP).** PoP es una relación de asignación de roles a usuarios de la conciencia de grupo, que es un mapeo de muchos a muchos entre usuarios y roles, cuando se cumplen un conjunto de reglas:

$$PoP \sqsubseteq A \times R \times Reg. \tag{11}$$

– **Política de Lugar (PoL).** PoL es una relación de la labor que realizan el Actor determinado la conciencia de grupo, cuando se cumplen un conjunto de reglas:

$$PoL \sqsubseteq A \times R \times T \times Rec \times Op \times Reg. \tag{12}$$

– **Política de Lugar (PoG).** PoG es una relación del lugar donde está el Actor, determinado la conciencia de grupo al cumplirse un conjunto de reglas:

$$PoL \sqsubseteq A \times R \times E \times G \times IU. \tag{13}$$

– **Política (P).** P representa un conjunto de políticas, que incluye Políticas de Persona (PoP), Políticas de Labor (PoL) y Políticas de Lugar (PoG):

$$P = PoP \sqcup PoL \sqcup PoG. \tag{14}$$

– **Asignación de Rol (AsigR).** AsigR es una relación de asignación de muchos a muchos entre Actor y Rol. Cada Actor podría desempeñar muchos Roles, y cada Rol podría ser desempeñado por muchos Actores:

$$AsigR \sqsubseteq A \times R. \tag{15}$$

– **Asignación de Recurso (AsigRec).** AsigRec es una relación de asignación de muchos a muchos entre Rol y Recurso. Cada Rol podría utilizar muchos Recursos, y cada Recurso podría ser utilizado por muchos Roles:

$$AsigRec \sqsubseteq R \times Rec. \tag{16}$$

– **Asignación de Operaciones (AsigOp).** AsigOp es una relación de asignación de muchos a muchos entre Operación y Recurso. Cada Operación se asigna a muchos Recursos, y cada Recurso estar asignado a muchas Operaciones:

$$AsigOp \sqsubseteq Op \times Rec. \tag{17}$$

Tabla 3. Enfoque semántico de políticas.

ESeP	ECoG	Política
PoP	*Presencia*	Sea $a_i \in A, i = 1, \dots, n \wedge g_h \in G, donde\ h = 1, \dots, q \wedge \{a_i, g_q\} \nexists AsigG \rightarrow G = \emptyset$ Sea $a_i \in A, i = 1, \dots, n \wedge g_h \in G, donde\ h = 1, \dots, q \wedge \{a_i, g_q\} \in AsigG \rightarrow G \neq \emptyset$
	Identidad	Sea $a_i \in A, i = 1, \dots, n \wedge g_h \in G, h = 1, \dots, q \wedge ecg_z \in ECG, z = 1, \dots, t \wedge \{a_i, g_q\} \in AsigG \rightarrow \{(a_i, a_j), g_q\} \in AsigG$ Sea $a_i \in A, i = 1, \dots, n \wedge r_l \in R, l = 1, \dots, k \wedge g_h \in G, h = 1, \dots, q \wedge \{a_i, r_l\} \in AsigR \wedge ecg_z \in ECG, z = 1, \dots, t \wedge \{a_i, g_q\} \in AsigG \wedge \{ecg_z, r_l\} \in AsigR \rightarrow \{(a_i, a_j), g_q\} \in AsigG \wedge \{\exists r_h \vert \{r_h, a_j\} \in AsigR \wedge \{ecg_t, r_h\} \in EleR\}$
	Autoría	Sea $a_i \in A, i = 1, \dots, n \wedge t_h \in T, h = 1, \dots, q \wedge \{a_i, t_h\} \in AsigT \rightarrow \{\exists ecg_m \vert \{ecg_m, a_i\} \in EleA\} \wedge \{\exists \mathrm{ecg_m} \vert \{\mathrm{ecg_m}, \mathrm{t_i}\} \in \mathrm{EleT}\} \wedge a_i \in A$
PoL	*Acción*	Sea $op_j \in Op, i = 1, \dots, o \wedge a_i \in A, i = 1, \dots, n \wedge r_h \in R, h = 1, \dots, k \wedge rec_l \in R, l = 1, \dots, m \wedge \{a_i, r_h\} \in AsigT \wedge \{op_j, rec_l\} \in AsigOp \rightarrow \{\exists ecg_m \vert \{ecg_m, op\} \in EleOp\} \wedge op \in Op$
	Inten_ ción	Sea $a_i \in A, i = 1, \dots, n \wedge t_h \in T, h = 1, \dots, q \wedge \{a_i, t_h\} \in AsigT \rightarrow \{\exists ecg_m \vert \{ecg_m, a_i\} \in EleA\} \wedge \{\exists \mathrm{ecg_m} \vert \{\mathrm{ecg_m}, \mathrm{t_i}\} \in \mathrm{EleT}\} \wedge t_h \in T$
	Arte_ facto	Sea $a_i \in A, i = 1, \dots, n \wedge r_h \in R, h = 1, \dots, k \wedge rec_l \in R, l = 1, \dots, m \wedge \{a_i, r_h\} \in AsigR \wedge \{r_h, rec_l\} \in AsigRec \rightarrow \{\exists ecg_y \vert \{ecg_y, rec\} \in EleRec\}$
PoG	*Ubica_ción*	Sea $a_i \in A, i = 1, \dots, n \wedge g_h \in G, h = 1, \dots, q \wedge e_j \in E, j = 1, \dots, p \rightarrow \{\exists t \in T \vert \{a_i, t\} \in AsigT\} \wedge \{a_i, g_q\} \in AsigG \wedge \{a_i, e_j\} \in AsigE$
	Mirada y Vista	Sea $a_i \in A, i = 1, \dots, n \wedge iu_j \in IU, j = 1, \dots, s \rightarrow \{\exists t \in T \vert \{a_i, t\} \in AsigT\} \wedge \{a_i, iu_j\} \in AsigIU\}$
	Alcance	Sea $a_i \in A, i = 1, \dots, n \wedge f_j \in F, j = 1, \dots, s \rightarrow \{\exists t \in T \vert \{a_i, t\} \in AsigT\} \wedge \{a_i, f_1\} \in AsigF\}$

– **Establecer ECG del Actor (EleA).** EleA es una relación de asignación de muchos a muchos entre Actor y ECG. Cada ECG podría ser asignado a muchos Actores y cada Actor podría asignarse a muchos ECG:

$$EleA \sqsubseteq ECG \times A. \tag{18}$$

3.2. Políticas para conciencia de grupo

El Enfoque Semántico de Políticas (ESeP) de la conciencia de grupo (véase la Tabla 3), se establece a partir del esquema general del enfoque semántico de políticas y de la definición formal establecida en la sección anterior, 3.1:

– **PoP.** La primera política de Persona (Tabla 3) está relacionada con el elemento de Conciencia de Grupo (ECoG) denominado *Presencia*, de la

pregunta ¿Hay alguien en el espacio de trabajo? y se refiere a los Actores. La segunda política (Tabla 3) concierne a *Identidad*, indica las preguntas ¿Quién está participando? y ¿Quién es ese?, los conceptos son Actor y Rol. La última (Tabla 3) atañe al elemento *Autoría*, de la pregunta ¿Quién está haciendo eso? y corresponde al trabajo del Actor. Además, para contestar a la pregunta ¿Quién estuvo ahí y cuándo? se almacenan los sucesos relacionados a los elementos de *Presencia, Identidad y Autoría*.

- **PoL.** La primera política de Labor (Tabla 3) se relaciona con el ECoG denominado *Acción*, de la pregunta ¿Qué están haciendo? y se refiere a Operación. La segunda política (Tabla 3) concierne a la *Intención*, de la pregunta ¿Cuál es el objetivo de esa acción? y es la Tarea que se realiza. La última (Tabla 3) atañe al elemento *Artefacto*, de la pregunta ¿En qué recurso están trabajando? y corresponde al Recurso utilizado. Además, para contestar a ¿Qué ha estado haciendo una persona?, ¿Cómo ocurrió esa operación? y ¿Cuándo sucedió ese evento? se almacenan los elementos de *Acción, Intención y Artefacto;* y se agrega un ECG del tiempo.
- **PoG.** La primera política de Grupo (Tabla 3) está relacionada con el ECoG denominado *Ubicación*, de la pregunta ¿Dónde están trabajando? y se refiere a los componentes del ESeP: Entorno y Grupo. La segunda política (Tabla 3) concierne al elemento *Alcance*, de la pregunta ¿Dónde puede llegar?, y es Fase. La última (Tabla 3) atañe a los elementos *Mirada y Vista*, de las preguntas ¿Dónde están mirando? y ¿Dónde pueden ver?, que corresponde a la Interfaz de Usuario. Además, para contestar a ¿Dónde ha estado una persona? y ¿Cómo ese artefacto tiene ese estado? se almacenan los elementos de *Ubicación, Mirada, Vista y Alcance*.

3.3. Aplicación de las políticas

El escenario del enfoque semántico será un Groupware, denominado EVA (Espacio Virtual Académico), que se centra en administrar cursos (véase la Fig. 2). El profesor puede crear uno o más grupos de acuerdo a los cursos que esté impartiendo, en cada uno puede cargar material del curso, descargar la tarea y responder preguntas a los estudiantes, mediante la publicación de mensajes, sobre un tema particular del curso. El estudiante ingresa a EVA y puede buscar, registrarse y unirse a un grupo (o a los que esté cursando), así como también puede descargar material, cargar tarea y responder o hacer preguntas o comentarios sobre algún tema del curso. Para dicho escenario se plantean tres contextos, el primero para la pregunta ¿quién?, el segundo para ¿qué? y el último para ¿dónde?:

- **Contexto 1 del Escenario, ¿Quién?.** La estudiante Coral ingresa al grupo correspondiente al curso de Tópicos Selectos de la Computación. Ella necesita saber ¿quién está ahí?, la presencia. Además, tendría que distinguir, al profesor de sus compañeros. Esto implica ser consciente, de la identidad de quién está en dicho curso, que rol está desempeñando ese actor y de qué tarea éste es autor; de esta manera, podrá interactuar adecuadamente con cada uno. También sería de utilidad, que supiera cuantas veces han ingresado los demás actores a dicho grupo, por ello, se necesita llevar un historial del

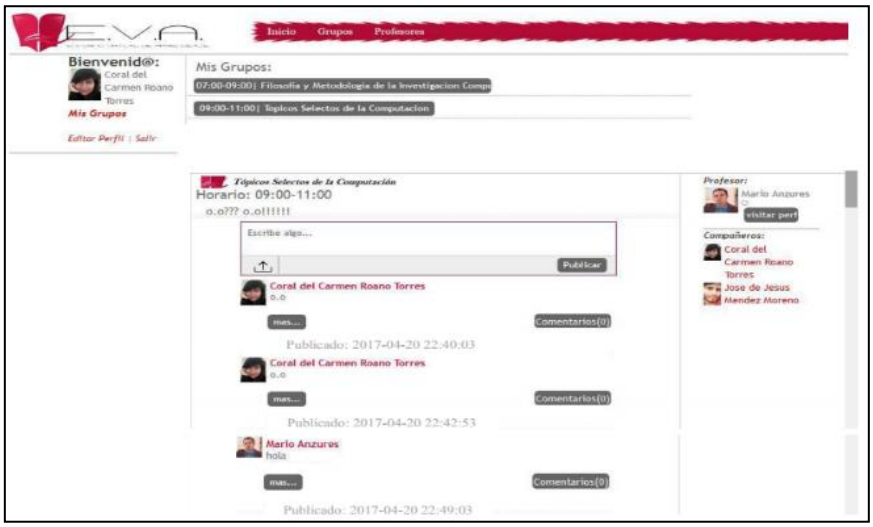


Fig. 2. Escenario de aplicación: Groupware EVA.

grupo. Por tanto, se determinan los elementos: a1= Coral Roano Torres; a2= Mario Anzures García: a3= José de Jesús Hernández Moreno; r1= Profesor; r2= Estudiante; g1= Tópicos Selectos de la Computación (TSC); g2= Filosofía y Metodología de la Investigación Computacional; t_1=Publicar; ecg_1=Profesor; ecg_2=Compañero; ecg_3=Coral; ecg_4=Mario, ecg_5=José y ecg_6=Públicado:[Fecha]. En la Tabla 4 se aprecian esos elementos juntos con la ECoG y parte de la ontología que gestiona la política de conciencia de grupo.

- **Contexto 2 del Escenario, ¿Qué?.** Coral ingresa al grupo y desea saber (intención) que dudas o comentarios (acción) han puesto en el grupo (artefacto) y quién las puso. Así, sí la duda ya fue contestada, podría hacer otra pregunta, agradecer la respuesta o pedir le aclaren la misma. También, ella, tal vez requiera consultar alguna respuesta de inicios del curso o anterior, por ello, es necesario, que se cuente con un historial. En consecuencia, se consideran los elementos de ESeP del **Contexto 1** junto con: op_1=Escribir publicación (EP), rec_1=cuadro de texto (CT) y ecg_7=Coral. En la Tabla 4 se observan dichos elementos juntos la ECoG y parte de la ontología que gestiona la política de conciencia de grupo.
- **Escenario 3, ¿Dónde?.** Coral necesita estar consciente ¿a qué grupo ha ingresado?) (ubicación), ¿dónde descarga el material, sube las tareas, escribe comentarios, etc.? (mirada y vista). Así como, consultar el material subido por el profesor, las tareas entregadas, y todos los comentarios hechos (alcance). En este último escenario. se consideran los elementos del **contxto 1 y 2,** así como IU=Grupo y Fase=Curso. En la Tabla 4 se observan dichos elementos juntos a los elementos de la ECoG y parte de la ontología, que gestiona la política de conciencia de grupo.

Tabla 4. Aplicación de políticas en los escenarios.

ECoG	Política	Ontología
Presencia	Sea $a_1, a_2, a_3 \in A \wedge g_1, g_2 \in G \wedge \{a_1,g_1\} \in AsigG \wedge \{a_2,g_1\} \in AsigG \wedge \{a_3,g_1\} \in AsigG \rightarrow G \neq \emptyset \wedge \{(a_1,a_2,a_3),g_1\} \in AsigG$	Actor{Coral}***pertenece****Grupo*{TSC}∧ Actor{Mario}***pertenece****Grupo*{TSC}∧ Actor{José}***pertenece****Grupo*{TSC}→TSC≠ ∅ ∧{(Coral,Mario, José),TSC}∈AsigG
Identidad	Sea $a_1,a_2 \in A \wedge r_1, r_2 \in R \wedge g_1, g_2 \in G \wedge ecg_1,ecg_2 \in G \wedge \{a_1,r_2\} \in AsigR \wedge \{a_2,r_1\} \in AsigR \wedge \{a_1,g_1\} \in AsigG \wedge \{a_2,g_1\} \in AsigG \wedge \{ecg_1,r_1\} \in AsigECG \wedge \{ecg_2,r_2\} \in AsigECG \rightarrow \{ (a_1,a_2,a_3), g_1\} \in AsigG \wedge \{\exists r_1,r_2 / \{(a_1,a_3),r_2\} \in AsigR \wedge \{a_2,r_2\} \in AsigR \wedge \{ecg_1,r_1\} \in AsigECG \wedge \{ecg_2,r_2\} \in AsigECG\}$	Actor{Coral}***desempeña****Rol*{Estudiante} ∧ Actor{Coral}***pertenece****Grupo*{TSC}∧ Actor{Mario}***desempeña****Rol*{Profesor} ∧ Actor{Mario}***pertenece****Grupo*{TSC}∧ Rol{Estudiante}***tiene***ECG{Compañero} ∧ Rol{Profesor}***tiene***ECG{Profesor} → {(Coral,Mario),TSC} ∧ { Rol{Estudiante}, ECG{Compañero}} ∧ {Rol{Profesor}, ECG{Profesor}}
Autoría	Sea $a_1, a_2, a_3 \in A \wedge t_1, t_2 \in T \rightarrow \{\exists ecg_3, ecg_4, ecg_6/\{(ecg_3,ecg_4,ecg_5),a_i\} \in EleA\} \wedge \{\exists ecg_6/\{(ecg_6,a_1\} \in EleT\} \wedge \{\exists ecg_6/\{(ecg_6,t_1\} \in EleT\}$	Actor{Coral}***realiza****Tarea*{Publicar} ∧ Actor{Coral}***tiene***ECG{Compañero} ∧ *Tarea*{Publicar***}tiene***ECG{Pbulicado:"*fecha*"}→Actor {Coral}***realiza*** *Tarea*{Publicar} ***tiene***ECG{Publicado:2017-04-22-40-03}
Acción	Sea $op_1 \in Op \wedge a_1,a_2,a_3 \in A \wedge r_1,r_2 \in R \wedge t_1 \in T \wedge \{a_1,r_2\} \in AsigR \wedge \{a_2,r_1\} \in AsigR \wedge \{a_3,r_2\} \in AsigR \wedge \{a_1,t_1\} \in AsigT \wedge \{a_2,t_1\} \in AsigT \wedge \{rec_1,r_1\} \in AsigRec \wedge \{rec_1,op_1\} \in AsigOp \rightarrow \{\exists ecg_3, ecg_4, ecg_6/\{ecg_6,op_1\} \in EleOp\}$	Actor{Coral}***realiza****Tarea*{Publicar} ∧ ***realiza****Tarea*{Publicar}***usa****Recurso*{CT} ∧ ***usa****Recurso*{CT}***aplica****Op*{EP}∧ *OP*{Op***}tiene***ECG{Pbulicado:"*fecha*"}→ Actor{Coral}***realiza****Tarea*{Publicar}***usa****Recurso*{CT} ***aplica****Op*{EP}***tiene***ECG{Publicado:2017-04-22-40-03}
Intención	Sea $a_1 \in A \wedge t_1 \in T \wedge \{a_1,t_1\} \in AsigT \rightarrow \{\exists ecg_6/\{ecg_6,t_1\} \in EleT\} \wedge \{\exists ecg_7/\{ecg_7,a_1\} \in EleA\}$	Actor{Coral}***realiza****Tarea*{Publicar} ∧ ***realiza****Tarea*{Publicar}***tiene***ECG {Publicado:2017-04-22-40-03} ∧ Actor{Coral}***tiene***ECG{Coral} → Actor{Coral}***tiene***ECG{Coral} ***realiza*** *Tarea*{Publicar}***tiene***ECG{Publicado:2017-04-22-40-03}
Artefacto	Sea $a_1 \in A \wedge r_2 \in R \wedge ,rec_1 \in Rec \wedge \{a_1,r_2\} \in AsigR \wedge \{rec_1,r_2\} \in AsigRec \rightarrow \{\exists ecg_7/\{ecg_7,rec_1\} \in EleRec\}$	Actor{Coral}***desempeña****Rol*{Estudiante} ∧ *Rol*{Estudiante}***usa****Recurso*{CT} → Actor{Coral}***desempeña****Rol*{Estudiante***usa****Recurso*{CT}***tiene***ECG{Publicado:2017-04-22-40-03}
Ubicación	Sea $a_1 \in A \wedge g_1 \in G \wedge \{a_1,g_1\} \in AsigG \rightarrow \{\exists t_1/\{a1,t_1\} \in AsigT \wedge \{a_1,g_1\} \in AsigG]$	Actor{Coral}***pertenece****Grupo*{TSC}→ Actor{Coral}***realiza****Tarea*{Publicar}***sellevaacaboen****Grupo*{TSC}
Mirada y Vista	Sea $a_1 \in A \wedge iu_1 \in IU \wedge \{a_1,iu_1\} \in AsigIU \rightarrow \{\exists t_1/\{a1,t_1\} \in AsigT \wedge \{a_1,iu_1\} \in AsigIU\}$	Actor{Coral}***colaboraen****IU*{Grupo}→ Actor{Coral}***realiza****Tarea*{Publicar}***elaboradaen****IU*{ Grupo}
Alcance	Sea $a_1 \in A \wedge f_1 \in F \wedge \{a_1,f1\} \in AsigF \rightarrow \{\exists t_1/\{a1,t_1\} \in AsigT \wedge \{a_1,f_1\} \in AsigF$	Actor{Coral}***trabajaen***Fase{Curso}→ Actor{Coral}***realiza****Tarea*{Publicar}***creadaen***Fase{Curso}

Los tres contextos del escenario planteado permitieron aplicar (véase la Tabla 4) las políticas especificadas (véase la Tabla 3); estableciendo que éstas:

- Se adaptan al dinamismo inherente en Groupware: 1) En un grupo pueden no existir actores, uno o más; debido a que éstos entran y salen de EVA. 2) Una tarea puede realizarse en igual o diferente tiempo, sobre un mismo o

distinto recurso; debido a la necesidad de los actores para alcanzar el objetivo común. 3) Un actor puede trabajar en el entorno, grupo, fase o interfaz de usuario, dependiendo del rol que estén desempeñando.

- Influyen en la coordinación y el desempeño de las tareas de manera positiva, como se expresó en [1]. Por ejemplo. se puede apreciar con el escenario presentado eobre el Groupware EVA, para que Coral colabore eficazmente durante su acceso a EVA, necesita información sobre la identidad (Contexto 1), ubicación (Contexto 3), actividades (Contexto 2), estado (Contexto 2) y más de los compañeros de su grupo (los tres contextos).
- Se establecen con los elementos necesarios para obtener la conciencia de grupo relacionada con Persona (PoP), Labor (PoL) o LuGar (PoG); así como definen un elemento de la conciencia de grupo para cada política; permitiendo que cada actor tenga un conocimiento de lo que sucede en todo momento en el Groupware.

4. Conclusiones y trabajo futuro

En este artículo, se ha presentado un enfoque semántico de políticas para gestionar la conciencia de grupo en Groupware; de tal manera que, se logra simplificar su adaptación y dinamismo a la naturaleza y necesidades cambiantes del grupo. Tres políticas fueron especificadas mediante una ontología, que están relacionadas con la conciencia de la persona (Política de Persona, PoP), Labor (Política de Labor, PoL) y lugar (Política de luGar, PoG). Dichas políticas están basadas en la revisión de la literatura, y particularmente, en las preguntas que establecen los elementos que constituyen la conciencia de grupo. De esta forma, se ha formulado una política para cada pregunta y planteado un escenario sobre el Groupware EVA; que ha servido para aplicar dichas políticas. Además, el modelo elaborado permitió comprender los elementos del enfoque semántico y las relaciones existentes entre éstos; proporcionando una vista estática, funcional y dinámica de tal enfoque. El trabajo futuro se centrará en especificar una arquitectura que sustente el enfoque semántico de políticas para gestionar la conciencia de grupo; así como generar una propuesta semiautomática de los widgets que deberían presentarse al usuario.

Referencias

1. Rupprecht, F.A., Kasakow, G., Aurich, J.C., Hamann, B.: Improving collaboration efficiency via diverse networked mobile devices. Journal on Multimodal User Interfaces, 12(2), pp. 91–108 (2018)
2. Gallardo, J., Bravo, C., Molina, A.I.: A framework for the descriptive specification of awareness support in multimodal user interfaces for collaborative activities. Journal on Multimodal User Interfaces, 12(2), pp. 145–159 (2018)
3. Gutwin, C., Greenberg, S.: A descriptive framework of workspace awareness for real-time groupware. CSCW Journal, 11, pp. 411–446 (2002)

4. Gutwin, C., Greenberg, S., Roseman, M.: Workspace awareness in real-time distributed groupware: framework, widgets, and evaluation. In: People and computers XI, Proceedings of the HCI (1996)
5. Tam, J., Greenberg, S.: A framework for asynchronous change awareness in collaborative documents and workspaces. I. J. Mac. St. 64(7), pp. 583–598 (2006)
6. Decouchant, D., Escalada-Imaz, G., Martinez, A.M., Mendoza, S., Muhammad, A.: Contextual awareness based communication and coauthoring proximity in the internet. Exp. Sys. Appl., 36, pp. 8391–8406 (2009)
7. Zhu, J., Tang, Y., Li, J., Liu, H.: A role hierarchical group awareness model and its time possibility analysis based fuzzy time. In: Proc. 13th Int. Conf. on Computer Supp. Cooper. Work in design, pp. 38–43 (2009)
8. Vieira, V., Tedesco, P., Salgado, A.C.: Towards an ontology for context representation in groupware. In: Fuks, H., Lukosch, S., Salgado, A.C. (eds.) CRIWG, LNCS 3706, Springer, pp. 367–375 (2005)
9. Gallardo, J., Molina, A.I., Bravo, C., Redondo, M.A., Collazos, C.A.: An ontological conceptualization approach for awareness in domain-independent collaborative modeling systems: Application to a model-driven development method. Expert Systems with Applications, 38, pp.1099–1118 (2011)
10. Branco, F., Gonçalves, R., Martins, J., Bessa, J., Baptista, A.: Computer Supported Cooperative Work - Exploratory Study on CSCW and Groupware Technologies and its Applicability in the Health Area. In: Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, M. Mendonça Teixeira (eds.), New Advances in Inf. Syst. and Techn., Advances in Int. Syst. and Comp., 445, Springer, pp. 379–389 (2016)
11. Wang, W., Reani, M.: The rise of mobile computing for Group Decision Support Systems: A comparative evaluation of mobile and desktop. Int. J. of Human-Computer Studies, 104, pp.16–35 (2017)
12. Wuertz, J., Alharthi, S.A., Hamilton, W.A., Bateman, S., Gutwin, C., Tang, A., Hammer, J.: A design framework for awareness cues in distributed multiplayer games. In: Proc. Conf. Human Factors in Comp. Syst., ACM, 243(14), pp. 1–243 (2018)
13. Blichmann, G., Meißner, K.: Customizing workspace awareness by non-programmers. In: Proc. of the ACM SIGCHI Symp. on Engineering Interactive Computing Systems, ACM, pp. 123–128 (2017)
14. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: some issues and experiences. Communications of the ACM, 34(1), pp. 39–58 (1991)
15. Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M., Paderewski, P.: Tutorial function groupware based on a workflow ontology and a directed acyclic graph. IEEE Latam Trans., 16(1), pp. 294–300 (2018)
16. Ellis, C.A., Gibbs, S.J.: Concurrency control in groupware systems. In: proceedings ACM SIGMOD Int. Conf on Management of Data, pp. 399–407 (1989)
17. Pinheiro, M.K., Souveyet, C.: Is group-awareness context-awareness? Converging Context-Awareness and Awareness Support. IJeC 15(3), pp. 1–19 (2019)
18. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. LNCS, Springer-Verlag, pp. 18–38 (2001)

19. Shen, H., Dewan, P.: Access control for collaborative environments. In: Proceedings of the ACM Conference on CSCW, pp. 51–58 (1992)
20. Greenberg, S., Marwood, D.: Real time groupware as a distributed system: In: Concurrency Control and Its Effect on the Interface ACM CSCW, pp. 207–217 (1994)
21. Dommel, H.P., Garcia-Luna-Aceves, J.J.: Floor control for multimedia conferencing and collaboration. ACM Multimedia 5(1), pp. 23–38 (1997)
22. Suri, N., Uszok, A.: Semantic web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. LNCS 2870, pp. 419–437 (2003)
23. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Int. J. Human Computer Studies, 43(5–6), pp. 907–928 (1995)
24. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological engineering with examples from the areas of knowledge management, e-Commerce and the Semantic Web. Springer (2004)
25. Kazakov, Y.: SRIQ and SROIQ are Harder than SHOIQ, descriptions logics. In: Proc. 11th Int. Conf. on Principles of Knowledge Rep. and Reasoning (2008)
26. Farquhar, A., Fikes, R., Rice, J.: The ontolingua server: a tool for collaborative ontology construction. Int. J. H-C Studies 46(6), pp. 707–727 (1997)
27. Dominguez, J., WebOnto, T.: Discussing, browsing, and editing ontologies on the web. In: Gaines, B.R., Musen, M.A. (eds.), Int. Workshop on Knowledge Acquisition, Modeling and Management, pp. 1–20 (1998)
28. Benchhofer, S., Horrocksm I., Goble. C., Stevens, R.: A reasonable ontology editor for the Semantic Web. In: Baader, F., Brewka, G., Eiter, T., LNAI 2174, Springer-Verlag, pp. 396–408 (2001)
29. Swartout, B., Ramesh, P., Knight, K., Russ, T.: Toward distributed use of largescale ontologies. In: Farquhar, A., Gruninger, M., Gómez-Pérez, A., Uschold, M., van der Vet, P. (eds.), AAAI'97 Spring Symp. Ont. Eng., pp. 138–148 (1997)
30. Musen, M.A.: An ontology-development platform for biomedical scientists. Stanford University (2000)
31. Horridge, I., Bechhofer, S.: The OWL API: a java API for OWL ontologies. Seman. Web J., pp. 11–21 (2011)
32. Prieto, A.E., Lozano, A.: An ontological approach to the hierarchical adaptation of workflows. IEEE LatinAmerican Transactions, 13(7), pp. 2278–2284 (2015)
33. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: collaborative ontology engineering for the semantic web. In: Horrocks I, Hendler JA (eds.) 1th Int. Sem. Web C. (ISWC'02), LNCS 2342, pp. 221–235 (2002)
34. Anzures-García, M., Sánchez-Gálvez, L.A.: PROMISE; proposing an ontological model for developing collaborative SystEms. Journal Intelligent & Fuzzy Systems, pp. 1–13 (2020)
35. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer, 29, pp. 38–47 (1996)

ISSN 1870-4069

Clasificación binaria de neuropatía diabética implementando el operador *PS*-Merge

América Velasco-Cétera[1], Eddy Sánchez-DelaCruz[1], Pilar Pozos-Parra[2]

[1]Tecnológico Nacional de México, Departamento de Posgrado e Investigación, Mexico
[2]Universidad Autónoma de Baja California, México

{amecetera,eddsacx}@gmail.com, maria.pozos@uabc.edu.mx

Resumen. En el campo de la inteligencia artificial, algunos algoritmos se diseñan para que, a partir de datos, se desempeñen métodos predictivos o descriptivos con la finalidad de ayudar a los expertos en la toma de decisiones. Este tipo de método de la inteligencia artificial es conocido como aprendizaje automático, existen métodos clásicos, los cuales son utilizados en la estadística con procesos complejos y sofisticados, principalmente para la clasificación y la regresión como: los algoritmos bayesianos, la regresión, la agrupación, los árboles de decisión y las redes neuronales, siendo esta última la más utilizada, para su implementación se requiere de un experto que programe los algoritmos considerando que se necesita ajustar gran variedad de parámetros para que un sistema aprenda correctamente. En esta investigación se realizó una clasificación binaria utilizando PS-Merge, un operador empleado para la fusión de creencias (conocimiento) y poco empleado en el aprendizaje automático. Los resultados mostraron una precisión superior al 80 % de instancias correctamente clasificadas de casos de neuropatía diabética, empleando tres nociones basadas en lógica proposicional: fusión de creencias, revisión de creencias y la noción de satisfacción parcial, lo que permitió combinar información resultante de diferentes fuentes contradictorias, durante el proceso no se requiere ajustar parámetros, además, se puede trazar las causas de los resultados, logrando así un aprendizaje automático explicable.

Palabras clave: Aprendizaje automático, clasificación, *PS*-Merge.

Binary Classification of Diabetic Neuropathy implementing the *PS*-Merge operator

Abstract. In the field of artificial intelligence, some algorithms are designed so that, from data, predictive or descriptive methods are performed

in order to help experts in decision making. This type of method of artificial intelligence is known as automatic learning, there are classic methods, which are used in statistics with complex and sophisticated processes, mainly for classification and regression as: Bayesian algorithms, regression, grouping, decision trees and neural networks, the latter being the most used, for its implementation requires an expert to program the algorithms considering that you need to adjust a variety of parameters for a system to learn correctly. In this research a binary classification was made using PS-Merge, an operator used for the fusion of beliefs (knowledge) and little used in automatic learning. The results showed an accuracy higher than 80 % of correctly classified instances of diabetic neuropathy cases, using three notions based on propositional logic: belief fusion, belief review and the notion of partial satisfaction, which allowed combining information resulting from different contradictory sources, during the process it is not required to adjust parameters, in addition, it is possible to trace the causes of the results, thus achieving an explainable automatic learning.

Keywords: Machine learning, classification, *PS*-Merge.

1. Introducción

Los trastornos que afectan al sistema nervioso se llaman neuropatías y generalmente afectan a las extremidades inferiores (piernas y pies), pero varían desde dolor y entumecimiento en los pies hasta problemas con la función de los órganos internos, como el corazón y la vejiga. Las personas diabéticas que no cuidan adecuadamente su control glucémico, corren el riesgo de desarrollar una neuropatía diabética, que es el resultado de la diabetes y afecta la forma en que las personas caminan, que conduce a una marcha alterada y a un deterioro progresivo [17]. El número de diabéticos en el mundo se estima en 285 millones, y se espera que alcance los 438 millones para 2030 [22].

Según las estimaciones de la Organización Mundial de la Salud, 422 millones de adultos en todo el mundo tenían diabetes en 2014, en comparación con 108 millones en 1980. En la población adulta, esta proporción aumentó de 4.7 % a 8.5 % [4]. La neuropatía diabética es el tipo más común de complicaciones microvasculares diabéticas, siendo una de las causas más común de amputación, falla autonómica, morbilidad y mortalidad [13].

Por otro lado, la Inteligencia Artificial (IA) constituye uno de los campos interdisciplinarios y transfronterizos donde convergen muchas ciencias. Se han introducido programas basados en IA que ayudan a los médicos a tomar decisiones [14], la mayoría basados en aprendizaje automático. Para procesar datos, construir e investigar sistemas que pueden aprender de los conjuntos de datos de entrenamiento y mejorar el proceso de clasificación y predicción [8]. Esta subdisciplina de la IA cada vez está más presente en tecnologías del entorno cotidiano, mediante técnicas que permiten reducir la intervención humana en la toma de decisiones.

Dado que la clasificación se usa más en el aprendizaje automático [12], se han propuesto diferentes algoritmos, tales como: funciones lineales, árboles de decisión, redes neuronales, clasificadores bayesianos, por mencionar algunos. Sin embargo, estos enfoques clásicos involucran funciones que los expertos deben calibrar manualmente, por ejemplo, el concertar gran variedad de parámetros y pesos, los cuales deben estar bien configurados para hacer que funcione correctamente el sistema.

Ahora, en lógica, se emplean técnicas de fusión que utilizan conocimiento, creencias o metas de un grupo de sujetos, así como creencias o metas de cada miembro del grupo. La *fusión de creencias* es una técnica basada en la lógica proposicional, que permite combinar información potencialmente conflictiva de diferentes fuentes definidas en un conjunto de bases de creencias inconsistentes, para generar una base de conocimiento única y consistente [6]. En cambio, *la revisión de creencias* se basa en el principio de separar el conocimiento de la información de diferentes fuentes para cambiar el estado del conocimiento en un marco lógico [19], o como una agregación de nuevo conocimiento en una base de conocimientos (KB) ya establecida [10], al cambiar el conocimiento original lo menos posible y manteniendo la consistencia de la KB. *La satisfacción parcial* es otra forma de medir la satisfacción de una fórmula.

Este concepto permite tener un valor de satisfacción en [0,1]. La explicación intuitiva para la satisfacción parcial indica que: es natural pensar que, si tiene una combinación de dos literales y sólo una está satisfecha, el agente estará satisfecho al 50 %. Al resumir esta idea, la satisfacción de la conexión de uno o más literales se puede medir en función de la suma de la interpretación de conjunciones y la suma de literales [1]. Este trabajo se inspira en [9], donde se usó una variante del operador PS-Merge para diagnosticar el cáncer oral. Los resultados muestran una precisión de más del 75 %, sin tener que ajustar parámetros en el conjunto de datos, determinando que el operador es adecuado para aplicaciones y tecnologías de aprendizaje automático en conjuntos de datos médicos.

Actualmente se carece de implementaciones con operadores que no utilicen la estadística y optimización sin la necesidad de hacer configuraciones en parámetros o iteraciones. En esta investigación, se realizó una implementación del PS-Merge [1], un operador de fusión relativamente nuevo y poco conocido, basado en la noción de satisfacción parcial, que combina la fusión de creencias y la revisión de creencias, aplicado a un conjunto de datos de la marcha de pacientes que padecen neuropatía diabética, para una clasificación binaria {sano, enfermo}. En la literatura las técnicas de *fusión de creencias*, *revisión de creencias* y *la noción de satisfacción parcial* se han utilizado por separado, éste es el primer intento utilizando PS-Merge en casos de neuropatía diabética, por lo que los resultados de esta investigación se plantean originales en el área.

La distribución del documento es la siguiente: la sección 2 presenta trabajos relacionados, en la sección 3 se presentan los materiales y métodos, la sección 4 presenta los resultados y la discusión. Por último, la sección 5 presenta conclusiones y trabajos futuros.

2. Trabajos relacionados

El uso del aprendizaje automático basado en algoritmos de IA para sistemas de diagnóstico y análisis ha guiado importantes mejoras en la predicción de enfermedades, en donde los sistemas pueden identificar si un paciente sufre una enfermedad de acuerdo a condiciones presentes, al proporcionar suficientes datos no sólo se puede detectar una enfermedad con la misma precisión que un patólogo humano sino que se puede anticipar a detectar en el paciente signos que puedan conducir a una enfermedad. No obstante, las tres nociones que utiliza PS-Merge, mantienen avances significativos como:

En [18], formalizaron el concepto de juego de revisión de creencias (BRG) para modelar la dinámica de las creencias de un grupo de agentes. Señalando un conjunto de propiedades que abordan varios temas de preservación, así como la capacidad de respuesta, la monotonicidad y la convergencia. Agregaron varias clases de políticas de revisión que se basan en la fusión de creencias de los operadores, considerando un caso en que todos los agentes utilizan la misma política de revisión. Desarrollaron un software que consiste en una interfaz gráfica que permite jugar con los BRGs considerando cualquiera de las 18 políticas de revisión de $\{ R^k_\Delta \mid k \in \{ 1, ..., 6 \}, \Delta \in \{ \Delta^{d_D,\Sigma}, \Delta^{d_H,\Sigma}, \Delta^{d_H,Gmin} \} \}$.

En [5] presentaron una solución genérica al problema de la revisión de los marcos de argumentación que se aplica a muchas argumentaciones prominentes de I-máximo semántica, con el fin de probar un teorema de representación para dos tipos de revisión. Combinando avances de argumentación y el cambio de creencias, es decir, tomaron conceptos de realizabilidad en la argumentación y la noción de cumplimiento como se utiliza en la revisión de Horn, los resultados son genéricos en el sentido que tienen una amplia gama de semántica.

En [7], fusionaron δ-ontologías usando la revisión de creencias y lógica inviable programación, extendiendo el marco de δ-ontologías al combinar la revisión de creencias, ontologías lógicas de argumentación y descripción, de dos ontologías tales que la unión de éstas podría ser inconsistente. Mientras que, en [3] análizaron el marco de los fragmentos de lógica propositiva considerando que el resultado del cambio de operación esté en el mismo fragmento que el cambio de creencia, así como, mantener el comportamiento del operador original sin cambios en caso de que entregue un resultado que ya encaja en el fragmento. Como resultado obtuvieron un enfoque que no se limita a un solo fragmento de lógica propositiva, sino que es aplicable a tres fragmentos de lógica propositiva.

En [2], describen un método de revisión que considera las bases de creencias "planas"sin información extra, adaptando la fusión de creencias bajo restricciones para llevar a cabo la revisión de las mismas, al comparar el Δ_{ps_μ} contra dos operadores de revisión de creencias: Dalal y BHQ, los resultados mostraron el potencial del enfoque de los PS, en donde las operaciones eran más simples así como más apropiadas en algunos escenarios y desarrollaron un prototipo de software de revisión de creencias.

En [21], ampliaron la noción de satisfacción, para incluir una "medida"de satisfacción y resolver problemas de la Primera Competencia Internacional de Horarios, donde fue necesario encontrar algunas heurísticas que permitieron reducir

el espacio de memoria. Integraron esta expansión en una fase de preprocesamiento que definió estados mentales consistentes, deseos (Belief-Desires-Intentions) de los agentes. El resultado fueron los estados que maximizan la Suma de la Satisfactibilidad Parcial del perfil, sin ser necesario definir un preordenamiento parcial.

En [17], se usó un conjunto de datos de una red de sensores, para lograr biomarcadores de marcha para pacientes con neuropatía diabética e individuos sanos, crearon un dataset que incluyó características de personas sanas o al contrario, así como el estado de salud conocido de cada caso, pero sin etiquetar. Los datos de prueba confirmaron la eficiencia de los modelos tras la aplicación de una búsqueda exhaustiva que combinó diversos algoritmos (*clasificadores ensamblados+aprendizaje profundo*) y la selección del que tuvo el máximo porcentaje de instancias correctamente clasificadas, en donde las instancias mostraron un alto grado de certeza con la existencia de atrofia en los músculos que conducen a una marcha anormal debido a la neuropatía diabética, logrando una precisión del 86.46 %.

3. Materiales y métodos

3.1. Conjunto de datos de neuropatía diabética

El conjunto de datos se utilizó en [17], en donde los autores realizaron una clasificación mediante la combinación de algoritmos ensamblados y aprendizaje profundo. Los datos se obtuvieron del departamento de servicios médicos de la Universidad Autónoma de Tabasco. Los registros se tomaron de la marcha de pacientes con neuropatía diabética y sanos, utilizando cinco sensores acelerómetros ADXL-335 en tres ejes (x, y, z), conectados a una tarjeta Arduino MEGA.2560; se colocó un sensor en la espalda baja, otro en la rodilla derecha, otro en la rodilla izquierda, otro en el tobillo derecho, y el último en el tobillo izquierdo del paciente. Asimismo, la tarjeta Arduino MEGA se configuró para el registro de 400 puntos en 5 minutos, con un total de diez pacientes con neuropatía y cinco pacientes sanos, y la lectura interna de las posiciones contiene 400 registros por paciente.

Los nombres y representaciones de las tres posiciones que ocupó el sensor son los siguientes: espalda baja (col-x, col-y, col-z), rodilla derecha (rodDer-x, rodDer-y, rodDer-z), rodilla izquierda (rodIzq-x, rodIzq-y, rodIzq-z), tobillo derecho (tobDer-x, tobDer-y, tobDer-z), y tobillo izquierdo (tobIzq-x, tobIzq-y, tobIzq-z). Para determinar si el registro pertenece a pacientes con neuropatía diabética, se agregó un valor positivo (TRUE) y falso (FALSE) de pacientes sanos, lo que generó un tipo de conjunto binario (enfermo, sano) con un total de 16 atributos. La Tabla 1 muestra las características de los pacientes: género, edad, peso, altura, años de sufrimiento y etiología. La Tabla 2 muestra un segmento del conjunto de datos, que representa las tres posiciones de cada sensor ocupado (col-z, col-y, col-x, rodDer-z, rodDer-y, rodDer-x, ...), y la columna 16 indica si el paciente tiene la enfermedad o es sano, el dataset contiene un total de 6000 registros y 16 columnas.

Tabla 1. Información de pacientes.

Género	Edad (años)	Peso (Kg)	Estatura (m)	Tiempo de padecimiento (años)	Causa
Masculino	54	89	1.70	5	Hereditario
Masculino	60	108	1.65	10	Alimentación
Femenino	56	99	1.60	4	Hereditario
Masculino	56	81.5	1.62	6	Hereditario
Masculino	62	73	1.57	15	Alimentación
Femenino	50	70	1.59	8	Hereditario
Masculino	58	102	1.61	6	Alimentación
Masculino	57	87.7	1.58	8	Alimentación
Femenino	61	90	1.65	3	Hereditario
Masculino	50	83.2	1.63	5	Hereditario
Femenino	35	72	1.61	0	Sano
Masculino	38	82	1.65	0	Sano
Masculino	45	95	1.67	0	Sano
Masculino	40	75	1.59	0	Sano
Femenino	29	59	1.55	0	Sano

Tabla 2. Dataset de neuropatía diabética (fragmento).

col-z	col-y	col-x	rodIzq-z	rodIzq-y	...	caso
1.65	1.74	1.99	1.60	1.69	...	NeuroDiabe
1.65	1.75	2.01	1.61	1.70	...	NeuroDiabe
1.66	1.75	2.01	1.61	1.70	...	NeuroDiabe
1.67	1.86	1.97	1.64	1.74	...	Sano

En el método, los datos se distribuyen de la siguiente manera: l_1 representa el sensor en la posición z (col-z), l_2 en la posición y (col-y), l_3 en la posición x (col-x), l_4 en la posición z (rodIzq-z), l_5 en la posición y (rodIzq-y), l_6 en la posición x (rodIzq-x), l_7 en la posición z (rodDer-z), l_8 en la posición y (rodDer-y), l_9 en la posición x (rodDer-x), l_{10} en la posición z (tobIzq-z), l_{11} en la posición y (tobIzq-y), l_{12} en la posición x (tobIzq-x), l_{13} en la posición z (tobDer-z), l_{14} en la posición y (tobDer-y), l_{15} en la posición x (tobDer-x), y l_{16} el caso del diagnóstico del paciente {enfermo o sano}.

3.2. Binarización del conjunto de datos

La binarización incluye la reducción de información, donde el único valor posible es verdadero (1) o falso (0) [20]. Se han propuesto diferentes métodos de binarización, pueden reducirse seleccionando un umbral, un promedio y otras operaciones clásicas para obtener resultados satisfactorios [15]. El método más simple de binarización es la determinación de umbrales, también llamado umbralización, y puede ser de manera global, local o adaptativa [11,16].

Considerando que el operador PS-Merge trabaja con datos de entrada y salida en formato binario, se binarizaron todas las variables del conjunto de datos, que consisten en 16 columnas y 6000 registros (como se describe en la Sección 3.1). Se aplicaron 8 algoritmos de agrupación robustos (*Spectral Clustering*, *Affinity Propagation Clustering*, *MiniBatchKMeans*, *MeanShift*, *Agglomerative Clustering*, *DBSCAN*, *OPTICS*, *KMeans*) para binarizar, e identificar cuál es la mejor separación en dígitos binarios. Así como tres métodos: primero, al tomar el valor de umbral. En seguida, al tomar el promedio global de cada columna. Finalmente, al tomar el valor promedio de 4 algoritmos de agrupación más adecuados para las dos categorías, es decir, de los 8 algoritmos de agrupación, cuatro pueden separar mejor los datos entre clases.

3.3. Operador *PS*-Merge

Antes de definir este paradigma proposicional, es necesario describir los conceptos utilizados.

Sea $\mathcal{L}$ un lengaje proposicional que consiste en: un conjunto de n variables proposicionales P=$\{p_1, p_2, p_3, ..., p_n\}$, y conectores lógicos ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$) para formar proposiciones compuestas. Hay dos formas normales, las disyuntivas (FND) y las conjuntivas (FNC). En el paradigma disyuntivo son disyunciones de conjunciones de literales, y el paradigma conjuntivo son conjunciones de disyunciones de literales, donde un literal (l) es una variable proposicional o su negación (p, $\neg p$, q, $\neg q$, ...).

Una cláusula es una disyunción de literales finita de la forma $l_1 \vee ... \vee l_n$, donde cada l_i con $i = (1, ..., n)$ es un literal.

Un término es una conjunción de literales finita de la forma $l_1 \wedge ... \wedge l_n$, donde cada l_i con $i = (1, ..., n)$ es un literal.

Una fórmula υ, está en forma normal disyuntiva (FND) si es de la forma $\upsilon_1 \vee \upsilon_2 \vee ... \vee \upsilon_n$ donde cada υ_i con $i = 1, ..., n$ es un término.

Una fórmula ψ, está en forma normal conjuntiva (FNC) si es de la forma $\psi_1 \wedge \psi_2 \wedge ... \wedge \psi_n$ donde cada ψ_i con $i = 1, ..., n$ es una cláusula.

Si K es una proposición representando una base de conocimientos o creencias, entonces Q_K representa la base de conocimiento equivalente en su FND.

Una interpretación, estado, o mundo es una función ω de P a 0, 1, estos dos valores son identificados con los valores clásicos de Verdadero (V) y Falso (F) respectivamente.

El conjunto de todas las posibles interpretaciones se denotará como $\mathcal{W}$ y sus elementos se denotarán como vectores de la forma $(\omega(p_1), ..., \omega(p_n))$. La cardinalidad del conjunto $\mathcal{W}$ es $\mid \mathcal{W} \mid = 2^n$.

Si $\omega(\upsilon) = 1$ para una determinada interpretación ω, una vez que ω se extiende a la conjunción y el resto de los conectores de manera convencional, la fórmula υ se denomina satisfecha o consistente. En este caso, ω es un modelo de υ.

Si tanto l como $-\,l$ están en un *multiset*, entonces contienen pares $\langle\, l, -\,l \,\rangle$ inconsistentes.

Cuando cualquier interpretación que hace que la premisa sea verdadera (operando izquierdo) también hace que la conclusión sea verdadera (operando dere-

cho), se produce una relación semántica ($\models$) de resultados. Cuando cada modelo de la premisa es también el modelo de conclusión. La conclusión es el resultado lógico de la premisa.

El método empleado utiliza la *fusión de creencias*, *revisión de creencias* y *noción de satisfacción parcial*. Para la fusión de creencias se emplea PS-Merge, el que considera bases de creencias inconsistentes, extrayendo información que no cause inconsistencias de las bases de creencias y sólo examina el mejor valor de los casos, está basado en la satisfacción parcial, la cual se define directamente desde la distancia del mundo w hasta la base K sin recurrir a la distancia entre los mundos. Para calcular la distancia del mundo a un perfil se emplea la suma de *la satisfacción parcial* de cada base de creencias. En [1] se define el operador como se muestra a continuación.

Definición 1. *La satisfacción parcial* se define a partir de una base de conocimiento normalizada en formato FND: Q_K, $\omega \in \mathcal{W}$ una interpretación cualquiera y $|P|$=n. La satisfacción parcial de Q_k para ω denotada por $\omega_{ps}(Q_k)$, se define como:

1. Si Q_K es una conjunción $C_1 \wedge ... \wedge C_s$ donde cada C_i es una literal, entonces:

$$\omega_{ps}(Q_K) = max\left\{\sum_{i=1}^{s} \omega_{ps}\left(\frac{\omega(C_i)}{s}\right), \left(\frac{n - |P(\bigwedge_{i=1}^{s} C_i|)}{2n}\right)\right\}, \tag{1}$$

donde: n: es el número de variables proposicionales del lenguaje considerado y s: es el número de literales que aparecen en la conjunción.

2. Si Q_K es una disyunción $D_1 \vee ... \vee D_r$ donde cada D_i es una conjunción de literales como:

$$\omega_{ps}(Q_K) = max\{\omega_{ps}(D_1), ..., \omega_{ps}(D_r)\}. \tag{2}$$

Cada caso de la base de datos es una fuente de información, es decir, cada caso (base de conocimiento) K es una conjunción de literales.

Definición 2. El operador Δ_{ps} se define a partir de los valores que maximizan la suma de la satisfacción parcial de las bases del perfil. Sea E un perfil de conocimiento obtenido de las bases de conocimiento $\{Q_{k_1}, ..., Q_{k_m}\}$, la fusión de E empleando satisfacción parcial, denotada por Δ_{ps} (E) es una función que va del conjunto de perfiles de creencias al conjunto de bases de conocimiento tal que:

$$mod(\triangle_{ps(E)}) = \left\{\omega \in \mathcal{W} | \sum_{i=1}^{m} \omega_{ps}(Q_{K_i}) \geq \sum_{i=1}^{m} \omega'_{ps}(Q_{K_i}) \; para \; todo \; \omega' \in \mathcal{W}\right\}. \tag{3}$$

3.4. Implementación del operador PS-Merge

El método trata cada caso del conjunto de datos como una fuente de información, y cada base de conocimiento tiene la siguiente forma: $K = l_1 \wedge l_2 \wedge l_3 \wedge$, ..., $\wedge\, l_{15} \rightarrow l_{16}$. Por ejemplo, cuando ocurre el caso 10, se representa como K_{10}=

(0,0,1,1,1,1,0,0,0,1,1,1,1,1,0,0), dado que los datos de entrada y salida están en formato binario.

Se identifican 16 variables en el conjunto de datos, y la fórmula general para obtener el número de combinación es: 2^n, es decir, las posibles combinaciones son 2^{16}= 65,536:
1) 0000000000000000
2) 0000000011000001
3) 0001100010000010
...
65,536) 1111111111111111

Si los casos están en FNC, deben transformarse a FND.

De acuerdo a la expresión a→b ≡ $\neg a \vee b$, se puede determinar una forma normal disyuntiva (FND) o forma normal conjuntiva (FNC) correspondiente a cada K que niega cada variable y conecta todos los elementos con disyunciones, $\neg l_1 \vee \neg l_2 \vee \neg l_3 \vee \neg l_4 \vee \neg l_5 \vee \neg l_6 \vee \neg l_7 \vee \neg l_8 \vee \neg l_9 \vee \neg l_{10} \vee \neg l_{11} \vee \neg l_{12} \vee \neg l_{13} \vee \neg l_{14} \vee \neg l_{15} \vee l_{16}$, después de la transformación de los casos en sus formas normales, se aplica el algoritmo, el objetivo que es el valor que debe ser y el resultado que es la evaluación obtenida por el algoritmo.

Al realizar la fusión se obtienen combinaciones, en donde a veces se puede obtener dos resultados 0, 1 con el mismo valor de atributos, dado que las características de los datos siempre vienen en paquetes considerando todas las variables, se pueden obtener ambas, pero no son contradicciones, es decir, el algoritmo puede encontrar una fórmula como $\{\neg p_1 \wedge \neg p_2 \wedge p_3 \wedge p_4 \wedge p_5 \wedge p_6 \wedge \neg p_7 \wedge \neg p_8 \wedge \neg p_9 \wedge p_{10} \wedge p_{11} \wedge p_{12} \wedge p_{13} \wedge p_{14} \wedge p_{15} \wedge p_{16}\}$ y $\{\neg p_1 \wedge \neg p_2 \wedge p_3 \wedge p_4 \wedge p_5 \wedge p_6 \wedge \neg p_7 \wedge \neg p_8 \wedge \neg p_9 \wedge p_{10} \wedge p_{11} \wedge p_{12} \wedge p_{13} \wedge p_{14} \wedge p_{15} \wedge \neg p_{16}\}$, lo cual es un diagnóstico ambiguo: el paciente tiene neuropatía diabética, pero en el conjunto de prueba es sano.

Se dividió la base de conocimiento en dos conjuntos, el conjunto de entrenamiento y el conjunto de pruebas, asignando el 70 % del conjunto de datos original al entrenamiento y el 30 % para el conjunto de pruebas, esto de manera aleatoria, la asignación quedó con 4200 casos de entrenamiento y 1800 de casos para prueba.

Ahora bien, en la primer fase se fusionaron 4200 casos del conjunto de entrenamiento. Para obtener el resultado de solución; se compararon los casos de la nueva base de conocimiento generada por *la fusión de creencias* con los casos del conjunto de prueba; si existían modelos con el mismo valor que la base obtenida por la fusión, podría ser 0, 1 o ambos.

Si encontró ambos, significa que en la base de fusión hay tanto positivo como negativo con las mismas variables y se representa como BB (Both-Both), en otras palabras, si en la fusión existen ambos casos 0 o 1, quiere decir que no sabe cómo dar solución, porque no hay ningún caso que pueda decir que sea 0 o 1, incluso, si una de las soluciones obtenida de la fusión coincide con un caso del conjunto de prueba, entonces es 0 y se interpreta con B, mientras que, al

segundo dígito para que sea 0, 1, se le asigna al primer dígito una B, para que no tenga 0 o ambos, porque puede que en el siguiente caso coincidan.

Para el tercer método binario, el algoritmo no encontró una respuesta en 2 casos, con 16 variables; otro aspecto del algoritmo fue que obtuvo 88 resultados de falsos negativos, en donde el resultado fue B, es decir, el algoritmo no encontró el valor correspondiente, para determinar si es un caso con neuropatía diabética o sano.

Para reducir los casos confusos se realizaron más pruebas. Mientras que, para el segundo método binario, el algoritmo encontró 65 resultados de falsos negativos, 254 resultados falsos positivos, dicho de otra manera, encontró casos de neuropatía diabética en donde los casos deberían ser sano. Asimismo, el algoritmo no encontró respuesta en 1 caso. Y finalmente para el primer método binario, el algoritmo encontró 56 casos falsos positivos, asignando un valor que se asimilara al valor que debería ser, en los casos en donde ambos tienen que es igual 0,1 entonces sí pudo definir un solo valor, ya sea que esté correcto o incorrecto, pero determinó un valor o se quedó como BB.

En otras palabras, se transformó lo que se inició como 0,1 a BB, en el caso que sea BB, ya sea que entró como BB desde el inicio, o que entró como 0,1, hace el proceso para que encuentre un resultado, si no lo encuentra, queda con los dígitos que inicio. Si son iguales ambos dígitos, y se asignó BB o encontró una solución, al segundo dígito se asigna una B, en el caso que ambas sean diferentes, sólo 1 se satisface y no ambas, sólo se le asigna B a una y no serán iguales, en el caso que ambas se satisfacen se asigna BB.

En la segunda fase, el algoritmo analizó los resultados obtenidos de la primer fase aplicando *la fusión de creencias*, los cuales son resultados de reducción de los B, en un momento dado con cada caso se recorre toda la base de datos para distinguir que tanto satisface a toda la base de datos, inicialmente nadie está satisfecho, aunque son 4200 registros, realmente en las combinaciones no se utilizan todas, se pueden encontrar algunas que se repiten, los resultados deben satisfacer las 16 variables para todos los casos.

Cuando en la primera entrada el máximo en satisfabilidad es 0, muy probablemente cambiará, por ejemplo, tenemos $K_{10} = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0$ y la base de datos en cuestión esté como K= 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1, pero si la suma parcial al menos tiene un valor que satisface parcialmente una pequeña porción de la base de datos en cuestión, es una solución, significa que satisface toda la base de datos de datos.

Analiza los 4200 casos hasta obtener una mejor solución, de las 65536 combinaciones sólo unas cuantas maximizan toda la satisfacción de la base de datos en cuestión, en efecto, cuando el algoritmo obtiene resultados que hacen que la mayoría de los casos estén satisfechos al máximo, en los casos en donde ambos tienen 0,1 sí pudo definir un solo valor para el conjunto de prueba, ya sea que este correcto o incorrecto, pero determinó un solo valor o quedó como BB, dando una solución correcta o incorrecta.

4. Resultados y discusiones

En esta investigación se aplicó el operador PS-Merge a un conjunto de datos con biomarcadores de la marcha de neuropatía diabética, con registros de personas con la enfermedad y personas sanas.

Una persona con neuropatía diabética que tiene un resultado positivo en la prueba y se denomina Verdadero Positivo (TP), mientras que una persona con neuropatía diabética, pero con una prueba negativa se denomina Falso Negativo (FN), una persona sin neuropatía diabética que tiene un resultado positivo se denomina Falso Positivo (FP), mientras que una persona sin neuropatía diabética, pero con una prueba negativa se denomina Verdadero Negativo (TN) [9]. La precisión se calculó a partir de la ecuación $((TP+TN)/(TP+FP+FN+TN))$. En la Tabla 3, se observa la matriz de confusión base.

Tabla 3. Matriz de confusión.

	Predijo sí	Predijo no
Real sí	Verdadero positivo	Falso positivo
Real no	Falso negativo	Verdadero negativo

Deriva una matriz de confusión para los tres métodos binarios, se utilizaron 4200 registros para cada método. Para el primer método el operador clasificó 4198, en donde, clasificó correctamente 1142 casos verdaderos positivos, 340 casos verdaderos negativos, por lo contrario, 88 casos falsos positivos y 228 falso negativo, y en 2 registros no considero correcto o incorrecto un resultado. La precisión se calculó a partir de la ecuación $((TP+TN)/(TP+FP+FN+TN))$, tal como, (1142+340) / (1142+88+228+340) = 82.42 %, (Véase en la Tabla 4).

Tabla 4. Matriz de confusión, primer método binario.

Clase	Sano en control	Enfermo
Sano en control	1142	88
Enfermo	228	340

En el segundo método binario, el operador clasificó correctamente 1124 casos verdaderos positivos, 356 casos verdaderos negativos, no obstante, el operador obtuvo 65 casos falsos positivos y 254 casos falsos negativos, así como 1 registro en el cual no encontró respuesta (Véase en la Tabla 5). La precisión fue (1124+356) / (1124+65+254+356) = 82.26 %.

Finalmente, para el tercer método binario, el operador clasificó correctamente 1132 casos verdaderos positivos, 340 casos verdaderos negativos, en cambio, 56 casos falsos positivos y 272 falso negativo (Véase en la Tabla 6). Asimismo, la precisión fue (1132+340) / (1132+56+272+340) = 81.77 %.

Tabla 5. Matriz de confusión caso 2.

Clase	Sano en control	Enfermo
Sano en control	1124	65
Enfermo	254	356

Tabla 6. Matriz de confusión caso 3.

Clase	Sano en control	Enfermo
Sano en control	1132	56
Enfermo	272	340

PS-Merge es un operador basado en lógica proposicional, que no se había implementado para clasificar datos de neuropatía diabética, el proceso utilizó de tres técnicas: *fusión de creencias*, *revisión de creencias* y la *noción de satisfacción parcial*. En esta investigación se logró clasificar correctamente datos de neuropatía diabética utilizando *la fusión de creencias*, obteniendo resultados de precisión de más del 80 %; resultados aceptables, en comparación a lo obtenido con técnicas de aprendizaje automático utilizando el mismo conjunto de datos [17], y al aplicar clasificadores ensamblados y aprendizaje profundo, con un resultado de precisión de más del 85 %. Ahora bien en [9], aplicaron el operador *PS*-Merge para casos de diagnóstico de cáncer oral, los resultados permitieron plantear la posibilidad de fusionar un conjunto de datos sin afinar parámetros, en donde los mejores resultados dan una precisión superior al 75 % de instancias correctamente clasificadas.

5. Conclusiones y trabajos futuros

En esta investigación hemos logrado clasificar clases binarias, implementando el operador *PS*-Merge, utilizando la noción de *fusión de creencias*, *revisión de creencias* y la *noción de satisfacción parcial*, en casos de neuropatía diabética, en donde los resultados mostraron un valor de precisión superior al 80 % de instancias correctamente clasificadas en los tres métodos binarios, por lo que es comparable a lo que se logró con técnicas de aprendizaje automático empleando el mismo conjunto de datos. En la Sección 3.2 presentamos una idea para binarizar el conjunto de datos. Asimismo, en la Sección 4 señalamos los resultados aplicando el operador *PS*-Merge, así como la matriz de confusión para cada uno de los métodos binarios. Se puede observar que el operador puede ser utilizado en cualquier conjunto de datos que esté representado en vectores binarios. Puesto que, el operador no utiliza la estadística y configuración alguna de parámetros, como lo realizan las técnicas de aprendizaje automático con algoritmos clásicos para datos del área médica, tampoco se requieren técnicas de preprocesamiento de datos, sólo se llevó a cabo una selección de variables para identificar el número de variables a fusionar.

Como trabajo a futuro se pretende:

- Implementar el operador PS-Merge en otros conjuntos de datos y comparar los resultados del operador con resultados de algoritmos clásicos.
- Ampliar el operador para clases multiclase ya que sólo considera clases binarias.

Referencias

1. Chávez-Bosquez, O., Pozos-Parra, P.: Modelando la toma de decisiones mediante fusión de creencias. Recta 15(2) (2014)
2. Chávez-Bosquez, O., Pozos-Parra, P., Ma, J.: ImplementingΔ ps (ps-merge) belief merging operator for belief revision. Computación y Sistemas 21(3), 419–434 (2017)
3. Creignou, N., Papini, O., Rümmele, S., Woltran, S.: Belief merging within fragments of propositional logic. ACM Transactions on Computational Logic (TOCL) 17(3), 1–28 (2016)
4. Cuevas, I.J.F., Núñez, Z.A.C., Ascencio, R.L., Vásquez, C.: Detección de neuropatía diabética periférica en adultos mayores de 60 años en el centro de salud"méxico bid"de colima, méxico. Archivos de medicina 14(4), 1 (2018)
5. Diller, M., Haret, A., Linsbichler, T., Rümmele, S., Woltran, S.: An extension-based approach to belief revision in abstract argumentation. International Journal of Approximate Reasoning 93, 395–423 (2018)
6. Everaere, P., Konieczny, S., Marquis, P.: On egalitarian belief merging. In: Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning (2014)
7. Gómez, S.A., Simari, G.R.: Merging of ontologies using belief revision and defeasible logic programming. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial 16(52), 16–28 (2013)
8. Ing, V.A., Salazar, J., Garicano, C., MgS, M.C.G.M., Añez, R., et al.: Una introducción a las aplicaciones de la inteligencia artificial en medicina: Aspectos históricos. Revista Latinoamericana de Hipertension 14(5), 590–600 (2019)
9. Kareem, S.A., Pozos-Parra, P., Wilson, N.: An application of belief merging for the diagnosis of oral cancer. Applied Soft Computing 61, 1105–1112 (2017)
10. Luna, I., De, G., Zacarias Flores, F., García García, A.D.: Algoritmo basado en modelos para la revisión de creencias entre formas normales conjuntivas. Computación y Sistemas 21(3), 435–448 (2017)
11. Molina, E., Diaz, J., Hidalgo-Silva, H., Chávez, E.: Algoritmos de binarizacion robusta de imágenes con iluminación no uniforme. Revista Iberoamericana de Automática e Informática industrial 15(3), 252–261 (2018)
12. Osisanwo, F., Akinsola, J., Awodele, O., Hinmikaiye, J., Olakanmi, O., Akinjobi, J.: Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT) 48(3), 128–138 (2017)
13. Pop-Busui, R., Boulton, A.J., Feldman, E.L., Bril, V., Freeman, R., Malik, R.A., Sosenko, J.M., Ziegler, D.: Diabetic neuropathy: a position statement by the american diabetes association. Diabetes care 40(1), 136—154 (2017)
14. Ramírez, R.B., Arellano, R.M., Arévalo, J.J.: la tecnología y la inteligencia artificial como futuro en el área médica. Universitas, Revista de Ciencias Sociales y Humanas (21), 185–190 (2014)

15. Rodriguez, R., Sosa, J.: Importance of the use of the parametric logarithm in a no-supervised strategy for the image binarization. IEEE Latin America Transactions 14(3), 1434–1439 (2016)
16. Sáenz, K.d.J.B.: Clasificación de los frutos de café según su estado de maduración y detección de la broca mediante técnicas de procesamiento de imágenes. Prospectiva 14(1), 15–22 (2016)
17. Sánchez-DelaCruz, E., Weber, R., Biswal, R., Mejía, J., Hernández-Chan, G., Gómez-Pozos, H.: Gait biomarkers classification by combining assembled algorithms and deep learning: Results of a local study. Computational and Mathematical Methods in Medicine (2019)
18. Schwind, N., Inoue, K., Bourgne, G., Konieczny, S., Marquis, P.: Belief revision games. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
19. Shear, T., Fitelson, B.: Two approaches to belief revision. Erkenntnis 84(3), 487–518 (2019)
20. Triana, N., Jaramillo, A.E., Gutiérrez, R.M., Rodríguez, C.A.: Técnicas de umbralización para el procesamiento digital de imágenes de gem-foils. Scientia et technica 21(4), 352–359 (2016)
21. Walulya, I., Palyvos-Giannas, D., Nikolakopoulos, Y., Gulisano, V., Papatriantafilou, M., Tsigas, P.: Viper: A module for communication-layer determinism and scaling in low-latency stream processing. Future Generation Computer Systems 88, 297–308 (2018)
22. Zhang, S.S., Wu, Z., Zhang, L.C., Zhang, Z., Chen, R.P., Huang, Y.H., Chen, H.: Efficacy and safety of pregabalin for treating painful diabetic peripheral neuropathy: a meta-analysis. Acta Anaesthesiologica Scandinavica 59(2), 147–159 (2015)

ISSN 1870-4069

Obtención de parámetros de la prueba de la estrella utilizando algoritmos genéticos y cómputo de alto rendimiento

Juan Miguel Pérez-García, Moisés Emmanuel Ramírez-Guzmán, Jorge González-García

Universidad Tecnológica de la Mixteca,
Instituto de Computación,
México

jnxho123d@gmail.com, {merg, jgonzal}@mixteco.utm.mx

Resumen. El uso de cómputo de alto rendimiento para la reducción del tiempo en la obtención de los parámetros para la prueba de la estrella usando algoritmos genéticos es de gran importancia debido a que estos algoritmos tardan un largo periodo de tiempo en dar respuesta si se ejecutan en computadoras convencionales, por lo que se opta en el uso de plataformas como CUDA. Además, las estructuras de datos que utilizan los algoritmos genéticos, se acoplan perfectamente a la estructura de ejecución de CUDA. La implementación del algoritmo, tomando en cuenta lo anterior, se lleva a cabo eficientemente y de esta manera se puede demostrar que la obtención de los parámetros se realiza en menor tiempo. Esta reducción de tiempo permite llevar a cabo una mayor cantidad de pruebas y hacer modificaciones a los parámetros del algoritmo para poder así reducir las aberraciones en el sistema óptico.

Palabras clave: Óptica, prueba de la estrella, algoritmos genéticos, CUDA.

Obtaining Parameters of the Star Test using Genetic Algorithms and High-Performance Computing

Abstract. The use of high-performance computation to reduce the time in obtaining the parameters for the star test using genetic algorithms is of great importance because these algorithms take a long period of time to respond if they are executed in Conventional computers, so you choose to use platforms like CUDA. Furthermore, the data structures that the genetic algorithms use fit perfectly into the CUDA execution structure. The implementation of the algorithm, taking into account the above, is carried out efficiently and in this way it can be shown that obtaining the parameters is performed in less time. This reduction in time allows

a greater number of tests to be carried out and modifications to the algorithm parameters to be able to reduce aberrations in the optical system.

Keywords: Optics, star test, genetic algorithms, CUDA.

1. Introducción

1.1. Prueba de la estrella

En el área de la física y de manera más específica, en la rama de la óptica, la prueba de la estrella es un método de medición de la calidad de una superficie óptica, de entre estas, las más comunes son los espejos cóncavos y los lentes. El mecanismo de esta prueba consta de un arreglo óptico experimental y un algoritmo de optimización. Del arreglo óptico se obtiene una imágen con las aberraciones del sistema y el algoritmo de optimización se encarga obtener los parámetros que generen una nueva imágen con menos aberraciones de tal manera que la calidad de la lente mejore; para llevar a cabo la optimización, se pueden utilizar metodos heurísticos como lo son los algoritmos genéticos.

Arreglo óptico. Este arreglo consiste en colocar una fuente puntual de luz a una distancia muy grande (S_0) y hacer incidir su luz a través del lente (L), observando o grabando el patrón de irradiancia generado a una distancia del foco (S_i) de la superficie, utilizando un dispositivo de carga acoplada (CCD por sus siglas en inglés) para registrar las ondas de luz en una imágen, así como se muestra en la Figura1. De este patrón de irradiancia se obtienen las aberraciones que se deben optimizar, las cuales son errores en el sistema óptico que generan alteraciones en la imágen generada por el CCD.

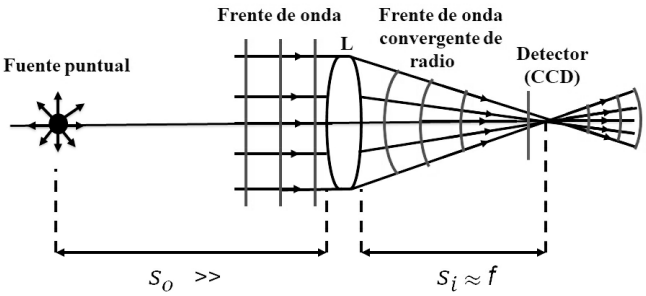


Fig. 1. Arreglo experimental de la prueba de la estrella para la prueba de lentes.

Método de optimización. Para la optimización de las aberraciones se usa el método heurístico de los algoritmos genéticos. Este método está basado en la evolución y la selección natural, que consiste en presentar un conjunto de posibles soluciones, llamado población, donde cada solución es un individuo. Durante varias generaciones se modifica la información de los individuos, simulando los principios del proceso de evolucion, es decir, la reproducción, recombinacion, mutación y la selección natural.

1.2. Algoritmos genéticos

Los Algoritmos Genéticos (AG) son técnicas de optimización y heurísticas de búsqueda, basados en la selección natural y el proceso evolutivo. La meta de estas técnicas, es encontrar la mejor solución posible o grupo de soluciones a un problema, con respecto a uno o más criterios. Un AG permite que una población compuesta por muchos individuos (soluciones) evolucionen bajo ciertas reglas a un estado que maximice su aptitud.

Los principales elementos que componen un AG se listan a continuación [1,2]:

- **Población:** Conjunto de soluciones factibles para un problema, cada una de estas se llama individuo
- **Población inicial:** Generación aleatoria inicial de soluciones factibles al problema dado
- **Selección:** Elección aleatoria y/o sesgada de individuos en base a lo que se considere "mejor"
- **Alteración de los individuos:** La generación de nuevos individuos en la población se puede hacer mediante dos operaciones: Mutación y Cruza
- **Representación:** Forma de representar a cada individuo para el problema tratado. Algunos ejemplos son: bits, enteros, rangos, permutaciones, etc.
- **Función de evaluación, error o aptitud:** Indica la calidad de un individuo en la población

Algunas de las limitaciones que se deben considerar al implementar un AG, son:

- Identificar la función de aptitud.
- Definir la representación para el problema.
- Se puede producir convergencia prematura.
- Se pueden detectar múltiples óptimos locales e inclusive no llegar a un óptimo global.

Antes de implementar un AG, se deben llevar a cabo los siguientes pasos:

1. Analizar la forma de representar las soluciones para el problema.
2. Identificar una función de aptitud.

Dados los puntos anteriores, la Figura 2 muestra gráficamente la implementación del AG general, considerando los siguientes pasos [1]:

1. Generar una población inicial de manera aleatoria con N elementos.
2. Decodificar el cromosoma de cada individuo y calcular su aptitud.
3. Utilizar un método de selección y aparear los cromosomas (realizar la cruza) para generar descendientes.
4. Aplicar las operaciones de mutación a los descendientes.
5. Aplicar un criterio de elitismo para generar la siguiente generación.
6. Si se cumple con algún criterio de parada terminar, en caso opuesto regresar al paso 2.

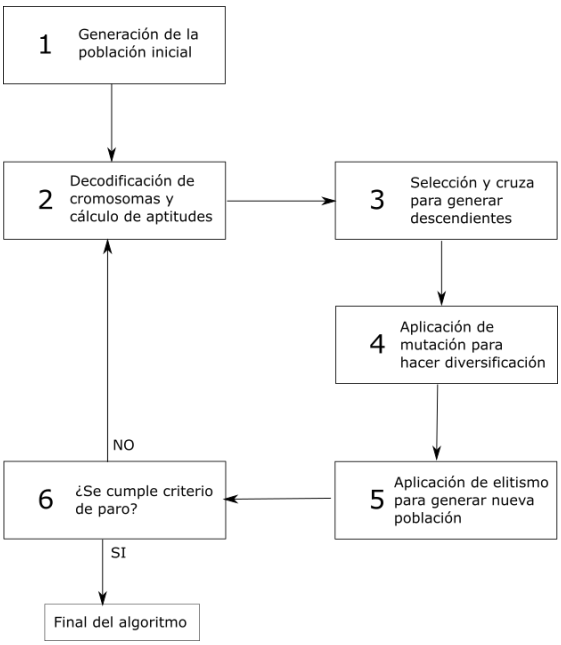


Fig. 2. Diagrama de flujo de un algoritmo genético general.

Como se mencionó anteriormente, los AGs pueden implementarse, de tal manera que se pueda realizar la búsqueda de soluciones de manera concurrida. Es gracias a esta característica, que se puede llevar a cabo la implementación del AG en plataformas de cómputo de alto rendimiento con programación paralela y concurrente, como lo es el caso de CUDA.

En 2019 Jhon Runwei Cheng y Mitsuo Gen, plantean un panorama general de las ventajas y problematicas del diseño de algoritmos genéticos sobre plataformas de uso de GPUs. Esta propuesta plantea representar los datos en el algoritmo de tal manera que los recursos se aprovechen de manera más óptima, ya sea definiendo kernels que esten a nivel de genes o a nivel de individuos, es decir, que un hilo se encargue de evaluar un gen o bien que evalue a un individuo completo.

Por otro lado los problemas que pueden existir en la paralelizacion son, la cantidad de bloques e hilos por bloque se vayan a utilizar para la implementación, así como el problema de la memoria compartida, ya que los dispositivos pueden tener limitación en este aspecto [9].

2. Programación paralela y distribuida

Para acelerar la ejecución de un programas diseñado originalmente de forma secuencial, se debe analizar la forma en que los datos son procesados. De esta forma se pueden detectar múltiples flujos de instrucciones independientes que pueden operar sobre los datos (MIMD). Así, se pueden explotar operaciones repetitivas sobre datos que pueden ser procesados vectorialmente usando un esquema de memoria compartida, enseguida se pueden visualizar tareas de más alto nivel que en este caso pueden ser ejecutadas de forma concurrente sobre procesos independientes aplicando un esquema de procesamiento distribuido. La programación paralela es un modelo para escribir programas que se ejecuten de manera concurrente sobre equipos con varios procesadores. La programación distribuida se ocupa cuando se deberá ejecutar un programa en varios equipos de cómputo que se comunican a través de una red de interconexión [3].

Usualmente, se desea que un programa paralelo pueda ser general y ejecutarse sobre una variedad grande de equipos de cómputo de manera eficiente. La implementación de programas paralelos puede llevarse a cabo usando bibliotecas que pueden invocarse desde lenguajes de programación secuenciales, extensiones a lenguajes o modelos nuevos de ejecución. El modelo de programación paralela usando memoria compartida puede implementarse usando la biblioteca OpenMP. Esta biblioteca contiene directivas que permiten la ejecución de regiones de código en paralelo.

Debido a que el presente trabajo utiliza programación paralela, a continuación se describen algunos términos importantes [4]:

- **Tarea**: es la unidad de trabajo al momento de ejecutarse un programa paralelo. Cada tarea se ejecuta secuencialmente pero existe concurrencia con respecto a otras tareas. Las tareas pueden clasificarse según su granularidad en tareas sencillas y tareas complejas.
- **Proceso**: Son unidades abstractas que realizan las tareas asignadas a los procesadores. Los procesos se comunican y sincronizan para realizar las tareas.
- **Procesador**: es una unidad de procesamiento que ejecuta un proceso.

Para realizar el proceso de paralelización como se muestra en la Figura 3 se deben llevar a cabo 4 tareas principales [4]:

- **Descomposición del problema** en tareas pequeñas. El ideal es que las tareas sean lo mas pequeñas posibles representando cantidades similares de trabajo, y evitando redundancia de proceso de cómputo y/o almacenamiento. La identificación de estas tareas debe poderse escalar con el tamaño del

problema. Los limites de esta descomposición están acotados por la Ley de Amdahl.
- **Mecanismo de asignación** por el cual las tareas se deben distribuir entre los procesos. Se deben considerar aspectos como equilibrio de la carga y si es necesario algún algoritmo de planificación buscando evitar en la medida de lo posible las comunicaciones y los puntos de detención (*deadlocks*) [5].
- **Orquestación** para estructurar la comunicación entre procesos y efectuar la sincronización a través de algunos patrones de comunicación: local/global, estático/dinámico, estructurado/no estructurado o síncrono/asíncrono.
- **Mapeo** es la asignación de procesos o hilos de ejecución a las unidades de procesamiento buscando minimizar el tiempo de ejecución.

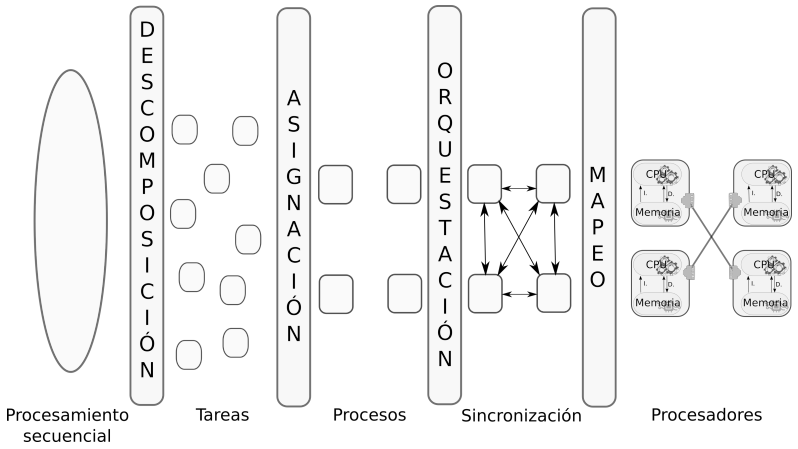


Fig. 3. Proceso de paralelizacion [4].

Al momento de buscar tareas que se pueden ejecutar de manera concurrente, es común detectar paralelismo en datos y tareas [6]:

- El **paralelismo en datos** ocurre cuando existen tareas que aplican la misma operación a diferentes elementos de un conjunto de datos. El ejemplo de la Figura 4 muestra que la operación de suma puede ser realizada de manera independiente, sobre los 100 elementos de los vectores y estas operaciones pueden ejecutarse de manera simultanea en diferentes procesadores.
- El **paralelismo de tareas** o funcional existe cuando se pueden ejecutar tareas de manera independiente sobre diferentes conjuntos de datos. En el ejemplo de la Figura 5, las asignaciones hacia **a** y **b**, así como **m** y **q** pueden ejecutarse de manera concurrente, sin embargo, **m** y **q** deben esperar a las asignaciones de **a** y **b** para poderse ejecutar, de la misma manera que h puede ser ejecutado hasta que **m** y **q** obtengan su valor correspondiente.

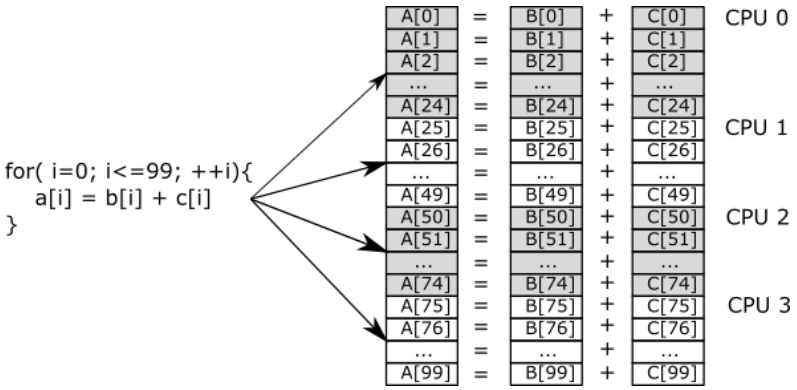


Fig. 4. Ejemplo del proceso de paralelizacion a nivel de datos. Se aprecia que el ciclo de asignaciones es dividido en cuatro bloques independientes, donde cada uno es llevado a cabo por una unidad de procesamiento.

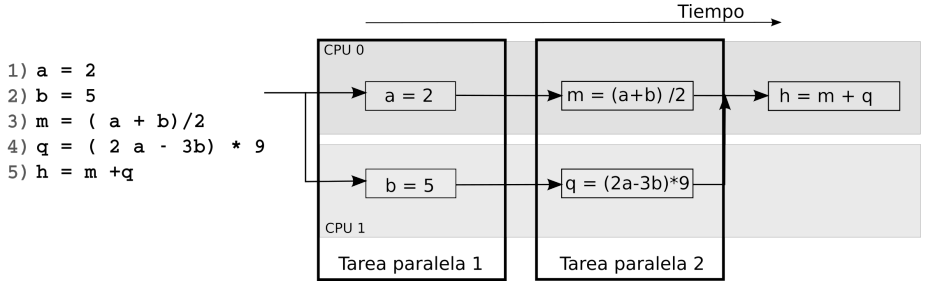


Fig. 5. Ejemplo del proceso de paralelizacion a nivel de tareas. Las primeras dos instrucciones pueden ejecutarse en paralelo. Al terminarse, pueden ejecutarse las siguientes dos también en paralelo, pero la quinta instrucción debe esperar hasta que terminen la cuarta y quinta instrucción.

A partir del año 2003, la industria de los semiconductores para el desarrollo del cómputo paralelo tomó dos caminos diferentes, el primero es la rama de los multiprocesadores, que consiste en mantener la velocidad de un programa secuencial ejecutando partes de este en varios procesadores (CPUs), tal es el caso de la plataforma que ofrece MPI y OpenMP, a través del paso de mensajes dentro de un cluster, que permite enviar parte de los datos y evaluarlos en diferentes computadoras. Por otro lado se encuentra la rama de las Unidades de Procesamiento Gráfico (GPUs), NVIDIA es una de las empresas que desarrolló esta tecnología, la cual consiste principalmente en manejar una gran cantidad de hilos en los GPUs y CUDA es el modelo de programación para la implementación de algoritmos en este modelo de cómputo paralelo.

En pruebas realizadas en 2012 la razón entre los multi-hilos de los GPUs y los multiprocesadores de los CPUs en el cálculo de números de punto flotante fue de casi 10.

Pero esta razón no necesariamente impacta en la velocidad de las aplicaciones ya que no todas las regiones de código pueden aprovechar esta mejora [7]. La diferencia principal entre estos dos modelos de cómputo paralelo, se debe principalmente en la filosofía del diseño de los chips. Por un lado, la arquitectura de un multiprocesador está orientada principalmente a la mejora del rendimiento de programas secuenciales. Mientras que los GPUs están orientados a las tareas con procesos concurrentes independientes. El ancho de banda de memoria es mayor en los GPUs, por lo que pueden mover más datos que en un multiprocesador, y esto permite comenzar a procesar de manera mas inmediata [7].

3. CUDA

CUDA es un modelo de programación paralela, cuyas siglas en inglés significan Arquitectura Unificada de Dispositivos de Cómputo, el cual fue desarrollado para implementar algoritmos que se puedan ejecutar en los GPUs de una tarjeta NVIDIA. CUDA C, es una extinción del lenguaje de programación C, añade palabras reservadas para el uso del cómputo paralelo, que además combina ambos sistemas de cómputo paralelo, GPUs y CPUs. Donde los CPUs son llamados host y con uno o varios CUDA device, que típicamente son los GPUs [7].

En este modelo de programación los algoritmos comúnmente presentan solo partes paralelizadas, es decir, en un segmento del programa, el host se encarga solo de preparar y presentar los datos resultantes, mientras que el procesamiento es llevado a cabo en el CUDA device.

Las estructuras de datos que se utilizan de manera mas eficiente en los GPUs, son las matrices y los vectores, esto se debe a que los GPUs están optimizados para el tratamiento de gráficos, como lo son las imágenes y vídeos. Por este motivo es preferible que para la implementación de algoritmos en esta plataforma se utilicen estas estructuras de datos, para modelar el problema que se desea solucionar. Se debe tomar en cuenta que los datos se representan en arreglos de una sola dimensión, es decir, vectores, por lo que al implementar un algoritmo que haga uso de matrices se debe considerar la forma de accesar a los datos.

3.1. Estructura de un programa en CUDA

La estructura de un programa en CUDA refleja la coexistencia de un host (CPU) y uno o más disposivos CUDA (GPUs) en la computadora. Las funciones y declaraciones de datos son definidos por palabras reservadas para CUDA. Las funciones y declaraciones de datos referentes a CUDA no son aceptados por un compilador de C, por lo que es necesario utilizar el compilador de NVIDIA, el cual es NVCC (NVIDIA C Compiler). Durante la compilación se separa el código del host y se utiliza el compilador estándar de C/C++ del host para su compilación y se ejecuta como un programa típico de C, mientras que NVCC se encarga de compilar el código CUDA, para luego ejecutarlo en los GPUs. En la Figura 6, se puede observar la separación de tareas de un programa en CUDA, cuando éste se compila.

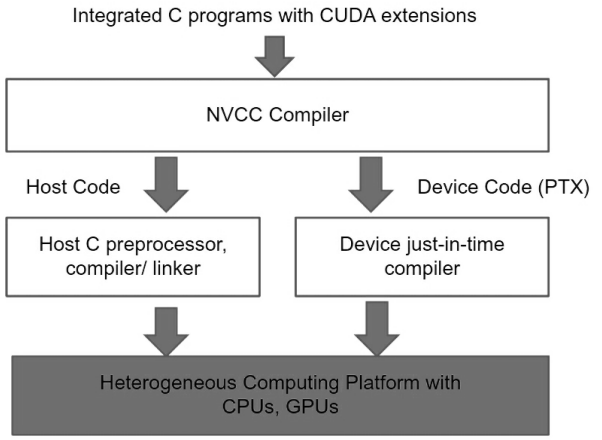


Fig. 6. Proceso de compilación de un programa en CUDA [7].

Las funciones de CUDA que se encargan de paralelizar los datos en los GPUs son los llamados Kernels. Para que un programa se pueda paralelizar en CUDA de manera eficiente es necesario que los datos sean paralelizables, es decir, los cálculos o modificaciones de los datos deben ser independientes y no tener relación con cálculos previos. En la Figura 7 se puede observar el ejemplo mas básico de un programa paralelizado en CUDA, la suma de vectores. Si se realiza la suma elemento a elemento de dos vectores A y B, y se guarda el resultado en un vector C, se puede notar que la suma de los elementos es totalmente independiente y por lo tanto se puede lanzar un hilo para sumar cada par de elementos y así realizar esta tarea de manera más eficiente.

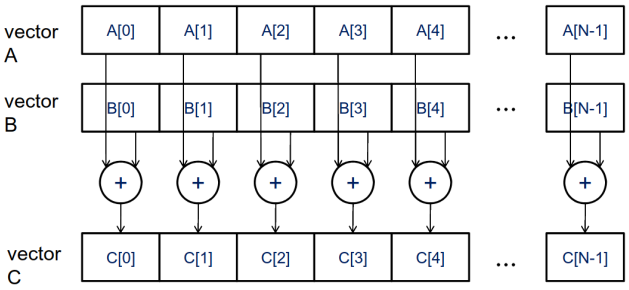


Fig. 7. Suma de dos vectores de forma paralela [7].

3.2. El kernel y los hilos

Como se mencionó anteriormente, los kernel son funciones de CUDA en las cuales se paralelizan los datos del programa. Cuando el host lanza el kernel, CUDA genera una malla de hilos, los cuales están organizados en dos niveles de jerarquía. Cada malla está organizada dentro de un arreglo de bloques de hilos (bloques). Todos los bloques tienen el mismo tamaño y se especifica cuando se lanza el kernel desde el host. El numero de hilos por bloque está disponible en la variable BlockDim. Por cuestiones de rendimiento del hardware, el tamaño de bloque debe ser múltiplo de 32.

Cada hilo en un bloque tiene un único valor dado por threadIdx. De tal manera que sea posible acceder a índices en los datos utilizando la instrucción **i = blockId.x*BlockDim.x + threadIdx.x**. Se puede declarar más de un kernel y se le puede lanzar más de una vez, dentro de un mismo programa [7]. En la Figura 8 se puede observar cómo se ejecutan los hilos al mismo tiempo dentro del código, en un mismo kernel.

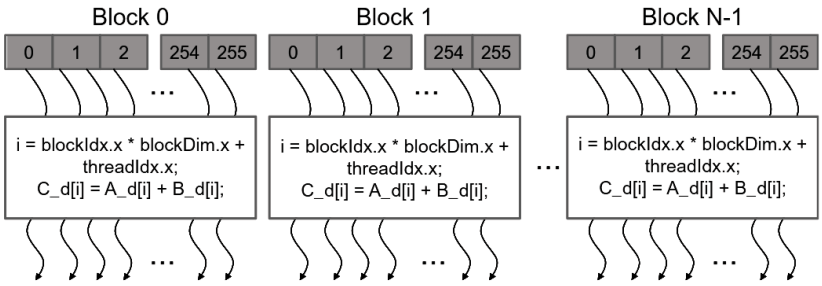


Fig. 8. Hilos de una malla ejecutándose al mismo tiempo en el código de un mismo kernel [7].

El uso de CUDA como herramienta para el desarrollo de investigaciones en la ciencia ha ido en aumento, en los últimos años debido a que esta plataforma ofrece una mejora de rendimiento en el procesamiento de datos, muy por encima que los procesadores más potentes. Como muestra el trabajo realizado por Huang XianLou y Yu ShuangYuan [8], quienes realizan la segmentación de imágenes, que se basa en cortes normalizados y cómputo paralelo con CUDA. Para la implementación de su algoritmo definieron cuatro kernels, (i) calcular la matriz de afinidad, (ii) reducir la matriz simétrica a una forma tri diagonal simétrica, (iii) Resolver el vector característico y (iv) discretizar el vector característico y seleccionar el punto del corte.

Para realizar pruebas, tomaron el Berkley Segmentation Dataset. La máquina que usaron estaba equipada con un CPU intel Core i5 de 2.8GHz, con 4GB de RAM y Sistema Operativo Windows 7.

El GPU usado fue una tarjeta NVIDIA GTX570 con 480 núcleos y 1.5GB de memoria principal, y con CUDA SDK 5.0 para la implementación del algoritmo. Los resultados experimentales muestran que el algoritmo puede segmentar imágenes de tamaño medio (200x160 píxeles) y tener una mejora de rendimiento, 2.3 veces más rápido, en comparación con la implementación orientada al uso de CPUs.

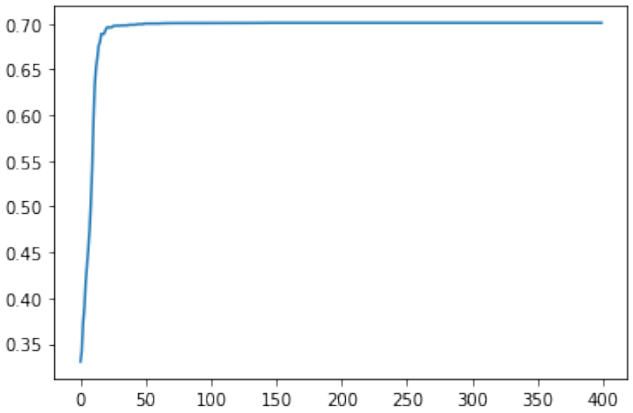


Fig. 9. Comportamiento de las aptitudes que resultan de aplicar el AG a una imágen de 600x500 píxeles, con poblacion de 1000 individuos y 400 genereciones.

4. Metodología

Los algoritmos genéticos pueden representar la la población utilizando matrices, las cuales permiten que se pueda implementar en modelos de programación paralela como CUDA. La metodología para la implementación del algoritmo genético para obtención de los parámetros de la prueba de la estrella se muestra a continuación.

1. Se llevan a cabo los pasos para la implementación de un algoritmo genético.
2. Una vez implementado el algoritmo de manera secuencial, se analizan el algoritmo para obtener las partes del código que se pueden paralelizar.
3. Se implementan los kernels necesarios para cada parte paralelizable del algoritmo.
4. Se realizan pruebas para verificar el rendimiento del algoritmo.

Este algoritmo recibe la información de una imágen con las aberraciones del sistema óptico y los parámetros en los que se llevará a cabo la simulación de la prueba de la estrella, y como salida darán los parámetros que describen la imágen optimizada de las aberraciones de sistema.

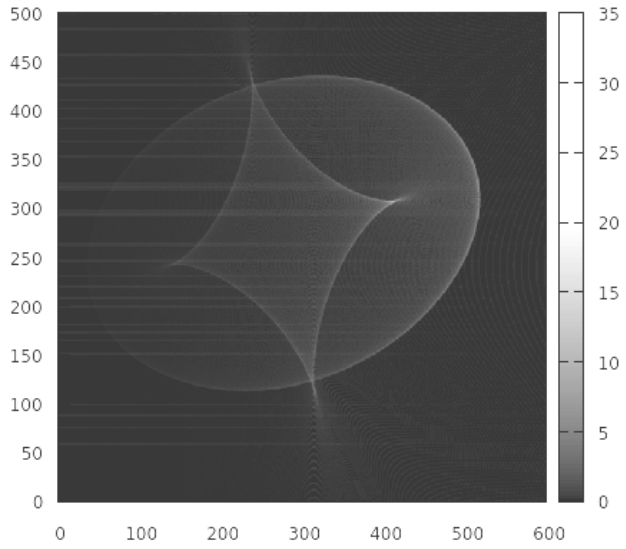


Fig. 10. Imágen original, obtenida del CCD antes de aplicar el AG.

4.1. Representación del problema

En la implementación del AG, cada indiviudo esta compuesto por un vector de 16 elementos, los cuales representan los coeficientes de la ecuacion, que genera la imágen optimizada de las aberraciones del sistema óptico.

La función de aptitud utiliza los polinomios de Zernik, para el coeficiente de correlacion entre un indiviudo y la imágen experimental obtenida del arreglo óptico, donde el coeficiente de correlación representa la aptitud del individuo. Para simular la cruza o reproducción de los individuos, se toman dos individuos y se mezclan a nivel de bits, estableciendo puntos de cruza de manera aleatoria y usandolos puntos para crear rangos, en los cuales la información se intercambia de un indivudo a otro. Esta cruza crea dos individuos nuevos, que contienen la informacion mezclada, de los individuos originales. Para mutar a los descendientes de la población, aleatoriamente se seleccionan algunos individuos y se invierten algunos de sus bits.

5. Resultados

Los resultados presentados a continuación, son un resumen de la ejecución del AG, así como el tiempo de ejecución de cada prueba y la mayor aptitud obtenida. Los parámetros para cada una de las ejecuciones del AG sobre una imágen de 500x500 píxeles son los siguientes:

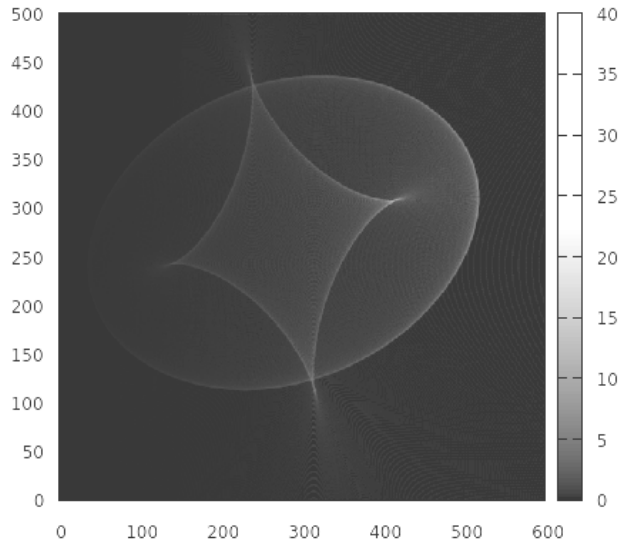


Fig. 11. Imágen de 600x500 píxeles después de aplicar el AG.

- No. de mutaciones: Se elige un número en cada generación de acuerdo al módulo de 6. De acuerdo al resultado del módulo, la cantidad de mutaciones es: 0=5, 1=10, 2=15, 3=20, 4=25, 5=1.
- Puntos de cruza: 100 inicialmente, con una reducción del 10 % a la cantidad por cada 10 generaciones.
- No. de individuos a mutar: comienza con el número total de la población y se reduce 10 % cada 10 generaciones.

La Tabla 1 muestra los resultados del algoritmo aplicado a una imágen de 500x500 píxeles, con una población de 400 individuos y 240 generaciones. El mejor resultado obtenido fue una aptitud de 0.98263 en la ejecución 3 del método de selección simple.

La Tabla **2** muestra los resultados del algoritmo aplicado a una imágen de 600x500 píxeles, con población de 1000 individuos y 400 generaciones. El mejor resultado obtenido, es una aptitud de 0.71741, que corresponde a la primera ejecución usando el método de selección Ruleta.

Como se observa en la figura 9 los AGs convergen muy rápido, pero la respuesta no necesariamente es buena o satisfactoria. La figura 10 muestra la imágen original obtenida del CCD, mientras que la figura 11 muestra el resultado al aplicar el AG, estas imágenes son de la prueba con la ímagen de 600x500 píxeles.

Tabla 1. Resultados de tiempo y aptitud del AG aplicando a una imágen de 500x500 píxeles.

Método	Tiempo(Hrs:min:seg)	Aptitud(por ejecución)
Simple	00:32:23	0.94494
	00:32:04	0.97820
	00:32:25	0.98263
Ruleta	00:31:48	0.59049
	00:32:24	0.96023
	00:32:28	0.83947
Torneo	00:33:11	0.89816
	00:32:30	0.85083
	00:31:51	0.81182
Ranking	00:32:22	0.84102
	00:30:25	0.92106
	00:32:08	0.54901

Tabla 2. Resultados de tiempo y aptitud de la aplicación del AG a una imágen de 600x500 píxeles.

Método	Tiempo(Hrs:min:seg)	Aptitud(por ejecución)
Simple	03:21:03	0.70999
	03:19:15	0.69499
	03:21:28	0.70221
Ruleta	03:18:52	0.71741
	03:21:19	0.61791
	03:20:38	0.67528
torneo	03:19:02	0.69572
	03:21:00	0.70252
	03:19:29	0.61625
Ranking	03:20:20	0.70905
	03:19:04	0.67239
	03:21:04	0.52251

6. Conclusiones

Este trabajo muestra que la implementación de un AG utilizando cómputo de alto rendimiento, puede lograr que el tiempo en el que el algoritmo de obtención de parámetros de la prueba de la estrella obtiene respuestas, sea menor al correspondiente a la implementación en computadoras convencionales. Los AG's al ser un método heurístico, para obtener buenos resultados hay que realizar muchas pruebas y ajustar el algoritmo, algunos ajustes pueden ser la cantidad de mutaciones, número de generaciones y tamaño de la población. Como trabajo futuro se pueden realizar implementaciones que ayuden a mejorar la respuesta del algoritmo, como lo es el cambio al método de optimización, una buena opción puede ser una red neuronal artificial.

Referencias

1. Haupt, R.L., Haupt S.E.: Practical Genetic Algorithms (2nd edition). Wiley-Interscience (2004)

2. Muñoz Duarte, A.: Metaheurısticas. Dykinson (2008)
3. Patterson, D., Hennessy, J.: Computer Architecture: A Quantitative Approach. 4th edition. Morgan Kaufmann, USA (2006)
4. Culler, D.E., Gupta, A., Singh, J. P.: Parallel Computer Architecture: A Hardware/Software Approach. 1st edition. Morgan Kaufmann Publishers, USA (1997)
5. Coulouris, G., Dollimore, J., Kindberg, T., Blair, G.: Distributed Systems: Concepts and Design. 5th edition. Addison-Wesley Publishing Company, USA (2012)
6. Quinn, M.: Parallel Programming in C with MPI and OpenMP. 1st edn. McGraw-Hill (2003)
7. Kirk, D.B., Hwu, Wen-Mei: Programming Massively Parallel Processors, A Hands-on Approach. Morgan Kaufmann (2013)
8. XianLou, H., ShuangYuan, Y.: Image segmentation based on Normalized Cut and CUDA parallel implementation. School of Computer and Information Technology, Bejiing Jiaotong University (2013)
9. Runwei-Cheng, J., Gen, M.: Accelerating genetic algorithms with GPU computing: A selective overview. Computer & Industrial Engineering, Elsevier (2018)

ISSN 1870-4069

Modelo híbrido fonético-neural para corrección en sistemas de reconocimiento del habla

Rafael Viana-Cámara, Mario Campos-Soberanis,
Diego Campos-Sobrino

SoldAI Research,
México

{rviana,mcampos,dcampos}@soldai.com

Resumen. El reconocimiento automático del habla (ASR) es un área relevante en diferentes ámbitos debido a que brinda un mecanismo de comunicación natural entre aplicaciones y usuarios. A menudo los ASR presentan errores en aplicaciones que usan un léxico propio de dominios específicos. Se han explorado diversas estrategias para reducir el error en ASR cerrados mediante posprocesamiento, destacando enfoques de corrección ortográfica y aprendizaje profundo. En el presente artículo se explora el uso de una red neuronal profunda para rectificar los resultados de un algoritmo de corrección fonética, aplicado a una base de datos de audios de televenta. Los resultados obtenidos muestran una reducción de la tasa de error de palabras (WER), tanto en la transcripción original, como en la corrección fonética, mostrando la viabilidad de los modelos de aprendizaje profundo para trabajar en conjunto con estrategias de corrección posprocesamiento en la reducción de errores cometidos por ASR en dominios específicos de lenguaje.

Palabras clave: Reconocimiento del habla, corrección fonética, redes neuronales profundas.

Hybrid Phonetic-neural Model for Correction in Speech Recognition Systems

Abstract. Automatic speech recognition (ASR) is a relevant area in multiple settings because it provides a natural communication mechanism between applications and users. ASRs often fail in environments that use language specific to determined domains. Some strategies have been explored to reduce errors in closed ASRs through post-processing, in particular automatic spell checking and deep learning approaches. In this article, we explore the use of a deep neural network to refine the results of a phonetic correction algorithm, applied to a teleshopping audio database. The results exhibit a reduction in the word error rate (WER), both in the original transcription and in the phonetic correction, that shows the feasibility of deep learning models to work together with

post-processing correction strategies in the reduction of errors made by closed ASRs in specific language domains.

Keywords: Speech recognition, phonetic correction, deep neural networks.

1. Introducción

Si bien los Sistemas de Reconocimiento del Habla (ASR por sus siglas en inglés), han madurado hasta el punto de contar con algunas implementaciones comerciales de calidad, el alto rango de error que presentan en dominios específicos impiden que esta tecnología sea adoptada ampliamente[2]. Lo anterior ha motivado que la corrección en ASR haya sido abundantemente estudiada en la literatura especializada. Los ASR tradicionales se integran de tres módulos relativamente independientes: modelo acústico, modelo de diccionario y modelo de lenguaje[10]. En fechas recientes también han cobrado auge los modelos extremo a extremo de aprendizaje profundo, en los que la división modular de un sistema tradicional no es tan clara [4]. A menudo los ASR en contextos comerciales se distribuyen como cajas negras en donde los usuarios tienen poco o nulo control sobre el modelo de reconocimiento de lenguaje, lo que impide optimizarlos utilizando datos de audio propios. Esto ocasiona que los modelos de poscorrección sean el paradigma utilizado para tratar con los errores producidos por los ASR de propósito general[3]. En entornos especializados de lenguaje donde con frecuencia se encuentran términos fuera de vocabulario, el reconocimiento contextual de palabras es de suma importancia y el grado de personalización de los modelos depende de las capacidades del ASR para adaptarse al contexto. Se ha experimentado con diferentes metodologías para realizar la corrección posprocesameinto de ASR cerrados incluyendo modelos de lenguaje y corrección fonética.

En este artículo se presenta un método para la corrección posprocesamiento en sistemas ASR aplicados a dominios específicos por medio de una red neuronal de memoria a corto y largo plazo (Long Short Term Memory o LSTM) que recibe como atributos de entrada, la salida de un proceso de corrección fonética, la transcripción original del ASR y los hiperparámetros del algoritmo de corrección. A continuación se realza el aporte de la corrección neural para la generación de un algoritmo híbrido que toma en cuenta tanto la corrección fonética, como la poscorrección de la misma, lo cual resulta en una estrategia efectiva para reducir el error en el reconocimiento del habla.

El artículo está estructurado de la siguiente forma: en la sección 2 se describen antecedentes del problema y trabajos relacionados; la sección 3 presenta la metodología utilizada en la investigación; la sección 4 describe el trabajo experimental realizado presentando sus resultados en la sección 5. Finalmente se proporcionan conclusiones y líneas de experimentación para trabajo futuro en la sección 6 del artículo.

2. Antecedentes

El problema de poscorrección en ASR ha sido abordado desde diferentes perspectivas. De manera general podemos hablar de tres diferentes tipos de errores que ocurren en el reconocimiento de audio: sustitución, donde una palabra en el discurso original es transcrita como un palabra diferente; el segundo es el borrado, en el que una palabra con respecto al discurso original no se presenta en la transcripción; y finalmente, inserción en donde una palabra que no aparece en el discurso original aparece en la transcripción [2]. Existen varios esfuerzos de investigación dirigidos a realizar la corrección de los errores en ASR utilizando técnicas de posprocesamiento, en particular una cantidad importante de éstas iniciativas involucra mecanismos de retroalimentación del usuario para aprender patrones de error [2]. Entre las estrategias para aprender esos patrones de error se ha considerado la reducción del problema de poscorrección de los ASR a un problema de corrección de faltas ortográficas.

El artículo [14] propone un modelo de corrección ortográfica basado en transformadores para corregir automáticamente los errores, especialmente aquellos de sustitución realizados por un sistema de reconocimiento de voz del idioma mandarín basado en *Connectionist Temporal Classification*(CTC por sus siglas en inglés). El proyecto se llevó a cabo utilizando los resultados de reconocimiento generados por los sistemas basados en CTC como entrada y las transcripciones de verdad como salida para entrenar un transformador con arquitectura codificador-decodificador, que es muy similar a la traducción automática. Los resultados obtenidos en una tarea de reconocimiento de voz en mandarín de 20,000 horas demuestra que el modelo de corrección ortográfica propuesto en el artículo puede lograr un Character Error Rate (CER) de 3.41 %, lo que resulta en una mejora relativa de 22.9 % y 53.2 % en comparación con los sistemas de línea de base que emplean CTC decodificados con y sin modelo de lenguaje respectivamente presentados en ese mismo artículo.

Una técnica de posprocesamiento versátil basada en la distancia fonética es presentada en [11]. En dicho artículo se integra el conocimiento del dominio con los resultados de ASR de dominio abierto, lo que conduce a un mejor rendimiento. En particular, la técnica presentada es capaz de hacer uso de restricciones de dominio utilizando diversos grados de conocimiento del mismo, que van desde restricciones de vocabulario puro a través de gramáticas o n-gramas hasta restricciones de las expresiones aceptables.

Un modelo de ASR como canal de transformación ruidoso es presentado por Shivakumar et al[10] donde se propone un sistema de corrección capaz de aprender de los errores agregados de todos los módulos independientes que constituyen el ASR e intentar corregirlos. El sistema propuesto utiliza el contexto a largo plazo mediante un modelo de lenguaje de red neuronal y puede elegir de mejor manera entre las posibles transcirpciones generadas por el ASR, así como reintroducir frases previamente podadas o no vistas (que se encuentran fuera del vocabulario). Proporciona correcciones en condiciones ASR de bajo rendimiento sin degradar ninguna transcripción precisa; tales correcciones pueden incluir transcripciones fuera de dominio y no coincidentes.

El sistema expuesto en el artículo proporciona mejoras consistentes sobre el ASR de línea de base, incluso cuando éste se optimiza a través de la restauración del modelo de lenguaje de red neuronal recurrente (RNN). Los resultados demuestran que cualquier mejora de ASR puede explotarse de forma independiente y que el sistema propuesto aún puede proporcionar beneficios en sistemas de reconocimiento altamente optimizados. El beneficio del modelo de lenguaje de la red neuronal se evidencia mediante el uso de 5-gramas que permite una mejora relativa de 1.9 % sobre la línea de base-1).

En el artículo [8] se modela la distorsión en la ortografía de un nombre debido al reconocedor de voz como el efecto de un canal ruidoso. Se sigue el marco de los modelos de traducción de IBM, donde el modelo es entrenado utilizando un texto paralelo de subtítulos y salida automática de reconocimiento de voz. También se realizan pruebas con un método basado en la distancia de edición de cadena. La eficacia de los modelos se evalúa en una tarea de recuperación de consulta de nombre. Los métodos presentados en el artículo dan como resultado una mejora del 60 % en F_1.

Un modelo de incrustación de palabras robusto a ruido es propuesto en [7]. Este supera a los modelos de uso común existentes como fasttext y word2vec en diferentes tareas. Se proponen extensiones para modelos modernos en tres tareas posteriores, es decir, clasificación de texto, reconocimiento de entidades con nombre y extracción de aspectos, estas extensiones muestran una mejora en la robustez al ruido sobre las soluciones existentes para diferentes tareas de NLP.

En [1] se utilizan estrategias de corrección fonética para corregir los errores generados por un sistema ASR. El trabajo citado convierte la transcripción del ASR a una representación en formato de Alfabeto Fonético Internacional (IPA por sus siglas en inglés). Los autores emplean un algoritmo de ventana deslizante para la selección de frases candidatas para su corrección, utilizando una estrategia de selección de candidatos mediante palabras contextuales. Las palabras de dominio específico son provistas mediante un contexto generado manualmente y la distancia de edición entre su representación fonética en formato IPA. Los autores reportan una mejora en el 30 % de las frases reconocidas por el servicio ASR de Google.

En [13] se presenta una extensión del trabajo anterior, experimentando con la optimización del contexto generado mediante algoritmos genéticos. Los autores muestran el desempeño de variantes del algoritmo de corrección fonética utilizando diferentes métodos de representación y selección de candidatos, además de diferentes contextos de palabras evolucionados genéticamente a partir de las transcripciones reales de los audios. De acuerdo a los autores se observó el mejor desempeño del algoritmo de corrección fonética utilizando IPA como representación fonética y una selección incremental por letras, logrando una mejora del WER relativo de un 19 %.

El presente artículo explora un enfoque neural que rectifica las correcciones sugeridas por un algoritmo configurable de corrección fonética. Se experimentó con diversas variantes de configuración del corrector con diferentes representaciones fonéticas de las transcripciones y modificando otros parámetros.

Las correcciones propuestas por este algoritmo se evalúan mediante un clasificador generado utilizando una red neuronal LSTM con salida binaria que indica si se debe aplicar la corrección ofrecida por el algoritmo de corrección fonética. El clasificador recibe como parámetros la transcripción original del ASR, la sugerencia de corrección ofrecida por el algoritmo, así como sus hiper parámetros y calcula una salida binaria. Lo anterior se realiza con la finalidad de reducir el número de correcciones erróneas realizadas por el algoritmo, permitiendo mejorar aún más la calidad de la corrección en enfoques de ASR de caja negra sin necesidad acceder a modelos acústicos o de lenguaje generados por el ASR original.

3. Metodología

Se utilizó un algoritmo correctivo basado en la representación fonética de transcripciones generadas por el sistema de reconocimiento del habla de *Google*. Como fuente de las transcripciones se utilizaron audios recopilados de un sistema de televenta de refrescos el cual se encuentra actualmente en producción, interactuando con usuarios mexicanos. Las transcripciones reales de los ejemplos fueron utilizadas como corpus para generar ejemplos con la transcripción original del ASR, así como la corrección propuesta, etiquetados en forma binaria, donde 1 representa que la corrección propuesta debe realizarse y 0 indica lo contrario. Para el etiquetado se calculó el WER de la transcripción hipotética del ASR y el WER de la corrección propuesta. En ambos casos el WER fue calculado con respecto a la transcripción real generada por un humano y se consideró que la corrección debe realizarse cuando el WER de la versión corregida es menor al WER de la transcripción del ASR. La base de datos fue aumentada con variantes de transcripción producidas por el corrector fonético, al ser utilizado con diferentes parámetros. Esta base de datos aumentada fue utilizada para entrenar un clasificador generado mediante una red neuronal LSTM cuyo objetivo es producir una salida binaria que indica si la corrección propuesta es recomendada.

3.1. Base de datos

Los audios de ejemplo fueron recopilados durante llamadas al sistema de televentas atendido por un agente inteligente. En dichas llamadas los usuarios emitían frases requiriendo diversos productos en diferentes tamaños y presentaciones, así como expresiones naturales de una interaccción de venta (confirmación, precios, etc.). Como parte del proceso se requiere la transcripción de la voz del usuario a texto para su posterior análisis por el sistema, para ello se utiliza el servicio ASR de *Google*. La transcripción real de la frase se realizó por medio de agentes humanos y sirvió como línea de base para evaluar las transcripciones hipotéticas del ASR mediante la métrica *Word Error Rate* (WER), la cual es considerada el estándar para ASR [2].

3.2. Preprocesamiento

Con el objetivo de minimizar el efecto de diferencias lexicográficas, así como facilitar la comparación fonética entre las transcripciones hipotéticas del ASR y las transcripciones reales, fue necesario realizar un preprocesamiento de normalización de texto consistente en: limpieza de símbolos y signos de puntuación, conversión del texto a minúsculas, conversión de números a texto y expansión de abreviaturas.

La etapa de limpieza inicial tiene como objeto la eliminación de ruido existente en las transcripciones, así como la reducción de caracteres a letras y dígitos. Por su parte, las dos últimas etapas del preprocesamiento tienen el efecto de expandir el texto a una forma explícita que facilita su conversión fonética, lo cual ayuda al desempeño del corrector.

3.3. Algoritmo de corrección fonética (PhoCo)

Para el desarrollo de está investigación se utilizó el algoritmo de corrección fonética (PhoCo por sus siglas en inglés) descrito en [1,13], el cual consiste en transformar el texto transcrito a una representación fonética y comparar segmentos de ésta, con versiones fonéticas de palabras y frases comunes en el dominio de aplicación para su posible reemplazo. Estas palabras y frases son denominadas *contexto*. La comparación se realiza utilizando un umbral de similitud de distancia de Levenshtein que determina si una corrección es sugerida o no. La transcripción fonética es un sistema de símbolos gráficos que representan los sonidos del habla humana, es usado como convención para evitar las peculiaridades de cada lengua escrita y representar aquellas lenguas sin tradición escrita [6]. Entre las representaciones fonéticas utilizadas se encuentran la del Alfabeto Fonético Internacional (IPA) y una versión de worldbet (Wbet) [5] adaptada al español de México [12]. De igual manera el algoritmo permite utilizar diferentes estrategias de selección de candidatos. Para éste artículo se utilizaron las configuraciones de ventana deslizante (Win) y selección incremental por caracteres (Let) como son descritos en [13].

3.4. Clasificador neural

Para poder descubrir patrones de error en la corrección fonética, se empleó una red neuronal profunda, que recibe como entrada la transcripción original del ASR, la frase de corrección candidata proporcionada por el PhoCo junto con los hiperparámetros del algoritmo. La salida de la red neuronal es un número binario que indica si se debe realizar la corrección propuesta. Las redes neuronales, en particular las recurrentes, se han utilizado de manera efectiva en tareas de clasificación y descubrimiento de patrones de texto, por lo que se decidió modelar el proceso de rectificación para el algoritmo de corrección fonética mediante una red neuronal. La arquitectura de la red neuronal se diseñó para robustecer la detección de patrones de palabras y el seguimiento de dependencias a corto y largo plazo, por lo que se generó una topología compuesta de la siguiente forma:

– Una capa de *embeddings* de tamaño 128,
– Una capa LSTM de 60 unidades ocultas,
– Una capa de *Max pooling*,
– Una capa densa de 50 unidades ocultas,
– Una capa densa de activación sigmoide de 1 unidad.

La arquitectura utilizada se ilustra en la Fig. 1, en donde se muestra el procesamiento de las diferentes capas de la red hasta producir una salida binaria, mediante una única neurona con activación sigmoide.

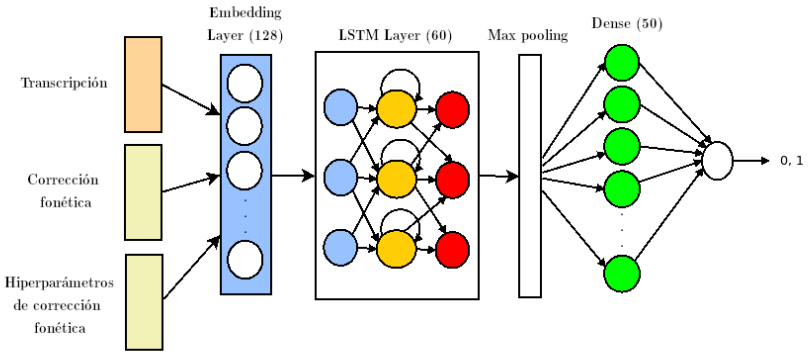


Fig. 1. Modelo del clasificador neural.

En primer lugar se tiene una capa de entrada que recibe la representación indexada del diccionario de palabras de la frase hipotética del ASR, así como de la sugerencia de corrección, y un valor numérico que indica el umbral utilizado por el PhoCo para producir su corrección candidata. Estas entradas son pasadas a una capa de *embeddings*, la cual añade una representación densa de las palabras que captura propiedades sintácticas y semánticas, las cuales han demostrado ser de utilidad en una gran cantidad de tareas de Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) [9]. A continuación se mandan las representaciones densas a una capa LSTM la cual posee propiedades importantes en el manejo de dependencias a largo plazo gracias a sus compuertas internas de actualización y olvido, que resultan de suma utilidad en la detección de patrones secuenciales de texto.

La capa de *Max pooling* funciona como un mecanismo de atención simplificado, muestreando las dependencias y entidades con mayor activación provenientes de la LSTM, propiciando la detección de características importantes en diferentes posiciones en el texto, lo cual ayuda a reducir la cantidad de datos necesarios para entrenar el modelo. A continuación se pasa por una capa densa

completamente conectada de 50 neuronas con activaciones RELU para calcular funciones compuestas por las características más relevantes muestreadas de la LSTM. Finalmente se pasa a una capa de salida de una sola neurona con función de activación sigmoide, la cual es recomendada para clasificación binaria. Se utilizó una función de pérdida de entropía cruzada binaria y se optó por una estrategia de optimización ADAM para ajustar la tasa de aprendizaje de manera adaptativa.

3.5. Algoritmo híbrido fonético-neural

El algoritmo híbrido se realizó ejecutando la corrección neuronal descrita en la sección 4.3 al algoritmo de corrección fonética, presentado en la sección 4.2. La idea central de este proceso es proveer un mecanismo de control para las posibles sustituciones erróneas que pudiera realizar el algoritmo de corrección fonética. Este enfoque permite adoptar estrategias más agresivas de corrección al poder configurar el umbral del algoritmo de corrección fonética estándar a un valor mayor y controlar los posibles errores de corrección (falsos positivos). El algoritmo consiste en realizar la corrección fonética de manera estándar y después evaluar la corrección candidata, junto con la transcripción original del ASR y los hiperparámetros del algoritmo fonético en el clasificador neural. Si el clasificador neural predice un valor mayor a 0.5 se procede a la corrección, en caso contrario se utiliza la transcripción del ASR.

4. Experimentación

En la presente sección se muestran los métodos usados para el entrenamiento del clasificador neural, la experimentación con la versión clásica del algoritmo de corrección fonética y la versión híbrida utilizando la salida del clasificador neural como factor de decisión para aceptar la corrección fonética propuesta, ilustrando los diferentes mecanismos implementados, según se describió en la sección 3 del documento.

4.1. Conjuntos de datos

Como fuente de datos para la experimentación se utilizó un total de 320 archivos de audio. Para cada uno de los audios se generaron dos transcripciones usando el ASR de Google con y sin contexto las cuales fueron almacenadas en una base de datos, conteniendo además la transcripción hecha de forma manual. Así, la base de datos contiene para cada audio dos frases hipotéticas generadas por el ASR y su transcripción real para efectos de evaluar el sistema. A continuación se realizaron diferentes hipótesis de corrección para cada ejemplo de audio utilizando diversas configuraciones del PhoCo, variando los parámetros de umbral entre 0.0 y 0.6 con un paso de 0.5, el tipo de representación utilizando IPA, texto simple y Wbet y el método de búsqueda utilizando estrategias de ventana deslizante y selección incremental de caracteres. De esta manera se

generaron 144 posibles correcciones para cada audio con lo que se consiguió una base de datos aumentada de 46,080 ejemplos para entrenar el clasificador neural. Las configuraciones listadas en la tabla se describen en [13]. Se añadió una etiqueta binaria, establecida a 1 cuando el WER de la corrección propuesta es menor al de la hipótesis del ASR y 0 en caso contrario. Los registros establecidos a 1 indican que la corrección propuesta afecta positivamente al WER.

4.2. Corrección fonética

Cada transcripción producida por el ASR en los datos de entrenamiento fue utilizada como fuente para un procedimiento de posprocesamiento correctivo basado en la transcripción fonética de texto. Dicho método de corrección fue utilizado con diferentes variantes y parámetros con lo que se obtuvieron para cada transcripción de ejemplo múltiples resultados, los cuales fueron registrados en la base de datos de entrenamiento aumentada con la estrategia presentada en la sección 4.1.

En los experimentos el parámetro de umbral fue variado usando una técnica de *GridSearch* en el rango de 0 a 0.6 en pasos de 0.05. Para el modo de representación se usaron tres variantes: IPA, texto simple y Wbet. Estas variantes en los parámetros para el corrector fonético dieron origen a variaciones en los resultados que se acumularon en la base de datos.

4.3. Clasificador neural

Para el entrenamiento del clasificador neural se dividió la base de datos aumentada descrita en la sección 4.1, en particiones aleatorias de entrenamiento, validación y prueba en porcentajes de 80 % para entrenamiento, 10 % para validación y 10 % para prueba. Se utilizó el conjunto de entrenamiento para generar diferentes modelos de redes neuronales, observando métricas de exactitud, precisión y exhaustividad, sobre los conjuntos de entrenamiento y validación, así como el área bajo la curva (AUC por sus siglas en inglés) característica operativa del receptor (ROC por sus siglas en inglés), la cual presenta un balance entre la tasa de verdaderos y falsos positivos y ofrece una métrica de desempeño para sistemas de clasificación. Se iteró sobre diferentes modelos empleando técnicas de regularización por abandono (*dropout*), con diferentes parámetros de probabilidad. Una vez se obtuvo el mejor modelo en el conjunto de validación, se procedió a evaluar el mismo en el conjunto de datos de prueba, para reportar las métricas de exactitud (*accuracy*), precisión (*precision*), exhaustividad (*recall*) y F_1 presentadas en la sección 5.1 del presente artículo. Los modelos fueron implementados utilizando tensorflow 2.0 y keras, implementados sobre un sistema opreativo Debian GNU/Linux 10 (buster) x86_64, provisto con una GPU Nvidia GTX 1080 TI de 11 GB.

4.4. Algoritmo híbrido fonético-neural

La experimentación con el algoritmo fonético neural, se realizó una vez entrenado el clasificador neural. Se procedió a examinar exhaustivamente con todos

los ejemplos de la base de datos el WER individual de cada una de las frases del ASR, del candidato a corrección fonética y de la salida del modelo fonético neural con respecto a la transcripción real. A continuación se procede a analizar el WER promedio de las frases para cada uno de los diferentes umbrales utilizados para generar la corrección fonética. En los resultados presentados en la sección 5.2 se reporta el respectivo WER promedio, así como las diferentes reducciones relativas del WER con respecto a la transcripción original.

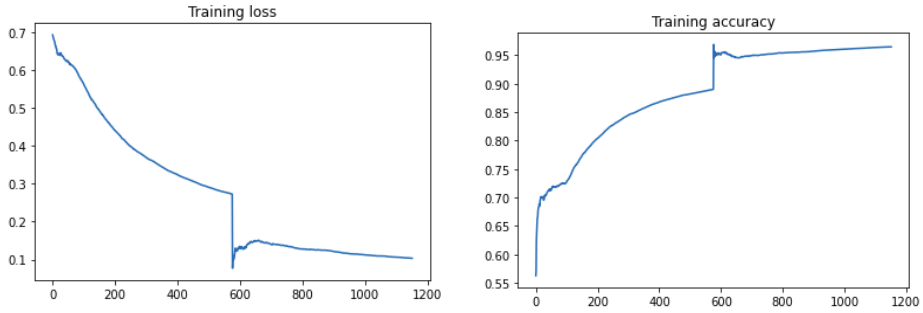
5. Resultados

En la presente sección se muestran los resultados del entrenamiento del clasificador neural, así como los comparativos entre la versión clásica del algoritmo de corrección fonética y la versión híbrida, ilustrando los diferentes valores WER promedio obtenidos de la transcripción del ASR, la corrección fonética y la corrección fonética-neural.

5.1. Clasificador neural

La red neuronal profunda se entrenó durante dos épocas con una técnica de *mini-batch* de tamaño 64, utilizando 36,863 datos obtenidos con los procedimientos descritos en la sección 4.1 y 4.3.

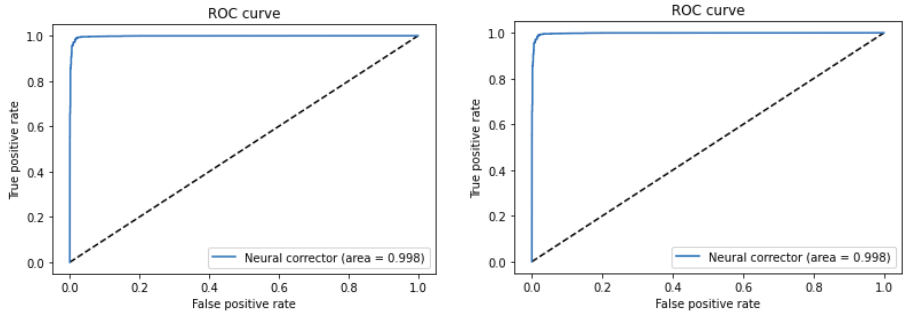
En la Fig. 2 se muestran las gráficas de la función de pérdida y la exactitud del modelo después del entrenamiento de cada uno de los lotes. La función de pérdida muestra algunas irregularidades debido a las particularidades de los diferentes lotes, sin embargo, se puede apreciar un descenso consistente en el error, en particular se nota una caída brusca alrededor del lote 550 hasta estabilizarse cerca del valor 0.1034. Un proceso similar ocurre con la exactitud de la red neuronal, la cual muestra un crecimiento sostenido, con un salto abrupto alrededor del lote 550, estabilizándose cerca del 0.9646.



(a) Pérdida (b) Exactitud

Fig. 2. Función de pérdida (a) y exactitud (b) en el entrenamiento de la red neuronal.

Una vez entrenado el mejor modelo neuronal obtenido de las diferentes fases de iteración se realizó la evaluación del mismo visualizando el área bajo la curva ROC cubierta por el modelo cuando realiza predicciones sobre los conjuntos de validación y prueba, misma que se ilustra en la Fig. 3 en donde se puede notar que se obtuvieron resultados satisfactorios cubriendo un 99 % del área.



(a) AUC validación (b) AUC prueba

Fig. 3. Área bajo la curva ROC para los conjunto de validación (a) y prueba (b).

Utilizando el modelo entrenado se calcularon métricas de exactitud, precisión, exhaustividad, así como el score F_1, utilizando el conjunto de prueba. Los resultados para las diferentes clases (0 y 1), así como el promedio realizado con la estrategia *macro average* el cual representa el promedio entre las diferentes clases. Se obtuvieron altos valores para todas las métricas superando el 95 % en cada una de ellas. El conjunto de prueba constaba de un 10 % del total de los datos traducido en 4,607 ejemplos de prueba. Los valores obtenidos para cada una de las métricas de evaluación de la red neuronal se muestra en la tabla 1, en donde llama particularmente la atención, el valor *macro average* F_1 el cual es de 98 %, siendo este un indicador de una alta eficiencia para el modelo del clasificador neural.

Tabla 1. Métricas de evaluación sobre el conjunto de datos de prueba.

Clase	**Precisión**	**Exhaustividad**	**F_1 score**	**Soporte**
0	0.99	0.99	0.99	3302
1	0.96	0.97	0.97	1305
Macro average	**0.98**	**0.98**	**0.98**	**4607**

5.2. Algoritmo híbrido fonético-neural

Los resultados de la experimentación descrita en la sección 4.3 se presentan a continuación. Los WER promedio de acuerdo a los diferentes umbrales son tomados del total de los 46,080 ejemplos. Cada uno de los valores de los umbrales utilizados para experimentar cuenta con 3,840 ejemplos. En la tabla 2, se muestran los WER promedio para los distintos umbrales así como la reducción relativa del WER para el algoritmo híbrido fonético-neural. La línea base obtenida mediante el ASR de *Google*, presentó un WER de 0.338, por lo que las reducciones relativas se realizan tomando ese valor como referencia.

Tabla 2. WER promedio y WER relativo del corrector fonético (PhoCo) y el modelo híbrido con relación al WER del ASR de Google.

Umbral	WER PhoCo	WER híbrido	$\mathbf{WER_{rel}}$ Google	$\mathbf{WER_{rel}}$ PhoCo
0.05	0.235	0.236	30.5 %	-0.1 %
0.10	0.235	0.236	30.5 %	-0.1 %
0.15	0.229	0.228	32.6 %	0.3 %
0.20	0.228	0.228	32.8 %	0.3 %
0.25	0.219	0.219	35.5 %	0.3 %
0.20	0.216	0.211	37.8 %	2.3 %
0.35	0.211	0.205	39.5 %	3.0 %
0.40	0.208	0.201	40.7 %	3.2 %
0.45	0.230	0.190	43.9 %	17.5 %
0.50	0.235	0.191	43.7 %	18.6 %
0.55	0.338	0.227	32.9 %	32.6 %
0.60	0.374	0.232	31.5 %	37.9 %
Promedio	**0.247**	**0.217**	**36.0 %**	**9.7 %**

A partir de los resultados presentados se observa que en configuraciones con umbrales pequeños (0.05 y 0.10) el WER relativo con respecto al algoritmo fonético original reduce, por lo cual el uso del clasificador neural no resulta una buena estrategia para realizar la corrección final, sin embargo, a partir de un umbral de 0.15 en adelante, muestra una mejora consistente con respecto al algoritmo fonético original, la cual aumenta notablemente conforme el valor del umbral crece, llegando a un máximo cuando el umbral también lo es y logrando una reducción del WER relativo con respecto a la versión fonética estándar de 37.9 %.

El WER relativo a la hipótesis proporcionada por el ASR de *Google*, muestra una reducción consistente, llegando a una reducción máxima de 43.9 % con un umbral del PhoCo establecido a 0.45. El algoritmo híbrido muestra reducciones consistentes del WER relativo tanto del ASR como de la transcripción fonética simple, exhibiendo una mejora promedio de 36 % y 9.7 % respectivamente. De igual manera el modelo híbrido logró obtener el WER mínimo con el umbral establecido a 0.45, disminuyendo el WER hasta 0.19, lo cual en comparación al

WER promedio del ASR de *Google*, representa una mejora del 14.8 % del WER absoluto y una del 43.9 % en términos relativos.

6. Conclusiones y trabajo futuro

A partir de los resultados obtenidos en la experimentación, se muestra la utilidad del algoritmo híbrido de corrección fonética-neural para reducir los errores de la transcripción de *Google*. Se observa que el algoritmo híbrido logra reducir el WER relativo hasta en un 43.9 %.

Se muestra una mejora consistente del algoritmo de corrección fonético-neural tanto sobre la transcripción del ASR de *Google*, como del algoritmo de corrección fonética simple. Se observó una reducción promedio del WER del algoritmo fonético simple de 9.7 %.

Las redes neuronales profundas, resultaron una excelente estrategia para el modelado de patrones de lenguaje en dominios específicos, exhibiendo un F_1 score de 0.98, así como un 99 % de área bajo la curva ROC.

Los aportes del clasificador neural resultan más notorios para valores del umbral de corrección fonética más altos, permitiendo configuraciones más agresivas para este algoritmo de corrección. Inclusive en esquemas donde el algoritmo fonético simple reduce su rendimiento debido a la corrección de ejemplos con falsos positivos, el uso a posteriori del clasificador neural resulta de utilidad para mantener un WER bajo en comparación al ASR de *Google*, como se puede observar en la tabla 2.

El corrector fonético es una estrategia viable para la corrección de errores en ASR comerciales alcanzando una mejora del WER relativo del 40.7 % con un umbral de 0.40. Con la aplicación del clasificador neural y el algoritmo híbrido se consigue reducir aún más el WER utilizando un umbral de 0.45 para el PhoCo consiguiendo una mejora en el WER relativo de 43.9 %. Este tipo de mejoras resultan importantes en ASR de uso comercial donde se necesitan grados de precisión cada vez más altos.

Dado que la arquitectura de corrección es independiente del sistema usado para la transcripción y del dominio de aplicación, la estrategia descrita es susceptible de ser extendida a diferentes sistemas ASR y dominios de aplicación. Es necesario, sin embargo, entrenar un clasificador neural para cada uno de los diferentes dominios, por lo que no se puede usar este enfoque para transferencia de conocimiento.

Los resultados encontrados muestran que es posible implementar una estrategia híbrida fonético-neural para la poscorrección de ASR en tiempo casi real. Dado que tanto el algoritmo de corrección fonética como el clasificador neural, son modelos computacionales susceptibles a escalamiento, se pueden emplear técnicas de integración de servicios web para realizar la poscorrección en sistemas ASR comerciales existentes.

Entre las líneas de investigación a futuro se requiere validar los resultados con corpus de diferentes dominios de aplicación, además se prevé la experimentación utilizando diferentes parámetros de la corrección fonética, incluyendo el

contexto y la incorporación de características del audio original. Otra línea de investigación es la comparación con algoritmos de aprendizaje profundo extremo a extremo, en donde un modelo neuronal profundo genere la corrección del ASR directamente.

Agradecimientos. A Carlos Rodrigo Castillo Sánchez, por su valiosa contribución al proporcionar la infraestructura para la experimentación del presente artículo.

Referencias

1. Campos-Sobrino, D., Campos-Soberanis, M., Martínez-Chin, I., Uc-Cetina, V.: Corrección de errores del reconocedor de voz de google usando métricas de distancia fonética. Research in Computing Science 148(1), 57–70 (2019)
2. Errattahi, R., Hannani], A.E., Ouahmane, H.: Automatic speech recognition errors detection and correction: A review. Procedia Computer Science 128, 32–37 (2018), http://www.sciencedirect.com/science/article/pii/S1877050918302187, 1st International Conference on Natural Language and Speech Processing
3. Feld, M., Momtazi, S., Freigang, F., Klakow, D., Müller, C.: Mobile texting: Can post-asr correction solve the issues? an experimental study on gain vs. costs. International Conference on Intelligent User Interfaces, Proceedings IUI (05 2012)
4. He, Y., Sainath, T.N., Prabhavalkar, R., McGraw, I., Alvarez, R., Zhao, D., Rybach, D., Kannan, A., Wu, Y., Pang, R., Liang, Q., Bhatia, D., Shangguan, Y., Li, B., Pundak, G., Sim, K.C., Bagby, T., Chang, S., Rao, K., Gruenstein, A.: Streaming end-to-end speech recognition for mobile devices. CoRR abs/1811.06621 (2018), http://arxiv.org/abs/1811.06621
5. Hieronymus, J.L.: Ascii phonetic symbols for world's languages: worldbet. Technical report, Bell Labs (1993)
6. Hualde, J.: The sounds of Spanish. Cambridge University Press (2005)
7. Malykh, V.: Robust to noise models in natural language processing tasks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 10–16. Association for Computational Linguistics, Florence, Italy (Jul 2019), https://www.aclweb.org/anthology/P19-2002
8. Raghavan, H., Allan, J.: Matching inconsistently spelled names in automatic speech recognizer output for information retrieval. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (01 2005)
9. Ruder, S.: A survey of cross-lingual embedding models. CoRR abs/1706.04902 (2017), http://arxiv.org/abs/1706.04902
10. Shivakumar, P.G., Li, H., Knight, K., Georgiou, P.G.: Learning from past mistakes: Improving automatic speech recognition output via noisy-clean phrase context modeling. CoRR abs/1802.02607 (2018), http://arxiv.org/abs/1802.02607
11. Twiefel, J., Baumann, T., Heinrich, S., Wermter, S.: Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing. In: Proceedings of the National Conference on Artificial Intelligence. vol. 2, pp. 1529–1535 (07 2014)

12. Varela, A., Cuayáhuitl, H., Nolazco-Flores, J.A.: Creating a Mexican Spanish Version of the CMU Sphinx-III Speech Recognition System. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) Progress in Pattern Recognition, Speech and Image Analysis. pp. 251–258. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
13. Viana-Cámara, R., Campos-Sobrino, D., Campos-Soberanis, M.: Optimización evolutiva de contextos para la corrección fonética en sistemas de reconocimiento del habla. Research in Computing Science 148(8), 293–306 (2019)
14. Zhang, S., Lei, M., Yan, Z.: Automatic spelling correction with transformer for ctc-based end-to-end speech recognition. ArXiv abs/1904.10045 (2019)

ISSN 1870-4069

Redes neuronales evolutivas con modelos de Lotka-Volterra

Fernando Javier Aguilar Canto

Universidad Autónoma de Yucatán
Facultad de Matemáticas,
México pherjev@gmail.com

Resumen. El entrenamiento de redes neuronales multicapa supone de dos partes diferenciadas: un ajuste de los pesos y una optimización de los hiperparámetros, tales como el número de capas y el número de neuronas en cada una de las mismas, la cual ha sido tradicionalmente elaborada de forma manual. En el presente artículo proponemos un algoritmo genético basado en el modelo ecológico de Lotka-Volterra, cuyo objetivo es permitir la evolución de la arquitectura de las capas densas (completamente conectadas) de la red para lo cual se concibieron dos variantes principales: aplicando cambios constantes en la estructura de las redes hijas (mutaciones constantes) o mediante cambios aleatorios. De forma experimental se observó que el empleo de aleatoriedad en el modelo arroja mejores resultados. Asimismo, fue posible mejorar una red convolucional dada optimizando sus capas densas y, las pruebas realizadas en la base de datos EMNIST lograron alcanzar 93.99 % de exactitud para el problema de clasificación de caracteres alfabéticos, mejorando a la red originalmente concebida. Finalmente, ofrecemos una prueba de la convergencia local del algoritmo propuesto, el cual además presenta un costo computacional polinomial, haciendo de nuestra propuesta una opción válida para el desarrollo de arquitecturas de forma automatizada.

Palabras clave: Redes neuronales prealimentadas, algoritmo evolutivo, redes neuronales evolutivas, redes neuronales convolucionales, optimización hiperparamétrica.

Evolutionary Neural Networks using Lotka-Volterra Models

Abstract. The training of multilayer neural network involves two different parts: an adjustment of the weights and an optimization of the hyperparameters, such as the number of layers and the number of neurons in each of them, which has been traditionally made manually. In this article, we propose a genetic algorithm based on the ecological Lotka-Volterra model, whose objective is to allow the evolution of the architecture of the dense (completely connected) layers of the network. For this, two main

variants were conceived: applying constant changes in the structure of the daughter-networks (constant mutations) or by means of random changes. Experimentally, it was observed that the use of randomness in the model yields better results. Also, it was possible to improve a given convolutional network by optimizing its dense layers and, the tests carried out in the EMNIST database achieved 93.99 % accuracy for the problem of classification of alphabetic characters, improving the network originally conceived. Finally, we offer a test of the local convergence of the proposed algorithm, which also presents a polynomial computational cost, making our proposal a valid option for the development of architectures in an automated way.

Keywords: Feedforward neural networks, evolutionary algorithm, evolutionary neural networks, convolutional neural networks, hyperparametric optimization.

1. Introducción

Las Redes Neuronales Prealimentadas (mejor conocidas como *Feedforward Neural Networks* o FNN) han sido exitosamente aplicadas en distintos contextos como Visión Computacional y Reconocimiento de Voz [12]. En gran medida, el éxito de la red depende de la arquitectura para abordar el problema específico. No obstante, en general, tanto para la elección de la arquitectura como el resto de los hiperparámetros, se carece de una fórmula explícita para su configuración óptima [22].

El presente trabajo aborda el problema de la optimización hiperparamétrica, en particular para el caso del diseño de las capas completamente conectadas, las cuales se han empleado en importantes redes neuronales profundas como las VGGs propuestas en [20]. El algoritmo presentado en este trabajo sugiere desarrollar un ecosistema artificial de redes neuronales, asignándoles un rol de presas siguiendo el modelo de Lotka-Volterra para efectuar su evolución mediante cambios graduales en su arquitectura. Con los depredadores eliminando a los ejemplares con menor exactitud sobre el conjunto de validación, las redes mejor adaptadas pueden preservar su estructura para las generaciones futuras. Siguiendo esta inspiración biológica, podemos desarrollar comunidades estables de redes neuronales que progresivamente vayan mejorando su respuesta al entorno, es decir, a los datos, para poder, a la postre, obtener una arquitectura adecuada para la resolución de un problema específico.

1.1. Redes Neuronales Prealimentadas

Las Redes Neuronales Prealimentadas consisten en varias capas de neuronas artificiales que realizan una operación de producto punto con sus entradas $\mathbf{x}$ (típicamente de la capa anterior o de la entrada del modelo) con los pesos $\mathbf{w}$ de la neurona, para luego aplicar una función de activación f, de forma que

calculan $y = f(\mathbf{w} \cdot \mathbf{x})$. La disposición en varias capas de neuronas sin conexiones recurrentes forma a las *redes multicapa prealimentadas* [1]. Una red neuronal prealimentada completamente conectada consta de una capa de entrada (*input layer*) y una capa de salida (*output layer*), los cuales dependen del problema modelado. De manera libre aparece tanto el número de capas ocultas (*hidden layers*) como el número de neuronas en cada una, siendo hiperparámetros de la red neuronal [22].

Los Teoremas de Aproximación Universal [10,15] sobre redes neuronales completamente conectadas, básicamente enuncian que una red neuronal de tipo FNN completamente conectada con una capa intermedia con suficientes neuronas pueden aproximar a cualquier función continua si la función de activación empleada no es un polinomio algebraico. Otros resultados matemáticos muestran la importancia de aumentar el número de capas frente al número de neuronas, lo cual, aunado al desarrollo de *Deep Learning* en términos prácticos nos inclina a considerar aumentar tal hiperparámetro [12] y no exclusivamente efectuar un ajuste con una red de una capa intermedia. Por ende, los modelos desarrollados deberán considerar tanto un aumento de neuronas en una capa como de número de capas.

1.2. Antecedentes

Las Redes Neuronales Evolutivas (*Evolutionary Neural Networks*) son un conjunto de estrategias adaptativas inspiradas en los mecanismos de selección natural con el objetivo de optimizar un aspecto de una red neuronal artificial. Reconocemos dos niveles básicos en los que podemos realizar optimización en redes neuronales: un nivel *paramétrico*, propio de los pesos adaptables de la red neuronal y un nivel *hiperparamétrico*, dirigido a las modificaciones no realizadas en los pesos de la red, sino en aspectos meramente externos como la arquitectura (número de capas intermedias con el número de neuronas), las funciones de activación (ReLU contra sigmoide, por ejemplo), los parámetros propios de los optimizadores (como la tasa de aprendizaje o el *momentum* es los gradientes acelerados [8]), entre otros aspectos.

El concepto de evolución ha sido aplicado en redes neuronales tanto a nivel de pesos (paramétrico, véase [18]) con una arquitectura fija, como a nivel hiperparamétrica [16]. Por dos razones centraremos el estudio en el caso de los hiperparámetros. La primera radica en el estado de solución de la optimización hiperparamétrica frente a la paramétrica. Métodos basados en el gradiente (tal como Adam) han demostrado su efectividad en el entrenamiento de los pesos de las redes neuronales [17]. Desde la formulación del Perceptrón en 1958 [19], se disponen de diferentes métodos para entrenar los pesos de la red neuronal. Sin embargo, tradicionalmente, la optimización hiperparamétrica se ha logrado de forma empírica [5,4]. Una segunda razón para centrarse en la optimización hiperparamétrica radica en su motivación biológica. De la Teoría de Hebb, se considera que los pesos de las neuronas reales son adaptables [2], mediante mecanismos de plasticidad neuronal [11], lo cual descarta que se ajusten por generaciones. Lo

que pudiese haber sido moldeado por la evolución es la arquitectura de las redes neuronales.

De acuerdo con Young *et al* [22], en la actualidad, se disponen de tres formas comunes para efectuar la selección de los hiperparámetros, las cuales, junto con otras propuestas, son la *búsqueda manual*, la *búsqueda por rejilla* (consiste en aplicar aumentos y decrementos constantes para explorar el espacio de los hiperparámetros, aunque tiene la desventaja de ser poco eficiente, en especial en espacios multidimensionales), la *búsqueda aleatoria* (propuesta por Bergstra y sus colaboradores en ([5,4]) consistente en modificar a los incrementos constantes por aleatorios, obteniendo mejores resultados) y los *algoritmos genéticos*.

Mención aparte merecen los llamados *algoritmos genéticos*, los cuales constan de tres fases diferenciadas: selección, cruzamiento y mutación. Algunas de estas fases, como selección y mutación son recogidas en el presente artículo. Estos algoritmos han sido implementados en el contexto de la optimización de arquitecturas de redes neuronales, en particular para el contexto de Redes Neuronales Convolucionales en [22,21], optimizando otros hiperparámetros como el número de filtros. Para el caso de reconocimiento de caracteres (que es abordado en este trabajo), existen propuestas de evolución centradas a las capas convolucionales que han sido aplicadas a las bases de datos MNIST y EMNIST. Entre dichas propuestas destacan [6,7] para MNIST y [3] para EMNIST.

2. Metodología

2.1. FNN evolutivas

Como se especificó en anteriormente, una optimización metaparamétrica de una FNN consiste en ajustar el número de capas y neuronas intermedias hasta lograr un mayor rendimiento de la red sobre el conjunto de validación. Las Redes FNNs pueden expresarse de manera abstracta como una sucesión $(s_1, s_2, \ldots, s_n, 0, \ldots)$ donde cada s_i representa el número de neuronas de la capa oculta i. Como el número de capas n es finito, entonces $s_i = 0$ para $i > n$. Por fines representativos, expresaremos la sucesión como $(s_1, \ldots, s_n)$ en lugar de la forma anteriormente presentada.

A la sucesión anteriormente presentada se le definirá como el *código genético* de la FNN. La razón del nombre obedece a que es posible introducir operaciones (*mutaciones*) al código con la finalidad de hallar una arquitectura apropiada, mediante una fisión binaria en dos redes hijas (que pueden ser entendidos como *organismos* en una *ecología artificial*). Tales mutaciones pueden ser de tres tipos:

– α: Agrega k neuronas a una capa aleatoriamente escogida:

$$(r_1, \ldots, r_i, \ldots, r_n) \to (r_1, \ldots, r_i + k, \ldots, r_n). \tag{1}$$

– β: Agrega una capa con k neuronas:

$$(r_1, \ldots, r_n) \to (r_1, \ldots, r_n, k). \tag{2}$$

– γ: Reduce k neuronas en una capa específica o la vuelve 0:

$$(r_1, \ldots, r_i, \ldots, r_n) \to (r_1, \ldots, \text{máx}(r_i - k, 0), \ldots, r_n). \tag{3}$$

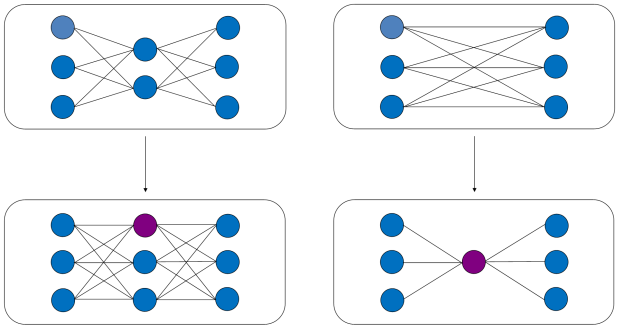


Fig. 1. Mutación de tipo α (izquierda) y β (derecha) con $k = 1$.

La elección de un tipo de mutación u otro será aleatorio, con una probabilidad p_x de elegir una mutación tipo x. Debido a que se tiene una ausencia de deleciones, se aconsejará iniciar una población de redes con el organismo (0).

De esta forma, el organismo inicial se "duplica" para crear a los organismos (0) y (k) hijos, los cuales se vuelven a duplicar para producir una nueva generación de organismos de forma recursiva (véase figura 2).

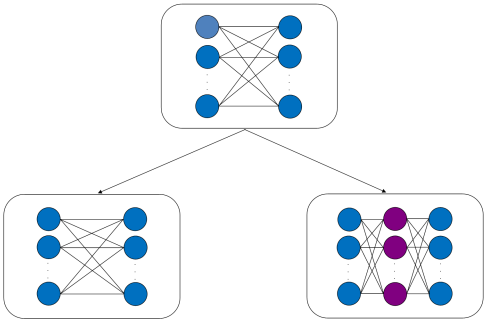


Fig. 2. Duplicación del organismo (0) en (0) y (k).

2.2. Modelo de Lotka-Volterra

Es posible efectuar una optimización simplemente efectuando sucesivas duplicaciones de organismos y tomar el rendimiento máximo de la última generación. Sin embargo, tal algoritmo tiene complejidad exponencial sobre el número m de generaciones ($O(m) = 2^m$). Para ello, introduciremos un *depredador artificial* cuyo fin será estabilizar la población de organismos, mediante la eliminación directa de los organismos más "débiles", *i.e.*, con menor rendimiento [1].

El número de depredadores y presas puede ser controlado mediante el modelo de Lotka-Volterra. Dado que se trata de un modelo abstracto, para fines prácticos únicamente necesitamos regular la población de organismos presa mediante "depredaciones" periódicas.

El clásico modelo de Predador-Presa de Lotka-Volterra es un sistema de dos ecuaciones diferenciales que representan la interacción ecológica entre dos especies, las cuales toman los roles de depredador y presa. Las ecuaciones están dadas por:

$$\frac{dx}{dt} = af(x) - bg(x,y)y, \tag{4}$$

$$\frac{dy}{dt} = cg(x,y)y - hy, \tag{5}$$

donde $x(t)$ denota la densidad de presas sobre el tiempo t, $y(t)$ es la densidad de depredadores sobre el tiempo, $f(x)$ es la razón de crecimiento poblacional de las presas en ausencia de depredadores, $g(x,y)$ representa a la respuesta funcional del depredador frente a su presa, a es la tasa de crecimiento de la población, b es la tasa de competencia, c representa la tasa de crecimiento del depredador y h a la tasa de muerte del depredador [13]. Considerando un modelo de crecimiento exponencial para la población de presas, se tiene que $f(x) = x$ y también podemos considerar una interacción $g(x,y) = x$.

Para nuestro modelo en particular, podemos tomar otras consideraciones que nos permiten fijar algunos parámetros del modelo. En ausencia de depredadores, el crecimiento de la presa está dado por $\frac{dx}{dt} = ax$. Como la reproducción de nuestros organismos presas se da por fisión binaria, entonces una solución propuesta a la ecuación diferencial es $f(t) = 2^t$, de donde observamos que $a = \ln 2$.

De esta forma, el modelo final está dado por las ecuaciones:

$$\frac{dx}{dt} = \ln 2 \; x - bxy, \tag{6}$$

$$\frac{dy}{dt} = cxy - hy. \tag{7}$$

Una simulación del modelo puede observarse en la figura 3.

[1] Una variante estocástica puede consistir en eliminar a los organismos con probabilidad en función de su rendimiento

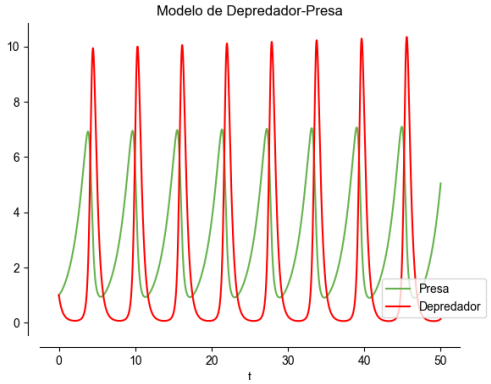


Fig. 3. Simulación del modelo de Lotka-Volterra utilizando método de Euler.

2.3. Discretización del modelo

Los parámetros b, c y h son libres para el sistema de las ecuaciones (5 y 6). Eligiendo adecuadamente los parámetros es posible efectuar una simulación de la dinámica poblacional que implica el aumento de la población presa hasta su declive.

De la simulación podemos observar que las soluciones analíticas del sistema son periódicas y dadas ciertas condiciones iniciales y parámetros predefinidos podemos calcular la amplitud, el periodo T, el máximo M y el mínimo m, dados por $M = \arg\max_{t\in[0,T]} x(t)$ y $m = \arg\min_{t\in[0,T]} x(t)$

La periodicidad de las soluciones es un hecho probado (veáse [14]). Los nuevos parámetros T, m, M son suficientes para realizar una aproximación discreta a la dinámica de la poblacón de presas, que es la que nos interesa. Tal aproximación será dado por los siguientes pasos:

1. Aumentar la población de presas exponencialmente utilizando $f(t) = 2^t$ (fisión binaria) hasta que $t = M$
2. Cuando $t = m$ reducir el número de presas hasta eliminar μ_m.
3. Aplicar $m \leftarrow m + T$ y $M \leftarrow M + T$ y repetir el paso 1.

Aquí $\mu_m = x(m)$, El anterior algoritmo nos permite disponer de una forma aproximada y estable para simular dicha dinámica. Más aún, podemos generalizar los parámetros T, m, M, μ_m, y establecerlos como iniciales en lugar de b, c y h. Una simplificación más extrema, puede fijar $m = M + 1$, de forma que en la siguiente iteración se efectúe la eliminación. De este modo, un algoritmo global para la optimización de los hiperparámetros está dado por el siguiente, siendo T, M, μ_m, t_{max} constantes de entrada:

En el anterior algoritmo, con las constantes de la simplificación del modelo, μ_m controla el número mínimo de presas, M controla el crecimiento inicial y T el crecimiento después de cada reducción. Una menor μ_m implica una mayor

Algoritmo 1 Evolución de redes neuronales con Modelo de Depredador-Presa.

Entradas: Parámetros t_{max}, M, μ_m, Conjuntos de entrenamiento y validación
Salidas: Configuraciones de capas de redes neuronales con mejor rendimiento

Initialization $t \leftarrow 0$
while $t < t_{max}$ **do**
  if $t = M$ **then**
    $t \leftarrow t + 1$
    $\mu \leftarrow$ número de organismos
    Eliminar $\mu - \mu_m$ organismos con menor rendimiento
  else
    Duplicar la población de presas
    Calcular el rendimiento de cada presa
  end if
  $t \leftarrow t + 1$
  $M \leftarrow M + T$
end while

depredación ya que es necesario reducir más presas. Una mayor M permite un mayor crecimiento de presas en el momento inicial y una mayor T permite un mayor crecimiento de presas después de las depredaciones.

2.4. Costo computacional de la solución propuesta

Como se mencionó anteriormente, el costo computacional del modelo (basado en el número de entrenamientos realizados, que es el paso considerablemente más costoso) en ausencia de depredadores está dado por $O(t) = 2^t$, ya que 2^t es el número de entrenamientos. En este contexto, el cálculo de la complejidad está dado por el número de entrenamientos que se realizan por cada generación de organismos. La presencia de depredadores permite reducir notablemente la complejidad del problema.

Las constantes del modelo son T, M, y μ_m, por lo que permanecerán como valores fijos. Si $t \leq M$, $O(t) = 2^t \leq 2^M$, por lo que la complejidad está acotada por una constante. Si $t > M$ se eliminan organismos cada periodo T dejando μ_m. Tras dicha eliminación los organismos, los restantes μ_m se siguen duplicando durante todo el periodo, alcanzando $2^T\mu_m$. Como esto ocurre cada $floor\left(\frac{t-M}{T}\right)$ veces, entonces una cota superior de la complejidad está dada por:

$$O(t) \leq 2^M + 2^T\mu_m\Big(\frac{t-M}{T} + 1\Big). \tag{8}$$

Lo anterior esclarece la importancia de la introducción de depredadores, al acotar de forma polinomial el problema.

3. Resultados

Para verificar la efectividad del algoritmo propuesto[2], se utilizó la base de datos de MNIST, consistente de dígitos manuscritos, dividiendo el dataset en 50000 imágenes para entrenamiento, 10000 para validación y 10000 para prueba. Los ajuste de los pesos se efectuaron minimizando la función de entropía cruzada categórica dispersa utilizando Adam. Todas las neuronas fueron dotadas de activación ReLU, exceptuando la última capa, donde se utilizó softmax. Dadas las características de la base de datos, se utilizó una entrada fija de 28×28 y una salidad de 10 neuronas correspondientes a las 10 clases de dígitos.

Se hicieron pruebas variando las constantes μ_m, M, T del algoritmo, que no son propiamente hiperparámetros (lo cual llevaría a una nueva optimización) sino constantes de calibración que influyen en el número de entrenamientos del modelo y el tiempo de ejecución. Su elección dependerá de las capacidades de los sistemas empleados. Las variantes del algoritmo son las siguientes:

1. $A(k, \mu_m, M, T)$: Algoritmo con k fija.
2. $B(k_{\text{máx}}, \mu_m, M, T)$: Algoritmo con k aleatoria elegida en el intervalo $[1, k_{\text{máx}}]$.
3. $C(k_{\text{máx}}, \mu_m, M, T)$: Algoritmo con k aleatoria elegida en $[1, k_{\text{máx}}]$ y utilizando una red neuronal convolucional de soporte.

La variante de tipo A está relacionada con la búsqueda por rejilla, ya que realiza aumentos graduales pero en múltiples direcciones y de forma controlada por las depredaciones, mientras que la variante de tipo B introduce el concepto de aleatoriedad descrito anteriormente para la búsqueda por rejilla. De acuerdo con la literatura [5,4], esta modificación podría mostrar resultados beneficiosos.

Por otro lado, la variante de tipo C involucra la introducción de una CNN seguida de una FNN. Dado que el trabajo propuesto se centra en la optimización de FNNs, las capas convolucionales y de pooling se mantuvieron constantes. Para optimizar dichas capas, es posible recurrir a las técnicas referenciadas en los Antecedentes. La red propuesta cuenta con la siguiente arquitectura:

1. Una capa convolucional con 169 filtros con kernels de tamaño 4×4 y activación ReLU. La entrada es de $28 \times 28 \times 1$.
2. Una capa convolucional con 36 filtros con kernels de 3×3 y activación ReLU.
3. Una capa de MaxPooling de 2×2.
4. Una capa densa de 30 unidades y activación ReLU.
5. Una capa de salida con 10 neuronas con activación Softmax.

La capa densa conforma a la red completamente conectada cuya arquitectura optimizarse. Dado que iniciamos con la neurona con código (30) entonces utilizaremos mutaciones γ. Para los otros casos, al iniciar en (0), solamente usaremos mutaciones α y β.

Para la red convolucional, se utilizó como optimizador a la entropía cruzada categórica, en lugar de la categórica dispersa. Además, para entrenar, debido al

[2] Los códigos están provistos en `https://github.com/Pherjev/NN-Evolutivas-Lotka-Volterra`

Tabla 1. Resumen de los entrenamientos indicando la configuración del entrenamiento, la arquitectura óptima calculada, la validación obtenida y la exactitud sobre el conjunto de prueba, así como el tiempo de ejecución empleado en segundos.

Algoritmo	Arquitectura	Validación	Prueba	Tiempo
$A(1, 9, 4, 3)$	(24)	0.9506	0.9622	2233.99039006
$A(30, 9, 4, 3)$	$(150, 60)$	0.9732	0.9806	1661.98341298
$A(30, 20, 6, 3)$	$(150, 90)$	0.9738	0.9788	3051.702986
$B(70, 9, 4, 3)$	$(271, 131)$	0.9768	0.9801	3053.07773709
$B(70, 20, 6, 3)$	(362)	0.9767	0.9837	6803.21170092
$C(30, 3, 2, 1)$	$(30, 33)$	0.9837	0.9896	6742.15097094

costo computacional, se empleó una iteración de entrenamiento en los algoritmos tipo C y dos iteraciones para los tipos A y B. De esta forma, se realizaron entrenamientos rápidos para muestrear la exactitud con escasas épocas.

Para probar el rendimiento final, se utilizó un entrenamiento con el conjunto de prueba usando 50 épocas. Todos las evoluciones se efectuaron usando 15 generaciones, excepto para $k = 1$, donde se emplearon 60 debido a que no mostraba una convergencia aún hacia un máximo local y su tiempo de ejecución en general fue menor. Un resumen de los resultados aparece en la tabla 1.

Tal como esperado, el algoritmo tipo C tuvo mejor exactitud tanto en la validación como en la prueba. Igualmente anticipado pero notorio es que el uso de aleatoriedad en B mostró mejoras significativas al rendimiento, aún mejores que aumentando μ_m, M, T, cambio que se reflejó más en el tiempo de ejecución y reportó poca o ninguna mejora. Ejemplos de gráficas de evolución de la exactitud y número de individuos se proporcionan a continuación.

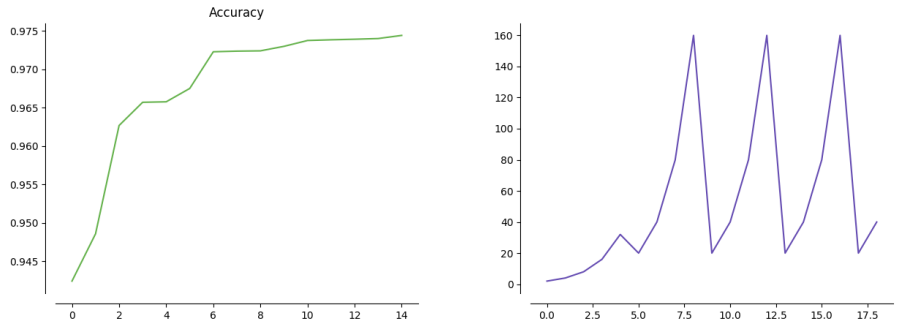


Fig. 4. Mejora de la exactitud (eje y, izquierda) por generación (eje x, izquierda), así como número de individuos en $A(30, 20, 6, 3)$. La diferencia entre los números de individuos de ambas gráficas radica en que se introduce una generación de redución para la gráfica del número de individuos.

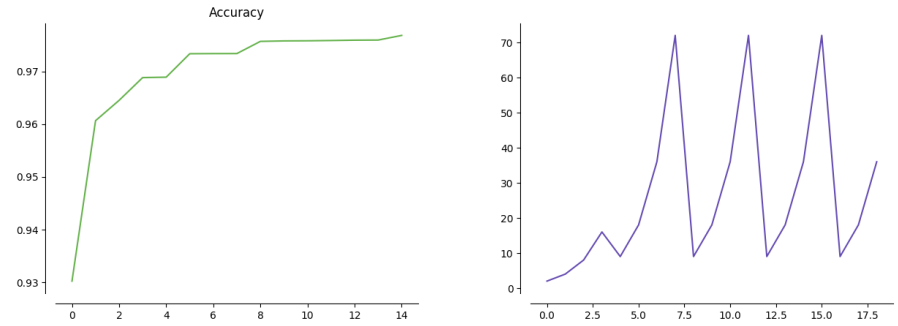


Fig. 5. Mejora de la exactitud (eje y, izquierda) por generación (eje x, izquierda), así como número de individuos en $B(70, 9, 4, 3)$.

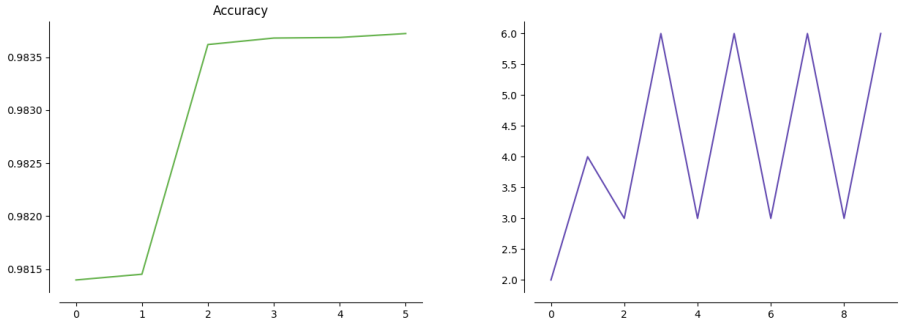


Fig. 6. Mejora de la exactitud (eje y, izquierda) por generación (eje x, izquierda), así como número de individuos en $C(30, 3, 2, 1)$.

3.1. EMNIST

Observando los resultados dados en la clasificación MNIST, se hicieron experimentos utilizando la base de datos de caracteres alfabéticos EMNIST propuesta por [9], utilizando 104000 imágenes para entrenamiento, 20800 para prueba y 20800 para validación. Se empleó la arquitectura convolucional propuesta anteriormente, efectuando el método de evolución $C(30, 3, 2, 1)$, el cual aportó los mejores resultados para MNIST. La evolución arrojó una arquitectura de (104), que amplía la propuesta original de (30) obteniendo una exactitud de validación de 0,9167. Se hizo un entrenamiento con 10 épocas, alcanzando 0,9263 de exactitud sobre el conjunto de prueba.

Un entrenamiento con el modelo originalmente propuesto arrojó 0,9209 de exactitud, que supera al 0,8515 logrado en el artículo original utilizando la técnica de *Online Pseudo-Inverse Update Method*. Una segunda red convolucional, basada en las arquitecturas VGG [20] fue implementada utilizando dos capas convolucionales de 64 filtros de 3×3, seguido de un MaxPooling de 2×2, luego otras dos capas convolucionales de 128 filtros de 3×3 y otro MaxPooling de 2×2 así como una capa densa de 256 neuronas. Esta arquitectura entrenada con 10 épocas logró una exactitud de 0.9379. Una evolución con los parámetros $C(70, 2, 2, 1)$ arrojó una estructura densa de $(361, 85)$, mejorando la exactitud a 0.9399 sobre el conjunto de prueba.

4. Conclusiones

En este proyecto se propuso un algoritmo de optimización hiperparamétrica centrada a la arquitectura de las redes FNN siguiendo una inspiración basada en el modelo de Lotka-Volterra sobre relación ecológica de depredadores y presas, dando a las redes neuronales un rol de presas frente a un depredador latente encargado de controlar su población. En términos de optimización, el rol del depredador se traduce en evitar una explosión combinatoria que supone permitir la reproducción no acotada de las presas.

En este artículo presentamos tres propuestas para realizar la optimización hiperparamétrica en la arquitectura de las capas densas: la primera consiste en realizar k aumentos en las capas para generar redes hijas, la segunda introduce aleatoriedad en la k, y la tercera, de forma separada, involucra incorporar alguno de estos algoritmos para modificar las capas densas presentes en una red neuronal convolucional dada. Para los problemas de clasificación que se abordaron (MNIST y EMNIST para caracteres), el uso de redes convolucionales es una opción más adecuada, lo cual se reflejó en los resultados.

En los casos aplicados fue posible mejorar el rendimiento de las redes, obteniendo arquitecturas optimizadas tanto partiendo de una red sin capas intermedias como desde una serie de capas densas ya definidas. En general, el uso de una k aleatoria arrojó mejores resultados que utilizando una k fija, tal como sugiere la literatura. Asimismo, para los problemas que se abordaron, fue conveniente utilizar una red inicial convolucional que variara las capas densas. Una observación relevante es que el cambio de las capas densas si bien reportó mejorías por medio de la evolución, se notó un aumento más fuerte de la exactitud mediante el empleo de capas convolucionales distintas, lo cual nos inclina a considerar incorporar un algoritmo evolutivo para las capas convolucionales o emplear las sugerencias presentadas en la literatura.

Nuestra propuesta, hasta ahora enfocada en las capas densas, el cual nos ha permitido obtener resultados beneficiosos en ambas bases de datos, tiene la convergencia hacia un mínimo local asegurada (véase Apéndice) para el caso $k = 1$, además de disponer de un costo computacional relativamente bajo que nos permite realizar varias pruebas sin llegar a una explosión combinatoria. Para términos prácticos, reuniendo tanto resultados teóricos como experimentales,

consideramos recomendable la realización de entrenamientos utilizando k aleatorias y realizar un ajuste final utilizando $k = 1$. Esta técnica evolutiva, aunque centrada en el caso de las redes FNNs, podría extrapolarse a otros contextos tanto de redes neuronales como para aspectos aún más generales[3].

Apéndice

Convergencia del algoritmo para k = 1

En esta sección probaremos la convergencia del algoritmo para $k = 1$ hacia un máximo local. Si bien el resultado no se generalizó para $k > 1$, podemos ajustar $k = 1$ en alguna iteración determinada. Para la formalización de la prueba, definamos el espacio ℓ_f de las sucesiones naturales finitas dado por:

$$\ell_f = \{(s_n)_{n\in\mathbb{N}} \mid s_n \in \mathbb{N}, \exists m : s_m = 0\}. \tag{9}$$

Los elementos de ℓ_f son las configuraciones de las redes neuronales. Definamos también una función evaluación que mide la exactitud de la red, dada por $e : \ell_f \to [0, 1]$, por ejemplo, exactitud sobre un conjunto de prueba fijo. Queremos mostrar que nuestro algoritmo converge para el caso $k = 1$ hacia un máximo local de e. En realidad esa situación puede no ocurrir, pero es posible garantizar la convergencia *casi seguramente*. En general, e puede tener no tener un máximo local, en cuyo caso, la sucesión de búsqueda se mantendrá creciente. En cada iteración i del algoritmo, tomamos un subconjunto finito $\mathcal{C}_i$ de ℓ_f y lo devuelve $\mathcal{C}_{i+1} \subset \ell_f$, también finito. Definamos la sucesión dada por:

$$m_i = \text{máx}\{e((s_n))|(s_n) \in \mathcal{C}_i\}. \tag{10}$$

Afirmamos que esta sucesión converge casi seguramente a un máximo local. Primeramente demostraremos que la sucesión (m_i) es no decreciente.

Lema 1 (m_i) *es no decreciente.*

Demostración. Sea $i \in \mathbb{N}$. Entonces como $\{e((s_n))|(s_n) \in \mathcal{C}_i\}$ es un conjunto finito, existe $(s_n) \in \mathcal{C}_i$ tal que $m_i = e((s_n))$. Si en la iteración $i + 1$ se duplican los individuos, (s_n) se conserva. En caso de que reduzcan, dado que (s_n) tiene la mejor exactitud, solamente puede ser eliminado si existe $(r_n) \in \mathcal{C}_i$ tal que $e((r_n)) = m_i$. Por lo tanto, en todos los casos $(s_n) \in \mathcal{C}_{i+1}$, o bien, $(r_n) \in \mathcal{C}_{i+1}$ de donde $m_i \leq m_{i+1}$ □

El Lema 1 se cumple para cualquer $k \in \mathbb{N}$. Para $k = 1$, debemos asegurar en nuestro algoritmo que si $(s_i) \in \ell_f$ cumple que $e((s_i)) = m_i$, entonces (s_i) no será depredado. Dada esta condición particular, podemos probar la convergencia casi segura:

[3] Agradecimientos al Dr. Carlos Francisco Brito-Loeza por el seguimiento de la investigación

Proposición 1 *Supongamos que la función e tiene un máximo global. Entonces* (m_i) *converge casi seguramente a un máximo local.*

Demostración. Por el Lema 1, la sucesión (m_i) es no decreciente. Ahora, veamos que si $m_i = m_j$ para $j > i$, entonces el algoritmo ha convergido a un máximo local. Por contradicción, supongamos que existe $(r_n) = \arg\max_{(s_n)\in\mathcal{C}_i} c((s_n))$, que no es máximo local, tal que $m_i = m_j$ para toda $j > i$. Como (r_n) no es máximo local, entonces existe (q_n) tal que $e((q_n)) > e((r_n))$ y $|(q_n) - (r_n)| = 1$, puesto que se tratan de sucesiones de naturales diferentes, por lo que difieren en solamente un punto. Notemos que $(r_n) \in \mathcal{C}_j$ para toda $j > i$. Por lo tanto, como $(q_n), (r_n) \in \ell_f$, ambas sucesiones son finitas. Si ambas se hacen 0 en el mismo índice, entonces se tiene que difieren en el índice t, de forma que:

$$(r_n) = (r_n^1, \dots, r_n^t, \dots, r_n^d, 0, \dots), \tag{11}$$

$$(q_n) = (r_n^1, \dots, r_n^t + 1, \dots, r_n^d, 0, \dots). \tag{12}$$

Sea p_α la probabilidad de ocurrencia de la mutación tipo α. La probabilidad de que no ocurra el evento r_n muta a q_n es de $p_\alpha \frac{d-1}{d} + p_\beta$. Por lo tanto, la probabilidad de que no ocurra i veces está dada por $(p_\alpha \frac{d-1}{d})^i$, de donde:

$$\lim_{i\to\infty} \left(p_\alpha \frac{d-1}{d} + p_\beta\right)^i = 0, \tag{13}$$

ya que la constante $p_\alpha \frac{d-1}{d} + p_\beta < 1$. Si ambas difieren en el índice en el que se hacen 0, entonces se tiene que:

$$(r_n) = (r_n^1, \dots, r_n^d, 0, 0, \dots), \tag{14}$$

$$(q_n) = (r_n^1, \dots, r_n^d, 1, 0, \dots), \tag{15}$$

de donde la probabilidad evento de la mutación de (r_n) a q_n es p_β, por lo que la probabilidad de que no ocurra es p_α, por lo que la probabilidad de que no ocurra i veces está dada por $(p_\alpha)^i$, de donde $\lim_{i\to\infty}(p_\alpha)^i = 0$.

Por lo tanto, (r_n) muta a (q_n) casi seguramente en alguna iteración $j > i$, de donde $(r_n) \in \mathcal{C}_j$ y por lo tanto $m_j \geq e((r_n)) > m_i$, que contradice nuestra hipótesis original. Por lo tanto, si el algoritmo se estanca, ha llegado a un mínimo local.

En el paso anterior vimos que (m_i) crece casi seguramente si se estaciona en un máximo local. Por lo tanto es de la forma:

$$m_1 = \cdots m_{1+o_1} < m_2 = \cdots . \tag{16}$$

Por lo que tenemos una subsucesión estrictamente creciente. Como existe un máximo global y el dominio es discreto, no puede crecer indefinidamente. Por lo tanto (m_i) converge a un máximo local □.

Referencias

1. Aggarwal, C.C.: Neural networks and deep learning. Springer 10, 978–3 (2018)
2. Aguilar Canto, F.J.: Eficacia de diferentes reglas hebbianas en el aprendizaje supervisado. Tecnología Educativa Revista CONAIC 7(1), 92–97 (2020)
3. Baldominos, A., Saez, Y., Isasi, P.: Model selection in committees of evolved convolutional neural networks using genetic algorithms. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 364–373. Springer (2018)
4. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of machine learning research 13(Feb), 281–305 (2012)
5. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Advances in neural information processing systems. pp. 2546–2554 (2011)
6. Bhandare, A., Kaur, D.: Designing convolutional neural network architecture using genetic algorithms. In: Proceedings on the International Conference on Artificial Intelligence (ICAI). pp. 150–156. The Steering Committee of The World Congress in Computer Science (2018)
7. Bochinski, E., Senst, T., Sikora, T.: Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3924–3928. IEEE (2017)
8. Botev, A., Lever, G., Barber, D.: Nesterov's accelerated gradient and momentum as approximations to regularised update descent. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 1899–1903. IEEE (2017)
9. Cohen, G., Afshar, S., Tapson, J., Van Schaik, A.: Emnist: Extending mnist to handwritten letters. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 2921–2926. IEEE (2017)
10. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2(4), 303–314 (1989)
11. Dayan, P., Abbott, L.F.: Theoretical neuroscience: computational and mathematical modeling of neural systems (2001)
12. Eldan, R., Shamir, O.: The power of depth for feedforward neural networks. In: Conference on learning theory. pp. 907–940 (2016)
13. Garain, K., Kumar, U., Mandal, P.S.: Global dynamics in a Beddington–DeAngelis Prey–Predator model with density dependent death rate of predator. Differential Equations and Dynamical Systems pp. 1–19 (2019)
14. Hadžiabdić, V., Mehuljić, M., Bektešević, J.: Lotka-Volterra model with two predators and their prey. Tem Journal 6(1), 132–136 (2017)
15. Hornik, K., Stinchcombe, M., White, H., et al.: Multilayer feedforward networks are universal approximators. Neural networks 2(5), 359–366 (1989)
16. Kavitha, K., Ramakrishnan, K., Singh, M.K.: Modeling and design of evolutionary neural network for heart disease detection. International Journal of Computer Science Issues (IJCSI) 7(5), 272 (2010)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Rocha, M., Cortez, P., Neves, J.: Evolutionary neural network learning. In: Portuguese Conference on Artificial Intelligence. pp. 24–28. Springer (2003)
19. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65(6), 386 (1958)

20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. van Wyk, G.J., Bosman, A.S.: Evolutionary neural architecture search for image restoration. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2019)
22. Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.H., Patton, R.M.: Optimizing deep learning hyper-parameters through an evolutionary algorithm. In: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments. pp. 1–5 (2015)

ISSN 1870-4069

Simulación y evaluación del comportamiento de peregrinos guadalupanos en un evento de sismo para planificar rutas de evacuación

José Abraham Montelongo Campos, Héctor Rafael Orozco Aguirre

Universidad Autónoma del Estado de México,
Centro Universitario UAEM Valle de México,
México

jmontelongoc414@alumno.uaemex.mx, {hrorozcoa, mquintanal}@uaemex.com

Resumen. La Basílica de Guadalupe cada año recibe a millones de peregrinos desde diversas partes del mundo. En caso de que se presente una situación de riesgo como un sismo, esta pudiera representar que peregrinos sufran un accidente e incluso fallezcan. En este artículo, se propone hacer uso de la plataforma AnyLogic y de su biblioteca de peatones para modelar y simular el comportamiento de peregrinos en un escenario 3D que represente a la Basílica de Guadalupe y sus alrededores. Se provocará una situación de sismo y se evaluará el comportamiento de los peregrinos tomando como caso de estudio el día 12 de diciembre que es cuando se celebra su fiesta anual y cuando acude una mayor concentración de personas. Esto permitirá las rutas de evacuación existentes y conocer los efectos provocados por la situación de riesgo.

Palabras clave: Agente, comportamiento, multitud, situación de riesgo.

Guadalupan Pilgrims' Behavior Simulation and Evaluation in an Earthquake Event for Planning Evacuation Routes

Abstract. Every year the Basilica of Guadalupe receives millions of pilgrims from different parts of the world. In the case of a risky situation such as an earthquake, this could represent that pilgrims suffer an accident and even die. In this paper, it is proposed to use the AnyLogic platform and its pedestrian library to model and simulate the pilgrims' behavior in a 3D scenario that represents the Basilica of Guadalupe and its surroundings. An earthquake will be provoked, and the behavior of the pilgrims will be evaluated taking as a case study the 12th of December which is when their annual festival is celebrated and when a greater concentration of people comes. This will allow evaluating the existing evacuation routes and knowing the effects caused by the risk situation.

Keywords: Agent, behavior, crowd, risk situation.

1. Introducción

La importancia de las simulaciones de evacuación para peatones en conjunto como una multitud es generalmente aceptada para diseñar edificios y para garantizar una buena seguridad en eventos masivos, donde pueda requerirse de una o más rutas de evacuación frente a una situación de riesgo o peligro como un desastre natural, por ejemplo, una inundación o terremoto, o bien, ante un acto terrorista. Se han desarrollado múltiples modelos para la simulación de peatones en las últimas décadas, entre los más populares que han sido empleados se encuentran los modelos basados en la fuerza, los autómatas celulares y las variaciones de los modelos de comportamiento y dirección de Reynold [1]. No obstante, en los últimos años, se han empleado otras alternativas que se adaptan mejor a la simulación de la dinámica de grupos en una sociedad, donde la más destacada es el empleo de los sistemas multiagente (SMA).

La simulación basada en agentes (SBA) se ha extendido a muchas áreas, incluida la sociología, biología, economía, física, química, ecología, aplicaciones industriales y simulación de multitudes en situaciones de riesgo o peligro como evacuaciones y desastres naturales. Para la mayoría de estas áreas, la SBA está reemplazando gradualmente las técnicas de microsimulación y la simulación orientada a objetos. Esto es debido a la capacidad de la SBA para capturar diferentes modelos dinámicos que generalmente consisten en interacciones entre entidades simples, conocidas como agentes a las cuales se les puede asignar un comportamiento individual o colectivo de manera estática o dinámica.

En [2], se dice que para reconocer una situación de riesgo y gestionar una evacuación exitosa de una multitud, entran en escena tres etapas clave como sigue:

1. Interpretación: se refiere al tiempo necesario para que miembros de una multitud puedan reconocer que existe un peligro para la misma.
2. Preparación: es el tiempo necesario para que cada miembro, o bien, un subgrupo de la multitud pueda decidir el curso de acción más apropiado.
3. Acción: es todo aquello que impacta en la eficiencia de la evacuación. Se refiere al tiempo que tardan una multitud en realizar una evacuación de un lugar, o bien, un traslado a otro sitio.

Acorde a [3], los criterios clave que pueden caracterizar de forma conjunta a una multitud son los siguientes:

- Tamaño: número de individuos reunidos.
- Densidad: multitud localizada en un área en particular.
- Tiempo: período de tiempo mensurable en el cual se reúne una multitud.
- Colectividad: tienen en común una identidad social, o bien, los mismos objetivos e intereses.

Los peregrinos representan un caso muy particular de multitud que comparte una misma ideología religiosa. Siendo estos en los cuales este artículo se centra. La Basílica Guadalupe (BG) ubicada en la Ciudad de México (CDMX) en México es uno de los lugares con más concentraciones de personas en el mundo, en donde más de 6 millones de personas acuden en peregrinación cada 12 de diciembre [4], fecha en que se festeja a la Virgen de Guadalupe, dejando una derrama económica de millones de pesos.

Al año se calcula que cerca de 20 millones entre peregrinos y visitantes acuden a ella [5], lo cual indica que cada día en promedio se concentran como multitud 54794.52 personas, cifra que excede en más de 5 veces la capacidad máxima de la BG en su interior que es 10 mil personas [6]. Una multitud de esta magnitud puede estar en peligro ante una situación de riesgo como lo puede ser un sismo, en la cual se pudieran perder vidas, o bien, resultar un considerable número de personas lesionadas por la estampida provocada ante una evacuación. Una estrategia de evacuación apropiada puede ayudar a resolver el problema y reducir aún más el riesgo de causalidades [7].

El antecedente más cercano a un evento de sismo como el que se simula en este artículo ocurrió el día 10 de diciembre de 2011, registrado a las 19:47:25 hora local (01:47, hora GMT), con magnitud de 6.5 grados Richter (recalculada) y con coordenadas de 17.85 N y 99.98 W a una profundidad de 58 km, el epicentro se ubicó a 55 km al NW de Chilpancingo, Guerrero. El movimiento fue fuertemente sentido en los estados de Guerrero, Oaxaca, Michoacán, Morelos, Puebla, Tlaxcala, D.F. e Hidalgo [8]. Como incidente mayor se tuvo el fallecimiento de tres personas a causa del sismo, un menor de 11 años y dos adultos [9]. Este artículo se centra en la simulación de dichos eventos para intentar prevenir el fallecimiento de personas.

El Centro de Comando, Control, Cómputo, Comunicaciones y Contacto Ciudadano de la CDMX, conocido como C5, el cual actualmente ofrece los servicios de: video monitoreo, altavoces, servicio de atención de llamadas de emergencia 911 CDMX, denuncia anónima 089 y LOCATEL 5658 1111. Todos los servicios del C5 operan las 24 horas, los 365 días del año. Los altavoces que tienen diversas cámaras de videovigilancia de la CDMX son una herramienta de difusión que permiten transmitir alertas e información de seguridad a la ciudadanía como: sismos, incidentes de alto riesgo, incluso el voceo de personas extraviadas [10] Además, cuenta con la Alerta Sísmica, la cual se activará cuando se detecte algún sismo que por su magnitud ponga en riesgo a la CDMX. También, cuenta con una aplicación para teléfonos inteligentes con sistema operativo iOS o Android, que conecta a las y los usuarios directamente al 911 de la CDMX, puede ser por llamada telefónica convencional o silenciosa y por chat [11].

En mayo de 1986, ocho meses después de los sismos de septiembre de 1985 que impactaron gravemente a la CDMX, se creó el Sistema Nacional de Protección Civil (SINAPROC), con la finalidad de establecer un sistema que permitiera a las autoridades y a la sociedad civil coordinarse de una manera eficiente y rápida en caso de un desastre [12]. El SINAPROC reúne métodos y procedimientos entre las dependencias del Gobierno de la República Mexicana, organizaciones de los diversos grupos voluntarios, sociales, privados y con las autoridades de la CDMX, de los estados y los municipios. Su propósito es efectuar acciones coordinadas para la protección de la población contra los peligros que se presenten en nuestro territorio, ya sean de origen natural, como los sismos o huracanes, o aquellos originados por la actividad humana que puedan eventualmente terminar en un desastre.

La responsabilidad principal del Centro Nacional de Prevención de Desastres (CENAPRED) consiste en apoyar al SINAPROC en los requerimientos técnicos que su operación demanda [13]. Sin embargo, no es sencillo el identificar un plan de acción óptimo basado en las limitaciones físicas del espacio urbano circundante, sobre todo de la capacidad de atención de los puntos de ingreso/egreso a la BG y los edificios a su alrededor.

Para poder lidiar con el problema antes descrito, el análisis, modelado y simulación de multitudes [14] mediante un SMA y la SBA ofrece una buena alternativa de solución ya que permite analizar y simular el comportamiento individual, o bien, colectivo de los peregrinos representados por agentes autónomos, tanto a una escala micro como a una macro. En este artículo, se emplea a la plataforma AnyLogic [15] para tales efectos, puesto que permite crear entornos virtuales de simulación a manera de escenarios 2D/3D, poblarlos con agentes, predefinir comportamientos para estos, o bien, generarlos de manera dinámica acorde a situaciones provocadas en el entorno, obtener estadísticas, entre más como analizar el comportamiento de los agentes. Con ello, es posible simular y evaluar situaciones de riesgo para juzgar sobre las ubicaciones más convenientes de puntos de evacuación (ingreso/egreso), rutas de evacuación a seguir y capacidad ideal de tales puntos. De esta manera, se creará un caso de estudio para simular y evaluar estrategias de evacuación de peregrinos ante una situación de riesgo representada por un sismo.

Con el uso de la Biblioteca de Peatones de AnyLogic (BPA) como herramienta de simulación y análisis de multitudes se permite modelar, visualizar y analizar con precisión el comportamiento de los flujos de multitudes en un entorno físico y eliminar sus posibles ineficiencias. Un peatón en un modelo AnyLogic se mueve de acuerdo con reglas físicas simuladas. Interactúa con los objetos circundantes, como paredes y escaleras, y evita posibles colisiones. Se pueden preasignar peatones con propiedades, preferencias y estados individuales. El conjunto de herramientas de la biblioteca incluye un mapa de densidad de flujo, contadores de peatones y elementos para calcular los tiempos de espera y servicio. La BPA es útil al simular dinámicas peatonales en entornos urbanos como lo es la BG. En una etapa preliminar de diseño del escenario virtual, los modelos de simulación de peatones ayudarán a evaluar la capacidad del entorno real para hacer frente a una carga planificada y cumplir con los requisitos de seguridad ante una evacuación por sismo. Las capacidades de la BPA serán fundamentales al evaluar problemas de capacidad, movilidad y accesibilidad. Durante la simulación del sismo y la concerniente evacuación, la herramienta de gestión de multitudes permitirá probar los cambios planificados y determinar la mejor ubicación de una ruta de evacuación.

2. Trabajos relacionados

Las grandes concentraciones de personas son un riesgo de seguridad, ya que no existe una mejor estrategia real para poner rápidamente a salvo a grandes cantidades de personas en caso de emergencia. Es probable que haya cuellos de botella cuando la multitud se mueva a través de las puertas, por ejemplo. La evacuación prolongada de la multitud puede llevar a las personas a estar expuestas a peligros por más tiempo. Un movimiento masivo impulsivo de personas, una estampida, también puede ser peligroso en sí mismo. No solo evitará un escape efectivo: también puede provocar lesiones aplastantes. Las personas a menudo entran en pánico en situaciones que justifican la evacuación y, como tal, el daño físico puede llegar a las personas ya que la multitud se empujará violentamente o incluso se atropellará entre sí [16].

Ante un sismo o terremoto, la falla de los componentes no estructurales y los desastres secundarios inducidos por la caída de escombros siguen siendo un problema

grave, lo que hace que la evacuación al aire libre sobre calles o carreteras sea más peligrosa. En [17], se propone un marco de simulación de evacuación de peatones que considera los escombros que caen inducidos por un terremoto, y también se establece un modelo de predicción asociado para calcular la distribución de escombros. Se adopta un modelo de fuerza social para realizar la simulación de una evacuación. Se realiza un caso de estudio del área de enseñanza ubicada en el campus de la Universidad de Tsinghua en China. El método propuesto puede ayudar a identificar áreas de alto riesgo entre las carreteras de evacuación, así como a proporcionar una base científica para la planificación urbana futura y el desarrollo de simulacros de evacuación de emergencia.

Los científicos sociales han criticado los modelos computacionales de flujos de peatones por su tratamiento de las multitudes psicológicas como meras agregaciones de individuos. De hecho, la mayoría de los modelos de dinámica de evacuación utilizan analogías de la física donde los peatones se consideran partículas. Si bien esto garantiza que los resultados de la simulación coincidan con fenómenos físicos importantes, como la desaceleración de la multitud con una densidad creciente, se ignoran los fenómenos sociales, como los procesos grupales. De acuerdo con lo dicho en [18], el contar con un buen modelo de locomoción realista es de vital importancia en escenarios de simulación de evacuaciones de peatones. Sin embargo, se suelen dejar de lado el contemplar aspectos del comportamiento social. Dicho de otra manera, estos aspectos y sus efectos han sido ampliamente examinados en la investigación empírica por psicólogos sociales, pero a menudo están ausentes en los modelos computacionales.

En una multitud, las personas tienen identidades sociales que comparten entre sí. El proceso de auto categorización determina las normas dentro de la multitud e influye en cómo las personas se comportarán en situaciones de evacuación. En la propuesta que fue dada en [1], se formula la aplicación de la identidad social en la simulación de peatones algorítmicamente. El objetivo es examinar si es posible trasladar el modelo psicológico a modelos computacionales de movimiento de peatones para que los resultados de la simulación correspondan a las observaciones de la psicología de masas. Esta propuesta fue restringida a un escenario específico que fue investigado a fondo por psicólogos de masas: el bombardeo y la posterior evacuación de un vagón subterráneo de Londres el 7 de julio de 2005.

Un ejemplo de un modelo de movimiento de peatones que incorpora el comportamiento social y psicológico es dado en [19]. Sin embargo, su enfoque está en la implementación del modelo y no en intentar emular los hallazgos empíricos de la psicología. La distancia de los modelos a las teorías subyacentes, la falta de información de modelado detallada y la cantidad de parámetros para calibrar, hacen que sea casi imposible replicar los modelos y compararlos con las observaciones. La representación de las trayectorias de peatones en [20] se enumeran y se definen brevemente los parámetros elegidos para el vector de entrada que se utilizará en cada uno de los algoritmos empleados para determinar la trayectoria del peatón. Estos parámetros se pueden dividir en dos grupos: parámetros descriptivos y parámetros estadísticos.

La implementación de las características/cualidades de un pasajero partimos de la definición que Anylogic da al Pedestrian, o viandante, denominado Ped (Pedestrian), están compuestas por una capacidad de aceleración y deceleración de movimiento, una velocidad de movimiento, y un ámbito de influencia [21]. En [7] proponen el uso de AnyLogic que incluye una biblioteca peatonal que se ocupa principalmente de diferentes aspectos de modelado de simulación de multitudes.

En [22] un modelo generalmente se prueba en varios escenarios de rendimiento. Cuando se considera un número tan grande de escenarios bajo evaluación, los datos pueden acumularse muy rápidamente y deben ganar cierta estructura dentro del sistema. Esto significa que se necesitan ciertos mecanismos para que este tipo de sistema maneje las grandes cantidades de datos generados e información estructurada. Como tal, es importante describir la diferencia entre los dos tipos de modelos en uso:

- Modelos estáticos: versiones de los modelos de edificios bajo evaluación de diseño.
- Modelos dinámicos: extensiones de los modelos estáticos, que aportan datos, información y conocimientos relacionados con el análisis adicional.

En la simulación propuesta en AnyLogic por [7], un modelador puede diseñar entornos espaciales para cualquier gran multitud. La geometría espacial se puede modelar con precisión hasta la escala de un entorno del mundo real. Además, el modelador puede utilizar el enfoque del MBA para describir la naturaleza de los individuos y sus interacciones en una multitud, definiendo atributos, por ejemplo, edad, peso, género, etc. AnyLogic proporciona varias distribuciones de probabilidad para inicializar una población de agentes y generar aleatoriamente valores iniciales de estos atributos basados en una distribución de probabilidad seleccionada. El comportamiento de los agentes se controla mediante diagramas de flujo de estado con transiciones activadas por tiempo o activadas por eventos [15].

3. Metodología

De acuerdo con [22], representar el conocimiento sobre el proceso de creación y análisis de escenarios de simulación de multitudes es un paso requerido para facilitar la minería de conocimiento. Esto generalmente requiere varias iteraciones de modelado y análisis y se basa en los modelos de información. Existen dos procesos principales involucrados:

1. Generación de escenarios: el proceso de comprender, las áreas y el entorno de los edificios para crear escenarios de simulación válidos a partir de esto, donde se hacen varias suposiciones de acuerdo con los requisitos de análisis.
2. Análisis de retroalimentación: el proceso de analizar los resultados del escenario y proporcionar retroalimentación para la toma de decisiones de diseño.

Actualmente, los SMA constituyen un área de creciente interés dentro de la inteligencia artificial, debido a la capacidad de adaptación para resolver problemas complejos no resueltos de manera satisfactoria mediante otras técnicas. Mediante el buen uso y aprovechamiento de las ciencias computacionales, en específico, en las áreas de la inteligencia artificial y la realidad virtual, se puede contribuir mediante estrategias de simulación 2D/3D basadas en un SMA para entender y resolver problemas del mundo real que engloben el comportamiento humano. Con la SBA, las entidades activas, conocidas como agentes, deben identificarse para definir su comportamiento. Estas entidades pueden ser personas, hogares, vehículos, equipos, productos o empresas, lo que sea relevante para el sistema. Se establecen conexiones entre ellas, se establecen las variables del entorno y se ejecutan simulaciones.

La dinámica global del sistema emerge entonces de las interacciones de los muchos comportamientos individuales y colectivos posibles de los agentes.

En este artículo, se propone la creación de un escenario virtual 2D/3D para la simulación de la BG y su entorno colindante, así como de los peregrinos que ahí arriban como tipo de multitud. Esto se hace mediante el uso primordial de la BPA (ver figura 1) para simular el comportamiento de los peregrinos. AnyLogic como herramienta de simulación soporta metodologías de simulación comunes: dinámica de sistemas (DS), sistemas de eventos discretos (SED), y el modelado basado en agentes (MBA) [23]. AnyLogic proporciona como ventajas para su elección una plataforma como herramienta para la creación rápida de prototipos a través de una interfaz amigable y fácil de usar, al igual que un entorno de desarrollo basado en el lenguaje Java y un conjunto de bibliotecas de componentes 2D/3D multipropósito (objetos y agentes), que en conjunto ayudan a acelerar el proceso de modelado proporcionando un entorno de ejecución para la simulación de un entorno y el análisis del comportamiento de quienes lo habitan a escala real.

En esencia con la BPA se dispondrá de lo siguiente:

1. Los peatones representados por peregrinos se mueven según un modelo de fuerza social. Eligen la ruta más corta, evitan colisiones con otros objetos mediante el análisis del entorno actual y toman decisiones sobre nuevos movimientos.
2. El comportamiento de los peregrinos se define mediante un diagrama de flujo de proceso, que es fácil de crear y comprender. Además de los bloques básicos para simular movimientos en un entorno físico, la biblioteca contiene elementos de marcado de espacio para colocar muros, servicios, atractores y escaleras, por nombrar algunos.
3. Los peregrinos pueden ser asignados previamente con propiedades individuales, preferencias y estados, como su sexo, edad o el lugar de interés al que se dirigen a visitar. Esto es posible debido al enfoque basado en agentes, que se implementa en modelos peatonales.
4. Las estadísticas de salida del modelo permiten evaluar la capacidad y el rendimiento de las instalaciones de la BG y su entorno colindante, contar a los peatones en diferentes zonas y medir los tiempos de espera y servicio.
5. Densidad de flujo peatonal de los peregrinos mostrada en un mapa de densidad a color. Este mapa se repinta constantemente de acuerdo con los valores reales: cuando la densidad cambia en algún punto, el color cambia dinámicamente para reflejar este cambio. El color rojo indica la densidad crítica y el color azul se usa para bajas densidades. Cuando la densidad en alguna área es cero, el mapa de densidad para esta área en particular no se dibuja en absoluto.
6. La BPA se puede integrar perfectamente con la Biblioteca de Tráfico Vial (BTV) para crear un modelo que considere el flujo e infraestructura vial de los alrededores de la BG dando un mayor realismo y riqueza visual al modelo virtual para asemejarlo más al mundo real.
7. AnyLogic admite la importación de objetos 3D almacenados en archivos X3D y VRML. Los modelos peatonales se pueden animar con activos 3D de personas y objetos circundantes.

Tabla 1. Bloques usados en diagramas de flujo de peatones.

Nombre del Bloque	Descripción
PedSource	Genera peatones. Se usa por lo general como un punto de inicio para el flujo de peatones. Este bloque también puede ser usado para generar grupos de peatones.
PedSink	Retira peatones entrantes. Es usado como punto final del flujo de peatones.
PedGoTo	Provoca que los peatones vayan a una ubicación especificada. La ubicación puede ser definida con figuras de marcación especial (línea objetivo o área), o como un punto con coordenadas dada.
PedService	Simula como los peatones reciben servicios en puntos designados. Los puntos de servicio son dibujados gráficamente usando figuras especiales de marcación (servicio con líneas, servicio con área).
PedWait	Hace que los peatones se dirijan a la ubicación especificada y esperen allí por un periodo especificado de tiempo. La ubicación puede ser definida por una meta, área o punto con coordenadas dadas.
PedSelectOutput	Dirige a los peatones entrantes a uno de los cinco puertos de salida según las relaciones o condiciones especificadas.
PedEnter	Acepta peatones generados en otros lugares (por ejemplo, en otro diagrama de flujo) y los introduce dentro del entorno simulado en una ubicación específica.
PedExit	Acepta peatones en el puerto de entrada. Retira peatones de un entorno simulado y los saca del mismo a través de un puerto de salida.
PedEscalator	Simula como los peatones son transportados por escaleras eléctricas.
PedChangeLevel	Mueve el flujo de peatones desde su piso o nivel actual a uno nuevo.
PedAreaDescriptor	Permite definir reglas y/o restricciones a los peatones estando dentro de un área rectangular o poligonal. Permite modificar la velocidad de los peatones y definir una velocidad máxima.
PedGroupAssemble	Organiza grupos a partir de la secuencia de peatones que ingresa al bloque. Permite definir criterios múltiples para la creación de grupos.
PedGroupChangeFormation	Hace que los grupos de peatones cambien su formación. Tres tipos de formaciones de grupo son admitidas: multitud, en cadena y frontal.
PedGroupDisassemble	Dispersa grupos de peatones. Cuando el líder de grupo entra a este bloque, el grupo se disuelve, de manera que los peatones disueltos sigan sus propias rutas.
PedSettings	Bloque opcional separado que permite especificar parámetros generales para todos los bloques de la BPA en el modelo.

8. Las capacidades de AnyLogic permiten cargar datos de archivos de dibujo CAD, archivos de forma y archivos de base de datos directamente al modelo peatonal de la BG.

La BPA permite general un diagrama de flujo del movimiento de los peatones a través de los bloques mostrados en la Tabla 1 y los elementos de marcado espacial dados en la Tabla 2. Estos bloques y elementos se muestran en la Fig. 1.

Tabla 2. Elementos de marcado espacial usados en diagramas de flujo de peatones.

Elemento	Descripción
Wall	Para dibujar paredes de formas complejas.
Rectangular Wall	Para dibujar áreas rectangulares no accesibles para los peatones.
Circular Wall	Para dibujar obstáculos circulares dentro del área simulada.
Target Line	Para definir el lugar donde aparecen los peatones en el espacio simulado.
Service with Lines	Para definir los servicios con líneas de cola donde los peatones esperan hasta que el servicio esté disponible.
Service with Area	Para definir los servicios con áreas de cola donde los peatones esperan hasta que el servicio esté disponible.
Rectangular Area	Para definir gráficamente el lugar donde aparecen los transeúntes en el espacio simulado; destino del movimiento peatonal, y peatones esperando lugar.
Polygonal Area	Para definir un área de forma compleja. Se comporta como el área rectangular.
Attractor	Para controlar la ubicación de los peatones dentro de un área, estos definen los puntos exactos donde los peatones esperarán dentro del área.
Escalator Group	Para simular cómo se transportan los peatones por escaleras mecánicas.
Pathway	Para restringir el movimiento de peatones al moverse a lo largo de un camino o ruta, simplemente define la pauta de movimiento.
Density Map	Para mostrar la densidad peatonal a colores. El color de cada punto del espacio corresponde a la densidad de corriente en esta área en particular.
Ped Flow Statistics	Para recopilar estadísticas sobre los peatones que cruzan la sección transversal dibujada.

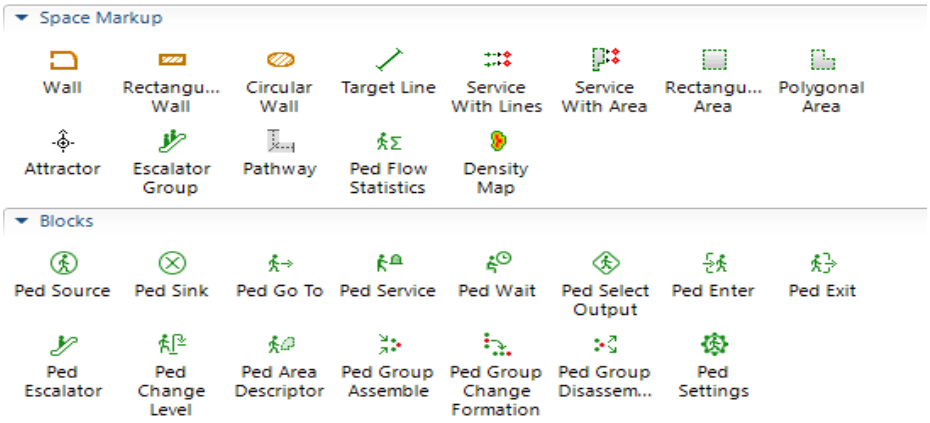


Fig. 1. Vista de los bloques y elementos de marcado espacial de la BPA.

4. Creación del escenario de simulación

Para la creación del escenario de simulación de la BG ante una evacuación por una situación de sismo en AnyLogic se hizo lo siguiente:

1. Definir los puntos de acceso/egreso o entrada/salida de los peregrinos: a través de estos los peregrinos se desplazan en puntos de interés (ver Fig. 2 definidos por números negros):

— Punto principal: localizado sobre la Calzada de Guadalupe que permite que los peregrinos ingresen/egresen a la Plaza Mariana.
— Punto secundario derecho inferior: sobre la Av. Fray Juan de Zumárraga por donde los peregrinos pueden ingresar/egresar de la Plaza Mariana y la Plancha de Imagen de la Virgen de Guadalupe.
— Punto lateral izquierdo inferior: ubicado en Calzada de los Misterios que conduce al ingreso/egreso de peregrinos a la parte inferior de la BG y la Plaza Mariana.
— 2 puntos laterales izquierdos centrales: por ellos los peregrinos ingresan/egresan a la parte trasera de la BG.
— Punto del Bautistero: como lo indica su nombre a través de él los peregrinos ingresan/egresan al Bautistero.
— Punto lateral derecho superior: para el ingreso/egreso de peregrinos a la Capilla del Pocito y Antigua Parroquia de Indios.

2. Definir puntos de reunión: ante el evento de sismo es en estos lugares donde los peregrinos van a reunirse previo a una evacuación (ver Fig. 2 definidos por números rojos):

— Punto de reunión de la Plaza Mariana: es el mayor de todos y a él se deben conducir los peregrinos que se encuentren en dicha plaza o dentro de la BG, el Templo Expiatorio y la Parroquia de la Santa María.
— Punto del Camino del Pocito: los peregrinos que se localicen cerca de éste o de la Capilla del Pocito o la Antigua Parroquia de Indios a él acudirán.
— Punto del Bautistero: todo peregrino ubicado cerca de él y del Museo de la BG ese será su punto de reunión.

3. Definir sitios de interés: estos representan los lugares a los que los peregrinos tienen particular intención de visitar y conocer (ver Fig. 3):

— BG: es el sitio de atracción principal al cual los peregrinos se encolarán para su ingreso. El 60% de los peregrinos que ingresan por el punto principal, punto secundario derecho inferior y punto lateral izquierdo inferior decidirán ir a este sitio. El 100% de los peregrinos que ingresan por los puntos laterales izquierdos centrales se dirigirán a la BG.
— Templo Expiatorio: conocido como la Antigua BG, es el segundo lugar de interés para los peregrinos. El 20% de los peregrinos que ingresan por el punto principal, punto secundario derecho inferior y punto lateral izquierdo inferior decidirán ir a este sitio.
— Antigua Parroquia de Santa María: es el tercer lugar de interés para los peregrinos. El 20% de los peregrinos que ingresan por el punto principal, punto secundario derecho inferior y punto lateral izquierdo inferior decidirán ir a este sitio.

Fig. 2. Escenario creado en AnyLogic para simular la afluencia de peregrinos y su comportamiento y evacuación ante un evento de sismo en la BG.

- Bautistero: El 50% de los peregrinos que ingresan por el punto del Bautistero de dirigirán a este sitio.
- Museo de la BG: El 50% de los peregrinos que ingresan por el punto del Bautistero de dirigirán a este sitio.
- Capilla de los Juramentos: El 33.33% de los peregrinos que ingresan por el punto lateral derecho superior elegirán a este sitio como de interés.
- Antigua Parroquia de Indios: El 33.33% de los peregrinos que ingresan por el punto lateral derecho superior elegirán a este sitio como de interés.
- Capilla del Pocito: El 33.33% de los peregrinos que ingresan por el punto lateral derecho superior elegirán a este sitio como de interés.
- Capilla del Cerrito: por este sitio sólo transitan peregrinos que hayan ingresado por los puntos de ingreso cercanos a ella.

4. Definir el diagrama de flujo del comportamiento peatonal de los peregrinos: este diagrama se generó con los bloques y elementos de marcado espacial de la BPA para definir el flujo de movimiento de los peregrinos en el escenario y para controlar la lógica de desplazamiento peatonal y de comportamiento ante el evento de sismo y su evacuación. En la Fig. 3 se muestra dicho diagrama.
5. Definir un flujo vehicular simple en las vialidades aledañas al escenario: se empleó la BTV para darle un mayor realismo al escenario de la BG y sus alrededores. A través de los componentes de dicha biblioteca se definió el comportamiento del flujo vial. Este flujo es complementario más no esencial para lo respectivo a la parte peatonal con los peregrinos.

En la Fig. 3, se muestra la lógica del comportamiento peatonal de los peregrinos mediante un grafo de conexión entre estados de navegación de los peregrinos por el escenario creado.

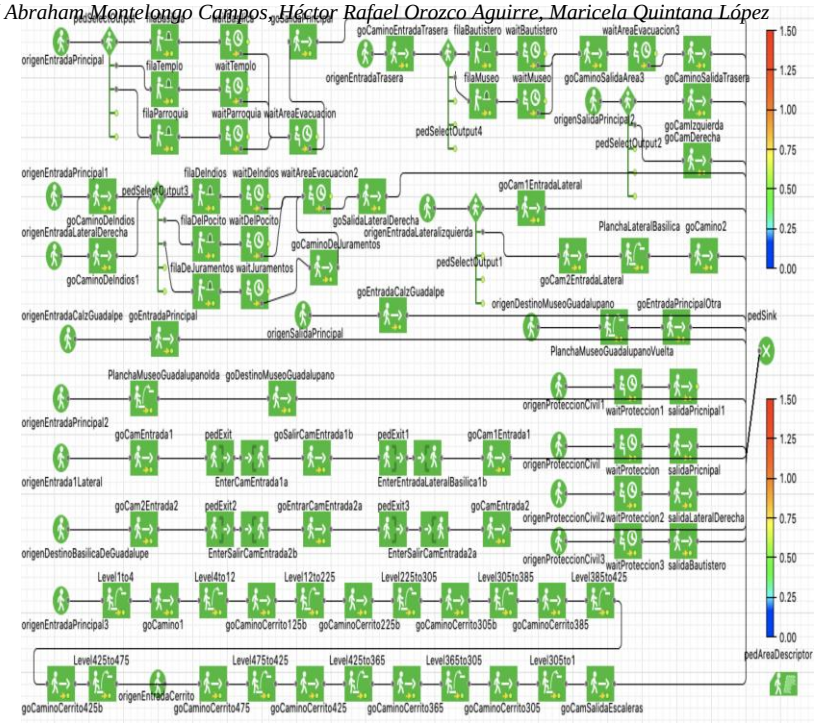


Fig. 3. Lógica del comportamiento peatonal de peregrinos en AnyLogic en el escenario creado.

Este grafo permite que haya afluencia peatonal de peregrinos a lugares de interés para que ingresen/egresen a ellos. Cuando se lance el evento de sismo los peregrinos deberán salir de dichos lugares para concentrarse en los puntos de reunión cercanos y esperar por la indicación de evacuación tras 5 minutos de espera para que abandonen el escenario egresando por los puntos cercanos.

4.1. Resultados

La Fig. 4 permite apreciar la simulación 3D del escenario de la BG y sus alrededores. La afluencia de peregrinos es de 39 peregrinos por minuto para que cada hora ingresen 2340 peregrinos y en 24 horas se tenga el aproximado al promedio diario de afluencia que es de 56160 peregrinos visitantes.

Por su parte, la Fig. 5 es para ver el comportamiento de los peregrinos ante la evacuación por un sismo. Para efectos de poder visualizar un ejemplo de evacuación ante un evento de sismo se muestra el caso para los peregrinos que arribaron en 1 hora.

Cabe mencionar que se empleó la versión estudiantil de AnyLogic para crear el escenario presentado en este artículo y se tienen limitantes en el número de agentes peregrinos que pueden generarse en el modelo.

Para poder tener un escenario de caso de estudio con un número de peregrinos cercano al que arriba cada 12 de diciembre a la BG, se debe emplear la versión de investigador bajo licencia.

Fig. 4. Simulación 3D en AnyLogic de una afluencia de peregrinos a la BG y sus alrededores.

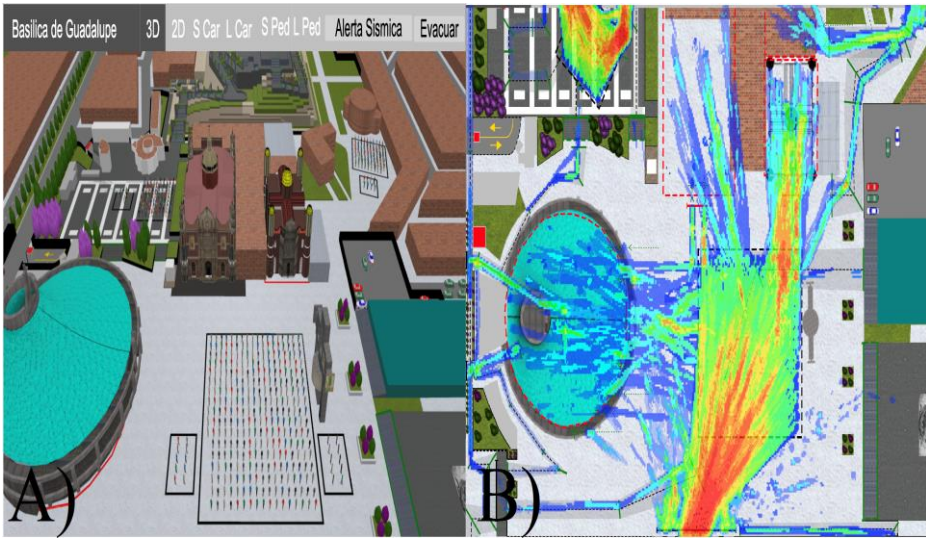


Fig. 5. Evacuación por sismo de peregrinos en la BG y sus alrededores en AnyLogic. A) Concentración en puntos de reunión cercanos. B) Mapa de densidad del flujo de evacuación de peregrinos.

En la simulación y tras emitirse la alerta sísmica, el tiempo total de arribo de todos los peregrinos a los puntos de reunión fue de 5 minutos y 45 segundos, mientras que el tiempo total de evacuación por sismo de los peregrinos desde los puntos de reunión fue de 8 minutos y 25 segundos. Estos tiempos deben mejorarse explorando alternativas de estrategias de reducción evaluando rutas de arribo a puntos de reunión y de evacuación más eficaces.

5. Conclusiones y trabajo a futuro

Hasta el momento, sólo se ha simulado el comportamiento de afluencia peatonal de peregrinos, así como la evacuación por una situación de riesgo dada por un sismo que ocurre tras una hora de afluencia. Sin embargo, es necesario analizar la capacidad de desalojo de los sitios de interés para saber en cuánto tiempo pueden los peregrinos salir, si lo harán de manera segura, si se suscitarán accidentes, o bien, hasta incluso muertes por el caos provocado. Además, cuando los peregrinos egresen por los puntos correspondientes, se debe hacer algo similar. Este análisis permitirá rediseñar el escenario para proponer un plan de acción óptimo que reduzca en lo mínimo posible las incidencias que se puedan suscitar y ver la posibilidad de reubicación de puntos de ingreso/egreso para un mejor control de la multitud de peregrinos. Para ello, un análisis estadístico de los datos recabados por la simulación y replanteando distintas simulaciones sobre el escenario serán la pauta que dirá qué es lo más adecuado para que en el mundo real ante un sismo como tal se tenga la mejor estrategia a seguir de manera adecuada.

En un trabajo futuro se extenderá la simulación a un escenario que incluya el número máximo de peregrinos para el 12 de diciembre y poder obtener resultados que sean concluyentes. Para enriquecer la simulación y tener una mejor comprensión de una situación de riesgo y los efectos de esta buscando que se lleguen a dar posibles cursos de acción para una mejor evacuación con la menor cantidad de incidencias posibles, es decir, su minimización, lo que se puede hacer para extender el escenario presentando en este artículo es principalmente lo siguiente:

- Incorporar diferentes comportamientos individuales y grupales en los peregrinos para representar una multitud más parecida a la real.
- Emplear la Biblioteca de Simulación de Procesos de AnyLogic (BSP) para tener un comportamiento dinámico y más apegado a la realidad ante una situación de riesgo para los agentes. Esto permitirá un detalle más fino ante una situación de riesgo como lo es un sismo.
- Emplear cámaras a nivel de piso y a manera de drones para tener otras vistas del comportamiento peatonal de los peregrinos y de su reacción ante la situación de riesgo y sus efectos.
- Recabar estadísticas de las tasas de accidentalidad o muerte posible de los peregrinos.
- Trabajar con esquemas inteligentes de control de evacuaciones ante una situación de riesgo.
- Evaluar las mejores ubicaciones de los puntos de ingreso/egreso.
- Mejorar los tiempos de arribo a los puntos de reunión y de evacuación para poder reducirlos mediante estrategias de simulación eficaces.

Referencias

1. von Sivers, I., Templeton, A., Künzner, F., Köster, G., Drury, J., Philippides, A., Bungartz, H. J.: Modelling social identification and helping in evacuation simulation. Safety Science, 89, pp. 288–300 (2016)

2. Ochoa Zezzatti, C.A., Rudomin, I., Vargas Solar, G., Espinosa-Oviedo, J.A., Pérez, H., Zechinelli Martini, J.L.: Humanitarian logistics and cultural diversity within crowd simulation. Computación y Sistemas, 21(1), pp. 7–21 (2017)
3. Zhou, B., Wang, X., Tang, X.: Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2871–2878 (2012)
4. Natgeo: National geographic en español. https://www.ngenespanol.com/traveler/el-12-de-diciembre-la-cdmx-recibe-peregrinos-de-todo-el-mundo/ (2017)
5. Sectur: https://www.gob.mx/sectur/prensa/mexico-entre-los-paises-mas-visitados-por-turismo-religioso-sectur (2016)
6. Artigas, J.B.: La Basílica del siglo XX en la Villa de Guadalupe (2010)
7. Mahmood, I.H.: Analyzing emergency evacuation strategies for mass gatherings using crowd simulation and analysis framework: Hajj scenario. In: Proceedings of the ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, pp. 231–240 (2017)
8. Servicio Sismologico Nacional: Reporte de sismo CDMX: Instituto de Geofísica. http://www.ssn.unam.mx/sismicidad/reportes-especiales/2011/SSNMX_rep_esp_20111210_guerrero_M65.pdf (2011)
9. UNIVERSITAM: https://universitam.com/academicos/sismologia/informe-sismologico-mundial-9-diciembre-2011-iran-meridional-2/ (2011)
10. Gobierno CDMX: Centro de Comando, Control, Cómputo, Comunicaciones y Contacto de la Ciudad de México. https://www.c5.cdmx.gob.mx/dependencia/acerca-de/el-c5-de-la-cdmx (2015)
11. Gobierno CDMX: Centro de Comando, Control, Cómputo, comunicaciones y Contacto de la Ciudad de México. https://www.c5.cdmx.gob.mx/canales-de-atencion-emergencias/app-9-1-1-de-la-cdmx (2017)
12. CENAPRED: https://datos.gob.mx/busca/organization/about/cenapred (2020)
13. SINAPROC: https://www.gob.mx/cenapred/articulos/que-es-el-sinaproc-y-como-se-consolido-en-nuetro-pais-enterate (2020)
14. De Gyves, O.T.: Comportamientos en simulación de multitudes: revisión del estado del arte. Research in Computer Science, 62, pp. 319–334 (2013)
15. Grigoryev, I.: AnyLogic in three days. AnyLogic (2018)
16. Isak-Peterson, E.J.: Simulation and evaluation of strategies for emergency evacuation of high-density crowds. Stockholm, pp. 4–5 (2018)
17. Lu, X., Yang, Z., Cimellaro, G.P., Xu, Z.: Pedestrian evacuation simulation under the scenario with earthquake-induced falling debris. Safety Science, 114, pp. 61–71 (2019)
18. Templeton, A., Drury, J., Philippides, A.: From mindless masses to small groups: conceptualizing collective behavior in crowd modeling. Review of General Psychology, 19(3), pp. 215–229 (2015)
19. Chu, M.L., Kincho, L.: Computational framework incorporating human behaviors for egress simulations. Journal of Computing in Civil Engineering, 27(6), pp. 699–707 (2013)
20. Grande, B.D.: Detección de comportamientos anómalos en la marcha de peatones mediante tecnicas de aprendizaje automatíco. pp. 29–34 (2017)

21. Benitez, Y.F.: Sistema de evaluación para toma de decisión en la modificación de una estación de tren via simulación informática. Universitat Autonoma de Barcelona, pp. 13–14 (2013)
22. Calin Boje, H.L.: Crowd simulation based knowledge mining supporting building evacuation design. Cardiff University, pp. 13–17 (2017)
23. Beatriz-Andres, R.S.: Modelado y Simulación de la Cadena de Suministro con AnyLogic. Universitat Politecnica de Valencia, pp. 60–61 (2016)

ISSN 1870-4069

Recocido simulado y el algoritmo de selección clonal aplicados al problema del agente viajero

Nayeli Joaquinita Meléndez Acosta

Universidad del Istmo,
México

nayelimelendez@ gmail.com

Resumen. Este artículo presenta una evaluación de los algoritmos de Recocido Simulado (RS) y Selección Clonal (CLONALG) aplicados al Problema del Agente Viajero, donde el objetivo es minimizar la distancia del recorrido. Los dos algoritmos han sido estudiados por separado para solucionar este problema y ambos han mostrado ser eficientes respecto a otros algoritmos, sin embargo, no existe comparación entre estos. El rendimiento de ambos algoritmos se determinó utilizando dos conjuntos de datos, el primero de estos entre ciudades generadas aleatoriamente. En la segunda prueba se consideraron instancias de ciudades tomadas de la librería TSPLIB. Los experimentos muestran que los dos algoritmos producen buenos resultados para encontrar la mejor ruta. Recocido Simulado obtuvo mejores soluciones en instancias con mayor cantidad de ciudades y en la mayoría de los casos se obtiene en menos iteraciones. En un futuro se pretende implementar variantes de los algoritmos originales y aplicarlos a problemas reales en municipios de la región.

Palabras clave: Algoritmos metaheurísticos, distancia, recorrido, sistema inmune, enfriamiento, temperatura.

Simulated Annealing and Clonal Selection Algorithm applied to the Traveling Salesman Problem

Abstract. This article presents an evaluation of the Simulated Annealing (SA) and Clonal Selection (CLONALG) algorithms applied to the Traveling Salesman Problem, where the objective is to minimize the distance of the route. The two algorithms have been studied apart to solve this problem and both have been shown to be efficient while compared to other algorithms, however, there is no comparison between them. The performance of both algorithms was determined using two data sets, the first of these between randomly generated cities. In the second test, instances of cities taken from the TSPLIB library were considered. Experiments showed that the two algorithms produce good results for finding the best route. Simulated Annealing obtained better solutions in instances with more cities and in most cases, it is obtained in fewer iterations. In the future, it is

intended to implement variants of the original algorithms and apply them to real problems in the municipal localities.

Keywords: Metaheuristic algorithms, distance, travel, immune system, cooling, temperature.

1. Introducción

Los algoritmos metaheurísticos han mostrado ser buenos para resolver el problema de agente viajero, ya que es un problema NP-difícil, este problema es muy simple con un pequeño número de ciudades, pero la complejidad computacional aumenta exponencialmente con el número de ciudades y no hay un algoritmo conocido que lo resuelva en tiempo polinomial [1, 2].

Los algoritmos metaheurísticos utilizan estrategias de búsqueda para explorar el espacio de búsqueda de manera eficiente, aun en espacios muy grandes. Este tipo de algoritmos, no garantizan que se encuentre una solución óptima, pero pueden proporcionar resultados muy buenos, casi óptimos [3, 4]. La necesidad de encontrar una solución rápida al PAV ha llevado a la creación de muchos algoritmos con resultados aproximados. Algunos algoritmos metaheurísticos utilizados para resolver este problema son: Algoritmos Genéticos (AG), el Recocido Simulado (RS), Optimización por Cúmulo de Partículas *(Particle Swarm Optimization, PSO*), Optimización por Colonia de Hormigas (*Ant Colony Optimization, ACO*), Algoritmo Colonia de Abejas Artificiales (*Artificial Bee Colony algorithm, ABC*), la Búsqueda Tabú y la Búsqueda Cuckoo, por mencionar algunos [2, 5 y 6].

En el PAV existen varias ciudades que debe recorrer un agente, el agente debe visitar cada ciudad solo una vez y regresar a la ciudad donde comenzó. El objetivo del PAV es encontrar en el menor número de iteraciones, el orden en que debe visitar a las ciudades, de modo que la distancia recorrida sea la menor en comparación con cualquier otro orden, reduciendo así el costo total del viaje. Además, este problema tiene muchas aplicaciones en el mundo real, tales como: cableado de computadoras, enrutamiento, comunicación inalámbrica, horarios de transporte, procesos de fabricación y programación de vehículos planificación y logística [2, 3].

Este articulo implementa dos enfoques heurísticos: el algoritmo de recocido simulado y el algoritmo de selección clonal (CLONALG) para resolver el problema del agente viajero, el rendimiento de ambos algoritmos se determinó utilizando dos conjuntos de datos, estos algoritmos ya han sido estudiados por separado para resolver PAV y ambos han mostrado ser eficientes respecto a otros algoritmos, sin embargo, no existe comparación entre ellos. El documento está organizado de la siguiente manera: la sección dos discute los trabajos relacionados. En la tercera sección se presenta una introducción al problema del agente viajero proporcionando una breve descripción sobre cómo funciona el PAV. La sección cuatro da detalles sobre los algoritmos implementados: RS y CLONALG. La sección cinco describe sobre la implementación de los algoritmos, esta sección se enfoca en pseudocódigo y describe el flujo de cada algoritmo.

La sección seis aborda las pruebas realizadas del PAV usando RS y CLONALG, junto con el análisis comparativo entre los dos algoritmos. Finalmente, en la última sección se muestran las conclusiones y trabajo futuro.

2. Trabajos relacionados

La necesidad de encontrar una solución rápida a los problemas de optimización, ha llevado a la creación de muchos algoritmos con resultados aproximados. Algunos de los algoritmos heurísticos más populares para resolver el problema del agente viajero son los algoritmos genéticos, el recocido simulado, la optimización de colonias de hormigas, la optimización por cúmulo de partículas, el algoritmo de búsqueda tabú [2, 5 y 6].

El diseño y el análisis de algoritmos efectivos para resolver el PAV sigue siendo muy importante debido a que se quiere mejorar su eficiencia para resolver otros problemas de optimización en general [7]. La combinación o creación de nuevos algoritmos se realiza con el objetivo de tratar mejor con el PAV, algunos trabajos relacionados donde han utilizado recocido simulado y el algoritmo de selección clonal son:

Sumathi en [2] realizó un análisis comparativo en función del número de iteraciones necesarias para alcanzar el camino óptimo, utilizaron dos algoritmos PSO y RS. Zhou en [6] propuso un nuevo algoritmo basado en recocido simulado y el algoritmo programación de expresión de genes (PEG) para resolver el PAV, utiliza recocido simulado para aumentar la diversidad de la PEG y mejorar la capacidad de búsqueda global. En los experimentos utilizaron seis instancias de referencia y los resultados mostraron que el algoritmo propuesto supera a otros algoritmos heurísticos conocidos, en términos de la mejor solución, la peor solución y el tiempo de ejecución del algoritmo. Pang en [7] diseñó un nuevo algoritmo utilizando CLONALG y la técnica de búsqueda local, este último fue empleado para acelerar la madurez del sistema, usaron varios problemas de referencia en TSPLIB para evaluar el rendimiento del algoritmo propuesto. Sus resultados mostraron que el algoritmo propuesto funciona mejor que el CLONALG estándar y un algoritmo genético.

Muthreja en [8] realizó la implementación del algoritmo de selección clonal y propuso una versión modificada para resolver el problema del agente viajero, la distancia más corta y el tiempo de convergencia más corto fueron utilizados para comparar el rendimiento de ambos algoritmos con dos algoritmos convencionales: un algoritmo genético y el algoritmo Ramificación y Poda (*Branch and Bound, BB*). Los resultados mostraron que el algoritmo de selección clonal y el propuesto dieron mejores soluciones. Lin en [9] propuso un algoritmo de búsqueda hibrido utilizando recocido simulado y el algoritmo de búsqueda tabú para resolver el PAV, además proponen parámetros adaptativos que el algoritmo puede ajustar automáticamente en función de los ejemplos, para probar su algoritmo utilizaron TSPLIB y los resultados mostraron que el algoritmo propuesto pudo encontrar soluciones satisfactorias en un periodo de tiempo razonable, además demostraron una mejora en la precisión y eficiencia comparado con otros algoritmos.

Para resolver el PAV además de los algoritmos implementados en este trabajo, se han utilizado otros algoritmos de búsqueda heurísticos, algunos trabajos relacionados son:

Wang en [10] propuso un algoritmo hibrido inteligente, combinó el algoritmo ABC y el algoritmo PSO, los resultados experimentales mostraron que el algoritmo hibrido es mejor comparado otros algoritmos similares, además tiene resultados satisfactorios en el PAV a gran escala. Myszkowski en [11] combinó los mecanismos de un algoritmo genético de manera única para resolver el PAV multiobjetivo. Wei en [12] realizó un estudio de tres algoritmos heurísticos para resolver el PAV: un AG, PSO y ACO, aplicó una estrategia de inicialización y búsqueda local para acelerar aún más la velocidad de búsqueda. Li en [13] utilizó el algoritmo ABC y le agrego una estrategia de búsqueda local para optimizar el camino, sus resultados mostraron que el algoritmo propuesto es muy confiable para resolver el PAV.

Hammouri en [14] utilizó un algoritmo metaheurístico denominado Algoritmo Libélula (*Dragonfly Algorithm, DA*), para evaluar su calidad se comparó con otros algoritmos metaheurísticos utilizando el mismo conjunto de datos tomado de TSPLIB, los resultados finales mostraron que el algoritmo propuesto es capaz de resolver eficientemente el PAV, produciendo resultados competitivos. Chandrawati en [15] utilizó el Algoritmo Luciérnaga (*Firefly Algorithm, FA*) para resolver tres problemas: planificación de rutas, enrutamiento de vehículos y el agente viajero. Juneja en [16] utilizó un algoritmo genético para encontrar una solución factible para el PAV. Jaradat en [17] propuso una solución para el PAV utilizando el Algoritmo Luciérnaga y el agrupamiento de K-Means, el enfoque propuesto comprende tres pasos principales: agrupar los nodos, encontrar la ruta óptima en cada grupo y volver a conectar los grupos.

Algunos trabajos relacionados para resolver otros problemas semejantes al PAV que han utilizado algoritmos heurísticos son:

Flórez en [18] presentó una evaluación de dos metaheurísticas para resolver el problema de planificación de trabajos (*Job Shop Scheduling*): sistemas inmunes artificiales y el ACO. El objetivo era encontrar la secuencia de trabajos que requirieran la menor cantidad de tiempo para ser ejecutada en las maquinas disponibles. Para medir el desempeño de ambos algoritmos compararon la calidad de las soluciones obtenidas con respecto a la mejor solución conocida. Vairamuthu en [19] utilizó CLONALG para resolver el problema de la planificación de N-trabajos, la implementación fue realizada en Matlab.

3. El problema del agente viajero

El problema del agente viajero se puede definir como un problema donde existen varias ciudades (nodos) y dada una ciudad de partida (nodo inicial), el agente viajero debe visitar cada cuidad solo una vez y regrese a la cuidad de partida, de manera que se minimice la distancia del recorrido [20, 21].

Matemáticamente se puede definir como un grafo $G = (V, E)$, los vértices son las ciudades y las aristas corresponden a las carreteras que unen dos ciudades [22]. V es un vector de ciudades $V = \{1, 2, 3, \dots, \mathrm{n}\}$ donde n es el número de ciudades. Entonces, $E = \{(\mathrm{i}, \mathrm{j}) : \mathrm{i}, \mathrm{j} \in \mathrm{V}, \mathrm{i} \neq \mathrm{j}\}$ es un conjunto de aristas. Una matriz de costos $C = (c_{ij})$ se define en E.

Los vértices son puntos $P_i = (x_i, y_i)$ y $P_j = (x_j, y_j)$. Entonces, la distancia entre estos puntos se define en (1):

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\,. \tag{1}$$

Por lo tanto, el problema se podría formalizar como minimizar (2):

$$f(x) = \sum_{i,j}^{n} c_{ij} \tag{2}$$

para encontrar la distancia total del recorrido más corto, visitando cada ciudad exactamente una vez y regresando a la ciudad de partida [1, 21].

Existen varios tipos de PAV, un PAV simétrico es cuando $c_{ij} = c_{ji}$, para todo i, j y asimétrico en caso contrario, para un PAV asimétrica de n ciudades existen $(n-1)!$ soluciones posibles, mientras que para un simétrico existen $\frac{(n-1)!}{2}$ soluciones posibles. En ambos casos no es posible realizar una búsqueda exhaustiva debido a que la cantidad de soluciones puede llegar a ser muy grande, ya que cuando n crece se hace imposible encontrar el camino más corto en tiempo polinomial [1, 2 y 23]. Otra variante es el PAV Aleatorio, donde las coordenadas de las ciudades se generan aleatoriamente en lugar de tener ejemplos estáticos ya disponibles en TSPLIB.

PAV sigue siendo investigado por tres razones. Primero puede modelar una gran cantidad de problemas del mundo real. Segundo se demostró que era un problema NP-difícil. Tercero los problemas NP- difícil son intratables es decir no se ha encontrado una manera eficiente de resolverlos [20]. Además, los problemas NP- difícil son más o menos equivalentes entre sí, si se logrará resolver uno de ellos, podría resolverse el resto.

4. Algoritmos metaheurísticos

En PAV la búsqueda exhaustiva es ineficiente debido al gran número de soluciones posibles, incluso cuando el problema es de tamaño moderado. Por lo tanto, el objetivo es obtener heurísticamente buenas soluciones en tiempo razonable.

4.1. Recocido simulado

Recocido Simulado es un algoritmo probabilístico inspirado en el recocido metalúrgico, que es una técnica de enfriamiento controlado para reducir defectos [1, 21]. Comienza con una mezcla de metal a alta temperatura y enfría lentamente la mezcla, permitiendo la formación de estructuras optimas a medida que el material se enfría [3].

El algoritmo se describe formalmente de la siguiente manera: comienza con una solución aleatoria x_p, luego se genera una solución vecina aleatoria x_n en cada iteración. Después se calcula la diferencia entre los valores de la función objetivo $\Delta f = f(x_n) - f(x_p)$. Si la solución vecina es una mejor solución (es decir, se mejora la función objetivo) entonces reemplazará a la solución actual.

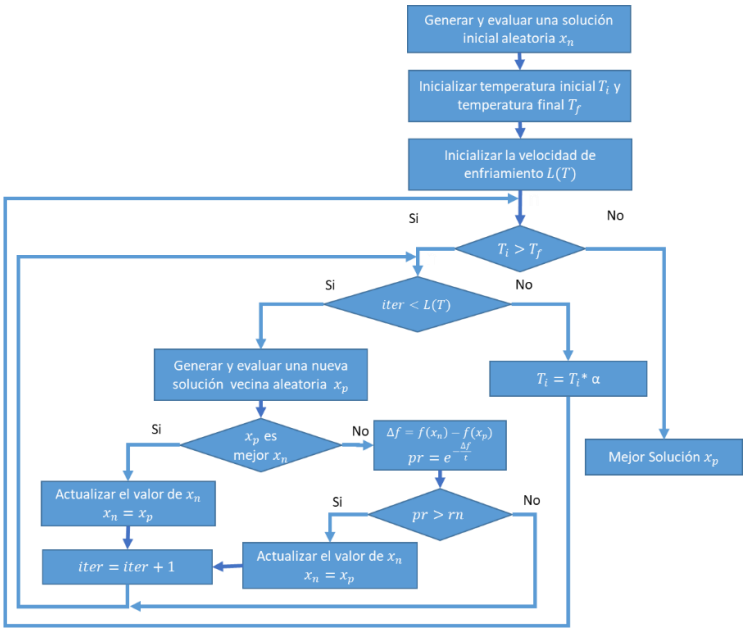


Fig. 1. Diagrama general de Recocido Simulado [1].

Si la solución es peor, se puede elegir reemplazar la solución actual dependiendo de una probabilidad definida en (3) que depende de la temperatura [1, 21 y 24]:

$$pr = e^{-\frac{\Delta f}{t}}, \tag{3}$$

donde pr se reduce a medida que el algoritmo avanza y (t) es la temperatura o el parámetro de control, esto significa que a medida que el algoritmo avanza, el parámetro de temperatura disminuye, dando a las peores soluciones una menor posibilidad de reemplazar a la solución actual [21, 24], es decir las peores soluciones en las primeras iteraciones del algoritmo evitan que la solución converja a un mínimo local, lo que significa que aceptar peores soluciones impide que el algoritmo quede atrapado en los mínimos locales y no pueda explorar más soluciones posibles. Esta aceptación se logra generando un número aleatorio (rn) entre $0 \leq rn \leq 1$ y después se compara con el umbral. Si $pr > rn,$ la solución actual es reemplazada por la solución vecina. El procedimiento se repite hasta que se cumpla la condición de paro.

Los parámetros que afectan el resultado del algoritmo son tres: la temperatura inicial (T_i), la velocidad a la que disminuye la temperatura (es decir, velocidad de enfriamiento) y la condición de parada del algoritmo [1, 21]. La velocidad de enfriamiento $L(t)$ es el número de iteraciones que usan la misma temperatura antes de disminuirla para buscar una mejor solución vecina, es decir se realiza una búsqueda local (conocida como Ciclo de Metrópolis). La velocidad de enfriamiento α es el grado de disminución de la temperatura conforme avanzan los ciclos hasta llegar a la temperatura final (T_f), se recomienda que este entre (0.8, 0.99). El algoritmo RS se describe en el diagrama de flujo de la Fig. 1.

Una ventaja de RS en comparación con otros métodos es su capacidad para no quedar estancado en mínimos locales, lo que significa que el algoritmo no siempre rechaza los cambios. Es decir, es un método probabilístico utilizado para obtener el mínimo de una función que puede tener varios mínimos locales [3].

4.2. Algoritmo de selección clonal

En los últimos años ha habido un gran interés en estudiar sistemas inspirados biológicamente para resolver problemas de optimización [8]. Los sistemas inmunes artificiales (SIA) son una clase de sistemas computacionalmente inteligentes inspirados en los principios y procesos del sistema inmune [19, 25].

El algoritmo de selección clonal es un algoritmo de optimización evolutivo inspirado en los mecanismos del sistema inmune biológico que tienen un tremendo potencial en muchas aplicaciones de optimización [8, 26].

El sistema inmune del cuerpo humano es capaz de reconocer las células extrañas que entran en un cuerpo humano que pueden causar enfermedades. Estas células extrañas se llaman antígenos, nuestro cuerpo aprende a neutralizar los efectos de los antígenos al comprender el patrón de comportamiento y oponiéndose a ellos para mantener la estabilidad del cuerpo. Las células producidas por el cuerpo humano para luchar con antígenos se denominan anticuerpos, el sistema inmune controla la acción del antígeno. El algoritmo de selección clonal tiene la funcionalidad de filtrar los anticuerpos en función de la afinidad. El grado de reconocimiento entre anticuerpo y antígeno se denomina afinidad. Cuanto mejor sea el reconocimiento, mayor será la afinidad y viceversa. Los anticuerpos capaces de luchar contra los antígenos son capaces de multiplicarse a través de un proceso de clonación proporcional a la afinidad, para luchar contra ellos [8].

La teoría de la selección clonal incluye tres procesos: proceso de selección clonal, proceso de proliferación, y proceso de maduración. Cuando un antígeno invade será reconocido por las células de sistema inmune con cierta afinidad. En la etapa de selección se activan las células con alta afinidad para proliferar, las cuales son estimuladas para producir grandes cantidades de clones, es decir realizar el proceso de proliferación. En la etapa final, en el proceso de maduración estos clones se pueden mutar y se convierten en células que secretan grandes cantidades de anticuerpos que retiene el patrón antígeno para futuras infecciones [27]. Así de manera semejante funciona el algoritmo de selección clonal (también llamado CLONALG), el cual se compone de cuatro procesos: selección, expansión clonal o proliferación, hipermutación (madurez de afinidad) y re-selección. La idea principal de CLONALG es que la mutación acumulativa puede dirigir la búsqueda a mejores soluciones. Este algoritmo emplea un operador de clonación e hipermutación en lugar de la operación de selección y cruce como en AG [7].

Básicamente el proceso del algoritmo de selección clonal consiste en generar aleatoriamente una población de soluciones candidatas; después seleccionamos un porcentaje de los mejores individuos, los cuales son clonados, luego a estos individuos se les aplica una hipermuta y finalmente seleccionamos la nueva población para la siguiente generación [26], este proceso se puede observar en la Fig. 2.

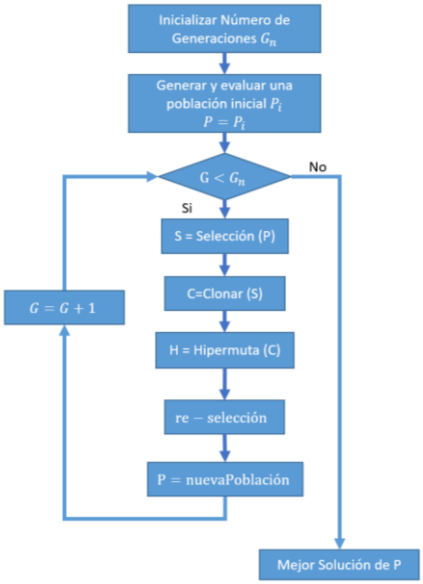


Fig. 2. Diagrama general del Algoritmo de Selección Clonal [8].

El CLONALG puede considerarse como un algoritmo evolutivo basado en la población. Cada individuo (anticuerpo) es esencialmente un buscador local óptimo, y la hipermutación puede considerarse como un proceso de exploración de vecindad ciega en el espacio de búsqueda en relación con el anticuerpo original. En el proceso de re-selección, los anticuerpos proliferados se seleccionan de acuerdo con sus valores de afinidad. Después de un número suficiente de generaciones, se espera que los anticuerpos encuentren las soluciones óptimas o las soluciones óptimas aproximadas [7].

El CLONALG ha sido ampliamente utilizado para resolver una variedad de problemas como reconocimiento de patrones, la optimización de funciones multimodales y el problema del agente viajero [7, 25].

5. Implementación

En esta sección se explica cómo se llevó a cabo la implementación de los algoritmos recocido simulado y el algoritmo de selección clonal para resolver el PAV. Ambos algoritmos están codificados en Scilab 6.0.1.

Un aspecto muy importante para ambos algoritmos es la representación de una solución al problema. Existen principalmente dos métodos para representar el recorrido del PAV: representación de adyacencia y representación de ruta.

Para ambos algoritmos se ha utilizado la representación de ruta, es decir, una solución es representada por un individuo de longitud n, donde n es el número de nodos en el problema. El tipo de representación usada es por permutaciones donde cada gen del cromosoma es una etiqueta del nodo de tal forma que ningún nodo puede aparecer dos veces en el mismo cromosoma.

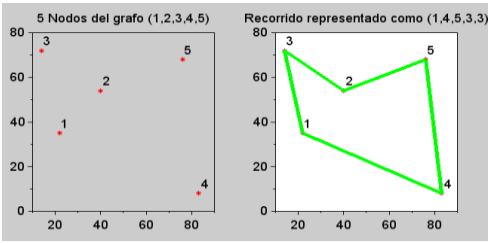


Fig. 3. (a) Nodos del grafo y (b) Recorrido representado por (1,4,5,2,3).

Por ejemplo, si {1, 2, 3, 4, 5} son las etiquetas de los 5 nodos del grafo, entonces un recorrido puede ser representado como (1, 4, 5, 2, 3), como se muestra en la Fig. 3.

5.1. Recocido simulado

Ahora nos centramos en la aplicación de recocido simulado al PAV. Primero se genera una semilla es decir una solución inicial (recorrido aleatorio). Para evaluar una solución se considera como función de aptitud definida en (2) donde $f(x)$ calcula el costo (o valor) del recorrido representado por una solución y la distancia entre estos puntos se define en (3).

Otro aspecto importante en este algoritmo es la generación de la solución vecina, la cual se crea eligiendo aleatoriamente dos posiciones en la solución y después se intercambian los genes (ciudades) localizados en dichas posiciones. Para el ejemplo anterior de 5 ciudades si se eligen las posiciones 2 y 4 tenemos [20]:

(1 4 5 2 3) Solución actual
  | | Posiciones seleccionadas
(1 2 5 4 3) Solución vecina

Para el enfriamiento, se ha utilizado un cambio de temperatura lineal. Esto significa que cada nueva temperatura se determina a partir de la temperatura anterior.

5.2. Algoritmo de selección clonal

El CLONALG se puede describir de la siguiente manera [7, 18 y 25]:

Paso 1. Generar aleatoriamente una población de individuos (M – anticuerpos); es un conjunto de soluciones potenciales, es decir un conjunto de recorridos, donde cada uno representa un posible recorrido,

Paso 2. Se ejecuta de manera iterativa. Para cada generación:

Paso 3. Determinar la afinidad de la población, es decir evaluar cada solución con respecto a la función f a optimizar de la ecuación (2) y seleccionar los n mejores individuos (selección).

Paso 4. Reproducir los n mejores individuos (la mitad de la población inicial) generando una población temporal de clones Mc, se clona con el mismo número de copias

Tabla 1. Parámetros usados para Recocido Simulado y CLONALG.

Recocido Simulado	Valores	CLONALG	Valores
Temperatura inicial	20-50	Número de Generaciones	500-5000
Velocidad de enfriamiento	20-50	Factor de clonación	50%
Grado de Enfriamiento	0.97	Factor de mutación	10%
Número de Ciudades	10-50	Número de Ciudades	10-50
		Tamaño de la población	200

(expansión clonal). La cantidad n de clones, está dado por la ecuación: $\mathrm{Mc} = (\beta * \mathrm{M})$, donde β es un factor de clonación y M es la cantidad de individuos. El factor de clonación el valor de 0,1.

Paso 5. Todas las copias clonadas se someten a un proceso de hipermutación con una tasa proporcional a su afinidad, cuanto mayor sea la afinidad, menor la tasa de mutación. Así se genera una nueva población de clones (C).

Paso 6. Determinar el valor de afinidad de todos los individuos mutados, se agregan estos individuos mutados a la población y se seleccionan M de estos individuos maduros para la siguiente generación (re-selección).

Paso 7. El algoritmo terminará después de un número de generaciones/iteraciones o si se ha encontrado la solución óptima para el caso de las instancias tomadas de TSPLIB.

6. Pruebas y resultados

Los dos algoritmos fueron desarrollados en SCILAB y han sido ejecutados en Windows 10 utilizando un CPU core i5 y 16 GB de RAM. Para comparar ambos algoritmos se realizaron dos pruebas. En la primera se hicieron experimentos sobre instancias de PAV generadas aleatoriamente de entre 10 y 30 ciudades. En la segunda se han probado en instancias de referencia entre 10 y 50 ciudades, las cuales fueron tomadas de TSPLIB [28].

A continuación, se definen los parámetros utilizados en las pruebas, para el recocido simulado: la temperatura inicial es de 20 a 50, la velocidad de enfriamiento es de 20 a 50 y el grado de enfriamiento es de 0.97. Para el algoritmo de selección clonal: el tamaño de la población es de 100 a 300, el factor de clonación 0.5, el factor de mutación 0.1 y un máximo de 500 a 5000 iteraciones.

El número de iteraciones fue determinado dependiendo del número de ciudades en cada problema. En ambos algoritmos los parámetros han sido determinados basados en las recomendaciones de la literatura y después de ejecutar los algoritmos, se obtuvieron las mejores soluciones utilizando los parámetros mostrados en las Tabla 1.

Para evaluar a los algoritmos se realizaron dos pruebas. En ambas pruebas los experimentos se realizaron 10 veces para cada caso, obteniendo el mejor recorrido, el peor, la media el número de generaciones del recorrido para cada algoritmo.

Tabla 2. Resultados para 10, 20 y 30 ciudades generadas aleatoriamente.

Número de nodos	Algoritmo	Mejor	Peor	Media	Número de generaciones
10	RS	270.35	270.35	270.35	40
	CLONALG	270.35	270.35	270.35	30
20	RS	382.5784	407.8863	391.7068	100
	CLONALG	382.5784	402.4819	391.8037	400
30	RS	510.4583	512.4537	511.3585	500
	CLONALG	510.3138	513.1765	511.5687	1000

Tabla 3. Resultados para instancias de referencia tomadas de TSPLIB [28].

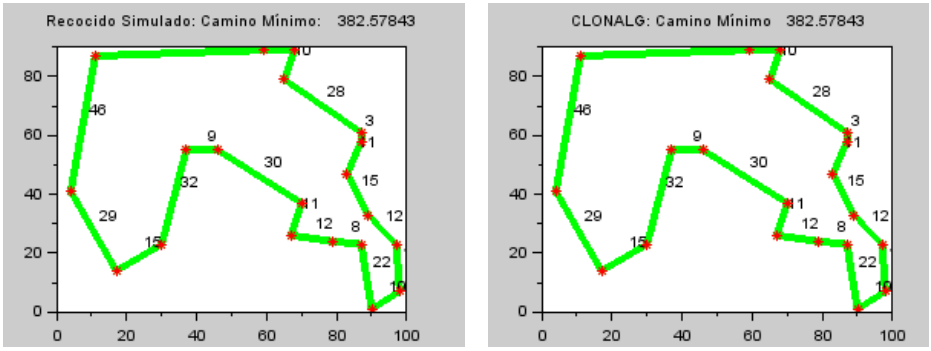
Instancia	Óptimo	Algoritmo	Mejor	Peor	Media
Ulysses16	74.1087	RS	75.08	79.23	76.96
		CLONALG	73.98	74.14	74.04
Ulysses22	75.6651	RS	78.4622	87.2333	83.2954
		CLONALG	75.7948	77.3688	76.4040
Bayg29	9074.1480	RS	9229.2592	9593.4947	9470.5960
		CLONALG	9074.1480	9879.7366	9594.0454
Att48	33523.709	RS	33936.386	34881.618	34398.193
		CLONALG	34700.472	38108.189	35727.915

En la primera prueba los algoritmos son aplicados a conjuntos de datos con diferente número de ciudades generadas aleatoriamente. Los resultados experimentales se presentan en la Tabla 2.

La Tabla 3 muestra los resultados de la segunda prueba se consideraron instancias entre 10 y 50 ciudades tomadas de la librería TSPLIB [28], que es una biblioteca de instancias de muestra para el TSP de varias fuentes y de varios tipos. Las instancias elegidas son de tipo PAV simétrico, donde la distancia del nodo i al nodo j es la misma que la del nodo j al nodo i. Además, se conocía la ruta óptima.

Cuatro instancias han sido seleccionadas de la librería TSPLIB, con diferente número de ciudades. La primera instancia es llamada Ulysses16, este tiene 16 ciudades de la Odisea de Ulises, que representa el viaje de Ulises en la época fenicia a las 16 maravillas del Mediterráneo y el recorrido mínimo es de 74.1087. Ulysses22 tiene 22 ciudades de la Odisea de Ulises y el recorrido mínimo es de 75.6651. Bayg29 tiene las distancias geográficas de 29 ciudades de Baviera, Alemania y el recorrido mínimo es de 9074.1480. Att48 tiene 48 capitales de los EE. UU y el recorrido mínimo es de 33523.709.

La Fig. 4 muestra los mejores resultados para 20 ciudades generadas aleatoriamente de la primera prueba, podemos observar en la Tabla 2 que ambos algoritmos encuentran el mejor camino para 10 y 20 ciudades, aunque RS para 20 ciudades encuentra la peor



(a) Recocido Simulado, mejor solución para 20 ciudades (b) CLONALG, mejor solución para 20 ciudades

Fig. 4. Mejores resultados para ciudades generadas aleatoriamente.

ruta (407.8863), aun así, su media de 391.7068 es casi igual a CLONALG con valor de 391.8037, indicando que RS encuentra más veces una mejor ruta.

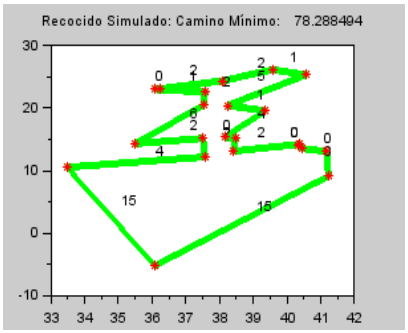
La Fig. 5 muestra los mejores resultados para los problemas ulysses22, bayg29 y att48 de la segunda prueba para ambos algoritmos.

Podemos observar en los resultados de la Tabla 2 y las imágenes de la Fig. 5 que el algoritmo de RS es mejor que CLONALG para un número de ciudades más grande, lo podemos ver en la instancia Att48 con 48 ciudades, donde RS encuentra la ruta más cercana a la mejor ruta con valor de 33936.386, además la media de RS (34398.193) en comparación con la de CLONALG (35727.915) muestran que RS encuentra buenas soluciones. Por otro lado, en la instancia bayg29 con 29 ciudades, RS también tiene mejor promedio (9470.5960) que CLONALG (9594.0454), aunque CLONALG encuentra la peor ruta con valor de 9879.7366, CLONALG sí encuentra la ruta optima (9074.1480), sin embargo, RS nunca encuentra la ruta optima. Ahora para instancias con un número pequeño de ciudades como en Ulysses16 y Ulysses22, CLONALG tiene mejores resultados encontrando la mejor ruta (73.9) para Ulysses16 y acercándose mucho a la mejor ruta para Ulysses22.

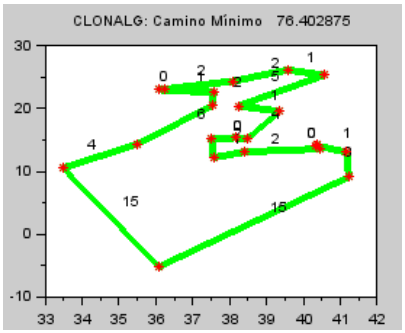
Recocido simulado casi siempre encuentra la mejor ruta en menos iteraciones, se puede observar en la Fig.6. La Fig. 6(a) muestra la evolución de la función objetivo de RS y CLONALG para 30 ciudades aleatorias, se muestra el costo del recorrido sobre el número de generaciones. También cabe destacar que en la mayoría de los casos RS encuentra una solución aceptable en menos generaciones. Finalmente, la Fig. 6(b) muestra la evaluación de RS y de CLONALG para 20 ciudades generadas aleatoriamente.

Podemos observar en la Fig. 6 que cuando las ciudades son generadas aleatoriamente el algoritmo de Recocido Simulado llega primero al valor óptimo, lo que no CLONALG, el cual necesita más generaciones para encontrar el valor óptimo.

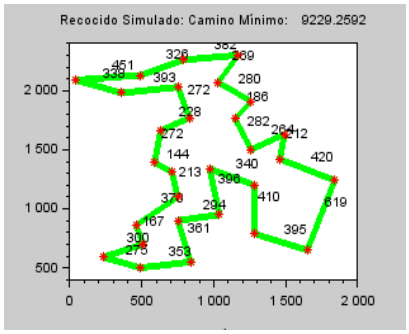
Recocido Simulado realiza una búsqueda más extensa, lo que le permite ser mejor buscando en instancias con un número de ciudades grande, pero a veces no encuentre la mejor solución.



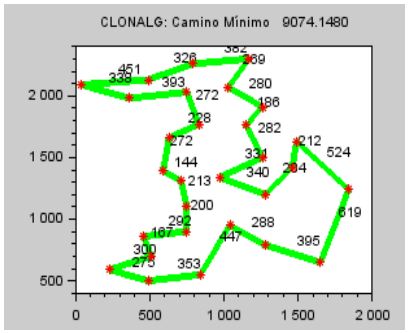
(a) Recocido Simulado, mejor solución para Ulysses22



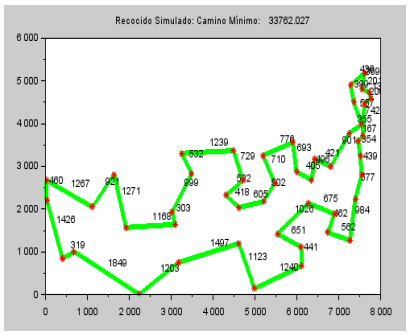
(b) CLONALG, mejor solución para Ulysses22



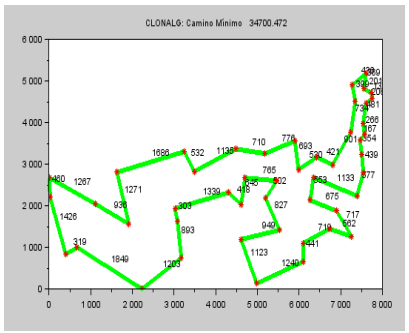
(c) Recocido Simulado, mejor solución para Bayg29



(d) CLONALG, mejor solución para Bayg29



(e) Recocido Simulado, mejor solución para Att48

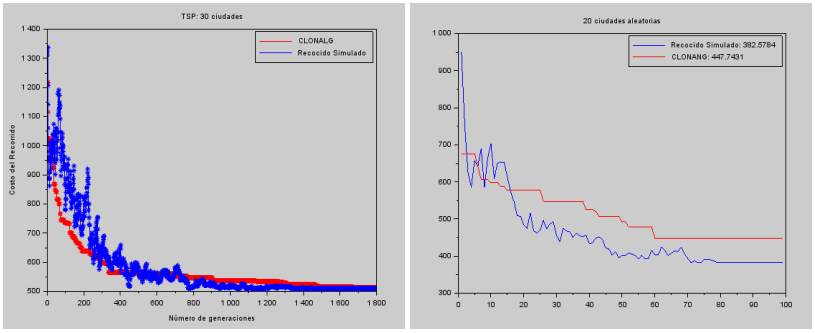


(f) CLONALG, mejor solución para Att48

Fig. 5. Mejores resultados. (a) Recocido Simulado y (b) CLONALG para Ulysses22. (c) Recocido Simulado y (d) CLONALG para Bayg29, (e) Recocido Simulado y (f) CLONALG para Att48.

7. Conclusiones y trabajo futuro

En esta investigación presentamos la implementación de dos algoritmos de optimización heurísticos, Recocido Simulado y el Algoritmo de Selección Clonal aplicados a resolver el Problema del Agente Viajero.



(a) 30 ciudades aleatorias (b) 20 ciudades aleatorias

Fig. 6. Evolución de los resultados de Recocido Simulado y CLONALG para (a) 30 ciudades generadas aleatoriamente y (b) 20 ciudades generadas aleatoriamente.

Para realizar una evaluación de ambos algoritmos se realizaron dos pruebas. En la primera prueba los algoritmos son aplicados a conjuntos de datos con diferente número de ciudades generadas aleatoriamente y la segunda prueba se realizó sobre instancias tomadas de la librería TSPLIB. Los algoritmos están codificados en Scilab.

Los dos algoritmos han sido estudiados por separado para solucionar este problema y ambos han mostrado ser eficientes respecto a otros algoritmos. En los resultados de este trabajo ambos algoritmos proporcionaron buenos resultados para problemas con un número de ciudades pequeños, pero si el tamaño aumenta pueden no encontrar la ruta optima.

En la primera prueba los dos algoritmos encuentran el mejor camino para 10 y 20 ciudades, aunque la diferencia entre el promedio de RS (391.7068) en comparación con CLONALG (391.8037) es mínima, saber que RS encuentra la peor ruta con valor de 407.8863, indica que RS encuentra más veces una mejor ruta.

En la segunda prueba, RS es mejor en la instancia Att48 con 48 ciudades, RS encuentra la ruta más cercana al óptimo con valor de 33936.386, además su media (34398.193) es mejor en comparación con la de CLONALG (35727.915), indicando que encuentra mejores soluciones. Pero para bayg29 con 29 ciudades CLONALG sí encuentra la ruta optima (9074.1480), sin embargo, RS nunca encuentra la ruta optima.

El tiempo de ejecución y la distancia de la ruta aumentan con un número creciente de ciudades. El algoritmo de Recocido Simulado realiza muchos saltos buscando una mejor solución, ocasionando que a veces no encuentre la mejor solución, pero más soluciones en el espacio de búsqueda, ocasionando que encuentra buenos resultados cuando el número de ciudades es mayor.

En un futuro se pretende probar con más instancias, así como implementar alguna variante de los algoritmos originales. También se pretende comparar su desempeño con otros algoritmos heurísticos presentes en la literatura como el Algoritmo Luciérnagas o el Algoritmo Libélula. Por último, se quiere aplicar a problemas reales de la región de Oaxaca. Algunas aplicaciones del PAV en situaciones de la vida real son en horarios de transporte, en procesos de fabricación, en ordenar n trabajos y reparto de productos

por mencionar algunos. Con esta investigación se pretende encontrar el algoritmo más apropiado y eficiente para aplicarlo a problemas reales en municipios de la región.

Referencias

1. Alhanjouri-Mohammed, A.: Optimization techniques for solving travelling salesman problem. International Journal of Advanced Research in Computer Science and Software Engineering, 7(3), IJARCSSE, pp. 165−174 (2017)
2. Sumathi, M., Rahamathunnisa, U., Anitha, A., Das, D., Nallakaruppan, M.K.: Comparison of particle swarm optimization and simulated annealing applied to travelling salesman problem. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(6), pp. 1578−1583 (2019)
3. Adewole, A.P., Otubamowo, K., Egunjobi, T.O.: A comparative study of simulated annealing and genetic algorithm for solving the travelling salesman problem. International Journal of Applied Information Systems (IJAIS), 4(4), pp. 6−12 (2012)
4. Shimomura, M., Takashima, Y.: Application of monte-carlo tree search to traveling-salesman problem. In: 20th Workshop on Synthesis and System Integration of Mixed Information Technologies, (SASIMI) Proceedings, pp. 352−356 (2016)
5. Panda, M.: Performance comparison of genetic algorithm, particle swarm optimization and simulated annealing applied to TSP. International Journal of Applied Engineering Research, 13(9), pp. 6808−6816 (2018)
6. Zhou, A., Zhu, L., Hu, B., Deng, S., Song, Y., Qiu, H., Pan, S.: Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. Information, 10 (2019)
7. Pang, W., Wang, K., Wang, Y., Ou, G., Li, H., Huang, L.: Clonal selection algorithm for solving permutation optimisation problems: a case study of travelling salesman problem. International Conference on Logistics Engineering, Management and Computer Science (LEMCS), Atlantis Press, pp. 575−580 (2015)
8. Muthreja, I., Kaur, D.: A comparative analysis of immune system inspired algorithms for traveling salesman problem. In: International Conference on Artificial Intelligence, ICAI'18, pp. 164−170 (2018)
9. Lin, Y., Bian, Z., Liu, X.: Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem. Appl. Soft Comput., 49, pp. 937–952 (2016)
10. Wang, Y.: Improving artificial bee colony and particle swarm optimization to solve TSP problem. In: International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Changsha, pp. 179–182 (2018)
11. Myszkowski, P.B., Laszczyk, M., Dziadek, K.: Non-dominated sorting tournament genetic algorithm for multi-objective travelling salesman problem. In: Federated Conference on Computer Science and Information Systems (FedCSIS) (2019)
12. Wei, F., Chen, W., Hu, X., Zhang, J.: An empirical study on evolutionary algorithms for traveling salesman problem. In: 9th International Conference on Information Science and Technology (ICIST), pp. 273–280 (2019)
13. Li, X., Zheng, Y.: Artificial bee colony algorithm and its application in traveling salesman problems. In: Chinese Control and Decision Conference (CCDC), Nanchang, China, pp. 1–5 (2019)

14. Hammouri, A.I., Samra, E.T.A., Al-Betar, M.A., Khalil, R.M., Alasmer, Z., Kanan, M.: A dragonfly algorithm for solving traveling salesman problem. In: 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, pp. 136–141 (2018)
15. Chandrawati, T.B., Sari, R.F.: A review of firefly algorithms for path planning, vehicle routing and traveling salesman problems. In: 2nd International Conference on Electrical Engineering and Informatics (ICon EEI), pp. 30–35 (2018)
16. Juneja, S.S., Saraswat, P., Singh, K., Sharma, J., Majumdar, R., Chowdhary, S.: Travelling salesman problem optimization using genetic algorithm. In: Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, pp. 264–268 (2019)
17. Jaradat, A., Matalkeh, B., Diabat, W., Solving traveling salesman problem using firefly algorithm and k-means clustering. In: IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, pp. 586–589 (2019)
18. Flórez, E., Díaz, N., Gómez, W., Bautista, L., Delgado, D.: Evaluación de algoritmos bioinspirados para la solución del problema de planificación de trabajos. I+D Revista de Investigaciones, 11(1), pp. 141–155 (2018)
19. Vairamuthu, M.K., Porselvi, S., Balaji, D.H., Babu, J.R.: Artificial immune system algorithm for multi objective flow shop scheduling problem. International Journal of Innovative Research in Science, Engineering and Technology, 3(3), pp. 1391–1395 (2014)
20. Chandekar, N., Jayachandran Pillai, M.: A comparative study of GA and ACO for solving travelling salesman problem. International Journal of Mechanical and Production Engineering (IJMPE), 5(11), pp. 34–37 (2017)
21. Sureja, N.M., Chawda, B.V.: Random travelling salesman problem using SA. International Journal of Emerging Technology and Advanced Engineering, 2(4), pp. 621–624 (2012)
22. Taiwo, O.S., Mayowa, O.O., Ruka, B.K.: Application of genetic algorithm to solve traveling salesman problem. International Journal of Advance Research, (IJOAR), 1(4), pp. 27–46 (2013)
23. Dwivedi, V., Chauhan, T., Saxena, S., Agrawal, P.: Travelling salesman problem using genetic algorithm. International Journal of Computer Applications (IJCA) National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI), pp. 25–30 (2012)
24. Elhaddad, Y.R., Sallabi, O.M.: A new hybrid genetic and simulated annealing algorithm to solve the traveling salesman problem. In: Proceedings of the World Congress on Engineering, WCE 2010, I (2010)
25. Revathi, M., Arthi, K.: Application of artificial immune system algorithms in dataset classification. International Journal of Innovative Research in Advanced Engineering (IJIRAE), 1(6), pp. 291–293 (2014)
26. Ortiz-Aguilar, L.D., Valadez, J.M., Soberanes, H.J., González, C.L., Ramírez, C.L., Soria-Alcaraz, J.A.: Comparativa de algoritmos bioinspirados aplicados al problema de calendarización de horarios. Research in Computing Science, 94, pp. 33–43 (2015)
27. Zhu, Y., Gao, S., Dai, H., Li, F., Tang, Z.: Improved clonal algorithm and its application to traveling salesman problem. IJCSNS International Journal of Computer Science and Network Security, 7(8), pp. 109–113 (2007)
28. Reinelt, G.: TSPLIB. http://comopt.ifi.uniheidelberg.de/software/TPLIB95/ (2020)

ISSN 1870-4069

Toward a Natural-Language Interface in Ontologies for Disambiguation

Alejandro Solís-Sánchez[1], Pablo Barraza Cornejo[1], Juan Acosta-Guadarrama[1,2], Juan Ferret[2]

[1] Universidad Autónoma de Ciudad Juárez, Mexico
[2] Instituto de Sistemas Filosóficos, USA

{al160509,al164006}@alumnos.uacj.mx, juan.acosta@uacj.mx

Abstract. With natural-language-interfaces-to-ontology (NLI2O) one can query a specific domain's data, stored into that particular conceptual structure, coded into specialized languages such as Ontology Web Language, OWL. To access such data, one requires complex query languages, such as SPARQL. NLI2Os translate human-language queries into SPARQL ones. In this work, we describe the architecture of an NLI2O. It receives a spoken question from an Android device. The spoken query becomes text and goes to a Java Server Page (JSP). The core of our interface runs on a JSP server, which receives the user's text query. It transforms it into a SPARQL query, later used to access, from the ontology to query, the information requested by the user. The interface uses several technologies to achieve that. Those are known as lemmatization, part-of-speech tagging, and tokenization. It also uses knowledge inference from the ontology of interest. It also uses clarification dialogues for disambiguation to answer user questions and revise the query. Besides that, some experiments suggest further studies like the disambiguation of an original text.

Keywords: Natural language interfaces to ontologies, disambiguation, SPARQL, OWL, semantic web.

1 Introduction

The Semantic Web is an extension of the current Web [2], where information maps to a well-defined meaning that can be understood both by computers and humans. The architecture of the Semantic Web that was established by Tim Berners-Lee makes use of ontologies as one of its principal components. An ontology is an explicit specification of conceptualization.

It allows us to understand the structure of knowledge better since it shows concepts and the relationships that exist between them [11]. Natural language[3]

[3] By natural language, we mean what people speak, such as English or any other.

interfaces (NLI) to ontologies translate a Natural Language query into a formal query language, employed to retrieve the knowledge expressed in one of the knowledge representation languages [7].

In the Semantic Web, ontologies store information and represent it through the Web Ontology Language, OWL[4], considered an extension of the *Resource Description Framework* RDF language [18]. Consulting knowledge stored in an ontology requires advanced skills for such purpose; an example is SPARQL (Protocol and RDF Query Language [20]) [14]. For this reason, interfaces that receive a query in natural language and transform it into a SPARQL query are necessary, with which they extract information users request.

The Semantic Web needs interfaces to query ontologies in a language closer to humans'. That facilitates interaction with sites or applications based on its architecture. In this work, we propose an architecture for an interface which, through queries in natural language, allows searching for information stored in an ontology, by using a mobile device for its operation. It is also capable of solving some kind of ambiguities through a simple clarification dialog, which is a promising feature to process non-monotonic languages in future works. The structure of this paper is the following. In Section 2, we add a description of related existing proposals; in Section 3, we describe our project; Section 4 describes the implementation and Section 5 the conclusions.

2 Related Works

There are different architectures proposed for the implementation of natural-language-interfaces-to-ontologies, NLI2O; they use different techniques of natural language processing (NLP) and formal query languages. This current work revisits and extends a preliminary version that appeared in [22], and we include further findings and software as a foundation of our claims[5]. In this section, we describe the architecture of some NLI2Os like FREyA [9], Gingseng [3], QuestIO [23], ORAKEL [4], and AquaLog [16], [22].

2.1 FREyA

FREyA (Feedback Refinement and Extended Vocabulary Aggregation) "is an NLI for querying ontologies which combines usability enhancement methods such as feedback and clarification dialogs," to improve recall and precision [10]. FREyA uses the knowledge available in the ontology to identify the terms found in the query, called *Ontology Concepts*—OC. If they are ambiguous, FREyA generates a clarification dialog in which the user chooses one of several alternatives. In the clarification dialog, the user selects an option that is stored and used to train the system. Starting from the OC, FREyA generates a set of triplets that converts to SPARQL queries to find the answer in the ontology. To

[4] Refer to `https://www.w3.org/TR/owl-features/`.
[5] Further examples and software are available at `http://k-lab.uacj.mx/NLI2O-20/`.

show the results to the user, it first identifies the type of response [9]. Then, it does a syntactic analysis and a search in the ontology to determine the *emphasis* of the question, for each answer type. A *focus* is a word or sequence of words that define a query and disambiguate it, showing what it is looking for [8]. The presentation of FREyA results includes a graph for visualization.

2.2 Ginseng

Ginseng (Guided Input Natural Language Search Engine) is another NLI based on a grammar of queries that extends dynamically through the structure of an ontology to guide users in the formulation of questions in a language similar to English. Based on the grammar, Ginseng translates queries to SPARQL, which allows its execution [3]. It provides query access to any OWL knowledge base. It guides the user to formulate questions, so there is no need to interpret them (logically o syntactically) and does not use any predefined vocabulary. Ginseng only knows the words defined by the currently considered ontologies, and the user has to follow it [3]. That can limit the possibilities of the user in general but ensures that all queries can get answered [3].

Ginseng's architecture consists of three parts: a multilevel grammar, an incremental parser, and an access layer to the ontology. The multilevel grammar consists of a static component that specifies the structures of generally possible query sentences and a dynamic one generated from the utilized ontologies. The static grammatical rules provide the structures and basic phrases for the questions in English. Each ontology loaded onto Ginseng creates the dynamic grammatical rules, and it uses them to extend the part of static syntax [3]. The incremental parser first uses the whole grammar to provide alternatives to the user while inputting the queries, and secondly to store information about how to build SPARQL queries. Finally, they use the Jena framework for the access layer to the ontologies. When the query execution becomes completed, Ginseng shows the generated SPARQL query and the results for the user [3]. The principal characteristic of Ginseng is that, unlike FREyA, queries that the user makes get controlled by a grammar based on the knowledge stored in the ontology. That gives rise to one of its advantages and equally its disadvantage since the grammar gives it control over what the user writes, and therefore, it avoids invalid questions, but at the same time limits the queries the user can make.

2.3 QuestIO System

QuestIO (Question-Based Interface to Ontologies) is an NLI to access structured information. It is domain independent and easy to use without training [23]. It is an open domain (or customizable to new ones with minimal cost), with undefined vocabulary; that is, it is derived automatically from the existing data in the knowledge base [23]. The system works by converting natural language queries to formal SeRQL (Sesame Rdf Query Language, pronounced "circle") questions [1], though other query languages can become used.

The QuestIO working process begins by processing the domain ontology, automatically creating a lexical dictionary from the obtained knowledge. From the ontology, the dictionary is capable of identifying mentions of classes, properties, instances, and property values associated with cases [23].

When QuestIO receives a query, the system does the following:

1. Make a linguistic analysis that consists of a morphological analysis of the text by a tokenizer and tagging tool.
2. Execute the ontological dictionary created at the system's start over the query's text. That creates annotations for all the mentions that the dictionary could identify—classes, properties, instances, and property values of data types.
3. Initiate an iterative transformation process to convert input text into a formal query. First, it separates the input text into tokens, determining each one's part of the discourse and adding annotations with its morphological route. Then, it tries to identify mentions of the ontology resources on the input text for generating a formal SeRQL query.
4. Finally, execute the query on the knowledge base and display the results.

QuestIO is an NLI that, unlike FREyA and Ginseng, uses SeRQL query language, which can be a disadvantage, since the official standard for consulting ontologies in the Semantic Web is SPARQL. Another problem is that it depends on a well-designed ontology for its operation, which generally requires a domain specialist for the ontology's design, making it less accessible to unskilled users.

2.4 ORAKEL

ORAKEL is an NLI to a knowledge base that transforms questions into a logical form [4] since it uses the F-logic language for representation of knowledge [15]. It receives a query in natural language as input, which gets converted to a first-order logic formula. Then, the logic formula gets transformed into the specific query language, which can be SPARQL or the F(rame)-Logic, F-logic, query language [5]. ORAKEL has the following main components.

1. Domain lexicon and general lexicon: a domain specialist creates both.
2. A knowledge base is composed of the domain ontology.
3. FrammeMapper: is a domain specialist that needs to know the underlying knowledge base.
4. Query Interpreter: builds a query formulated by the user to a logical form concerning domain predicates.
5. Query Converter: implemented in Prolog. It receives queries in logical form and translates them to knowledge base language (F-logic).
6. Answer generation.

Unlike other NLI shown in this work, ORAKEL is compatible with F-logic language for knowledge representation. Its main advantage is that it was designed to port NLI between domains efficiently. The disadvantage is that it requires a domain specialist for the lexicon generation.

2.5 AquaLog

Aqualog [16] is a portable question-answer system that takes as input queries expressed in natural language and an ontology and returns answers extracted from semantic markup compatible with the available ontology. Aqualog architecture can get characterized as a cascade model in which a natural language query gets translated into a set of intermediate representations based on triplets. Such triplets shall be compatible with the ontology. The triplet model Aqualog uses is of the form subject-predicate-object [16]. The main Aqualog modules are the linguistic component and the similarity relationship service.

The purpose of the linguistic component is to convert the natural language query to a query in triplets. Aqualog uses the GATE [6] infrastructure and resources to analyze the question as part of a linguistic component. GATE returns a set of syntactical annotations associated with the input query. These annotations include information about sentences, tokens, nouns, and verbs. Aqualog extends the set of annotations returned by GATE, identifying terms, relationships, question indicators (what/who/when/...), and patterns or types of questions.

Aqualog presents a solution that combines different strategies to give sense to a natural language query concerning the universe of discourse covered by the ontology. Using the GATE framework gives it the ability to improve the processing of natural language queries.

Previously analyzed NLI use ontologies to store knowledge, coded in XML files, and extracted through a query language. Except for QuestIO (which uses SeRQL), other interfaces use SPARQL, recommended by the W3C for the Semantic Web[6].

3 Proposed Architecture

The currently developed interface receives a natural language query made by the users in English, translates it to SPARQL [20], and through it, extracts the required information from the user's ontology. Its architecture has two main modules: a) knowledge-generating module (KGM) and b) interface module (IM). The KGM is in charge of generating the knowledge the IM needs to answer the users' natural language queries. In Section 3.1, we describe the KGM, and Section 3.2 describes the IM. In Figure 1, we present the proposed interface's general architecture.

3.1 Knowledge-generating Module

For the IM to be able to answer natural language queries, It is necessary to provide it of knowledge about the structure of the ontology the user wants to query and about the conversational linguistic domain. To facilitate portability to different user ontologies, the KGM is in charge of generating this knowledge, trying to minimize user intervention while configuring the IM.

[6] Refer to https://www.w3.org/blog/SW/2008/01/15/sparql_is_a_recommendation/

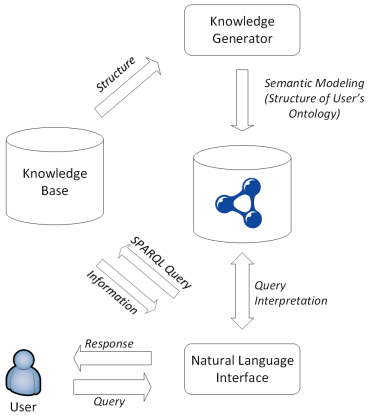


Fig. 1. General Architecture of Proposed Interface.

Firstly, the user indicates to the KGM the ontology they wish to query. Then, the KGM accesses the ontology and analyzes its structure; that is, it identifies the classes, object and data, and the way these elements are related. For this purpose, we used SPARQL query language, coded in OWL [18]. Through SPARQL queries, it can identify the features that make up the ontology's structure and access the stored information, generally modeled individually, that is, as instances of the ontology's classes.

As the next step, the identified elements get modeled semantically by using a semantic representation defined a priori for this purpose. The semantic representation was designed firstly in the software Protégé [25]. Then, to integrate the structure of this semantic representation in the KGM, we used the framework Apache Jena[7], which allows managing ontologies from the Java programming language, in which we designed our proposal.

Next, each of the modeled elements names get associated with synonyms and with words that share the same lemma. That gets done to provide the IM with linguistic knowledge about the domain of the user's ontology. The generated vocabulary becomes part of the semantic model.

Lastly, after generating the semantic model, the KGM stores it in an ontology, created dynamically. This ontology contains the knowledge the IM needs to answer natural language queries formulated by users. In Figure 2, we present the architecture corresponding to KGM, and in Figure 3, we show a fragment of the generated semantic model.

We generated the presented fragment from the ontology Geography.owl[8], used in the evaluation of the FREyA interface by Damljanovic [9]. At the same time, we generated this ontology from the deductive database made by Raymond J. Mooney [24], which contained information about the geography of the United

[7] Refer to `https://jena.apache.org/about_jena/about.html`.
[8] Refer to `https://github.com/danicadamljanovic/freya`.

States and got distributed as example data in Turbo Prolog 2.0. In Figure 4, we show the structure of the ontology Geography.owl.

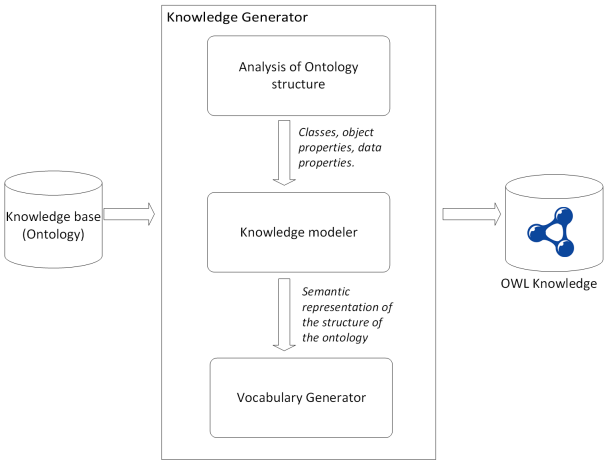


Fig. 2. Knowledge-Generating Module Architecture.

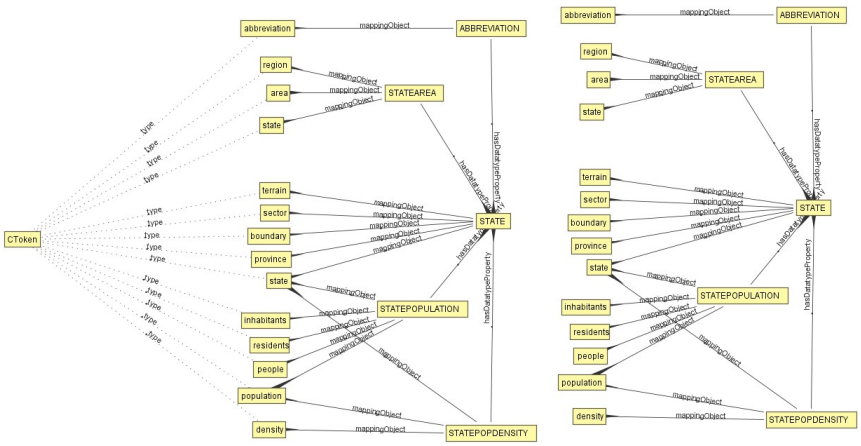


Fig. 3. Generated semantic model fragment.

The classes *CClass, CObjectProperty, CDatatypeProperty*, and *CToken*, make up the semantic representation of Figure 3, besides the object properties *hasDatatypeProperty, mappingObject*, and others.

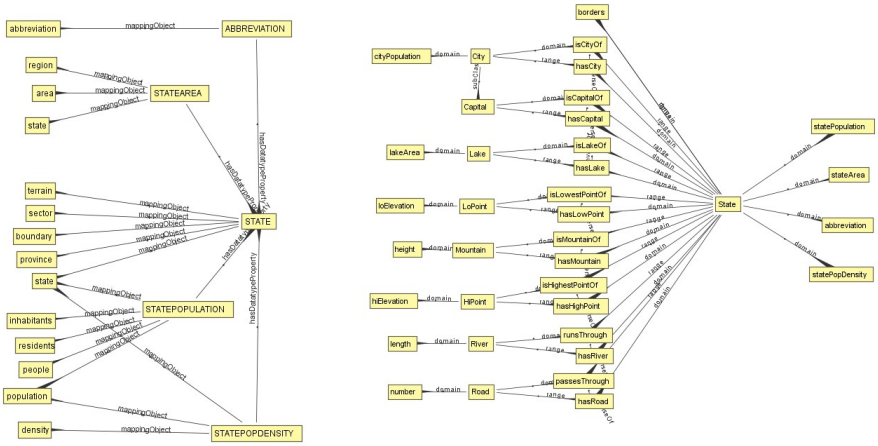


Fig. 4. Ontology structure for Geography.owl.

Figure 3 models the classes *State, Lake, City, Capital, Lake, LoPoint, Mountain, HiPoint, River*, and *Road*, shown in Figure 4 as individuals of the *CClass* class. Figure 3 models object properties *Borders, isCityOf, hasCity, isCapitalOf, hasCapital, isLakeOf, hasLake, isLowestPointOf, hasLowestPoint, isMountainOf, hasMountain, isHighestPointOf, hasHighestPoint, runsThrough, hasRiver, passesThrough* and *hasRoad*, shown in Figure 4, as individuals of the *CObjectProperty* class. Figure 3 models data properties *statePopulation, stateArea, abbreviation, statePopDensity, cityPopulation, lakeArea, loElevation, height, hiElevation, length* and *number*, shown in Figure 4, as class individuals *CDatatypeProperty*.

The class *CToken* is useful for modeling the interface vocabulary. Considering Figure 3, if the user introduced the word "*population*," IM would know that they are possibly referring to individuals *STATE* (CClass), *STATEPOPULATION* (CObjectProperty), or *STATEPOPDENSITY* (CObjectProperty). In this way, the IM uses the knowledge generated to interpret user's queries and generating the SPARQL query.

3.2 Interface Module

The Interface Module (IM) aims at translating the user's natural language query to a SPARQL query, with which it extracts from the user's ontology the requested information. Its operation is in Figure 5. The first step done by IM to generate the SPARQL query is to process the natural language of the user's query.

That phase consists of separating the user's query into a set of words, or isolated *tokens*. Afterward, it tags these tokens according to their grammatical function in the user's query. Lastly, it obtains the lemma for each one of the

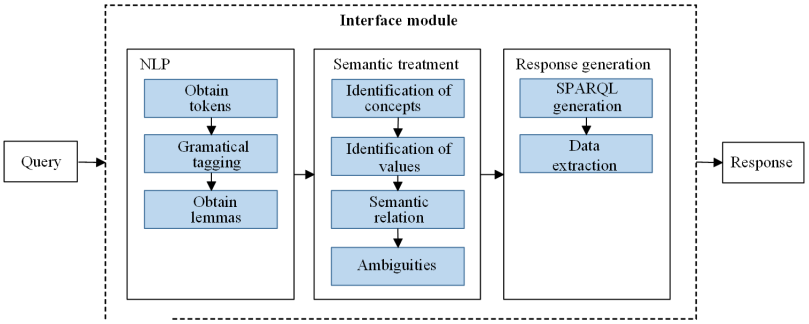


Fig. 5. Interface module operation.

tokens. We achieve this phase by using Freeling [19], a suite of tools for processing natural language which supports different languages, including English and Spanish. Supposing the user introduces the query "*What is the population density of Wyoming?*", the result of this phase is in Table 1, as in [22].

Table 1. Example of natural language processing.

Token	Lemma	PoS	Meaning PoS
What	what	WP	type= interrogative pos= verb
is	be	VBZ	pos= pronoun vform= personal person= 3
the	the	DT	pos= determiner
population	population	NN	pos= noun type= common num= singular
density	density	NN	pos= noun type= common num= singular
of	of	IN	pos=preposition
Wyoming	Wyoming	NP	pos= noun type= proper
?	?	Fit	pos= punctuation type= question mark punctenclose= close

The second step achieved by the IM is a semantic treatment. In this phase, each token, tagged grammatically as a noun or adjective, becomes analyzed. Analysis means to identify if any token maps to a concept, namely, to an individual of the class *CToken* from the ontology generated by the KGM (as it was in the world *population* example at the end of Section 3.1). That gets done through an exact matching of characters. Through the model presented in Figure 3, the IM identifies if the mapping refers to a class, text property, or data property. In case any of these tokens did not map to some element

in the ontology, it searches in the user's ontologies to identify if it is a value. In the previous example, *Wyoming* is a value, since the IM identifies that an instantiated individual from the class *State*. Correctly identifying the mappings and values is essential to generate the SPARQL query. In Table 2, we present the result of this step.

Table 2. Example of identification of mappings and values.

Token	Lemma	PoS	Meaning PoS
What	what	WP	
is	be	VBZ	
the	the	DT	
population	population	NN	Token=http://www.uacj_nlp.com/schema/nlp. OWL#population Class=http://www.uacj_nlp.com/schema/nlp. OWL#CDatatypeProperty Object=http://www.uacj_nlp.com/schema/nlp. OWL#STATEPOPULATION Token=http://www.uacj_nlp.com/schema/nlp. OWL#population Class=http://www.uacj_nlp.com/schema/nlp. OWL#CDatatypeProperty Object=http://www.uacj_nlp.com/schema/nlp. OWL#STATEPOPDENSITY Token=http://www.uacj_nlp.com/schema/nlp. OWL#citypopulation Class=http://www.uacj_nlp.com/schema/nlp. OWL#CDatatypeProperty Object=http://www.uacj_nlp.com/schema/nlp. OWL#CITYPOPULATION
density	density	NN	Token=http://www.uacj_nlp.com/schema/nlp. OWL#density Class=http://www.uacj_nlp.com/schema/nlp. OWL#CDatatypeProperty Object=http://www.uacj_nlp.com/schema/nlp. OWL#STATEPOPDENSITY Token=http://www.uacj_nlp.com/schema/nlp. OWL#population Class=http://www.uacj_nlp.com/schema/nlp. OWL#CDatatypeProperty Object=http://www.uacj_nlp.com/schema/nlp. OWL#in order toTo
of	of	IN	
Wyoming	Wyoming	NP	Individual=http://www.mooney.net/ geo#wyoming Class=http://www.mooney.net/geo#State Label=wyoming@en Individual=http://www.mooney.net /geo#wyomingMi Class=http://www.mooney.net/geo#City Label=wyoming@en
?	?	Fit	

Subsequently, it aims to identify if there is some relationship between any of the tokens. It determines that if, between its mapping, there exists some in common, besides following some *a priori* defined rules. In case of being affirmative, it combines both tokens into a composite one, keeping only the matching or related mappings and discarding the rest of them. The objective is to simplify the generation process of the SPARQL query. As we see in Table 2, the tokens *population* and *density* have in common the STATEPOPDENSITY mapping, which is why the IM combines them into a single token.

Then, the IM processes the identified values. If the latter belong to more than one class, it could cause an *ambiguity issue*. In such a case, *Wyoming* belongs to two of them, *State* (*wyoming*) and *City* (*wyoming*), so it tries to solve the ambiguity identifying the classes that match those of the rest of the queries' elements.

Since the IM determined, on the grounds of Figure 3, that STATEPOPDENSITY is a data property that belongs to STATE, it discards the class *City*. In the case the IM cannot solve the ambiguity, it shows a clarification dialog to the user so they can decide on it. In Table 3, we present the results of such a semantic treatment phase.

The third step is to generate the SPARQL query with the identified elements after the semantic treatment phase, shown in Table 3, and to use such query to extract, from the user's ontology, the requested information. To generate the

Table 3. Example of the semantic treatment phase.

Token	Lemma	PoS	Meaning PoS
What	what	WP	
is	be	VBZ	
the	the	DT	
population density	population density	NN	Token=http://www.uacj_nlp.com/schema/nlp. OWL#population Class=http://www.uacj_nlp.com/schema/nlp. OWL#CDatatypeProperty Object=http://www.uacj_nlp.com/schema/nlp. OWL#STATEPOPDENSITY
of	of	IN	
Wyoming	Wyoming	NP	Individual=http://www.mooney.net/ geo#wyoming Class=http://www.mooney.net/geo#State Label=wyoming@en
?	?	Fit	

SPARQL query, it follows a series of a priori defined rules. For example, based on Figure 3 and Table 3, the IM identified that STATEPOPDENSITY is a data property related to the STATE class. Such elements are defined in the user's ontology by the IM as <http://www.mooney.net/geo#statePopDensity> and as <http://www.mooney.net/geo#State>. Also, *Wyoming* is an individual of the class *State*. For that reason, the IM generated the following SPARQL query, where the variable that forms part of the *Select* clause comes up out of the combined token *population density*, which becomes transformed into *?population_density*. In Figure 6, we show results given by the IM using the generated SPARQL query.

```
Select ?population_density
Where {
        <http://www.mooney.net/geo#statePopDensity>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
        <http://www.w3.org/2002/07/owl#DatatypeProperty>.
        <http://www.mooney.net/geo#wyoming>
        <http://www.mooney.net/geo#statePopDensity>
        ?population_density.

        <http://www.mooney.net/geo#wyoming>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
        <http://www.mooney.net/geo#State>
}
```

population_density
0.20830059

Fig. 6. Result of the query "*What is the population density of Wyoming*" using the ontology Geography.owl.

4 Implementation

In this section, we describe the implementation of the NLI proposed in this paper.

We describe the software components and hardware devices where they locate. The NLI is based on a client-server architecture, uses an Android app as

a graphical interface and an application server where the query processing gets done. The server works through Java Server Pages (JSP) [12] as a technology for its implementation, which is why we use Apache Tomcat [26], a Java-based application container.

The display diagram of Figure 7 shows the software components and hardware devices used. The system implements the following software components: An Android application, an event controller for inter-component interaction, a knowledge base, a natural language processing component, and another to generate and execute SPARQL queries on the knowledge base.

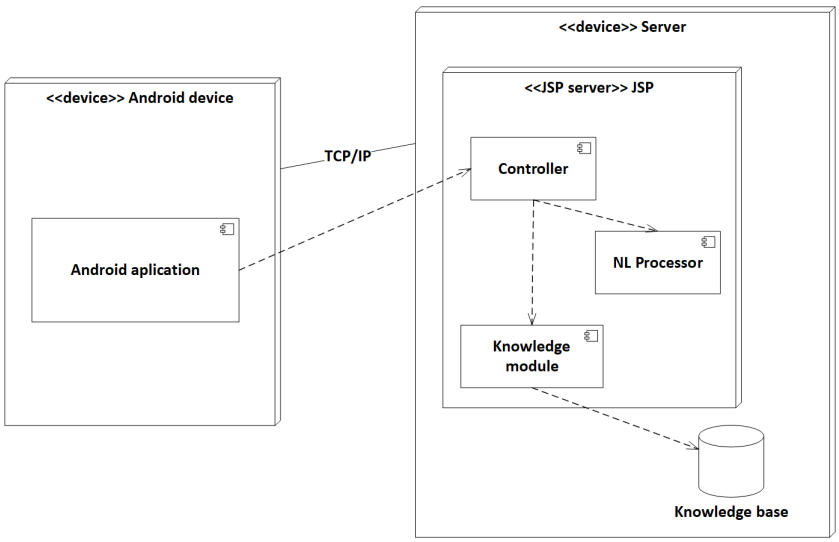


Fig. 7. Display diagram.

The general process for every input query is according to the following steps:

1. The Android app receives the query in a voice phrase as input, converts it into text, and sends it to the controller for processing.
2. The controller receives the query and sends it to the NL processor.
3. The NL processor receives the text query, and out of it, it obtains the tokens, grammatical tags, and lemmas. Finally, it sends the controller a set of all of them.
4. Once the controller receives the set of tokens, tags, and lemmas, it sends it to the knowledge module.
5. The knowledge module is in charge of generating SPARQL queries and executing them on the ontology to get a response. Lastly, it sends the response back to the controller.
6. The controller sends a response to the Android application to show it to the user.

For the implementation of each component, we used several development tools. We implemented the Android app with the Android Studio IDE, using native libraries to convert voice queries to text. One can run it in devices that use Android as an operating system.

The controller is JSP technology [13] implementation. Receives and makes requests to other components to manage all of the system's processes, from receiving the query to sending the answer to the Android application.

The NL processor is implemented in the Java language.After reviewing several natural language processing tools like the open-source library Freeling [19], the library Apache OpenNLP[9], and the Stanford CoreNLP [17] tools, we chose Freeling to implement this component since, unlike the others, it has better support for the English and Spanish languages.

The knowledge base is in ontology form, and in OWL [18] and RDF [21] files, to access it, we use the SPARQL query language. We implemented the knowledge module using the Java programming language and Apache Jena, which is a Java framework for querying knowledge bases stored in RDF and OWL structures and allows managing the ontology at runtime.

The NLI allows selecting the previously created ontology it is desired to query. Using the framework Jena, the NLI accesses the ontology and analyzes its structure to reply to the user's natural language queries subsequently.

5 Conclusions

With the upcoming of Semantic Web, the use of ontologies has taken importance, since they intended to use as a knowledge representation medium to facilitate the interoperability of information on the Web. The NLI2O are adequate tools to access knowledge stored in the ontologies. Its development could facilitate user resource localization, communication between informatics applications, information lookup, disambiguation, and other activities on the Web.

In this work, we propose the architecture of an NLI that uses knowledge bases on ontologies, which receives voice queries from an Android device. Then, it converts them to text, applies natural-language and knowledge-representation-processing techniques to generate the corresponding SPARQL queries. Finally, it extracts information from the ontology.

Even though the modeling and semantic processing by the NLI has allowed answering queries formulated in natural language by users, some areas of opportunity were detected.

As future works, we proposed improving the knowledge representation system, as well as the NLI semantic processing and F-logic. We also suggest upgrading the natural language processing, so the NLI is capable of answering queries that make comparisons, negotiations, automatic disambiguation, dates, and numbers. Subsequently, we propose evaluating the performance of the NLI against FREyA using the ontology Geography.owl and its respective corpus of queries called Geoquery 250 and Geoquery 880.

[9] Refer to `https://opennlp.apache.org`.

As future long-term works, we propose addressing complex problems related to natural language features. Some of them are anaphora, and intersentential ellipsis, to mention a few. The anaphora is when we refer to a previously mentioned entity, and its resolution is essential to answer questions that relate to the same object in different forms. The ellipsis occurs when we omit one or more words in a sentence and are understood by having been mentioned earlier.

References

1. Aduna, J.B., Aduna, A.K.: SeRQL: A second generation RDF query language. SWAD-Europe Workshop on Semantic Web Storage and Retrieval (2003)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (May 2001)
3. Bernstein, A., Kaufmann, E., Kaiser, C., Kiefer, C.: Ginseng: A guided input natural language search engine for querying ontologies. Department of Informatics, University of Zurich, Switzerland (2006)
4. Cimiano, P.: ORAKEL: A natural language interface to an F-Logic knowledge base. In: Meziane, F., Métais, E. (eds.) Natural Language Processing and Information Systems. pp. 401–406. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
5. Cimiano, P., Haase, P., Heizmann, J.: Porting natural language interfaces between domains: An experimental user study with the ORAKEL system. International Conference on Intelligent User Interfaces, Proceedings IUI (2008)
6. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: An architecture for development of robust HLT applications. In: 40th Annual Meeting on ACL. pp. 168–175. ACL '02, Association for Computational Linguistics, USA (2002)
7. Damljanovic, D.: Natural Language Interfaces to Conceptual Models. Ph.D. thesis, The University of Sheffield (2011), `http://etheses.whiterose.ac.uk/1630/2/Damljanovic{%}2C{_}Danica.pdf`
8. Damljanovic, D., Agatonovic, M., Cunningham, H.: Identification of the question focus: Combining syntactic analysis and ontology-based lookup through the user interaction. 7th LREC (May 2010)
9. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) The Semantic Web: Research and Applications. pp. 106–120. Springer, Berlin, Heidelberg (2010)
10. Damljanovic, D., Agatonovic, M., Cunningham, H.: FREyA: An interactive way of querying linked data using natural language. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) The Semantic Web: ESWC 2011 Workshops. pp. 125–138. Springer, Berlin, Heidelberg (2012)
11. Guarino, N., Oberle, D., Staab, S.: What is an Ontology? In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 0–17 (2009)
12. Hunt, J.: Java Server Pages, pp. 361–370. Springer, London (2002)
13. Jayson Falkner, K.W.J.: Servlets and JavaServer Pages^{TM}: The J2EE^{TM} Technology Web Tier. Addison-Wesley Professional (2003)
14. Kaufmann, E., Bernstein, A.: How useful are natural language interfaces to the semantic web for casual end-users? In: Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) The Semantic Web. pp. 281–294. Springer, Berlin, Heidelberg (2007)

15. Kifer, M., Lausen, G.: F-Logic: A higher-order language for reasoning about objects, inheritance, and scheme. In: Proceedings of the 1989 ACM SIGMOD. pp. 134–146. '89, ACM, New York, NY, USA (1989)
16. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the Semantic Web. In: Gómez-Pérez, A., Euzenat, J. (eds.) The Semantic Web: Research and Applications. pp. 546–562. Springer, Berlin, Heidelberg (2005)
17. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL System Demonstrations. pp. 55–60 (2014)
18. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. World Wide Web Consortium (February 2004)
19. Padró, L.: Analizadores multilingües en FreeLing. Linguamatica 3(2), 13–20 (December 2011)
20. Prud'hommeaux, E., Seaborne, A., W3C, Laboratories, H.P., Bristol: SPARQL Query Language for RDF. World Wide Web Consortium (January 2008)
21. Schreiber, G., Amsterdam, V.U., Raimond, I., BBC: RDF 1.1 Primer. World Wide Web Consortium (February 2014), `https://www.w3.org/TR/rdf11-primer/`
22. Solís, A., Florencia, R., Acosta, J., López, F.: Interfaz de lenguaje natural para deducción de información almacenada en ontologías . Research in Computing Science 147(6), 189–205 (2019)
23. Tablan, V., Damljanovic, D., Bontcheva, K.: A Natural Language Query Interface to Structured Information. In: The Semantic Web: Research and Applications: 5th ESWC, Tenerife, Canary Islands, Spain, June 1–5. Springer (2008)
24. Tang, L.R., Mooney, R.J.: Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing, pp. 466–477. Machine Learning: ECML 2001, Springer Berlin Heidelberg (2001)
25. Tudorache, T., Nyulas, C., Noy, N.F., Musen, M.A.: Webprotégé: A collaborative ontology editor and knowledge acquisition tool for the web. Semantic Web 4(1), 89–99 (2011), `10.3233/sw-2012-0057`
26. Vukotic, A., Goodwill, J.: Introduction to Apache Tomcat 7, pp. 1–15. Apress, Berkeley, CA (2011)

ISSN 1870-4069

Diseño industrial e inteligencia artificial: Proyección del diseño industrial a 100 años de su creación como disciplina y sus aplicaciones con IA hoy

Sandra Rodríguez-Mondragón, Oscar Herrera-Alcántara,
Luis Jorge Soto-Walls, Manuel Martín Clavé-Almeida

Universidad Autónoma Metropolitana,
Unidad Azcapotzalco,
México

{sandra.rgz.mondragon, luissotowalls,
mclavealmeida}@gmail.com, oha@correo.azc.uam.mx

Resumen. La definición de Diseño industrial como lo conocemos actualmente surge en 1919 junto con la Bauhaus, y la presente investigación muestra un análisis histórico de la forma en que se desarrollan diversas definiciones y concepciones de la disciplina como tal, en ella se busca esclarecer el concepto desde el punto de vista histórico, pasando por la definición actual y haciendo un planteamiento de su proyección incursionando en temas de inteligencia artificial. Esta propuesta de proyección del Diseño industrial (DI) se plantea por medio de aplicar técnicas de Inteligencia Artificial (IA), así el desarrollo de productos por medio del uso de algoritmos evolutivos para la optimización de simulaciones. Este método busca perfeccionar sistemas que ya cuentan con cierta utilidad para llegar a un desarrollo máximo de utilidad por medio de IA.

Palabras clave: Diseño industrial, innovación, tecnología, algoritmos evolutivos, inteligencia artificial, optimización.

Industrial Design and Artificial Intelligence: Projection of Industrial Design after 100 Years of its Creation as a Discipline and its Applications in Modern AI

Abstract. The definition of Industrial Design as we know it today arises in 1919 together with the Bauhaus, and this research shows a historical analysis of the way in which various definitions and conceptions of the discipline are developed as such, in it we seek to clarify the concept from the historical point of view, going through the current definition and making an approach to its projection by entering into topics of artificial intelligence. This proposal for the projection of Industrial Design (DI) is proposed through the application of Artificial Intelligence (AI) techniques, as well as the development of products through the

use of evolutionary algorithms for the optimization of simulations. This method seeks to improve systems that already have a certain utility to reach a maximum development of utility through AI.

Keywords: Industrial design, innovation, technology, evolutionary algorithms, artificial intelligence, optimization.

1. Introducción

El termino diseñador industrial aparece por primera vez en 1907 para describir el trabajo realizado por el arquitecto Peter Behrems dentro de la empresa *Allgemaine Elektricitäts-Gesellschaft* (AEG) como artista consultor; él es considerado el padre del Diseño Industrial en Alemania y además el creador de la identidad corporativa.

La definición de Diseño Industrial como lo conocemos actualmente surge junto con la Bauhaus (Weimar, Dessau y Berlín, 1919-1935) [1]; que fue la primera escuela en incluir la especialidad de Diseño Industrial como una profesión reconocida por el estado.

Así el manifiesto de la Bauhaus cita a la disciplina del Diseño Industrial como se muestra a continuación:

> *"La recuperación de los métodos artesanales en la actividad constructiva, elevar la potencia artesana al mismo nivel que las Bellas Artes e intentar comercializar los productos que, integrados en la producción industrial, se convertirían en objetos de consumo asequibles para el gran público" [2].*

Actualmente, la definición de Diseño Industrial de la Organización Mundial del Diseño (*World Design Organization, WDO*) en 2015 es la siguiente:

> *El diseño industrial es un proceso estratégico de resolución de problemas que impulsa la innovación, genera éxito empresarial y conduce a una mejor calidad de vida a través de productos, sistemas, servicios y experiencias innovadores. El diseño industrial cierra la brecha entre lo que es y lo que es posible. Es una profesión transdisciplinaria que aprovecha la creatividad para resolver problemas y crear soluciones con la intención de mejorar un producto, sistema, servicio, experiencia o negocio. En esencia, el diseño industrial ofrece una forma más optimista de mirar el futuro al reformular los problemas como oportunidades. Vincula la innovación, la tecnología, la investigación, los negocios y los clientes para proporcionar un nuevo valor y una ventaja competitiva en las esferas económica, social y ambiental [3].*

1.1. Definiciones del diseñador industrial en algunas instituciones educativas

En el contexto internacional existe un gran número de instituciones que tienen programas institucionales de Diseño Industrial (DI). En México, de acuerdo con Moreno [4], según el Consejo Mexicano para la Acreditación de Programas de Diseño, A.C. (COMAPROD) [5], existen 22 programas de licenciatura en Diseño Industrial en México acreditados por este organismo, dónde 9 son instituciones públicas y el 13 son privadas. Sin embargo, consultando otros organismos se observan hasta 46 universidades en el país que imparten la carrera de diseño industrial. Para propósitos de

este artículo, es relevante mencionar algunas de las instituciones que se presentan a continuación, en particular en la Universidad Autónoma Metropolitana (UAM) [6 y 7], la Universidad Nacional Autónoma de México (UNAM) [8] y la Universidad Autónoma del Estado de México (UAEM) [9] ya que cuentan con un número considerable de egresados de la carrera de Diseño Industrial. Aquí, se comenta sobre los objetivos de los planes de estudio, dado que cada una de estas instituciones cuenta con un programa de la carrera de Diseño Industrial y han desarrollado su propio concepto de disciplina mismo que expresan en sus planes y programas:

La UAM, Unidad Azcapotzalco, en el plan de estudios de la carrera en Diseño Industrial desde 2005, tiene por objetivo:

> *Formar un profesional integrado, con una conciencia crítica de su actividad disciplinaria y de la relación existente entre éste y la sociedad; que conozca racionalmente, mediante su proceso de diseño de productos, el compromiso, el enfoque, las funciones, aportaciones y la conciencia con la cual se diseña y realiza un producto socialmente útil [10].*

La UAM, Unidad Xochimilco, por su parte tiene como objetivo general de la carrera en Diseño industrial:

> *Formar diseñadores industriales altamente calificados, capaces de incorporarse al desarrollo industrial y científico-tecnológico a través de la reflexión crítica; de la generación de alternativas de diseño de objetos-bienes materiales; de la estética; de alternativas para el cambio tecnológico; de los mercados nacionales e internacionales y de la protección del medio ambiente. Asimismo, deberá ser capaz de coadyuvar a la resolución de necesidades mayoritarias y al desarrollo armónico, plural, justo y solidario de la sociedad mexicana [11].*

La UNAM, realiza la siguiente descripción del profesional en Diseño Industrial:

> *El Diseñador Industrial es el profesional que genera objetos-producto de fabricación industrial, que con su preparación y conocimientos contribuye al mejoramiento de la calidad de vida de las personas.*
>
> *Participa en la definición de los aspectos estéticos, funcionales y productivos del objeto e incrementa su valor.*
>
> *Desarrolla su actividad en tres áreas principales: dentro de una industria o empresa; llevando a cabo actividades de consultoría de manera independiente o creando su despacho, o bien; como diseñador emprendedor estableciendo una empresa.*
>
> *Asimismo, es responsable de: * Crear nuevas propuestas y soluciones de productos acordes con los requerimientos de un mercado globalizado que demanda productos mejor planeados y diseñados, con el objeto de lograr el óptimo aprovechamiento de la infraestructura industrial.*
>
> ** Proponer innovaciones e incluir tendencias en los procesos de manufactura, cuya originalidad les permita competir con productos importados [12].*

La UAEM, realiza la siguiente descripción del profesional de Diseño Industrial:

> *Un diseñador industrial propone y soluciona problemáticas del día a día, sociales y del trabajo; genera infinidad de propuestas de innovación en el área de la ergonomía. […]*

> *El diseño industrial resuelve necesidades, soluciona problemas, embellece el espacio. Un diseñador industrial es el profesionista capaz de dejar un legado en la cultura material de las diferentes sociedades del mundo, considerando los entornos naturales y artificiales. […].*
>
> *Un diseñador industrial es un visionario del futuro. Con la ayuda de las nuevas tecnologías, tu capacidad creativa no tendrá límites. El campo laboral es amplio y variado, puedes ofrecer tus servicios en departamentos de planeación y desarrollo de nuevos productos en diversas industrias, así como en las áreas de manufactura, producción, proyectos, investigación, etc. [13].*

Como se puede ver, todas coinciden en que dicha disciplina tiene un fin primario de diseño de objetos y productos industriales, con una conciencia crítica y reflexiva que sea un aporte en beneficio de la sociedad a nivel global. En dicho contexto se parte de un ambiente tecnológico que implica la capacidad creativa en la resolución de problemas por medio de productos. Pero, pese a que las definiciones implican la profesión como una disciplina creativa e innovadora, en todas ellas no se aprecia una clara inclusión de Inteligencia Artificial (IA), sin embargo la IA ha sido adoptada inconscientemente en la sociedad actual en el mundo y en el Diseño Industrial no es la excepción, sólo que no se ha tomado conciencia de su integración, por ejemplo: en aeronáutica se utiliza optimización evolutiva para el diseño de estructuras robustas, en aplicaciones médicas a nivel de prótesis personalizadas con un efecto en usabilidad eficiente y convivencia humano-computadora, sistemas de geolocalización para el desarrollo de vehículos autónomos, teléfonos inteligentes, entre otras aplicaciones. Este trabajo pone de manifiesto la importancia de hacerlo explícito en los planes y programas y en todo el ámbito del DI.

1.2. Inteligencia artificial

Parte de los alcances de la Inteligencia Artificial (IA) y algunas de sus características son que [14]:

- Es un área de la ciencia bastante nueva (1956)
- Su objetivo son las capacidades que consideramos Inteligentes
- Las aproximaciones siguen diferentes puntos de vista
- Sus influencias son diversas (Filosofía, Matemática, Psicología, Biología)
- Involucra una gran cantidad de áreas tanto genéricas (aprendizaje, percepción, resolución de problemas) como específicas (jugar al ajedrez, *diagnosticar enfermedades, conducir automóviles, etc.).*

A finales de siglo XX, de acuerdo con Berzal, la IA tiene cuatro campos de acción [15]:

a) *Sistemas que actúan como humanos, "El estudio de cómo hacer computadoras que hagan cosas que, de momento, la gente hace mejor" (Rich y Knight, 1991).*

b) *Sistemas que piensan como humanos, "El esfuerzo por hacer las computadoras pensar…máquinas con mentes en el sentido amplio y literal" (Haugeland, 1985).*

Tabla 1. Áreas de trabajo de IA.

Básicas	Específicas
Representación del conocimiento	Planificación de tareas Tratamiento del Lenguaje Natural Razonamiento Automático
Resolución de problemas, búsqueda	Sistemas Basados en el Conocimiento Percepción Aprendizaje Automático Agentes autónomos

c) *Sistemas que actúan racionalmente. "Un campo de estudio que busca explicar y emular el comportamiento inteligente en términos de procesos computacionales" (Schalkpff, 1990).*

d) *Sistemas que piensan racionalmente. "El estudio de las facultades mentales a través del estudio de modelos computacionales" (Charniak y Mc Dermott, 1985)*.

Ahora bien, para continuar con la identificación los sistemas de IA, algunas áreas de trabajo se presentan en la Tabla 1.

Como se puede ver en la Tabla 1, la IA tiene muchas vertientes y prácticamente cualquier área de conocimiento puede incluir IA, en el Diseño Industrial aún no se indica como incluir la IA, en contraste, la Industria 4.0 no existiría sin el concepto de IA, sino que ésta acepta y adopta como pieza fundamental las técnicas de IA para el desarrollo de sistemas de producción, de esta forma ¿cuál sería el nombre o la versión del Diseño Industrial dónde se incluya la IA? Puede ser DIIA (Diseño industrial e Inteligencia Artificial), tal como en su momento se desarrolló el Diseño con el uso de dibujo asistido por computadora que dio lugar al CAD (Diseño Aasistido por Computadora).

2. Diseño industrial e inteligencia artificial

En síntesis, el Diseño Industrial se basa en el desarrollo de productos con la aplicación de métodos creativos y el uso de tecnologías, por su parte, la Inteligencia artificial, se basa en los procesos de conocimiento y resolución de problemas realizados por el ser humano; así, la interrelación de estas disciplinas radica en resolver problemas de diseño con métodos innovadores que sustituyan algunas funciones del ser humano por medio de tecnologías de IA, una de las principales herramientas que han facilitado los procesos de diseño actuales, son el uso y análisis en *big data*.[1]

[1] Big data es un término que describe el gran volumen de datos – estructurados y no estructurados – que inundan una empresa todos los días, https://www.sas.com/es_mx/insights/big-data/what-is-big-data.html, ultimo acceso 05/02/20

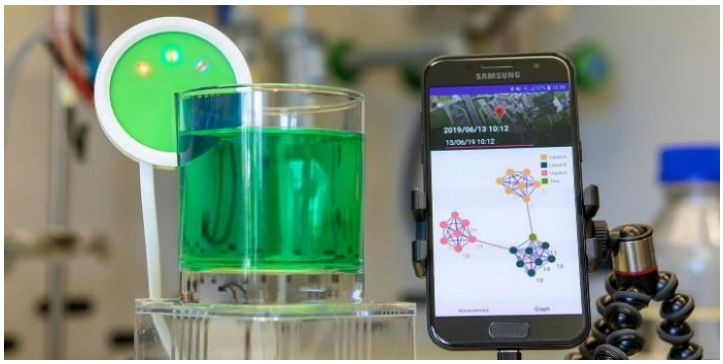


Fig. 1. Lengua electrónica (*IBM Research en Suiza) [16]*.

Fig. 2. Robots aspiradores Roomba® [17].

2.1. Ejemplos de diseño resueltos por medio de IA

Lengua electrónica

> *Investigadores de IBM Research en Suiza han creado un dispositivo que analiza y reconoce multitud de líquidos en menos de un minuto. Esa nueva tecnología se podrá aplicar, entre otros, en el sector biomédico y en análisis de la calidad del agua. Este dispositivo cuenta con un algoritmo de aprendizaje automático que le permitirá en un futuro, analizar muestras de orina de una persona y ayudar a obtener una evaluación de la huella dactilar metabólica (ver Figura 1) [16].*

Este ejemplo se basa en principios de diseño biónico o biomímesis, que resuelven problemas de diseño por medio de imitar las funciones de organismos vivos.

Aspirador robotizado

Estos aspiradores emplean un sistema de sensores inteligentes que guían al robot aspirador por toda la casa, a lo largo de las paredes, alrededor de los muebles y por debajo de sillones y camas para ayudar a limpiar los pisos en profundidad. Además, cuentan con tecnología que le permite identificar las áreas de mayor tránsito para optimizar su funcionamiento y el uso de energía, es programable para operar en horarios de poca actividad (ver Figura 2).

De acuerdo con Ma. Ángeles Domínguez [14], la IA está cambiando el trabajo de los diseñadores y tiene, algunas de las siguientes consecuencias:

- La IA acelerará el proceso de desarrollo del producto/servicio.
- A medida que el desarrollo se acelera, también los clientes tendrán nuevas expectativas sobre la línea de tiempo y el progreso.
- Las herramientas de diseño y los flujos de trabajo tendrán que ser más veloces para adaptarse a un entorno más dinámico.
- Las experiencias con la IA también deben diseñarse, tal vez como una nueva oportunidad de negocio.
- La IA puede reducir la necesidad de interfaz de usuario física/gráfica y su interacción.

2.2. Caso de aplicación de IA en diseño industrial

Diseño paramétrico

De acuerdo con Sergio Marrero y José Martínez, el diseño paramétrico consiste en:

> *Transformarla descripción primaria de un problema en una nueva descripción de cierto conjunto de objetos que, adecuadamente vinculados y puestos en funcionamiento, han de resolver el problema inicial [18].*

Ello sugiere realizar un diseño conceptual que define el problema de diseño y un diseño detallado que debe contar con un modelo matemático para describir el objeto por medio de ecuaciones y valores, esto permite identificar las características dimensionales y geométricas del objeto. Para implementar IA en este proceso se parte del modelo matemático del objeto y una matriz de confusión que opera empleando las variables dependientes e independientes del modelo.

Definición de variables

En el diseño, es preponderante la definición de formas primitivas o básicas del modelado e identificar éstas como variables primarias que regirán la forma de un objeto. Reconocer estas variables permite conocer las cualidades formales[2] del objeto y poder definir una proyección de mejora por medio de simulación.

Al definir con claridad las variables primarias, el proceso de simulación da la posibilidad de identificar la pertinencia de la forma del objeto, y ello ayuda a mejorar su función. Ahora bien, un sistema complejo de simulación consta de variables primarias y secundarias, dónde las variables secundarias o dependientes se modifican con base en las variables primarias.

A continuación, se muestra parte del proyecto "Sistema modular para conformación de escultura cerámica monumental" [19]. Así, en la Tabla 2, se observa la matriz de algunas de las variables dimensionales del objeto y en la Figura 3 la relación de variables con respecto a su forma. Cabe aclarar que en la Figura 3, se muestra el objeto en vista bidimensional y en el sistema de simulación ya se contemplan todas las variables en un ambiente tridimensional.

[2] Refiriéndonos a la forma geométrica del objeto.

Tabla 2. Matriz de variables.

Variables		Valor predicho	
Independientes	A	15	15
	B	60	60
Dependientes	a	60	=A
	b	15	=B
	c	60	=A
	d	15	=B

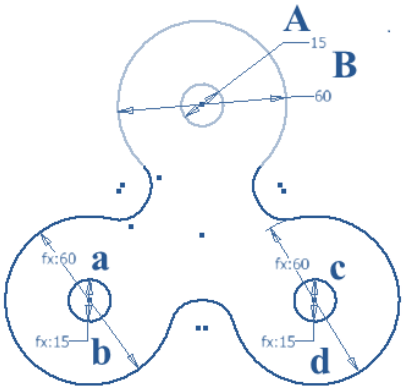


Fig. 3. Modelo b, aplicación de modelado paramétrico (Rodríguez, S. 2014) [19].

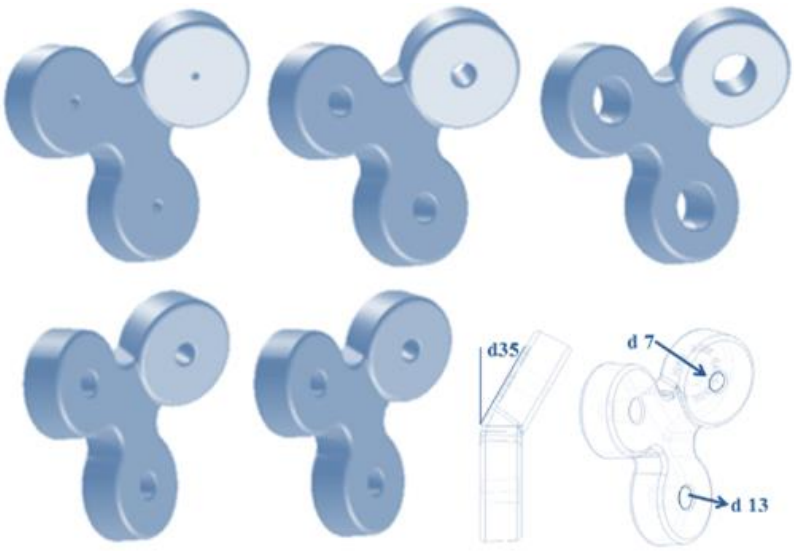


Fig. 4. Resultados (Rodríguez, 2014) [19].

Tabla 3. Variables físicas del material para el proceso de simulación del análisis de elementos finitos.

Variables	Definición
Material	Concreto, agregado expuesto
Densidad	2.40731 g/cm^3
Masa	587.682 gr
Área	34,443.7 mm^2
Volumen	244,124 mm^3
Centro de gravedad	X = 0.00000938643 mm Y = 7.70724 mm Z = 0.894215 mm

Tabla 4. Variables físicas del material para el proceso de simulación del análisis de elementos finitos en el sistema de módulos cerámicos.

Variables	Definición	
Densidad de masa	2.40731 g/cm^3	
Fuerza de rendimiento	2.41329 MPa	
Resistencia a la tracción	2.41329 MPa	
Estrés	El coeficiente de Poisson	0.167 *ul*
	Cortes de módulo	9.96144 *GPa*

Para continuar con el ejemplo en la figura 4, se muestran algunos de los resultados obtenidos al aplicar distintos valores a las variables primarias del objeto.

Para la optimización de este objeto, se somete la forma a diferentes esfuerzos en un sistema de simulación de Autodesk Inventor ®[3], para poder realizar la simulación, es necesario definir las cualidades físicas del material del objeto. En este caso se propuso un material con cualidades similares a cerámica de alta temperatura, dicho material es el concreto agregado expuesto. Las características físicas del objeto con material se presentan en la Tabla 3.

Resultados de la simulación

Para realizar el proceso de simulación, es necesario definir variables y restricciones en el sistema de simulación, lo que refiere al conjunto de objetos esta escultura. Las variables del sistema se presentan en la Tabla 4. En la primera simulación se definen características generales de densidad de masa, fuerza de rendimiento, resistencia a la tracción y estrés.

Dado que el sistema de simulación trabaja con el comportamiento de varios módulos sometidos a esfuerzos, también es importante definir variables de tipo de carga en condiciones operativas de gravedad (ver Tabla 5).

[3] Autodesk, Inc., es una compañía dedicada al software de diseño en 2D y 3D para las industrias de manufacturas, infraestructuras, construcción, medios y entretenimiento y datos transmitidos vía inalámbrica.

Fig. 5. Estrés.

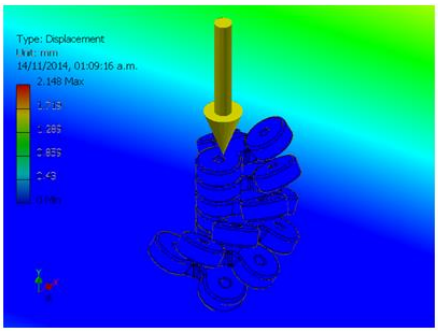


Fig. 6. Desplazamiento.

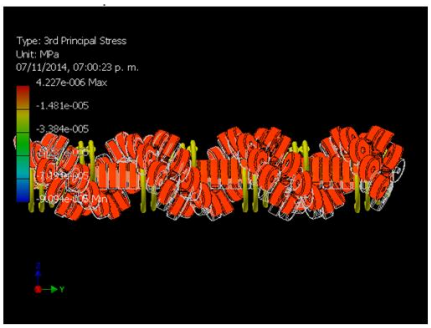


Fig. 7. Estrés.

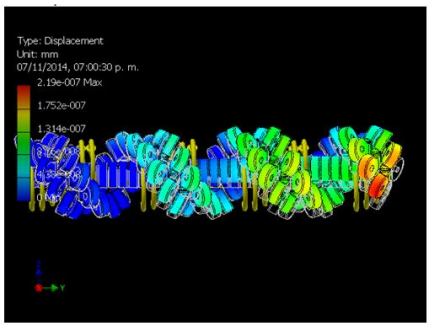


Fig. 8. Desplazamiento.

Tabla 5. Variables de tipo de carga en condiciones operativas con fuerza de gravedad.

Variables	Definición
Magnitud	9,810.000 mm/s^2
Vector X	0.000 mm/s^2
Vector Y	-9,810.000 mm/s^2
Vector Z	0.000 mm/s^2

Ahora bien, en las Figuras 5 a 8, podemos ver las gráficas de simulación que permiten validar el comportamiento del objeto al someterlo a ciertos esfuerzos empleando acomodo radial de los módulos. Las Figuras 5 y 6, muestran un modo de acomodo vertical conformado por 6 módulos; las Figuras 7 y 8, muestran un acomodo horizontal conformado por 35 módulos. En este ejemplo, dado que el objeto se proyectó con materiales cerámicos para producir una escultura modular, es importante evaluar el estrés y el desplazamiento, el primero que pueda inferir la posible fractura de los módulos escultóricos y el segundo una pérdida de equilibrio.
Finalmente, en la Figura 9, se muestran algunos de los resultados obtenidos después de dicho análisis.

En este caso, el diseño está evaluado por medio de simulación con Inventor profesional de AUTODESK®, para ser funcional como módulo de una escultura cerámica, sin embargo, los valores de la Tabla 2, fueron desarrollados con ajuste

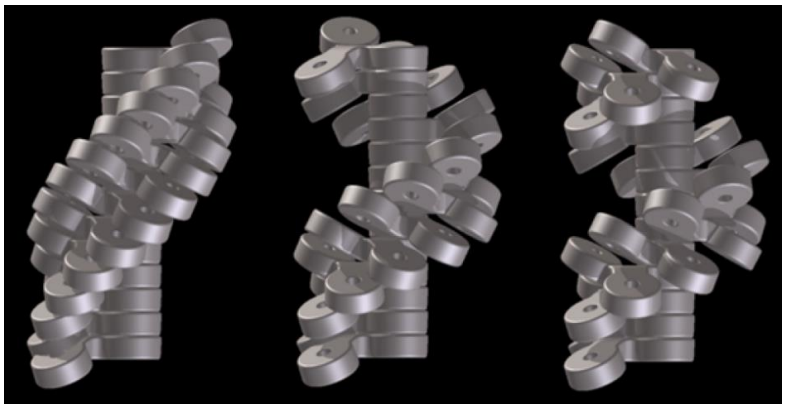


Fig. 9. Ensamble con acomodo radial [4].

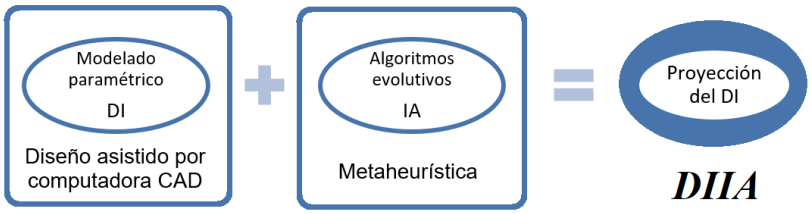


Fig. 10. Proyección del Diseño Industrial.

manual e incremento de módulos siempre que se mantenga la estabilidad en el proceso de simulación. Consideramos que estas restricciones pueden expresarse en forma de reglas. Para realizar el planteamiento del problema de diseño por medio de IA, es necesario definir con claridad, para este caso, el número de módulos, peso, estabilidad, superficie de contacto o exposición, etc., y la propuesta de mejora es por medio de alguna técnica de optimización.

2.3. Diseño industrial y algoritmos evolutivos

Ahora bien, como una propuesta de proyección del Diseño Industrial aplicando técnicas de IA, se sugiere trabajar el desarrollo de productos por medio del uso de algoritmos evolutivos [20] para la optimización de simulaciones. Este método busca perfeccionar sistemas que ya cuentan con cierta utilidad para llegar a un desarrollo máximo de utilidad por medio de IA.

Es importante trabajar estos métodos en combinación con el modelado paramétrico, así el manejo de variables matemáticas nos lleva a trabajar con métodos metaheurísticos [21], dado que los algoritmos evolutivos suelen ubicarse en este grupo de algoritmos (ver Figura 10).

Dado que los algoritmos evolutivos están basados en una población de datos, en el caso del modelado paramétrico, la matriz de variables dependientes e independientes pueden ser sometidas a estos algoritmos para llegar a la optimización del diseño. De acuerdo con lo anterior, el DI evoluciona al implementar IA, después del proceso de diseño. Esto permite no sólo realizar simulación para prever problemas de diseño, sino que, al llegar a la solución del problema de diseño, delegar a IA la optimización del diseño final del producto de DI.

Si bien este trabajo no presenta una solución a un caso particular del diseño paramétrico del caso de estudio en la Tabla 2, ya se tiene identificado el enfoque para que con apoyo de la IA se puede resolver y, más aún, deja patente la idea general de cómo el DI puede adoptar técnicas de IA que con métodos tradicionales o manuales serían muy difícil de abordar.

3. Trabajo futuro

Se espera realizar este tipo de ejecuciones con técnicas de optimización e ilustrar los resultados en contraste con los diseños actuales. Parte de la metodología que pueden adoptar los diseñadores industriales sugiere incluir todos o algunos de los siguientes aspectos:

1. Requerimientos de diseño cuantificables
2. Diseño paramétrico
3. Restricciones formales y funcionales
4. Identificación de funciones para optimización
5. Aplicar técnicas de optimización con posible apoyo de simuladores
6. Validación e interpretación de resultados

Se sugiere que las variables para trabajar con las funciones de optimización sean las secundarias que produce el simulador, en este caso de AUTODESK®[4], como son: densidad, masa, área, volumen, centro de gravedad, etc.

Esta metodología aplica a un diseño paramétrico bajo un enfoque de optimización, sin embargo, es muy importante mencionar que no es el único enfoque, porque la IA tiene muchas otras vertientes, por ejemplo, también se puede abordar el diseño basado en opiniones de los usuarios aplicando técnicas de procesamiento de lenguaje natural [22], por medio de encuestas de usabilidad [23], entre otras técnicas. También se pueden aplicar enfoques con cálculo simbólico [24] y manejo de reglas a través de ontologías [25], técnicas de visión por computadora [26], etc.

4. Conclusiones

Después de revisar estas definiciones y ejemplos, podemos decir que:

[4] Autodesk, Inc., es una compañía dedicada al software de diseño en 2D y 3D para las industrias de manufacturas, infraestructuras, construcción, medios y entretenimiento y datos transmitidos vía inalámbrica.

- El diseño industrial es la disciplina orientada a la creación y al desarrollo de los productos industriales. En esta actividad se aplica como aporte principal la creatividad y la innovación. Para esto, es necesario tener conocimiento de las tecnologías (actuales y anteriores), y aplicarlas en la resolución de problemas de diseño.
- Debido a que los productos industriales, tienen un alcance masivo, idealmente deben resolver necesidades sociales y ser ecológicamente responsables. Así, los aspectos relevantes a nuestro tiempo son de carácter tecnológico y medioambiental.
- La definición de la *Bauhause* es vigente, ya que el objetivo de llevar lo artesanal y lo artístico a los productos industriales es una de las metas del diseño industrial actualmente.
- Para implementar IA en objetos de diseño industrial, idealmente se debe trabajar diseño paramétrico que facilite el análisis del objeto por medio de valores numéricos que pueden ser sometidos a evaluación por medios de IA y que aporten soluciones de forma eficiente.
- La IA en el diseño industrial reduce tiempo en la toma de decisiones para el desarrollo de productos, dado que permite identificar problemas formales rápidamente desde la etapa de simulación y facilita la implementación de variables críticas en dicho proceso.
- Los algoritmos evolutivos, permiten la optimización de objetos de diseño cuando se trabaja simulación en el prototipado 3D, de esta forma no sólo se reducen o eliminan problemas de diseño estructural, dimensional, etc., sino que se puede llegar al mejor diseño posible con los parámetros determinados por el diseñador.

Referencias

1. La Bauhaus: Historia del arte. https://historia-arte.com/articulos/la-bauhaus (2020)
2. Gropius, W.: The New Architecture and the Bauhaus. Faber & Faber (1935)
3. WDO: The world design organization was formerly known as international council of societies of industrial design WDO. South Corea. http://wdo.org/about/definition/industrial-design-definition-history/ (2015)
4. Moreno, M.: Factores que permiten impulsar la participación en los mercados internacionales los productos desarrollados por el diseño industrial mexicano. Tesis de doctorado, Universidad Autónoma de Nuevo León (2019)
5. COMAPROD: https://www.comaprod.com/ (2020)
6. UAM: Resultados de la encuesta aplicada a egresados de la Licenciatura en Diseño Industrial, Unidad Azcapotzalco. http://www.uam.mx/egresados/Informes/P19LDI.pdf, (2013)
7. UAM: Resultados de la encuesta aplicada a egresados de la Licenciatura en Diseño Industrial, Unidad Xochimilco http://www.uam.mx/egresados/estudios/acreditacionlic/Unidad_Xochimilco/5_2011/Plan_87_Licenciatura_en_Diseno_Industrial_XOC_CAD_2011.pdf (2011)
8. UAM: 3er. Informe de actividades Facultad de arquitectura https://drive.google.com/file/d/1bT71ewTjxcx0E997ygkfUEIhWtTXinXE/view (2019)

9. UAEM: Estadística universitaria. http://web.uaemex.mx/universidatos/5121/5121/5121/DOP19/EBUAEM19.pdf (2020)
10. UAM: Plan de estudios de la carrera de diseño industrial de la UAM-Azcapotzalco. https://www.azc.uam.mx/docs/site/licenciaturas/19_3Industrial.pdf (2019)
11. UAM: Plan de estudios de la carrera de diseño industrial de la UAM-Xochimilco, http://www2.xoc.uam.mx/oferta-educativa/divisiones/cyad/licenciaturas/dindustrial/presentacion/ (2019)
12. UNAM: Plan de Estudios 2004 de la Licenciatura en Diseño Industrial. Recuperado de: https://arquitectura.unam.mx/uploads/8/1/1/0/8110907/parte1.pdf (2020)
13. UAEM: Descripción del profesional de Diseño industrial. Recuperado de: http://denms.uaemex.mx/exporientavirtual/wp-content/uploads/2014/12/Licenciatura_Diseno_Industrial.pdf (2020)
14. Domínguez, M.A.: Inteligencia Artificial y diseño: un mix cada vez más habitual. https://graffica.info/inteligencia-artificial-y-diseno-un-mix-habitual/ (2019)
15. Berzal, F.: Inteligencia Artificial. https://elvex.ugr.es/decsai/computational-intelligence/slides/A1%20AI.pdf (2016)
16. IBM: Lengua electrónica. https://www.agenciasinc.es/var/ezwebin_ site/storage/images/noticias/una-lengua-electronica-identifica-tipos-de-liquido-con-inteligencia-artificial/6542265-1-esl-MX/Una-lengua-electronica-identifica-tipos-de-liquido-con-inteligencia-artificial.jpg (2019)
17. Roomba: Robots aspiradores. https://www.irobot.mx/roomba (2019)
18. Marrero, S., Martínez, J.: Estrategia para el diseño paramétrico basado en modelos. Ingeniería Mecánica, 11(3), pp. 39–46 (2011)
19. Rodríguez, S.:Sistema Modular para la conformación de escultura cerámica monumental. Tesis de maestría, UAM Azcapotzalco (2014)
20. Holland, J.H.: Adaptation in natural and artificial systems. Ann Harbor: University of Michigan Press (1975)
21. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys, 35(3), pp. 268–308 (2003)
22. Goldberg, D.E.: Genetic algorithms in search optimization and machine learning, Addison-Wesley (1989)
23. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press (1996)
24. Fogel, D.B.: Evolutionary Computation. The Fossil Record. Selected Readings on the History of Evolutionary Algorithms (1998)
25. Sánchez, J.: Investigación y análisis de ontologías para el manejo de información en el World Wide Web. https://deim.urv.cat/~itaka/PFCs/JSanchez.pdf (2002)
26. Alegre, E., Pajares, G., Escalera, A.: Conceptos y métodos en visión por computador. Grupo de visión del CEA. https://intranet.ceautomatica.es/sites/default/files/upload/8/files/ConceptosyMetodosenVxC.pdf (2016)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx