EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 150 No. 7
July 2021

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
Olga Kolesnikova, ESCOM-IPN, Mexico
Rafael Guzmán, Univ. of Guanajuato, Mexico
Juan Manuel Torres Moreno, U. of Avignon, France

Editorial Coordination:
Griselda Franco Sánchez

RESEARCH IN COMPUTING SCIENCE, *Año 21, Volumen 150, No. 7, 09 de Julio de 2021, 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 Othón 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 Nacional del 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 Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 09 de Julio de 2021.*

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicacion. Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politecnico Nacional.

RESEARCH IN COMPUTING SCIENCE, *Year 21, Volume 150, No. 7, July 09, 2021, is a monthly publication edited by the National Polytechnic Institute through the Center for Computing Research. Av. Juan de Dios Bátiz S/N, Esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, C.P. 07738, Mexico City, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor in charge: Dr. Grigori Sidorov. Reservation of Exclusive Use Rights of Title No. 04-2019-082310242100-203. ISSN: pending, both granted by the National Copyright Institute. Responsible for the latest update of this issue: the Computer Research Center, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Last modified on July 09, 2021.*

The opinions expressed by the authors do not necessarily reflect the position of the publication's editor. The total or partial reproduction of the publication's contents and images is strictly prohibited without prior authorization from the National Polytechnic Institute.

Advances in Artificial Intelligence

Oscar Herrera-Alcántara (ed.)



Instituto Politécnico Nacional,
Centro de Investigación en Computación, México 2021

ISSN: en trámite

Copyright © Instituto Politécnico Nacional 2021
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

Table of Contents

Page

Explorando la variabilidad en el tamaño de la población de un algoritmo genético para resolver el problema del agente viajero 7
Laura Michelle Báez Villegas, Santiago Omar Caballero Morales

Aplicación de red neuronal artificial para estimar el impago de préstamos bancarios 19
Walter Jeremías López Flores

Gestión y presentación de información de una biblioteca por medio de XML 33
Oscar Eduardo González Ramos, Mireya Tovar Vidal

Hacia un semáforo de riesgo personal para sismos en la república mexicana 45
Rodrigo Vázquez Espino, Wulfrano Arturo Luna-Ramírez, Anasol Peña-Rios, Miguel Capllonch Juan

Propuesta de distribución y agrupamiento de Centros de Salud en el Estado de Aguascalientes basado en algoritmo K-Means modificado considerando la patología de Diabetes Mellitus 59
Pablo Rodríguez de León, María Dolores Torres Soto, Aurora Torres Soto

Análisis de un láser de fibra óptica dopada con Erbio y modulación de bombeo por medio de grafos de visibilidad 73
Jorge Luis. García-Rito, Jesús R. Sevilla-Escoboza

Prevención de riesgos de trabajo mediante la detección de emociones relacionadas con el estrés laboral ocupando técnicas de visión artificial para el cumplimiento de la NOM-35-STPS-2018 83
Efrén Uriel Aguilar Juárez, Brenda González Torres, Fernando Munguia Reyes

Amputación de datos de registros geofísicos con mecanismo de pérdida MCAR 95
César E. Santos-Vázquez, Leonardo Morales-Collado, Juana Canul-Reich, José Hernández-Torruco

DABATestors un algoritmo basado en CUDA para reducción del tiempo de obtención de testores 109
Daniel Alejandro Barajas Aranda, Aurora Torres Soto, María Dolores Torres Soto

Revisión del estado del arte en técnicas de minería de flujos de datos 119
María Yesenia Zavaleta-Sánchez, Edgard Iván Benítez-Guerrero

Modelo basado en agentes para exploración y optimización de trayectorias mediante cómputo Bio-Inspirado 133
Marco A. Jiménez-Limas, Emmanuel Salcedo-Carrillo, Fernanda Borjas, Amaury Salcedo-Carrillo

Aplicación móvil para el análisis de vocalizaciones del perro doméstico (canis familiaris) 147
Humberto Pérez-Espinosa, Yuvila M. Sanzón Rosas, Emanuel Aguilar Benitez

Separador industrial PET mediante técnicas de visión computacional 161
Irving Misael Caballero Aquino, Axel Adrián Casarrubias Calvillo, Rubén Daniel Vera León

Construcción de un modelo automático para la detección de armas en el transporte público 173
Eduardo Enrique Roldán-Juárez, Gabriela Ramírez-de-la-Rosa, Carlos Joel Rivero-Moreno

Comparación del rendimiento del algoritmo basada en forrajeo de bacterias con Matlab y Octave 187
Adrian García-López, Betania Hernández-Ocaña, Oscar Chávez-Bosquez, José Hernández-Torruco, Juana Canul-Reich

COMPOSAIN: Aplicación de un algoritmo genético para el diseño de contrapuntos musicales sin restricción de integridad de figuras 199
Natan Vilchis-Tavera, Adriana Lara

Estudio sobre sistemas tutores inteligentes basados en gamificación 213
Aldo Cisneros, Yasmín Hernández, Alicia Martínez, Javier Ortiz, Hugo Estrada

Análisis de rendimiento de particionamiento por medoides y medianas 227
María Beatriz Bernábe Loranca, Jonathan Ramos Flores, Rogelio González Velázquez, Gerardo Martínez Guzmán, Juan Carlos Duque Cardona

Enfoques y técnicas de análisis en estudios experimentales para neuromarketing, una revisión de la literatura 243
Pamela Stephanie Rivera Pérez, Leslie Imelda Callejas Dorantes, Alejandro del Rey Torres Rodriguez

Sistema de detección de intrusos en el CAN bus de automóviles utilizando una red neuronal multicapa 257

Juan Carlos Venegas-Segura, Ponciano Jorge Escamilla-Ambrosio, Alberto Jorge Rosales-Silva, Floriberto Ortiz-Rodríguez, Narciso Muñoz-Aguirre

Previsión de ventas basado en un modelo híbrido de pronóstico gris y un modelo de red neuronal artificial 271

Kennya Jeamileth Trochez Aguilar, Rodolfo Eleazar Pérez Loaiza, Perfecto Malaquías Quintero Flores

Modelado de un sistema de inferencia difusa aplicado al monitoreo de emisiones contaminantes de tráfico de flotillas de repartición de productos 285

Melina Ramos Sánchez

Modelo operador lineal en toma de decisiones: sensibilidad a ganancias y pérdidas 299

Laurent Avila-Chauvet, Diana Mejía Cruz

Generador de problemas de ecuaciones no lineales con gamificación 309

Hugo Pablo Leyva, Rafaela Blanca Silva López

Modelando y optimizando el problema de selección de portafolio de múltiples objetivos y parámetros difusos triangulares 323

Luis Mario Velasco Ocejo, Héctor Joaquín Fraire Huacuja, Fausto Antonio Balderas Jaramillo

Tratamiento de señales electroencefalográficas de imaginación motora con fines de clasificación 337

Alexis Edmundo Gallegos Acosta, María Dolores Torres Soto, Aurora Torres Soto, Eunice Esther Ponce de León Sentí

Identificación de potenciales “Hotspots” usando métodos de clasificación 351

Humberto Reyes San Pedro, Humberto Cervantes Maceda, Abel García Nájera

Plataforma digital para la seguridad escolar mediante el control de acceso automatizado basado en el reconocimiento facial 365

Cynthia B. Perez, Jesus R. Villavicencio, Jessica Beltran, Roberto Limon-Ulloa, Perla Duran, Samuel Torres, Bryan García

Recopilación y análisis de datos para la implementación de un sistema que ayude a la predicción de intenciones de robo en personas 379
Juan Carlos Sánchez Navarro, Blanca Estela Pedroza Méndez, José Federico Ramírez Cruz, Rubén Xochitiotzi Cuatecontzi, Abel Martínez Gorospe

Diseño y desarrollo de un robot de servicio con un control de velocidad PID 391
Rosario Baltazar, Martín Sánchez, Miguel Ángel Casillas, Liliana Sánchez, Arnulfo Alanis

Peligros y ventajas del diseño de instrumentos de medición auxiliado con estrategias de selección de atributos automáticas 405
Mario E. Marin, Julio C. Ponce, Ángel E. Muñoz, Alejandro Padilla

Máquina de estado líquido con reservorio de neuronas pulsantes a tiempo discreto para clasificación espacio-temporal 419
Christian Alba-Cisneros, Andrés Espinal, Manuel Ornelas-Rodríguez, Horacio Rostro-Gonzalez, Marco Sotelo-Figueroa, Jorge Soria-Alcaraz, Héctor Puga-Soberanes, Raúl Santiago-Montero

ISSN 1870-4069

Explorando la variabilidad en el tamaño de la población de un algoritmo genético para resolver el problema del agente viajero

Laura Michelle Báez Villegas, Santiago Omar Caballero Morales

Universidad Popular Autónoma del Estado de Puebla,
México

lauramichele.baez@upaep.edu.mx,
santiagoomar.caballero@upaep.mx

Resumen. El Problema del Agente Viajero (o Travelling Salesman Problem, TSP) es uno de los principales modelos de ruteo dentro del campo logístico. Su complejidad del tipo NP-duro hace que problemas de gran escala sean difíciles de resolver en tiempo razonable. Para ello resulta de mucha utilidad el diseño y la aplicación de algoritmos de búsqueda local o metaheurísticas. En este trabajo, el TSP se aborda mediante el diseño de un Algoritmo Genético que considera una población fija y una población variable como mecanismo para mejorar la convergencia hacia un óptimo local. Los resultados indican que, si bien una población variable puede hacer más rápida la ejecución del Algoritmo Genético y ayuda a evitar que el algoritmo permanezca en un óptimo local de manera prematura, su velocidad de convergencia disminuye con respecto al caso de una población fija (aunque esto involucre un mayor tiempo de ejecución). Esta observación puede ser de utilidad para el ajuste de los parámetros de estos algoritmos para el TSP.

Palabras clave: Problema del agente viajero, algoritmos genéticos, población variable.

Exploring Variability in Population Size of a Genetic Algorithm for Solving the Travelling Salesman Problem

Abstract. The Travelling Salesman Problem (TSP) is one of the main routing models within the logistics field. Its NP-hard complexity makes large-scale problems difficult to solve in a reasonable time. For this, the design and application of local search algorithms or metaheuristics is very useful. In this work, the TSP is approached by designing a Genetic Algorithm that considers a fixed population and a variable population as a mechanism to improve convergence towards a local optimum. The results indicate that, although a variable population can speed up the execution of the Genetic Algorithm and help prevent the algorithm from remaining in a local optimum prematurely, its speed of convergence decreases with respect to the case of a fixed population (although this involves a longer execution time). This observation may be useful for adjusting the parameters of these algorithms for the TSP.

Keywords: Travelling salesman problem, genetic algorithms, variable population.

1. Introducción

El Problema del Agente Viajero, o Travelling Salesman Problem (TSP) es uno de los problemas más importantes dentro de la logística y de la optimización. Este problema modela el escenario de un agente viajero que parte de su ciudad de origen, visita exactamente una vez cada ciudad de un conjunto asignado, y finalmente regresa a su ciudad de origen (punto de partida) [1].

Un recorrido con estas características se conoce como Circuito Hamiltoniano, Ruta o Tour. Por lo tanto, el objetivo del TSP consiste en encontrar el Circuito Hamiltoniano de menor costo o distancia total. La Figura 1 presenta un ejemplo de solución para el TSP con ocho ciudades [1]. Debido a lo anterior, la relevancia del TSP no solo se presenta en los aspectos de distribución, sino también en los traslados dentro de los almacenes y plantas de producción [1].

De igual manera, el TSP es la base de problemas más extensos como el Problema de Ruteo Vehicular o Vehicle Routing Problem (VRP) que puede entenderse como un TSP múltiple, en el cual se tienen restricciones de demanda y capacidad (véase Figura 2) [2, 7]. Si bien el TSP/VRP tienen amplia aplicación en la distribución, la solución de estos es difícil cuando se trata de redes de gran tamaño.

Esto es debido a su complejidad computacional que es del tipo NP-duro, lo cual hace que el tiempo de ejecución de un algoritmo de solución crezca exponencialmente conforme el tamaño del problema crezca linealmente [1, 5, 8]. ¡Por ejemplo, un TSP con 8 ciudades tiene un total de 8! = 40320 soluciones posibles. ¡En cambio, un problema con sólo una ciudad adicional tiene un total de 9! = 362880 soluciones posibles.

Por ello, las formulaciones matemáticas mediante programación lineal entera y algoritmos exactos como Branch & Bound pueden generar soluciones en tiempo razonable en redes con menos de 200 nodos, ciudades o ubicaciones [3]. Para obtener soluciones cercanas al óptimo dentro de un tiempo razonable se ha realizado investigación en el campo de los algoritmos de búsqueda local o metaheurísticas.

Para los problemas de ruteo, algoritmos como Búsqueda Tabú (BT) [4], Algoritmos Genéticos (AG) [5], Clarke & Wright (Ahorros) [6], entre otros. En el presente trabajo se considera a los AG para resolver redes del tipo TSP con más de 200 nodos. Si bien esta consideración ya se ha abordado en la literatura especializada, en estos se establecen estructuras fijas para el tamaño de la población, la cual es el medio principal de memoria de soluciones y que, dependiendo de su tamaño, puede hacer más lenta la operación del AG.

Es por esto que el AG propuesto considera una población de tamaño variable con el fin de hacer más rápido el AG, y depurar la población para evitar la convergencia hacia un óptimo local de manera prematura.

Los avances del presente trabajo se describen en las siguientes secciones: en la Sección 2 se presentan las bases de los AG mientras que en la Sección 3 se presentan los detalles del AG propuesto con población variable.

Los resultados se presentan en la Sección 4 y la discusión de estos, incluyendo el trabajo futuro, se presentan en la Sección 5.

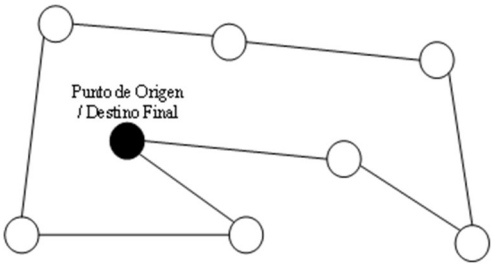


Fig. 1. Ejemplo de Solución para el TSP (basado en [1]). La ruta solución inicia y termina en un punto o ubicación, y cada ubicación es visitada sólo una vez.

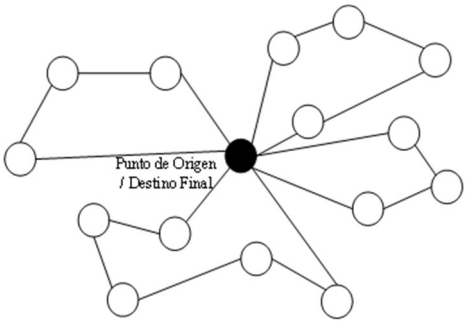


Fig. 2. Ejemplo de Solución para el VRP (basado en [2]). La solución consiste de un conjunto de rutas que inician y terminan en un punto o ubicación, y cada ubicación es visitada sólo una vez y por una sola ruta. Las demandas de estas ubicaciones no deben exceder la capacidad del vehículo que cubre la ruta asignada a éstas.

2. Algoritmos genéticos para problemas de ruteo

Los Algoritmos Genéticos o AG consisten de diversas técnicas de búsqueda que imitan el proceso evolutivo de selección natural o "sobrevivencia del ser más apto". Dichos algoritmos han sido útiles para encontrar soluciones a diversos problemas combinatorios complejos. Los pasos de un AG se describen en la Tabla 1.

Nótese que la idea de un AG es mejorar soluciones candidatas por medio de la simulación de mecanismos de evolución tales como la selección, cruzamiento y mutación. También, al ser algoritmos de búsqueda local, su desempeño depende de los mecanismos utilizados para explorar el espacio de solución como son los operadores específicos para cruzamiento y mutacion [8, 9]. Dentro de las diferentes variantes de estos operadores, el método de cruzamiento es el que más influye en diversificar la estructura de las soluciones.

Aquí es importante señalar que el cruzamiento está sujeto a las representaciones de cromosomas de las soluciones, las cuales pueden ser secuencias binarias, de números enteros, reales o secuenciales [10]. Para los problemas de ruteo en donde las soluciones consisten de secuencias de números enteros únicos (permutaciones), el método de cruzamiento conocido como Cruzamiento de Emparejamiento Parcial o Partially Matched Crossover (PMX) se ha utilizado ampliamente [10,11,12]. Otro aspecto es el

Tabla 1. Pasos generales de un AG.

Paso 0	Inicialización de la población: conjunto inicial de soluciones factibles (individuos) para mejorarse a lo largo del proceso del AG
Paso 1	Evaluación de cada individuo de la población para determinar su "elegibilidad" (por ejemplo, esto puede ser el valor de función objetivo o distancia total de la solución)
Paso 2	Proceso iterativo de búsqueda del AG (generaciones) que se repite hasta que se cumpla la condición de terminación o paro. a. Seleccionar padres (pares de soluciones de la población para generar (engendrar) otras soluciones (hijos). b. Reproducción de padres mediante operadores de cruzamiento y mutación. c. Evaluar "elegibilidad" de los individuos engendrados (hijos). d. Actualizar población con los mejores padres e hijos para la siguiente generación. e. Devolver el mejor individuo y mejores valores en población y promedio para evaluar convergencia. f. Evaluar si se cumple condición de terminación, de lo contrario, seguir con la próxima iteración (generación).
Paso 3	Regresar mejor individuo en la población final

tamaño de la población, la cual puede restringir o saturar la capacidad de estos operadores para diversificar soluciones. Ambas situaciones pueden dar como resultado "estancamientos" en soluciones locales de baja calidad, aunque el AG se ejecute durante un número extenso de generaciones.

Por ello, es necesario explorar alternativas o configuraciones en los mecanismos del GA para evitar este "estancamiento" en soluciones locales. Como propuesta de este trabajo, se considera el cambio dinámico en el tamaño de la población para crear una interrupción y reinicio de las soluciones candidatas para impulsar la búsqueda en otros espacios de solución y reducer el riesgo de "estancamiento". Los detalles técnicos del AG en función de los components de la Tabla 1 se describen en las siguientes secciones.

3. Algoritmo genético con población variable para el TSP

3.1. Paso 0

Se define inicialmente una población con P=100 individuos o soluciones para el TSP en donde el 50% son generadas de manera aleatoria y el 50% son generadas mediante un algoritmo voraz (Greedy Algorthm). Como ejemplo de una solución, se tiene el siguiente considerando 12 ubicaciones:

[1 – 9 – 6 – 5 – 10 – 2 – 4 – 8 – 12 – 3 – 11 – 7 – 1].

Nótese que esta solución implica una ruta que comienza en la ubicación "1", procede con la ubicación "9", y de manera secuencial continua con las ubicaciones "6", "5", "10", "2", "4", "8", "12", "3", "11" y "7" para finalmente regresar a la ubicación de origen "1", la cual en el context logístico y de distribución usualmente corresponde al centro de distribución, almacén, o centro de acopio. Este formato de solución

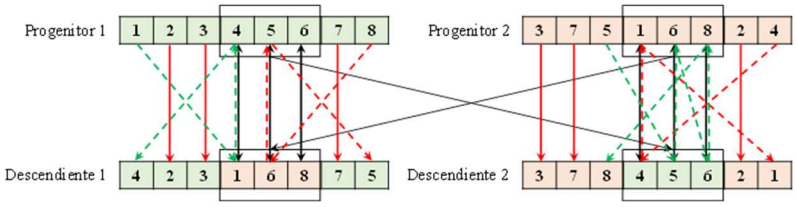


Fig. 3. Ejemplo de cruzamiento PMX (basado en [13]).

corresponde a la representación cromosómica del AG para la cual se deben seleccionar operadores de cruzamiento y mutación apropiados.

3.2. Paso 1

Una vez teniendo la población inicial se procede a evaluar la elegibilidad de cada individuo. En este caso, la elegibilidad depende de la distancia total asociada a la ruta. Para ello, se considera la suma de distancias Euclideanas entre cada ubicación dentro de la ruta.

Dentro del contexto de modelación matemática, la distancia total se puede expresar mediante la siguiente función:

$$C_T = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \, . \tag{1}$$

En donde n es el número total de nodos o ubicaciones a visitar en el problema, c_{ij} es la distancia o costo de viajar al nodo j desde i, (por ejemplo, la distancia Euclideana entre las ubicaciones i y j) y x_{ij} es la variable binaria de decisión que es igual a "1" si el camino entre i y j se considera dentro de la ruta solución al TSP, y es igual a "0" en caso contrario. Nótese que resolver este problema consiste en encontrar los valores de la variable de decisión x_{ij} que minimicen la función C_T. Esto es, resolver la función objetivo $min\ C_T$.

Al tener la población con estos valores de elegibilidad de cada individuo se hace un ordenamiento en base a éste ultimo de manera descendente. Con ello, los individuos (rutas) con menor distancia total se ubicarán al inicio de la población en tanto que aquellos con las mayores distancias (y, por lo tanto, menos elegibles para resolver el TSP) se ubicarán al final de la población.

3.3. Paso 2

a. **Selección:** Para comenzar la reproducción de individuos es necesario escoger un método de selección apropiado para las soluciones progenitoras. Para este propósito se considera el método de "torneo" en donde los individuos con mejor valor de elegibilidad son candidatos para reproducción. También, de manera aleatoria, una solución se elegirá de la parte inferior de la población para reproducción con una solución de la parte superior.

Tabla 2. Instancias de la base de datos TSPLIB usadas para pruebas.

n	instancia	*n*	instancia
51	eil51.tsp	198	d198.tsp
52	berlin52.tsp	200	kroA200.tsp
70	st70.tsp	200	kroB200.tsp
76	eil76.tsp	225	ts225.tsp
76	pr76.tsp	225	tsp225.tsp
99	rat99.tsp	226	pr226.tsp
100	kroA100.tsp	262	gil262.tsp
100	kroB100.tsp	264	pr264.tsp
100	kroC100.tsp	280	a280.tsp
100	kroD100.tsp	299	pr299.tsp
100	kroE100.tsp	318	lin318.tsp
100	rd100.tsp	400	rd400.tsp
105	lin105.tsp	417	fl417.tsp
107	pr107.tsp	439	pr439.tsp
124	pr124.tsp	442	pcb442.tsp
127	bier127.tsp	493	d493.tsp
130	ch130.tsp	574	u574.tsp
136	pr136.tsp	575	rat575.tsp
144	pr144.tsp	654	p654.tsp
150	ch150.tsp	657	d657.tsp
150	kroA150.tsp	724	u724.tsp
150	kroB150.tsp	783	rat783.tsp
152	pr152.tsp	1002	pr1002.tsp
159	u159.tsp	1060	u1060.tsp
195	rat195.tsp	1084	vm1084.tsp

b. Reproducción: Al tener la selección de pares de individuos para reproducción, la generación de nuevos individuos (soluciones descendientes o hijos) se obtiene mediante los operadores de cruzamiento y mutación.

Como operador de cruzamiento se considera el método PMX, el cual hace una combinación factible de fracciones de cada solución progenitora. Un ejemplo de cómo funciona este operador se presenta en la Figura 3.

En cuanto a los operadores de mutación, se consideran a los métodos de inversión de sub-secuencias entre dos puntos, e intercambio de dos puntos.

c. Evaluación de elegibilidad: Una vez obtenidas las soluciones descendientes el valor de elegibilidad de cada una se obtiene mediante la suma acumulada de las **distancias** Euclideanas entre cada ubicación dentro de la ruta equivalente.

d. Actualización de la población: Al tener todas las soluciones evaluadas, se hace una unión de las soluciones en la población con aquellas de los descendientes. Esta población total se ordena en orden descendente y se toman las mejores P soluciones para integrar la población para la siguiente generación. Nótese que esto va a ocasionar que, tanto el valor de la mejor solución como el del promedio de las soluciones en la población, se mejoren con respecto a la generación presente.

Inicialmente el AG se ejecuta durante X generaciones con una población fija de P soluciones. Al llegar a este punto se comienza la dinámica de cambio de tamaño de

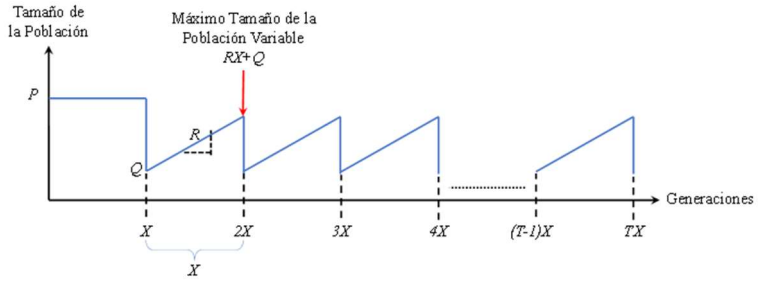


Fig. 4. Dinámica de cambio de tamaño de población para el AG.

población mediante un reinicio a Q soluciones, las cuales se incrementarán a una razón de R unidades por generación. Esta dinámica se ilustra en la Figura 4. Nótese que el proceso de reinicio del tamaño de la población a Q soluciones se realiza cada X generaciones y que, dependiendo de la razón de incremento en R unidades, se puede llegar a tener una población con un máximo de $R \times X + Q$ soluciones que puede ser mayor que la población inicial P.

e. **Cálculo de convergencia y registro de mejor solución:** Al final de cada generación se hace un registro de la mejor solución, la cual, de acuerdo al esquema de actualización de la población, corresponde a la primera solución en la misma. El valor de elegibilidad de esta solución también se registra para llevar un control de la convergencia del AG.

f. **Condición de terminación:** Como condición de terminación se define un número máximo de generaciones K. Para este caso, y para propósitos de evaluar el enfoque del cambio de población, se consideran K=1000 generaciones.

4. Resultados

El AG fue codificado en OCTAVE y fue ejecutado en una computadora Laptop Intel Core i7-5500U CPU a 2.40 GHz con 8GB RAM. La tasa de cruzamiento por PMX fue del 60% en tanto que la de mutación por intercambio e inversion fue del 30%. Las pruebas se realizaron con las instancias de la base de datos TSPLIB [14]. Los detalles de estas instancias se muestran en la Tabla 2. Como se puede observar, se consideraron 50 instancias en donde 25 instancias tienen de 51 a 195 ubicaciones, y 25 instancias tienen de 198 a 1084 ubicaciones.

Para estas instancias, la Figura 5 muestra el error promedio con respecto a las mejores soluciones. Nótese que se muestra la convergencia del error con el AG con población constante (P=100) y con reinicio de la población (población variable con reinicio a Q = 50 soluciones). Para este caso se consideró X = 20 generaciones y R = 5 individuos (para tener una población máxima con 20×5 + 50 = 150 soluciones).

Como se puede observar, los resultados obtenidos con P=100 muestran el mejor desempeño dentro del rango de las 1000 generaciones con un error promedio de 8.5%. El AG con reinicio de la población tiene un error promedio mayor con 9.8%. Si bien, el AG con reinicio se ejecuta más rápido que el AG con población constante, en ambos

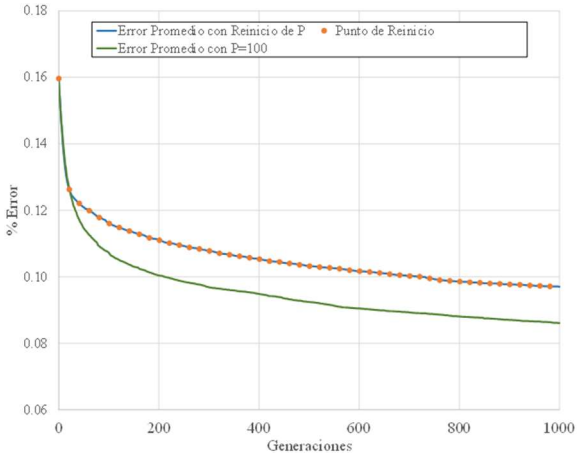


Fig. 5. Convergencia del error promedio con el AG con población fija (P=100) y población variable (reinicio de P cada X=20 generaciones).

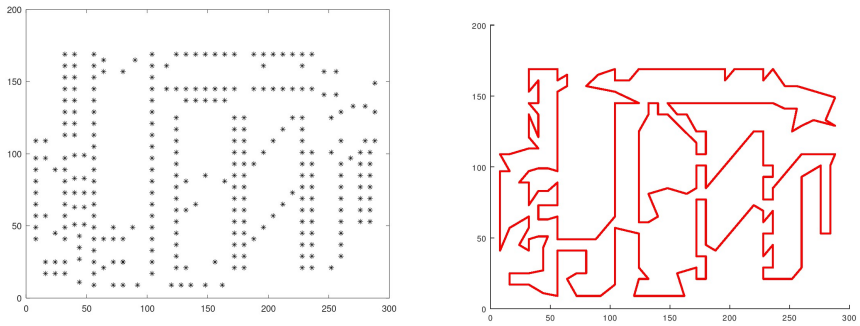


Fig. 6. Solución obtenida para la instancia a280.tsp con el AG con P=100 y 2000 generaciones.

casos se observa un decremento en el error, lo cual corresponde con una convergencia progresiva sin estancamiento hacia un óptimo local.

No obstante, se puede observar que la pendiente de convergencia es más pronunciada (o acelerada) en el caso con *P* constante. Cuando hay reinicio en *P*, la pendiente tiene una desaceleración a partir de la generación 600, lo cual disminuye la pendiente de la convergencia. Con estos resultados se puede estimar que el cambio en el tamaño de la población puede retrasar la convergencia hacia el óptimo local, lo que implicaría considerar un número mayor de generaciones.

Por otro lado, se verifica que la aplicación de los AG con el operador de PMX pueden brindar soluciones con un error promedio del 8.5% en menos de 1000 generaciones. Al analizar el patrón de convergencia, se puede esperar una disminución mayor del error al extender el número de generaciones. Como ejemplo de esto, la Figura 6 presenta la solución (ruta TSP) final obtenida para la instancia a280.tsp con 2000 generaciones y

P=100. Esta solución tiene un error de 6% y presenta un patrón definido característico del TSP.

5. Conclusiones y trabajo a futuro

En el presente trabajo se consideró un AG para resolver redes TSP con más de 200 nodos. En particular, el AG integró el concepto de reinicio de población o población de tamaño variable con el fin de hacer más rápido el AG, y depurar la población para evitar la convergencia hacia un óptimo local de manera prematura.

Si bien el AG con esta integración fue más rápido por el reinicio de población a un tamaño más pequeño, dentro de un rango de 1000 generaciones y 50 instancias de prueba, no se obtuvo una mejora significativa con respecto al AG con tamaño de población fija o constante.

Esta observación es importante para delimitar el ajuste de parámetros de los AG para resolver el TSP. Si bien el tamaño de la población influye en la velocidad y calidad de las soluciones, es más importante mejorar los operadores utilizados para generación de soluciones candidatas (reproducción).

Como trabajo a futuro, se pretende hacer una extensión del operador PMX para hacer el cruzamiento entre varias soluciones [10]. Esto se espera que pueda mejorar la velocidad de convergencia en presencia de poblaciones pequeñas, lo cual contribuiría en el campo de los micro AG (μAG).

Referencias

1. González-Velarde, J. L., Ríos-Mercado, R. Z.: Investigación de operaciones en acción: Aplicación del TSP en problemas de manufactura y logística. Ingenierías, vol. 2, no. 4, pp. 18–23 (1999)
2. Arboleda-Zúñiga, J., Xiomara-López, A., Lorena-Lozano, Y.: El problema de ruteo de vehículos [VRP] y su aplicación en medianas empresas colombianas. Ingenium, vol. 10, no. 27, pp. 29–36 (2016)
3. Liu, W., Li, X., Luo, N., Chen, X.: Common grounding optimization for CVRP. In: 8th International Conference on Industrial Electronics and Applications, ICIEA 2013, pp. 1755–1758 (2013) doi: 10.1109/ICIEA.2013.6566652
4. Ho, S. C., Gendreau, M.: Path relinking for the vehicle routing problem. J Heuristics, 12, pp. 55–72 (2006)
5. Kumar, V., Panneerselvam, R.: A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator. International Journal of Computational Intelligence Research, vol. 13, no. 7, pp. 1717–1733 (2017)
6. Pichpibul, T., Kawtummachai, R.: a heuristic approach based on Clarke-Wright algorithm for open vehicle routing problem. The Scientific World Journal, vol. 2013, pp. 1–11 (2013) doi: 10.1155/2013/874349
7. González-Vargas, G., González-Aristizábal, F.: Metaheurísticas aplicadas al ruteo de vehículos. Un caso de estudio. Parte 1: Formulación del problema. Revista Ingeniería E Investigación, vol. 26, no. 3, pp. 149–156 (2006)
8. Méndez, A., Simón, S., Palumbo, D., Chiachera, E., Carnero, M.: Dos Enfoques para la solución del problema de ruteo de vehículos (CVRP): Aplicación a un Caso Real de Recolección de Residuos. Mecánica Computacional, pp. 9367–9377(2010)

9. Kora, P., Yadlapalli, P.: Crossover operators in genetic algorithms: A review. International Journal of Computer Applications, vol. 162, no. 10, pp. 34–36 (2017)
10. Ting, C. K., Su, C. H., Lee, C. N.: Multi-parent extension of partially mapped crossover for combinatorial optimization problems. Expert Systems with Applications, vol. 37, no. 3, pp. 1879–1886 (2010)
11. Deep, K., Mebrahtu, H.: Variant of partially mapped crossover for the travelling salesman problems. International Journal of Combinatorial Optimization Problems and Informatic, vol. 3, no. 1, pp. 47–69 (2012)
12. Hussain, A., Muhammad, Y. S., Sajid, M. N.: A simulated study of genetic algorithm with a new crossover operator using traveling salesman problem. Journal of Mathematics, vol. 51, no. 5, pp. 61–77 (2019)
13. Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., Dizdarevic, S.: Genetic algorithms for the travelling salesman problem: A Review of representations and operators. Artificial Intelligence Review, vol. 13, no. 2, pp. 129-170 (1999) doi:10.1023/a:1006529012972

ISSN 1870-4069

Aplicación de red neuronal artificial para estimar el impago de préstamos bancarios

Walter Jeremías López Flores

Universidad Tecnológica Centroamericana,
Honduras

wjlopez@unitec.edu

Resumen. El objetivo de este estudio es construir un modelo de Red Neuronal Artificial para la evaluación del riesgo crediticio en un banco de Honduras, mediante la aplicación de técnicas de aprendizaje de máquina supervisado y minería de datos a su cartera crediticia de personas naturales, para estimar si un cliente con determinadas características puede caer en impago de su préstamo. De una población de 38,156 préstamos y 30 características disponibles, 11 cuantitativas y 19 cualitativas se eligieron las variables independientes más representativas mediante un análisis de correlación, estableciendo un modelo base de tipo Perceptrón Multicapa con 6 variables en la capa de entrada, al que se fueron agregando las características de mayor relación con la variable dependiente *EnMora* en la capa de salida, para analizar su efecto sobre la capacidad predictiva del modelo constituido por 4 capas densas de 20 y 24 nodos y una capa central entre estas de tipo *dropout*. Se aplicó sobremuestreo SMOTE al subconjunto de entrenamiento para balancear las clases de la variable dependiente, cuyo modelo final presenta según las matrices de confusión y reportes de clasificación una exactitud de 99%, precisión de 99.43%, sensibilidad del 99.45%, especificidad de 95.15%, una curva ROC cuyo AUC es de 98.41% y una curva *Precision-Recall* con un AUC de 97.46%.

Palabras clave: Aprendizaje de máquina, red neuronal artificial, perceptrón multicapa, riesgo de crédito, impago de préstamos.

Application of Artificial Neural Network to Estimate Bank Loan Default

Abstract. The objective of this study is to build an Artificial Neural Network model for the evaluation of credit risk in a Honduran bank, by applying supervised machine learning techniques and data mining to its loan portfolio of natural persons, to estimate whether a client with certain characteristics may default on his loan. From a population of 38,156 loans and 30 available characteristics, 11 quantitative and 19 qualitative, to choose the most representative independent variables through a correlation analysis, establishing a Multilayer Perceptron base model with 6 variables in the input layer, to which they were adding the characteristics of greater relation with the dependent variable *default* on the output layer, to analyze its effect on the predictive capacity of the model consisting of 4 dense layers of 20 and 24 nodes and a central layer between these of dropout type. SMOTE oversampling was applied to the training

subset to balance the classes of the dependent variable, whose final model presents, according to the confusion matrices and classification reports, an accuracy of 99%, precision of 99.43%, sensitivity of 99.45%, specificity of 95.15%, a ROC curve whose AUC is 98.41% and a Precision-Recall curve with an AUC of 97.46%.

Keywords: Machine learning, artificial neural network, multilayer perceptron, credit risk, loans default.

1. Introducción

El impacto de eventos como la crisis financiera mundial de 2008 y la pandemia mundial de Covid-19 a finales de 2019 genera, entre otros efectos, desaceleración y recesión prolongada en la economía global, fenómenos de los cuales es difícil recuperarse tanto para las personas, empresas y países. Lo anterior conlleva muchos riesgos implícitos, siendo uno de ellos el riesgo de crédito, pues tanto a nivel individual como empresarial, se requiere financiamiento oportuno para superar las adversidades que estas situaciones presentan en las necesidades de las personas y las empresas o particulares que las suplen, pero al presentarse dificultades para generar los ingresos necesarios dentro de ese ciclo económico, que les permitan ser sujetos a crédito ante la banca comercial, se presenta un doble riesgo para los bancos: que sus clientes con préstamos vigentes caigan en mora, y así mismo, ¿cómo evaluar nuevas solicitudes de crédito ante la incertidumbre, para pronosticar qué clientes potenciales con ciertas características a evaluar, puedan caer en mora si no se cuenta con historial crediticio a lo interno del banco?.

Además, al considerar que la Inteligencia Artificial (IA) ha sido una de las áreas de las ciencias computacionales que mayor desarrollo han tenido desde mediados del siglo pasado, especialmente en su rama del aprendizaje automático o de máquina (ML por sus siglas en inglés), el cual tiene una amplia gama de aplicación en diferentes áreas del conocimiento e industrias, a nivel empresarial y financiero, incluyendo la gestión integral del riesgo; es oportuno conducir estudios que permitan a los bancos innovar y mejorar sus niveles de productividad y certeza en la toma de decisiones, especialmente en la colocación de préstamos, que es su negocio principal.

El objetivo de este estudio es proponer un modelo para la evaluación crediticia de los clientes en un banco en Honduras, que pueda ser reproducible para todo el sistema financiero nacional, entendiendo como clientes a las personas naturales que tienen préstamos vigentes y/o solicitan nuevas aplicaciones de crédito, para determinar si pueden o no caer en el impago de los mismos, construyendo una Red Neuronal Artificial a partir de su base de datos interna, diseñada con las bibliotecas de IA disponibles para el lenguaje de programación Python.

2. Marco teórico

El riesgo de crédito es la probabilidad de que una entidad no haga frente, parcial o totalmente, a su obligación de devolver una deuda o rendimiento convenido a su vencimiento sobre un instrumento financiero, ya sea por quiebra, iliquidez o cualquier

otra razón. La evaluación de este tipo de riesgo se basa en la probabilidad de que el prestatario incumpla con sus obligaciones, situación que en el argot financiero se dice que ocurre un *default*, el cual se relaciona con los ciclos económicos así: reduciéndose durante períodos de expansión económica, al mantenerse tasas de impago bajas, y sucediendo lo contrario en períodos de contracción económica [1].

2.1. Uso de redes neuronales artificiales en la estimación de riesgo crediticio

Las Redes Neuronales Artificiales (ANN, por sus siglas en inglés) emulan la estructura y comportamiento del cerebro, utilizando procesos de ML para encontrar solución a diversos problemas, aplicando algoritmos matemáticos que encuentran relaciones no lineales entre conjuntos de datos, soliendo utilizarse como herramientas para predecir tendencias y/o clasificar conjuntos de datos [2].

Existen muchos tipos de ANN, las cuales varían según su aplicación, específicamente las de tipo Perceptrón Multicapa (MLP por sus siglas en inglés), son definidas por [3] como aquellas compuestas por varias capas, generalmente divididas en tres grupos: una capa de entrada (*input layer*), una o varias capas ocultas (*hidden layers*) y una capa de salida (*output layer*); también, constan de pesos entre sus conexiones, la función de activación, el valor para el sesgo (*bias*) y generalmente, se realiza un cálculo del error entre la salida deseada y la real para ajustar los pesos de forma inversa (*backpropagation*).

Respecto al número de capas, estas deben ser mayores si se trata de un problema no lineal y complejo, pero generalmente, un problema se puede representar bastante bien usando una o dos capas ocultas; y el número de neuronas por capa, usualmente se establece por ensayo y error, aunque existe también el criterio de considerar como valor referencial el promedio entre el número de entradas y salidas [4]. Las capas son densas cuando conectan cada neurona en una capa con todas las salidas de la capa anterior, también se conocen como capa completamente conectada [5].

Hay varios tipos de aprendizaje de máquina: supervisado, no supervisado y por refuerzo [6]. Cuando las redes son supervisadas, es requerido que el usuario especifique la salida deseada; a partir de ello, la red aprende a detectar la relación entre las entradas y salidas que fueron suministradas, siguiendo un proceso adaptativo e iterativo; y cuando la red ya ha sido entrenada, se le presentan nuevos datos, que no haya visto antes y se prueba para comprobar la bondad del conjunto de pesos, y cuando ofrece un rendimiento óptimo ya está lista para trabajar [2].

En el estudio [2] se aplica una metodología de medición de riesgo de crédito basado en un modelo MLP a partir de los datos de una cartera comercial, realizando un análisis exploratorio univariante y luego se fueron cruzando variables del cliente, del crédito y del comportamiento con la variable binaria *default* (fallidos y no fallidos), se investigó la correlación entre ellas, para establecer las características particulares del modelo de clasificación y determinar las ponderaciones necesarias para establecer la probabilidad de impago. El uso de ANN para la calificación y evaluación de créditos ha sido efectivo durante la última década, pues su capacidad en tales aplicaciones se debe a la forma en que opera la red y la disponibilidad de datos de entrenamiento, lo cual es más evidente cuando se utilizan redes MLP basadas en el algoritmo de aprendizaje de retropropagación [7].

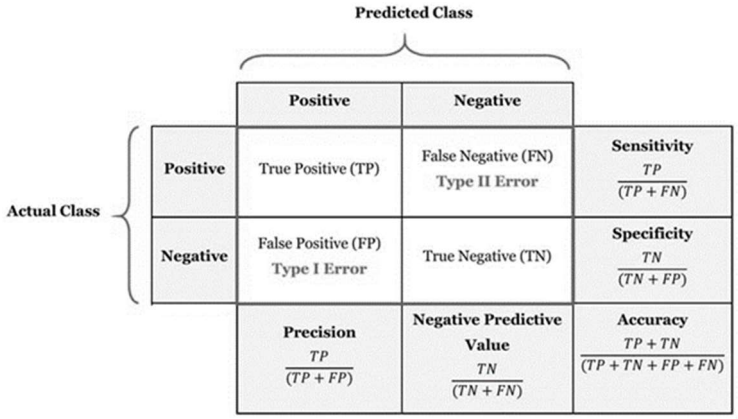


Fig. 1. Esquema de una matriz de confusión y fórmulas de sus étricas [15].

Cuando se alimenta la información de un solicitante de crédito a la ANN, los atributos (respuestas del solicitante a un conjunto de preguntas o características) se toman como entrada a la red y se toma una combinación lineal de ellos con pesos arbitrarios, los atributos se combinan linealmente y están sujetos a una transformación no lineal representada por una determinada función de activación, luego se alimentan como entradas en la siguiente capa y así hasta la función final que produce valores que se pueden comparar con un punto de corte para la clasificación; cada caso de entrenamiento se envía a la red y se compara con el valor observado, el residual se propaga a la red y los pesos se modifican en cada capa según la contribución que cada peso hace al valor de error [7].

En cuanto a la selección de funciones de activación para los modelos ANN, según [4] se realiza por ensayo y error, de acuerdo al problema a resolver o al criterio del investigador, pues aún no existe un criterio estándar en la literatura; comúnmente la función logística o sigmoidea es utilizada con buenos resultados, aunque también se emplean las funciones *heaviside*, *signum*, lineal, *piece-wise linear*, tangente hiperbólica, ReLU (*Rectified Linear Unit*) y *rectifier softplus* descritas en [8], las cuales según [9] son llamadas también funciones de transferencia y se definen como la suma ponderada de la entrada se transforma en una salida en una capa de la red, y propone la siguiente metodología de selección de funciones para los nodos: si son de capa oculta, para MLP o CNN (*Convolutional Neural Network*) usar ReLU y en redes recurrentes usar sigmoidea o tangente; luego para la capa de salida, si es un problema de regresión, usar lineal y para clasificación, si es binaria, sigmoidea; y si es multiclase, *softmax* o sigmoidea.

Los métodos de regularización estocástica, como el abandono (*dropout*) son una forma eficaz de evitar el ajuste excesivo o sobre-entrenamiento (*over-fitting*) y se utilizan a menudo en la práctica debido a su simplicidad [10]. La capa de abandono en una red, establece aleatoriamente las unidades de entrada en 0 con una frecuencia de velocidad a cada paso durante el tiempo de entrenamiento, lo que ayuda a evitar el sobreajuste; las entradas que no están configuradas en 0 se escalan en $1/(1 - tasa)$ de modo que la suma de todas las entradas no cambia [11]. Los modelos de Keras, tienen dos modos: entrenamiento y prueba, los mecanismos de regularización se desactivan al

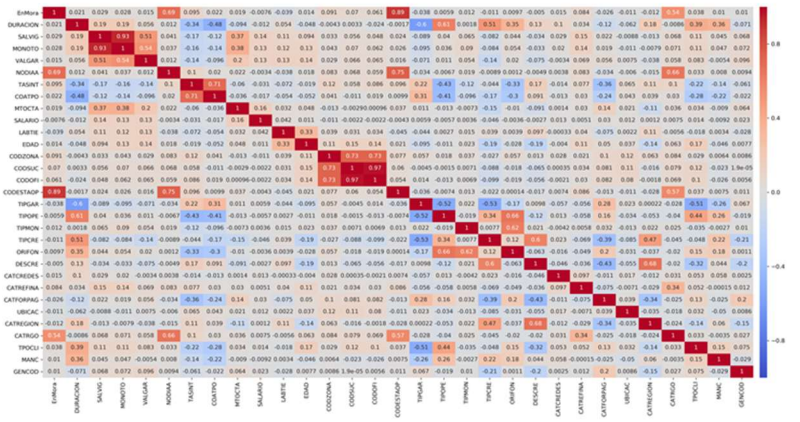


Fig. 2. Matriz de correlación de variables para el modelo MLP.

momento de prueba, son reflejados en la pérdida de tiempo de entrenamiento, pero no en la pérdida de tiempo de prueba; además, la pérdida de entrenamiento es el promedio de las pérdidas para cada lote de datos de entrenamiento, durante la época (*epoch*) actual, debido a que su modelo cambia con el tiempo, la pérdida en los primeros lotes de una época es generalmente mayor que en los últimos lotes, resultando al final en una pérdida menor. En los modelos ANN, una época es un límite arbitrario, definido como una pasada sobre todo el conjunto de datos, que se utiliza para separar el entrenamiento en distintas fases, para el registro y evaluación periódica [12].

El propósito de las funciones de pérdida dentro de las ANN, es calcular la cantidad que un modelo debe buscar minimizar durante el entrenamiento. La función probabilística de pérdida llamada entropía cruzada binaria (*binary crossentropy*), calcula la pérdida entre etiquetas verdaderas y predichas cuando solo hay dos clases de etiquetas (0 y 1), para cada ejemplo debería haber un único valor de punto flotante por predicción [13].

Existe un algoritmo para optimización basada en gradientes de primer orden de funciones objetivas estocásticas, y en estimaciones adaptativas de momentos de orden inferior llamado *Adam*, que es un método sencillo de implementar, computacionalmente eficiente, con pocos requisitos de memoria, invariante al cambio de escala diagonal de los gradientes y muy adecuado para problemas con grandes cantidades de datos o parámetros, así como para objetivos no estacionarios y problemas con gradientes muy ruidosos o dispersos [10].

2.2. Matriz de confusión, métricas y curvas ROC y Precision-Recall

En los problemas de decisión binaria, el clasificador etiqueta ejemplos como positivos o negativos. La decisión tomada por el clasificador se puede representar en una estructura conocida como matriz de confusión o tabla de contingencia, la cual tiene cuatro categorías: los verdaderos positivos (TP) son observaciones correctamente etiquetadas como positivas, los falsos positivos (FP) se refieren a ejemplos negativos etiquetados incorrectamente como positivos (llamados error tipo I); los verdaderos negativos (TN) corresponden a los negativos etiquetados correctamente como

negativos y finalmente, los falsos negativos (FN), que se refieren a ejemplos positivos etiquetados incorrectamente negativos (llamados error tipo II) [14]. A continuación, se muestra la estructura de una matriz de confusión y sus principales métricas.

La precisión o *Precision*, indica la fracción de los casos predichos correctamente que resultaron realmente positivos, la cual es una métrica útil en los casos en que los FP son una preocupación mayor que los FN; mientras que la sensibilidad o *Sensitivity* (también conocida como *Recall*), indica la fracción de los casos positivos reales que pudo predecir correctamente el modelo, siendo una métrica útil en los casos en que los FN triunfan sobre los FP [16].

Otra métrica importante que surge de combinar *Precision* y *Recall* es el *F1-score* o puntuación F1, que por definición es la media armónica entre estas; la cual, según [17] se usa ampliamente para medir el éxito de un clasificador binario cuando una clase es rara, mientras que otras puntuaciones como *micro average* (micropromedio), *macro average* (macropromedio) y el promedio $F1$ por instancia, se utilizan en la clasificación de múltiples etiquetas y también demuestran cómo esta media armónica puede ser expresada también por sustitución como una función de cuenta de TP, FP y FN como se indica en la Ecuación (1):

$$F1 = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} = \frac{2TP}{2TP+FP+FN}. \quad (1)$$

En la práctica, cuando se intenta aumentar la precisión de los modelos, la sensibilidad también disminuye y viceversa; el puntaje $F1$ captura ambas tendencias en un solo valor y brinda una idea combinada de estas métricas, alcanzando su valor máximo cuando $Precision = Recall$ [16].

En la actualidad, el ML se ha alejado de simplemente presentar resultados de precisión al realizar una validación empírica de nuevos algoritmos, especialmente cuando estos generan probabilidades de valores de clase, por lo que se recomiendan utilizar curvas ROC (*Receiver Operator Characteristic*) que muestran cómo el número de ejemplos positivos clasificados correctamente varía con el número de ejemplos negativos clasificados incorrectamente, y curvas *Precision-Recall* (PR) como una alternativa a las curvas ROC para tareas con un gran sesgo en la distribución de clases [14].

2.3. Conjuntos de datos desequilibrados y sobremuestreo SMOTE

Los conjuntos de datos desequilibrados son un problema común dentro del ML, ya que la mayoría de modelos de clasificación tradicionales no pueden manejarlos, pues ocurre un alto costo de clasificación errónea a menudo en la clase minoritaria, porque el modelo intentará clasificar toda la muestra de datos en la clase mayoritaria; para abordar este problema se usa una técnica llamada *oversampling* (sobremuestreo) que permite crear una muestra de clase específica para que la distribución de clases del conjunto de datos original pueda equilibrarse.

La técnica SMOTE (*Synthetic Minority Oversampling Technique*) es una de las técnicas de sobremuestreo más populares, la cual funciona encontrando un grupo de k vecinos más cercanos en el espacio de características, luego encuentra aleatoriamente un punto dentro de este grupo, y finalmente usa el promedio ponderado para "falsificar" el nuevo punto de datos [18].

Tabla 1. Diseño de los modelos MLP utilizados en el estudio.

Estructura			**Parámetros**					
Capa		Nodos	Modelo base	1: base + CATRGO	2: base + ESTAOP	3: base + NODIAA	4: base + CATRGO y NODIAA	5: Todas
Entrada		$X =$	6	7	7	7	8	9
Oculta (densa)	1	20	140	160	160	160	180	200
Oculta (densa)	2	24	504	504	504	504	504	504
Oculta (dropout)	3	24	0	0	0	0	0	0
Oculta (densa)	4	20	500	500	500	500	500	500
Oculta (densa)	5	24	504	504	504	504	504	504
Salida (densa)		1	25	25	25	25	25	25
Parámetros totales:			1,673	1,693	1,693	1,693	1,713	1,733
Parámetros entrenables:			1,673	1,693	1,693	1,693	1,713	1,733
Parámetros no entrenables:			0	0	0	0	0	0

Tabla 2. Resultados de entrenamiento de los modelos MLP diseñados.

Epoch	**Modelo base**		**1: base + CATRGO**		**2: base + ESTAOP**		**3: base + NODIAA**		**4: base + CATRGO y NODIAA**		**5: Todas**	
	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
1	0.3647	0.8926	0.3607	0.8901	0.3625	0.8940	0.0855	0.9765	0.1185	0.9730	0.4976	0.9554
2	0.3421	0.8966	0.2810	0.9107	0.1695	0.9451	0.0493	0.9858	0.0471	0.9879	0.0482	0.9875
3	0.3366	0.8966	0.2696	0.9133	0.0341	0.9912	0.0436	0.9880	0.0399	0.9893	0.0403	0.9894
4	0.3366	0.8966	0.2606	0.9172	0.0217	0.9950	0.0405	0.9889	0.0373	0.9903	0.0361	0.9911
5	0.3341	0.8966	0.2595	0.9154	0.0173	0.9959	0.0402	0.8996	0.0366	0.9907	0.0329	0.9915
Score	0.3331	89.66%	0.2489	92.03%	0.0051	100.0%	0.0411	99.23%	0.0327	99.12%	0.0317	99.00%

Para encontrar la distancia entre los puntos vecinos, se discute en [19] que la función de distancia euclidiana es inapropiada para atributos nominales, y otra alternativa llamada VDM es inapropiada para atributos continuos, por lo ninguna es suficiente por sí misma para un conjunto de datos que utilice ambas; ante lo cual, describen una función adecuada de distancia heterogénea llamada *Heterogeneous Value Difference Metric* (HVDM) con la Ecuación (2):

$$HVDM(x, y) = \sqrt{\sum_{a=1}^{m} {d_a}^2 (x_a, y_a)}. \quad (2)$$

En Python existe un paquete llamado SMOTE-NC que permite crear datos sintéticos tanto para variables categóricas como continuas dentro de un conjunto de datos,

decidiendo las categorías de una nueva muestra generada eligiendo la más frecuente de los vecinos más cercanos presentes durante la generación [20].

El procedimiento de validación cruzada después del sobremuestreo conduce a resultados sobre optimistas, mientras que realizarlo en los conjuntos de entrenamiento en cada iteración de un procedimiento de validación cruzada es la forma correcta de validar los resultados en escenarios desequilibrados [21].

3. Metodología

El estudio es de enfoque cuantitativo, alcance explicativo y de tipo experimental; los datos provienen de fuente secundaria, extraídos de la base de datos interna de un banco de Honduras mediante consulta electrónica del reporte de Central de Información Crediticia (CIC) que se entrega al ente regulador: la Comisión Nacional de Banca y Seguros (CNBS) de dicho país, con saldos de los préstamos otorgados y que aún no han sido cancelados, los cuales van desde el año 2000 hasta diciembre de 2020.

El conjunto de datos se compone de 51,696 registros y 30 campos de los que se identificarán las variables independientes más adecuadas para construir el modelo.

En cuanto a las herramientas utilizadas: la limpieza de los datos, su preparación y exploración preliminar fue realizada en Excel y se empleó el *framework* Anaconda para la manipulación de datos, diseño, entrenamiento, prueba y visualización de un modelo MLP en lenguaje Python 3.7 con las bibliotecas de IA Keras y Scikit-Learn codificando los algoritmos de ML supervisado en el editor Jupyter Notebook 6.0.

3.1. Población y muestra

La población a estudiar estará compuesta solo por los préstamos concedidos a personas naturales; el tipo de muestra será censal, por lo que no se requiere definir tamaño muestral ni utilizar tipo de muestreo alguno, ya que se analizará todo el universo completo.

Al delimitar la cartera crediticia filtrando únicamente los préstamos de personas naturales, esta queda establecida en 38,156 observaciones, de las cuales solo el 10.38% (3,959) se encuentran en mora y las restantes 34,197 no lo están (clase *EnMora* = 0), que sería la clase mayoritaria con un 89.62% de las observaciones.

3.2. Identificación y selección de variables

La variable dependiente para el modelo es de tipo dicotómica y llamada *EnMora*, la cual indicará si el préstamo está en mora con el valor 1, o no lo está con el valor 0, en función de un arreglo bidimensional de variables predictoras, compuesto por 11 cuantitativas y 19 cualitativas, para un *dataframe* de 30 variables disponibles, analizando la intensidad y dirección de la relación de estas con la variable dependiente mediante la matriz de correlación de la Fig. 2.

Tabla 3. Matrices de confusión con los resultados de entrenamiento y prueba.

Modelo		Entrenamiento				Prueba			
		0	1	Total	Loss/Acc	0	1	Total	Loss/Acc
Base	0	23,946	0	23,946	0.3316	10,251	0	10,251	0.3336
	1	2,763	0	2,763	89.66%	1,196	0	1,196	89.55%
	Total	26,709	0	26,709		11,447	0	11,447	
1: base + CATRGO	0	23,567	379	23,946	0.2489	10,070	181	10,251	0.2497
	1	1,750	1,013	2,763	92.03%	730	466	1,196	92.04%
	Total	25,317	1,392	26,709		10,800	647	11,447	
2: base + ESTAOP	0	23,946	0	23,946	0.0051	10,251	0	10,251	0.0051
	1	1	2,762	2,763	100.0%	0	1,196	1,196	100.0%
	Total	23,947	2,762	26,709		10,251	1,196	11,447	
3: base + NODIAA	0	23,869	77	23,946	0.0411	10,221	30	10,251	0.0463
	1	128	2,635	2,763	99.23%	70	1,126	1,196	99.13%
	Total	23,997	2,712	26,709		10,291	1,156	11,447	
4: base + CATRGO y NODIAA	0	23,936	10	23,946	0.0327	10,227	4	10,251	0.0402
	1	224	2,539	2,763	99.12%	120	1,076	1,196	98.91%
	Total	24,160	2,549	26,709		10,347	1,080	11,447	
5: Todas	0	23,942	4	23,946	0.0317	10,248	3	10,251	0.0384
	1	262	2,501	2,763	99.00%	135	1,061	1,196	98.79%
	Total	24,204	2,505	26,709		10,383	1,064	11,447	

Tabla 4. Reportes de clasificación de los modelos con datos de prueba.

Modelo base

	precision	recall	f1-score	support
0	0.90	1.00	0.94	10251
1	0.00	0.00	0.00	1196
accuracy			0.90	11447
macro avg	0.45	0.50	0.47	11447
weighted avg	0.80	0.90	0.85	11447

Modelo 1: base + CATRGO

	precision	recall	f1-score	support
0	0.92	0.99	0.95	10251
1	0.82	0.21	0.34	1196
accuracy			0.91	11447
macro avg	0.87	0.60	0.65	11447
weighted avg	0.91	0.91	0.89	11447

Modelo 2: base + ESTAOP

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10251
1	1.00	1.00	1.00	1196
accuracy			1.00	11447
macro avg	1.00	1.00	1.00	11447
weighted avg	1.00	1.00	1.00	11447

Modelo 3: base + NODIAA

	precision	recall	f1-score	support
0	0.99	1.00	1.00	10251
1	0.98	0.94	0.96	1196
accuracy			0.99	11447
macro avg	0.99	0.97	0.98	11447
weighted avg	0.99	0.99	0.99	11447

Modelo 4: base + CATRGO y NODIAA

	precision	recall	f1-score	support
0	0.99	1.00	0.99	10251
1	1.00	0.90	0.95	1196
accuracy			0.99	11447
macro avg	0.99	0.95	0.97	11447
weighted avg	0.99	0.99	0.99	11447

Modelo 5: Todas

	precision	recall	f1-score	support
0	0.99	1.00	0.99	10251
1	1.00	0.89	0.94	1196
accuracy			0.99	11447
macro avg	0.99	0.94	0.97	11447
weighted avg	0.99	0.99	0.99	11447

El número de días de atraso del préstamo (NODIAA) es la única variable independiente cuantitativa que presenta una relación fuerte con la mora (0.69), mientras que solo dos características cualitativas presentan una relación importante con la

variable respuesta, siendo la más fuerte el estado de operación (ESTAOP) con 0.89, seguida de la categoría de riesgo (CATRGO) con 0.54. Todas las restantes no llegan a 0.1, por lo que no son significativas como para considerar que su efecto sea influyente dentro del modelo; a pesar de ello, se seleccionaron las siguientes 6 variables por considerarse de interés por los analistas de crédito del banco para predecir el comportamiento de la mora, dado que representan características inherentes tanto de los préstamos: duración del préstamo en años (DURACION), el monto del capital otorgado (MONOTO), y la tasa de interés (TASINT); así como de las personas: años de laborar (LABTIE), edad del cliente (EDAD) y su género (GENCOD).

4. Resultados

Se construyó un modelo base de tipo MLP con 4 capas densas ocultas y una capa *dropout* intermedia, utilizando funciones de activación de tipo ReLU, mientras que la capa de salida tendrá un solo nodo, con una función de activación sigmoidea, ideal para predecir una variable binaria. Para el modelo base se tomaron las 6 variables de entrada seleccionadas anteriormente, por ser atributos básicos inherentes a los préstamos y clientes, y también porque son conocidas de antemano, aún para nuevos clientes que tomarán un préstamo por primera vez.

La escala de la variable MONOTO, se cambió, normalizándola en lugar de usar sus valores originales. Las 3 variables que resultaron con mayor correlación se fueron agregando al modelo base para analizar de qué manera contribuyen a explicar la mora, tanto individualmente como en conjunto, combinado en distintos experimentos las variables en la capa de entrada, así como el número de nodos que conforman cada capa oculta hasta alcanzar los mejores resultados (ver Tabla 1).

El conjunto de datos se dividió en dos subconjuntos: el de entrenamiento conformado por el 70% de las unidades muestrales que equivale a 26,709 préstamos, y el de prueba con el 30% restante equivalente a 11,447. El entrenamiento de los modelos, se realizó durante cinco *epochs*, utilizando el optimizador *Adam* y la función de pérdida *binary crossentropy* (ver Tabla 2). Tanto la función de pérdida se va reduciendo en todos los modelos con respecto al base, así como la exactitud de todos va aumentando en cada época.

Al comparar los resultados de clasificación y métricas entre los modelos con los datos de entrenamiento y prueba para evaluar la capacidad de clasificación de cada uno (ver Tabla 3), se demuestra que no hubo *over-fitting*, por lo que se presentan los reportes de clasificación de los modelos de prueba (ver Tabla 4). Los resultados, reflejan que el modelo base, con las características proporcionadas fue incapaz de aprender a clasificar los préstamos cuando la clase $EnMora = 1$, lo cual se explica por la baja correlación que mantienen con la variable dependiente, además de ser la clase minoritaria.

Cabe reportar que se realizó una prueba en el modelo base con todas las demás variables que fueron excluidas, obteniendo resultados similares, por lo que ninguna de ellas contribuye a mejorar el aprendizaje de este para clasificar la mora. Al agregar la categoría de riesgo del préstamo al modelo base, este comienza a aprender a clasificar los que caen en mora, pero comete muchos errores de Tipo I y Tipo II, situación que se reduce sustancialmente al sustituir esta variable por el número de días de atraso en el experimento 3, el cual aprende a predecir la clase minoritaria con buenos resultados.

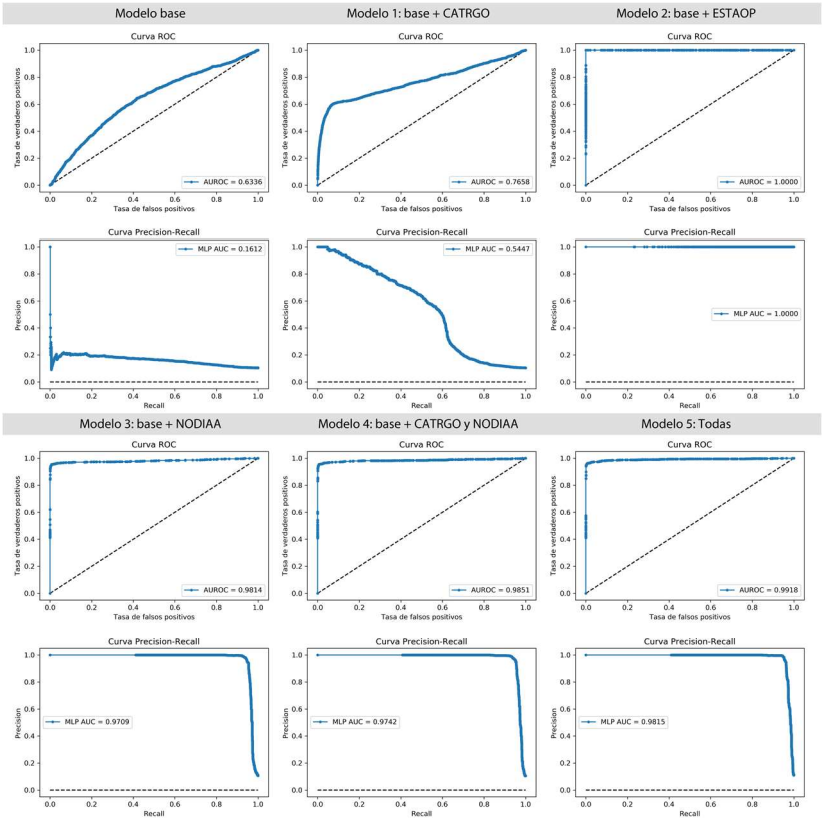


Fig. 3. Curvas ROC y PR de los modelos MLP con los datos de prueba.

Tabla 5. Comparación de resultados del modelo MLP optimizado con SMOTE.

Modelo 3 con datos de prueba entrenado con clases desbalanceadas					**Modelo 3 con datos de prueba entrenado con clases balanceadas mediante SMOTE-NC**			
	0	**1**	**Total**	**Métricas**	**0**	**1**	**Total**	**Métricas**
0	10,221	30	10,251	Sen = 0.9971	10,195	56	10,251	Sen = 0.9945
1	70	1,126	1,196	Esp = 0.9415	58	1,138	1,196	Esp = 0.9515
Total	10,291	1,156	11,447	Acc = 0.9913	10,253	1,194	11,447	Acc = 0.9900
Pre = 0.9932			Neg Pre = 0.9740		Pre = 0.9943		Neg Pre = 0.9531	

	precision	recall	f1-score	support
0	0.99	1.00	1.00	10251
1	0.97	0.94	0.96	1196
accuracy			0.99	11447
macro avg	0.98	0.97	0.98	11447
weighted avg	0.99	0.99	0.99	11447

	precision	recall	f1-score	support
0	0.99	0.99	0.99	10251
1	0.95	0.95	0.95	1196
accuracy			0.99	11447
macro avg	0.97	0.97	0.97	11447
weighted avg	0.99	0.99	0.99	11447

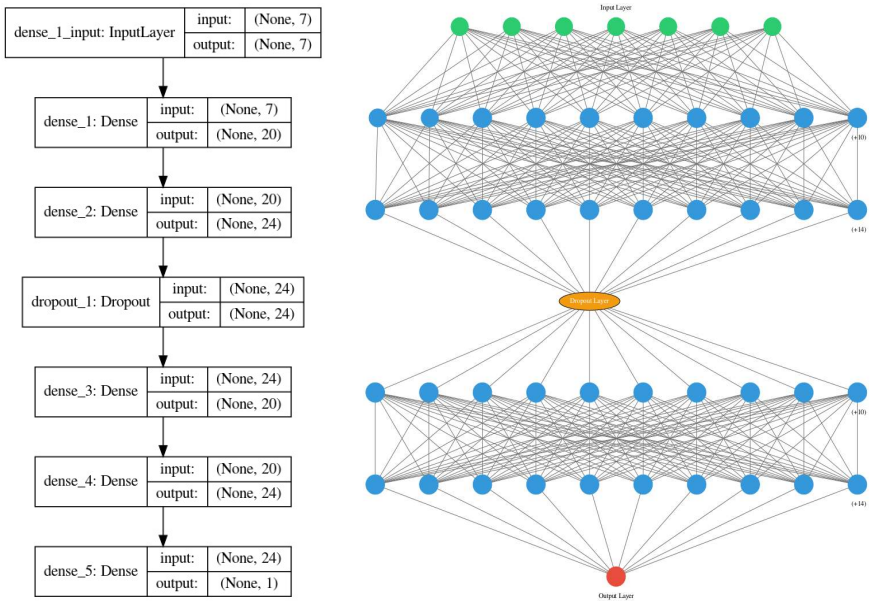


Fig. 4. Diagrama del modelo MLP seleccionado.

El modelo aprende a identificar los préstamos que caen en mora de manera perfecta en el experimento 2 dado que conoce el estado de la operación, pero esto debido a que todos los que tienen estado vigente se encuentran al día, y cualquier estado distinto implicaría que el préstamo ha caído en mora, por lo que no es una característica idónea para predecir; también, al agregar todas las variables restantes al modelo base se obtienen resultados casi perfectos, pero esto porque está incluida ESTAOP, y es como si las otras le restaran exactitud (ver Fig. 3).

En base a los resultados obtenidos, se selecciona como preferido el modelo 3, que presenta un AUC bajo la curva ROC (AUROC) de 98.14%, un AUC de 97.09% bajo la curva PR, y una exactitud de 99.13%; aunque los modelos 4 y 5 obtuvieron mayores valores en dichos indicadores, dicho experimento 3 presenta el *Recall* y *F1-score* más altos para la clase minoritaria, con 0.94 y 0.96 respectivamente, observando también que en los siguientes modelos estos fueron disminuyendo, por lo que se elige este como el modelo MLP preferido, el cual se representa gráficamente en la Fig. 4.

4.1. Sobremuestreo SMOTE

A pesar que el modelo seleccionado demuestra en sus resultados, tasas altas de reconocimiento sobre la clase minoritaria, se le aplica un sobremuestreo SMOTE al subconjunto de datos de entrenamiento, para analizar si presenta alguna mejora al entrenar con clases balanceadas, para lo cual se utilizó el paquete SMOTE-NC de la biblioteca *imblearn* de Python, indicándole que la variable GENCOD era categórica, quedando cada clase binaria con 23,946 observaciones para un *dataframe* de 47,892 registros. Al volver a entrenar el modelo 3 con los nuevos datos sintéticos equilibrados

en sus clases de salida, durante 5 e*pochs*, y luego proporcionarle los datos de prueba se obtienen los resultados presentados en la Tabla 5, se alcanzan prácticamente los mismos valores de exactitud, precisión y sensibilidad; la capacidad de predicción de valores negativos baja a 95.5%, un 2.15% menos con respecto al mismo modelo entrenando sin sobremuestreo, pero la especificidad sube en 1.06%, lo cual indica una ligera mejora en la capacidad de categorizar la clase minoritaria al utilizar el sobremuestreo SMOTE para entrenar el modelo seleccionado.

Los errores de Tipo I y II se reducen y mejoran las curvas de clasificación después de aplicar SMOTE para optimizar el modelo elegido, su AUROC subió en 0.27% de 0.9814 a 0.9841, y el AUC de la curva PR sube 0.37%, de 0.9709 a 0.9746.

5. Conclusiones y trabajo a futuro

El estudio entrega un modelo MLP construido en Python a partir de los datos internos que conforman la cartera de préstamos de personas naturales de un banco hondureño. Dicha herramienta es de gran utilidad práctica para los analistas de crédito, ya que les permitirá evaluar si un cliente con ciertas características ingresadas en las variables de entrada seleccionadas, puede caer en mora del préstamo que tiene vigente, o que está solicitando en caso de ser nuevo, siendo el número de días de atraso la variable que más contribuye a explicar la mora para que el modelo aprenda a clasificar los préstamos o solicitudes que pueden caer en impago.

El modelo seleccionado aprende a clasificar las categorías binarias de la variable de salida de manera adecuada, inclusive de la clase minoritaria sin necesidad de aplicar sobremuestreo, aunque su capacidad de clasificación mejora ligeramente cuando las etiquetas de esta se encuentran balanceadas.

La investigación al estar delimitada a un solo banco, no provee un modelo generalizado para todo el sistema financiero del país en cuestión; sin embargo, la metodología propuesta puede replicarse en cualquier otro banco, cooperativa o empresa financiera del sistema nacional, o de cualquier otro país, con solo cambiar los datos de entrada y ajustando el algoritmo a estos. De igual manera, el ente regulador, puede diseñar y entrenar un modelo general, a partir de la base de datos consolidada que mantiene de todas las entidades supervisadas del sistema financiero dentro de su CIC.

En investigaciones futuras, se pueden considerar variables macroeconómicas dentro del modelo, u otras variables exógenas, para no limitarlo a los datos internos de un banco u otra institución financiera.

Referencias

1. Saavedra-García, M. L., Saavedra García, M. J.: Modelos para medir el riesgo de crédito de la banca. Cuadernos de Administración, vol. 23, no. 40, pp. 295–319 (2010)
2. Pérez-Ramírez, F. O., Fernández-Castaño, H.: Las redes neuronales y la evaluación del riesgo de crédito. Revista Ingenierías, vol. 6, no. 10, pp. 77–91 http://udem.scimago.es/index.php/ingenierias/article/view/225. (2007)
3. Peña, M., Orellana, J.: Red neuronal para clasificación de riesgo en cooperativas de ahorro y crédito. Congreso de Ciencia y Tecnología, vol. 13, no. 1, pp. 121–124 (2018) doi:10.24133/cctespe.v13i1.710

4. Llano, L., Hoyos, A., Arias, F., Velásquez, J.: Comparación del desempeño de funciones de activación en redes Feedforward para aproximar funciones de datos con y sin ruido. Avances en Sistemas e Informática, vol. 4, no, 2, pp. 79–87 (2007)
5. Google.: Glosario sobre aprendizaje automático. Obtenido de Developers Google: https://developers.google.com/machine-learning/crash-course/glossary
6. Lee, I., Shin, Y. J.: Machine learning for enterprises: Applications, algorithm selection, and challenges. Business Horizons, vol. 63, 2, pp. 157–170 (2020) doi: 10.1016/j.bushor.2019.10.005
7. Khashman, A.: Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. Expert Systems with Applications, vol. 37, pp. 6233–6239 (2010) doi: 10.1016/j.eswa.2010.02.101
8. Raschka, S.: Activation functions for artificial neural networks. Obtenido de mlxtend: https://rasbt.github.io/mlxtend/user_guide/general_concepts/activation-functions (2016)
9. Brownlee, J.: How to choose an activation function for deep learning. Obtenido de Machine Learning Mastery: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/ (2021)
10. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization (2017) arXiv.org. doi: arckiv.org/pdf/1412.6980v9.pdf
11. Keras. (s.f.).: Dropout layer. Obtenido de Keras API reference: Regularization layers: https://keras.io/api/layers/regularization_layers/dropout/.
12. Keras. (s.f.).: Keras FAQ. Obtenido de Keras.io: https://keras.io/getting_started/faq/
13. Keras. (s.f.).: Probabilistic losses. Obtenido de Keras API Reference: Losses: https://keras.io/api/losses/probabilistic_losses/
14. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning - ICML '06 (2006) doi:10.1145/1143844.1143874
15. Data Science.: What is confusion matrix and advanced classification metrics? Obtenido de Data Science and Machine Learning: https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html. (2019)
16. Bhandari, A.: Everything you should know about confusion matrix for machine learning. Obtenido de Medium Analytics Vidhya: https://analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/ (2020)
17. Lipton, Z. C., Elkan, C., Narayanaswamy, B.: Thresholding classifiers to maximize F1 score. ArXiv.org (2014) doi:arxiv.org/abs/1402.1892
18. Jiang, P., Zhang, J., Zou, J.: Credit card fraud detection using auto encoder neural network. arXiv.org (2019) doi:arxiv.org/abs/1908.11553
19. Wilson, D. R., Martínez, T. R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research, vol. 6, pp. 1–34 (1997)
20. Aguilar, F.: SMOTE-NC in ML categorization models for imbalanced datasets. Obtenido de medium analytics vidhya https://medium.com/analytics-vidhya/smote-nc-in-ml-categorization-models-fo-imbalanced-datasets-8adbdcf08c25 (2019)
21. Senoane-Santos, M., Pompeu-Soares, J., Henriques-Abreu, P., Araújo, H. Santos, J.: Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches. IEEE Computational Inteligence Magazine, vol. 13, no. 4, pp. 59–76 (2018) doi:10.1109/MCI.2018.2866730

ISSN 1870-4069

Gestión y presentación de información de una biblioteca por medio de XML

Oscar Eduardo González Ramos, Mireya Tovar Vidal

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

oscare.gonzalez@viep.com.mx, mtovar@cs.buap.mx

Resumen. En la actualidad, la información en internet crece radicalmente día a día, por ello surge la necesidad de tener organizada la información de modo que pueda ser obtenida de manera fácil y rápida. En este artículo se presenta una propuesta de estructuración de la información de una biblioteca, en específico libros, por medio de XML, presentando dicha información de manera visual en una plataforma web, donde el XML es el núcleo del proyecto, de aquí se obtienen y se almacenan los datos. Con la ayuda del API DOM se acceden a los datos almacenados en el XML y por medio de PHP se procesan para obtener un funcionamiento de ABC (Altas, Bajas y Cambios), accediendo al nodo raíz y a partir de este se consultan sus hijos y las hojas de estos. Obteniendo de esta manera una plataforma con la cual se puede tener un control y gestionar los elementos de una biblioteca.

Palabras clave: Web semántica, XML, DOM.

Management and Presentation of Information From a Library by Means of XML

Abstract. Today, information on the internet grows radically every day, which is why there is a need to have information organized so that it can be obtained easily and quickly. This article presents a proposal for structuring the information of a library, in specific books, in XML format, presenting said information visually on a web platform, where XML is the core of the project, from here the data. With the help of the DOM API, the data stored in the XML are accessed and through PHP it is processed to obtain an ABC operation (Additions, Removals and Changes), the root node is accessed and from this its children are consulted and the leaves of these. Being in this way that you have a platform with which you can have control and manage the elements of a library.

Keywords: Semantic web, XML, DOM.

1. Introducción

La web semántica es una extensión de la web como se conoce hoy en día, donde la información se encuentra bien definida, a tal grado que permite una mejor coordinación entre humanos y máquinas [2]. La necesidad de gestionar contenidos informáticos de una manera rápida y organizada hoy en día es absolutamente necesaria. XML y las especificaciones derivantes ofrecen una gama amplia de soluciones a diferentes problemas que se tienen en el campo de la computación.

La forma de hacer referencia a los atributos de formato en la edición de textos se realizaba mediante una notación específica desarrollada ex profeso, tratándose de marcas que se añadían al documento original para indicar información, definiendo la sintaxis según el caso [4]. XML no es más que un lenguaje de etiquetas creado para codificar otros lenguajes que definen la estructura y el contenido de documentos. Hoy en día las aplicaciones del lenguaje XML son bastantes, ya que se ha usado para definir archivos de configuración, protocolos de comunicación, entre otros.

Algunos de sus usos más extendidos son el intercambio de datos entre sistemas, base de datos o conversores. En el caso de intercambio de datos entre sistemas, uno de los objetivos fundamentales es permitir la posibilidad de realizar intercambio de datos de manera estructurada entre sistemas distintos. XML al ser un lenguaje estandarizado y de texto plano, hace que la transferencia sea bastante ágil e independiente sin importar la plataforma o el lenguaje en la que se tienen implementado los sistemas [5].

Como base de datos, XML permite almacenar información de manera estandarizada, los cuales pueden ser tratados prácticamente por cualquier lenguaje de programación. Su manejo es relativamente más sencillo comparado con los gestores de bases de datos. Como conversor, actualmente existen muchos formatos que ofrecen dicha conversión a XML, ya sean PDF, HTML, .text, .docx o XHTML. XML es uno de los lenguajes más utilizados en el tema de la administración y recuperación de la información, ya que es un lenguaje estandarizado y los desarrolladores pueden crear estructuras propias para el manejo y manipulación de los datos que se quieran representar [10].

XML facilita el tratamiento de los datos, ya que los programas desarrollados en cualquier lenguaje de programación pueden ser asistidos mediante el uso de APIs, que no son más que partes de código que implementan las funcionalidades básicas para el manejo y tratamiento de los datos, para XML existen DOM y SAX. DOM (Document Object Model) es recomendado por la W3C, ya que da una representación orientada a objetos basada en árboles del documento XML, estableciendo nodos de información (datos o metadatos). SAX (Simple API for XML) tiene menos capacidad que el DOM, recorre el documento y avisa cuando encuentra una etiqueta y cuando encuentra un contenido específico (eventos).

Ambas APIs cargan el documento en formato XML, por otro lado, el paso de documentos XML a HTML, se realiza por medio del estándar XSL (Extended Stylesheets Language), cuyo núcleo es XSLT (Extended Stylesheets Language Transformations) [4]. En este trabajo se presenta el desarrollo de un sistema de control bibliotecario, donde en una primera instancia se han utilizado únicamente libros como parte de los recursos de la biblioteca, creando un ABC (Altas, Bajas y Cambios) en los datos que se almacenan en un archivo con formato XML, para lograr así un control.

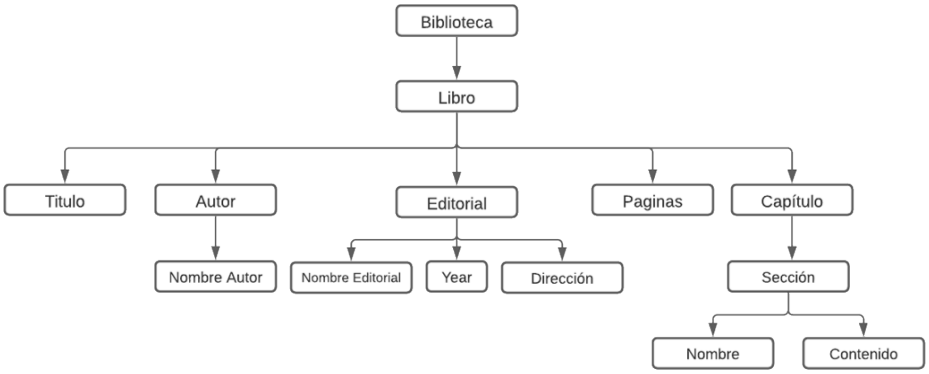


Fig. 1. Árbol Biblioteca.

Para lograr esto sin el uso de una base de datos como MYSQL, se utilizó el Lenguaje de programación PHP junto con el API DOM que se proporciona para el manejo de datos XML, que se mencionó anteriormente. En específico se usó una variante para PHP llamada SimpleXmlElement, la información se representa a través de un sitio web implementado bajo HTML5 y PHP para el manejo del XML, así como también se presenta la alternativa de mostrar la información por medio de XSLT.

El artículo está organizado de la siguiente manera: en la sección 2 correspondiente al estado del arte, se muestran algunos trabajos relacionados que manipulan XML para dar solución a alguna situación, desde trabajos de tesis, hasta pequeñas apps de gestión de usuarios. En la sección 3 se plantea una propuesta para dar solución a el almacén de datos de una biblioteca, en específico para esta etapa se manipulan datos de libros relacionados a la web semántica, donde se específica el árbol de la estructura para el XML.

En la sección 4 se muestra la implementación de la plataforma usando archivos XML como almacén de datos para una aplicación, en vez de usar una base de datos tradicional, la aplicación es web, la cual está desarrollada bajo HTML5, Bootstrap 4, PHP 7 para manipular el DOM con SimpleXmlElement y Datatables con JQuery para las consultas y filtros. Finalmente, en la sección 5 se tienen las conclusiones del trabajo realizado y las referencias consultadas.

2. Estado del arte

A continuación se presentan trabajos relacionados al nuestro, donde como base se maneja un archivo XML como núcleo para el almacenamiento de datos y por medio de Android o HTML se muestran los datos que se tienen almacenados. En [7] se tiene como objetivo principal la creación de una aplicación que cree un archivo XML basado en un DTD, donde el contenido es aportado por el usuario.

Se transcriben documentos generados en la cámara del Parlamento de Andalucia, posteriormente se generar archivos XML y PDF. La estructuración inicial en forma DTD indicará al sistema la estructura que deberá usarse para que el usuario ingrese datos para cada uno de los documentos que se quieran tratar.

```
<!DOCTYPE biblioteca [
    <!ELEMENT biblioteca (libro+) >
    <!ELEMENT libro (titulo, autor+, editorial, paginas, capitulo+) >
        <!ATTLIST libro ISBN CDATA #REQUIRED>
    <!ELEMENT titulo (#PCDATA) >
    <!ELEMENT autor (nombreAutor) >
        <!ELEMENT nombreAutor (#PCDATA) >
    <!ELEMENT editorial (nombreEditorial, year, direccion) >
        <!ELEMENT nombreEditorial (#PCDATA) >
        <!ELEMENT year (#PCDATA) >
        <!ELEMENT direccion (#PCDATA) >
    <!ELEMENT paginas (#PCDATA) >
    <!ELEMENT capitulo ( seccion+ ) >
        <!ATTLIST capitulo nombre CDATA #REQUIRED>
        <!ELEMENT seccion (nombre, contenido) >
        <!ELEMENT nombre (#PCDATA) >
        <!ELEMENT contenido (#PCDATA) >
]>
```

Fig. 2. DTD Biblioteca.

Por medio de una interfaz dinámica adaptada a cada tipo de estructura, permitiendo al usuario crear el documento que se quiera tratar de acuerdo a la estructura establecida anteriormente por el DTD. Finalmente, el programa permitirá la exportación en formato XML de los datos que fueron introducidos, así como de igual manera en formato PDF respetando las reglas establecidas.

En [9] se tiene un sistema que facilitará a un profesional sin conocimientos informáticos, generar una serie de contenidos educativos sobre temas de su especialidad, los contenidos se producen en un formato abstracto y fácilmente manipulable, de modo que sean reutilizados en diferentes escenarios educativos e interfaces de usuario.

Estos formatos se basan en XML para los cursos y una serie de transformadores que permitan la distribución dosificada de cursos por email, servidores Web, CDROMS o papel. La información se representa por medio de un navegador web estándar, proporcionando al alumno un gran número de facilidades de navegación, búsqueda y autoevaluación.

En [3] se pretende estructurar la información de manera adecuada de datos obtenidos de la Facultad de Ciencias de la Computación en relación con los alumnos, profesores y materias de el área de Posgrado, construyen y analizan un diagrama de árbol para el diseño de archivos XML.

Con ayuda de XPath manipulan estos archivos, accediendo a su estructura iniciando en un nodo raíz, seguido de los hijos y las hojas del árbol, donde la principal funcionalidad del archivo XML es la de almacenar datos en vez de utilizar una base de datos. Para lograr la manipulación, implementan un sistema de software encargado del manejo de la información mediante archivos XML, donde se eliminan, insertan y modifican datos.

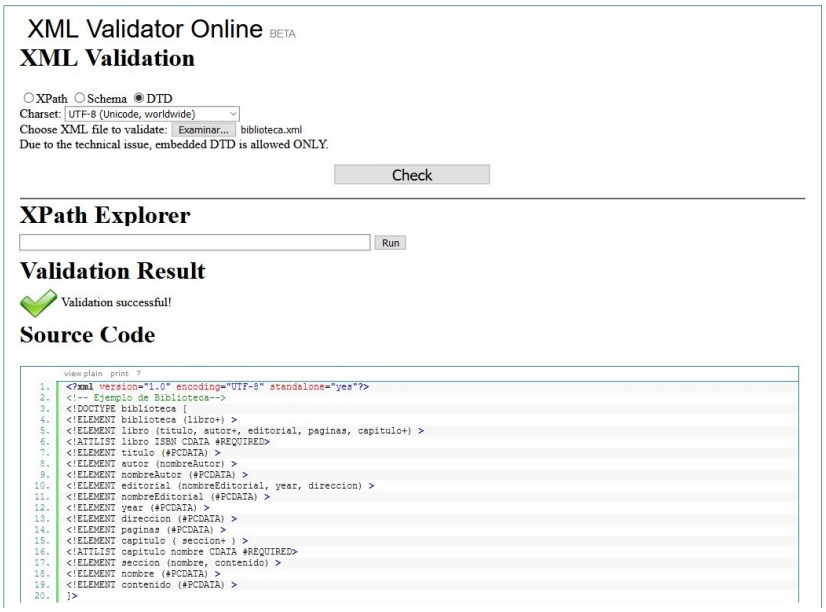


Fig. 3. DTD Biblioteca.

En [6] se diseña e implementa un sistema de gestión de contenidos aprovechando las propiedades que brinda XML para la gestión y manejo de contenidos de servicios de información de CITMATEL (empresa de tecnologías de la información), teniendo al XML como núcleo del proyecto y utilizándolo en lugar de cualquier base de datos tradicional. En [1] se desarrolla una aplicación en Android para la gestión de un documento XML, utilizando el método DOM.

Tiene como objetivo principal crear un sistema de ficheros XML que contengan los registros de clientes de una empresa, en la interfaz se pueden realizar diferentes opciones como editar, modificar o eliminar algún registro. En este trabajo se presenta una propuesta que permite administrar la información de los libros de una biblioteca por medio de una plataforma web, donde su núcleo es un archivo XML que contiene los datos.

Por medio del uso del API DOM y con la integración del lenguaje de programación PHP es como se pueden realizar altas, bajas, cambios y consultas de la información, con esto se pretende lograr una aportación importante en el área de manejo y recuperación de la información, además de que no solamente se puede implementar para bibliotecas, sino para cualquier otro tipo de proyecto que requiera control de información, como gestión de empleados, o inventarios, entre otros.

3. Diseño de la propuesta

El uso de archivos XML para este trabajo son de vital importancia, ya que son el núcleo para que se tenga un funcionamiento correcto al momento de querer realizar alguna acción con los datos que contenga, de tal manera que por medio de

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:element name="biblioteca">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="libro" type="libroType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="libroType">
        <xs:sequence>
            <xs:group ref="infoLibro" />
        </xs:sequence>
        <xs:attribute name='ISBN' type='xs:string'/>
    </xs:complexType>
    <xs:group name="infoLibro">
        <xs:sequence>
            <xs:element name="titulo" type="xs:string"/>
            <xs:element name="autor" type="autorType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="editorial" type="editorialType" minOccurs="0" maxOccurs="1"/>
            <xs:element name="paginas" type="xs:integer"/>
            <xs:element name="capitulo" type="capituloType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:group>
```

Fig. 4. XSD Biblioteca.

esta plataforma se logre tener un control de los datos representativos de elementos que pueden contenerse en una biblioteca. Se requiere en primera instancia crear una estructura para el control de elementos que pudieran existir en una biblioteca, en una primera fase de este proyecto, se contemplan solamente a libros. Para obtener la estructura y una buena organización de los datos se creó un diagrama de árbol, de tal manera que permita representar una jerarquía de los datos que pueda contener un libro.

Una vez teniendo el árbol ayudará a crear una representación de la estructura por medio de XML. Se pretende desarrollar una plataforma que ayude a gestionar y tener un control a modo de inventario de los libros que se puedan tener en una biblioteca, la plataforma está desarrollada con HTML5 para crear la estructura del sitio y sea visible en navegadores web. Se utiliza Boostrap 4 para que la interfaz visual sea llamativa y amigable para el usuario, siendo este un freamework de CCS3.

Se utiliza de igual manera un archivo XML en lugar de una base de datos tradicional para el almacenamiento de los datos, aprovechando las propiedades para acceder a los datos por medio del API DOM. Para hacer uso del API se integra el lenguaje de programación PHP7 para el manejo de los datos abstraídos por medio del API, lo que ayuda a la manipulación para el alta baja y cambios que se quieran realizar en algún elemento almacenado.

A continuación, se muestra toda la información detallada del proceso para la creación de la plataforma, iniciando por el diseño del árbol para la estructura de una biblioteca, aclarando que en esta primera fase se contemplan solamente libros. En la Figura 1, se puede apreciar el árbol propuesto donde se tiene como nodo raíz a Biblioteca y como único hijo a Libro, a su vez los hijos de Libro son Título, Autor, Editorial, Páginas y Capítulo.

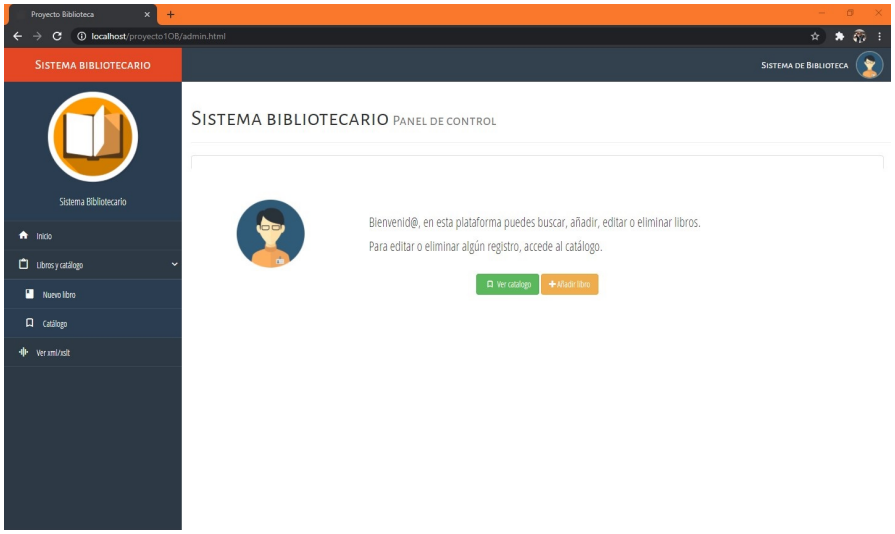


Fig. 5. Vista Dashboard.

De estos últimos Autor tiene como hijo a Nombre Autor, Editorial tiene como hijos a Nombre Editorial, Year (año), Dirección y Capítulo tiene como hijo a Sección, que a su vez tiene como hijos a Nombre y Contenido. Para comprobar la validez o que el XML este bien formado hay que validarlo, para esto se establecieron las reglas de construcción de documentos XML por medio de la definición de tipo de documento (DTD). De igual manera se creó la validación por medio de XSD, el cual nos permite describir la estructura y las restricciones de los contenidos de los documentos XML de manera más precisa, para no solamente quedarnos con una validación.

A continuación, se procede a describir el DTD y el XSD para este trabajo, mostrando de igual manera que las validaciones realizadas son satisfactorias con la ayuda de un par de herramientas de validación. Como se puede observar en la Figura 2 se define la raíz biblioteca que puede contener a su vez nodos hijos que serían libros, se específica que puede ser 1 o más. Posteriormente para cada nodo libro se definen sus hijos que serían título, autor, editorial, páginas y capítulo.

En el caso de autor tiene como hijos a NombreAutor, editorial tiene como hijos a nombreEditorial, year (año) y direccion, capítulo tiene como hijos a sección, se definen su tipo de dato y los atributos como el ISBN para identificar a cada libro y el nombre de capítulo en caso de querer identificarlos. Para validar el DTD se utilizó una herramienta web llamada XML Validator Online (ver Figura 3) [8].

En el caso de XSD, crece relativamente en cuanto a líneas de código, como se puede mostrar en la Figura 4, donde se define el esquema y a partir de ahí se define el elemento biblioteca que tendrá un tipo complex con una secuencia, en esta secuencia se define el elemento libro indicando que puede aparecer cero o muchas veces, este elemento libro es de tipo libroType, el cual es un tipo complex y se define un grupo para la información del libro que sería el título, autor, editorial, páginas y capítulo. En el caso de autor, editorial y capítulo se definen igual como tipo complex y se crean sus grupos para definir sus elementos anidados.

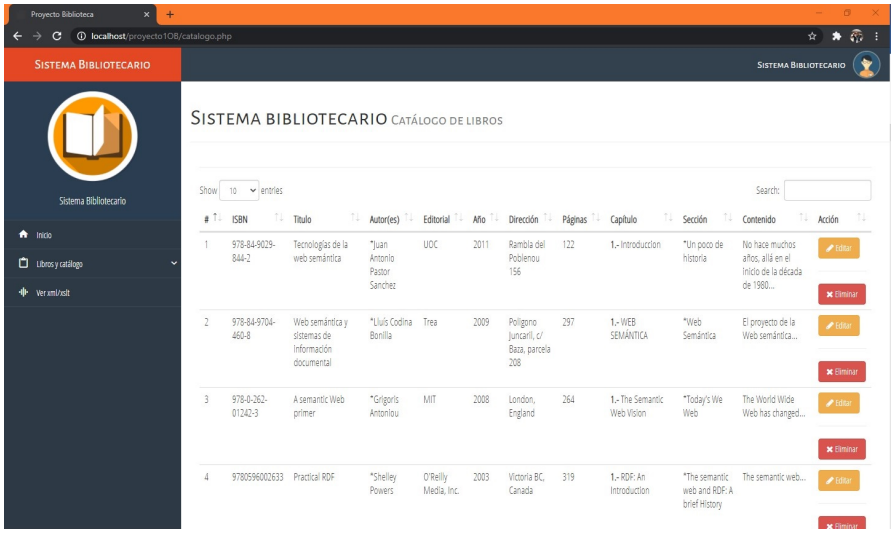


Fig. 6. Vista Dashboard.

4. Resultados

A partir de la estructura creada del XML, se realizó una interfaz de Plataforma para administrar los libros en una biblioteca. A continuación, se describe cada apartado de la plataforma con su respectivo funcionamiento. La plataforma cuenta con cuatro funciones básicas de cualquier sistema de gestión, que son altas, bajas, modificación y búsquedas. Para lograr estas funciones se realizaron adaptaciones de HTML5, PHP, Bootstrap 4, SimpleXmlElement y JQuery. A continuación, se detalla cada parte de la plataforma con las funcionalidades mencionadas anteriormente.

4.1. Dashboard (Inicio)

En la Figura 5 se puede observar el Dashboard o página principal del sitio, mostrando un menú lateral con los ítems correspondientes a Inicio, Libros y catálogo (Nuevo libro, Catálogo) y Ver XML/XSLT. El enlace de inicio nos mandará al Dashboard, Libros y catálogos tiene dos sub ítems, en el caso de Nuevo libro, nos abrirá una nueva vista con un formulario para añadir un nuevo libro. El sub item Catálogo nos mostrará una vista con una tabla que contiene todos los detalles de los libros registrados donde se podrá eliminar o editar algún elemento.

El item final Ver XML/XSLT nos abrirá una ventana con la vista del XML, pero en la versión de XSLT. Se tiene un header donde se muestra el nombre de la plataforma “Sistema Bibliotecario”. En la parte central se tiene el cuerpo del sitio, para el inicio de la plataforma como se muestra, aparecen indicaciones generales de lo que puede hacer el usuario, así como un par de botónes, el verde “Ver catálogo” mandará a la vista de los registros almacenados, mientras que “Añadir libro” nos mostrará la vista con el formulario para agregar un libro, esto a manera de que sean llamativos y de fácil acceso para el usuario, sin necesidad de usar el menú de navegación lateral.

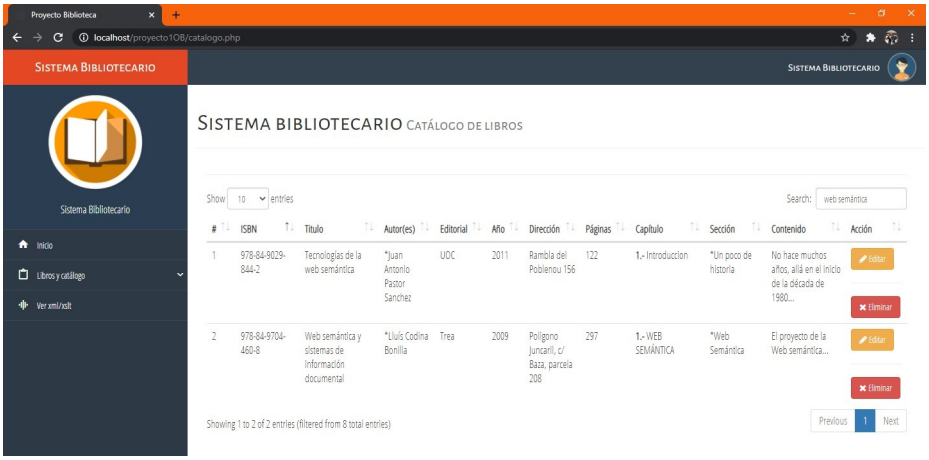


Fig. 7. Búsqueda: Web semántica.

4.2. Catálogo

En la Figura 6 se muestra la vista de Catálogo, donde se tiene en una tabla los registros que se extrajeron del XML mostrando el ISBN, Título, Autores, Editorial, etc. En la parte derecha por item se tienen dos botones por item en la tabla que son Editar y Eliminar. Editar nos llevará a una nueva vista donde se visualiza un formulario idéntico al de añadir libro, pero con los datos precargados del elemento a editar, mientras que el botón de eliminar, borrará el elemento de la plataforma y del XML.

La tabla donde se muestran los elementos, contiene en la cabecera iconos de flechas que representan filtros. Al dar clic en alguna flecha ordenará los elementos de la tabla de manera ascendente o descendente. En el lado derecho sobre la tabla, se tiene un buscador, al escribir alguna palabra mostrará únicamente los elementos que contengan la palabra que se ingresó en la consulta en cualquiera de las columnas de la tabla. Véase un ejemplo de consulta en la Figura 7, en la que se busca por la frase “web semántica”.

4.3. Nuevo Libro

En la Figura 8 se muestra la vista correspondiente a el alta de libros. La figura muestra un formulario vacío solicitando los datos para dar de alta el libro.

4.4. Editar

En la Figura 9, se muestra el formulario correspondiente a la edición de un elemento de los registros, a diferencia del alta, aquí en cada input (caja de texto), se muestran los datos que se tienen almacenados del registro a editar.

4.5. Eliminar

En la Figura 7 se muestra la vista del catálogo donde se explicó que aparecen los botones de editar y eliminar. Para eliminar un registro, basta con dar clic en un botón de eliminar y automáticamente borrará el elemento tanto de la plataforma como del XML.

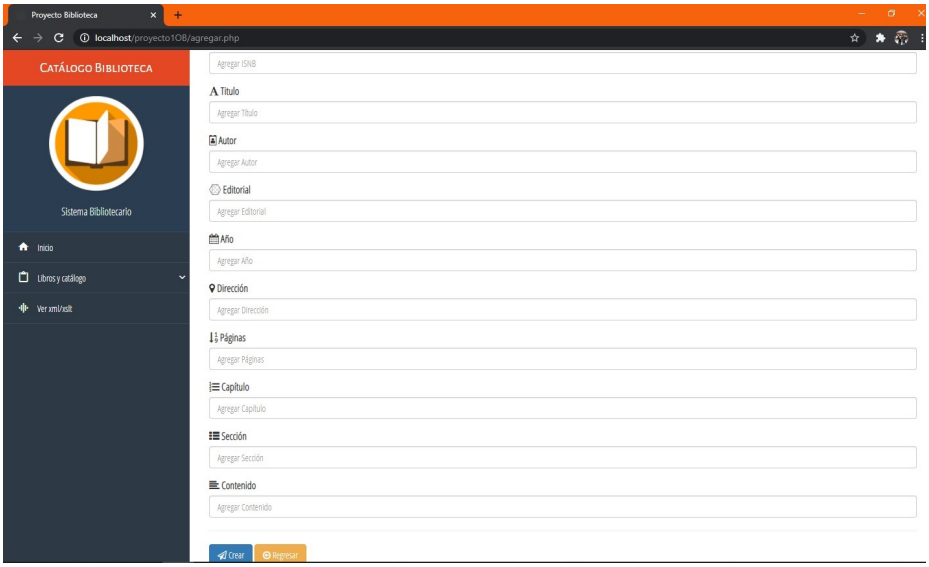


Fig. 8. Alta de Libro.

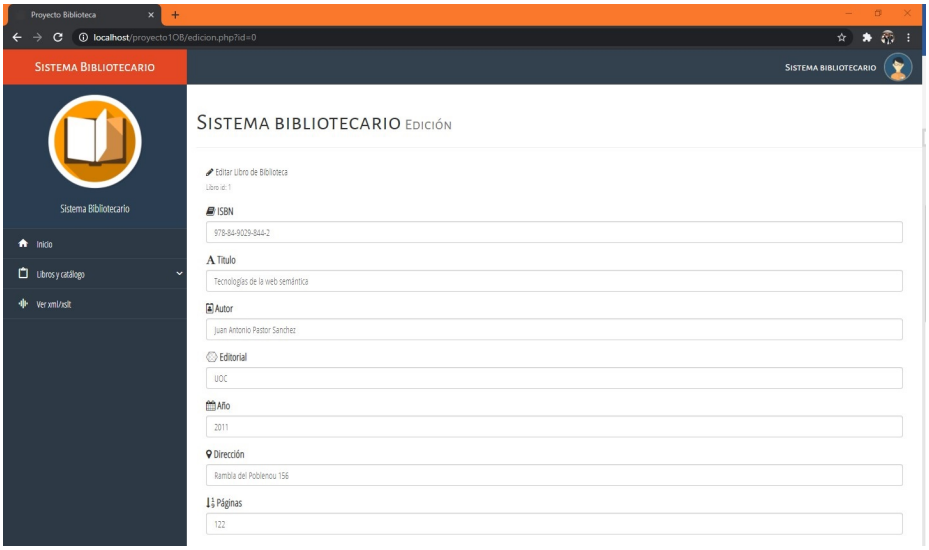


Fig. 9. Edición de un elemento.

Para mostrar los elementos se manda a llamar el archivo biblioteca1.xml, por medio de una función de la librería SimpleXmlElement. Posteriormente, con un foreach se recorren todos los objetos creados y se mandan a imprimir combinando código HTML con PHP. Para eliminar un elemento se espera un id que corresponde al elemento que se desea eliminar, por medio de un objeto creado se obtiene el elemento del XML y por medio de una función llamada unset se elimina ese elemento.

Colección de Libros de Biblioteca

ISBN	Titulo	Autor	Editorial	Año	Dirección	Paginas	Capítulo	Sección
978-0-262-01242-3	A semantic Web primer	Grigoris Antoniou	MIT	2008	London, England	264	-The Semantic Web Vision- -Structured Web Documents: XML-	-**Today's We Web**- -The World Wide Web has changed...- -**The XML Languag**- -Knowledge management concerns itself with acquiring...-
978-3-662-49995-5	Handbook on Ontologies...	Steffen Staab	SpringerLink	2009	Berlin Heidelberg	811	-Ontology Representation Languages- -Ontologies: Semantics for the Web -	-**What Is an Ontology?**- -The word "ontology" is used with different senses in different communities...- -**Description Logics**- -In this chapter, we explain what description logics are and why they make good ontology languages...-
978-3-598-25180-1	Knowledge Representation in the Social Semantic Web ...	Katrin Weller	De Gruyter	2010	West Sussex PO19 8SQ, England	442	-Knowledge Representation and Indexing: Background and Future- -Ontologies: Semantics for the Web -	-**Knowledge Representation** - -The term Knowledge representation is defined differently...- - **Ontologies: Definition and Background** - -The heterogeneous (scientific) communities dealing...-
978-1-84753-192-6	La Web Semántica...	Santiago Márquez Solís	SpringerLink	2007	Berlin Heidelberg	215	-Introducción- -Conceptos básicos sobre la Web actual -	-**Justificación y contexto**- -El desarrollo de Internet ha propiciado que prácticamente...- -**Los orígenes de la Web**- -La World Wide Web o simplemente Web como comúnmente se le denomina...-
978-84-9836-780-5	MANUAL DE WEB SEMANTICA	J. Grigoris Antoniou	COMARES	2011	Polígono Juncaril, c/ Baza, parcela 208	328	-VISIÓN DE LA WEB SEMÁNTICA- -DOCUMENTOS WEB CON ESTRUCTURA: EL LENGUAJE XML-	-**La Web actual**- -La Web actual...- -**El lenguaje XML**- -La representación de la información...-
1-4120-3448-5	OWL: Representing Information Using the Web Ontology Language	Lee W. Lacy	Trafford Publishing	2005	Victoria BC, Canada	264	-Semantic Web Overview- -Ontologies-	-**Current Web**- -In recent years we have...- -**Ontologies** - **Enablers of the Semantic Web**- -The semantic web explicit semantics...-

Fig. 10. Visualización XSLT Librería (Biblioteca).

Finalmente se guardan los elementos en el mismo archivo para guardar los cambios. Para guardar los elementos de la edición, se reciben inicialmente por medio de un formulario los datos que se desean cambiar por el método POST de HTML. Posteriormente se crea un objeto con la librería SimpleXmlElement y a cada objeto que hace referencia a cada nodo del XML se asignan los nuevos valores.

4.6. Ver XML/XSLT

En la Figura 10 se muestra la vista del XML con XSLT, en la plataforma basta con dar clic en el ítem del menú Ver XML/XSLT para que abra esta vista. Si se añadieron, editaron o eliminaron libros del registro se muestran de igual manera los datos actualizados.

5. Conclusiones

La información estructurada a través de XML ayuda a tener cierto orden y se ve de cierta manera estéticamente entendible. En la plataforma de sistema bibliotecario se puede acceder a los elementos del XML para editar, crear, borrar o consultar algún libro en el registro, lo cual pretende ser de ayuda para el personal que labore en alguna biblioteca y quiera realizar una especie de inventario. Cabe mencionar que no podemos comparar el uso de XML con un Gestor de Base de Datos en cuanto al tema del almacén de información, ya que en primera instancia depende total y completamente de la cantidad de datos a manipular y que tan complejo sea el sistema a querer desarrollarse. Continuando con las herramientas de la web semántica, otra alternativa de estructuración semántica de la información modelada es el formato OWL(Web Ontology Language) que se utiliza para el desarrollo de ontologías. La información que se uso es de libros que realmente existen, se hizo una búsqueda en la web para encontrar los libros, todos pertenecientes al área de la web semántica.

La creación de la estructura en XML para el almacén de los libros facilitó la manipulación de los datos, además de que por medio de PHP se logró la adaptación de XML con HTML para poder interactuar con los datos. En cuanto a los resultados que se obtuvieron con la implementación de la plataforma, estos son satisfactorios ya que se ha logrado crear una plataforma funcional a la altura de una plataforma que utiliza base de datos para el almacenamiento de la información.

Como trabajo a futuro se puede implementar y desarrollar la plataforma para que este más completa, manipulando distintos datos como usuarios, estantes, áreas u otras cosas que se puedan agregar o que pertenezcan a una biblioteca, se deberá modificar el árbol presentado en esta primera versión para lograr el cometido, además de la creación de otros modelos de representación semántica, como son las ontologías.

Referencias

1. Android Academy: Proyecto: Tratamiento XML en android utilizando DOM (2015) academiaandroid.com/proyecto-tratamiento-xml-en-android-dom-i/
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American, vol. 284, no. 5, pp. 34–43 (2001)
3. Cancino-Gordillo, J., Méndez-Sánchez, L., Valdez-Tovar, F., Tovar-Vidal, M.: Representación de información de una institución educativa a través de XML. Research in Computing Science, vol. 149, no. 8, pp. 867–878 (2020)
4. Echagüe-Burgos, J.: XML (Extensible Markup Language) (2014)
5. Ibermática.: XML, la solución ante la diversidad de formatos (2017)
6. Luaces-Torres, L., Menéndez-Senarega, A.: Sistema de gestión de contenidos XML. La Habana: Facultad de Ingeniería Industrial Centro de Estudios de Ingenieria y Sistemas (2006)
7. Romero-Vaquero, E.: Desarrollo de una aplicación para la transcripción de sesiones parlamentarias. Granada, Universidad de Granada (2015)
8. Studio, S. N.: XML Validator Online, Beta. Google Engine. xmlvalidator.new- studio.org/ (2009)
9. Sáes-Torres, A.: Herramientas para la definición y manipulación de materiales educativos basadas en XML (2003)
10. Tolsa, H., Bordignon, F.: Introducción a la recuperación de información. Universidad Nacional de Luján (2007)

ISSN 1870-4069

Hacia un semáforo de riesgo personal para sismos en la república mexicana

Rodrigo Vázquez Espino[1], Wulfrano Arturo Luna-Ramírez[1], Anasol Peña-Rios[2], Miguel Capllonch Juan[3]

[1] Universidad Autónoma Metropolitana-Cuajimalpa, División de Ciencias de la Comunicación y Diseño, Departamento de Tecnologías de la Información, México

[2] University of Essex, School of Computer Science and Electronic Engineering, Reino Unido

[3] Investigador independiente

mcjrcmc@gmail.com,acpena@essex.ac.uk, {2163030986,wluna}@cua.uam.mx

Resumen. Los desastres naturales son fenómenos meteorológicos que se producen en todo el mundo como resultado de la actividad de los elementos y la evolución del ambiente. Estos son catastróficos para todos los afectados, pero no todos los experimentan de igual manera. El semáforo de riesgo propuesto en este documento permite calcular, el índice de riesgo que presenta una persona ante un desastre natural a través de una aplicación web. Este prototipo considera como caso de estudio los eventos sísmicos. Este cálculo personalizado considera variables fisiológicas, sensoriales y de ubicación del usuario junto con los datos proporcionados por los sistemas oficiales de monitoreo ambiental. De esta forma, el semáforo de riesgos permite informar de manera visual y personalizada el riesgo que presenta el usuario, mediante tres colores diferenciados.

Palabras clave: Gestión de desastres naturales, aplicaciones web, integración de sistemas, sistemas de alerta temprana y gestión de riesgos.

Towards a Personal Risk Traffic Light System for Earthquakes in Mexico

Abstract. Natural disasters are meteorological phenomena that occur throughout the world as a result of the activity of the elements and the evolution of the environment. These are catastrophic for everyone affected but not everyone experiences them in the same way. The personal risk traffic light system proposed in this paper allows calculation of a person's risk index over a natural disaster through a web application. This prototype focuses on seismic events as a use case. The personalized calculation considers users' physiological, sensorial and location variables alongside data provided by official environmental monitoring systems. In this way, the risk traffic light reports the user's risk in a visual and personalized manner through three different colors.

Keywords: Natural disaster management, web applications, systems integration, early warning and disaster risk information systems.

1. Introducción

Históricamente México ha sido susceptible a una extensa cantidad de desastres naturales debido a que está situado entre cinco placas tectónicas, doce volcanes activos, dos océanos y ocurren abundantes lluvias durante todo el año, por lo tanto, es común que se presenten eventos como sismos, inundaciones, huracanes, deslaves, erupciones y tornados [1].

Para ilustrar lo anterior, cabe recordar que el 19 de septiembre de 2017 un sismo de magnitud 7.1 ocasionó la muerte de aproximadamente 369 personas y el desplome de 52 edificios, tan solo en la Ciudad de México. En el mismo día, pero 32 años antes, un terremoto de magnitud 8.1 ocasionó aproximadamente 40,000 pérdidas humanas [2]. La ciencia y tecnología no pueden impedir que ocurran estos sucesos naturales, pero aprovechando los datos que proporcionan los organismos de monitoreo oficiales de desastres naturales y climatológicos, se puede generar información valiosa para la seguridad de quien la utilice a través de un adecuado tratamiento de esta.

En la actualidad existen sistemas de alerta ante desastres naturales [3, 4], pero es difícil conocer con certeza qué tan vulnerable es cada usuario ante diversos eventos, ya que la finalidad de los sistemas de alerta es informar al mayor número de usuarios posibles. Esto conlleva a una falta de personalización de información. Si bien todos estamos expuestos a algunos de los desastres expresados con anterioridad, no significa que compartamos el mismo índice de riesgo ante dichos eventos.

Para poder generar información personalizada y que aporte un valor adicional a la emitida por los sistemas de alerta oficiales, se requiere realizar consultas, colectas análisis y almacenamiento de información oficial, información proveniente de fuentes de monitoreo ambiental y datos personales de las personas que utilicen este semáforo. Lo anterior implica mantener, adaptar o crear un formato para que los datos obtenidos permanezcan organizados y estructurados y así facilitar su almacenamiento y su análisis.

El proyecto aquí presentado, tiene el fin de ayudar a las personas a conocer su vulnerabilidad ante los peligros naturales a que están expuestos. Este prototipo calcula el índice de riesgo que presentan los desastres naturales a manera de simulación (con datos precargados de usuarios).

Se enfoca en el caso de los sismos, y toma en cuenta datos históricos de estos eventos, así como una serie de categorías para representar individuos con diferentes capacidades físicas y sensoriales.

2. Trabajos relacionados

Los semáforos de riesgo son instrumentos de comunicación y alertamiento que permiten definir la gravedad del suceso al que se encuentran expuestas las personas ante ciertos acontecimientos. Análogamente a un semáforo de tránsito, generalmente los niveles presentados por los semáforos de riesgo están divididos en 3 partes, cada una representada por un color que se asocia a su grado de riesgo.

Por ejemplo, el semáforo de alerta volcánica del Centro Nacional de Prevención de Desastres (CENAPRED) [5], divide el riesgo en tres colores cada uno con un número de fases, las cuales están ligadas a su actividad y al daño que puede ocasionar, cada una

Tabla 1. Comparativa entre Sistemas de Alerta para Monitoreo de Desastres y el Semáforo de Riesgo Personal (SRP) propuesto.

	PAGER	**ENS**	**ShakeCast**	**Cenapred**	**SSN**	**SGIRPC**	SRP
Alerta temprana ante la detección de un desastre	✓	✓			✓		
Puede ser consultado en la página web	✓	✓	✓	✓	✓	✓	✓
Cuenta con una aplicación móvil		✓		✓			✓
Considera la ubicación geográfica del usuario	✓	✓	✓	✓			✓
Considera limitaciones motrices del usuario							✓
Considera el número de pisos del inmueble donde se encuentra el usuario							✓

de estas fases requieren de diferentes acciones que deben de atender los habitantes de la zona volcánica. Las fases del semáforo en orden ascendente son verde, amarillo y rojo. El color verde denota la ausencia de actividad o presencia esporádica y leve de fumarolas y sismos, sin necesidad de evacuar.

El color amarillo representa una mayor presencia de actividad volcánica, sismos, exhalaciones de ceniza en dónde existe la posibilidad de evacuación así que se incita a la población a prepararse para salir de sus asentamientos. Finalmente, el color rojo representa la actividad altamente peligrosa del volcán y la evacuación para salvaguardar la vida de los habitantes.

Otro de los sistemas de alerta más importantes en México es el centro de monitoreo del Servicio Sismológico Nacional (SSN) de la Universidad Nacional Autónoma de México (UNAM), el cual provee de información histórica y de monitoreo en tiempo real de una red de 61 estaciones de monitoreo sísmico esparcidas en la República Mexicana.

El SSN crea reportes preliminares en caso de sismos y alerta al Centro de Comando, Control, Cómputo, Comunicaciones y Contacto Ciudadano de la Ciudad de México (C5) en caso de detectar un sismo peligroso con el fin de emitir la alerta sísmica en sus altavoces [4]. El SSN también cuenta con un canal RSS (Sindicación Realmente Simple por sus siglas en inglés) en el cual se encuentran los últimos 15 sismos detectados. Los dos ejemplos mencionados pertenecen a los sistemas de alerta nacional cuyo propósito es notificar a la mayor cantidad de personas ante un desastre.

A nivel personal existen sistemas privados que ofrecen un mayor nivel de personalización como SkyAlert, y SASSLA para los sismos, los cuales cuentan con

redes de monitoreo propias y son utilizados por medio de aplicaciones para teléfonos móviles.

Esto les permite utilizar la ubicación GPS (Sistema de Posicionamiento Global por sus siglas en inglés) del usuario para poder calcular el riesgo y el arribo aproximado del evento. Sin embargo, estos sistemas carecen de una orientación sobre las condiciones fisiológicas y sensoriales de sus usuarios, lo que las hace tener un carácter más generalista que no se ajusta a las necesidades particulares de una persona en una situación determinada.

Además, tales sistemas se centran en propagar las alertas en caso de sismos, y también carecen de un indicador tipo semáforo de riesgo, como el que aquí se presenta. Es importante señalar que hasta donde los autores conocen, no existe un semáforo de riesgo personal en el caso de sismos operando en la República Mexicana, ni a nivel internacional, pues si bien existen sistemas que miden el impacto de los sismos en poblaciones o localidades determinadas, estos esfuerzos parecen tener un mayor énfasis la diseminación de la alerta y las estimaciones de riesgos de índole económica y material a nivel de colectividades (incluso post-evento), dejando de lado al individuo y las situaciones personales de vulnerabilidad en sus valoraciones.

Ejemplos notorios son: el Sistema de Evaluación Rápida para Responder ante de Sismos Globales (PAGER por sus siglas en inglés) [6] que usa internamente un sistema de graduación de cuatro colores (verde, amarillo, anaranjado, y rojo); el Sistema de Notificación de Sismos (ENS por sus siglas en inglés), sistema de envío de alertas personalizadas sobre sismos que considera ciertas disposiciones geográficas definidas por el usuario [6, 7]; o ShakeCast [6, 8], un sistema para evaluación de situaciones post-evento que contrasta las condiciones de los inmuebles donde está el usuario, enfocado en la mitigación y respuesta tras los sismos.

La Tabla 1 muestra un comparativo entre los sistemas de alerta para monitoreo de desastres mencionados y la propuesta que aquí se presenta, resaltando las características de personalización indicadas anteriormente.

3. Modelado del semáforo

El semáforo opera con base en el cálculo del índice de riesgo personal (IR) de un determinado usuario ante un desastre natural. El IR se calcula a partir de dos componentes: el peligro detectado P y el grado de vulnerabilidad del usuario V.

En este trabajo se han desarrollado dos modelos de semáforo: 1) un modelo lineal, en el que P y V son ponderados por un factor de normalización y 2) un modelo polinomial.

3.1. Cálculo lineal

Inicialmente se desarrolló un modelo lineal para el semáforo. De manera simple, esta relación se muestra en la Eq. (1):

$$IR = (P + V) / FN, \quad (1)$$

donde:

IR = Índice de riesgo.

P = Peligro.

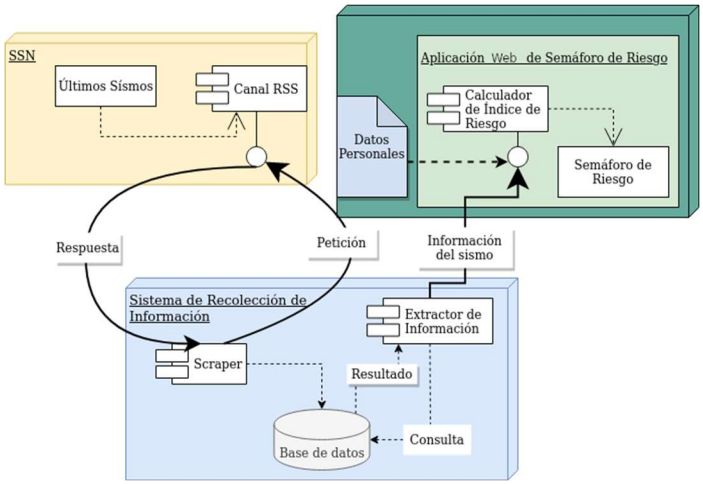


Fig. 1. Módulos de Procesamiento de Información.


```
−<rss version="2.0">
 −<channel>
    <title>Últimos sismos registrados por el SSN</title>
    <link>http://www.ssn.unam.mx</link>
    <description>Reporte de los últimos sismos en México</description>
   −<item>
    −<title>
       3.6, 49 km al SUROESTE de PINOTEPA NACIONAL, OAX
     </title>
    −<description>
       <p>Fecha:2020-10-29 16:54:39 (Hora de M&eacute:xico)<br/>Lat/Lon: 16.03/-98.37<br
       />Profundidad: 5.0 km </p>
     </description>
    −<link>
       http://www2.ssn.unam.mx:8080/jsp/localizacion-de-sismo.jsp?latitud=16.03&longitud=-98.37&
       prf=5.0 km&ma=3.6&fecha=2020-10-29&hora=16:54:39&loc=49 km al SUROESTE de
       PINOTEPA NACIONAL, OAX &evento=1
     </link>
     <geo:lat>16.03</geo:lat>
     <geo:long>-98.37</geo:long>
   </item>
   −<item>
    −<title>
       3.5, 57 km al SUROESTE de PINOTEPA NACIONAL, OAX
```


Fig. 2. Ejemplo de Datos de Sismo (SSN).

V = Vulnerabilidad del individuo.

FN = Factor de Normalización.

El factor de normalización tiene como propósito el equilibrar el peso del peligro y la vulnerabilidad, para este prototipo se considera que ambas variables tienen igual peso, por lo que este valor asume como 2.

3.2. Cálculo polinomial

El modelo lineal, que fue planteado como primera aproximación, presenta dos inconvenientes:

1. No contempla que incluso para valores muy altos de peligro, el índice de riesgo de una persona con vulnerabilidad mínima no es suficiente para generar una alerta y viceversa, personas con alta vulnerabilidad no serían alertadas en presencia de un peligro mínimo.

Tabla 2. Relación entre magnitud (n) y peligro del sismo (escala Mercalli).

Magnitud	Peligro
n < 4	1
4 ≤ n < 5	2
5 ≤ n < 6	4
6 ≤ n < 6.5	5
6.5 ≤ n < 7	6
7 ≤ n < 7.5	7
7.5 ≤ n < 8	8
8 ≤ n	10

Tabla 3. Vectores de características de vulnerabilidad.

V1	**V2**
V1 = [Mag, Slat, Slong]	V2 = [dfis, dsen, u, Plat, Plong]
Donde: Mag = Magnitud del Sismo Slat = Latitud del sismo Slong = Longitud del sismo	Donde: dfis = Discapacidades Físicas dsen = Discapacidades Sensoriales u = Ubicación de la persona en relación con el inmueble Plat = Latitud de la persona Plong = Longitud de la persona

2. Si se intenta solventar el problema anterior usando el modelo lineal cambiando FN, el modelo incurre en el error de generar índices de riesgo indebidamente bajos para personas en situaciones de suficiente peligro y vulnerabilidad.

Para resolver estos inconvenientes, se introdujo un modelo polinomial, según se define en la Eq. (2):

$$IR = a_0 + a_1P + a_2V + a_3P^2 + a_4V^2 + a_5PV, \quad (2)$$

donde:

$a_i$son coeficientes obtenidos por regresión contra un conjunto (P, V, IR) de casos hipotéticos o teóricos. Para el ajuste de estos parámetros en este trabajo, se escogieron solamente cuatro tuplas de valores (P, V, IR) de manera que estos reflejaran los casos ya mencionados del modelo lineal:

$p_1 = (P_{min} + 0.5, V_{max}, 5),$
$p_2 = (P_{max}, V_{min} + 0.5, 5),$
$p_3 = \left(P_{min} + 0.35P_{ptp}, V_{min} + 0.35V_{ptp}, 5\right),$
$p_4 = \left(P_{min} + 0.6P_{ptp}, V_{min} + 0.6V_{ptp}, 7\right),$

donde:

$P_{min} = 1,$
$P_{max} = 10,$
$V_{min} = 1.5,$
$V_{max} = 10,$

Tabla 4. Ponderación de distancia (d) entre epicentro del sismo y la persona.

Radio de distancia entre epicentro y persona (km)	Valor
d < 80	2.5
80 ≤ d < 140	2
140 ≤ d < 200	1.5
200 ≤ d < 300	1
300 ≤ d < 400	0.5
d ≥ 400	0

Tabla 5. Ponderación de la ubicación (u) de la persona al momento del sismo.

Ubicación	Valor
Del séptimo piso en adelante de un inmueble	2.5
Del cuarto al sexto piso de un inmueble	2
En el tercer o segundo piso de un inmueble	1.5
Planta baja de un inmueble	1
Zona despejada	0.5

$P_{ptp} = P_{max} - P_{min}$,
$V_{ptp} = V_{max} - V_{min}$.
Los valores obtenidos para los coeficientes usando estos puntos son:

$$(a_0, a_1, a_2, a_3, a_4, a_5) = (0.23698655, 0.59396298, 0.63429914, -0.02897832, -0.02830087, 0.02829105).$$

3.3. Cálculo de la vulnerabilidad

Para calcular V, considerando que el semáforo toma como caso de estudio los eventos sísmicos, se toman en cuenta distintos aspectos de vulnerabilidad del usuario durante tales situaciones. El prototipo de semáforo de riesgo personal propuesto en este artículo se compone de dos módulos para el procesamiento de información (Fig. 1), cada uno de estos módulos se encarga de procesar datos de diferente naturaleza y fuente para poder realizar el cálculo del IR. Los módulos son:

- Sistema de Recolección de Información. Se encarga de consultar los registros más recientes de desastres y fenómenos naturales.
- Aplicación Web de Semáforo de Riesgos. Consulta los datos personales de los usuarios para poder calcular su vulnerabilidad en conjunto con los datos recopilados por el primer módulo, de esta manera el índice de riesgo puede ser calculado y mostrado a través de una página web.

3.4. Sistema de recolección de información

El Sistema de Recolección de Información proporciona uno de los insumos para realizar la estimación del color del semáforo de riesgo personal que se le presenta a un

Tabla 6. Ponderación de capacidades físicas de la persona (dfis).

Capacidades Físicas	Valor
No puede desplazarse.	2.5
Depende de la ayuda de otras personas para desplazarse	2
Utiliza silla de ruedas o muletas para desplazarse	1.5
Puede moverse con ligeras complicaciones, sin necesidad de ayuda	1
Puede evacuar en caso de emergencia sin ayuda	0.5

Tabla 7a. Ponderación de capacidades sensoriales de la persona (dsen).

Capacidades Sensoriales	Valor
No puede escuchar alarmas ni visualizar rutas de evacuación	2.5
Se le dificulta ver y/o escuchar alarmas e indicaciones de evacuación	1.5
Puede escuchar indicaciones y alarmas claramente y observar rutas de evacuación	0.5

Tabla 8. Ponderación de colores de semáforo de riesgo.

Índice de Riesgo	Color
$IR \leq 5$	Verde
$5 < IR < 7$	Amarillo
$IR \geq 7$	Rojo

usuario. Para realizar esta estimación, el Sistema debe proporcionar datos del peligro que representa un evento sísmico. De esta forma, se obtienen los datos de los últimos sismos ocurridos en México, cuya información se obtiene del canal RSS que el Servicio Sismológico Nacional pone a disposición en su sitio Web. Esta información se consulta de manera automática y continua, para tener un pequeño acervo de datos actualizados de los últimos 15 sismos reportados.

El acervo conserva el formato usado por el Sismológico Nacional (XML). En la Fig. 2 se muestra un ejemplo de la consulta de los datos de un sismo provenientes del RSS del Sismológico Nacional.

El Sistema de Recolección de Información, implementa un extractor de información (conocido comúnmente como *data scraper*) implementado en el lenguaje `Python`, que hace uso de las bibliotecas `Pandas` y `Geocoder`.

3.5. Aplicación Web de semáforo de riesgos

En el prototipo que aquí se presenta, se utilizaron los datos recabados de sismos con diferentes magnitudes y epicentros (Sección 3.1), los cuales se utilizan como datos de entrada para calcular el IR ante un sismo. Los datos personales se simulan tomando como punto de partida algunas de las ciudades más vulnerables ante sismos en México para representar la ubicación de los usuarios [1].

Así mismo, se utilizaron diferentes valores para representar individuos con distintas capacidades físicas y sensoriales (Tablas 6 y 7) para determinar estos valores se consideró la información sobre vulnerabilidad física descrita en [8]. Como ya se indicó, el cálculo del IR implica conocer el peligro P y la vulnerabilidad presentada V.

Tabla 7b Caso 1 Semáforo en Verde.

V1	V2	P	V	IR_Lin	Color	IR_Pol	Color
{7.8, 19.51, -104.45}	{0.5, 0.5, 1.0, 19.394068, -99.096056}	5	2.0	3.5	Verde	3.920649	Verde

Tabla 9. Caso 2 Semáforo en Amarillo.

V1	V2	P	V	IR_Lin	Color	IR_Pol	Color
{7.8, 19.51, -104.45}	{1.0, 1.5, 1.5, 21.125008, -101.685959}	5	4.5	4.75	Verde	5.400146	Amarillo

Tabla 10. Caso 3 Semáforo en Rojo.

V1	V2	P	V	IR_Lin	Color	IR_Pol	Color
{7.8, 19.51, -104.45}	{2.0, 2.5, 2.5, 19.596019, -103.754735}	5	9.5	7.25	Rojo	7.297857	Rojo

Para el cálculo de P se propone una relación entre magnitud y peligro del sismo inspirada en la escala Mercalli [6-8]. Esta relación puede apreciarse en la Tabla 2.

Para el cálculo de V, se crean dos vectores de características que representan tanto los datos obtenidos del sistema de monitoreo sísmico como los datos personales del usuario (Tabla 3). Específicamente:

- V1: representa los valores de magnitud, latitud y longitud asociados a un sismo.
- V2: características fisiológicas y sensoriales, así como la ubicación del usuario al momento del sismo.

El cálculo de V se consideró tomando en cuenta los siguientes factores:

- La distancia (d) entre el usuario (Plat, Plong) y el epicentro del sismo (Slat, Slong), usando la fórmula del semiverseno (Eq. 3).
- Su ubicación (u) dentro de un inmueble.
- Su capacidad para desplazarse (dfis).
- Su capacidad visual y auditiva (dsen).

Estos factores se ponderan de acuerdo con una graduación que estima la vulnerabilidad, y cada uno de ellos se representa como un número entero, lo cual conforma un vector de vulnerabilidades, es decir V. Las ponderaciones se muestran en las Tablas 4, 5, 6 y 7.

La fórmula del semiverseno (Harvesine) calcula la distancia entre dos posiciones considerando la curvatura terrestre. El semiverseno se define en la Eq. 3:

$$a = sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cdot \cos(lat2) \cdot sin^2\left(\frac{\Delta long}{2}\right),$$

$$c = 2 \cdot atan2\left(\sqrt{a}, \sqrt{1-a}\right), \tag{3}$$

$$d = R \cdot c\ ,$$



Fig. 3a. Operación del semáforo en un evento.

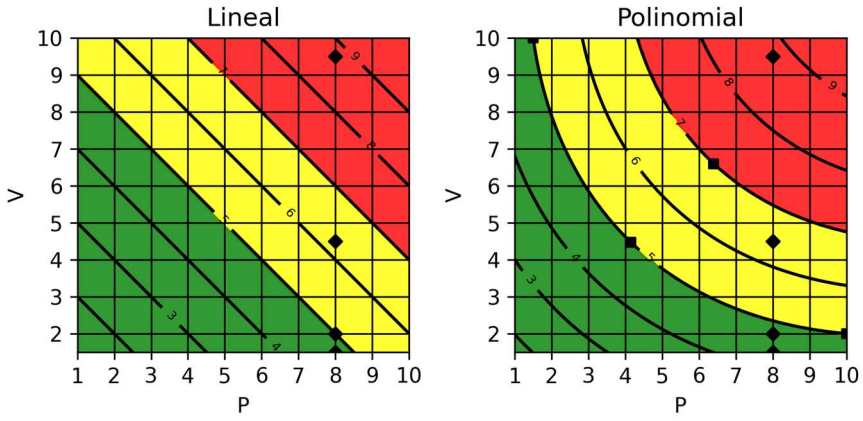


Fig. 3b. Comparación de los resultados de IR para una malla de valores (P, V) entre los modelos lineal (izquierda) y polinomial (derecha) para los casos de estudio (representados por rombos). En la figura del modelo polinomial se muestran los puntos hipotéticos usados para ajustar los parámetros (cuadrados).

donde:

$\Delta lat = lat2 = lat1,$
$\Delta long = long2 - long1,$

R = el radio de la Tierra (promedio r = 6,371 km).

La Tabla 4 corresponde a la ponderación propuesta para el cálculo del valor de distancia. Se toma como referencia lo expuesto en [10, 11, 12] para estimar que la

máxima vulnerabilidad se presenta a una distancia de menos de 100 km del epicentro, y a medida que la distancia crece la vulnerabilidad decrece.

La Tabla 5 toma como referencia lo expuesto en [13], la ubicación de la persona dentro de un inmueble ya que solo los primeros 2 pisos de un inmueble podrán evacuar durante el sismo, también indica que el tiempo máximo para evacuar un inmueble es de 40 segundos, por lo que la movilidad de la persona es importante.

En la Tabla 6 se proponen los valores asignados al impacto que tiene las capacidades motrices de la persona.

Análogamente, en la Tabla 7a se hace una estimación acerca de sus capacidades sensoriales frente a un evento sísmico.

A partir de las ponderaciones recopiladas en V, y de la estimación de P que representa un sismo, se aplica la Eq. 1, con lo que se obtiene el IR. Lo que prosigue es determinar el color del semáforo a ser presentado al usuario. Dicho color indicaría el riesgo que el evento sísmico representa para un usuario en un momento determinado, conforme sus características personales y las del evento. Para representar los colores del semáforo, se utiliza la ponderación presentada en la Tabla 8.

Para poder apreciar mejor el cálculo del IR, a continuación, se presentan casos de ejemplo de los vectores de información, su IR resultante y su color de semáforo. Estos casos están representados en la Fig. 3b para el semáforo usando los cálculos lineal y polinomial.

Caso 1. Sismo de magnitud 6.2, detectado en La Resolana, Jalisco, el individuo se encuentra en la planta baja de un inmueble en Iztacalco en la Ciudad de México, a 561 km de distancia del epicentro. La persona puede moverse libremente y puede ver y escuchar instrucciones de evacuación.

Caso 2. Sismo de magnitud 6.2, detectado en La Resolana, Jalisco, el individuo se encuentra en el tercer piso de un edificio en León, Guanajuato, a 340 km de distancia del epicentro. La persona puede moverse con ligeras complicaciones y se le dificulta ver indicaciones de evacuación.

Caso 3. Sismo de magnitud 6.2, detectado en La Resolana, Jalisco, el usuario se encuentra en un séptimo piso en Alista, Jalisco, a 73 km de distancia del epicentro. Depende de la ayuda de otras personas para desplazarse, y no puede escuchar instrucciones de evacuación.

El prototipo aquí presentado, muestra el semáforo personal de manera gráfica vía una página web, implementada en el manejador de contenidos `Django`[1]. La página despliega el semáforo personal para un usuario ante un sismo determinado, así como información sobre el evento: datos del sismo, datos fisiológicos de la persona, y su IR, que determina el color actual del semáforo para un evento.

Un ejemplo del semáforo operando se ilustra en la Fig. 3a, en ella, se puede ver un caso donde el semáforo marca el color verde para una medición particular. Se muestran además los datos asociados al cálculo del IR y el color del semáforo para ese evento (ver las tablas de valores de referencia indicadas):

- Magnitud del sismo detectado en la escala de Mercalli (Tabla 3).
- Datos de la vulnerabilidad del usuario:

1 Proyecto Django - https://www.djangoproject.com/

- Usuario con autonomía suficiente para poder evacuar un edificio en caso de emergencia (Tabla 6).
- Capacidad de escucha de indicaciones y alarmas y visualización de letreros de rutas de evacuación (Tabla 7b).
- El usuario se ubica en una planta baja (Tabla 5).
- El IR calculado.

Además, para una mejor representación de la situación, el semáforo proporciona un mapa del epicentro para ubicar al usuario en el evento sísmico (Tabla 4).

4. Conclusiones y trabajo a futuro

En este artículo se presenta el prototipo de un semáforo de riesgo personal en caso de desastres naturales, que toma como caso de estudio los eventos sísmicos en la República Mexicana, por la importancia que estos tienen para el país. Si bien existen semáforos y sistemas de alerta personalizados, este trabajo se diferencia de ellos en que hace una caracterización más profunda de las condiciones del usuario que aquellas reportadas en la literatura.

Toma como base la determinación del índice de riesgo personal, que considera tanto una estimación del peligro que representa un evento de desastre natural (en este caso sismos), así como las condiciones fisiológicas y sensoriales de un usuario determinado durante tales eventos. De esta forma, se logra una mejor determinación del riesgo personal, a través del índice.

El prototipo toma información de sismos provenientes de medios oficiales (Servicio Sismológico Nacional) e información simulada de usuarios para realizar el cálculo del índice de riesgo personal, mismo que se muestra gráficamente en una página Web junto con información del evento y del cálculo.

El prototipo continúa su desarrollo, y se tienen identificadas las siguientes tareas como trabajo a futuro:

- Refinar el cálculo del riesgo personal. Ahora, el índice de riesgo se calcula mediante dos aproximaciones, una lineal y una polinomial. Hay dos opciones a considerar para continuar con su estudio:
  - Uso de técnicas de Aprendizaje Automático para adecuar el cálculo de los coeficientes polinomiales.
  - La representación de valores de pertenencia a las clases que representan los niveles de riesgo, peligro y vulnerabilidad utilizando lógica difusa.
- Tomar datos de usuarios en tiempo real, relativos a sus características de vulnerabilidad, por lo que se plantea su implementación en dispositivos móviles que recolectan los datos requeridos.
- Evaluación del riesgo personal con datos de usuarios en tiempo real.
- Evaluación de la usabilidad del semáforo con una población de muestra con distintas características.

Finalmente, se planea extender el semáforo de riesgo personal a otras instancias de desastres naturales relevantes para México y sus distintas localidades, tales como inundaciones, huracanes, entre otros.

Referencias

1. CENAPRED.: Diagnóstico de peligros e identificación de riesgos de desastres en México. Atlas Nacional de Riesgos de la República Mexicana, vol. 1, no. 4 (2014) http://www.cenapred.gob.mx/es/Publicaciones/archivos/36-DIAGNSTICODEPELIGROSEIDENTIFICACINDERIESGOSDEDESASTRESENMXICO.PDF
2. Quezada-Reyes, A.: El Sismo de 1985 en cifras. Servicio Sismológico Nacional, https://web.archive.org/web/20080408054816/http://www.ssn.unam.mx/website/jsp/Carteles/sismo85.jsp
3. Luna-Ramirez, W. A., Peña-Ríos, A., Ferreyra, E., Ortiz-Hernandez, G., Jaimez-Gonzalez, C. R.: Hacia un sistema de seguridad y alerta temprana de riesgos basado en agentes cognitivos. Research in Computing Science, Special Issue on "Intelligent Systems and Optimization", México: Instituto Politécnico Nacional (IPN), vol. 147, no. 8, pp. 135–148 (2018)
4. Cuéllar, A., Espinosa-Aranda, J., Suárez, R., Ibarrola, G., Uribe, A., Rodríguez, F., Islas, R., Rodríguez, G., García, A., Frontana, B.: The mexican seismic alert system (SASMEX): Its alert signals, broadcast results and performance during the M7.4 Punta Maldonado earthquake of march 20th, 2012. In Early warning for geological disasters, Springer, Berlin, Heidelberg, pp. 71–87 (2014) doi: 10.1007/978-3-642-12233-0_4
5. Centro nacional de prevención de desastres. https://www.gob.mx/cenapred
6. Wald, D. J., Jaiswal, K. S., Marano, K. D., Bausch, D.: Earthquake impact scale. Natural Hazards Review, vol. 12, no. 3, (2011) doi: 10.1061/(ASCE)NH.1527-6996.0000040
7. Wald, L. A., Wald, D. J., Schwarz, S., Presgrave, B., Earle, P. S., Martinez, E., Oppenheimer, D.: The USGS earthquake notification service (ENS): Customizable notifications of earthquakes around the globe. Seismological Research Letters, vol. 71, no. 1, pp. 103–110 (2008) doi: 10.1785/gssrl.79.1.103
8. Wald, D., Lin, K. W., Porter, K., Turner, L.: Shakecast: Automating and improving the use of shakemap for post-earthquake decision-making and response. Earthquake Spectra, vol. 24, no. 2, pp. 533–553 (2008b) doi: 10.1193/1.2923924
9. Sitio Web del Servicio Sismológico Nacional. http://www.ssn.unam.mx/
10. Sitio web del servicio SkyAlert. https://skyalert.mx/
11. Sitio web del servicio SASSLA. https://www.sassla.mx/
12. Renda, E., RoZAS-Garay, M., Moscardini, O.: Manual para la elaboración de mapas de riesgo (1st ed.). Programa naciones unidas para el desarrollo PNUD argentina y ministerio de seguridad de la nación de argentina. (2017) https://drive.google.com/file/d/0BwHRtG8cxrk_blFjZndlOWg3eDQ/view
13. Servicio Geológico Mexicano. Escalas de los sismos (2017) https://www.sgm.gob.mx/Web/MuseoVirtual/Informacion_complementaria/Escalassismos.html
14. Anexo 5.1.5-a. Bases técnicas para el cálculo de la prima de riesgo y la pérdida máxima probable de los seguros de terremoto de la circular única de seguros y fianzas de 2015. Comisión Nacional de Seguros y Fianzas, Decreto por el que se expide la Ley de Instituciones de Seguros y de Fianzas y se reforman y adicionan diversas disposiciones de la Ley sobre el Contrato de Seguro. 4 de abril de 2013, http://www.cnsf.gob.mx/CUSFELECTRONICA/CUSF/Index y http://www.cnsf.gob.mx/CUSFELECTRONICA/CUSF/A_5_1_5_A

    García-Soto, A. D., Pozos-Estrada, A., Hong, H., Gómez-Martínez, R.: Estimación del peligro sísmico debido a sismos interplaca e inslab y sus implicaciones en el diseño sísmico.

Revista de Ingeniería Sísmica, vol. 86, pp. 27-54 (2012) [fecha de Consulta 15 de Abril de 2021]. ISSN: 0185-092X. https://www.redalyc.org/articulo.oa?id=61824407002

15. García, D., Singh, S., Herraiz, M., Ordaz, M., Pacheco, J.: Inslab earthquakes of central Mexico: Peak ground-motion parameters and response spectra. Bulletin of The Seismological Society of America - BULL SEISMOL SOC AMER, vol. 95, pp. 2272–2282 (2005) doi: 10.1785/0120050072
16. Manual de protección civil ante casos de emergencia, contingencias y desastres del STCONAPRA. Secretaría de Salud/STCONAPRA (2017) https://www.gob.mx/cms/uploads/attachment/file/252641/Manual_de_Protecci_n_Civil_STCONAPRA.pdf

ISSN 1870-4069

Propuesta de distribución y agrupamiento de Centros de Salud en el Estado de Aguascalientes basado en algoritmo K-Means modificado considerando la patología de Diabetes Mellitus

Pablo Rodríguez de León[1], María Dolores Torres Soto[2], Aurora Torres Soto[2]

[1] Posgrado e Investigación, Aguascalientes, México

[2] Benemérita Universidad Autónoma de Aguascalientes, México

pablo.rdz@live.com, {mdtorres, atorres}@correo.uaa.mx

Resumen. Este artículo presenta la propuesta de rediseño de la distribución de unidades de salud del Instituto de Servicios de Salud del Estado de Aguascalientes (ISSEA) en jurisdicciones cuyo objetivo, es la reducción de las distancias de un centro a su jurisdicción y mejorar la atención. Actualmente, el ISSEA cuenta con 89 centros de salud agrupados en 3 Jurisdicciones, esta distribución tiene el problema de no estar organizada en base a distancias de los centros de salud con respecto de la ubicación de su oficina de jurisdicción responsable, esta configuración toma en cuenta el municipio de pertenencia de la unidad para determinar la Jurisdicción responsable. La propuesta de redistribución presentada en este trabajo utilizó la ubicación de las unidades de salud en base a su longitud y latitud geográficas, y como resultado del análisis se propone incluir una jurisdicción 4 que atendería a 14 centros de salud, los cuales captan más de 65,000 pacientes. Los resultados fueron obtenidos mediante la implementación de una versión modificada del algoritmo K-means, con centroides fijos y variables, manejando distancias calculadas por medio de semiverseno o fórmula de Haversine. Se determinó la ubicación de la jurisdicción 4, se obtuvo el agrupamiento de las unidades de salud en 4 grupos (uno por jurisdicción), obteniendo una mejor distribución de las unidades de salud, así como de los pacientes pertenecientes a cada una de éstas.

Palabras clave: K-means modificado, Centroides fijos y variables, Distancia Haversine.

Proposal for Distribution and Grouping of Health Centers in the State of Aguascalientes based on the K-Means Modified Algorithm Considering the Pathology of Diabetes Mellitus

Abstract. This paper presents the proposal for the redesign of the distribution of health units of the Institute of Health Services of the State of Aguascalientes. (ISSEA) in jurisdictions whose objective is to reduce distances and improve care.

The ISSEA has 89 health centers grouped in 3 jurisdictions, this distribution has the problem of not being organized based on the distances of the health centers with respect to the location of its office of responsible jurisdiction, this configuration considers the municipality belonging to the unit to determine the responsible jurisdiction. The redistribution proposal presented in this work used the location of the health units based on their geographic longitude and latitude, and because of the analysis it is proposed to include a jurisdiction 4 that will serve 14 health centers which attract more than 65,000 patients. The results were obtained by implementing a modified version of the K-means algorithm, with fixed and variable centroids, handling distances calculated by means of the semiversin or Haversine formula. The location of jurisdiction 4 was determined, the health units were grouped into 4 groups (one per jurisdiction), obtaining a better distribution of the health units, as well as the patients belonging to each of them.

Keywords: K-means, Fixed and variable centroids, Haversine distance.

1. Introducción

En la actualidad, el ISSEA cuenta con 89 centros de salud distribuidos en sus 11 municipios y con 3 oficinas jurisdiccionales[1], las cuales fueron ubicadas en los municipios de Aguascalientes, Rincón de Romos y Calvillo. Esta ubicación data de principios de los años 90. Las jurisdicciones cuentan con oficinas administrativas, almacenes, centros de distribución, facilidades de comunicación, etc. Su principal tarea es la de administrar y monitorear al personal, administrar y distribuir insumos, equipo y medicamentos a los centros de salud.

La propuesta que se presenta en este trabajo, reduce a la mitad la distancia máxima de cada centro de salud respecto de su jurisdicción de responsabilidad, ya que actualmente la distancia máxima de un centro de salud con respecto a la jurisdicción de su responsabilidad es de 54km lineales y 70 km por carretera.

Esta investigación cuenta con la información de ubicación (longitud y latitud) de 89 centros de salud y 3 jurisdicciones, también se tiene información del total de pacientes pertenecientes a cada unidad de salud; así como información importante de pacientes catalogados como probables y confirmados con respecto de la patología de diabetes mellitus.

Esta información fue recolectada mediante un proceso automatizado por medio de servidores MS SQL Server[2]. Este trabajo incluye la implementación de un algoritmo K-means modificado, que trabaja con distancias Haversine (que calcula la distancia entre dos puntos con la fórmula de Haversine determinando la distancia de círculo máximo entre dos puntos de una esfera dadas su longitud y latitud).

Dicho algoritmo trabaja con 4 centroides, de los cuales, tres son centroides fijos y uno variable.

Los centroides fijos usan las coordenadas de latitud y longitud de las tres oficinas jurisdiccionales existentes actualmente en ISSEA y el centroide variable representa la nueva jurisdicción.

Mediante el algoritmo K-means modificado, se generó la ubicación de la jurisdicción 4 en esta jurisdicción se agruparon 14 centros de salud de los cuales 6 pertenecían a la jurisdicción 1 y 8 pertenecían a la jurisdicción 2. Esta nueva jurisdicción 4 puede

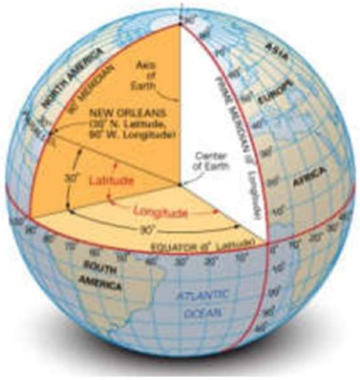


Fig. 1. El hemisferio muestra la diferencia entre las líneas de latitud y longitud [7].

atender a 65,568 pacientes de los cuales 26,336 eran de la jurisdicción 1 y 39,232 de la jurisdicción 2, respecto de la patología de diabetes mellitus. La jurisdicción 4 atenderá a 41,213 pacientes confirmados con la patología.

2. Conceptos relacionados

2.1. K-means

Este algoritmo fue creado por MacQueen en 1967 y es el algoritmo de agrupación más famoso y empleado ya que a pesar de su simpleza es eficaz. La agrupación permite el reconocimiento de tipologías o grupos en los cuales los elementos comparten gran similitud entre ellos y muchas diferencias con los elementos de los de otros grupos [3].

K-means lleva a cabo un procedimiento de clasificación de un conjunto de elementos en un determinado número K de clústeres o grupos, anticipadamente definido. Se le da el nombre de K-means o K–medias debido a que representa cada uno de los grupos por la media de los puntos de los elementos, o dicho de otra manera por sus centroides. La interpretación por medio de centroides tiene la utilidad de que es un concepto estadístico y gráfico reconocido. Un clúster por lo tanto es determinado por su centroide el cual se encuentra justo en el centro de los elementos que componen el clúster[4].

Algunos autores como Cifuentes Ramos [5] dividen la técnica de K-means en las siguientes 4 etapas:

1 Se eligen al azar los K elementos que conforman los K clúster o grupos iníciales. El valor inicial del centro para cada clúster k es X_i, y hasta este momento este sería el único objeto perteneciente al clúster o grupo.

2 Se debe asignar los elementos al clúster. Cada elemento se asigna al objeto más cercano, tomando en cuenta una medida de distancia.

3 Una vez que todos los elementos son situados, se calculan los centroides de K clústeres. Estos nuevos clústeres representan la media del total de elementos designados al clúster.

4 Volver a hacer las etapas 2 y 3 hasta que no se tenga diferencias en reasignaciones.

El algoritmo siempre finaliza, pero eso no garantiza el llegar a la solución óptima. Se debe tomar en cuenta que para tener mejores resultados se debe ejecutar el algoritmo varias veces para minimizar la sensibilidad a la elección aleatoria de K centroides iníciales.

2.2. Fórmula de Haversine

La fórmula de Haversine es una ecuación importante en la navegación, que proporciona la distancia del círculo máximo entre dos puntos en una esfera, partiendo de su longitud y latitud [6].

El cálculo de la distancia de un punto a otro en la superficie de la tierra se ve afectado por la curvatura de ésta. La fórmula de Haversine es un método para calcular la distancia estimada de manera adecuada y precisa de puntos geográficos [7], a continuación en la fórmula 1 se muestra el cálculo de la distancia Haversine:

$$d = 2r.arcsen\sqrt{sen^2\frac{\theta_2-\theta_1}{2} + \cos\theta_1 .\cos 2 .sen^2\frac{\varphi_2-\varphi_1}{2}}, \quad (1)$$

donde

$r = 6271\ km$

θ es la latitud

φ es la longitud

La longitud es una línea que mide distancia entre el oeste y este de la tierra desde la línea del meridiano principal.

La latitud es la línea que mide la distancia entre el norte y el sur de la Tierra frente a la línea del ecuador. En la siguiente figura 1 podemos ver la situación entre las líneas de latitud y longitud.

Google Maps platform tiene disponible una función en su javaAPI para el cálculo de distancias de un punto a otro, en su página menciona textualmente que:

Con Google Maps platform, se puede eliminar algunas conjeturas de la imagen cuantificando distancias a través de la distancia en línea recta y la distancia de la ruta. Cada uno utiliza un enfoque diferente y resuelve diferentes problemas o acciones de los usuarios.

El método más simple para calcular la distancia se basa en matemáticas avanzadas. Conocida como la fórmula de Haversine, utiliza trigonometría esférica para determinar la distancia del gran círculo entre dos puntos. Usando el equivalente de JavaScript de la fórmula de Haversine, podemos determinar la longitud de la polilínea y la distancia recta entre dos marcadores" [8].

La empresa Here We Go, especialista en el desarrollo de aplicaciones de navegación y mapas, publicó en su página de desarrollador una herramienta de SDK de java script en la cual utilizan e implementan dentro de sus funciones el uso de la fórmula de Haversine para cálculo de distancia en tiempo real entre dos puntos, y menciona que,[9] cuando tenemos dos marcadores y estamos trazando una línea recta entre éstos, si queremos calcular la distancia, haremos uso de la fórmula de Haversine, porque hay

Tabla 1. Distribución Actual de Población.

Jurisdicción	Municipios
I	Aguascalientes, Jesús María, el Llano y San Francisco de los Romo
II	Cosío, Tepezalá, San José de Gracia, Pabellón de Arteaga y Asientos
III	Calvillo

Tabla 2. Distribución de unidades a cargo.

Jurisdicción	Unidades a cargo
I	44
II	30
III	15
Total	89

que recordar, que la Tierra no es plana, y para hacernos la vida más fácil, HERE JavaScript SDK se encarga de implementar esto por nosotros.

2.3. Primer nivel de atención en salud del estado de Aguascalientes

El Instituto de servicios de salud del estado de Aguascalientes tiene dentro de la dirección de servicios de salud dos niveles de atención, En el primer nivel de atención se encuentra la división de tres jurisdicciones sanitarias, las cuales atienden 89 centros de salud, en el segundo nivel de atención se encuentra la división de 5 hospitales generales [10]. Esta investigación utiliza la información que corresponde al primer nivel de atención y sus 3 jurisdicciones sanitarias actuales.

Como se mencionó en la introducción, una Jurisdicción sanitaria tiene las tareas de administrar y monitorear al personal, administrar y distribuir insumos, equipo y medicamentos a los centros de salud. La jurisdicción sanitaria 1 está ubicada en el municipio de Aguascalientes y es responsable de los centros de salud pertenecientes a los municipios de Aguascalientes, Jesús María, el Llano y San Francisco de los Romo.

La jurisdicción 2 está situada en el municipio de Rincón de Romos, y tiene a su cargo los centros de salud de los municipios de Cosío, Tepezalá, San José de gracia, Pabellón de Arteaga y Asientos. La jurisdicción sanitaria 3 tiene a su cargo los centros de salud pertenecientes al municipio de Calvillo, de esta manera se distribuye la carga de responsabilidad de los centros de salud del estado de Aguascalientes, en la tabla 1 se muestra esta información.

Actualmente la jurisdicción sanitaria 1 tiene a su cargo 44 centros de salud, la jurisdicción sanitaria 2 tiene a su cargo 30 centros de salud y la jurisdicción 3 tiene a su cargo 15 centros de salud, como se muestra en la tabla 2.

Las jurisdicciones sanitarias han sido agregadas según el crecimiento en cantidad de centros de salud, pero sin tomar en consideración la distancia desde el centro de salud a la oficina jurisdiccional.

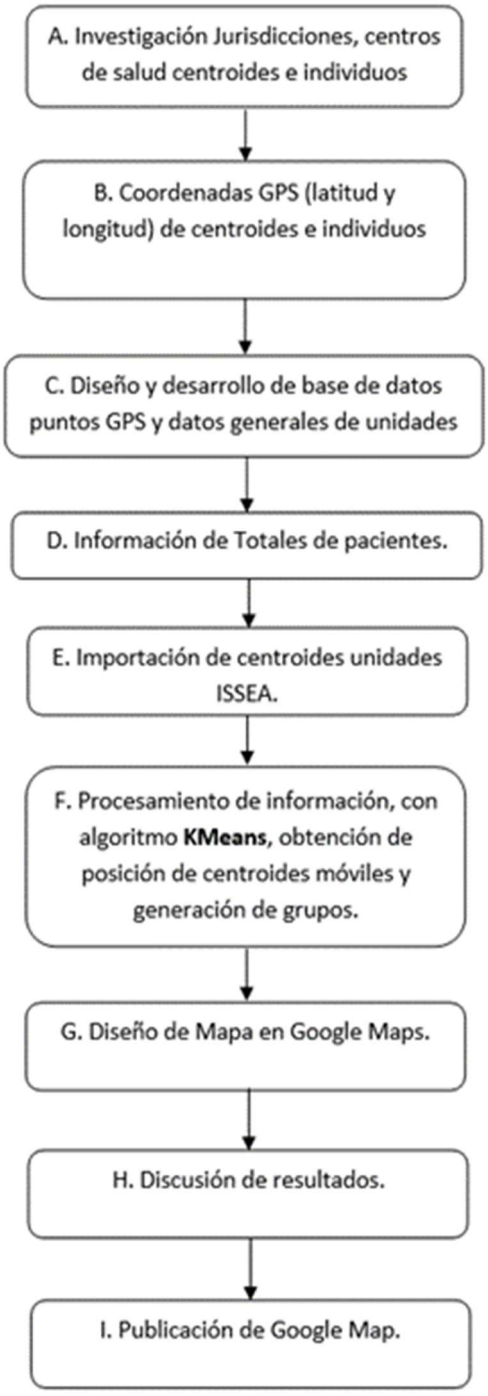


Fig. 2. Metodología Implementada.

3. Trabajos relacionados

3.1. Clustering K-Means para mejorar la calidad de servicios de organizaciones modernas

En el trabajo de Mamani Rodríguez [4] se señala que la minería de datos distribuida es una disciplina de alto interés para los investigadores ya que las organizaciones centralizadas actuales tienen ciertas limitaciones, menciona que las organizaciones competitivas deben mantenerse dispuestas al cambio, a la mejora continua de los servicios pero respetando los estándares de la industria.

Por ello están interesadas en utilizar herramientas informáticas que apoyen sus objetivos. Los investigadores plantean el desarrollo de un prototipo basado en minería de datos distribuida y la adaptación de un algoritmo de agrupamiento distribuido basado en la técnica K-means. La propuesta se basa en la detección de patrones de comportamiento basado en un proceso de negocio.

El estudio evalúa los datos nominales en repositorios en las sucursales de la organización y define un modelo dimensional para hacer un proceso de negocio específico, este modelo es implementado en los repositorios de información de la organización. Otro trabajo es el de Basyir [7] en el cual se creó una herramienta que determina los servicios de emergencia más cercanos mediante la posición geográfica del sujeto, este trabajo emplea el cálculo de la distancia de Haversine y agrupación con algoritmo de K-means.

4. Material y métodos

4.1. Metodología

Esta investigación fue desarrollada la siguiente metodología que se presenta en la Fig.2.

A continuación, se presentan los pasos a seguidos de la metodología ilustrada anteriormente en la figura 2.

A. Investigación de ubicación de Jurisdicciones, cabeceras municipales, centros de salud los cuales serán centroides e individuos de interés.

B. Extracción de coordenadas GPS (latitud y longitud) de cada centroide fijo, unidades ISSEA y determinación de posiciones iníciales de centroides móviles.

C. Diseño y desarrollo de base de datos para almacenamiento de puntos de localización GPS y datos generales de unidades del ISSEA y datos de cantidades de atención de pacientes.

D. Importación de información de datos GPS de unidades del ISSEA y centroides a base de datos.

Tabla 3. Ejemplo de Información obtenida de repositorios ISSEA.

Nombre	Ubicación Longitud, Latitud	Probables diabetes	Confirmados diabetes	Total, Pacientes	Id Unidad	Jurisdicción
C. N. OJOCALIENTE	21.886044,-102.220608	358	239	4,975	80	1
C. GREMIAL	21.899112,-102.290868	369	815	13,317	2	1
C. PROGRESO	21.904467,-102.268230	915	994	16,025	10	1
C. CONSTITUCIÓN	21.930177,-102.283669	1,282	1,111	27,053	3	1
C. G PORTUGAL	21.868991,-102.269119	890	1,039	15,773	9	1
C. INSURGENTES	21.849031,-102.310384	1,289	1,949	24,827	5	1

E. Obtención, procesamiento y Almacenamiento de Información de Totales de pacientes.
F. Procesamiento de información por medio de software desarrollado K-Means modificado para agrupación.
G. Proceso de diseño de Mapa en Google Maps, aplicación de Chrome apps.
H. Análisis, revisión y discusión de resultados.
I. Proceso de publicación de mapa en Google Map.

4.2. Preparación de la información

Para la obtención de datos generales de las unidades de salud y de totales de pacientes pertenecientes a estas unidades se obtuvo acceso por parte del ISSEA a la información de cada una de las bases de datos mediante tareas programadas para conseguir la información más reciente y actual.

A continuación, se describe el proceso de obtención, recolección y preparación de datos de los 89 centros de salud del ISSEA.

1. Desarrollo, diseño y creación de base datos *Datos_recoleccion* en servidor MS SQL Server[2].

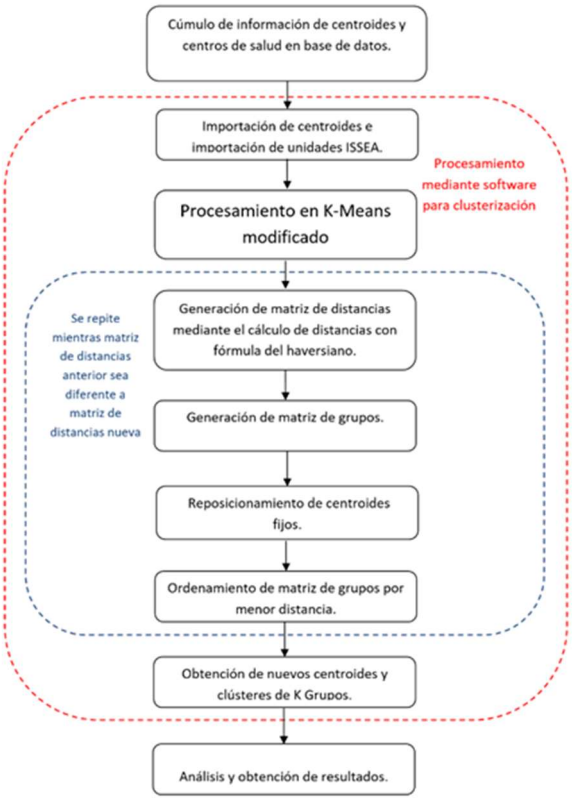


Fig.3. Marco de trabajo para procesamiento de información.

2. Creación y diseño de tablas contenedoras de datos de recolección en base de datos *Datos_recoleccion*.
3. Desarrollo y pruebas en una base de datos de prueba de consultas (query's) de los datos deseados a recolectar.
4. Generación de script para la inserción y actualización de datos obtenidos por las consultas (query's) en archivo *Datosrecoleccion.sql.*
5. Programación de tareas de recolección en MS SQL Server mediante la ejecución de tareas programadas y servicios de transformación de datos (DTS) los cuales se ejecutan en cada restauración de las bases de datos de las 89 unidades de salud del ISSEA. Estas restauraciones se realizan de manera automática en días laborales, mediante enlaces dedicados.
6. Revisión de confirmación de ejecución de tareas programadas y datos de las ejecuciones, tasa de transferencia, porcentajes, estadísticas, etc.
7. Revisión de los datos insertados o actualizados diariamente en las tablas de *Datos_recoleccion*.

En la Tabla 3, se muestra un ejemplo de los datos almacenados en la Tabla *Datos _recolección.*

5. Resultados

Como resultado de esta investigación se tiene el diseño, desarrollo e implementación del marco de trabajo para procesamiento de información, el cual se describe en la figura 3.

En la figura anterior se presentaron los pasos a seguir para llevar a cabo el marco de trabajo para procesamiento de información, a continuación, se muestra detalladamente el orden y secuencia de los pasos a seguir del marco de trabajo.

Procesamiento de información por medio de software desarrollado para agrupación.

1. Importación de centroides e importación de unidades ISSEA.
2. K-means:
   a) Generación de matriz de distancias mediante el cálculo de distancias con fórmula del Haversine.
   b) Generación de matriz de grupos.
   c) Posicionamiento de centroides fijos.
   d) Ordenamiento de matriz de grupos por menor distancia.
   e) Cálculo de nueva posición de centroides.
   f) Repetir de a-e hasta no tener diferencias en Matriz de grupos anterior.
3. Obtención de nuevos centroides y clústeres de K Grupos.

Adicionalmente se diseñó, desarrollo e implemento el siguiente algoritmo K-means modificado para centroides fijos y móviles:

m: cantidad total de centroides.
n: cantidad de centroides móviles.

1. IMPORTA individuos.
2. IMPORTA centroides **fijos** y **móviles**.
3. MIENTRAS la matriz de grupos anterior diferente con la matriz de grupos nueva.
   a) LIMPIA matriz de grupos anterior con ceros.
   b) LIMPIA matriz de centroides nuevos.
   c) RECORRE todas las líneas de individuos y COMPARA con centroides y GENERA matriz de distancias resultantes.
   d) RECORRE todas las líneas de distancias y COMPARA cual es la menor y GENERA matriz de grupos nueva.
   e) ESTABLECE centroides **fijos** 1-m.
   f) RECORRE todas las líneas de grupos (m+1)-n y SUMA los que son 1 en sus respectivos individuos y GENERA nueva posición de centroides móviles.
   g) COMPARA matriz de grupos anterior con matriz de grupos nueva.
4. PREGUNTA, ¿son diferentes la matriz de grupos anterior con la matriz de grupos nueva?
5. SI.
   a) LIMPIA matriz de grupos nueva con ceros.
   b) IGUALA matriz de centroides anteriores a matriz de centroides nuevos.
   c) LIMPIA matriz de centroides nuevos.
   d) REPITE 3 y 4.

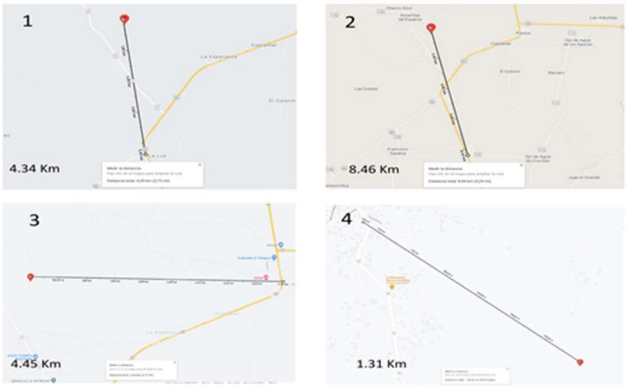


Fig. 4. Localización de posibles ubicaciones de oficina Jurisdiccional 4.

Tabla 4. Distribución Actual de Población.

Jurisdicción	Unidades	Probables diabetes	Confirmados diabetes	Total, Pacientes
I	44	25,430	26,933	461,058
II	30	8,172	8,177	126,633
III	14	6,812	6,103	71,213
Todas	89	40,414	41,213	658,904

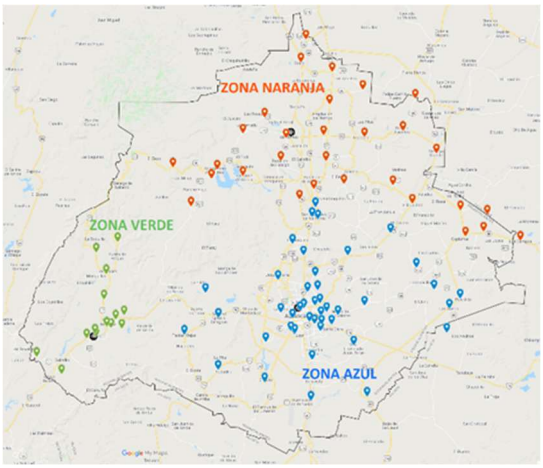


Fig. 5. Distribución y ubicación actuales de centros de salud ISSEA.

6. NO.
   a) TERMINA.

Como resultado del algoritmo, se propone la inclusión de la jurisdicción 4 como centroide nuevo, contando así con 4 agrupaciones en total, esta decisión es de carácter

Tabla 5. Distribución de Población Propuesta.

Jurisdicción	Unidades	Probables diabetes	Confirmados diabetes	Total, Pacientes
I	35	23,868	25,105	430,463
II	26	6,609	6,530	91,660
III	14	6,812	6,103	71,213
IV	14	3,125	3,475	65,568
Todas	89	40,414	41213	658,904

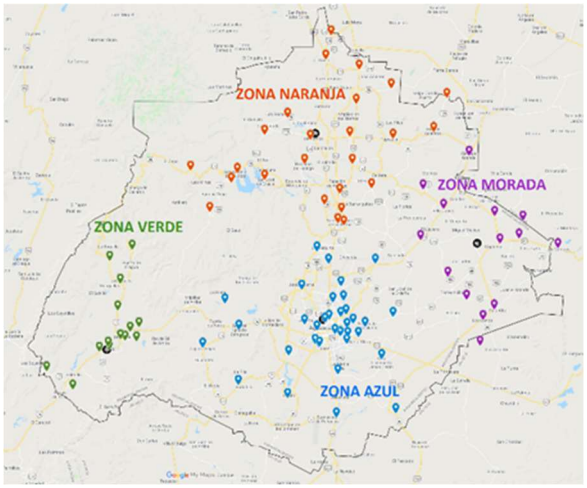


Fig. 6. Nueva agrupación propuesta de centros de salud ISSEA.

económico (pocos centroides o jurisdicciones). Para la ubicación de la nueva jurisdicción se proponen 4 comunidades o ubicaciones posibles ya que son comunidades cercanas al punto determinado por el algoritmo y están habilitadas con servicios básicos. Las ubicaciones son: La Luz, Terremoto, Pilotos y Amarillas de Esparza. La distancia del punto "Jurisdicción 4" determinado por el algoritmo (22.037063, -102.003551) al que llamaremos punto de atención o PDA a comunidades cercanas con los servicios necesarios para operar una oficina Jurisdiccional son:

1. PDA a La luz 4.34km con 870habitantes,
2. PDA a El Terremoto 8.46km con 354 habitantes,
3. PDA a Pilotos 4.45km 1282 con habitantes,
4. PDA a Amarillas de Esparza1.31km con 808 habitantes.

En la figura 4 se observa las distancias en mapa de cada una de las comunidades antes mencionadas. Actualmente, los 89 centros de salud atienden a un total de 658,904 pacientes registrados distribuidos de la manera mostrada en la tabla 4. En la figura 4, se presenta la distribución actual de centros de salud por cada jurisdicción, de color azul

los pertenecientes a la jurisdicción 1, en color naranja los pertenecientes a la jurisdicción 2 y en color verde los pertenecientes a la jurisdicción 3.

La propuesta de atención a pacientes registrados para cada una de las jurisdicciones quedaría redistribuida como se presenta en la tabla 5. En la figura 5, se presenta la propuesta de distribución de los 89 centros de salud en 4 jurisdicciones, en color azul los pertenecientes a la jurisdicción 1, en color naranja los pertenecientes a la jurisdicción 2, en color verde los pertenecientes a la jurisdicción 3 y en color violeta los pertenecientes a la nueva jurisdicción 4.

6. Conclusiones y discusión

En base a los resultados obtenidos, se puede concluir que el algoritmo K-means modificado, es una herramienta robusta que se puede adecuar a diferentes situaciones como son la distribución y agrupamiento. En este caso se adaptó para funcionar con centroides fijos y móviles dada la particularidad de la, y se obtuvo un resultado muy conveniente. Mediante la jurisdicción 4, se reduciría la carga de trabajo de las jurisdicciones 1 y 2 en un 6.6% y un 27.6% respectivamente, atendiendo 30,595 personas de la jurisdicción 1 y 34973 de la jurisdicción 2.

En el análisis de las posibles localidades para establecer la jurisdicción 4, se concluye que no existe ninguna regla de restricción sobre la posición de la oficina jurisdiccional, sin embargo, se encontró que la conformación de una jurisdicción sanitaria está determinada por la regionalización, con el fin de planificar la prestación de servicios, proporcionar el uso óptimo de recursos y responder a las necesidades de atención a la salud particulares[11], por lo cual, se concluye que dadas las características de la localidad de Pilotos, municipio de Asientos, es la localidad ideal ya que a diferencia de las otras posibles localidades, cuenta con una población de más de 1000 personas, tiene los servicios (luz, agua, drenaje, teléfono) básicos necesarios para operar una jurisdicción, cuenta con acceso carretero mediante la carretera federal 43 y además cuenta con un centro de salud actualmente funcionando, con las facilidades para poder instaurar la oficina jurisdiccional 4.

Para trabajos futuros y con el fin de experimentar diferentes resultados, se considera trabajar con más de un centroide variable e integrar y posicionar geográficamente otras nuevas jurisdicciones en el estado, así como trabajar con uno o varios centroides fijos, según sea el caso de jurisdicciones que se necesiten conservar en su posición geográfica actual, además de la comparación de resultados con otras técnicas de búsqueda exhaustiva.

Referencias

1 Díaz de León-Castañeda, C., Góngora-Ortega, J.: eSalud en servicios de salud públicos en México: Estudio de caso, vol. 32, e21256 (2020) doi: 10.22198/rys2020/32/1256

2 Microsoft. “Microsoft SQL Server 2008 R2 SP2 - Express Edition” (2008) https://www.microsoft.com/es-mx/download/details.aspx?id=30438

3 Salcedo-López, F. H.: Minería de datos para segmentación de clientes en la empresa tecnológica Master PC. pp. 184 (2015) [Online], Available: http://dspace.unl.edu.ec/jspui/ bitstream/ 123456789/10462/1/Chamba Jiménez%2C Sairy Fernanda.pdf

4 Mamani, Z.: Aplicación de la minería de datos distribuida usando algoritmo de clustering K-Means para mejorar la calidad de servicios de las organizaciones modernas. pp. 150 (2015)
5 Cifuentes-Ramos, F. A.: Clasificación automática de Tweets utilizando K-NN y K-Means como algoritmos de clasificación automática, aplicando TF-IDF y TF-RFL para las ponderaciones. Pontif. Univ. Católica Valparaíso Fac. Ing. Esc. Ing. Informática (2016)
6 Chopde, N., Nichat, M.: Landmark based shortest path detection by using A* and Haversine formula. International Journal of Innovative Research in Computer and Communication Engineering, vol. 1, no. 2, pp. 298–302 (2013) [Online], Available: http://www.ijircce.com/ upload/ 2013/april/17_V1204030_Landmark_H.pdf.
7 Basyir, M., Nasir, M., Suryati, S., Mellyssa, W.: Determination of nearest emergency service office using Haversine formula based on android platform. EMITTER International Journal of Engineering Technology, vol. 5, no. 2, pp. 270–278 (2018) doi: 10.24003/emitter. v5i2.220
8 Rode, S.: How to calculate distances on a map with the maps javascript API (2019) https:// clo ud. google.com/blog/products/maps-platform/how-calculate-distances-map-maps-java script-api
9 Raboy, N.: Distance calculations between points with the HERE javascript SDK. https://developer.here.com/blog/distance-calculations-between-points-with-the-here-javasc ript-sdk.
10 1D. Administrativo, primer nivel de atención (2020) https://www.aguascalientes.gob. mx/isea/prinivel.html.
11 Poder ejecutivo del Estado.: Manual de organización tipo de jurisdicción sanitaria. Gaceta del Gobierno del Estado de México, no. 308, pp. 40 (2011)

ISSN 1870-4069

Análisis de un láser de fibra óptica dopada con Erbio y modulación de bombeo por medio de grafos de visibilidad

Jorge Luis. García-Rito[1], Jesús R. Sevilla-Escoboza[1]

[1] Centro Universitario de los Lagos, UDG. Jalisco, México

jorgeluis95033@gmail.com, jesus.sescoboza@academicos.udg.mx

Resumen. La caracterización de sistemas no lineales es un campo abierto, durante los últimos 30 años se han desarrollados herramientas formales cálculos y herramientas computacionales que permiten identificar diferentes tipos de comportamientos y avanzar en su aprovechamiento en diversas aplicaciones. Este trabajo se enfoca en la puesta a punto de un algoritmo que parta exclusivamente de series temporales, independientemente del modelo, como una primera aproximación se usa un láser de fibra dopada con Erbio con modulación de bombeo debido a su gran cantidad de estados dinámicos, pasando de estados cuasi-periódicos, caos y la presencia de multiestabilidad. Específicamente se parte del análisis tradicional por medio del diagrama de bifurcación y la frecuencia entre pulsos, para después implementar la técnica de grafo de visibilidad y la medida de agrupamiento en redes complejas para identificar los diferentes estados dinámicos.

Palabras clave: Laser, caos, grafos, redes complejas.

Analysis of an Erbium-Doped Optic Fiber Laser and Pump Modulation Using Visibility Graphs

Abstract. The characterization of non-linear systems is an open field, over the last 30 years formal calculation and computational tools that allow the identification of different types of behaviors have been developed in order to advance in their use in various applications. This work focuses on the fine-tuning of an algorithm that starts exclusively from time series, regardless of the model. As a first approximation an Erbium-doped fiber laser with pump modulation is used due to its large number of dynamic states, such as quasi-periodic states, chaos and the presence of multistability. Specifically, it begins with the traditional analysis by bifurcation diagrams and the frequency between pulses and later implements the visibility graph technique and clustering measure in complex networks to identify the different dynamic states.

Keywords: Laser, chaos, graphs, complex networks.

1. Introducción

Desde su invención, se han desarrollado diversas aplicaciones para los sistemas láser, actualmente tanto los láseres de emisión constante como los de emisión pulsada, son utilizados en la industria, medicina, tratamientos cosméticos y diversas áreas de investigación como fuentes de luz [1-3]. Es importante mencionar que este avance en aplicaciones continúa evolucionando, lo cual ha presentado situaciones en las que los sistemas láseres deben ser mejorados para su aplicación adecuada. Una de estas situaciones la podemos describir de la siguiente forma: los pulsos láseres carecen de estabilidad temporal entre cada emisión además de variaciones en la amplitud de pulsos, lo cual, al este también ser un sistema sensible a condiciones iniciales, puede llegar a presentar comportamientos caóticos.

A lo largo de la historia se han desarrollado técnicas para llevar sistemas caóticos a estados estables, modificando un parámetro del sistema, cuidadosamente seleccionado [4]. Estas mismas técnicas han sido implementadas en un sistema láser, demostrando que es posible mejorar dicha estabilidad en esos sistemas [5]. Sin embargo, aún nos encontramos en la necesidad de una inestabilidad menor para aplicaciones donde se requiera mayor precisión además de una forma de caracterizarla.

En este trabajo se propone una métrica, utilizando grafos de visibilidad y técnicas de análisis de redes, para la caracterización de la estabilidad de los pulsos láseres generados por modulación en la corriente de bombeo en un láser de Erbio. Dando como resultado una forma gráfica, como analítica de la estabilidad de nuestro sistema.

2. Trabajos relacionados

Se ha demostrado la capacidad de llegar a controlar sistemas caóticos. En [6] se utilizan distintos métodos de control de sistemas multiestables utilizando seguimiento de trayectorias, reforzamiento y modulación armónica. Con la validez de este trabajo, tenemos certeza de las aplicaciones en sistemas distintos, tales como un sistema láser. Una de estas técnicas antes mencionadas, la modulación armónica, se ha aplicado en sistemas láser, en el trabajo [4] se logró llevar un sistema caótico a un estado periódico estable utilizando una variación de un parámetro, dicho parámetro fue cuidadosamente seleccionado.

Posteriormente Rajashi Roy y colaboradores realizaron una modificación a este método para lograr controlar un láser de estado sólido [5]. El continuo avance de la tecnología láser viene de la mano con nuevos desafíos a solucionar. Comportamientos caóticos no deseados han sido observados en láseres de estado sólido de diferentes materiales [7]. Estos comportamientos resultan ser no deseados para la aplicación requerida.

Los trabajos mencionados anteriormente tienen el objetivo de convertir o estabilizar un comportamiento en específico, pero se carece de la herramienta basada en la dinámica resultante para evaluar el comportamiento final, es por esto por lo que en este trabajo se propone el uso de los grafos de visibilidad para caracterizar la dinámica final. Antes de continuar debemos mencionar que la herramienta más común que podemos encontrar en los análisis de sistemas dinámicos son los diagramas de bifurcación, estos

son una representación gráfica del comportamiento de un sistema cuando se varía un parámetro de este [8].

Para la construcción de dicho diagrama es necesario variar, en pequeñas cantidades, un parámetro específico del sistema a analizar y observar el estado obtenido con cada variación. Esto nos brinda una imagen general del comportamiento del sistema. La importancia de estas gráficas radica en la capacidad de analizar de manera experimental los diferentes estados del sistema dependiendo del parámetro a variar.

Tomando el modelo matemático de un láser de Erbio de estado sólido de [9] y utilizando estas distintas herramientas, observamos el comportamiento del sistema láser al someterlo a una señal moduladora, variando su amplitud y frecuencia, para generar una salida pulsada. La estabilización de una salida pulsada ha sido un amplio tema de investigación debido a la gran cantidad de áreas en las que se pueden aplicar dichos pulsos estabilizados. Este tipo de experimentos nos dará como resultado una serie temporal en donde tenemos la amplitud de los pulsos láser. Existen distintas formas de interpretar la información de las series temporales, una de ellas es convertirlas en un grafo de visibilidad.

En [10] se presenta un método de análisis de series temporales, el cual consiste en un algoritmo que convierte dicha serie en un grafo de visibilidad. Las propiedades de estos grafos nos permiten manejar cualquier serie temporal como una red compleja. Una denominación de estados grafos es Grafo de Visibilidad Natural (NVG, por sus siglas en inglés) [11].

Se forma evaluando cada punto de la serie temporal entre sí y, siguiendo el criterio de evaluación, se realiza una conexión si dicho criterio se cumple. Dándonos como resultado, una red en la cual, cada nodo es un valor de la serie temporal y el peso de cada enlace entre ellos nos da su grado de visibilidad. Con esta nueva interpretación de los datos, podemos utilizar herramientas de análisis de redes para obtener información relevante. Se ha demostrado la robustez de los grafos de visibilidad ya que la dinámica de los sistemas es fácilmente distinguible a pesar de la presencia de ruido, además pueden ser aplicados en una gran variedad de áreas.

Una red, por definición es un conjunto de entidades conectadas entre sí [12]. Al encontrarse en gran parte de nuestro entorno, el análisis de redes y manipulación de estas es de gran importancia. Dentro de estas, podemos obtener información de sus nodos mediante diversas herramientas de análisis, ya sea a una escala microscópica, mesoscópica y macroscópica.

3. Metodología

La metodología utilizada para esta propuesta se describe a continuación:

Utilizando las siguientes ecuaciones de modelado de un láser de Erbio procedemos a variar, de manera analítica, la potencia de bombeo y el ángulo de polarización para, de esta forma, analizar el cambio que generan estas variaciones. Las ecuaciones que describen el sistema son las siguientes:

$$\frac{dS}{d\tau} = -S[1 + mCos(wt) - D], \tag{1}$$

$$\frac{dD}{d\tau} = -\gamma[D - A + SD], \tag{2}$$

$$(i) = \frac{l_{(k)}}{\frac{(p_k(p_k - 1))}{2}}. \quad (3)$$

Estas son las ecuaciones simplificadas del modelo inicial tomado de [9] donde, S es la intensidad normalizada y D es la inversión de población, mientras que, los parámetros m y w son la amplitud de bombeo y la frecuencia de modulación respectivamente.

Para el análisis se utiliza el Grafo de Visibilidad, el cual se ha utilizado como herramienta para estimar el exponente de Hurst movimiento browniano fraccional [10]. Estos grafos se crean a partir de una serie temporal, en este caso hablamos de la serie de pulsos emitidos por el láser.

Cada punto en la serie consta de su magnitud y el instante del tiempo en el que se presenta. Para crear nuestro grafo de visibilidad debemos seguir el siguiente criterio de visibilidad. Dos valores arbitrarios (x_1, y_1) y (x_2, y_2) tendrán visibilidad y, consecuentemente estarán conectados, si cualquier otro dato (x_3, y_3) colocado entre ellos cumple lo siguiente:

$$y_3 < y_2 + \frac{(y_1 - y_2)(x_2 - x_3)}{x_2 - x_1}. \quad (4)$$

Cada uno de los puntos son evaluados entre sí y conectados si cumplen el criterio anterior, dando como resultado el grafo de visibilidad. A los nodos de las redes resultantes se le calcula el coeficiente de agrupamiento individual para luego obtener el valor promedio característico de la red [13], este índice cuantifica qué tanto está de agrupado, o mejor dicho interconectado con los nodos vecinos. Valores altos de agrupamiento indican una alta conectividad entre los nodos, formando así un grupo de mayor tamaño, mientras que valores bajos representan grupos más pequeños.

Para obtener el valor del coeficiente de agrupamiento individual [14]:

- Para cada nodo v_i de la red se obtiene el número de conexiones que tiene el nodo, es decir se calcula el grado del mismo $p_{(k)}$, lo cual es equivalente a encontrar los vecinos del nodo.
- Se encuentra el número de enlaces de los vecinos l_k.
- Se calcula el agrupamiento individual por medio de la ecuación siguiente:

$$C(i) = \frac{l_{(k)}}{\frac{(p_k(p_k - 1))}{2}}. \quad (5)$$

- Finalmente se calcula el valor promedio.

3.1. Resultado de diagrama de bifurcación

Para la construcción de los diagramas de bifurcación se realizan variaciones de la amplitud de modulación y de la frecuencia de modulación, la integración numérica se realiza utilizando el software MATLAB con el método de integración de Runge-Kutta de cuarto orden, con un tamaño de paso de $1x10^{-1}$ y un total de 2^{17} pasos, variando la amplitud de modulación de 0.01 a 1 u. a.; mientras que la frecuencia de modulación en el rango de 0.18 a 0.30 u. a.

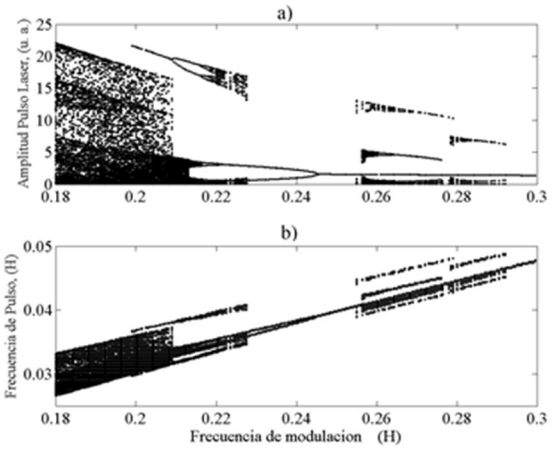


Fig. 1. Diagrama de bifurcación cuando la amplitud de la señal de modulación es m = 1.

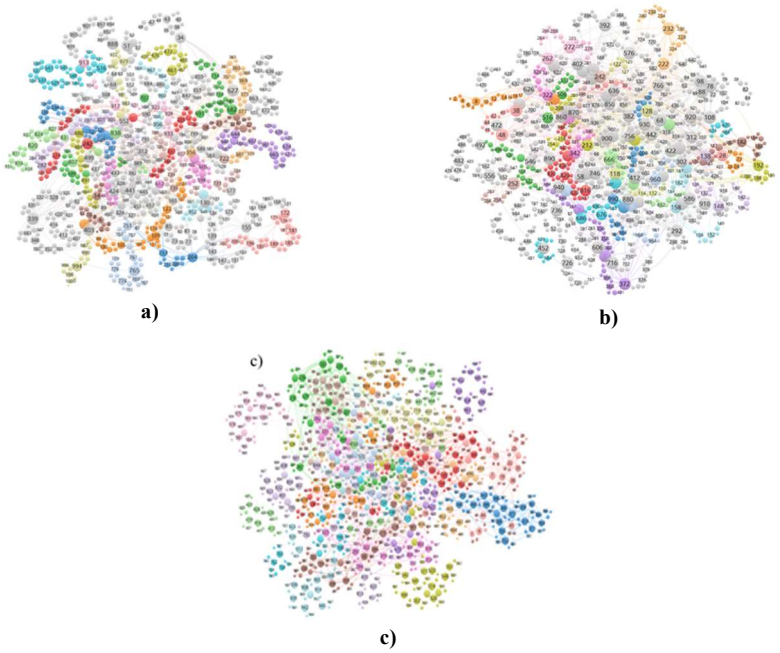


Fig. 2. Representación visual de las redes obtenidas por el algoritmo de grafos de visibilidad natural aplicados a las series temporales cuando a) m = 0.18, b) m = 0.24 y c) m = 0.25.

Usando estos parámetros se obtiene el siguiente diagrama de bifurcación: Analizando la figura 1 (a), observamos que existe una región caótica desde 0.18 hasta 0.21, sin embargo, posterior a eso, podemos destacar que se presentan múltiples comportamientos coexistentes, es decir en estas regiones dependiendo de las condiciones iniciales podemos encontrar dos regímenes dinámicos distintos. En el caso

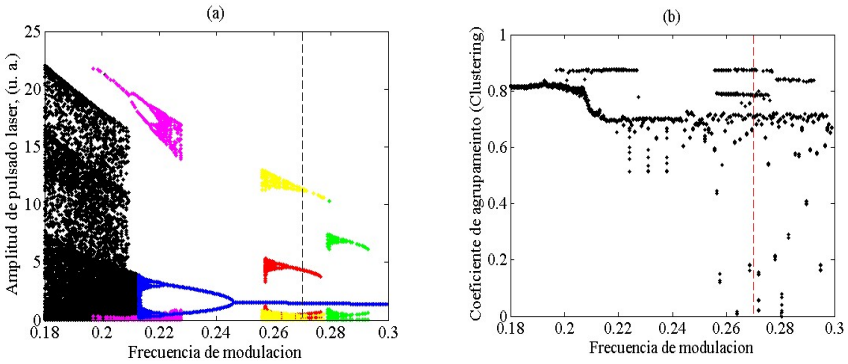


Fig. 3. Diagrama de bifurcación con los diferentes comportamientos que se pueden encontrar con las variaciones antes mencionadas. La figura b) muestra el agrupamiento.

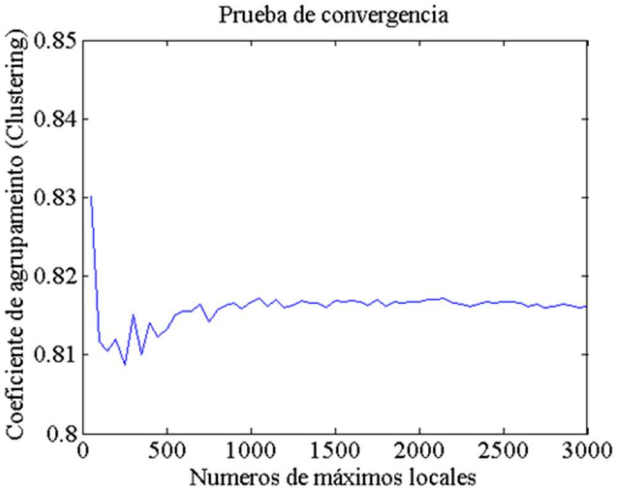


Fig. 4. Prueba de convergencia. Al aumentar el número de máximos observamos que el agrupamiento tiende a un valor, lo cual nos da una certeza de la cantidad de máximos necesaria para minimizar el error.

de la figura 1 (b) se calculó el tiempo entre las emisiones láser, de nueva cuenta podemos encontrar regiones donde las distancias entre las emisiones láser son caóticas, es decir existe una variación considerable en la frecuencia, hasta regiones donde el tiempo de pulsado pasa a ser periódico. Con el objetivo de evaluar la efectividad del grafo de visibilidad seleccionamos tres puntos del diagrama de bifurcación, cuando $m = 0.18, m = 0.24$ y cuando $m = 0.25$.

Utilizamos el algoritmo de conversión de una serie temporal a un grafo de visibilidad. De esta forma obtendremos una red compleja a la cual aplicaremos el algoritmo antes descrito para evaluar el coeficiente de agrupamiento y de esta manera proponer su utilización en experimentos reales donde no sea posible obtener los exponentes de Lyapunov para poder evaluar la complejidad del comportamiento.

3.2. Grafos de visibilidad

La figura 2 es la representación gráfica de la red compleja obtenida de las series temporales en las cuales los colores representan las comunidades vecinas, es decir con quien tiene más afinidad un nodo. Se pueden obtener distintas características de estas redes, sin embargo, lo ideal para nuestro objetivo es obtener el coeficiente de agrupamiento promedio.

En la figura 2, debemos notar la presencia en cada una de las redes de un gran grupo gris, este representa la comunidad de vecinos más grande a la cual se conectan todos los demás, en la figura 2 (a) cuando $m = 0.18$ tenemos un agrupamiento alto, ya que en ese punto el sistema presenta comportamientos caóticos, lo que significa que gran cantidad de puntos, o bien, nodos, están conectados entre sí.

En el caso $m = 0.24$, la serie temporal corresponde a un periodo doble, donde únicamente tenemos dos máximos locales característicos, el valor del agrupamiento decae, debido a la aparición de pequeños grupos, que se acentúan en el caso de $m = 0.25$, donde se tiene una serie temporal periódica donde el valor de un solo máximo local se repite, por lo las comunidades son cada vez más visibles.

3.3. Diagrama de bifurcación y agrupamiento

En la figura 3. (a) podemos observar el diagrama de bifurcación con los diferentes comportamientos que se pueden encontrar. Este modelo presenta una dinámica muy rica ya que se encuentran diversas regiones de coexistencia de atractores en la figura 3(b) se muestra el coeficiente de agrupamiento y observamos cómo cada uno de estos atractores tiene un valor propio.

Finalmente realizamos una prueba de convergencia en la cual tomamos de 0 a 3000 máximos locales y calculamos el agrupamiento. Se realizó de esta forma debido a que esta métrica de redes es sensible a la cantidad de puntos para evaluar. El resultado se observa en la siguiente figura 4. Podemos observar que el coeficiente de agrupamiento tiende a un valor a partir de haber tomado 1500 o más puntos.

4. Conclusiones y trabajo a futuro

Al modular la fuente de bombeo de un sistema láser podemos obtener una gran cantidad de comportamientos dinámicos, tales como periódicos, caóticos y multiestables. Manipular dos parámetros, tales como la amplitud de la señal moduladora y la frecuencia de esta, da como resultado diversas variaciones en la amplitud de pulsos y en su tiempo de repetición.

Implementar los grafos de visibilidad como herramienta para la manipulación e interpretación de la serie temporal resultante de modular el sistema láser, nos permite obtener información relevante desde otra perspectiva lo cual abre la posibilidad de determinar, mediante el cálculo del coeficiente de agrupamiento, una forma de caracterizar dicha salida. Sin embargo, podrían existir otras métricas o técnicas de manipulación de redes para complementar el resultado obtenido y realizar una caracterización sólida.

La importancia de utilizar el grafo de visibilidad permite explorar con mayor precisión experimentos reales, ya que la mayoría de los láseres y sistemas dinámicos carecen de modelos que representan de forma real a los dispositivos reales, por lo cual implementar técnicas como el exponente de Lyapunov, dimensión fractal del atractor entre otras no requiere un mayor costo computacional, por lo cual este trabajo permite explorar la posibilidad de análisis en tiempo real.

Como conclusión particular de este análisis podemos afirmar que el análisis de las redes construidas a partir de series temporales permite inferir la naturaleza de las mismas, en señales altamente complejas (por ejemplo, señales caóticas) muestran un alto nivel de agrupamiento por lo tanto una alta conectividad entre los nodos, semejante a redes aleatorias, señales periódicas muestran un índice bajo de agrupamiento por lo cual podemos encontrar la presencia de comunidades más definidas.

Referencias

1. Hutton-Carlsen, K., Esmann, J., Serup, J.: Tattoo removal by Q-switched yttrium aluminium garnet laser: Client satisfaction. Journal of the European Academy of Dermatology and Venereology, vol. 31, no. 5, pp. 904–909 (2017) doi: 10.1111/jdv.14124
2. Rothschild, P. S., Hooshmand, J., Allen, P. L., Vote, B. J.: Comparison of the neodymium: YAG rate after femtosecond laser–assisted and phacoemulsification cataract surgery. Journal of Cataract & Refractive Surgery, vol. 44, no. 3, pp. 407–408 (2018) doi: 10.1016/ j.jcrs.2018.01.017
3. Alarfaj, M. A., Hasen, A. A., Al-Yami, S. S.: Traumatic subhyaloid macular hemorrhage with complete resolution following neodymium-doped yttrium aluminium garnet laser. American journal of ophthalmology case reports, vol. 9, pp. 85–87 (2018) doi: 10.10 16/j.ajoc.2018.01.020
4. Ott, E., Grebogi, C. Yorke, J.: Controlling Chaos. Physical review letters, vol. 64, pp. 1196 (1990) doi: 10.1103/PhysRevLett.64.1196
5. Roy, R., Murphy, T., Maier, T., Gills, Z., Hunt, E.: Dynamical control of a chaotic laser: Experimental stabilization of a globally coupled system. Physical Review Letters, vol. 68, pp. 1259 (1992) doi: 10.1103/PhysRevLett.68.1259
6. Sevilla-Escoboza, R., Pisarchik, A. N., Jaimes-Reátegui, R., Huerta-Cuellar, G.: Selective monostability in multi-stable systems. In: Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 471, no. 2180 (2015) doi: 10.1098/ rspa.2015.0005
7. Tang, J., Bai, Z., Zhang, D., Qi, Y., Ding, J., Wang, Y., Lu, Z.: Advances in all-solid-state passively Q-Switched lasers based on Cr4+: YAG saturable absorber. Photonics, Multidisciplinary Digital Publishing Institute, vol. 8, no. 4, pp. 93 (2021) doi: 10.3390/ photonics8040093
8. Buscarino, A., Fortuna, L., Frasca, M.: Essentials of nonlinear circuit dynamics with MATLAB® and laboratory experiments. CRC Press (2017)
9. Metayer, C., Serres, A., Rosero, E. J., Barbosa, W. A. S., De Aguiar, F. M., Leite, J. R., Tredicce, J. R.: Extreme events in chaotic lasers with modulated parameter. Optics express, vol. 22, no. 17, pp. 19850–19859 (2014) doi: 10.1364/OE.22.019850
10. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuno, J. C.: From time series to complex networks: The visibility graph. In: Proceedings of the National Academy of Sciences, vol. 105, no. 13, pp. 4972–4975 (2008) doi: 10.1073/pnas.0709247105
11. Iacovacci, J., Lacasa, L.: Sequential visibility-graph motifs. Physical Review E, vol. 93, no. 4, pp. 042309 (2016) doi: 10.1103/PhysRevE.93.042309

12. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D. U.: Complex networks: Structure and dynamics. Physics reports, vol. 424, no. 4-5, pp. 175–308 (2006) doi: 10.1016/j.physrep.2005.10.009
13. Watts, D. J., Strogatz, S. H.: Collective dynamics of 'small-world'networks. Nature, vol. 393, no. 6684, pp. 440–442 (1998) doi: 10.1038/30918
14. Rubinov, M.: Sporns, O.: Complex network measures of brain connectivity: Uses and interpretations. Neuroimage, vol. 52, no. 3, pp. 1059–1069 (2010) doi: 10.1016/j.neuroimage.2009.10.003

ISSN 1870-4069

Prevención de riesgos de trabajo mediante la detección de emociones relacionadas con el estrés laboral ocupando técnicas de visión artificial para el cumplimiento de la NOM-35-STPS-2018

Efrén Uriel Aguilar Juárez, Brenda González Torres, Fernando Munguia Reyes

Universidad Politécnica del Valle de México,
División de Ingeniería Mecatrónica y Mecánica Automotriz,
México

{efren.aguilar.juarez, brenda.gonzalez.torres, Fernando.munguia.reyes}@upvm.edu.mx

Resumen. El estrés laboral se manifiesta con un conjunto de reacciones perjudiciales, tanto físicas como emocionales, que se originan cuando las exigencias del trabajo sobrepasan las capacidades, recursos o disponibilidad de tiempo de los empleados. De igual manera, el estrés emocional, juega también un papel importante dentro de todo esto, ya que muchas veces los problemas personales, familiares y/o sociales, mezclados con diversas situaciones dentro del área de trabajo, pueden llegar a suscitar accidentes en donde no solo la integridad física y emocional de la persona se vean afectadas, sino también de terceros. EstresNet, busca ser una alternativa para la prevención de riesgos e innovación para el mejoramiento de las condiciones laborales. Este fue creado mediante el uso de técnicas de Deep Learning, donde se utilizó un total de 2558 imágenes para su entrenamiento y 281 para el test, dichas imágenes fueron obtenidas por cuenta propia, de familiares y amigos. Se obtuvo una precisión del entrenamiento del 100% y una precisión de test del 92.882% para detección de emociones. EstresNet, está dirigido para toda empresa que esté interesada en la salud mental de su plantilla laboral y el bienestar dentro de ella.

Palabras clave: NOM-35-STPS-2018, estrés, riesgo, detección, EstresNet, CNN, Matlab, YOLOv2.

Preventing Work Risks by Detecting Emotions Related to Work Stress Occupying Machine Vision Techniques for Compliance with NOM-35-STPS-2018

Abstract. Work stress manifests itself with a set of harmful reactions, both physical and emotional, that arise when the demands of work exceed employees' capabilities, resources, or time availability. Similarly, emotional stress also plays an important role within all this, since often personal, family and/or social problems, mixed with various situations within the workplace, it can lead to

accidents where not only the physical and emotional integrity of the person is affected, but also of third parties. EstresNet seeks to be an alternative for risk prevention and innovation for improving working conditions. This was created by using Deep Learning techniques, a total of 2558 images were used for their training and 281 for the test, these images were obtained on their own, from family and friends. 100% training accuracy and 92.882% test accuracy were achieved for emotion detection. EstresNet, is aimed at any company that is interested in the mental health of its workforce and the well-being within it.

Keywords: NOM-35-STPS-2018, stress, risk, detection, EstresNet, CNN, Matlab, YOLOv2.

1. Introducción

Según datos de la Organización Mundial de la Salud (OMS), México ocupa el primer lugar a nivel mundial en estrés laboral, con 75% de su fuerza laboral bajo esta condición. Le siguen China (73%) y Estados Unidos (59%). Si bien, la salud del trabajador es principalmente la que resiente el impacto de los riesgos psicosociales llegando a provocar afectaciones a la salud física, salud mental, trastornos psicosomáticos, etc. También tiene repercusiones en los costos económicos por medio de accidentes de trabajo, pérdida de materiales, bajo rendimiento y baja productividad, debido a que gran cantidad de los accidentes están asociados a factores de riesgo psicosocial en los centros de trabajo [1]. Los accidentes laborales y las enfermedades ocupacionales generan pérdidas cercanas al 4% del Producto Interno Bruto global cada año [2].

Organismos internacionales como la OMS, reconocen que los trastornos mentales afectan a la capacidad de trabajo y ocasionan pérdidas de productividad, por otra parte, también señalan que las empresas que ponen en práctica iniciativas para promover la salud mental de sus trabajadores obtienen como resultado un aumento en la productividad [3]. En México la secretaria del trabajo y prevención social, por medio de la NORMA Oficial Mexicana NOM-035-STPS-2018, tiene como objetivo establecer los elementos para identificar, analizar y prevenir los factores de riesgo psicosocial, así como para promover un entorno organizacional favorable en los centros de trabajo [4].

Dicho lo anterior conviene precisar que esta norma no contempla una evaluación psicológica de los trabajadores ni establece ninguna obligación para identificar trastornos mentales, conocer variables psicológicas internas del individuo, medir el estrés en los trabajadores o evaluar su perfil psicológico, lo cual está fuera de su alcance. Dicha norma contempla a los exámenes médicos y evaluaciones psicológicas como una medida de aplicación general, sin embargo dado que los factores de riesgo psicosocial pueden provocar ciertos trastornos de la mente, requiere para su aplicación que se cumplan al menos las siguientes condiciones: que se compruebe la exposición del trabajador a violencia laboral y/o factores de riesgo psicosocial, que el centro de trabajo reciba quejas por prácticas opuestas al entorno organizacional favorable , denuncias por actos de violencia laboral y por último que existan signos o síntomas que denoten alguna alteración a la salud del trabajador[5].

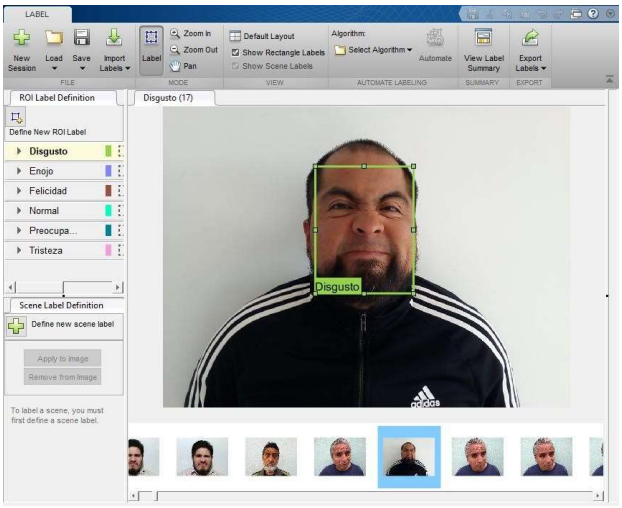


Fig. 1. Etiquetado en Image Labeler de Matlab.



Fig. 2. Arquitectura ResNet-18 con YOLO v2.

Se propone una herramienta que reconoce emociones en los individuos por medio de sus expresiones faciales, siendo estas clasificadas en seis categorías: disgusto, enojo, preocupación, tristeza, felicidad y normal. Las primeras cuatro categorías son consideradas negativas asociadas al estrés y las dos restantes positivas. Las emociones son un indicador de la salud del ser humano que pueden denotar síntomas de posibles trastornos. Ya que las emociones dirigen gran parte de nuestra vida, con ellas tomamos la mayoría de las decisiones; determinan nuestra atención, memoria, percepción, la forma en la que nos relacionamos con otras personas y con nuestro entorno, su

Tabla 1. Imágenes ocupadas para el entrenamiento.

Categoría	Número de Imágenes
Disgusto	419
Enojo	425
Felicidad	448
Normal	433
Preocupación	421
Tristeza	412

Tabla 2. Imágenes ocupadas para la prueba.

Categoría	Número de Imágenes
Disgusto	48
Enojo	56
Felicidad	43
Normal	45
Preocupación	41
Tristeza	48

consideración resulta ineludible al momento de determinar si se encuentra bajo el influjo de algún riesgo psicosocial y por tanto la probabilidad de cometer algún accidente laboral sea mayor [6].

Es conveniente señalar que esta herramienta no permite diagnosticar una enfermedad o trastorno, sino que sólo ayuda a identificar la necesidad de que un trabajador sea canalizado para que reciba la atención correspondiente. Por el momento únicamente es un sistema de detección de emociones ligadas a los accidentes laborales, por medio de redes neuronales ocupando técnicas de Deep Learning y YOLOv2 (You Only Look Once V2) en un sistema de visión artificial mediante MATLAB con un alto grado de eficacia. El nombre preliminar de este sistema es: "EstresNet". Se prevé que este sistema sea ideal para implementarse en centros de trabajo donde la operación tenga una alta dependencia del capital humano.

La siguiente fase de desarrollo contempla un software que recabe la información generada por el sistema de visión artificial, así como otros datos estadísticos del personal, relativo a su productividad e incidencias para hacer un análisis comparativo que permita tener un panorama más amplio de su desempeño. Este programa será administrado por el área de recursos humanos, le permitirá actuar adecuadamente, así como prever riesgos derivados de conductas patológicas no tratadas, de esta manera también, incrementar la utilidad, a la vez que se reducen los gastos de seguro médico por un personal sano y con la atención oportuna.

2. Trabajos relacionados

México es uno de los países que cuenta con un alto índice de trabajadores con estrés laboral, ya que las organizaciones no cuentan con las condiciones necesarias para el

buen desarrollo de los trabajadores. Según un estudio realizado por la Universidad Nacional Autónoma de México, el 85% de las organizaciones mexicanas no cuentan con las condiciones adecuadas para que sus colaboradores tengan un balance entre vida personal y trabajo, lo cual propicia trastornos físicos y psicológicos [7].

Sin embargo, no hay investigaciones relacionadas o dirigidas a la aplicación de la detección de emociones en un entorno laboral, ya que los trabajos presentados tienen un enfoque al reconocimiento de emociones por medio de fotografías, orientados a la parte de interacción social, como el cuidado de personas mayores en un ambiente inteligente o para apoyar la didáctica de enseñanza aprendizaje [8].

La mayoría de estas investigaciones han sido desarrolladas en otros países como España, Ecuador y Brasil, desde el año 2015; algunos han sido desarrollados haciendo uso de Python [9].

En México no se han desarrollado gran cantidad de investigaciones o proyectos en el área de reconocimiento de emociones, los que se encuentran disponibles han sido realizadas por áreas psicológicas solo implementando el reconocimiento facial por imágenes [10].

Un artículo publicado por CIO México menciona que la visión artificial tendrá un gran impacto en la fabricación industrial en años próximos, haciendo uso de chips, software, y estándares de Machine Learning [11], sin embargo, esto tiene un enfoque más al aumento y mejora en la producción industrial, dejando fuera los aspectos sociales dentro de esta.

Esto denota que las investigaciones realizadas en el país van dirigidas a la parte industrial como en robótica, donde se hace uso de Machine Learning o Deep Learning, cabe mencionar que las publicaciones encontradas han sido de universidades, tales como UNAM e IPN, que cuentan con recursos para el desarrollo de estos proyectos, ya que el hacer uso del reconocimiento facial por medio de imágenes, demanda recursos computacionales con los cuales no todos cuentan debido al alto costo, siendo este un impedimento para el desarrollo, sin mencionar, el hecho de la adquisición de conocimientos de una programación que permita el desarrollo de un código, una base de datos adecuada a los requerimientos, un buen etiquetado, entrenamientos, etc. Representando un área de oportunidad en el desarrollo del proyecto, enfocado a una parte importante del sector industrial, que es el capital humano.

3. Metodología y materiales

3.1. Categorías

Para la elaboración de EstresNet se designaron 6 categorías a identificar, tomando en cuenta que el estrés se puede presentar de varias formas, en este caso, nos basamos en las expresiones faciales para conocer su estado emocional.

Preliminarmente se tomaron en cuenta 4 emociones negativas (tristeza, preocupación, disgusto y enojo) y 2 emociones positivas (felicidad y normal), con la finalidad, de poder clasificar el estado emocional de las personas.

Tabla 3. Opciones de entrenamiento.

Opciones de Entrenamiento	Configuración Seleccionada
Solver	adam
MiniBatchSize	32
InitialLearnRate	0.0001
MaxEpochs	100
LearnRateDropFactor	0.01
LearnRateDropPeriod	30
Shuffle	every-epoch
DispatchInBackground	true
ExecutionEnvironment	auto

Tabla 4. Características de la computadora usada para el entrenamiento.

Tarjeta Gráfica	Procesador	Memoria RAM
4GB GeForce GTX 1050Ti	AMD RYZEN 5 3400G	16GB

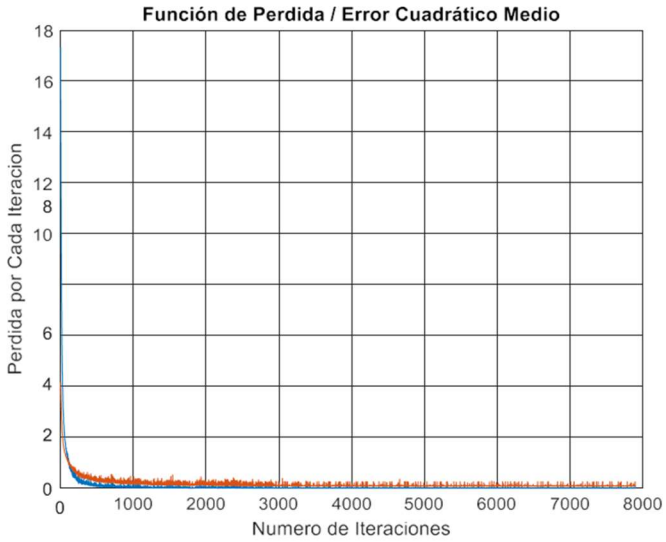


Fig. 3. Función de perdida y error cuadrático medio con solver “adam” a 100 épocas.

3.2. Base de datos (entrenamiento y test)

Para el entrenamiento de la red se ocuparon 2558 imágenes (Tabla 1), las cuales se obtuvieron con la colaboración de amigos y familiares. Para el test de la red, se ocuparon 281 imágenes (Tabla 2), estás se obtuvieron de la misma forma que las imágenes de entrenamiento.

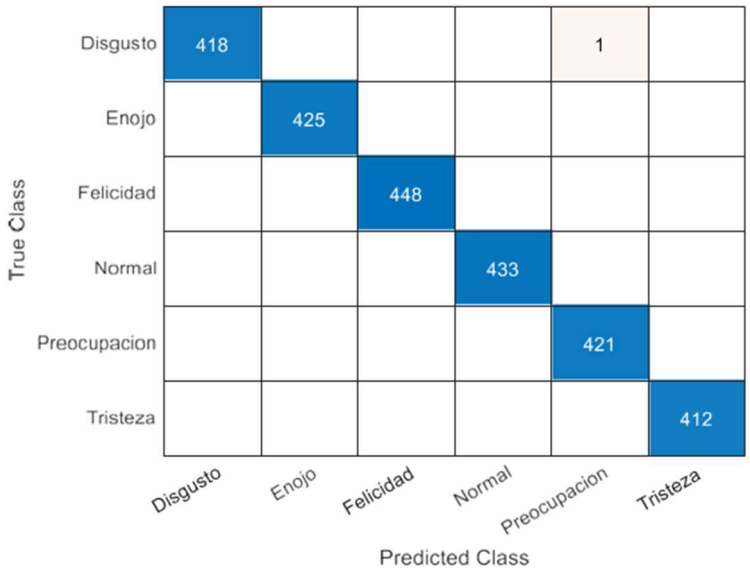


Fig. 4. Matriz de confusión (entrenamiento).

Tabla 5. Comparación de imágenes (entrenamiento).

Categoría	Número Total de Imágenes	Número Total de Imágenes Entrenadas
Disgusto	419	419
Enojo	425	425
Felicidad	448	448
Normal	433	433
Preocupación	421	421
Tristeza	412	412

Tabla 6. Total de imágenes vs. total de imágenes entrenadas.

Número Total de Imágenes	**Número Total de Imágenes Entrenadas**
2558	2558

3.3. Métodos propuestos para el desarrollo

Una red neuronal convolucional [12] (CNN o ConvNet) es una arquitectura de red para Deep Learning que aprende directamente de los datos, sin necesidad de extraer características manualmente. Las CNN proporcionan una arquitectura óptima para descubrir y aprender características principales en imágenes y datos de series temporales. Para este proyecto se ocupó como red convolucional "ResNet-18", ya que esta es muy eficaz respecto a el entrenamiento y su coste computacional es medio-alto,

esto la hace mejor comparada con otras redes. ResNet-18, es una red neuronal convolucional que tiene 18 capas de profundidad.

Puede cargar una versión previamente entrenada de la red entrenada en más de un millón de imágenes de la base de datos ImageNet [13]. Con esto la red entrenada puede clasificar imágenes en diversas categorías de objetos, animales, etc. Se implementó un detector YOLO (You Only Look Once) v2 [14] a la red ResNet- 18, el cual reconoce objetos específicos en imágenes, basándose en las imágenes de entrenamiento, las cuales fueron etiquetadas en la aplicación Image Labeler [15] de Matlab, esta aplicación permite etiquetar datos reales en una colección de imágenes.

En la imagen se muestra un ejemplo del cómo se realizó la etiqueta a cada una de las imágenes, de lado izquierdo se pueden observar las diferentes categorías que conforman a este proyecto.

3.4. Entrenamiento

Se llevaron a cabo diferentes entrenamientos para conocer la eficacia y precisión del funcionamiento de EstresNet, para esto, se ocuparon las siguientes opciones de entrenamiento:

En los diferentes entrenamientos que se llevaron a cabo, se utilizó también el solver "sgdm", sin embargo, los entrenamientos realizados con este, no se obtuvieron buenos resultados y en algunos casos no se detectaba ninguna emoción, incluso utilizando las mismas imágenes que se utilizaron para el entrenamiento.

Por lo cual, se decidió cambiar de solver, además de cambiar los valores de ciertos aspectos de las opciones de entrenamiento, como las épocas y el MiniBatchSize. En total, se realizaron 8 entrenamientos, de los cuales, obtuvimos mejores resultados con aquellos que se realizaron con el solver "adam". Para realizar los entrenamientos se utilizó una computadora de escritorio con las siguientes características (Tabla 4).

Una vez, teniendo el entrenamiento adecuado, se obtuvo la representación gráfica de la función de perdida y error cuadrático medio (Fig. 3.) y la matriz de confusión respectiva (Fig. 4.).

Comparando la matriz de confusión con la cantidad de imágenes totales con las que se llevó a cabo el entrenamiento, obtuvimos los siguientes resultados, los cuales se muestran en la Tabla 5 y Tabla 6.

Las imágenes no entrenadas son aquellas que al momento del entrenamiento no se encontró ninguna relación con los datos y, por ende, estas no fueron consideradas para el entrenamiento. Pudimos observar que, para el entrenamiento, ninguna imagen fue excluida o tomada como no entrenada, por lo cual podemos decir que la precisión de entrenamiento para cada categoría es de un 100%.

3.5. Test

Una vez hecho el entrenamiento, procedimos a probar el funcionamiento de EstresNet, por medio del procesamiento de las 281 imágenes que conforman el test. De lo cual obtuvimos los siguientes resultados, los cuales se muestran en la Fig. 5.

Comparando la matriz de confusión con el número total de imágenes detectadas (Tabla 7 y Tabla 8) pudimos obtener la precisión de test por cada categoría (Tabla 9).

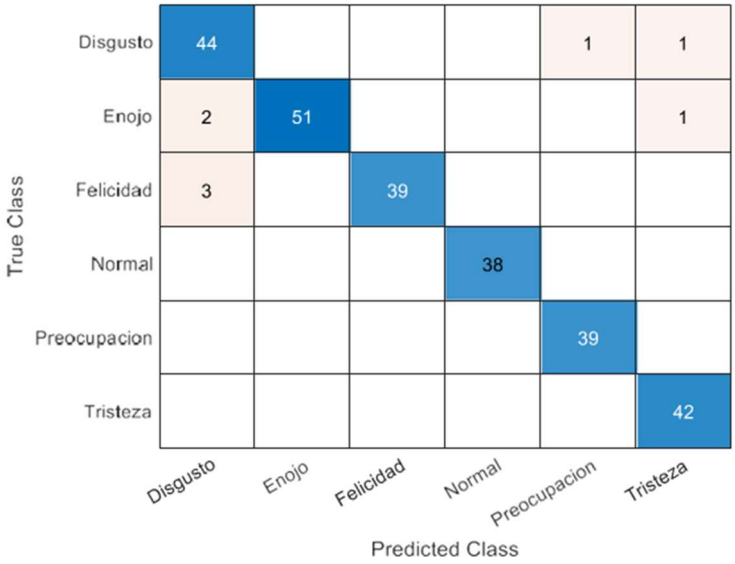


Fig. 5. Matriz de confusión (Test).

Tabla 7. Comparación de imágenes (Test).

Categoría	Número Total de Imágenes	Número Total de Imágenes Entrenadas
Disgusto	48	46
Enojo	56	54
Felicidad	43	42
Normal	45	38
Preocupación	41	39
Tristeza	48	42

Tabla 8. Total, de Imágenes Vs Total de Imágenes Entrenadas.

Número Total de Imágenes	Número Total de Imágenes Entrenadas
281	261

3.6. Resultados

Con relación a los resultados de las imágenes del test, se muestran a continuación algunos resultados de la detección de estas por medio de EstresNet.

4. Conclusiones y trabajo a futuro

El desarrollo de EstresNet se implementó utilizando varias técnicas para detectar las emociones que reflejan las personas por medio de sus expresiones faciales. El objetivo

Tabla 9. Precisión del test por categoría.

Categoría	Precisión
Disgusto	99.525%
Enojo	99.5317%
Felicidad	99.7773%
Normal	98.4091%
Preocupación	99.5272%
Tristeza	98.5646%



Fig. 6. Detección de cada categoría.

del proyecto pretende ser una alternativa para la detección y prevención de posibles riesgos vinculados con el estado emocional y psicosocial de las personas.

Realizando la comparación del funcionamiento de EstresNet con las estrategias implementadas hoy en día, se llega a la conclusión de que el desempeño de EstresNet tiene una mayor eficiencia si se usa de manera adecuada, de forma alterna, analizando los beneficios generados, se observa que diversas empresas sin importar el tamaño que estén constituidas, no cuentan con sistemas o estrategias para prevenir accidentes a causa de las emociones de los trabajadores, ocasionando un riesgo inminente ante los trabajados de las empresas.

Investigando y relacionando la NOM-035-STPS-2018, se genera la incertidumbre acerca de cómo los factores ambientales y del entorno laboral podrían ser determinantes en la conducta y emociones del trabajador, y cómo se podría diferenciar los problemas causados por estos entornos considerando aquellos que son por problemas emocionales como el estrés, ansiedad, depresión, tristeza, etc., Por lo cual, se concluye que la implementación de EstresNet dependerá del capital humano adecuado los cuales se encargarán de llevar un seguimiento de los trabajadores, tanto de su estado emocional como su desempeño laboral y social, dando seguimiento y teniendo una mejor gestión en el estado emocional de los trabajadores.

Primordialmente, este proyecto se pensó para su uso dentro de la industria, principalmente, para empresas que dependan del capital humano y su efectividad, sin

embargo, analizando las oportunidades que este podría tener, se ha contemplado la posibilidad de aumentar las categorías de EstresNet, teniendo un mayor rango para su uso, generando la posibilidad de incorporación al sector social, escolar, comercial, dentro de aeropuertos, etc.

El trabajo futuro para mejorar el rendimiento de EstresNet debe ser relacionado directamente con el área de seguridad e higiene organizacional que podría llevar el seguimiento del estado emocional de los trabajadores mexicanos y prevenir riesgos de trabajo con base a la NOM-035-STPS-2018. En adición a esto, es posible ampliar la base de datos con fotografías de personas con diferentes rasgos para poder tener un mayor rango de detección.

Referencias

1. Patlán-Pérez, J.: Factores psicosociales, clasificación, identificación y consecuencias en la salud laboral. Instituto Mexicano del Seguro Social, http://www.imss.gob.mx/sites/all/statics/salud/estreslaboral/1erjornada/06-Factores-Consecuencias.pdf, (Año desconocido)
2. Comisión Nacional de los Derechos Humanos: Destaca CNDH importancia de la cultura de la prevención, para revertir más de 500,000 accidentes laborales, 12,622 enfermedades profesionales y 1,408 fallecimientos. https://www.cndh.org.mx/sites/default/files/documentos/2019-04/Com_2019_159.pdf, (2019)
3. Organización Mundial de la Salud (OMS): Día mundial de la salud mental 2017. La salud mental en el lugar de trabajo, https://www.who.int/mental_health/world-mental-health-day/2017/es/ (2017)
4. Campa-Cifrián, R. R.: Factores de riesgo psicosocial en el trabajo - Identificación, análisis y prevención. Secretaría del Trabajo y Previsión Social: Norma Oficial Mexicana NOM-035-STPS-2018, http://asinom.stps.gob.mx:8145/ upload/nom/48.pdf (2018)
5. Gobierno de México: Norma Oficial Mexicana NOM-035-STPS-2018, Factores de riesgo psicosocial. Identificación, análisis y prevención, https://www.gob.mx/cms/uploads/attachment/file/503381/NOM035_guia.pdf (2019)
6. González, I.: Neuromarketingla: Qué son los análisis de expresión facial y cómo funcionan. https://neuromarketing.la/2016/12/los-analisis-expresion-facial-funcionan/ (2016)
7. Dirección General de Comunicación Social: 374 Millones de accidentes en el trabajo al año en el mundo. https://www.dgcs.unam.mx/boletin/bdboletin/2020_379.html (2020)
8. Monasor, E. L.: Detección de emociones orientada a mejorar la calidad de vida y el cuidado de personas mayores en ambientes inteligentes. Universidad de Castilla, La Mancha: https://ruidera.uclm.es/xmlui/bitstream/handle/10578/23320/TESIS%20Lozano%20Monasor.pdf?sequence=1&isAllowed=y (2019)
9. Lozano, G.: Reconocimiento de emociones (eigenfaces, fisherfaces, LBPH) Python-OpenCV. OMES: https://omes-va.com/reconocimiento-de-emociones-opencv-python/#comments (2020)
10. Iglesias-Hoyos, S., del Castillo Arreola, A., Muñoz-Delgado, J. I.: Reconocimiento facial de expresión emocional: Diferencia por licenciaturas. Universidad Nacional Autónoma de México: https://www.redalyc.org/pdf/3589/358950159003.pdf (2016)
11. Cortés, M.: El impacto de la visión artificial en la fabricación industrial se acentuará en los próximos años. CIO México: https://cio.com.mx/el-impacto-de-la-vision-artificial-en-la-fabricacion-industrial-se-acentuara-en-los-proximos-anos/ (2020)
12. MathWorks: Redes neuronales convolucionales. 3 Cosas que Necesitas Saber, https://la.mathworks.com/discovery/convolutional-neural-networkmatlab.html?s_tid=srchtitle

13. ImageNet, http://www.image-net.org
14. MathWorks, Yolov2ObjectDetector, https://la.mathworks.com/help/vision/ref/yolov2objectdetector.html
15. MathWorks. Image Labeler, https://la.mathworks.com/help/vision/ ref/imagelabeler-app.html?searchHighlight=image%20labeler&s_tid=srchtitle

ISSN 1870-4069

Amputación de datos de registros geofísicos con mecanismo de pérdida MCAR

César E. Santos-Vázquez, Leonardo Morales-Collado,
Juana Canul-Reich, José Hernández-Torruco

Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencia y Tecnologías de la Información,
México

{csv.santos92, lmc580, jcanulreich}@gmail.com,
jose.hernandezt@ujat.mx

Resumen. El tratamiento de datos faltantes en el área de registros geofísicos es una tarea crucial, por lo cual el desarrollo de una metodología que simule la presencia de los mismos, como lo es la amputación de datos, permite contar con un punto de referencia a la hora de implementar nuevas técnicas enfocadas en la solución de este tipo de situaciones. En esta investigación se propone un flujo de trabajo para el análisis, procesado y amputación de datos de registros geofísicos. Mediante este análisis se obtuvo una muestra de los datos amputados con el mecanismo de pérdida MCAR y se comparó con las distintas distribuciones logísticas para el mismo mecanismo. En esta propuesta se utiliza tanto un lenguaje de código abierto, así como librerías accesibles para todo público, por lo cual replicar esta experimentación no presenta ningún inconveniente.

Palabras clave: Amputación, datos perdidos, datos nulos, registros geofísicos.

Well Logs Data Amputation with MCAR Loss Mechanism

Abstract. The treatment of missing data in the area of geophysical well logs is a crucial task, which is why the development of a methodology that simulates the presence of the same, such as the data amputation, allows to have a point of reference at the time to implement new techniques focused on solving this type of situation. In this research, a workflow is prposed for the analysis, processing and data amputation from geophysical well logs. Through this analysis, a sample of the amputee data with the MCAR loss mechanism was obtained and compared with the different logistic distributions for the same mechanism. In this proposal, an open source language is used, as well as libraries accesible to all audiences, so replicating this experiment does not present any inconvenience.

Keywords: Amputation, missing data, null data, well logs.

1. Introducción

El manejo de valores faltantes en un set de datos es una problemática típica de la etapa de preprocesado. El autor [5] deja en claro las relaciones existentes entre las variables y su variabilidad exponiendo ejemplos sobre como se relaciona una variable con la taza de cambio de la otra. En el área de registros geofísicos también existen procesos que necesitan la preparación de los datos para poder realizar estudios o interpretaciones petrofísicas.

La pérdida de datos en curvas de registros geofísicos es un problema común que se debe corregir, cuando es posible, durante la etapa de control de calidad. Algunas de las causas de estas pérdidas de información se deben a las condiciones físicas que pueden sufrir los equipos de medición de registros, llevándolos a un mal funcionamiento, a las condiciones y características del agujero, a la velocidad excesiva en el momento en que se toma el registro, al manejo de la información previo a ser entregada para realizar los estudios, entre otras, como menciona [3].

La simulación de datos faltantes, permite desarrollar y evaluar métodos que abordan esta problemática. Contar con un procedimiento de amputación de datos que pueda aplicarse sobre datos completos es importante, ya que en primera instancia (antes de la etapa de procesado) estos suelen contar con valores faltantes. Particularmente la información de registros geofísicos es privada para algunas empresas, por lo que conseguir datasets públicos puede ser una tarea difícil.

En los casos de las empresas que sí hacen públicos sus datasets, estos ya están preprocesados y típicamente sus posibles valores faltantes ya han sido resueltos. Este estudio es parte de una investigación más amplia, para la cual no se disponía de datasets con valores faltantes, por lo que se decidió identificar una técnica para generarlos, es decir realizar amputación de datos. Al realizar la revisión de la literatura relacionada se observó que no se cuenta con una metodología establecida para la amputación de datos de registros geofísicos.

Por lo tanto, establecer tal metodología, que sirva de referencia para futuras investigaciones en la industria energética y en el área de investigación científica, es la principal contribución de este manuscrito. En el articulo [9] se menciona que algunas de las prácticas actuales para la amputación de datos no son del todo correctas, ya que la validez de la metodología y la estadística aplicada en dichos métodos de amputación pueden no ser apropiadas para dicho propósito.

Finalmente concluye que la amputación multivariable, suele generar amputaciones mucho más fiables, al mismo tiempo que permite regular la aplicación del método de amputación utilizado. En la literatura actual [7], señala que el proceso de amputación de datos es la base para los experimentos de imputación, pero si no se realiza adecuadamente puede resultar en conclusiones inválidas. Propone utilizar el desafiante mecanismo de pérdida, MCAR, para trabajos futuros.

Por otra parte, en [1], se menciona que los datos faltantes mas allá de ser casos aislados, son prácticamente un regla en la vida real. Se hace mención a estudios publicados durante los años 1998 a 2004, en los cuales se encontró que el 48 % de los estudios presentó datos faltantes, y el 36 % no, quedando un restante indeterminado. De este modo se justifican los esfuerzos realizados con el fin de mejorar los métodos para tratar este tipo de problemáticas.

Tabla 1. Definición de probabilidades de pérdida según el mecanismo de pérdida. Donde y: matriz de datos entera, y_{obs}: observaciones de y, y_{mis}: observaciones perdidas, R: matriz de pérdida y q: vector de parámetros que describen la relación entre la pérdida, R y el conjunto de datos y.

Mecanismo de pérdida	Definición de probabilidad
MNAR	$p(R\|y_{obs}, y_{mis}, q)$
MAR	$p(R\|y_{obs}, q)$
MCAR	$p(R\|q)$

Este artículo está estructurado de la siguiente manera: En la Sección 2 se realiza una definición formal de los mecanismos de pérdida involucrados en el manejo de datos faltantes. La Sección 3 presenta el conjunto de datos utilizado, algunas características de su estructura y formato; muestra el hardware y software utilizado para realizar los experimentos y establece la elección del mecanismo de pérdida para este experimento.

En la Sección 4 se define la implementación del método de amputación de datos aplicado al conjunto de datos utilizado, empezando por el preprocesado de la información para adecuarla a las necesidades del experimento, seguido del desarrollo del método de amputación y finalizando con una verificación gráfica de los datos ya amputados.

La Sección 5 presenta los datos amputados, comparando los resultados del método utilizando distintos tipos de distribución de probabilidad. Finalmente, la Sección 6 concluye el artículo resaltando la importancia del uso del mecanismo de pérdida MCAR para la amputación de datos de registros geofísicos y propone algunas direcciones para futuras investigaciones.

2. Mecanismo de pérdida para amputación de datos

La investigación realizada se centra en la generación y posterior análisis de datos faltantes en datos de registros geofísicos. Regularmente las herramientas presentes en software petrofísico ofrecen una solución no del todo útil como lo es la imputación de datos faltantes utilizando la media de dicho atributo en el conjunto de datos, del mismo modo los enfoques clásicos en otras disciplinas ofrecen desde la eliminación de las instancias con presencia de valores nulos hasta la imputación por medio de una regresión lineal.

Por lo tanto, contar con una metodología para la experimentación en esta disciplina podría ser de gran ayuda para referenciar futuras experimentaciones o propuestas de solución a este tipo de problemática, al tener un punto de comparación.

2.1. Mecanismos de pérdida en datos faltantes

En primera instancia se debe definir el mecanismo de pérdida bajo el cual sería lo más lógico que se presentaran datos faltantes en un conjunto con respectiva naturaleza del mismo. Fox et al [2] definen tres categorías para clasificar los datos faltantes:

1. **MCAR** (Missing completely at random): cuando la probabilidad de que un dato se pierda no está relacionada a ninguna variable en el conjunto de datos.

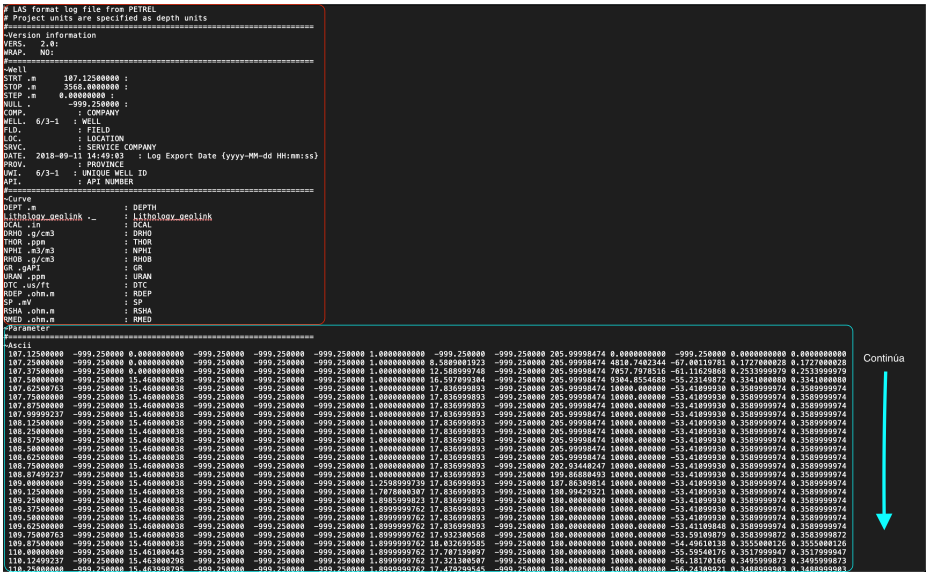


Fig. 1. Vista de un archivo con extensión `.las`.

2. **MAR** (Missing at random): cuando la probabilidad de que hagan falta datos en una variable está relacionada con otra(s) variable(s) del conjunto de datos.
3. **MNAR** (Missing not at random): sucede cuando la probabilidad de la pérdida de datos está relacionada con la variable en sí misma.

A pesar de que estos mecanismos tienen nombres que podrían sugerir una causa probable de datos perdidos, el concepto realmente engloba la relación estadística entre las observaciones (variables) y la probabilidad de pérdida de datos. En la tabla 1 se muestran las definiciones formales (matemáticas) de los mecanismos de pérdida.

Los autores en [4] ejemplifican tres tipos de mecanismos de pérdida. Basado en esa diferenciación se puede clasificar la problemática de este estudio (gaps en registros geofísicos) dentro del mecanismo de pérdida MCAR (missing completely at random), debido a que la probabilidad de que se presenten datos nulos es la misma para todo el conjunto de datos.

3. Métodos y materiales

3.1. Conjunto de datos

El conjunto de datos a utilizar pertenece al proyecto de plataforma continental de Noruega realizado por la empresa Geolink, el cual contiene datos de más de quinientos pozos. Estos datos contienen información geológica y petrofísica sobre hidrocarburos del sector noruego. Los datos se encuentran disponibles a través de GitHub en el repositorio de Lukas Mosser[1].

[1] https://github.com/LukasMosser/geolink_conjuntodedatos

```
$ VERSION   : num 2
$ WELL      :'data.frame':    13 obs. of  4 variables:
 ..$ MNEM       : chr [1:13] "STRT" "STOP" "STEP" "NULL" ...
 ..$ UNIT       : chr [1:13] "m" "m" "m" "" ...
 ..$ VALUE      : chr [1:13] "107.12500000" "3568.0000000" "0.00000000" "-999.250000" ...
 ..$ DESCRIPTION: chr [1:13] "" "" "" "" ...
$ CURVE     :'data.frame':    14 obs. of  4 variables:
 ..$ MNEM       : chr [1:14] "DEPT" "Lithology_geolink" "DCAL" "DRHO" ...
 ..$ UNIT       : chr [1:14] "m" "_" "in" "g/cm3" ...
 ..$ API.CODE   : chr [1:14] "" "" "" "" ...
 ..$ DESCRIPTION: chr [1:14] " DEPTH" " Lithology_geolink" " DCAL" " DRHO" ...
$ PARAM     : NULL
$ OTHER     : chr ""
$ LOG       :'data.frame':    27688 obs. of  14 variables:
 ..$ DEPT            : num [1:27688] 107 107 107 108 108 ...
 ..$ Lithology_geolink: num [1:27688] NA NA NA NA NA NA NA NA NA ...
 ..$ DCAL            : num [1:27688] 0 0 0 15.5 15.5 ...
 ..$ DRHO            : num [1:27688] NA NA NA NA NA NA NA NA NA ...
 ..$ THOR            : num [1:27688] NA NA NA NA NA NA NA NA NA ...
 ..$ NPHI            : num [1:27688] NA NA NA NA NA NA NA NA NA ...
 ..$ RHOB            : num [1:27688] 1 1 1 1 1 1 1 1 1 1 ...
 ..$ GR              : num [1:27688] NA 8.58 12.59 16.6 17.84 ...
 ..$ URAN            : num [1:27688] NA NA NA NA NA NA NA NA NA ...
 ..$ DTC             : num [1:27688] 206 206 206 206 206 ...
 ..$ RDEP            : num [1:27688] 0 4811 7058 9305 10000 ...
 ..$ SP              : num [1:27688] NA -67 -61.1 -55.2 -53.4 ...
 ..$ RSHA            : num [1:27688] 0 0.173 0.253 0.334 0.359 ...
 ..$ RMED            : num [1:27688] 0 0.173 0.253 0.334 0.359 ...
$ PATH      : chr  "**************************************** /6_3-1.las"
$ ATTRIBUTES:'data.frame':    1 obs. of  8 variables:
 ..$ well       : chr "6/3-1"
 ..$ null       : num -999
 ..$ start      : num 107
 ..$ start_units: chr "m"
 ..$ stop       : num 3568
 ..$ step       : num 0
 ..$ step_units : chr "m"
```

Fig. 2. Estructura de un archivo `.las` en R Studio.

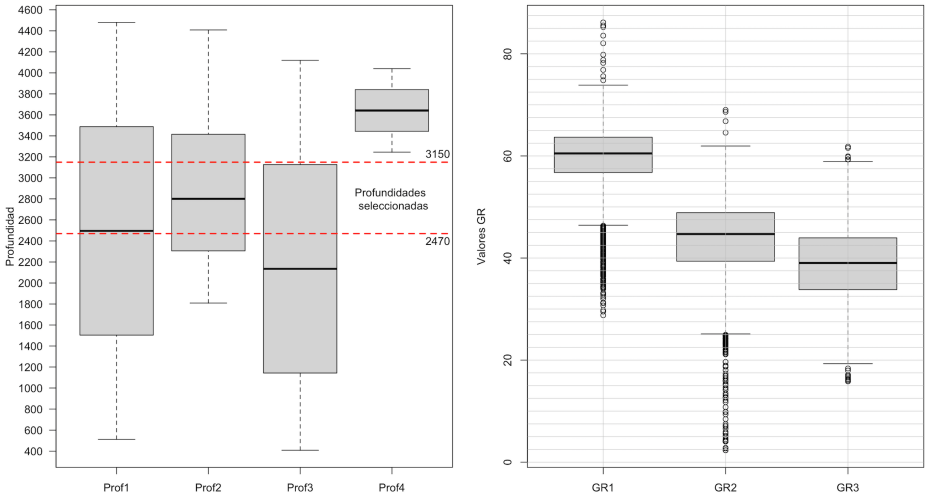
La información de cada pozo se almacena en un archivo con formato **.LAS**, dentro de tal archivo se encuentran los parámetros para diversas respuestas de señal (curvas). La figura 1 muestra la estructura interna de un archivo **.LAS**, el recuadro rojo delimita la zona de encabezado del pozo, dentro de esa zona se puede encontrar información como, por ejemplo, la versión del formato, el inicio y fin del pozo, la localización, fecha, nombre y se hace especial referencia a la nomenclatura utilizada para los datos faltantes o nulos, señalada con el número negativo “-999.250000”.

Del mismo modo se observa que dentro del encabezado de pozo (recuadro rojo) se encuentra información sobre el arreglo de curvas (señales) que incluye este archivo, estas situadas en la parte inferior del encabezado (apartado señalado como curve). Por otra parte en el recuadro azul se muestran los datos apilados por columna, estas columnas llevan un orden descendente (como se encuentra indicado en el apartado curve).

3.2. Hardware

Para la realización de esta investigación se utilizó un equipo de computo Apple MacBook Air 13” 2018 el cual cuenta con las siguientes especificaciones:

– Procesador: 1.6 GHz Intel Core i5 de dos núcleos.
– Memoria: 8GB 2133 MHz LPDDR3.
– Almacenamiento: 256 GB SSD.
– Gráficos: Intel UHD Graphics 617, 1536 MB.
– Sistema Operativo: macOs Catalina 10.15.4.



(a) (b)

Fig. 3. Distribución de datos en el intervalo seleccionado.

Tabla 2. Proporción de datos nulos generados.

Proporción	Nulos
3 %	139
10 %	443
15 %	683
25 %	1084
30 %	1301

3.3. R y RStudio

El lenguaje utilizado fue R en su versión 4.0.1. Por otro lado la utilización del lenguaje se realizó a través del IDE (entorno de desarrollo integrado) RStudio [6], con la versión Orange Blossom 1.2.5033.

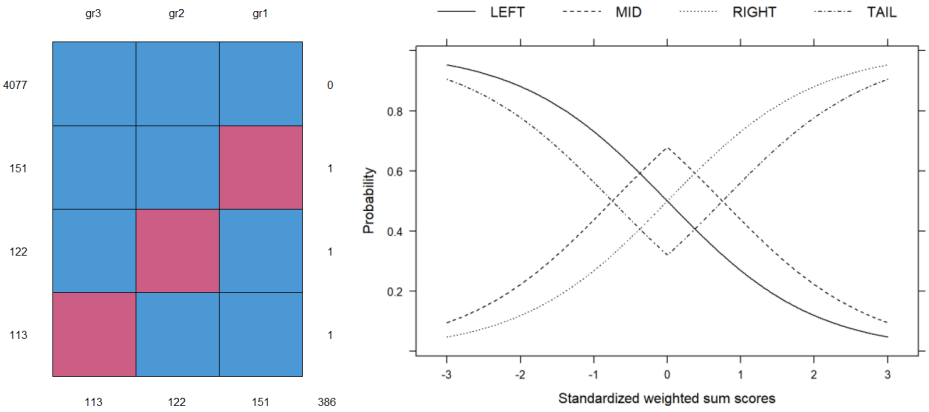
3.4. Missing Completely at Random (MCAR)

Dentro de los mecanismos de pérdida, se denomina como Missing Completely at Random (MCAR) a aquellas situaciones en las que una matriz de pérdida **M** no presenta ninguna relación con la matriz de datos **X**, como se aprecia en [7] y los parámetros que describen la relación entre ambas matrices **q**.

Atendiendo a la naturaleza de los datos, de registros geofísicos, este mecanismo es el que mejor se ajusta a un panorama real, ya que la aleatoriedad de las situaciones presentes durante la recolección de los registros son situaciones muy comunes y se presentan con facilidad:

$$p(\mathbf{M}|\mathbf{X},\mathbf{q}) = p(\mathbf{M}|\mathbf{q}). \tag{1}$$

Del mismo modo la configuración y la tasa de pérdida, son dos factores muy importantes cuando se amputan valores a un conjunto de datos.



(a) Patrón de pérdida del 3 %. (b) Distribución logística.

Fig. 4. Patrón de Pérdida y Tipos de Distribución Logística.

La configuración se refiere a si los datos faltantes se presentan solamente en una variable o se presentan en dos o más variables del conjunto de datos, pudiendo ser univariable o multivariable. Por otra parte la tasa de pérdida representa el porcentaje de datos faltantes, es posible referirse a estos por variable o por el conjunto de datos completo.

Por ejemplo si se tuviera un conjunto de datos constituido por una matriz de 5 variables y 100 observaciones ($X[i_{1..,100},\ j_{1..,5}]$) y cada variable tuviera 20 valores faltantes, la tasa de pérdida sería 20 %, ya que si se cuenta con 100×5 se tendrían 500 valores en el conjunto de datos, de los cuales $20 \times 5 = 100$ son faltantes, esto es $100/500 = 0{,}2$ es decir, el 20 % de los valores presentes en la matriz de datos **X** son valores faltantes.

4. Implementación del método de amputación

4.1. Preprocesado de datos

La librería **lastools** permite utilizar archivos con extensión **.LAS** en R. La figura 2 muestra la estructura interna de un archivo con esta extensión dentro del ambiente. Los datos se extrajeron en un dataframe, que contiene los valores del registro y su correspondiente columna de profundidad.

En la figura 3a se muestra un boxplot en el cual se comparan las profundidades, donde se observa que los pozos 1, 2 y 3 coinciden en el intervalo de los 2470 a los 3150 metros, por lo cual se utilizó esta sección para delimitar un conjunto de datos de trabajo. Por otra parte en la figura 3b se muestra el boxplot de los valores de la curva GR de cada pozo seleccionado, en el intervalo definido anteriormente.

4.2. Amputación de datos

La función **ampute()** de la librería **mice** permite realizar la amputación multivariable, es decir amputar datos en las diferentes variables del dataset, de una sola vez y seleccionar el mecanismo, patrón y proporción de pérdida de datos.

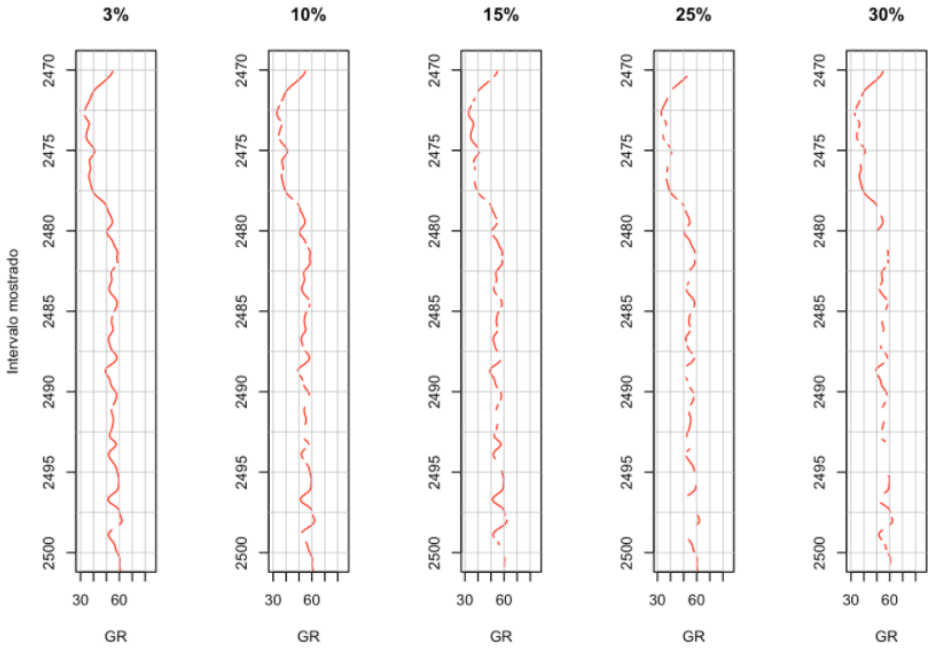


Fig. 5. Curvas con diferentes porcentajes amputados.

Genera la pérdida de datos deseada, de acuerdo a los parámetros indicados. Las proporciones de pérdida para este experimento fueron del 3, 10, 15, 25 y 30 por ciento, con un mismo tipo de distribución (**MID**). Es importante mencionar que la función **ampute()** no admite conjuntos con NA's. El resultado es un objeto de clase **mads** (multivariate amputed data set), el cual contiene los datos amputados, los originales, el patrón de pérdida y el tipo de distribución.

En la figura 4a se observa el patrón de pérdida generado. La primer fila representa todos los casos completos, mientras que la segunda fila representa 151 casos en los que la variable "gr1" contiene datos faltantes, la tercer fila representa 122 casos incompletos para la variable "gr2" y finalmente la cuarta fila representa 113 casos en los que la variable "gr3" cuenta con faltantes. Los datos faltantes se muestran en rojo y los datos observados en azul.

Otro parámetro importante para la función **ampute()** es **type**, el cual como explican los autores en [9], se refiere al tipo de distribución de probabilidad logística, la cual puede ser de 4 tipos: LEFT, MID, RIGHT, TAIL. Según el valor que se escoja para este parámetro se determinará el comportamiento de la función **ampute()**, basado en la distribución logística. La figura 4b nos muestra la relación entre la distribución logística y el tipo de pérdida escogido.

Verificación de datos amputados. Una vez que el conjunto de datos fue amputado, se verifica si realmente se generaron Na´s en su contenido. El número de observaciones totales es de 4463 y se cuenta con 3 variables; la tabla 2 muestra la cantidad de datos amputados por porcentaje de pérdida indicado en la función **ampute()**.

Tabla 3. Localización de Valores Amputados.

Loc. Valores Amputados por porcentaje							Valores Amputados por porcentaje				
Prof.	**GR 3 %**	**GR 10 %**	**GR 15 %**	**GR 25 %**	**GR 30 %**	**GR_Orig**	**gap 3 %**	**gap 10 %**	**gap 15 %**	**gap 25 %**	**gap 30 %**
12850	54.86	54.86	54.86	54.86	54.86	54.86	×	×	×	×	×
12851	53.98	53.98	53.98	×	53.98	53.98	×	×	×	53.98	×
12852	52.48	52.48	52.48	52.48	52.48	52.48	×	×	×	×	×
12853	50.55	50.55	×	50.55	50.55	50.55	×	×	50.55	×	×
12854	48.42	48.42	48.42	48.42	48.42	48.42	×	×	×	×	×
12855	46.20	46.20	46.20	46.20	46.20	46.20	×	×	×	×	×
12856	43.95	43.95	43.95	43.95	43.95	43.95	×	×	×	×	×
12857	41.87	41.87	41.87	41.87	41.87	41.87	×	×	×	×	×
12858	40.24	40.24	40.24	×	40.24	40.24	×	×	×	40.24	×
12859	39.13	39.13	×	39.13	39.13	39.13	×	×	39.13	×	×
12860	38.32	38.32	×	×	38.32	38.32	×	×	38.32	38.32	×
12861	37.56	37.56	37.56	37.56	×	37.56	×	×	×	×	37.56
12862	36.72	36.72	36.72	36.72	36.72	36.72	×	×	×	×	×
12863	35.80	×	×	35.80	35.80	35.80	×	35.80	35.80	×	×
12864	34.83	34.83	34.83	34.83	34.83	34.83	×	×	×	×	×
12865	33.92	33.92	33.92	33.92	33.92	33.92	×	×	×	×	×
12866	33.16	33.16	33.16	33.16	×	33.16	×	×	×	×	33.16
12867	×	32.77	32.77	32.77	32.77	32.77	32.77	×	×	×	×
12868	33.05	33.05	33.05	33.05	33.05	33.05	×	×	×	×	×
12869	34.11	34.11	34.11	×	×	34.11	×	×	×	34.11	34.11

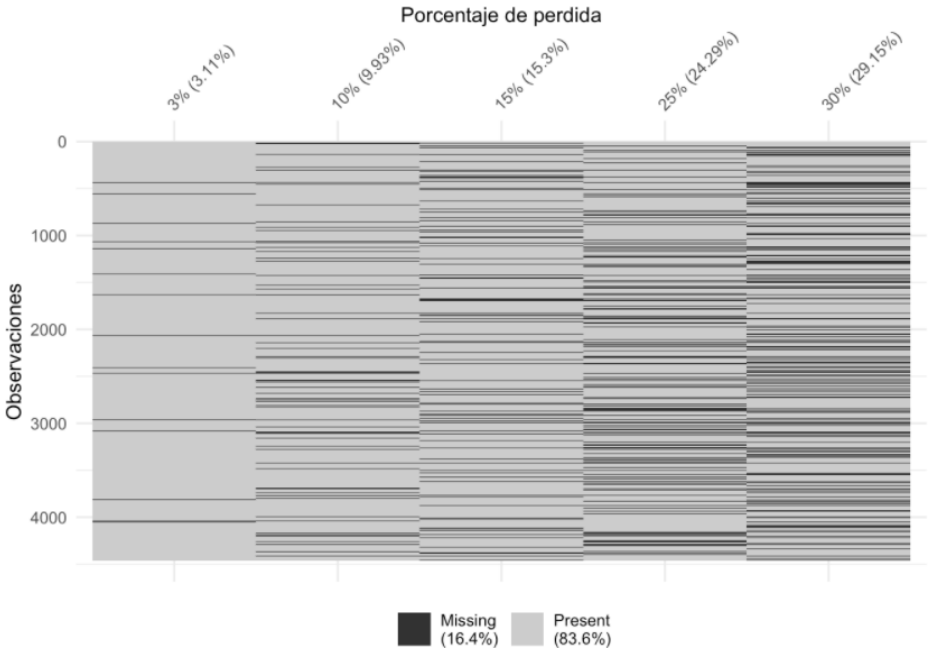


Fig. 6. Patrón de pérdida MCAR.

4.3. Representación gráfica de los datos amputados

La figura 5 muestra una gráfica de todas las curvas amputadas con los diferentes porcentajes de pérdida seleccionados, se observan las partes en las que se generaron datos nulos. Los datos corresponden a la misma curva, del mismo pozo.

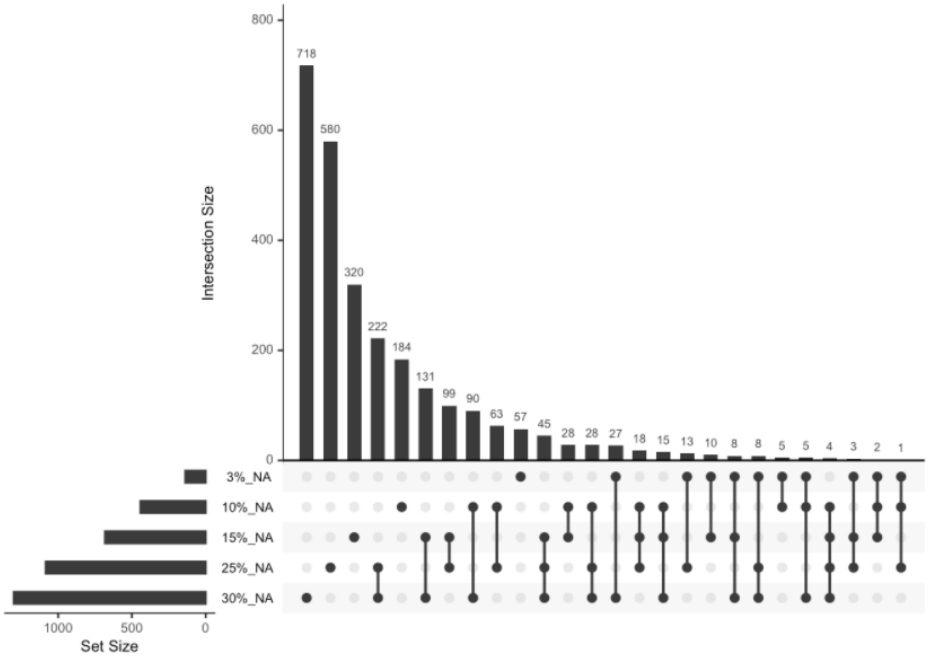


Fig. 7. Acumulación de datos amputados por columna (Porcentaje de pérdida).

La tabla 3 muestra la impresión de los datos con los valores amputados, en comparación con el valor original en cada parte. Nótese que las instancias de GR que contienen NA's coinciden con las instancias de gaps que contienen valores, es decir son inversas entre sí. De este modo se identifican los intervalos con datos faltantes y se señalan dentro del conjunto de datos.

La figura 6 muestra los patrones de pérdida generados por la función **ampute()**. Es posible apreciar los porcentajes de pérdida por columna (variables), esto también permite comprobar que el mecanismo de pérdida no presenta ningún patrón reconocible y que el mecanismo de pérdida aplicado por la función **ampute()** es **MCAR**.

Por otra parte, la figura 7 muestra las veces que el conjunto de datos presenta NA's, se observa que las columnas 15 % y 30 % coinciden 131 veces, mientras que 25 % y 30 % coinciden 222 veces. Conforme el tamaño de intersección crece, la incidencia por columna se va aislando.

5. Resultados

Una vez realizada la amputación, en la tabla 4 se puede observar una muestra de las instancias 30-60; a simple vista se puede percibir la ausencia de un patrón sistemático, lo cual indica que las instancias fueron amputadas (celdas sombreadas) aleatoriamente. En la figura 8 se presentan gráficamente los datos amputados, con la distribución logística tipo Right, mostrados en la tabla 4 (Right). Se aprecia que los gráficos de las curvas coinciden en las zonas con presencia de datos nulos.

Tabla 4. Muestra de los datos amputados con mecanismo MCAR por tipos de distribución de probabilidad logística Right, Left, Mid y Tail.

Right				Left				Mid				Tail			
Obs.	**GR1**	**GR2**	**GR3**	**Obs.**	**GR1**	**GR2**	**GR3**	**Obs.**	**GR1**	**GR2**	**GR3**	**Obs.**	**GR1**	**GR2**	**GR3**
30	36.47	46.57	51.15	30	×	46.57	51.15	30	36.47	46.57	51.15	30	36.47	46.57	51.15
31	38.15	49.38	52.41	31	38.15	49.38	52.41	31	38.15	49.38	52.41	31	×	49.38	52.41
32	39.78	49.40	53.92	32	39.78	49.40	53.92	32	39.78	49.40	53.92	32	39.78	49.40	53.92
33	40.86	×	54.33	33	×	46.88	54.33	33	×	46.88	54.33	33	40.86	46.88	54.33
34	40.95	46.88	53.19	34	40.95	46.88	53.19	34	40.95	46.88	53.19	34	×	46.88	53.19
35	39.92	49.66	51.46	35	39.92	49.66	51.46	35	39.92	49.66	51.46	35	39.92	×	51.46
36	38.29	50.16	50.47	36	38.29	50.16	50.47	36	38.29	50.16	50.47	36	38.29	50.16	50.47
37	37.06	×	50.73	37	37.06	50.25	50.73	37	37.06	50.25	50.73	37	37.06	50.25	50.73
38	36.81	×	51.55	38	36.81	50.72	51.55	38	36.81	50.72	51.55	38	36.81	50.72	51.55
39	37.23	48.81	51.96	39	37.23	48.81	51.96	39	×	48.81	51.96	39	37.23	48.81	51.96
40	37.69	×	51.61	40	37.69	46.88	51.61	40	37.69	46.88	51.61	40	37.69	46.88	51.61
41	37.69	43.01	51.03	41	37.69	43.01	51.03	41	37.69	43.01	51.03	41	37.69	43.01	51.03
42	×	46.28	51.00	42	×	46.28	51.00	42	37.15	46.28	51.00	42	×	46.28	51.00
43	36.56	×	51.61	43	36.56	45.19	×	43	36.56	45.19	51.61	43	36.56	45.19	51.61
44	36.42	×	52.15	44	36.42	46.59	52.15	44	36.42	46.59	52.15	44	36.42	46.59	52.15
45	36.80	47.91	51.65	45	36.80	×	51.65	45	36.80	47.91	51.65	45	36.80	47.91	×
46	37.38	51.19	49.51	46	×	51.19	49.51	46	37.38	51.19	49.51	46	37.38	51.19	49.51
47	37.85	54.43	45.72	47	37.85	54.43	45.72	47	37.85	54.43	×	47	37.85	×	45.72
48	38.23	49.75	40.70	48	38.23	×	40.70	48	38.23	49.75	40.70	48	38.23	49.75	40.70
49	38.81	48.41	35.34	49	38.81	48.41	35.34	49	38.81	48.41	35.34	49	38.81	48.41	35.34
50	39.70	47.54	30.96	50	39.70	47.54	30.96	50	×	47.54	30.96	50	39.70	×	30.96
51	40.90	50.75	28.75	51	40.90	50.75	28.75	51	40.90	50.75	28.75	51	×	50.75	28.75
52	42.61	50.69	×	52	42.61	50.69	×	52	42.61	×	29.23	52	42.61	50.69	×
53	44.85	53.19	32.02	53	44.85	53.19	32.02	53	44.85	×	32.02	53	44.85	53.19	32.02
54	×	51.00	36.15	54	47.19	51.00	×	54	×	51.00	36.15	54	47.19	51.00	36.15
55	49.04	53.59	40.61	55	49.04	53.59	40.61	55	49.04	×	40.61	55	49.04	53.59	40.61
56	50.09	52.09	44.94	56	50.09	52.09	44.94	56	50.09	52.09	44.94	56	×	52.09	44.94
57	50.69	52.66	48.96	57	50.69	52.66	×	57	50.69	52.66	48.96	57	×	52.66	48.96
58	51.39	52.09	52.27	58	51.39	52.09	52.27	58	51.39	52.09	52.27	58	51.39	52.09	×
59	52.32	51.07	54.19	59	52.32	51.07	54.19	59	52.32	×	54.19	59	52.32	51.07	54.19
60	53.27	52.35	54.59	60	53.27	52.35	54.59	60	53.27	52.35	54.59	60	×	52.35	54.59

Por ejemplo, la curva **gr1** (negro) presenta 2 gaps, que corresponden a las observaciones 42 y 54. La curva **gr2** (rojo) presenta los gaps correspondientes a las observaciones 33, 37, 38, 40, 43 y 44, en el caso de la observación 39, esta no es graficada puesto que no tiene datos antes y después.

Finalmente, la curva **gr3** (azul) presenta 1 gap correspondiente a la observación 52. La experimentación realizada para llegar a estos resultados es replicable en su totalidad, sin embargo es importante mencionar que los resultados pueden variar un poco, dado que el mecanismo de pérdida es completamente aleatorio. No obstante, la metodología utilizada funciona para cualquier conjunto de datos de registros de pozos.

6. Conclusiones

Se logró establecer un procedimiento completamente replicable, adecuado y de fácil acceso para la comunidad científica y petrofísica, con el cual es posible simular la presencia de datos faltantes. Sin importar cuál sea la finalidad que tenga cada usuario, puede aplicar este procedimiento.

Right			
Obs.	**GR1**	**GR2**	**GR3**
30	36.47	46.57	51.15
31	38.15	49.38	52.41
32	39.78	49.40	53.92
33	40.86	×	54.33
34	40.95	46.88	53.19
35	39.92	49.66	51.46
36	38.29	50.16	50.47
37	37.06	×	50.73
38	36.81	×	51.55
39	37.23	48.81	51.96
40	37.69	×	51.61
41	37.69	43.01	51.03
42	×	46.28	51.00
43	36.56	×	51.61
44	36.42	×	52.15
45	36.80	47.91	51.65
46	37.38	51.19	49.51
47	37.85	54.43	45.72
48	38.23	49.75	40.70
49	38.81	48.41	35.34
50	39.70	47.54	30.96
51	40.90	50.75	28.75
52	42.61	50.69	×
53	44.85	53.19	32.02
54	×	51.00	36.15
55	49.04	53.59	40.61
56	50.09	52.09	44.94
57	50.69	52.66	48.96
58	51.39	52.09	52.27
59	52.32	51.07	54.19
60	53.27	52.35	54.59

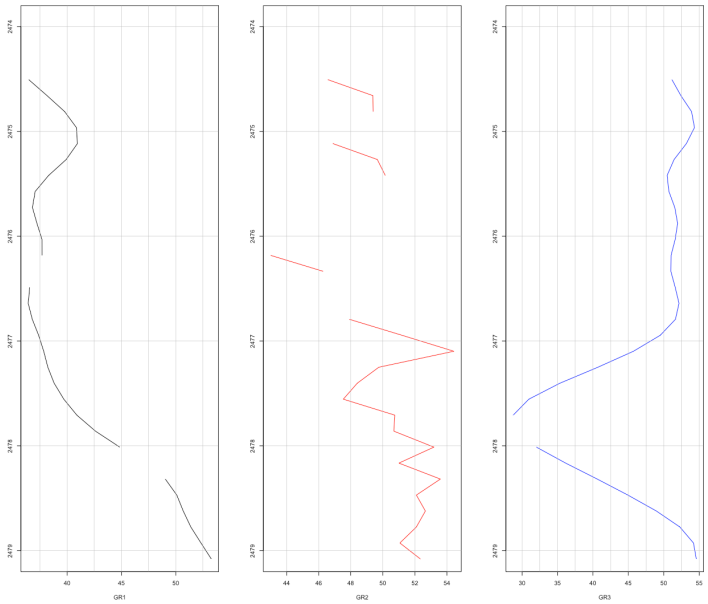


Fig. 8. Comparación de las curvas amputadas con probabilidad logística *Right* y la ocurrencia de datos nulos en el conjunto de datos.

Se pretende extender esta investigación hasta la aplicación de métodos de imputación, tomando los resultados obtenidos como base fundamental para el establecimiento de una metodología que sirva para replicar un procedimiento adecuado de amputación (simulación de datos faltantes) con fines de desarrollo de nuevas técnicas de imputación y el mejoramiento de las actuales.

Los datos utilizados aseguran la reproducibilidad de esta metodología, al ser de libre acceso. La función de amputación (**ampute()**) presenta un límite para el porcentaje de pérdida deseado (no más del 30 %), puesto que una presencia muy alta de faltantes en un dataset, lo convierte en un conjunto de datos de mala calidad.

Por otro lado los tipos de probabilidad logística juegan un papel sumamente importante a la hora de la amputación, ya que definen los rangos dentro de los cuales se encuentran los datos con mayor probabilidad de ser amputados según el tipo de esta distribución de probabilidad.

El estudio y la aplicación de nuevos paradigmas de la computación en conjunto con técnicas clásicas implica el desarrollo de experimentación controlada y referenciada, por lo que contar con un proceso de muestreo y análisis de los datos disponibles es una ventaja y un avance en la prueba de nuevas alternativas para el tratamiento de datos faltantes en registros geofísicos. De este modo, podrá ser más certera la comparación de resultados.

El mecanismo de pérdida MCAR es, en la opinión de los autores, el que más se ajusta a la naturaleza de los datos de registros geofísicos, debido a que los procesos naturales implican la incidencia de muchas variables, las cuales son difíciles (por no decir imposibles) de modelar de manera exacta.

Aunque este trabajo se enfocó al mecanismo de pérdida MCAR, es posible para otros conjuntos de datos, simular los mecanismos, patrones (MNAR, MAR, según la naturaleza de los datos), frecuencia, proporción y distribución logística de pérdida, relacionados a la propia naturaleza de cada conjunto de datos.

Referencias

1. Dong, Y., Joanne-Peng, C. Y.: Principled missing data methods for researchers. SpringerPlus, Springer Science and Business Media LLC, vol. 2, no. 1 (2013) doi: 10.1186/2193-1801-2-222
2. Fox, G.A., Negrete-Yankelevich, S., Sosa, V. J.: Ecological statistics: Contemporary theory and application. Oxford University Press (2015) doi: 10.1093/acprof:oso/9780199672547.001.0001
3. Lopes Rui L., Alípio, J.: Mind the Gap: A well log data analysis (2017)
4. Mallol Roselló, P. J.: Importancia del tratamiento de datos perdidos. Aplicación en estudios longitudinales pequeños, Universitat Oberta de Catalunya (2017)
5. Pyle, D.: Data preparation for data mining (1999)
6. RStudio Team. RStudio: Integrated Development Environment for R. RStudio (2020)
7. Santos, M. S., Pereira, R. C., Costa, A. F., Soares, J .P., Santos, J., Abreu, P. H.: Generating synthetic missing data: A review by missing mechanism. IEEE Access, vol. 7, pp. 11651–11667 (2019) doi: 10.1109/access.2019.2891360
8. Schouten, R. M., Lugtig, P., Vink, G.: Generating missing values for simulation purposes: A multivariate amputation procedure. Journal of Statistical Computation and Simulation, vol. 88, no. 15, pp. 2909–2930 (2018) doi: 10.1080/00949655.2018.1491577
9. Schouten, R. M., Vink, G.: The dance of the mechanisms: How observed information influences the validity of missingness assumptions. Sociological Methods and Research, SAGE Publications, vol. 50, no. 3, pp. 1243–1258 (2018) doi: 10.1177/0049124118799376

ISSN 1870-4069

DABATestors un algoritmo basado en CUDA para reducción del tiempo de obtención de testores

Daniel Alejandro Barajas Aranda, Aurora Torres Soto,
María Dolores Torres Soto

Benemérita Universidad Autónoma de Aguascalientes,
México

alengot@hotmail.com

Resumen. El cálculo de testores es un problema de tipo exponencial, por lo que procesar grandes conjuntos de datos para obtener los testores, es una tarea costosa computacionalmente, puede consumir mucho tiempo, llegando a tardar meses en espacios de soluciones de 2^{40}-1. Este trabajo presenta al algoritmo DABATestor, el cual está basado en el paradigma CUDA como una propuesta para la reducción del tiempo de procesamiento. Dicho algoritmo ha sido comparado con un algoritmo secuencial usando matrices de 2x4, 15x7, 57x14, 8421x354, en donde se han obtenido reducciones del orden de una tercera parte del tiempo de procesamiento requerido para la identificación del 100% de los testores de cada matriz. Este algoritmo se ha programado en Python y se pone a disposición de la comunidad científica.

Palabras clave: Testores, CUDA, paralelización, lógica combinatoria, reconocimiento de patrones.

DABATestors an Algorithm Based on CUDA to Reduce Time to Obtain Testors

Abstract. The calculation of testors is an exponential problem, so processing large data sets to obtain them is a computationally expensive, since it is of exponential order and time-consuming task, taking months in 2^{40}-1 solutions' spaces. In this work, the DABATestor algorithm is presented, it is based on the CUDA paradigm, as a proposal to reduce processing time. This algorithm has been compared with a sequential algorithm using 2x4, 15x7, 57x14, 8421x354 matrices, where results of at least one-third of the processing time have been obtained. This algorithm has been programmed in Python and it is put available to the scientific community.

Keywords: Testors, CUDA, parallelization, combinatorial logic, pattern recognition.

1. Introducción

La obtención exhaustiva de los testores de una matriz de datos es una tarea ardua de gran complejidad computacional, que ha sido abordada desde diferentes perspectivas. En este presente artículo, se presenta el algoritmo DABATestors, que está basado en CUDA (Compute Unified Device Architecture - Arquitectura Unificada de Dispositivos de Cómputo). DABATestors, es un algoritmo exhaustivo que funciona con los procesadores nVidia para la obtención en paralelo del 100% de los testores de un grupo de datos.

La solución exhaustiva de gran parte de los problemas del mundo real, es extraordinariamente costosa computacionalmente, como lo es el cálculo de todos los testores asociados a una matriz de aprendizaje. En este documento, se presenta DABATestors, un algoritmo de paralelización determinista de escala exterior que usa el paradigma de programación CUDA para la obtención del conjunto de todos los testores asociados a una matriz de aprendizaje, este algoritmo fue sometido a prueba con matrices de distintos tamaños.

Para evaluar el desempeño del algoritmo DABATestors, se comparó contra un algoritmo secuencial utilizando un GPU Tesla P100-PCIE-16GB con 4 diferentes matrices ejecutando 30 réplicas con cada una de ellas.

1.1. Reconocimiento de patrones

Es una disciplina científica que busca la clasificación de objetos en un número de categorías o clases [1]. Los objetos pueden ser: imágenes, señales, sonidos o cualquier tipo de dato. Desde los años 60's esta disciplina se ha desarrollado gracias al avance de la tecnología, dejando de ser solo una parte teórica de la estadística [1]. Las principales aplicaciones del reconocimiento de patrones son: visión por computadora, reconocimiento de caracteres, diagnóstico asistido por computadora y reconocimiento de voz, entre otras.

La teoría de testores, la cual es una subárea del reconocimiento de patrones, se formuló como una de las direcciones científicas independientes de la cibernética matemática a mediados de los años 50's en la ex-unión de las repúblicas socialistas soviéticas (URSS) [2]. Un testor es un conjunto de características capaz de distinguir objetos provenientes de diferentes clases disjuntas, de manera, que ningún objeto de una clase puede ser confundido con alguno de otra [3].

Los testores han sido aplicados en varios ámbitos, por ejemplo: en medicina se empleó en la determinación de factores asociados con lesiones pulmonares agudas relacionadas con transfusiones sanguíneas (TRALI); se aplicaron mediante un algoritmo evolutivo híbrido [4]. También se han aplicado en astronomía para la estimación de parámetros estelares [5], así como en ciencias computacionales, donde han sido utilizados para reducir las dimensiones de modelos de redes neuronales [6]. Los testores son una herramienta muy útil, sin embargo, su cálculo es un proceso tardado que crece exponencialmente con cada una de las variables que tenga la matriz a procesar [7].



Fig. 1. CPU vs GPU [10].

Para el lector interesado en profundizar sobre testores y los conceptos relacionados con esto, se recomienda consultar a Schucloper (Introducción a la Teoría de Testores J Ruíz-Shulcloper, E Alba, M Lazo - Serie Verde Cinvestav-IPN, 1994).

1.2. Cómputo paralelo

La evolución que han tenido las tarjetas gráficas ha hecho factible el empleo de dos o más procesadores para la solución computacional de un problema, mediante su división en componentes independientes que pueden resolverse de manera paralela.

Lo anterior, bajo el principio de que los grandes problemas se pueden dividir en otros más pequeños [8].

Entre las diferentes razones que existen para el uso del cómputo en paralelo, se pueden destacar tres [9]:

- La transformación y creación de algoritmos de modo paralelo para resolución de problemas.
- La capacidad del cómputo paralelo de disminuir el tiempo de cómputo necesario en problemas de grandes volúmenes de datos o cálculos intensivos.
- El límite físico alcanzado por las computadoras secuenciales.

1.3. CUDA

CUDA son las siglas de Arquitectura Unificada de Dispositivos de Cómputo (Compute Unified Device Architecture), que es una plataforma de computación en paralelo y un conjunto de herramientas de desarrollo, creadas por nVidia que permiten la codificación en GPU (graphics processing unit). En la Figura 1 se muestra que un GPU contiene más núcleos ALU (arithmetic logic unit- unidad aritmética lógica) que un CPU y que éstos comparten la memoria, lo que funciona conceptualmente para cálculos altamente paralelos porque la GPU puede ocultar las latencias de acceso a la memoria en lugar de acceder a ella a través de grandes cantidades de memoria caché.

Dentro de los GPU nVidia, existe una jerarquía de subprocesos, cada bloque de subprocesos puede programarse en cualquiera de los multiprocesadores disponibles dentro de una GPU, en cualquier orden, de forma simultánea o secuencial, de modo que un programa CUDA compilado pueda ejecutarse en cualquier número de multiprocesadores como se ilustra en la Figura 2 [10]. En esta arquitectura el programador solo tiene que definir los bloques de procesos, y automáticamente el GPU se encarga de asignarlos a los núcleos que tenga disponibles.

Tabla 1. Matrices de Prueba.

Matriz	Registros	Variables	Casos estudiados del Conjunto Potencia
M1	57	14	16384
M2	8421	354	100000, 10000000
M3	2	4	16
M4	15	7	128

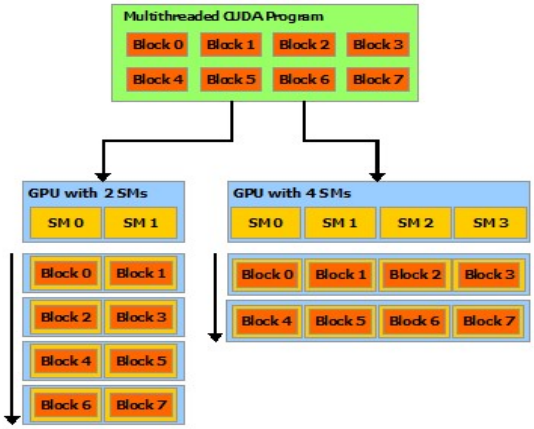


Fig. 2. Escalabilidad Automática [10].

Un programa multiproceso, se divide en bloques de subprocesos que se ejecutan independientemente uno del otro, de modo que una GPU con más procesadores ejecutará automáticamente el programa en menos tiempo que una GPU con menos procesadores [10].

2. Materiales y métodos

El algoritmo DABATestors fue implementado bajo el paradigma CUDA en el lenguaje Python. Este requiere de la librería numba y el uso de CUDA Toolkit versión 8 o superior para su correcto funcionamiento. Adicionalmente, se utilizaron las librerías numpy y pandas para el uso de matrices.

El algoritmo recibe como insumos una matriz básica que contiene la información sintetizada sin los superconjuntos de la matriz de diferencias, y un conjunto potencia el cual está formado por todas las combinaciones binarias del número de variables que componen el problema: 2^n-1 posibilidades, donde **n** es el número de variables que contiene el problema.

Cabe resaltar que la matriz básica se debe tener previamente en memoria, mientras que la matriz del conjunto potencia se define en tiempo de ejecución. La implementación del algoritmo consiste en establecer el proceso general para la obtención de un solo testor, y este proceso se paraleliza en varios hilos pertenecientes a varios bloques en la

```
Definir función (IsTestor)
        Identificar hilo de GPU
        Obtener número de testor guardar en decimal
        Mientras decimal // !=0
                Arreglo1 en [-j]= decimal % 2
                Incrementar j
        Arreglo1 [-j] = decimal
        Band=1
        Bucle hasta total de filas de matriz básica (j)
                Band2=0
                Bucle hasta total de columnas de matriz básica (k)
                Si arreglo1 [k] ==1
                        Si matriz básica [j,k]==1
                                Band2++
                Si band2==0
                        Band1=0
        Si band1 ==1
                Retornar true
        Si no
                Retornar false
Definir dimensión de malla y bloques
Definir tamaño de paso
Llamar función con arreglo del tamaño de paso empezando desde 0
Si último número llamado < 2^(número de columnas matriz básica)
        Tamaño de paso *2
        Llamar función con arreglo del tamaño de paso empezando en último número llamado
Repetir desde el si
```

Fig. 3. Pseudocódigo del Algoritmo DABATestors.

cuadrícula del GPU. El algoritmo se puso a prueba con diversas matrices (Tabla 1) obteniendo los testores para problemas conocidos con anterioridad (M1, M2), donde se trabaja datos de personas con tendencia suicida [11]; explorando el espacio total de soluciones en la matriz M1; y en la matriz M2 se verificaron 100,000 y 10,000,000 de casos del conjunto potencia. También se usaron matrices M3 y M4 en donde se exploró el espacio completo de soluciones.

La evaluación del algoritmo se llevó a cabo en un GPU Tesla P100-PCIE-16GB, en donde se ejecutó 30 veces para cada matriz. El algoritmo secuencial también fue replicado 30 veces. El pseudocódigo de la herramienta desarrollada (bajo el paradigma CUDA), se muestra a continuación (Fig. 3).

Se puede apreciar que el pseudocódigo define el tamaño de bloques y mallas a trabajar por el GPU, y define el tamaño de paso (tamaño del subconjunto del conjunto potencia) a utilizar en la iteración, enseguida llama a la función IsTestor enviando subconjuntos del conjunto potencia. En la función identifica el hilo utilizado por el GPU para tomar el registro del conjunto potencia y verificar si es o no testor, transformando el número a binario y posteriormente comparándolo con la matriz básica.

3. Resultados

Se obtuvo el tiempo de procesamiento de las diversas matrices mediante el algoritmo DABATestors y un algoritmo secuencial que utiliza la misma función descrita en el pseudocódigo, pero de forma secuencial. Mediante la comparación de los tiempos de

```
@cuda.jit
def testor(mb,mto,bina,col,fil,lim): #matriz basica, matriz de testores
                  #y salida, vector de binario, columnas, filas, limite
  i = cuda.threadIdx.x + cuda.blockIdx.x * cuda.blockDim.x

 #obtención del número a verificar testor
  decimal=mto[i,0]

#crear binario
  j=1
  while decimal // 2 != 0:
    bina[i,-j]  = (decimal % 2)
    decimal = decimal // 2
    j+=1
  bina[i,-j]=decimal

#verificar is testor con la matriz básica
  istestor=1
  for j in range(fil):
    cad0=0
    for k in range (col):
      if bina[i,k]==1:
        if mb[j,k]==1:
          cad0=cad0+1
    if cad0==0:
      istestor=0

    if istestor!=0:
      mto[i,1]=1
    else:
      mto[i,1]=0
#fin de la función paralela

#defición del bloque de hilos
griddim = 1000, 1
blockdim = 1000, 1
i=0
paso=1000000

#creación de los arreglos
mto=numpy.arange((paso*i), (paso*(i+1)), 1, dtype=numpy.int32)
mto=mto.reshape((paso, 1))
mto=numpy.insert(mto, 1, 0, axis=1)
bina = numpy.zeros((paso, col), dtype=numpy.int32)
mto_out = cuda.device_array_like(mto)

mto_out[:]=mto[:]
lim=2**col
lim=0
mto_out.copy_to_host(mto)

#ciclo de ejecución para obtención de testores
for j in range(200):
  mto=numpy.arange((paso*i), (paso*(i+1)), 1, dtype=numpy.int32)
  mto=mto.reshape((paso, 1))
  mto=numpy.insert(mto, 1, 0, axis=1)
#llamada a la función paralela
  testor[griddim, blockdim](mb,mto_out,bina,col,fil,lim)

  mte=pd.DataFrame(mto)
  mte=mte[mte[1] == 1]
```

Fig. 4. Algoritmo DABATestors.

procesamiento, entre el algoritmo DABATestors y el secuencial, se estableció el punto donde el uso del algoritmo DABATestors es similar que uno secuencial (conjunto potencia de (2^7)-1). Uno de los resultados más importantes de este trabajo es el código de la herramienta DABATestors (Fig. 4). Así como la comprobación de su efectividad.

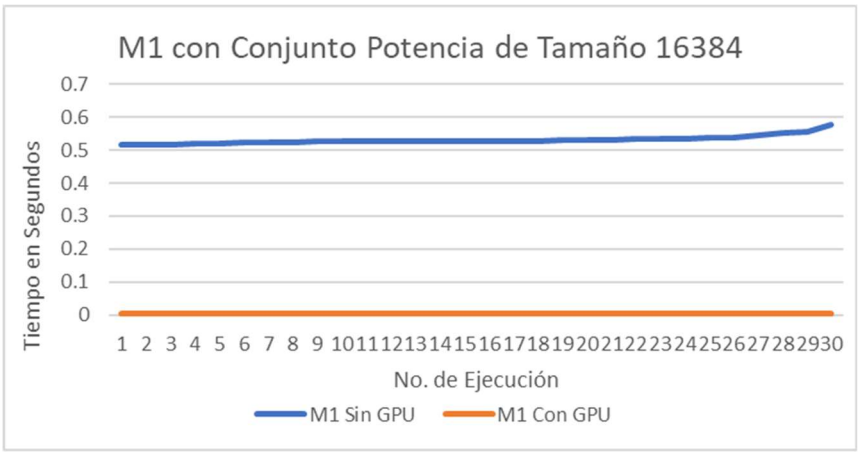


Fig. 7. Tiempo de procesamiento de la matriz M1 con 16384 casos del conjunto potencia.

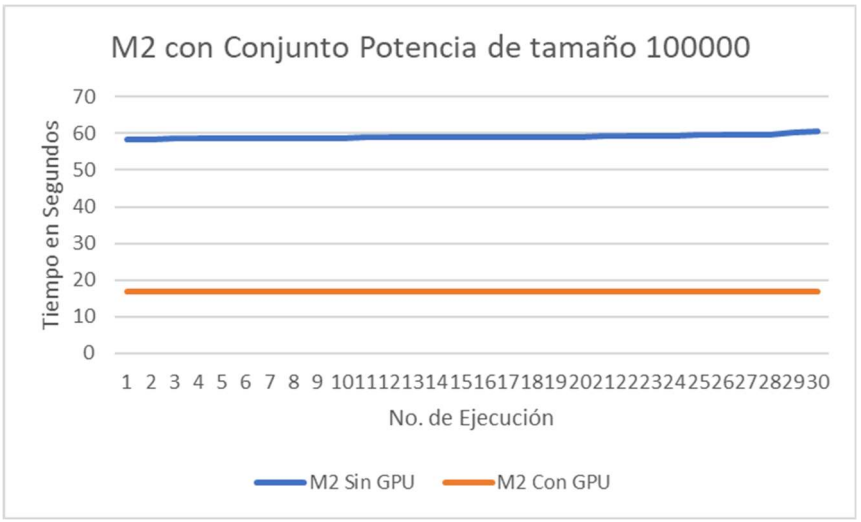


Fig. 8. Tiempo de procesamiento de la matriz M2 con 100,000 casos del conjunto potencia.

En la figura 4 se puede observar que el tamaño de paso es de 1,000,000, esto se define de esta manera ya que el procesador utilizado no soporta un número mayor de datos en paralelo, pudiendo aumentar o disminuir este número según el procesador disponible. Se realizaron pruebas de desempeño en las que se midió el tiempo de ejecución durante 30 réplicas.

La función testor descrita en el algoritmo se ejecutó de manera secuencial y paralela mediante CUDA en un GPU Tesla P100-PCIE-16GB, obteniendo mediciones para las matrices M1, M2, M3 y M4 mencionadas previamente.

En la figura 5 se observa el tiempo promedio de las pruebas efectuadas en los diferentes casos. En el procesamiento de casos del conjunto potencia (CP) de tamaño 16, 6384 y 100,000 filas respectivamente el tiempo de procesamiento con CUDA (línea naranja)

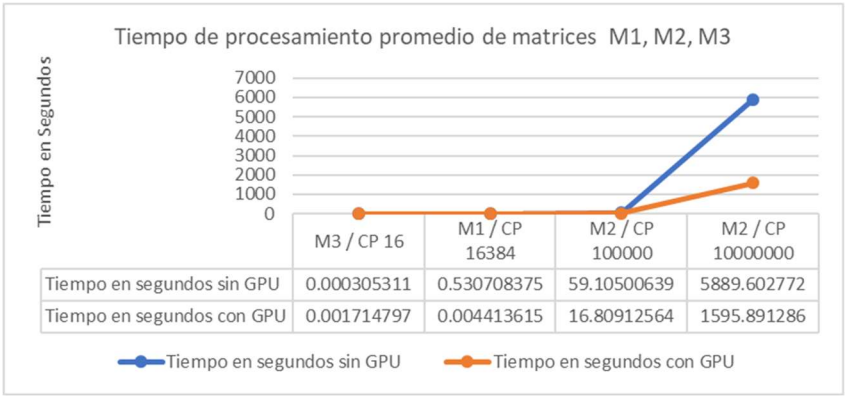


Fig. 5. Tiempo de procesamiento promedio de matrices M3, M1, M2.

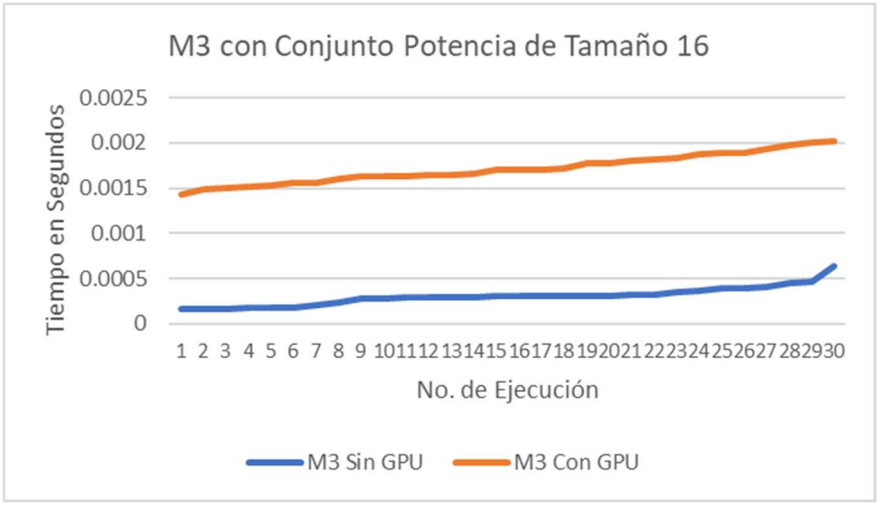


Fig. 6. Tiempo de procesamiento de la matriz M3 con 16 casos del conjunto potencia.

aparenta ser similar a el procesamiento secuencial, sin embargo, cuando el espacio de soluciones es grande (10,000,000), se puede apreciar un desempeño muy bueno en el procesamiento con CUDA.

En la figura 6 se observa que el uso del algoritmo DABATestors no es una buena opción cuando se tiene un grupo de datos pequeño, ya que el tiempo de procesamiento fue mayor al requerido en el procesamiento secuencial (línea azul).

Esto ocurrió debido a que los tiempos requeridos para cargar los datos en memoria son mayores que el tiempo que tarda en procesarlos.

En las figuras 7 y 8, se observa la diferencia entre el tiempo de procesamiento, entre el algoritmo DABATestors y el secuencial. Se puede apreciar que los tiempos requeridos con el uso de GPU (línea naranja) son significativamente inferiores a los requeridos sin el uso de este (línea azul), mostrando resultados de al menos un tercio del tiempo requerido por un algoritmo secuencial. En el caso de la figura 9 se realizó el

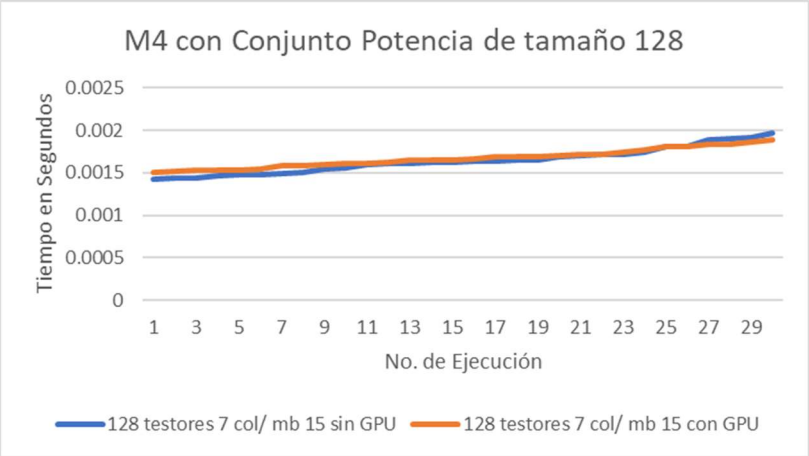


Fig. 9. Tiempo de procesamiento de la Matriz M4 con 128 casos del conjunto.

análisis de la matriz M4 de sus 128 testores posibles. Se observa que el tiempo de procesamiento con el uso del GPU (línea naranja) y sin él (línea azul) son muy semejantes. La grafica muestra los resultados de 30 ejecuciones diferentes, donde se obtuvieron tiempos muy similares, mostrando que es conveniente dejar de usar el procesamiento secuencial y empezar a usar el procesamiento en paralelo en matrices de características similares a la matriz M4 cuyo conjunto potencia es de (2^7)-1 filas, sin importar en gran medida el número de registros de la matriz básica.

Del grupo de experimentos expuestos se observa que cuando el conjunto potencia crece (a partir de (2^7)-1) filas, el uso del algoritmo DABATestors es una opción muy buena, ya que se reduce el tiempo de procesamiento. Con el algoritmo DABATestors se mejoró significativamente el tiempo de procesamiento logrando reducirlo a un tercio del requerido por su contraparte secuencial en problemas con conjuntos potencia de al menos (2^{14})-1.

4. Conclusiones y trabajo a futuro

Los problemas del mundo real raramente son abordados con técnicas exhaustivas debido a su alto costo computacional; un buen ejemplo es la identificación del 100% de los testores asociados a una matriz. En este trabajo, se presenta una alternativa que logra reducir el tiempo de procesamiento requerido por un algoritmo exhaustivo y secuencial en al menos una tercera parte para problemas con conjuntos potencia de (2^{14})-1 filas por lo bajo.

Se ha demostrado que el uso del algoritmo DABATestors provee resultados muy convenientes en cuanto al tiempo de ejecución, no obstante, con problemas con pocas variables, es mejor usar una técnica secuencial; debido al tiempo requerido en la copia de datos a la memoria del GPU, sin embargo, cuando el tiempo de procesamiento es superior al tiempo de transferencia de datos a la memoria, el uso del algoritmo DABATestors representa una mejor alternativa que un algoritmo secuencial.

Cuando los problemas tienen una complejidad algorítmica muy elevada y se requiere que la solución sea determinista, el uso del GPU es una buena alternativa, ya que permite reducir el tiempo de procesamiento sustancialmente. En el caso de la obtención de los testores, el algoritmo propuesto brinda una reducción del tiempo de procesamiento, requiriendo un tercio del tiempo del algoritmo secuencial.

En este trabajo se presenta un algoritmo que está a disposición de la comunidad científica. Los resultados presentados se obtuvieron con un GPU Tesla P100-PCIE-16GB con diversas matrices obteniendo mejoras de tiempo significativas en matrices básicas de al menos 14 columnas.

Actualmente se está trabajando en la obtención del total de testores típicos como parte de una nueva propuesta de este algoritmo.

Referencias

1. Prandi, D., Gauthier, J. P.: A semidiscrete version of tehe Citti-Petitot-Sarti model as a plausible model for anthropomorphic image reconstruction and pattern recognition, Springer Briefs in Mathematics (2018)
2. Dmitriev, A. N., Zhuravlev, Y. I., Krendeleiev, F. P.: On mathematical principles of object and phenomena classification, Discrete Analysis, pp.3–15 (1966)
3. Shulcloper, J. R., Guzmán, A., Martínez, J. F.: Enfoque lógico combinatorio al reconocimiento de patrones. Instituto Politécnico Nacional, Mexico (1999)
4. Torres, M. D., Torres, A., Cuellar, F., Torres, M. L., Ponce de León, E., Pinales, F.: Evolutionary computation in the identification of risk factors. Case of TRALI (2014)
5. Santos, J. Á., Carrasco, A., Martínez, J. F.: Selección de características usando testores típicos aplicada a la estimación de parámetros estelares. Computación y Sistemas vol. 8, pp. 15–23 (2004)
6. Vázquez, R A., Godoy-Calderón, S.: Using testor theory to reduce the dimension of neural network models. Spec Issue Neural Networks Assoc Memories vol. 28, pp. 93–103 (2007)
7. Alganza, Y. S., Porrata, A. P.: Lex: Un nuevo algoritmo para el calculo de los testores tipicos. Ciencias Matemáticas, vol. 21, no. 1 (2003)
8. Almasi, G. S.: Highly parallel computing. Benjamin/Cummings series in computer science and engineering (1989)
9. De Giusti, A., Naiouf, M., Sanz, C., Tinetti, F., De Giusti, L, Bertone, R.: Procesamiento paralelo y distribuido en tratamiento masivo de datos. Un workshop de investigadores en ciencias de la computación, pp. 364-367 (2002)
10. Programming Guide : CUDA toolkit documentation. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#introduction. Accessed 26 Mar 2020
11. Barajas, D.: Identificación de factores de riesgo determinantes en el suicidio en Aguascalientes mediante la técnica de testores típicos. Universidad Autónoma de Aguascalites (2017)

ISSN 1870-4069

Revisión del estado del arte en técnicas de minería de flujos de datos

María Yesenia Zavaleta-Sánchez,
Edgard Iván Benítez-Guerrero

Universidad Veracruzana,
Facultad de Estadística e Informática,
México

{yzavaleta, edbenitez}@uv.mx

Resumen. Actualmente, con el avance de la tecnología y el uso de diferentes dispositivos electrónicos, en muchas aplicaciones se generan grandes cantidades de datos de manera continua a altas velocidades llamados flujos de datos, (data streams, en inglés), que requieren ser procesados en cortos lapsos de tiempo. Las técnicas de minería de flujos de datos (Data Stream Mining, DSM por sus siglas en inglés) permiten extraer conocimiento e identificar patrones ocultos en flujos de datos continuos. El objetivo principal de este trabajo es presentar una revisión del estado del arte en el área de minería de flujos de datos para identificar modelos, técnicas y herramientas para el procesamiento de flujos de datos. Se realizó un mapeo sistemático con 69 artículos recuperados de cuatro bases de datos electrónicas. Como resultado se identificó el uso de modelos de series de tiempo y modelo de ventanas deslizantes para representación y procesamiento, respectivamente. Clasificación, conteo de frecuencias y agrupación como las técnicas de DSM más utilizadas. Herramientas para su manejo como Apache Hadoop, Apache Spark y Esper. Finalmente, para trabajos futuros la detección de eventos usando DSM.

Palabras clave: Flujo de datos, minería de flujos de datos, mapeo sistemático de la literatura.

State of the Art Review of Data Stream Mining Techniques

Abstract. Nowadays, with the advancement of technology and the use of different electronic devices, in many applications large amounts of data are generated continuously at high speeds called data streams, which need to be processed in short time lapses. Data Stream Mining techniques (DSM) allow you to extract knowledge and identify hidden patterns in continuous data streams. The main objective of this work is to present a review of the state of the art in the DSM area to identify models, techniques and tools for data stream processing. A systematic literature review was carried out with 69 articles retrieved from four electronic databases. As a result, the use of time

series models and sliding window models for representation and processing, respectively, was identified. Classification, frequency counting and clustering as the most used DSM techniques. Tools for its management such as Apache Hadoop, Apache Spark and Esper. Finally, for future work the event detection using DSM techniques.

Keywords: Data stream, data stream mining, a systematic literature review.

1. Introducción

Actualmente, con el avance de la tecnología y el uso de diferentes dispositivos electrónicos, en muchas aplicaciones se generan grandes cantidades de datos de manera continua a altas velocidades llamados flujos de datos, (data streams, en inglés), que requieren ser procesados en cortos lapsos de tiempo. Los flujos de datos son secuencias infitas de datos que se generan continuamente en el tiempo. Ejemplos de estos son los que se generan en transacciones bancarias, redes sociales, sensores, correo electrónico, entre otros [2, 36].

Asimismo, [2, 9] definen un flujo como una secuencia ilimitada de tuplas de la forma (s, t) ordenada por t, donde s, es una tupla relacional y t es la marca de tiempo de la tupla. Más formalmente, un flujo de datos S es una secuencia masiva de datos $x^1, x^2, ..., x^N$, es decir, $S = \{x^i\}_{i=1}^{N}$, potencialmente ilimitada $(N \rightarrow \infty)$. Cada flujo de datos se describe mediante un vector de atributo n-dimensional $x^i = [x_j^i]_{j=1}^{n}$ que pertenece a un espacio Ω de atributos que puede ser continuo, categórico o mixto [81].

En los últimos años, se ha desarrollado un creciente interés en el estudio de los flujos. Algunos casos de aplicación incluyen datos generados por redes de sensores, datos meteorológicos, análisis del mercado de valores y monitoreo del tráfico de la red, entre otros. Estas aplicaciones involucran conjuntos de datos que son demasiado grandes por lo que generalmente se almacenan en dispositivos secundarios.

Extraer conocimiento útil de los flujos de datos es un desafío, DSM debe considerar las siguientes restricciones: Los flujos llegan contínuamente, es decir, no hay control sobre el orden en que se deben procesar, el tamaño de una secuencia puede ser ilimitado, después de haber sido procesados los flujos son descartados y posiblemente el proceso de generación de flujos no es estacionario, es decir, su distribución de probabilidad puede cambiar con el tiempo [35].

En general, cuando se pretende extraer información o reconocer patrones de comportamiento en grandes cantidades de datos se recurre al uso de técnicas de minería de datos. Por las características de los flujos de datos, las técnicas de minería de datos tradicionales no pueden aplicarse directamente a las secuencias de datos. Esto se debe a que la mayoría requiere múltiples escaneos de datos para extraer información, lo que no es realista para los flujos de datos que pueden cambiar con el tiempo.

Por lo que surge la necesidad de adaptar dichas técnicas para dar un mejor tratamiento a estas estructuras de datos. De ahí se deriva la motivación de esta investigación y la importancia del uso de técnicas de DSM cuyo objetivo es extraer conocimiento y patrones ocultos de flujos de datos continuos [74]. En la literatura, existen otros trabajos no recientes donde se ha abordado el tema de DSM.

Tabla 1. Resultados de búsqueda avanzada.

	ACM	**IEEE**	**SD**	**SL**	**Total**
Búsqueda inicial	111	252	453	244	1066
Primera selección	6	59	78	42	185
Segunda selección	5	22	55	6	88
Selección final	5	16	43	5	69

Por ejemplo, en el 2003 en el trabajo de [41] se hace énfasis en los sistemas para la gestión de flujos, modelos de datos y lenguajes para las consultas continuas.

En 2005, [34] presentaron los fundamentos teóricos para el análisis de flujos de datos. Por otra parte, en trabajos más recientes [68] abordan los métodos para identificar patrones frecuentes en flujos de datos, [74] presentan un survey sobre agrupación y clasificación, [38] hacen una revisión del estado del arte para agrupación y [59] elaboran una revisión sistemática de la literatura para comparar herramientas y tecnologías para el análisis de flujos de datos masivos.

Debido a lo anterior, en este trabajo se presentan los resultados de un mapeo sistemático de la literatura basado en el análisis de 69 artículos publicados en bases de datos electrónicas en el periodo del 2010 al 2019, con la finalidad de conocer aspectos relevantes del área como modelos de datos y de procesamiento de flujos de datos, técnicas de preprocesamiento de flujos de datos, técnicas de DSM, herramientas para su gestión y los temas de investigación pendientes por abordar.

El resto del artículo está organizado de la siguiente forma, en la sección 2 se describe el método de investigación usado para la costrucción del mapeo sistemático, en la sección 3 se presentan los resultados obtenidos para cada pregunta de investigación planteada; y finalmente, en la sección 4 se resumen las conclusiones del estudio.

2. Método de investigación

Para la construcción del mapeo sistemático se tomaron en consideración los lineamientos propuestos por [56]. Las categorías utilizadas en un estudio de mapeo se basan en información de las publicaciones (nombres de autores, afiliaciones de autores, fuente de publicación, tipo de publicación, fecha de publicación, etc.) y/o información sobre los métodos de investigación utilizados [57]. Los pasos esenciales del proceso de nuestro estudio fueron la definición de preguntas de investigación, la búsqueda sistematizada de trabajos relevantes en bases de datos electrónicas, la selección de trabajos, el análisis del contenido de los artículos y los resultados del mapeo.

2.1. Preguntas de investigación

- P1. ¿Cuáles son los modelos que comúnmente se utilizan con flujos de datos?
- P2. ¿Cuáles son las técnicas de preprocesamiento de flujos de datos?
- P3. ¿Cuáles son las técnicas de la minería de flujos de datos más utilizadas?
- P4. ¿Cuáles son las herramientas que se emplean con mayor frecuencia?
- P5. ¿Cuáles son las principales temáticas pendientes por abordar en el área?

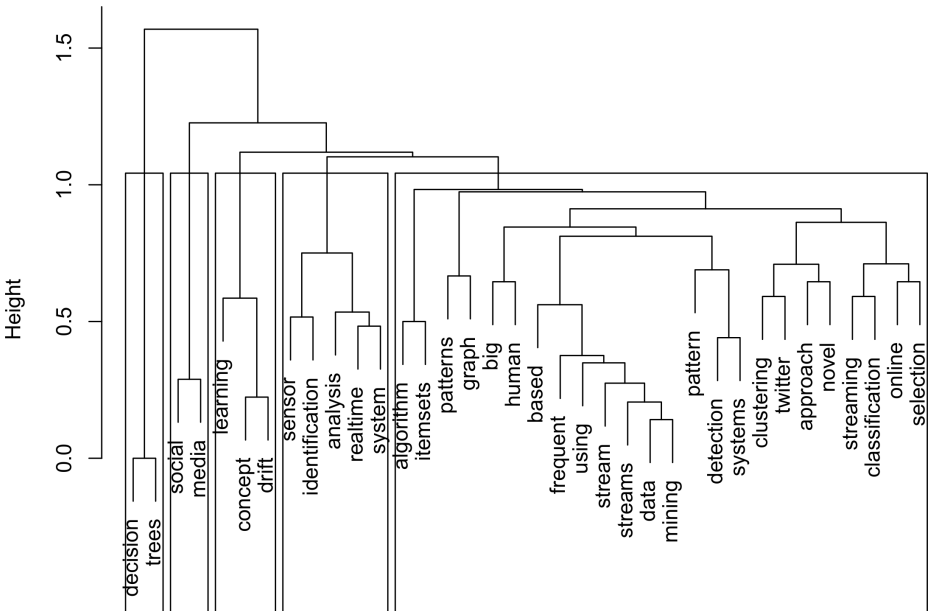


Fig. 1. Dendograma de temáticas abordadas en títulos de artículos.

2.2. Proceso de búsqueda

La estrategia de búsqueda incluyó el uso de cuatro bases de datos científicas de acceso Universitario: ACM Digital Library[1](ACM), IEEE Xplore[2](IEEE), ScienceDirect[3](SD) y SpringerLink[4](SL). En cada base se utilizó la cadena de búsqueda "DATA STREAM MINING". **Criterios de inclusión**: Journals en inglés publicados del 2010 al 2019. Para las bases de datos que lo permitieron, se realizó un refinamiento adicional limitando la búsqueda (ACM-PDF en el área de Computing Sciences in Colleges de enero del 2010 a agosto del 2019; SD–búsqueda en título, resumen y palabras clave).

Como resultado de la búsqueda inicial se encontraron 1066 artículos (111 ACM, 252 IEEE, 453 SD y 244 en SL). En la primera selección se revisó el título, resumen y palabras clave de los documentos, obteniendo como resultado 185 artículos y excluyendo un total de 881 artículos. En la segunda selección, se revisó la introducción y conclusiones de los 185 trabajos, eligiendo 88 artículos.

Finalmente, se realizó una revisión completa de los artículos para identificar si respondian al menos una de las preguntas de investigación con lo que se descartaron 19 artículos y se consideraron para el estudio 69 artículos (5 ACM, 16 IEEE, 43 SD y 5 en SL), ver Tabla 1.

[1] https://dl.acm.org/

[2] https://ieeexplore.ieee.org/Xplore/home.jsp

[3] https://www.sciencedirect.com/

[4] https://link.springer.com/

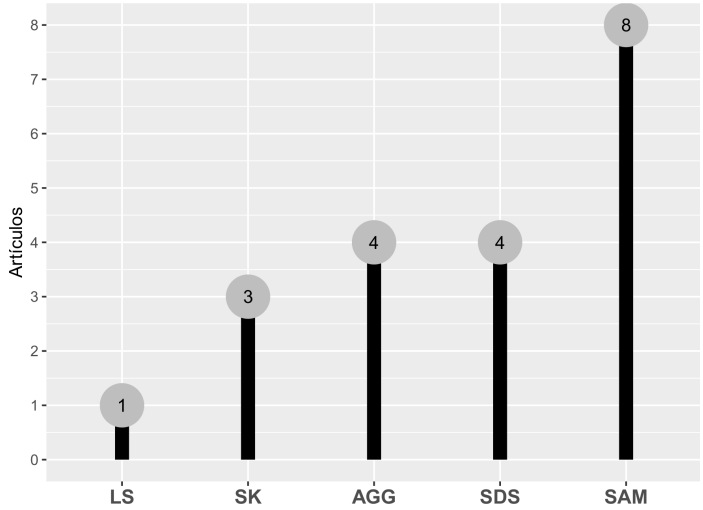


Fig. 2. Lollipop sobre técnicas de preprocesamiento de flujos.

Criterios de exclusión: Se excluyeron 997 artículos que, después de una revisión detallada, no respondían al menos una de las preguntas de investigación planteadas. De cada artículo se extrajo: Título, autores, año de publicación, palabras clave, modelos de datos y de procesamiento, técnicas de preprocesamiento, técnicas DSM, herramientas utilizadas para la minería de flujos y trabajos futuros.

Se construyó una base de datos con las variables de los 69 artículos. Para las variables cualitativas (modelos, técnicas de preprocesamiento y técnicas DSM) se realizaron tablas de frecuencias o gráficos de barras (lollipop) para representar la distribución de las categorías y clasificar los trabajos revisados.

Finalmente, para las variables textuales (título, herramientas y trabajos futuros) se aplicaron técnicas de minería de textos como dendogramas para agrupación jerárquica de palabras y un Trigrama para crear una red semántica y buscar asociaciones entre grupos de 3 palabras sobre trabajos futuros. Cabe resaltar que todos los análisis se realizaron con el Software R Project Versión 4.0.3[5]. En la Figura 1 se muestra un Dendograma donde se agrupan las temáticas principales que se abordan en los títulos de los artículos.

3. Resultados

P1. ¿Cuáles son los modelos que comúnmente se utilizan con flujos de datos? De acuerdo con los trabajos de [39, 72, 42], existen al menos cuatro formas en que un flujo de datos puede representar una señal subyacente $A : [1...N] \rightarrow R$: Modelo de serie temporal, modelo de caja registradora, modelo de torniquete y modelo agregado. En el modelo de serie temporal (time series model, en inglés) cada observación proporciona directamente el valor de la señal subyacente y aparece en orden creciente.

[5] https://www.r-project.org/

Tabla 2. Resumen de técnicas de minería de flujos de datos.

Técnica	Artículos	Autores
Classification	34	[64], [91], [18], [12], [6], [8], [30], [73], [71], [93], [70], [62], [17], [23], [52], [85], [32], [14], [31], [49], [5], [1], [60], [92], [65], [7], [19], [78], [21],[79], [90], [46], [63], [76]
Frequency counting	15	[94], [75], [50], [61], [84], [28], [20], [24], [83], [16], [77], [29], [44], [27], [22]
Clustering	13	[86], [95], [43], [88], [87], [37], [26], [13], [15], [69], [82], [55], [33]
Time series analysis	6	[89], [86], [66], [45], [47], [54]

En el modelo de caja registradora (cash register model, en inglés) cada flujo representa un incremento positivo al valor anterior de la señal para obtener el nuevo valor. El modelo de torniquete (turnstile model, en inglés) es una generalización del modelo de caja registradora al permitir que los valores que se incrementan del rango parcial sean negativos.

Finalmente, en el modelo agregado (aggregate model, en inglés) cada elemento del flujo contiene un rago de valores para un valor particular en el dominio de A.

Según la investigación de [96, 51], hay tres modelos de procesamiento de flujos de datos: Landmark, Damped y Sliding Windows. El modelo landmark extrae todos los conjuntos de elementos frecuentes en todo el historial de flujos de datos desde un punto de tiempo específico llamado landmark hasta el presente.

El modelo damped o time-fading model, extrae conjuntos de elementos frecuentes en los flujos de datos en los que cada transacción tiene un peso y este peso disminuye con la edad. El modelo de ventanas deslizantes (sliding windows, en inglés) encuentra y mantiene conjuntos de elementos frecuentes en ventanas deslizantes cuyo tamaño está predefinido y se puede decidir de acuerdo con las aplicaciones y los recursos del sistema.

Elegir qué tipo de modelo se debe usar depende en gran medida de las necesidades de la aplicación. En los trabajos revisados, el modelo de flujo de datos más utilizado fue el modelo de series de tiempo [89, 86, 66, 45, 47, 54]. Por su parte, los modelos de procesamiento de datos identificados fueron dos: el modelo de ventana deslizante [65, 29, 22]; y el modelo de ventana landmark [28].

P2. ¿Cuáles son las técnicas de preprocesamiento de flujos de datos? En el trabajo de [34], se describe a las técnicas de preprocesamiento como enfoques generales para procesar flujos de datos antes de aplicar DSM. El muestreo (del inglés, sampling, SAM) es el proceso de elección probabilística de un elemento o dato para ser procesado o no [25]. Load shedding (LS), se refiere al proceso de descartar una secuencia de flujos de datos en consultas contínuas.

Sin embargo, es difícil de usar con algoritmos de minería de flujos porque elimina fragmentos de flujos de datos que podrían usarse en la generación de modelos [11]. Sketching (SK) permite proyectar aleatoriamente un subconjunto de las características [10]. Synopsis data structures (SDS), consiste en aplicar técnicas de resumen a la secuencia entrante para su posterior análisis. El análisis de wavelet, histogramas, cuantiles y momentos de frecuencia son algunos ejemplos [10, 40].

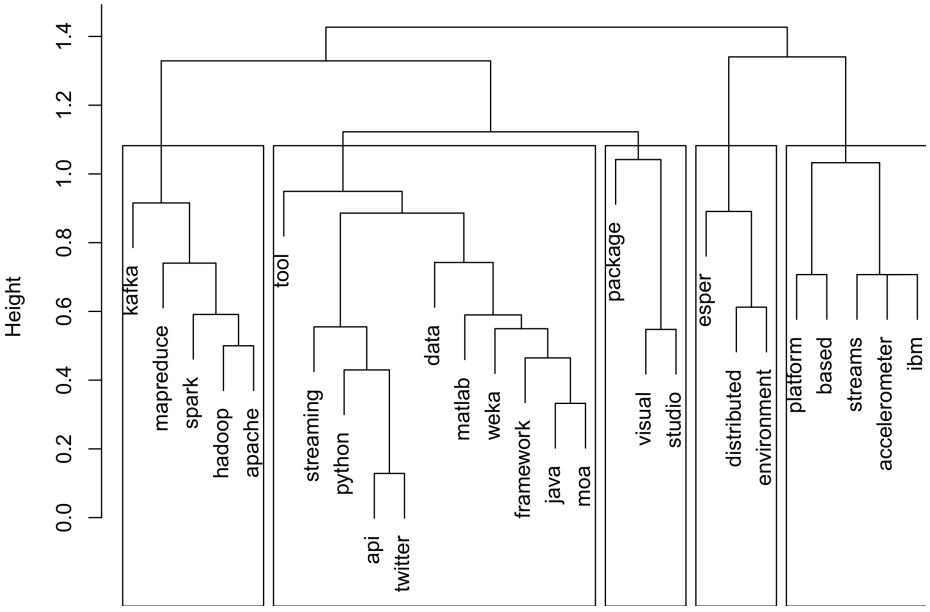


Fig. 3. Dendograma sobre herramientas para minería de flujos de datos.

Por último, la agregación (en inglés, aggregation, AGG), se obtiene al calcular medidas estadísticas como las media y la varianza que resumen el flujo entrante [4, 3]. En nuestro estudio se identificó SAM en 8 artículos: [91, 95, 80, 23, 85, 32, 83, 63] seguida de SDS [64, 94, 88, 78] y AGG [87, 92, 19, 46] en 4 artículos cada una; SK [58, 73, 93] en 3 artículos y LS en 1 artículo [24], ver Figura 2.

P3. ¿Cuáles son las técnicas de la minería de flujos de datos más utilizadas? La agrupación (clustering, en inglés), permite agrupar datos con un comportamiento similar. Se puede considerar como una partición o segmentación de elementos en grupos que pueden o no ser disjuntos.

En muchos casos, la respuesta a un problema de agrupación no es única, es decir, se pueden encontrar muchas respuestas, e interpretar el significado práctico de cada agrupación puede ser difícil [4]. La clasificación, (classification, en inglés), asigna datos en grupos predefinidos (clases). Su diferencia con clustering es que, en esta, el número de grupos está predeterminado y fijo [53].

Por otra parte, la técnica de conteo de frecuencias, (frequency counting, en inglés), permite el conteo de frecuencias para identificar conjuntos de elementos frecuentes. Sin embargo, aunque la minería de conjuntos de elementos frecuentes se ha estudiado ampliamente en la minería de datos, extenderla a los flujos de datos es un desafío, especialmente para los flujos con distribuciones no estáticas [51].

Por último, la técnica de análisis de series de tiempo, (time series analysis, en inglés), modela flujos de datos que contienen solo valores numéricos como series de tiempo. Las tareas de minería de flujos en series temporales se pueden clasificar brevemente en dos tipos: detección de patrones y análisis de tendencias [48, 67].

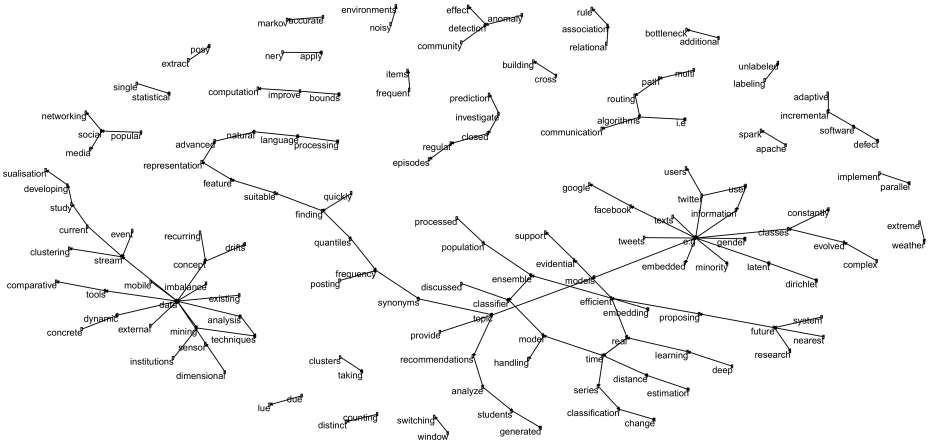


Fig. 4. Trigrama de trabajos futuros en DSM.

Derivado de nuestro análisis se encontró que la clasificación es la técnica de DSM que se emplea con mayor frecuencia. Como se observa en la Tabla 2, la distribución del uso de las técnicas de DSM fue: 34 artículos de clasificación, 15 de conteo de frecuencias, 13 de agrupación y 6 artículos sobre análisis de series de tiempo.

P4. ¿Cuáles son las herramientas que se emplean con mayor frecuencia? Los resultados nos muestran el uso de Apache Hadoop, Apache Spark, Mapreduce y Kafka. Así como tambien las Api´s de Twitter y Python para la generación de flujos. Otras herramientas relevantes que figuran en los trabajos son los frameworks de MOA y Java, Weka, Esper y MATLAB. Por otra parte, tambien se resalta el uso de una plataforma de IBM que no es opensource pero es una de las más completas para el procesamiento de flujos (ver Figura 3).

P5. ¿Cuáles son las principales temáticas pendientes por abordar en el área? En la Figura 4 se muestra un Tri-grama para identificar agrupaciones de tres términos más asociados a Data Stream Mining.

En el lado izquierdo se muestra una de las ramificaciones principales donde se observa la agrupación de tres términos Clustering-Stream-Event, es decir la agrupación de flujos de datos para la detección de eventos es un área de la minería de flujos que actualmente esta tomando ventaja en la investigación pues se requiere realizar análisis considerando la variación en la distribución de los flujos (Concept drift) y su efecto en la conformación de los clusters que también varían con la llegada de nuevos flujos.

También se requiere realizar análisis comparando diferentes herramientas que permitan el desarollo de la visualización de las agrupaciones.

Por otra parte, en el lado derecho de la red se observa una ramificación más grande que conecta varios temas de interés con el uso de técnicas de series de tiempo y clasificación en la minería de flujos: Técnicas de procesamiento de lenguaje natural (PLN), aprendizaje profundo (deep learning) y métodos de ensamble para la clasificación de flujos, evolución de clases complejas, el uso de series de tiempo para clasificación y su aplicación en flujos de textos que se generan en redes sociales como Facebook, Twitter o consultas en la Web como Google.

4. Conclusiones

Recientemente, se cuenta con aplicaciones en las que se generan flujos de datos a tasas muy altas con variaciones en el tiempo y posiblemente impredecibles.

La señal emitida por la entrada de los flujos de datos se puede modelar de diferentes formas siendo mas representativo el modelo de ventana deslizante.

Asímismo, aunque existen diferentes enfoques para realizar un preprocesamiento de los flujos, las técnicas mas utilizadas son el muestreo, sinopsis y agregación. Por otra parte, para realizar la extracción de conocimiento e información ocultos en flujos, se necesita el empleo de técnicas de DSM que permiten realizar tareas específicas como el agrupamiento, clasificación, conteo de frecuencias y análisis de series de tiempo, siendo de mayor uso la técnica de clasificación. Cabe resaltar que existen diferentes herramientas que permiten las generación de flujos y su procesamiento como son; Apache Hadoop, Apache Spark y MOA.

Finalmente, el análisis de flujos de datos masivos, aplicando técnicas de DSM aun tiene varios desafíos que resolver como es el caso de la agrupación de flujos para la detección de eventos, aplicaciones relacionadas con PLN, aprendizaje profundo, el uso de series de tiempo para clasificación y su aplicación en flujos de textos que se generan en redes sociales, entre otros.

Agradecimientos. Al Consejo Nacional de Ciencia y Tecnología (CONACYT) de México en el marco del proyecto de Cátedras "Infraestructura para Agilizar el Desarrollo de Sistemas Centrados en el Usuario" (Ref. 3053). También por la beca de Doctorado número 743385 del primer autor, así como a la Universidad Veracruzana.

Referencias

1. Adeniyi, D. A., Wei, Z., Yongquan, Y.: Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method. Applied Computing and Informatics, vol. 12, no. 1, pp. 90–108 (2016) doi: 10.1016/j.aci.2014.10.001
2. Aggarwal, C. C.: Data streams: models and algorithms. vol. 31. Springer Science & Business Media (2007) doi: 10.1007/978-0-387-47534-9
3. Aggarwal, C. C., Han, J., Wang, J., Yu, P. S.: A framework for projected clustering of high dimensional data streams. In: Proceedings of the Thirtieth international conference on Very large data bases, vol. 30, pp. 852–863 (2004)
4. Aggarwal, C. C., Philip, S. Y., Han, J., Wang, J.: A framework for clustering evolving data streams. In: Proceedings of the 2003 Very Large Data Base Endowment Inc. Conference, pp. 81–92 (2003) doi: 10.1016/b978-012722442-8/50016-1
5. Ait-Alla, A., Lütjen, M., Lewandowski, M., Freitag, M., Thoben, K. D.: Real-time fault detection for advanced maintenance of sustainable technical systems. Procedia CIRP, vol. 41, pp. 295–300 (2016) doi: 10.1016/j.procir.2016.01.015
6. Akbar, A., Kousiouris, G., Pervaiz, H., Sancho, J., Ta-Shma, P., Carrez, F., Moessner, K.: Real-time probabilistic data fusion for large-scale iot applications. IEEE Access, vol. 6, pp. 10015–10027 (2018) doi: 10.1109/access.2018.2804623
7. Amphawan, K., Soulas, J., Lenca, P.: Mining top-k regular episodes from sensor streams. Procedia Computer Science, vol. 69, pp. 76–85 (2015) doi: 10.1016/j.procs.2015.10.008

8. Anupama, N., Jena, S.: A novel approach using incremental oversampling for data stream mining. Evolving Systems, vol. 10, no. 3, pp. 351–362 (2019) doi: 10.1007/s12530-018-9249-5
9. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: Semantic foundations and query execution. The VLDB Journal, vol. 15, no. 2, pp. 121–142 (2006) doi: 10.1007/s00778-004-0147-z
10. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 1–16 (2002) doi: 10.1145/543613.543615
11. Babcock, B., Datar, M., Motwani, R.: Load shedding techniques for data stream systems. In: Proceedings of the 2003 Workshop on Management and Processing of Data Streams, vol. 577 (2003)
12. Benjelloun, F. Z., Oussous, A., Bennani, A., Belfkih, S., Lahcen, A. A.: Improving outliers detection in data streams using LiCS and voting. Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 10 (2021) doi: 10.1016/j.jksuci.2019.08.003
13. Bhagat, A., Kshirsagar, N., Khodke, P., Dongre, K., Ali, S.: Penalty parameter selection for hierarchical data stream clustering. Procedia Computer Science, vol. 79, pp. 24–31 (2016) doi: 10.1016/j.procs.2016.03.005
14. Bodyanskiy, Y. V., Tyshchenko, O. K., Kopaliani, D. S.: Adaptive learning of an evolving cascade neo-fuzzy system in data stream mining tasks. Evolving Systems, vol. 7, no. 2, pp. 107–116 (2016) doi: 10.1007/s12530-016-9149-5
15. Bones, C. C., Romani, L. A., de Sousa, E. P.: Improving multivariate data streams clustering. Procedia Computer Science, vol. 80, pp. 461–471 (2016), doi: 10.1016/j.procs.2016.05.325
16. Braun, P., Cameron, J. J., Cuzzocrea, A., Jiang, F., Leung, C. K.: Effectively and efficiently mining frequent patterns from dense graph streams on disk. Procedia Computer Science, vol. 35, pp. 338–347 (2014) doi: 10.1016/j.procs.2014.08.114
17. Cerquitelli, T.: Predicting large scale fine grain energy consumption. Energy Procedia, vol. 111, pp. 1079–1088 (2017) doi: 10.1016/j.egypro.2017.03.271
18. Chao, S. C., Lin, K. C. J., Chen, M. S.: Flow classification for software-defined data centers using stream mining. IEEE Transactions on Services Computing, vol. 12, no. 1, pp. 105–116 (2016) doi: 10.1109/tsc.2016.2597846
19. Chen, X., Vorvoreanu, M., Madhavan, K.: Mining social media data for understanding students' learning experiences. IEEE Transactions on learning technologies, vol. 7, no. 3, pp. 246–259 (2014) doi: 10.1109/TLT.2013.2296520
20. Cuzzocrea, A., Han, Z., Jiang, F., Leung, C. K., Zhang, H.: Edge-based mining of frequent subgraphs from graph streams. Procedia Computer Science, vol. 60, pp. 573–582 (2015) doi: 10.1016/j.procs.2015.08.184
21. Czarnowski, I., Jedrzejowicz, P.: Ensemble classifier for mining data streams. Procedia Computer Science, vol. 35, pp. 397–406 (2014) doi: 10.1016/j.procs.2014.08.120
22. Dai, C., Chen, L.: An algorithm for mining frequent closed itemsets in data stream. Physics Procedia, vol. 24, pp. 1722–1728 (2012) doi: 10.1016/j.phpro.2012.02.254
23. Dao, M. S., Nguyen Gia, T. A., Mai, V. C.: Daily human activities recognition using heterogeneous sensors from smartphones. Procedia computer science, vol. 111, pp. 323–328 (2017) doi: 10.1016/j.procs.2017.06.030
24. Desai, D., Joshi, A.: A deviant load shedding system for data stream mining. Procedia Computer Science, vol. 45, pp. 118–126 (2015) doi: 10.1016/j.procs.2015.03.103
25. Domingos, P., Hulten, G.: A general method for scaling up machine learning algorithms and its application to clustering. In: Proceedings of the Eighteenth International Conference on Machine Learning, vol. 1, pp. 106–113 (2001)

26. Drosou, A., Kalamaras, I., Papadopoulos, S., Tzovaras, D.: An enhanced graph analytics platform (gap) providing insight in big network data. Journal of Innovation in Digital Ecosystems, vol. 3, no. 2, pp. 83–97 (2016) doi: 10.1016/j.jides.2016.10.005
27. Erra, U., Frola, B.: Frequent items mining acceleration exploiting fast parallel sorting on the GPU. Procedia Computer Science, vol. 9, pp. 86–95 (2012) doi: 10.1016/j.procs.2012.04.010
28. Farhat, A., Gouider, M. S., Said, L. B.: New algorithm for frequent itemsets mining from evidential data streams. In: Proceedings of the 20th International Conference KES, pp. 645–653 (2016) doi: 10.1016/j.procs.2016.08.246
29. Farzanyar, Z., Kangavari, M., Cercone, N.: Max-FISM: Mining (recently) maximal frequent itemsets over data streams using the sliding window model. Computers & Mathematics with Applications, vol. 64, no. 6, pp. 1706–1718 (2012)
30. Fong, S., Li, J., Song, W., Tian, Y., Wong, R. K., Dey, N.: Predicting unusual energy consumption events from smart home sensor network by data stream mining with misclassified recall. Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 4, pp. 1197–1221 (2018) doi: 10.1007/s12652-018-0685-7
31. Fong, S., Liu, K., Cho, K., Wong, R., Mohammed, S., Fiaidhi, J.: Improvised methods for tackling big data stream mining challenges: Case study of human activity recognition. The Journal of Supercomputing, vol. 72, no. 10, pp. 3927–3959 (2016) doi: 10.1007/s11227-016-1639-5
32. Fong, S., Wong, R., Vasilakos, A. V.: Accelerated PSO swarm search feature selection for data stream mining big data. IEEE Transactions on Services Computing, vol. 9, no. 1, pp. 33–45 (2015) doi: 10.1109/tsc.2015.2439695
33. Gaber, M. M., Krishnaswamy, S., Gillick, B., AlTaiar, H., Nicoloudis, N., Liono, J., Zaslavsky, A.: Interactive self-adaptive clutter-aware visualisation for mobile data mining. Journal of Computer and System Sciences, vol. 79, no. 3, pp. 369–382 (2013) doi: 10.1016/j.jcss.2012.09.009
34. Gaber, M. M., Zaslavsky, A., Krishnaswamy, S.: Mining data streams: A review. ACM Sigmod Record, vol. 34, no. 2, pp. 18–26 (2005) doi: 10.1145/1083784.1083789
35. Gama, J.: Knowledge discovery from data streams, vol. 12. IOS Press (2010), doi: 10.3233/ida-2008-123
36. Gama, J., Gaber, M. M.: Learning from data streams. Processing techniques in sensor networks. Springer (2007) doi: 10.1007/3-540-73679-4
37. Gao, T., Li, A., Meng, F.: Research on data stream clustering based on FCM algorithm 1. Procedia Computer Science, vol. 122, pp. 595–602 (2017) doi: 10.1016/j.procs.2017.11.411
38. Ghesmoune, M., Lebbah, M., Azzag, H.: State-of-the-art on clustering data streams. Big Data Analytics, vol. 1, no. 1, pp. 13 (2016) doi: 10.1186/s41044-016-0011-3
39. Gilbert, A. C., Kotidis, Y., Muthukrishnan, S., Strauss, M.: Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In: Proceedings of the 27th International Conference on Very Large Data Bases, vol. 1, pp. 79–88 (2001)
40. Gilbert, A. C., Kotidis, Y., Muthukrishnan, S., Strauss, M. J.: One-pass wavelet decompositions of data streams. IEEE Transactions on knowledge and data engineering, vol. 15, no. 3, pp. 541–554 (2003) doi: 10.1109/tkde.2003.1198389
41. Golab, L., Özsu, M. T.: Issues in data stream management. ACM Sigmod Record, vol. 32, no. 2, pp. 5–14 (2003) doi: 10.1145/776985.776986
42. Golab, L., Ozsu, M. T.: Data stream management. Morgan & Claypool Publishers (2010)
43. Groß-Klußmann, A., König, S., Ebner, M.: Buzzwords build momentum: Global financial twitter sentiment and the aggregate stock market. Expert Systems with Applications, vol. 136, pp. 171–186 (2019) doi: 10.1016/j.eswa.2019.06.027
44. Guo, J., Zhang, P., JianlongTan, Guo, L.: Mining hot topics from twitter streams. Procedia Computer Science, vol. 9, pp. 2008–2011 (2012) doi: 10.1016/j.procs.2012.04.224

45. Hassan, M. H., Tizghadam, A., Leon-Garcia, A.: Spatio-temporal anomaly detection in intelligent transportation systems. Procedia Computer Science, vol. 151, pp. 852–857 (2019) doi: 10.1016/j.procs.2019.04.117
46. Hemalatha, C. S., Vaidehi, V.: Frequent bit pattern mining over tri-axial accelerometer data streams for recognizing human activities and detecting fall. Procedia Computer Science, vol. 19, pp. 56–63 (2013) doi: 10.1016/j.procs.2013.06.013
47. Hu, Y., Jiang, Z., Zhan, P., Zhang, Q., Ding, Y., Li, X.: A novel multi-resolution representation for streaming time series. Procedia Computer Science, vol. 129, pp. 178–184 (2018) doi: 10.1016/j.procs.2018.03.069
48. Indyk, P., Koudas, N., Muthukrishnan, S.: Identifying representative trends in massive time series data sets using sketches. In: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 363–372 (2000)
49. Jankowski, D., Jackowski, K., Cyganek, B.: Learning decision trees from data streams with concept drift. Procedia Computer Science, vol. 80, pp. 1682–1691 (2016) doi: 10.1016/j.procs.2016.05.508
50. Jaysawal, B. P., Huang, J.-W.: PSP-AMS: Progressive mining of sequential patterns across multiple streams. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 13, no. 1, pp. 1–23 (2018) doi: 10.1145/3281632
51. Jiang, N., Gruenwald, L.: Research issues in data stream association rule mining. ACM SIGMOD Record, vol. 35, no. 1, pp. 14–19 (2006) doi: 10.1145/1121995.1121998
52. Kazanskiy, N., Protsenko, V., Serafimovich, P.: Performance analysis of real-time face detection system based on stream data mining frameworks. Procedia Engineering, vol. 201, pp. 806–816 (2017) doi: 10.1016/j.proeng.2017.09.602
53. Khan, M., Ding, Q., Perrizo, W.: k-nearest neighbor classification on spatial data streams using p-trees. In: Advances in Knowledge Discovery and Data Mining, pp. 517–528 (2002) doi: 10.1007/3-540-47887-6_51
54. Khodabakhsh, A., Arí, I., Bakír, M., Ercan, A. O.: Multivariate sensor data analysis for oil refineries and multi-mode identification of system behavior in real-time. IEEE Access, vol. 6, pp. 64389–64405 (2018), doi: 10.1109/access.2018.2877097
55. Kianfar, J., Edara, P.: A data mining approach to creating fundamental traffic flow diagram. Procedia-Social and Behavioral Sciences, vol. 104, no. 1, pp. 430–439 (2013) doi: 10.1016/j.sbspro.2013.11.136
56. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University, vol. 33, no. 2004, pp. 1–26 (2004)
57. Kitchenham, B. A., Budgen, D., Brereton, O. P.: Using mapping studies as the basis for further research–a participant-observer case study. Information and Software Technology, vol. 53, no. 6, pp. 638–651 (2011) doi: 10.1016/j.infsof.2010.12.011
58. Ko, J. M., Hong, S. R., Choi, J. Y., Kim, C. O.: Wafer-to-wafer process fault detection using data stream mining techniques. International Journal of Precision Engineering and Manufacturing, vol. 14, no. 1, pp. 103–113 (2013), doi: 10.1007/s12541-013-0015-0
59. Kolajo, T., Daramola, O., Adebiyi, A.: Big data stream analysis: A systematic literature review. Journal of Big Data, vol. 6, no. 1, pp. 47 (2019) doi: 10.1186/s40537-019-0210-7
60. Krawczyk, B.: GPU-accelerated extreme learning machines for imbalanced data streams with concept drift. Procedia Computer Science, vol. 80, pp. 1692–1701 (2016) doi: 10.1016/j.procs.2016.05.509
61. Kusumakumari, V., Sherigar, D., Chandran, R., Patil, N.: Frequent pattern mining on stream data using hadoop cantree-gtree. Procedia Computer Science, vol. 115, pp. 266–273 (2017) doi: 10.1016/j.procs.2017.09.134
62. Le, T., Stahl, F., Gaber, M. M., Gomes, J. B., Di Fatta, G.: On expressiveness and uncertainty awareness in rule-based classification for data streams. Neurocomputing, vol. 265, pp. 127–141 (2017) doi: 10.1016/j.neucom.2017.05.081

63. Li, P., Wu, X., Hu, X.: Mining recurring concept drifts with limited labeled streaming data. In: Proceedings of 2nd Asian conference on machine learning, vol. 3, pp. 1–32 (2012) doi: 10.1145/2089094.2089105
64. Liang, C., Li, M., Liu, B.: Online computing quantile summaries over uncertain data streams. IEEE Access, vol. 7, pp. 10916–10926 (2019) doi: 10.1109/access.2019.2891550
65. Lifna, C., Vijayalakshmi, M.: Identifying concept-drift in twitter streams. Procedia Computer Science, vol. 45, pp. 86–94 (2015)
66. Lim, S., Tucker, C. S.: Mining twitter data for causal links between tweets and real-world outcomes. Expert Systems with Applications: X, vol. 3 (2019) doi: 10.1016/j.eswax.2019.100007
67. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pp. 2–11 (2003) doi: 10.1145/882082.882086
68. Liu, H., Lin, Y., Han, J.: Methods for mining frequent items in data streams: An overview. Knowledge and information systems, vol. 26, no. 1, pp. 1–30 (2009) doi: 10.1007/s10115-009-0267-2
69. Liu, Y., Ma, P., Cui, H.: Design and development of fpga-based high-performance radar data stream mining system. Procedia Computer Science, vol. 55, pp. 876–885 (2015) doi: 10.1016/j.procs.2015.07.147
70. Miholca, D. L., Czibula, G., Crivei, L. M.: A new incremental relational association rules mining approach. Procedia Computer Science, vol. 126, pp. 126–135 (2018) doi: 10.1016/j.procs.2018.07.216
71. Molano-Pulido, J., Jiménez-Guarín, C.: SEAbIRD: Adaptable daily living activity identification from sensor data streams. Procedia Computer Science, vol. 130, pp. 939–946 (2018) doi: 10.1016/j.procs.2018.04.093
72. Muthukrishnan, S.: Data streams: Algorithms and applications. Foundations and Trends® in Theoretical Computer Science, vol. 1, no. 2, pp. 117–236 (2005) doi: 10.1561/0400000002
73. Nalavade, J. E., Murugan, T. S.: HRNeuro-fuzzy: Adapting neuro-fuzzy classifier for recurring concept drift of evolving data streams using rough set theory and holoentropy. Journal of King Saud University-Computer and Information Sciences, vol. 30, no. 4, pp. 498–509 (2018) doi: 10.1016/j.jksuci.2016.11.005
74. Nguyen, H. L., Woon, Y. K., Ng, W. K.: A survey on data stream clustering and classification. Knowledge and Information Systems, vol. 45, no. 3, pp. 535–569 (2015) doi: 10.1007/s10115-014-0808-1
75. Peng, M., Zhu, J., Wang, H., Li, X., Zhang, Y., Zhang, X., Tian, G.: Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding. ACM Transactions on Knowledge Discovery from Data, vol. 12, no. 3, pp. 1–26 (2018)
76. Qu, Z. Y., Zhang, L., Ding, G. L.: A data stream mining algorithm used in power grid disturbance identification. Physics Procedia, vol. 24, pp. 966–970 (2012) doi: 10.1016/j.phpro.2012.02.145
77. Rashid, M. M., Gondal, I., Kamruzzaman, J.: ACSP-tree: A tree structure for mining behavioral patterns from wireless sensor networks. In: Proceedings of the 38th Annual IEEE Conference on Local Computer Networks, pp. 691–694 (2013) doi: 10.1109/lcn.2013.6761312
78. Rutkowski, L., Jaworski, M., Pietruczuk, L., Duda, P.: Decision trees for mining data streams based on the gaussian approximation. IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 1, pp. 108–119 (2013) doi: 10.1109/tkde.2013.34
79. Rutkowski, L., Pietruczuk, L., Duda, P., Jaworski, M.: Decision trees for mining data streams based on the McDiarmid's bound. IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 6, pp. 1272–1279 (2012) doi: 10.1109/tkde.2012.66

80. Shi, L. L., Liu, L., Wu, Y., Jiang, L., Hardy, J.: Event detection and user interest discovering in social media data streams. IEEE Access, vol. 5, pp. 20953–20964 (2017) doi: 10.1109/access.2017.2675839
81. Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., De Carvalho, A. C., Gama, J.: Data stream clustering: A survey. ACM Computing Surveys, vol. 46, no. 1, pp. 1–31 (2013)
82. Spezzano, G., Vinci, A.: Pattern detection in cyber-physical systems. Procedia Computer Science, vol. 52, pp. 1016–1021 (2015) doi: 10.1016/j.procs.2015.05.096
83. Srimani, P., Malini, M. P.: Frequent item set mining using INC_MINE in massive online analysis frame work. Procedia Computer Science, vol. 45, pp. 133–142 (2015) doi: 10.1016/j.procs.2015.03.105
84. Subbian, K., Aggarwal, C., Srivastava, J.: Mining influencers using information flows in social streams. ACM Transactions on Knowledge Discovery from Data, vol. 10, no. 3, pp. 1–28 (2016) doi: 10.1145/2815625
85. Sun, Y., Tang, K., Minku, L. L., Wang, S., Yao, X.: Online ensemble learning of data streams with gradually evolved classes. IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 6, pp. 1532–1545 (2016) doi: 10.1109/tkde.2016.2526675
86. Tajalizadeh, H., Boostani, R.: A novel stream clustering framework for spam detection in twitter. IEEE Transactions on Computational Social Systems, vol. 6, no. 3, pp. 525–534 (2019) doi: 10.1109/TCSS.2019.2910818
87. Tennant, M., Stahl, F., Rana, O., Gomes, J. B.: Scalable real-time classification of data streams with concept drift. Future Generation Computer Systems, vol. 75, pp. 187–199 (2017) doi: 10.1016/j.future.2017.03.026
88. Vadivukarassi, M., Puviarasan, N., Aruna, P.: A framework of keyword based image retrieval using proposed hog_sift feature extraction method from twitter dataset. Procedia Computer Science, vol. 132, pp. 1422–1431 (2018), doi: 10.1016/j.procs.2018.05.073
89. Weng, Y., Liu, L.: A collective anomaly detection approach for multidimensional streams in mobile service security. IEEE Access, vol. 7, pp. 49157–49168 (2019) doi: 10.1109/access.2019.2909750
90. Won, S., Cho, I., Sudusinghe, K., Xu, J., Zhang, Y., Van Der Schaar, M., Bhattacharyya, S. S.: A design methodology for distributed adaptive stream mining systems. Procedia Computer Science, vol. 18, pp. 2482–2491 (2013), doi: 10.1016/j.procs.2013.05.425
91. You, D., Wu, X., Shen, L., Deng, S., Chen, Z., Ma, C., Lian, Q.: Online feature selection for streaming features using self-adaption sliding-window sampling. IEEE Access, vol. 7, pp. 16088–16100 (2019) doi: 10.1109/access.2019.2894121
92. Yu, K., Ding, W., Simovici, D. A., Wang, H., Pei, J., Wu, X.: Classification with streaming features: An emerging-pattern mining approach. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 9, no. 4, pp. 1–31 (2015) doi: 10.1145/2700409
93. Za'in, C., Pratama, M., Lughofer, E., Ferdaus, M., Cai, Q., Prasad, M.: Big data analytics based on panfis mapreduce. Procedia Computer Science, vol. 144, pp. 140–152 (2018) doi: 10.1016/j.procs.2018.10.514
94. Zhang, S., Zhang, Y., Yin, L., Yuan, T., Wu, Z., Luo, H.: Mining frequent items over the distributed hierarchical continuous weighted data streams in internet of things. IEEE Access, vol. 7, pp. 74890–74898 (2019) doi: 10.1109/access.2019.2911573
95. Zhou, Z., Zhang, X., Zhou, X., Liu, Y.: Semantic-aware visual abstraction of large-scale social media data with geo-tags. IEEE Access, vol. 7, pp. 114851–114861 (2019) doi: 10.1109/access.2019.2935471
96. Zhu, Y., Shasha, D.: Efficient elastic burst detection in data streams. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 336–345 (2003) doi: 10.1145/956750.956789

ISSN 1870-4069

Modelo basado en agentes para exploración y optimización de trayectorias mediante cómputo bioinspirado

Marco A. Jiménez-Limas[1,2,3], Emmanuel Salcedo-Carrillo[4], Fernanda Borjas[2], Amaury Salcedo-Carrillo[5]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

[2] Universidad Nacional Autónoma de México,
Facultad de Ciencias,
México

[3] Universidad Autónoma de la Ciudad de México,
Posgrado en Ciencias de la Complejidad,
México

[4] Instituto Politécnico Nacional,
Escuela Nacional de Ciencias Biológicas,
México

[5] Instituto Politécnico Nacional
Centro de Investigación y de Estudios Avanzados,
Departamento de Fisiología, Biofísica y Neurociencias,
México

{marcojimenez, fernandaborjas}@ciencias.unam.mx,
escsal97@gmail.com, amaury.salcedo@cinvestav.mx

Resumen. El comportamiento colectivo durante el forrajeo es un fenómeno ubicuo en la naturaleza; diversos enfoques se han propuesto para modelar este tipo de comportamiento, dentro de los que destacan los modelos basados en agentes (MBA). En el presente trabajo se realizó un modelo de exploración de un espacio desconocido de dos dimensiones utilizando tres tipos de caminatas aleatorias (LF, URW, GRW) para evaluar su eficacia de búsqueda. Posteriormente, se realizó un MBA en tres dimensiones en el que el espacio a explorar consistió en tres componentes principales: punto de partida (*home*), punto de llegada (*target*) y obstáculos. A los agentes se les implementaron reglas de colectividad (separación, alineación y cohesión) y una caminata aleatoria tipo LF para explorar el espacio y así llegar al *target*. Se realizaron 30 simulaciones, que se detenían una vez que un agente alcanzaba el target, para posteriormente generar un grafo utilizando la información recabada por cada agente (la trayectoria recorrida por agente y la posible comunicación local). Se aplicó el algoritmo ACO sobre el grafo obtenido para encontrar la ruta más corta al objetivo, y se comparó la ruta optimizada con la realizada por el agente que alcanzó el *target* para evaluar si había diferencias significativas entre sus distancias. Las distancias optimizadas presentaron una diferencia importante al compararlas contra las originales y se logró una disminución del 22 % en la distancia recorrida del *home* al *target* utilizando el

método planteado. El uso de MBA y caminatas aleatorias, en especial los LF, resultan una forma adecuada de describir la dinámica del forrajeo en animales, ya que toman en cuenta la interacción entre los componentes del sistema y con el ambiente, dando lugar a comportamientos emergentes que no pueden ser descritos a partir únicamente de conductas individuales.

Palabras clave: Inteligencia colectiva, caminata aleatoria, forrajeo.

Agent-Based Model for Exploration and Path Optimization Using Bio-inspired Computing

Abstract. Collective behavior during foraging is a ubiquitous phenomenon in nature; several approaches have been developed to model this type of behavior, within which stand out Agent-Based Models (ABM). In the present work, a two-dimensional exploration model of an unknown space was carried out using three types of random walks (LF, URW, GRW) to evaluate its search efficiency. Later, a three-dimensional ABM was developed, in which the space to explore has three main components: home, target and obstacles. The agents were implemented collective rules (separation, alignment, and cohesion) and a LF random walk to explore the space and thus reach the target. 30 simulations were carried out, which stopped once an agent reached the target, to later generate a graph using the information collected by each agent. The ACO algorithm was applied on the obtained graph to find the shortest path to the target, and the optimized route was compared with the route taken by the agent who reached the target to evaluate if there were significant differences between their distances. The optimized paths presented an important difference when compared against the originals, and a 22% decrease in the distance traveled from home to the target was achieved using the proposed method. The use of ABM and random walks, especially the LF, are an adequate way to describe the dynamics of foraging in animals, since they consider the interaction between the components of the system and with the environment, giving rise to emergent behaviors that cannot be described solely based on individual behaviors.

Keywords: Collective intelligence, random walks, foraging.

1. Introducción

La presencia de comportamiento colectivo en ciertos grupos de animales es un fenómeno ubicuo en la naturaleza. Se han brindado diversas explicaciones a este tipo de comportamiento, dentro de las que destacan la evitación de depredadores y el forrajeo [1]. Este tipo de agrupaciones pueden ser vistos como sistemas colectivos, los cuales son sistemas complejos de agentes que interactúan entre sí y en los que cada agente tiene una tarea propia a realizar [2].

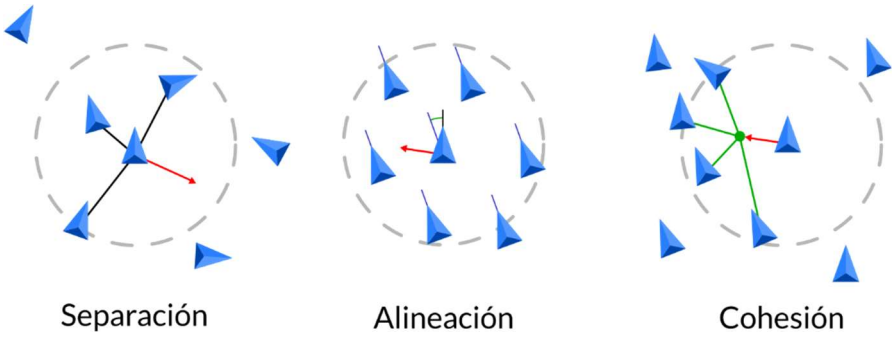


Fig. 1. Reglas de colectividad: Separación, Alineación y Cohesión. Dado un agente con un rango de visión, su vecindad se define como todos aquellos agentes que se encuentran a su alrededor dentro del radio dado por el rango de visión.

Algunos ejemplos de estos sistemas son las colonias de hormigas, termitas o las colmenas de abejas, las cuales están constituidas de muchos individuos con una función específica [3]. Otros sistemas que presentan comportamiento colectivo sin una división de la labor son los bancos de peces o las parvadas de pájaros.

El estudio del comportamiento colectivo en sistemas de animales es ampliamente utilizado dentro del ámbito del cómputo bioinspirado, especialmente para el desarrollo de algoritmos de exploración y de optimización. Uno de los primeros acercamientos al modelado del comportamiento colectivo de animales fue la simulación *Boids*, desarrollada por Craig Reynolds, la cual tuvo como objetivo la replicación del comportamiento de las parvadas de aves o de los bancos de peces utilizando geometría computacional 3D [4].

Este modelo implementa tres reglas simples a nivel local que dan lugar a la colectividad: i) *separación:* cada agente debe mantener una distancia entre sí para evitar la sobre agrupación; ii) *alineación:* cada agente ajusta su posición para que corresponda con el alineamiento promedio del resto de los agentes; iii) *cohesión:* cada agente intenta moverse a la posición promedio de otros agentes. Estas reglas describen como cada agente reacciona respecto a la ubicación y velocidades del resto de agentes cercanos en el Flock (Figura 1).

Cada agente es independiente, ya que posee una velocidad, posición y orientación propia. Sin embargo, estas propiedades se ven alteradas por las reglas de colectividad, cuando hay otros agentes en su vecindad (la cual se caracteriza por una distancia medida desde el centro del agente y un ángulo medido desde la dirección de vuelo del agente). Esto ocasiona que cualquier agente que se localice fuera de esa región sea ignorado (Figura 1).

En cuanto a los algoritmos bioinspirados de exploración, un método ampliamente utilizado es el uso de caminatas aleatorias (RW) [5]. Dentro de ellas podemos encontrar caminatas aleatorias uniformes, gaussianas o con difusión anómala (como los vuelos de Lévy). Las RW son procesos estocásticos que describen el movimiento de agentes o partículas en un espacio, en el que los agentes se mueven a una distancia determinada por una distribución de probabilidad y un ángulo aleatorio.

Las caminatas aleatorias uniformes (URW por sus siglas en inglés) tienen un paso que viene dado por una distribución uniforme; en las gaussianas (GRW por sus siglas en inglés), la longitud del paso está dada por una distribución normal; en los vuelos de Lévy

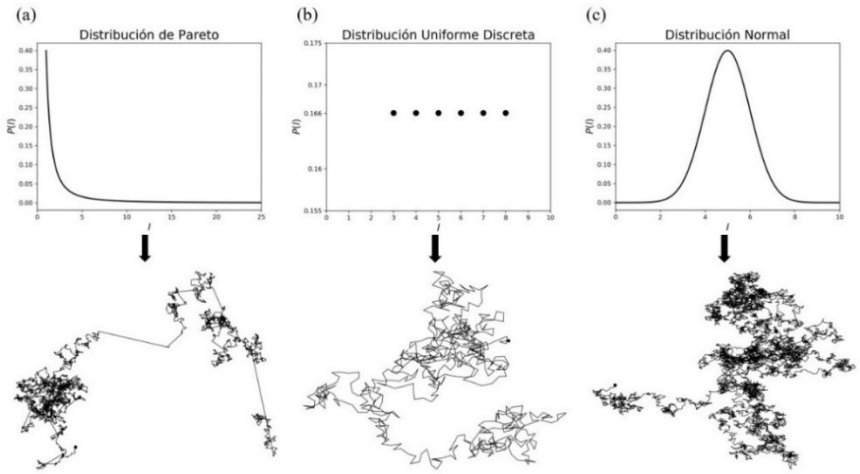


Fig. 2. (a) Distribución de Pareto que da lugar al vuelo de Lévy; (b) Distribución uniforme discreta que da lugar a una caminata aleatoria uniforme; (c) Distribución normal que da lugar a una caminata aleatoria gaussiana.

(LF por sus siglas en inglés), la longitud del paso está dada por una distribución libre de escala (anómala), como puede ser una distribución de Pareto (Figura 2).

Existen también diversos algoritmos bioinspirados para optimizar (maximizar o minimizar) una función, dentro de los cuales están los basados en hormigas (ACO: Ant Colony Optimization) o abejas (ABC: Artificial Bee Colony) [6]. Dentro del campo de la inteligencia artificial, un paradigma bioinspirado es la Inteligencia de enjambre (Swarm intelligence), la cual es definida como "la inteligencia colectiva emergente de los grupos de agentes simples" [7, 8, 9]. El ACO es una metaheurística que engloba una serie de técnicas de optimización inspiradas en el comportamiento colectivo de las hormigas durante el forrajeo, mediante la cual es posible dar solución a problemas computacionales, como la búsqueda de la trayectoria más corta entre dos nodos en un grafo [10].

La modelación basada en agentes es una herramienta ampliamente utilizada para estudiar comportamientos colectivos de distintos grupos de animales. Dentro de algunos trabajos que se relacionan con estos tópicos destaca el realizado por Sellers y colaboradores [11] se utilizó un MBA para modelar y simular el forrajeo colectivo de babuinos, con el cual obtuvieron resultados que son concordantes con el comportamiento etológico de este grupo de animales.

Por otro lado, Robinson y colaboradores [12] hicieron un MBA para entender el rol de las feromonas en un grupo social de hormigas durante el forrajeo, y encontraron que diferentes tipos de feromonas tienen efectos determinantes en la dinámica del forrajeo colectivo. Otros enfoques toman en cuenta el ambiente en el que se desenvuelven estos grupos de animales, como el estudio desarrollado por Rivière y colaboradores [13], quienes desarrollaron un MBA para modelar y simular el forrajeo colectivo de abejas utilizando datos del ambiente como el tiempo local y las fuentes de néctar cercanas; sus resultados mostraron la formación de procesos autoorganizados de selección de las mejores fuentes de néctar disponibles por parte de los agentes.

En este trabajo se presenta un modelo basado en agentes (MBA) capaz de simular en 3D la exploración y optimización de trayectorias utilizando como base el comportamiento de sistemas colectivos. Para la exploración, se utilizaron caminatas aleatorias combinadas con el modelo *Boids,* variando el tipo de caminata aleatoria para encontrar aquella que realiza una búsqueda más eficiente. Para la optimización de la trayectoria se utilizó el algoritmo ACO.

2. Métodos

El trabajo se llevó a cabo en tres partes, en la primera se exploraron tres tipos de caminatas aleatorias (uniformes, gaussiana y vuelos de Lévy) y se examinó cuál de las tres fue la más eficiente en cuanto al tiempo de convergencia (número de iteraciones en el que los agentes alcanzaron el objetivo; es decir, a la fuente de alimento). En la segunda parte se realizó un modelo basado en agentes, en el cual se implementó el modelo *Boids* [4] para simular el forrajeo colectivo en un espacio tridimensional utilizando los tres tipos de caminatas aleatorias. De esta simulación se obtuvo un grafo cuyos nodos fueron las posiciones que tuvieron todos los agentes durante el forrajeo. Finalmente, se aplicó el ACO en dicho grafo para encontrar la ruta más corta del origen al objetivo.

2.1. Forrajeo mediante caminatas aleatorias

La parte de la exploración del espacio durante el forrajeo se modeló usando caminantes aleatorios, ya que este tipo de movimiento es el que rige el forrajeo de varios animales en presencia de incertidumbre en el ambiente [14, 15, 16].

Para esta parte, se probó cada uno de los tres tipos de caminata aleatoria utilizando un solo agente, al cual se le implementaron las condiciones correspondientes a cada caminata aleatoria: para la uniforme, se utilizó una distribución uniforme discreta que tomaba los valores 4, 5 y 6 con la misma probabilidad; para la gaussiana, se utilizó una distribución normal con una media de 5 y desviación estándar de 1; y para los vuelos de Levy, se utilizó una distribución de Pareto con $\mu = 2$, paso mínimo de 1 y paso máximo de 25. El agente contaba con un rango de visión r_v y se desplazaba en un espacio bidimensional. En el espacio se colocaron dos puntos importantes: el punto de partida (*home*) y el objetivo (*target*). El *target* representaba la fuente de alimento que el agente estaba buscando.

El modelo de forrajeo usado fue el descrito en [17]; dado un espacio D-dimensional, un agente con un rango de visión r_v y un *target* (Figura 3):

1. Si hay un *target* ubicado dentro del rango de visión, entonces el agente se mueve en una línea recta hacia él. Si hay más de un *target*, se mueve hacia el más cercano.
2. Si no hay un *target* dentro del radio de visión, entonces en su *j*-ésimo paso el agente elige una dirección al azar y una distancia l_j dependiendo del tipo de caminata aleatoria que tenga. Luego se mueve incrementalmente al nuevo punto, buscando continuamente un *target* dentro de su radio de visión durante su desplazamiento. Si no detecta un *target*, se detiene después de recorrer la distancia l_j y elige una nueva dirección y una nueva distancia l_{j+1}. De otra manera se dirige al *target* como en el paso 1.

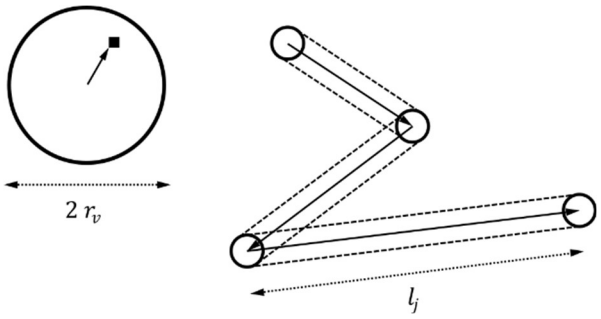


Fig. 3. Modelo de forrajeo [17]: si un *target* (cuadrado obscuro) se encuentra dentro del rango de visión, entonces el agente se mueve directamente hacia el objetivo; de otra forma, el agente escoge una dirección aleatoria y una longitud de paso de acuerdo al tipo de caminata aleatoria que posee y se desplaza buscando continuamente el *target* dentro de su rango de visión.

Se generaron diferentes escenarios para probar la eficiencia de búsqueda de los agentes utilizando los tres tipos de caminatas aleatorias. Para cada una de ellas, se implementaron tres escenarios que variaban en cuanto a la distancia entre el *home*, en la posición (0,0), y el *target*, en las posiciones (10,10): correspondiente a la distancia corta; (-50,0): distancia media; y (90,-90): distancia larga.

El espacio en el que se generaron los escenarios fue un cuadrado centrado en el punto (0,0) cuyos lados tenían longitud de 200 (su rango iba desde -100 hasta 100). Para cada uno de los tratamientos (combinación tipo de caminata y posición del *target*), se colocó a un agente en el espacio y se fijó un límite de 1000 iteraciones para llegar al *target* (el experimento se paraba si el agente no lo alcanzaba en este tiempo, ya que se consideró que este límite era suficiente para evaluar la eficiencia de la caminata)

El valor del rango de visión r_v utilizado fue de 1. Cada uno de los tratamientos se repitió 100 veces y se obtuvo un promedio del número de iteraciones que le tomaba al agente alcanzar el *target*. A partir de los resultados de estas exploraciones, se eligió el tipo de caminata aleatoria más eficiente para la optimización de trayectorias que se describen más adelante.

2.2. Modelo basado en agentes para forrajeo colectivo

Dado que el objetivo del presente trabajo fue modelar el forrajeo colectivo presente en sistemas de animales de la manera más realista posible, se implementó un modelo basado en agentes en tres dimensiones utilizando el lenguaje de programación NetLogo [18], el cual es un software para modelado en 2D y 3D multi-agente, que permite generar código de fácil manejo que puede asociarse con una interfaz gráfica para variar los parámetros programados de manera dinámica durante la simulación.

Se utilizó como base el modelo realizado por [19]. Este modelo utiliza las reglas de colectividad descritas anteriormente (Figura 1a) como base para el comportamiento de los agentes. Este modelo fue modificado para que los agentes pudieran tener distintos tipos de caminatas aleatorias. Además, se añadieron obstáculos al espacio, y a los agentes se les implementó la capacidad de evadir obstáculos y de identificar *targets* dentro de su rango de visión.

Se generó un espacio en tres dimensiones y una interfaz gráfica (GUI por sus siglas en inglés), el espacio consistía en un universo cúbico de 64,000 *patches* o parcelas, con limitación de localización, es decir, con contención en el movimiento de los agentes dentro del espacio.

Se agregaron 40 obstáculos en el espacio con posiciones aleatorias (con la posibilidad de aumentar el número de éstos), además se colocó la posición de *home* y de *target*. A la GUI se le agregaron *sliders* y *switches*, que permitían variar los valores de los parámetros para controlar el tipo de comportamiento deseado en la simulación (tipo de caminata aleatoria, rango de visión, de separación, de alineación y de cohesión).

Para la simulación, se utilizaron 500 agentes, las coordenadas del *home* (0, 0, 0), las coordenadas del *target* (10, -10, 10) y tipo de caminata LF. Con estos parámetros establecidos, se corrió la simulación hasta el momento en el que uno de los agentes tenía en su rango de visión al objetivo. Las posiciones de cada agente se guardaron en un archivo durante cada iteración, y una vez concluida la simulación se utilizó el archivo de posiciones para generar un grafo, cuyos nodos correspondían a las posiciones que tuvieron los agentes a lo largo de la simulación.

Posteriormente, se calcularon las distancias entre nodos y se establecieron sus conexiones, para finalmente calcular la ruta más corta del *home* al *target* en el grafo utilizando el ACO.

2.3. Optimización de trayectorias

Como se mencionó anteriormente, para la optimización se utilizó el algoritmo ACO, el cual consiste en optimizar utilizando como base el comportamiento de forrajeo de las hormigas. Éstas no conocen el territorio al inicio del forrajeo, por lo que empiezan a buscar el alimento de manera aleatoria. Una vez que encuentran la fuente de alimento, regresan a su hogar dejando un rastro de feromona. En esta búsqueda, diferentes hormigas pueden encontrar diferentes rutas a la fuente de alimento, sin embargo, permanece el problema de cuál de todas es la más cercana a su nido.

Esto se resuelve con la concentración de feromonas en los rastros, ya que el camino más corto es aquel en el que las feromonas tardarán más en evaporarse. Al tener más concentración de feromonas, más hormigas van siguiendo este rastro aumentando así su concentración de feromona. De manera estocástica, algunas hormigas salen de esta trayectoria y generan una nueva ruta; de ser una ruta más corta que la original, la concentración de feromona de la ruta tardará menos en evaporarse ocasionando que las hormigas opten por tomar esta ruta.

Este proceso se repite hasta que las hormigas logran encontrar la ruta más corta en su exploración del espacio [10]. El ACO aplica estas ideas sobre un grafo, en el cual las hormigas van recorriendo los nodos para encontrar la ruta más corta entre dos puntos. El algoritmo inicia con una cantidad *n* de hormigas exploradoras definidas por el usuario y con una matriz de concentración de feromonas, donde la concentración inicial también está definida por el usuario.

Para aplicar el ACO, una vez concluida la simulación se obtuvo un archivo con las posiciones espaciales de cada agente en cada iteración. El objetivo del presente trabajo fue encontrar la ruta más corta que un agente puede tomar para llegar del *home* al *target*; dado que se desconoce el espacio, solo es posible obtener información de los puntos por los que pasaron los agentes durante la simulación, y la optimización de la ruta más corta

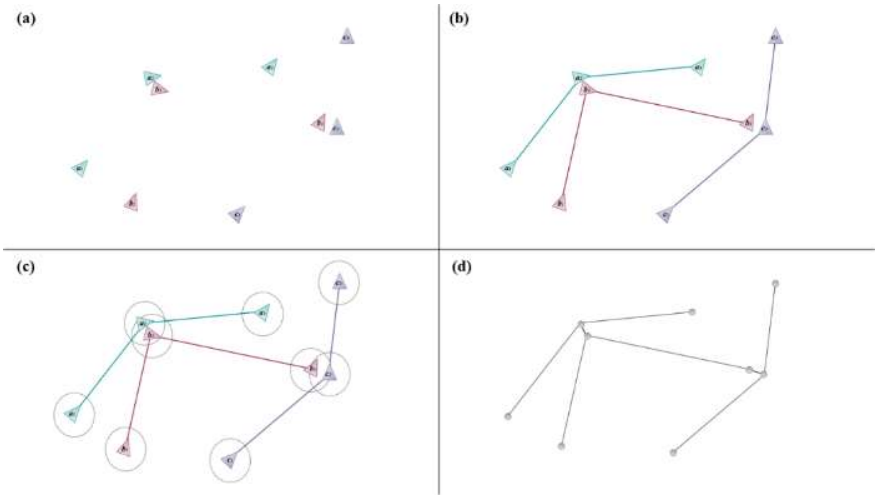


Fig. 4. Proceso de generación del grafo. Se toma el caso de tres agentes *a*, *b* y *c* en tres iteraciones. (a) Se identifican como nodos las posiciones que cada agente tuvo en cada iteración. (b) Se unen los nodos que corresponden a las posiciones de cada agente en las tres iteraciones; así se genera un subgrafo por cada agente correspondiente a su trayectoria recorrida. (c) Para cada nodo se hace una búsqueda de radio *r* a su alrededor para detectar la presencia de nodos cercanos y de ser el caso, se genera una conexión a dichos nodos. (d) El grafo resultante contiene los subgrafos formados por las trayectorias de cada agente y las uniones entre nodos que cuyas posiciones fueron cercanas.

solo puede hacerse sobre dichos puntos conocidos. Para ello, fue necesario convertir las posiciones que fueron teniendo los agentes en un grafo cuyos nodos representan dichas posiciones. Estos nodos fueron conectados mediante aristas utilizando los siguientes criterios.

1. Sea $A = \{a_1, a_2, \ldots, a_n\}$ el conjunto de todos los agentes, donde a_i representa al agente i. Sea $T = \{t_1, t_2, \ldots, t_{m-1}\}$, donde t_j representa la j-ésima iteración; en la simulación se consideraron m iteraciones. Un nodo representa la posición de a_i al tiempo t_j. Se conectó el nodo correspondiente a a_i en el tiempo t_j con el nodo correspondiente a a_i en el tiempo t_{j+1}, para todo agente a en A y para todo t en T (Figura 4a). Esto permitió generar una trayectoria por cada agente que conectaba a todas las posiciones que tuvo ese agente durante la simulación (Figura 4b).
2. Del paso anterior se obtuvo un grafo disconexo que contenía n subgrafos correspondientes a las rutas que siguieron los n agentes; sin embargo, para obtener una ruta más corta es necesario tener un grafo más denso, que utilice la información recabada por todos los agentes. Se procedió a conectar los nodos entre ellos de acuerdo con la distancia que los separaba, para lo cual se seleccionó un parámetro r el cual corresponde a un radio; para cada nodo a_i, se obtuvieron los nodos que estaban a una distancia menor o igual a r respecto a su posición (Figura 4c). Dichos nodos se conectaron a a_i, lo que permitió generar un grafo más denso sobre el cual se llevó a cabo la optimización (Figura 4d).

A partir de la matriz de adyacencias correspondiente al grafo generado se calcularon las distancias entre los nodos conectados para obtener una matriz de distancias. Dicha matriz es la entrada que recibe el algoritmo ACO, por lo que con ella fue posible aplicar la optimización para encontrar la ruta más corta. Los valores de los parámetros utilizados para el ACO fueron los siguientes: $Q = \alpha = \beta = 1$, ρ (coeficiente de evaporación) = 0.01 y la matriz de feromonas fue inicializada con un valor de 0.1.

Se utilizaron 10 hormigas para la optimización y 1000 iteraciones. El proceso descrito en las secciones 2.2 y 2.3 se repitió 30 veces para evaluar si en efecto el modelo estaba encontrando una ruta más corta que la del agente que llegó al *target*.

3. Resultados

3.1. Forrajeo mediante caminatas aleatorias

A partir de la exploración realizada utilizando diferentes tipos de caminatas aleatorias (LF, URW y GRW) y con diferentes distancias del *home* al *target*, se observó que, para distancias cortas, la URW fue el tipo de caminata más eficiente, con un promedio de iteraciones de 20 para alcanzar el *target* (Figura 5a). Para distancias medias y largas, el LF resultó ser el tipo de caminata más eficiente, con un promedio de iteraciones para alcanzar el *target* de 153 y 378 respectivamente (Figura 5b, c).

En los tres escenarios, se pudo observar que la GRW fue la que tuvo un peor desempeño en cuanto a las iteraciones que le tomó llegar al *target*; de hecho, se encontró que para distancias largas este tipo de caminata no logró llegar al objetivo en el límite de iteraciones impuesto (1000 iteraciones; Figura 5c).

3.2. Modelo basado en agentes para forrajeo colectivo

En la Figura 6 se observa el ambiente en el que se desarrolló la simulación del MBA. Los cubos verdes representan los obstáculos añadidos al entorno para brindarle al sistema un poco más de realismo; el punto de partida (*home*) se encuentra en las coordenadas (0,0,0), es decir, en el centro del espacio: el punto rojo representa el objetivo que los agentes debían alcanzar (Figura 6a). Una vez realizada la exploración (que concluía cuando el primer agente llegaba al *target*), se optimizaba la ruta del *home* al *target* siguiendo los pasos descritos en la sección 2.3 y dicha ruta se visualizaba gráficamente como se muestra en la Figura 6b.

En la Figura 7 su observan las comunidades de agentes en un ambiente sin obstáculos y sin *target* para observar la dinámica colectiva. En ausencia de reglas de colectividad, se observa una comunidad dispersa que abarca un mayor volumen de exploración (Figura 7a); mientras que al aplicar las reglas de colectividad puede observarse la formación de un cumulo cuyo comportamiento asemeja a la forma en que algunos sistemas de animales llevan a cabo el forrajeo (Figura 7b). Se decidió mantener las reglas de colectividad debido a que le brindan realismo al modelo, ya que aparentan la forma en que este proceso de forrajeo es llevado a cabo en sistemas naturales.

A partir del modelo basado en agentes implementado en NetLogo, se observó que las comunidades de agentes creadas tuvieron un comportamiento colectivo similar independientemente del tipo de caminata aleatoria elegida. Sin embargo, esto no

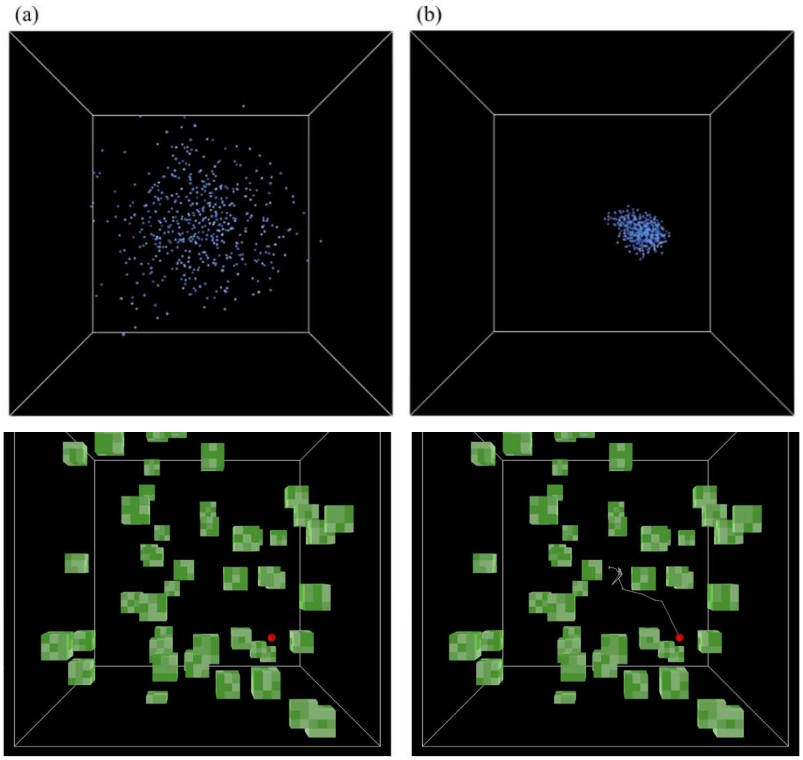


Fig
targ
ır el

Fig
de
glas

Fig. 6. (a) Ambiente en el que se desarrolló la simulación del MBA. (b) Representación de la trayectoria optimizada en el ambiente utilizando el ACO.

necesariamente implica que el tipo de caminata aleatoria implementada en los agentes no influya en el comportamiento a nivel individual y colectivo para llevar a cabo el forrajeo.

En el caso de que un agente se separe del resto, el tipo de caminata que este posee puede ser determinante en su eficacia de exploración del espacio.

Asimismo, se observó que, al implementar las reglas de colectividad desde el inicio, los agentes tendían a explorar el espacio en conjunto en lugar de dispersarse y abarcar más área de exploración.

Por ende, si estas reglas se mantenían durante toda la simulación, el grafo resultante hubiera abarcado menos espacio ya que todas las trayectorias hubieran estado concentradas, por lo que se decidió únicamente mantener la regla de separación durante todo el transcurso de la simulación y aplicar las reglas de cohesión y alineamiento una vez pasadas las primeras diez iteraciones. De esta forma se generó una mayor cantidad de cúmulos de agentes que aumentaron el volumen de exploración.

3.3. Optimización de trayectorias

A partir de las 30 repeticiones de la simulación y posterior optimización descritas en las secciones 2.2 y 2.3, se obtuvo la distancia recorrida por el primer agente que logró

Tabla 1. Resultados de las repeticiones de las rutas originales y optimizadas.

Ruta	Promedio de distancias	Desviación estándar	Error estándar
Original	62.24	25.11	4.58
Optimizada	48.75	20.17	3.68

alcanzar el *target* en cada simulación (en adelante "ruta original") y la distancia optimizada mediante el algoritmo ACO (en adelante "ruta optimizada"). El promedio de la distancia de las rutas optimizadas para las 30 repeticiones fue considerablemente menor al promedio de distancias de las rutas originales, tuvo una menor desviación estándar y un menor error estándar (Tabla 1).

Para comparar las distancias obtenidas de las rutas originales contra las de las rutas optimizadas, se realizó una prueba estadística de *t*. El resultado de la prueba indicó que las medias de las muestras difieren significativamente ($p < 0.01$). Estos resultados indican que las distancias optimizadas presentan una diferencia importante al compararlas contra las originales, y que se logró una disminución aproximada del 22 % en la distancia recorrida del *home* al *target* al optimizar la ruta utilizando el método planteado.

4. Discusión

Los resultados obtenidos de la exploración utilizando diferentes tipos de caminatas aleatorias permitieron identificar que el tipo de caminata más eficiente para la búsqueda de objetivos en un ambiente desconocido cuando estos se hallan en distancias cortas respecto al punto de partida es la URW. Mientras que los vuelos de Lévy representan el tipo de caminata más efectiva cuando las distancias entre el *home* y el *target* son más grandes. Esto puede deberse al hecho de que la caminata tipo LF está regida por una distribución de Pareto, la cual es una distribución anómala, es decir, una distribución libre de escala (sin escala característica).

Esto ocasiona que los pasos dados por los caminantes no tienen una longitud característica, permitiéndoles dar pasos muy variables en cuanto a su longitud. Al observar la dinámica seguida por los caminantes regidos por una caminata tipo LF (Figura 2a), puede notarse que la exploración realizada por los agentes consta de pasos regulares en un intervalo de tiempo y pasos de longitud muy larga que les permite salir completamente del área de exploración en la que se encontraban.

Este comportamiento permite a los agentes llegar a más zonas durante su exploración (pasos largos y súbitos) y en cada una de ellas realizar una búsqueda fina (pasos cortos y regulares). Este tipo de caminata permite equilibrar el área de exploración y la precisión de la búsqueda.

Precisamente por esta razón, para el MBA en tres dimensiones se decidió utilizar los vuelos de Lévy, pese a que el comportamiento colectivo de los agentes era similar independientemente del tipo de RW aplicado. Esto se hizo así ya que, si algún agente se llegaba a separar del grupo, este tipo de caminata resulta más eficiente para que pueda hacer una exploración del espacio efectiva [17].

Los resultados obtenidos de la optimización mostraron que la distancia recorrida del *home* al *target* disminuyó de manera significativa al optimizar la ruta con el método planteado. Este resultado es importante porque toma en cuenta las interacciones entre los agentes que pueden darse en una exploración, ya que la conexión de nodos cercanos entre sí al momento de generar el grafo permite tomar en cuenta la información recabada colectivamente, pero a nivel local, semejando comportamientos similares presentados en algunos entes en la naturaleza, como las bacterias que forman los *biofilms* o la comunicación en colonias de insectos sociales [20, 21].

5. Conclusiones y trabajo a futuro

El uso de modelos basados en agentes y de caminatas aleatorias, en especial aquellas con distribuciones libres de escala (como son los vuelos de Lévy), resultan una forma adecuada de describir la dinámica que presentan algunos grupos sociales de animales durante el forrajeo.

Esto debido a que toman en cuenta la interacción entre los componentes del sistema y con el ambiente, dando lugar a comportamientos emergentes que no pueden ser descritos a partir únicamente de conductas individuales. Asimismo, el uso del ACO permite hacer una optimización eficiente en cuanto a tiempo de cómputo, dado que los grafos obtenidos pueden ser muy grandes y representar un problema al utilizar otro tipo de algoritmos de ruta más corta.

En futuros trabajos, puede explorarse el comportamiento del sistema en ausencia de reglas de colectividad y evaluar si la reducción de la distancia al llevar a cabo la optimización se mantiene bajo este escenario. En caso de que esto suceda, sería interesante determinar qué parámetros o configuraciones del ambiente ocasionan que los sistemas naturales presenten este tipo de comportamientos colectivos. Queda por esclarecer la forma en que los agentes o individuos interactúan con el ambiente y como este influye en la manera en que estos se organizan.

Referencias

1. Boyer, D., Miramontes, O., Ramos-Fernández, G., Mateos, J. L., Cocho, G.: Modeling the searching behavior of social monkeys. Physica A: Statistical Mechanics and its Applications, vol. 342, no. 1-2, pp. 329–335 (2004) doi: 10.1016/j.physa.2004.04.091
2. Namatame, A.: Adaptation and evolution in collective systems. World Scientific (2005)
3. Gotwald, H.: Army ants the biology of social predators. Cornell University Press, Ithaca, NY (1995)
4. Reynolds, C.: Flocks, herds, and schools: A distribuited behavioral model. SIGGRAPH '87: In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, vol. 21, no. 4, pp. 25–34 (1987) doi: 10.1145/37401.37406
5. Xia, F., Liu, J., Nie, H., Fu, Y., Wan, L., Kong, X.: Random walks: A review of algorithms and applications. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 2, pp. 95–107 (2019) doi: 10.1109/TETCI.2019.2952908
6. Duan, H., Li, P.: Bio-inspired computation in unmanned aerial vehicles. Springer Berlin Heidelberg (2014)

7. Bonabeau, E.: Agent-based modeling: Methods and techniques for simulating human systems. In: Proceedings of The National Academy of Sciences, vol. 99, no. 3, pp. 7280–7287 (2002) doi: 10.1073/pnas.082080899
8. Russell, S., Norving, P.: Artificial intelligence a modern approach (2003)
9. Kennedy, J., Eberhart, R.: Swarm Intelligence. Elsevier (2001)
10. Dorigo, M., Di Caro, G.: Ant colony optimization: A new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation-CEC99, vol. 2, pp. 1470–1477 (1999)
11. Sellers, W. I., Hill, R., Logan, B.: Biorealistic simulating baboon foraging using agent-based modeling (2007)
12. Robinson, E. J., Ratnieks, F. L., Holcombe, M.: An agent-based model to investigate the roles of attractive and repellent pheromones in ant decision making during foraging. Journal of theoretical Biology, vol. 255, no. 2, pp. 250–258 (2008)
13. Rivière, J., Alaux, C., Le Conte, Y., Layec, Y., Lozac'h, A., Rodin, V., Singhoff, F.: Toward a complete agent-based model of a honeybee colony. Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection, PAAMS 2018, Communications in Computer and Information, Springer, Cham, vol. 887, pp. 493–505 (2018) doi: 10.1007/978-3-319-94779-2_42
14. Abbott, J. T., Austerweil, J. L., Griffiths, T. L.: Random walks on semantic networks can resemble optimal foraging. Neural Information Processing Systems Conference, vol. 122, no. 3, pp. 558 (2015)
15. James, A., Plank, M. J., Edwards, A. M.: Assessing Lévy walks as models of animal foraging. Journal of the Royal Society Interface, vol. 8, no. 62, pp. 1233–1247 (2011)
16. Fronhofer, E. A., Hovestadt, T., Poethke, H. J.: From random walks to informed movement. Oikos, vol. 122, no. 6, pp. 857–866 (2013)
17. Viswanathan, G. M., Raposo, E. P., Da Luz, M. G. E.: Lévy flights and superdiffusion in the context of biological encounters and random searches. Physics of Life Reviews, vol. 5, no. 3, pp. 133–150 (2008) doi: 10.1016/j.plrev.2008.03.002
18. Wilensky, U.: NetLogo flocking model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/ models/Flocking (1998)
19. Wilensky, U.: NetLogo. Center for connected learning and computer-based modeling. Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/ (1999)
20. Jacob, E. B., Becker, I., Shapira, Y., Levine, H.: Bacterial linguistic communication and social intelligence. TRENDS in Microbiology, vol. 12, no. 8, pp. 366–372 (2004)
21. Ben-Jacob, E.: Social behavior of bacteria: from physics to complex organization. The European Physical Journal B, vol. 65, no. 3, pp. 315–322 (2008)

ISSN 1870-4069

Aplicación móvil para el análisis de vocalizaciones del perro doméstico (canis familiaris)

Humberto Pérez-Espinosa[1,2], Yuvila M. Sanzón Rosas[1], Emanuel Aguilar Benitez[1]

[1] Centro de Investigación Científica y de Educación Superior de Ensenada, Unidad de Transferencia Tecnológica, México

[2] Consejo Nacional de Ciencia y Tecnología, México

hperez@cicese.mx

Resumen. El ladrido es una vocalización muy distintiva del perro. Es muy común y un medio de interacción entre los perros y con los humanos. Sin embargo, el alcance de su análisis automatizado por técnicas computacionales, así como las posibles aplicaciones a las que pueden dar lugar, han sido poco exploradas. En este artículo, describimos el proceso para desarrollar una aplicación capaz de estimar el nivel de alarma en el ladrido de los perros. Para el entrenamiento del sistema usamos una base de datos con más de 10,000 ladridos. Dichos ladridos fueron obtenidos aplicando estímulos positivos y negativos a los perros. Las muestras de ladridos se caracterizaron acústicamente utilizando una herramienta de procesamiento de señales que extrae grandes conjuntos de características. Generamos subconjuntos óptimos de descriptores acústicos para alimentar algoritmos de aprendizaje automático y obtener modelos de clasificación entrenados. Evaluamos dichos modelos y comparamos el rendimiento de clasificación de diferentes algoritmos. La clasificación obtenida supera los resultados previamente reportados en trabajos similares. A partir de estos modelos entrenados se diseñó e implementó una arquitectura que incluye un API y una aplicación móvil que puede ser usada con propósitos de seguridad doméstica y cuidado de las mascotas.

Palabras clave: Aprendizaje máquina, clasificación de ladridos, aplicación móvil.

Mobile Application for the Analysis of Domestic Dog (Canis Familiaris) Vocalizations

Abstract. Barking is a very distinctive vocalization of the dog. It is very common and a means of interaction between dogs and towards humans. However, the scope of its automated analysis by computational techniques, as well as the possible applications to which it can give rise, have been little explored.

In this article, we describe the process of developing an application capable of estimating the alarm level in dogs' barking. For system training, we use a database with more than 10,000 barks. These barks were obtained by applying positive and negative stimuli to the dogs. The bark samples were characterized acoustically using a signal processing tool that extracts large sets of characteristics. We generate optimal subsets of acoustic descriptors to feed machine learning algorithms and obtain trained classification models. We evaluate these models and compare the classification performance of different algorithms. The classification obtained exceeds the results previously reported in similar works. Based on these trained models, an architecture was designed and implemented that includes an API and a mobile application that can be used for home security and pet care purposes.

Keywords: Machine learning, dog bark classification, mobile application characterization.

1. Introducción

El perro ha sido compañero del hombre desde hace aproximadamente 15,000 años [5], inicialmente el propósito que tenían los perros era ayudar en la cacería, posteriormente una vez que los humanos se asentaron y comenzaron a practicar tanto la agricultura como la ganadería a los perros se les entrenó para cumplir el rol de guardianes de ganado.

En la actualidad, los perros siguen estando presentes en la sociedad principalmente como acompañantes, pero también se desempeñan como rescatistas, guardianes o asistentes entre otras importantes labores. En años recientes, se ha estudiado la manera interpretar sus ladridos para entenderlos, puesto que ellos son capaces de comunicarse con los humanos ya sea por medio de vocalizaciones o por medio de movimientos corporales.

Así mismo, los perros domésticos son capaces de entender a los humanos a través del lenguaje corporal [8]. Incluso, los perros son capaces de reaccionar a distintas situaciones y comunicarlas por medio de sus ladridos [14]. Investigaciones previas han tratado de identificar los tipos de ladridos ante contextos agradables y desagradables para el perro, otras se han enfocado en monitorear y reconocer el comportamiento del perro [7].

Con estos antecedentes el estudio aquí presentado se enfocó en usar los ladridos del perro para generar una alarma, es decir, identificar si los ladridos que emite el perro es por una situación agradable como estar jugando o si los ladridos advierten de una amenaza como estar ante la presencia de un extraño. Para esto fue necesario experimentar con distintas características que dieran mejores resultados tanto para identificar vocalizaciones del perro como para discriminar si un ladrido es amigable o de alarma.

2. Trabajos relacionados

La detección y clasificación de sonidos ambientales puede ser de gran utilidad en varios escenarios, como sistemas de vigilancia, asistencia, y monitorización. En el trabajo de [11] desarrollaron un sistema de clasificación de eventos acústicos para sistemas de vigilancia. En lugar de extraer características en marcos de tiempo corto, hacen una búsqueda sobre toda la duración del evento acústico. Se usó una base de datos de 8 clases que incluía gritos humanos, disparos y ladridos de perro.

Dentro de lo que se conoce como sistemas de asistencia existen varios trabajos en la literatura. La aplicación para Android Audition [1] fue desarrollada para asistir a personas con discapacidades auditivas reconociendo los sonidos a su alrededor. Los algoritmos utilizados de aprendizaje automático para su implementación fueron AdaBoostM1 que funciona como clasificador y REPTree como un alumno débil.

La validación del sistema se realiza en cuatro condiciones ambientales diferentes (ruido alto y bajo, fuente de sonido cercana y lejana) en este trabajo no se obtuvieron resultados satisfactorios, ya que la precisión fue 26,25 % para la detección en las cuatro condiciones. Otra aplicación con la misma orientación de asistencia a usuarios con discapacidad auditiva es HomeSound [4] que consta de un micrófono y una pantalla, y utiliza varios dispositivos de captura de audio instalados en los hogares.

El prototipo muestra los sonido acontecidos con historiales de formas de onda que representan el volumen y el tono. Una de las características del sistema es la clasificación automática de sonido. En la evaluación de desempeño se comprobó un aumento en la concientización sobre los sonidos en el hogar, pero las frecuentes clasificaciones automáticas erróneas no fueron bien recibidas por los usuarios.

Ambas aplicaciones para asistencia a usuarios con discapacidad auditiva incluyen la detección de ladridos como una característica esencial que les permite a los usuarios darse cuenta de eventos relevantes, por ejemplo si alguien está a la puerta o si el perro necesita algo. Otro trabajo que cae dentro del área de ambientes asistidos es el de [13], donde se investigó si el ladrido de un perro se puede distinguir del silencio u otros sonidos, como el ruido de un animal o del tráfico, o controlar un mecanismo de cierre de ventana en un escenario de hogar inteligente.

Estos trabajos se enfocan en detectar ladridos en el sonido ambiental. Sin embargo, existen otros trabajos que realizan un análisis más detallado de los ladridos tratando de clasificar diferentes tipos de ladrido. Recientemente se ha analizado el contexto y la emoción percibida en los ladridos explorando conjuntos de características acústicas buscando predecir la intensidad emocional de las secuencias de ladridos [3].

Los autores demuestran que el análisis acústico basado en aprendizaje automático puede alcanzar un rendimiento superior al nivel humano para clasificar el contexto en el cual fue producido un ladrido. Kim et al [6] propuso una forma de clasificar eventos acústicos de los perros eficienctizando los recursos sin degradación significativa de la precisión.

Para entrenar sus modelos usaron datos de intensidad de los sonidos mediante el uso de un sensor de ruido en lugar de datos de un sensor de sonido. En [9] por medio de los ladridos crean modelos para identificar la raza, el sexo y el contexto al que fue expuesto el perro.

Tabla 1. Número de muestras en los conjuntos de entrenamiento y prueba.

Clases	Entrenamiento	Prueba
Ladridos	5,645	6,957
Chillidos	911	389
Gruñidos	666	688
Sonidos	23,685	8,755
Total	**30,907**	**16,789**

Dicho estudio compara el desempeño de los algoritmos de aprendizaje supervisado: Naive Bayes, classification trees, k-nearest neighbors y logistic regression en conjunto con características extraídas usando la herramienta Praat teniendo una precisión del 55.5 % en clasificación de contextos. Por otra parte en [10] obtiene una precisión del 43 % usando el sistema de procesamiento de señales de audio EDS para obtener en conjunto de características acústicas.

Estos trabajos muestran el interés que existe en usar la identificación y clasificación de ladridos del perro doméstico para diversas aplicaciones. En este trabajo proponemos la creación de una aplicación orientada a la seguridad doméstica que tiene como principal característica el estimar el nivel de alarma que un perro transmite a través de sus ladridos.

3. Datos usados

Para entrenar los modelos de clasificación automática en los cuales se basa el método propuesto se usaron dos bases de datos de vocalizaciones del perro doméstico que generó CICESE-UT en colaboración con la empresa Efecto Mescalina. Se cuenta con grabaciones de 102 perros de diferentes razas. Para obtener los datos se aplicó un conjunto de estímulos con el objetivo de generar reacciones en las que se aprecien diferentes comportamientos como agresividad, miedo, alegría, entre otros.

3.1. Base de datos de perros Mescalina 2015

Para obtener los datos se pidió la autorización de los propietarios para grabar a sus mascotas y someterlos a una serie de estímulos de acuerdo a las condiciones explicadas a continuación:

1. Juego: El propietario estimula al perro utilizando los objetos o juguetes con los que normalmente juega, esperando lograr vocalizaciones.
2. Alerta normal ante presencia de un extraño: se comienza a tocar repetidamente el timbre del domicilio y dar golpes lo suficientemente fuertes a la puerta esperando provocar que el perro ladre. Esto se repite las veces que sea necesario.
3. Agresión: Una persona entra al domicilio, llamando la atención del perro y acercándose amenazadoramente a él, golpeando los pies contra el piso y dando palmadas para lograr que el perro se sienta agredido.

Tabla 2. Conjunto de audios para la detección de alarma.

Clases	Número de muestras
Tranquilo	12
Activo	9
Alerta	58
Alarma	33
Total	**112**

4. Agresión al dueño dentro del domicilio: se solicita al propietario que se acerque y se simula un ataque, haciendo también ruidos como palmadas y golpes al piso. Para lograr una mejor reacción de parte del perro, de ser necesario, se le pide al dueño que grite pidiendo auxilio o el nombre del perro.

En esta base de datos se tienen 6,077 ladridos segmentados a nivel individual con un promedio de duración 1 segundo. La base de datos Mescalina 2015 tiene treinta y siete individuos con sus nombres correspondientes. Los ladridos corresponden a las grabaciones de 11 razas de perros comunes en los hogares. En esta base de datos las edades de perros se etiquetaron en meses. Desde los 6 meses hasta 72.

3.2. Base de datos de perros Mescalina 2017

La base de Mescalina 2017 también la generó CICESE-UT[3] con el fin agregar más muestras al corpus original. Se añadieron nuevos contextos de grabación, los cuales se describen a continuación:

1. Llegada a casa: se solicita al propietario que salga del domicilio, se aleja con el experimentador, posteriormente regresan, y desde afuera de la casa le habla cariñosamente a su perro, pero sin abrir la puerta.
2. Simulación de paseo: el propietario realiza la rutina normal que precede al paseo. De ser necesaria una mejor reacción, se estimula al perro con llamadas cálidas o simulando abrir la puerta hasta que se emocione y produzca vocalizaciones.
3. Tristeza/ansiedad por separación: el propietario ata al perro con la correa a un árbol y se aleja de su vista. Para lograr un mejor resultado se le pide al dueño que se despida de él.
4. Asustado en el parque: mientras el perro continúa atado al árbol, el experimentador o algún otro extraño se acerca amenazadoramente al perro.

En esta base de datos se tienen 6,948 ladridos segmentados a nivel individual con un promedio de duración 1 segundo. La base de datos Mescalina 2017 tiene sesenta y cinco individuos. Los ladridos corresponden a las grabaciones de cuatro razas de perros comunes en los hogares mexicanos Chihuahua, French Poodle, Schnauzer y mestizos. Para este estudio se utilizaron los audios segmentados a nivel de vocalización individual. El formato del audio es WAV PCM monocanal con una frecuencia de muestreo a 16 kHz.

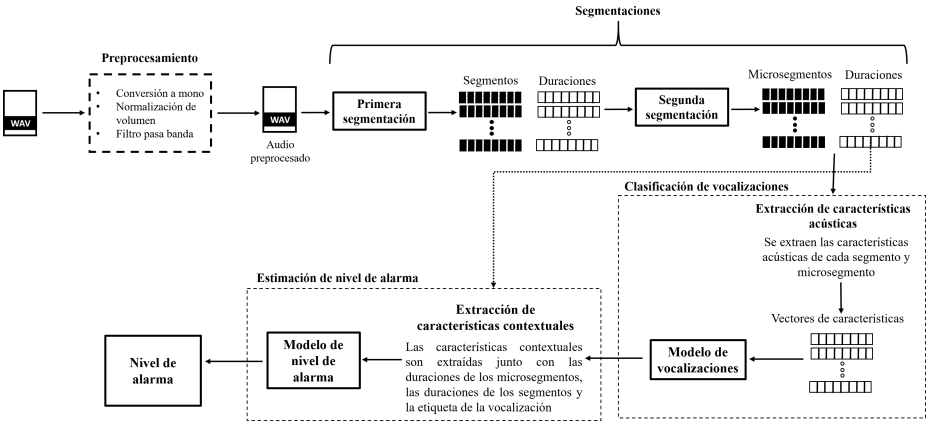


Fig. 1. Esquema de detección de un nivel de alarma.

De acuerdo al teorema de muestreo de Nyquist-Shannon, si se utilizan 16 kHz como frecuencia de muestreo, la frecuencia máxima de la señal sería de 8,000 Hz. De acuerdo al trabajo de [2] se han registrado vocalizaciones de perro que llegan a alcanzar 5,000 Hz, por lo que la frecuencia de muestreo es suficiente.

4. Método para la estimación de nivel de alarma

El método propuesto para determinar el nivel de alarma a partir de ladridos, incluye dos modelos de clasificación que son aplicados uno después del otro. El primero, identifica el tipo de vocalización y el segundo modelo que estima el nivel de alarma en las vocalizaciones. Para entrenar cada uno de estos modelos se prepararon dos conjuntos de audios. El primero, al cual llamamos conjunto para clasificación de vocalizaciones, esta enfocado en diferenciar los diferentes tipos de vocalizaciones que emiten los perros utilizando las siguientes categorías: ladridos, chillidos, y gruñidos.

Además, se incluyeron en el entrenamiento todos los sonidos del ambiente capturados durante la recolección de los datos. A esta categoría se le denominó: sonido. Los segmentos de audio tienen una duración de 0.3 a 5 segundos. Este conjunto de audios se dividió en entrenamiento y prueba, ver Tabla 1.

El segundo conjunto de audios, al cual llamamos conjunto para detección de alarma, fue formado por 112 audios con una duración de 15 segundos y etiquetados con los posibles estados de nivel de alarma: tranquilo, activo, alerta y alarma, ver Tabla 2.

A diferencia del conjunto para clasificación de vocalizaciones que contiene segmentos de vocalizaciones individuales, el conjunto para detección de alarma, son una sucesión de vocalizaciones representativas de cada uno de los niveles de alarma de interés. Estos 112 audios fueron seleccionados de forma manual por las mismas personas que aplicaron los estímulos descritos en la sección anterior.

Enseguida se describe el flujo, tratamiento de los datos y construcción de modelos para encontrar el nivel de alarma de un audio (Figura 1). Este proceso consta de cuatro fases: fase de pre-procesamiento, segmentación, extracción de características acústicas, clasificación de vocalizaciones y predicción del nivel de alarma.

Tabla 3. Características acústicas.

Características	**No. características**
MFCC	12
Delta's MFCC	12
LPCC	12
Delta's LPCC	12
Zero Crossing Rate	1
Log Energy	1
Total características	**50**

4.1. Fase de pre-procesamiento

El audio de entrada debe estar en formato WAV, tener una duración de 15 segundos a una tasa de muestreo de 16kHz. A este audio se le aplican los siguientes procedimientos:

1. Convertir el audio a un solo canal.
2. Aplicar filtro para la normalización del volumen.
3. Aplicar filtro pasa banda de 150Hz a 3500Hz.

Dado que el ancho de banda del ladrido es menor a 3,000Hz se conservan todas las frecuencias menores a este valor. En el trabajo de [12] se observa que los ladridos de distintos perros se encuentran en el intervalo de frecuencia de 1,000 Hz a 2,000 Hz en su mayoría, aunque otros tipos de vocalizaciones del perro pueden alcanzar frecuencias por arriba de ese rango.

4.2. Fase de segmentación

El audio resultante de la fase de preprocesamiento, es segmentado por primera vez con el fin de encontrar grupos de vocalizaciones, es decir periodos en donde se den varias vocalizaciones continuas sin una pausa perceptible entre ellas. El resultado de esta segmentación son tuplas de vectores, donde:

– El primer elemento es un vector que contiene señal de cada segmento del audio.
– El segundo elemento es un vector que contiene los tiempos de inicio y fin de cada segmento.

Enseguida, a cada uno de los segmentos del audio se le aplica una segunda segmentación con el objetivo de obtener micro segmentos, que representan vocalizaciones individuales. Es decir, un micro segmento contiene un sólo ladrido, chillido o gruñido.

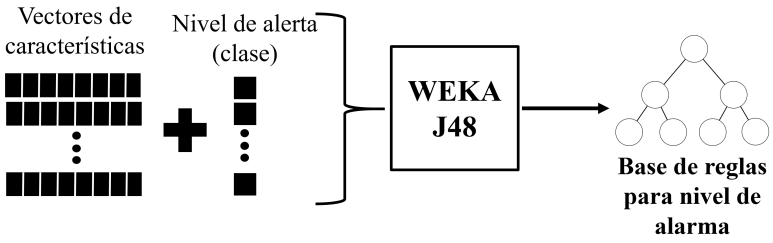


Fig. 2. Proceso de generación de reglas para un nivel de alarma.

4.3. Extracción de características acústicas

Después de aplicar los procedimientos de la fase de segmentación se caracteriza el audio. Como se mencionó anteriormente, la tarea de determinar un nivel de alarma se basa en dos etapas primordiales, determinar el tipo de vocalización y estimar el nivel de alarma. Para ello, fue necesario obtener características diferentes para cada una de las dos etapas. En el caso del conjunto para clasificación de vocalizaciones la información se representó con las características acústicas de bajo nivel mencionadas en la Tabla 3.

En el caso del conjunto para detección de alarma, se determinó que las características contextuales o también llamadas de alto nivel obtenidas debían estar relacionadas con el contexto de la grabación, por lo cual se consideran características basadas en el número de vocalizaciones presentes en un periodo y en la manera en que se agrupan. A continuación se listan las nueve características de este tipo que fueron usadas:

– Número de ladridos.
– Número de chillidos.
– Número de gruñidos.
– Número de grupos de ladridos.
– Duración de los ladridos.
– Duración de silencio.
– Promedio log energy de ladridos.
– Promedio de decibelios en ladridos.
– Promedio del pitch de ladridos.

4.4. Clasificación de vocalizaciones

El modelo para detección de vocalizaciones fue construido y evaluado empleando los conjuntos de datos que se describen en la Tabla 1. Ambos conjuntos de datos fueron caracterizados de acuerdo a la tabla 3 y se evaluaron los siguientes cinco algoritmos de aprendizaje automático de WEKA:

– Decision three algorithm - J48.
– Function algorithm - SVM (polynomial kernel).

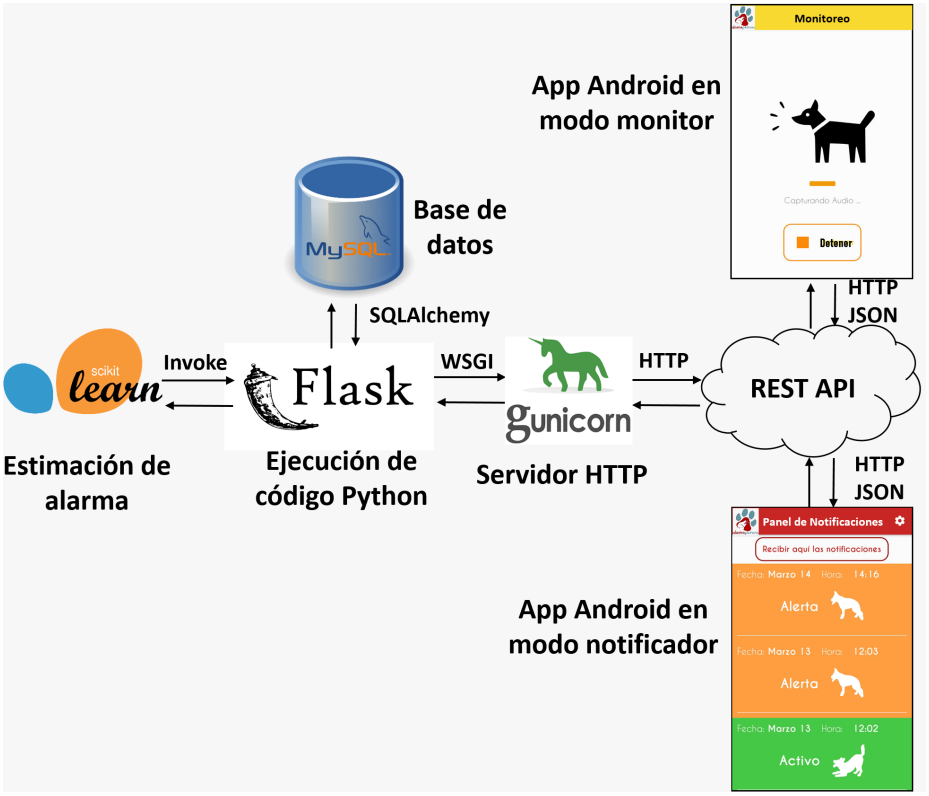


Fig. 3. Arquitectura del sistema.

– Bayesian algorithm - Naive Bayes.
– Ensemble algorithm - Random Forest.
– Ensemble algorithm - Bagging (REPTree).

Además se emplearon los siguientes cuatro algoritmos de la biblioteca de aprendizaje automático para python Scikit-Learn:

– Random Forest.
– KNN.
– Gradient Boosting.
– Extratrees.

En el proceso de detección de alarma, una vez obtenidos los micro segmentos, se extraen las características acústicas de cada uno, de manera que ahora se tienen los vectores de características acústicas. Estos son clasificados por el modelo de vocalizaciones para predecir si un segmento es un ladrido, un chillido, un gruñido o un sonido ambiental.

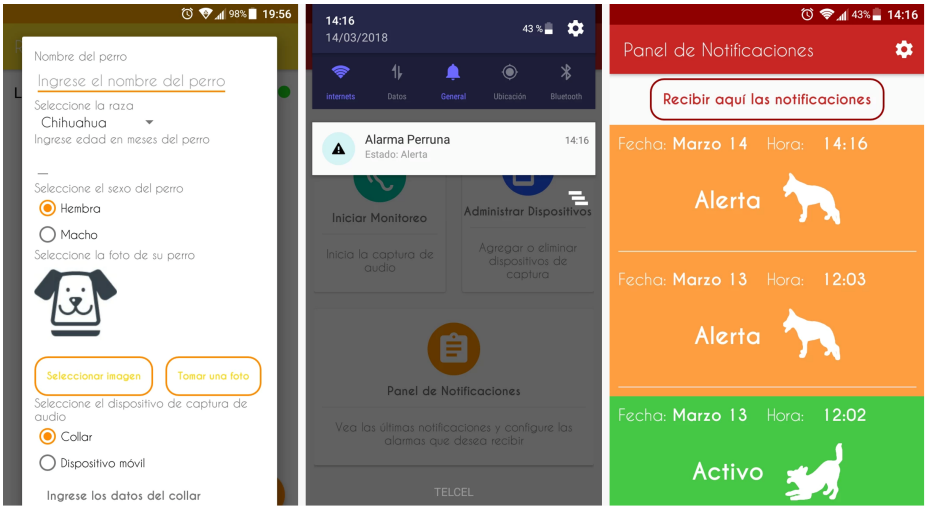


Fig. 4. Capturas de pantalla de la aplicación Alarma Perruna.

4.5. Determinación del nivel de alarma

Terminada la fase de clasificación de vocalizaciones, se realiza la extracción de características contextuales. Enseguida se describe el proceso de generación de estas reglas para el cálculo de nivel de alarma. Dicho proceso se basa en obtener las características contextuales sobre el conjunto de audios para detección de alarma. A cada vector de características se le agregó el nivel de alarma correspondiente de cada audio.

Las características contextuales extraídas junto con el nivel de alarma asignado, fueron alimentados al algoritmo J48 de WEKA, para que de esta manera se obtuviese el modelo de clasificación a utilizar en la determinación del nivel de alarma correspondiente a un audio de 15 segundos (ver figura 2). Una vez que se tienen estas reglas, son aplicadas sobre las características resultantes en el paso anterior, de esta manera se obtiene el nivel de alarma correspondiente al audio de entrada.

5. Arquitectura del sistema

El sistema se compone de dos elementos, el primero es una Interfaz de Programación de Aplicaciones (API) bajo el paradigma de software basado en red conocido como Transferencia de Estado Representacional (REST). El segundo es una aplicación móvil (Android) que puede ejercer el rol de monitor para la captura y envío de los audios del perro, o el rol de notificador para la recepción y visualización de notificaciones por parte del API REST. Cuando la aplicación funciona en modo de monitor, se encarga de la obtención de muestras acústicas, las cuales son enviadas al API, para su procesamiento.

Tabla 4. Clasificación de vocalizaciones.

	Clasificador	**F-measure**
Weka	J48	0.762
	SVM	0.827
	Naive Bayes	0.614
	Random Forest	0.866
	Bagging	0.840
Scikit-Learn	KNN	0.833
	Random Forest	0.886
	Gradient Boosting	0.815
	ExtraTree	0.854

Cuando la aplicación móvil funciona en modo de notificador, recibe y muestra información al usuario sobre el estado actual del perro. Además, mediante la aplicación se hace el registro del usuario y el registro de otros dispositivos de captura de audio, como podrían ser collares o pecheras desarrollados por terceros.

La REST API, lleva a cabo el almacenamiento de datos de los usuarios en la base de datos, el procesamiento de audio, la clasificación del audio de entrada, la identificación de un nivel de alarma y el envío de la notificación al dispositivo notificador. En la Figura 3 se muestra la arquitectura que hace uso de un conjunto de herramientas (MySQL, Gunicorn, Flask) que permiten un control flexible sobre cada componente del front y back end.

El servidor que realiza el procesamiento del audio interactúa con las aplicaciones mediante una API basada en Python que se encarga de ejecutar los algoritmos y métodos necesarios para segmentar, caracterizar y clasificar el audio. Una vez que el servidor recibe un archivo de audio de la aplicación monitor, se activa el procesamiento, y se envía un mensaje asincrónico a la aplicación notificadora.

6. Aplicación alarma perruna

Como se mencionó anteriormente cuando la aplicación funciona en modo de monitor se encarga de la obtención de muestras acústicas, las cuales son enviadas por medio del API, para su tratamiento. Las principales funcionalidades de la aplicación son:

1. Registro: La primera vez que se usa la aplicación se debe registrar el usuario con un correo electrónico y contraseña. Una vez registrado el usuario, inicia sesión y deberá acceder a la opción Registro de perros en la pantalla principal para registrar a uno o más perros. Por cada perro se debe especificar su nombre, raza, sexo, y fotografía (ver Figura 4 (a)). Además, por cada perro registrado se debe dar de alta el dispositivo que capturará sus vocalizaciones, que puede ser un dispositivo móvil en el cual se ejecute la aplicación en modo monitor o un dispositivo vestible, como un collar o una pechera. Mediante la REST API se lleva a cabo el almacenamiento de la información de registro en la base de datos.

Tabla 5. Matriz de confusión clasificación de nivel de alarma.

	Tranquilo	**Activo**	**Alerta**	**Alarma**
Tranquilo	12	0	0	0
Activo	1	7	1	0
Alerta	0	1	48	9
Alarma	0	0	14	19

2. Monitoreo: para iniciar el monitoreo del sonido ambiental, en la pantalla principal de la aplicación se pulsa el botón Iniciar monitoreo. En ese momento la aplicación empieza a captura el sonido ambiental, y lo envía al sistema de procesamiento cada 15 segundos.

   Durante el monitoreo la aplicación muestra en pantalla un mensaje que dice capturando audio, una barra de progreso que se reinicia cada 15 segundos y un botón con la etiqueta Detener (esta pantalla se muestra en la parte superior de la Figura 3). El dispositivo deberá colocarse en un lugar de la casa donde se espere que el perro se mantenga cerca y las vocalizaciones sean capturadas con suficiente intensidad.

3. Notificaciones: En este apartado de la aplicación se reciben las notificaciones (ver Figura 4 (c)). A esta pantalla se accede pulsando la opción Panel de notificaciones en la pantalla principal. Cuando la aplicación está en modo notificador puede estar trabajando en segundo plano y notificar al usuario como se muestra en la Figura 4 (b).

   En este apartado la aplicación cuenta con opciones de configuración para indicar que niveles de alarma quiere recibir el usuario. En dicha pantalla el usuario marca los estados del perro que desea que se le notifiquen. Por ejemplo, únicamente el nivel de alarma máximo.

7. Resultados

El sistema fue evaluado en las dos etapas de clasificación de las que consta. En la Tabla 4 se muestra el desempeño en la primera etapa, es decir en la tarea de clasificación de vocalizaciones y sonidos ambientales correspondiente a cada algoritmo de aprendizaje y de acuerdo al valor F-measure obtenido.

Como se puede observar el mejor desempeño se obtuvo con el algoritmo Random Forest de Scikit-Learn. Para la evaluación de la segunda etapa, debido a que el tamaño del corpus es reducido, sólo 112 muestras, se optó por evaluar el modelo realizando 10-fold cross-validation. Usando este esquema de validación se obtuvo un F-measure de 0.764.

A continuación se muestra la matriz de confusión resultante de aplicar este proceso sobre los datos descritos en la Tabla 5. En esta matriz de confusión se puede apreciar que existe una mayor confusión entre las clases alerta y alarma.

8. Conclusiones y trabajo futuro

En este trabajo se desarrolló un sistema para la estimación del nivel de alarma transmitido en los ladridos del perro doméstico. El sistema está compuesto por una aplicación móvil y un sistema de procesamiento, y clasificación de la señal de audio que se comunican por medio de un API. Usando descriptores acústicos de bajo nivel como MFCCs, LPCC, Zero Crossing Rate y Log Energy, que son usados para entrenar un modelo de clasificación de tipos de vocalizaciones con el algoritmo de aprendizaje máquina Random Forest, se logró un F-measure de 0.886. Lo cual puede considerarse como un muy buen resultado para este tipo de tareas de clasificación.

A partir de la clasificación del tipo de vocalización y de otros descriptores de alto nivel en segmentos de audio de 15 segundos se entrenó un árbol de decisión que determina el nivel de alarma detectado en las vocalizaciones para ser notificado al usuario. Esta estimación mostró un F-measure de 0.764 y un mayor nivel de confusión en las clases de alerta y alarma, lo cual podría no ser tan grave en la práctica, ya que de acuerdo a la orientación de la aplicación, lo importante es identificar cuando el perro puede haber detectado algo inusual en su entorno.

Como trabajo futuro es necesario realizar una evaluación de la aplicación en un ambiente real, ya que en la evaluación reportada se usaron sólo muestras de la base de datos previamente colectada, pero no se usaron muestras obtenidas a través de la aplicación y enviadas por medio del API.

La aplicación Alarma Perruna muestra un desempeño en la clasificación de vocalizaciones caninas suficiente para ser usada de manera eficaz por dueños de perros que dejan a sus mascotas en casa y desean saber que sucede mientras están fuera. Mediante el uso de esta aplicación los dueños de los perros pueden saber si existe algo inusual en su hogar, por ejemplo si un extraño intenta entrar, o el perro tiene algún problema y tomar alguna medida al respecto.

Agradecimientos. Este trabajo de investigación se ha llevado a cabo en el contexto del programa “Cátedras CONACyT ”financiado por el Consejo Nacional de Ciencia y Tecnología (CONACyT).

Referencias

1. Ayu Melati, A. I. S., Karyono, K. K.: Audio detection (audition): Android based sound detection application for hearing-impaired using AdaBoostM1 classifier with REPTree weaklearner. In: Proceedings of the Asia-Pacific Conference on Computer Aided System Engineering, pp. 136–140 (2014) doi: 10.1109/apcase.2014.6924487
2. Frommolt, K. H., Gebler, A.: Directionality of dog vocalizations. The Journal of the Acoustical Society of America, no. 1, pp. 561–565 (2004) doi: 10.1121/1.1758971
3. Hantke, S., Cummins, N., Schuller, B.: What is my dog trying to tell me? the automatic recognition of the context and perceived emotion of dog barks. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5134–5138 (2018) doi: 10.1109/icassp.2018.8461757

4. Jain, D., Mack, K., Amrous, A., Wright, M., Goodman, S., Findlater, L., Froehlich, J. E.: Homesound: An iterative field deployment of an in-home sound awareness system for deaf or hard of hearing users. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12 (2020) doi: 10.1145/3313831.3376758
5. Kevin, S. J.: The Domestication, Behaviour and Use of the Dog. Springer Netherlands (2006) doi: 10.1007/978-1-4020-4362-8_1
6. Kim, Y., Sa, J., Chung, Y., Park, D., Lee, S.: Resource-efficient pet dog sound events classification using lstm-fcn based on time-series data. Sensors, vol. 18, no. 11, pp. 4019 (2018) doi: 10.3390/s18114019
7. Ladha, C., Hammerla, N., Hughes, E., Olivier, P., Ploetz, T.: Dog's life: Wearable activity recognition for dogs. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, pp. 415–418 (2013) doi: 10.1145/2493432.2493519
8. Lakatos, G., Gácsi, M., Topál, J., Miklósi, Á.: Comprehension and utilisation of pointing gestures and gazing in dog–human communication in relatively complex situations. Animal Cognition, vol. 15, no. 2, pp. 201–213 (2012) doi: 10.1007/s10071-011-0446-x
9. Larranaga, A., Bielza, C., Pongrácz, P., Faragó, T., Bálint, A., Larranaga, P.: Comparing supervised learning methods for classifying sex, age, context and individual mudi dogs from barking. Animal cognition, vol. 18, no. 2, pp. 405–421 (2015) doi: 10.1007/s10071-014-0811-7
10. Molnár, C., Kaplan, F., Roy, P., Pachet, F., Pongrácz, P., Dóka, A., Miklósi, Á.: Classification of dog barks: A machine learning approach. Animal Cognition, vol. 11, no. 3, pp. 389–400 (2008) doi: 10.1007/s10071-007-0129-9
11. Nguyen, Q., Choi, J.: Matching pursuit based robust acoustic event classification for surveillance systems. Computers & Electrical Engineering, vol. 57, pp. 43–54 (2017) doi: 10.1016/j.compeleceng.2016.11.007
12. Sakamoto, S.: Sound attenuation devices for dogs barking (estimation of aperture ratio and experimental study of silencer). International Journal of Mechanical Engineering and Applications, vol. 2, no. 1, pp. 18–24 (2014), doi: 10.11648/j.ijmea.20140201.14
13. Schneider, M., Jokisch, O.: Towards a robust analysis and classification of dog barking. Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung, pp. 117–124 (2020)
14. Yin, S.: A new perspective on barking in dogs (cannis familiaris). Journal of Comparative Psycology, vol. 16, pp. 189–193 (2002)

ISSN 1870-4069

Separador industrial PET mediante técnicas de visión computacional

Irving Misael Caballero Aquino, Axel Adrián Casarrubias Calvillo, Rubén Daniel Vera León

Universidad Politécnica del Valle de México, México

{irving.caballero.aquino, axel.casarrubias.calvillo, ruben.vera.leon}@upvm.edu.mx

Resumen. En el ámbito del sector industrial existen amplias aplicaciones a la recolección y clasificación de plásticos, así como el PET, aunque, la principal razón por la que este suele ser recolectado es para reciclarlo. El PET usualmente se segmenta en dos categorías: PET cristalino y PET de color, dicha segmentación por lo regular consiste en una técnica en donde un grupo de personas están al pendiente del lugar donde se junta el PET y lo va separando por categoría (cristal y de color) para así llevar a cabo el proceso de reciclaje. Nosotros logramos desarrollar una alternativa para la clasificación de PET con ayuda de inteligencia artificial logrando detectar los tipos o bien la clase a la cual corresponde. Se pretende implementar ésta técnica de clasificado conectando una cámara que será la extensión del clasificador para la aplicación de inteligencia artificial y con ello, poder identificar a alta velocidad el tipo de PET y definir a que clase pertenece, reduciendo tiempo y esfuerzo en el proceso de clasificación de PET y satisfaciendo las necesidades de reciclaje, cuidado del medio ambiente, reducción de basura en las calles, así como el cuidado del calentamiento global, las cuales son problemáticas actuales que necesitan de soluciones.[1]. Ya que la revolución e innovación de la tecnología cada vez es más eficiente, este proyecto es competente para participar y aplicarlo en el sector de INDUSTRIA 4.0 [2, 3]. La norma ISO 14001 [4] en algunos países es exigente ante todas las industrias cuando se trata de ecología y medio ambiente, por lo que nos dimos la tarea de implementar procesos más rápidos y eficientes ya que es una de las operaciones o procesos más comunes. Es por esto que nuestro clasificador de PET es una de las mejores herramientas gracias a la implementación de inteligencia artificial [5].

Palabras clave: PET, innovación, industria 4.0, medio ambiente, ISO, calentamiento global, reciclar

Pet Industrial Separator Using Computer Vision Techniques

Abstract. In the field of the industrial sector there are extensive applications of collecting and classifying plastics, as well as PET, albeit the mean reason is for recycling. PET is usually segmented in two categories: crystalline and colored pet this consists of a technique where a group of people are on the slope of the

place where the pet meets and separates it by class (glass) and (colored) in order to carry out the process of recycling. We managed to develop an alternative for PET classification with the help of artificial intelligence managing to detect the types or the class with which they are corresponding to. It is intended to use this classifier connected to a camera which is an extension of the classifier for the artificial intelligence application, as well as identify at high speed the type of PET and also define which class the PET belongs to, in order to reduce time end effort on the PET classifying process, satisfying some necessities like recycling, taking care of the environment, and excess rubbish on the streets and also with the care of global warming which are nowadays problems that need answers. Since the revolution and innovation of technology is becoming more efficient this project is competent to participate and apply in the sector of INDUSTRIA 4.0 ISO 14001 standard in some countries is demanding in all industries when it co mes to ecology and environment, consequently, we were tasked with implementing faster and more efficient processes as it is one of the most common operation or processes, worldwide. That's why our PET classifier is one of the best tools thanks to the implementation of artificial intelligence.

Keywords: PET, innovation, industry 4.0, environment, ISO, global warming, recycle.

1. Introducción

Hoy en día la aplicación y resecación de plásticos y PET es muy común ya que tiene una gran variedad de potencial a desempeñar. El plástico se ha hecho tan omnipresente que resulta difícil creer que apenas llevamos produciéndolo a escala industrial desde hace solo unas décadas.

Su historia es más reciente de lo que se podría pensar. Los primeros plásticos totalmente sintéticos empezaron a fabricarse a comienzos del siglo XX: la baquelita, ideada en 1907, fue el primero de ellos, pero no fue hasta después de la II Guerra Mundial cuando arrancó la fabricación y utilización masiva de todo tipo de materiales de ese tipo [6].

El tereftalato de polietileno (PET) es un polímero plástico que se obtiene mediante un proceso de polimerización de ácido tereftálico y mono etilenglicol. Es un polímero lineal, con un alto grado de cristalinidad y termoplástico en su comportamiento, lo cual lo hace apto para ser transformado mediante procesos de extrusión, inyección, inyección-soplado y termoformado. Sus principales características son:

- Es totalmente reciclable.
- Cristalinidad y transparencia, aunque admite cargas de colorantes.
- Buen comportamiento frente a esfuerzos permanentes.
- Alta resistencia al desgaste.
- Muy buen coeficiente de deslizamiento.
- Buena resistencia química.
- Excelente barrera a CO2, aceptable barrera a O2 y humedad.
- Compatible con otros materiales barrera que mejoran en su conjunto la calidad de barrera de los envases y por lo tanto permiten su uso en marcados específicos.

– Aprobado para su uso en productos que deban estar en contacto con productos alimentarios.

Las propiedades físicas del PET y su capacidad para cumplir diversas especificaciones técnicas han sido las razones por las que el material haya alcanzado un desarrollo relevante en la producción de fibras textiles y en la producción de una gran diversidad de envases, especialmente en la producción de botellas, bandejas, flejes y láminas [7].

El reciclaje es una de las alternativas utilizadas en la reducción del volumen de los residuos sólidos. Se trata de un proceso que consiste básicamente en volver a utilizar materiales que fueron desechados y que aún son aptos para elaborar otros productos o volver a fabricar los mismos.

El PET es uno de los subproductos que más se recupera debido a que tiene un alto precio de venta, aproximadamente $6.80 por kg, el cual depende del mercado internacional, además de que se ha establecido un amplio canal de captación para su retorno a la industria.

El PET puede reciclarse múltiples veces, pero para uso alimentario solo se permite un primer nivel, pasado éste se utiliza para una amplia variedad de productos finales como: fibra, fibra de relleno textil, correas, y botellas y envases para usos no alimentarios como detergentes y productos fitosanitarios [1].

El PET se obtiene a partir de materiales de embalaje, como envases de comida y botellas de refrescos o de agua mineral. Para alcanzar un alto grado de calidad e higiene, los procesadores necesitan una corriente de pelets prácticamente pura. Esto se logra retirando productos de colores no deseados, aluminio, otros metales y muchas más sustancias extrañas [8].

2. Trabajos relacionados

Hoy en día, existen máquinas que se encargan de clasificar no solo PET, sino que también otros productos, aunque por lo regular, estas máquinas son enfocadas para ver si los productos cumplen con un color estándar o un aspecto en específico y estás máquina suelen ser robustas debido al gran trabajo, así como con cierto costo elevado a cubrir para empresas pequeñas.

Las clasificadoras ópticas de la serie Sortex son prueba de ello pues realizan las funciones descritas, son equipadas con los desarrollos tecnológicos más innovadores, esta serie detecta impurezas, partículas extrañas y colores no deseados, y los descarta de forma eficaz [8, 9].

De acuerdo con el avance tecnológico y la necesidad de implementar nuevos métodos para ser competentes en la INDUSTRA 4.0, se ha impulsado a que se utilice Deep Learning para obtener una mayor identificación y exactitud en nuestros procesos, esto queda evidenciado en [11], donde se propone la integración de aprendizaje profundo o Deep Learning en los procesos de reciclaje desarrollando un prototipo de sistema automatizado que integra visión artificial para la selección de empaques de plástico, vidrio y lata para la selección de material reciclable, debido al posible incremento de residuos para los próximos años en Colombia, lo que sugiere que la falta de un sistema eficiente de reciclaje propicia al desecho de elementos potencialmente

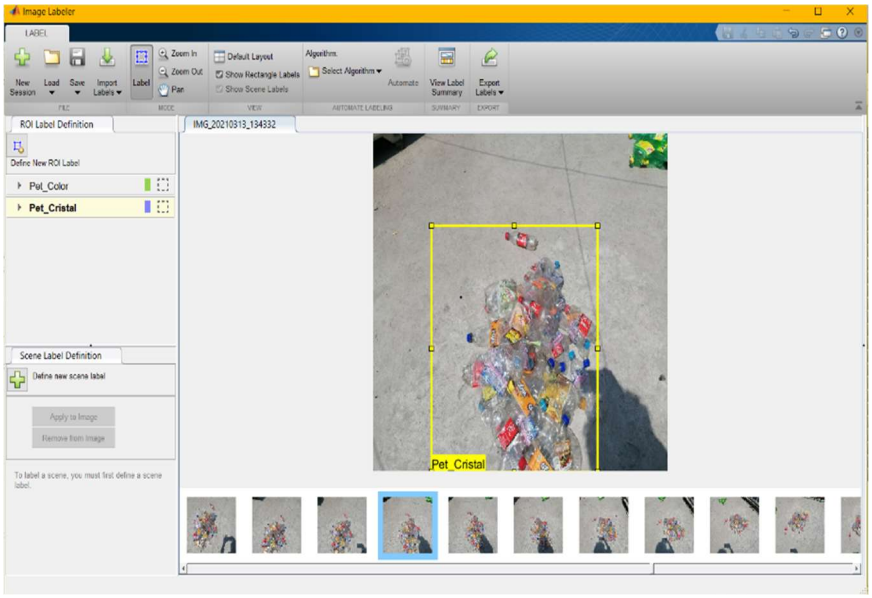


Fig. 1. Sesión de etiquetado en Matlab.

provechosos en beneficio de la industria y el medio ambiente. Ya que, con el proceso desarrollado manualmente, no se optimizan los recursos como tiempo y cantidad de materiales clasificados, por lo cual el porcentaje de residuos reciclados es bajo en este país.

Producto de esta práctica improvisada en los vertederos, se presentan consecuencias como daños ambientales, problemas de salubridad a la población, así como desaprovechamiento de recursos y materias primas. Esto indica que, debido al tradicional proceso de reciclaje, se está perdiendo materia prima, que eventualmente, puede reintegrarse al ciclo de vida útil, esto por la falta de implementación de un sistema de control que facilite el proceso de clasificación [11].

Una vez planteada la problemática y diseñado tanto el entrenamiento como la implementación del hardware [11], concluye que en el aplicativo se necesita diseñar una interfaz que permita a usuarios que no cuenten con el hardware probar la capacidad de clasificación de la red neuronal convolucional, con la finalidad de despertar la curiosidad por el aprendizaje profundo y sus aplicaciones.

El diseño de un hardware a escala expuso resultados positivos, lo que indica que este prototipo se puede adaptar a procesos de clasificación de residuos para contribuir con el cuidado del medio ambiente en Colombia implementando Deep Learning [11].

En [12] se muestra información importante con aspectos relacionados al desarrollo del proyecto, pues [12] emplea técnicas de Deep Learning y You Only Look Once V2 (YOLO V2) mediante MATLAB [13], si bien la solución no se relaciona en cuanto al eje temático abordado, este es un trabajo importante, ya que con estas técnicas se desarrolla el proyecto, lo cual nos permite tener una referencia de resultados favorables [12].

Tabla 1. Imágenes ocupadas para entrenar Clasificador de PET.

Clases	Número de Imágenes
PET Cristal	1000
PET Color	1000



Fig. 2. Arquitectura ResNet-18 con YOLO V2.

3. Materiales y métodos propuestos

Las redes neuronales convolucionales (CNN o ConvNets) son herramientas esenciales para el aprendizaje profundo y son especialmente útiles para tareas de clasificación de imágenes, detección de objetos y reconocimiento. Las CNN se implementan como una serie de capas interconectadas [18].

Para poder realizar la clasificación, se implementó un detector You Only Look Once (YOLO) v2 [10]. Como red neuronal base, se empleó ResNet-18 [14], la cual es una red neuronal convolucional que tiene 18 capas de profundidad, puede cargar una versión preentrenada de la red entrenada en más de un millón de imágenes desde la base de datos ImageNet [15]. La red preentrenada puede clasificar imágenes en 1000

Entrenar a YOLO V2 object detector

```
opts=trainingOptions('adam',"MiniBatchSize",32,...
    "InitialLearnRate",0.0001,...
    "MaxEpochs",20,"LearnRateSchedule","piecewise",...
    'LearnRateDropPeriod',75,...
    "Shuffle","every-epoch",...
    "L2Regularization",0.0001,"Verbose",true,"VerboseFrequency",50,...
    'DispatchInBackground', true,...
    'ExecutionEnvironment','auto')
```

Fig. 3 Configuración de entrenamiento.

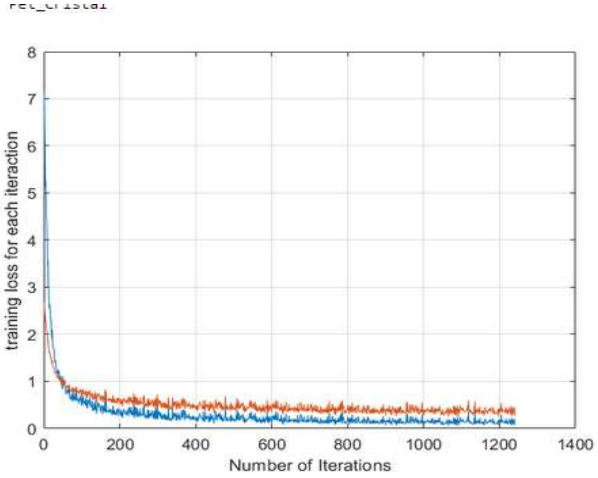


Fig. 4. Función de pérdida con 20 épocas.

categorías de objetos. Para el entrenamiento, se proponen los siguientes requisitos detallados a continuación.

3.1. Categorías

El proyecto CALSIFICADOR DE PET empleó una base de datos de 2000 imágenes en 2 ROI labels: PET CRISTAL Y PET COLOR en la sesión de etiquetado de Matlab (Fig.1), aunque, previamente, la carpeta principal de la base de datos se dividió en dos en relación a su categoría (PET cristal y PET color), cada una de estas carpetas con 1000 imágenes.

Para evitar la pérdida de información al momento de etiquetar mediante el preprocesamiento de imágenes, se estableció un rango de dimensiones en todas las imágenes de 224 x 224 a 780 x 780, todas las imágenes se ajustaron al tamaño máximo establecido con la finalidad de acelerar el tiempo de entrenamiento, ya que, previamente se realizaron otros entrenamientos con imágenes sin reestablecer su tamaño, lo cual generó un tiempo prolongado de entrenamiento, pese a tener pocas categorías, además, el restablecimiento del tamaño, ayudó a igualar el tamaño de las fotografías, pues había fotografías horizontales y verticales en las carpetas. A pesar de que se restableció el formato de las imágenes, la resolución de las imágenes no fue afectada.

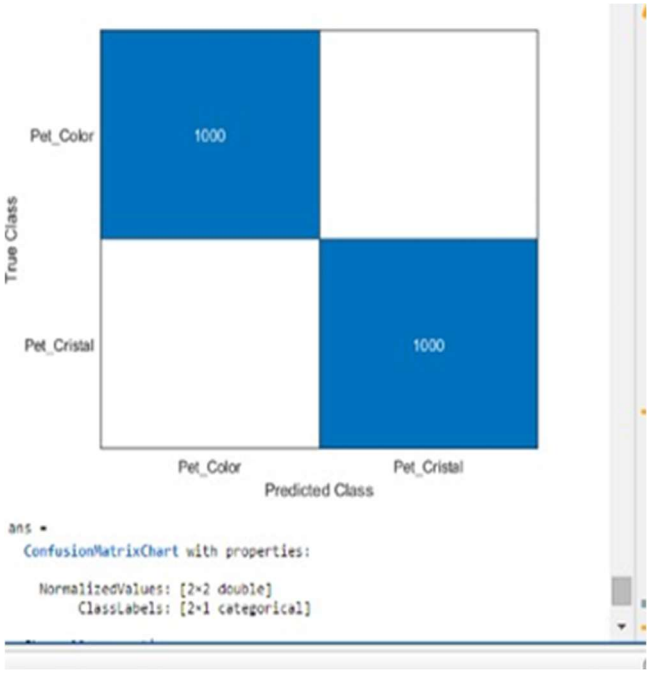


Fig. 5. Matriz de Confusión del entrenamiento petrining.

Tabla 2. Comparación de entrenamiento.

Categoría	Imágenes Totales	Imágenes entrenadas
PET Cristal	1000	1000
PET Color	1000	1000

3.2. Dataset

Como se mencionó en el punto anterior, se utilizó una base de datos con un total de 2000 imágenes (Tabla 1). Los Dataset fueron recopilados por los participantes de este proyecto a partir de fotografías tomadas por ellos mismos, no obstante, hubo una Dataset previa en donde se recopilaron 5000 imágenes, sin embargo, se seleccionaron solo 2000 imágenes debido a sus características, además de ser un número razonable pues no era lo suficientemente grande para retrasar el entrenamiento de la red.

Por otro lado, se eligieron las 2000 imágenes para evitar confusiones, pues ciertas imágenes de la primera Dataset, tenían dos tipos de PETS combinados en una sola foto, lo cual generaba un conflicto al generar la matriz de confusión al estar almacenadas en una carpeta aparte.

Es importante aclarar que algunas imágenes empleadas en las pruebas son en su mayoría imágenes diferentes de la Dataset empleada, ya sea de la versión de 5000 imágenes o descargadas de Internet.



Fig. 6. Detección 1 PET CRISTAL.



Fig. 7. Detección 2 PET CRISTAL.



Fig. 8. Detección 3 PET CRISTAL.

3.3. Arquitectura

Para implementar el detector YOLO v2 en ResNet18 se estableció en inputLayer un tamaño mínimo de [224 224 3], posteriormente se generó una capa ReorgLayerSource en el nodo de Res3a_relu con la finalidad de obtener los datos procesados de las imágenes a través de las capas convolucionales y ReLu para poder realizar una detección ROI mediante los GroundTruth, posteriormente se genera una ReorgConcatenationLayer para finalmente ingresarlo al detector YOLO.

Reorg Layer sirve para combinar características de nivel medio y características de alto nivel y precisión de clasificación de getter a través de diferentes formas, como la combinación máxima y la suma . En YOLO v2, el autor usa una Reorg Layer para remodelar el tensor de salida a fin de que dos salidas de tensor puedan concatenarse juntas [17].

3.4. Entrenamiento

Se realizaron distintos entrenamientos con ciertas variantes en las bases de datos, la base de datos de 2000 imágenes funcionó, las anteriores pruebas tuvieron una



Fig. 9. Detección PET COLOR.



Fig. 10.Detección 2 PET COLOR.



Fig. 11. Detección 3 PET COLOR.

Tabla 3. Comparación de precisión de TESTS.

Categoría	Precisión del TEST
PET Cristal 1	.919
PET Cristal 2	.912
PET Cristal 3	.902
PET Colores 1	.886
PET Colores 2	.900
PET Colores 3	.867

configuración similar a la (Fig.3), pero con cambios en las épocas entre entrenamientos, siendo que partimos con la base de 5000 imágenes y 5 épocas, el primer entrenamiento generó un Mini-batch Loss de .7, una cantidad que podía ser mejorada, se optó por reducir la base de datos, es así que llegamos a la Dataset con su número de imágenes anteriormente mencionada que generó un entrenamiento rápido y efectivo.

Una vez tomada la decisión de usar una Dataset de 2000 imágenes, se realizaron ciertos entrenamientos, siendo 20 épocas y un MiniBatch Size de 32 suficientes para generar una Mini-batch Loss de .2.

Al momento de realizar los entrenamientos, se ocupó una Laptop Hp Pavilon Gaming con Tarjeta gráfica GeForce GTX 1660 Ti with Max-Q Design, durante 20 épocas, obteniendo la función de perdida expuesta en (Fig.4), donde podemos ver una respuesta satisfactoria con esas épocas al irse reduciendo.

Con este entrenamiento se obtiene la matriz de confusión (Fig.5), se puede apreciar que no hay ninguna imagen que no se haya entrenado, lo cual es favorable, pues indica que no hubo pérdidas y que la red pudo aprovechar cada imagen para satisfacer los resultados del Object Detector. Una vez comparada la matriz de confusión con la cantidad total de imágenes con las que se entrenaron, se obtuvo lo indicado en la Tabla 2.

4. Resultados

Los modelos desarrollados fueron evaluados en el conjunto de prueba que consta de 6 imágenes tomadas de bases de datos descartadas anteriormente y algunas sacadas de la Dataset empleada.

Estas imágenes que se busca detectar con el entrenamiento indicado, mostraron, al ser introducidas en el código para pruebas, que PET es el que se encuentra en la imagen, ya sea de color o de cristal, es por lo que se usaron 6 imágenes en total, 3 para cada categoría para poner a prueba el entrenamiento.

Con relación a los resultados en las 3 primeras imágenes introducidas, se muestra que en todos los casos la detección marca que son PET cristal, puede haber ciertos errores, especialmente en la Fig.7 pues, se evitó que se pongan 2 Vounding Boxes al mismo tiempo con la condición de que se marque la opción que tiene más probabilidades al ser mayor de .7 y solo marca PET cristal, al ser el que es más visible en la imagen. En las Fig. [9, 10, 11] se aprecian las pruebas con imágenes que contengan en su mayoría PET de Color, por lo que, los resultados se inclinan y detectan PET COLOR.

Mediante Deep Learning se generó la clasificación, así mismo se entrenó con diferentes bases de datos, ya que contamos con una base de datos de 5,000 imágenes diferentes capturadas por nosotros mismos, luego de hacer los primeros entrenamientos, se mostraba un exceso de salidas al tener 3 carpetas, en la que una contenía imágenes donde se localizaban tanto PET cristal como PET de color en una misma imagen, lo cual no nos permitía generar una matriz de confusión, pues era difícil de interpretar al tener imágenes en que no pertenecían en su totalidad a un solo tipo de PET, por lo que la Matriz de Confusión quedaba desordenada y parecía que no había hecho un entrenamiento adecuado, obstaculizando al conocimiento de los resultados, optándose por una base de datos más pequeña enfocándonos en imágenes puramente de un solo tipo.

Una vez realizado el entrenamiento y las pruebas, en estas hubo una precisión óptima (Tabla 3), dicha conclusión la inferimos al ver el número que se encuentra en las etiquetas, pues indica el número de certeza que tiene el detector de que sea PET cristal o PET de color, aunque se puede observar en la detección 2 de PET cristal que solo

marca un área en un solo tipo e ignora el hecho de que se encuentra PET de color, lo cual se debe a las configuraciones realizadas en el entrenamiento.

5. Conclusiones y trabajo futuro

En este proyecto, se implementaron técnicas Deep Learning [13] y You Only look Once (YOLO) con MATLAB. para la detección y clasificación del PET este clasificador es un proyecto totalmente competente que, una vez desarrollado en un ambiente tangible que no solo se quede en detección de objetos, puede ser empleando principalmente para la industria 4.0 [16].

El proyecto, como ya se mencionó, puede ser aplicado en cualquier sector dedicado al reciclaje y está dirigido principalmente para reducir el trabajo al momento de separar el PET en lugares donde se emplea la mano de obra y como una alternativa a reducir gastos en empresas medianas que emplean técnicas como sensores o máquinas clasificadoras industriales. Si bien los resultados son buenos, aún queda mucho trabajo para implementarlo de manera física en tiempo real, donde no solo sea detectar por medio de fotos, si no por medio de video en tiempo real, además de vincular más funcionalidades a esta técnica como la creación de bases de datos donde se registren las cantidades de cada tipo de PET que pasen frente a la cámara.

Asimismo, es importante llevar a cabo más entrenamientos a fin de que aumente la eficacia de la técnica hasta un nivel con errores mínimos o bien, modificar el código de pruebas para que puedan aparecer más de un Vounding Box, con lo cual se determinarían tanto PET de Colores como PET cristal en una sola imagen. Es importante, poder generar otras alternativas a distintos problemas que puedan ser aplicados en la industria 4.0 mediante la implementación de Deep Learning.

Referencias

1. La falacia de la industria en la lucha contra la contaminación plástica: Estado de Reciclaje en México, https://cdn2.hubspot.net/hubfs/2617486/Greenpeace/templates/pages/alianza/estado-del-reciclaje-reporte-completo.pdf (2019)
2. AS news (2020) https://www.asnews.mx/
3. Molina, I.: México se encamina a ser líder de la industria 4.0 en América Latina. https://mexicoindustry.com/noticia/mexico-se-encamina-a-ser-lider-de-la-industria-40-en-america-latina
4. ISO 14001 Gestión medioambiental. https://www.normas-iso.com/iso-14001/ (2021)
5. Certificación ISO 14001. http://www.fao.org/3/ad818s/ad818s08.htm (2021)
6. Direct industry.: https://www.directindustry.es/fabricante-industrial/clasificador-botellas-188024.html (2021)
7. Acoplásticos. https://www.acoplasticos.org/index.php/mnu-pre/opm-bus-pref/36-opc-fag-pre4 (2021)
8. La clasificación de los plásticos. https://gestoresderesiduos.org/noticias/la-clasificacion-de-los-plasticos (2020)
9. Búler: https://sortex.com.ar/clasificacion-de-plasticos/clasificacion-de-pet/ (2021)
10. MathWorks object detection using YOLO v2 deep learning. https://la.mathworks.com/help/deeplearning/ug/objectdetection-using-yolo-v2.html? s_tid=blogs_rc. 6 (2020)

11. Prototipo de sistema automatizado con visión artificial para la selección de empaques de plástico, vidrio y lata en el proceso de reciclaje. http://repositorio.ucundinamarca.edu.co/bitstream/handle/20.500.12558/2497/Trabajo%20de%20grado.pdf?sequence=3&isAllowed=y
12. Gómez, M. A., Hernández, A., I., Santiago, K. A., Varela, A.: Alternativa para la detección Covid-19 mediante rayos X en México ocupando técnicas de Deep Learning y YOLO V2. (2020)
13. MathWorks Object detection using deep learning. https://la.mathworks.com/help/vision/ug/getting-started-with-object-detection-using-deep-learning.html (2020)
14. Resnet 18. https://la.mathworks.com/help/deeplearning/ref/resnet18.html
15. ImageNet. http://www.image-net.org (2021)
16. Gobierno de México. https://www.gob.mx/se/articulos/el-desarrollo-de-la-industria-4-0-en-mexico?idiom=es (2021)
17. Mao, L.: YOLO v2 reorg layer explained. https://leimao.github.io/blog/Reorg-Layer-Explained/ (2021)
18. Redes neuronales convolucionales. https://la.mathworks.com/discovery/convolutional-neural-network-matlab.html. (2020)

ISSN 1870-4069

Construcción de un modelo automático para la detección de armas en el transporte público

Eduardo Enrique Roldán-Juárez, Gabriela Ramírez-de-la-Rosa, Carlos Joel Rivero-Moreno

Universidad Autónoma Metropolitana,
Departamento de Tecnologías de la Información,
México

{gramirez, crivero}@cua.uam.mx,
eerj1311@gmail.com

Resumen. El delito de asaltos a mano armada en el transporte público es un problema común en las grandes ciudades. La detección de estos eventos es importante en la respuesta policiaca. En este artículo se propone una metodología para detectar armas de fuego en imágenes obtenidas de video-vigilancia en el transporte público. La metodología propuesta incorpora: i) la utilización de una arquitectura existente basada en redes neuronales para la construcción del modelo de detección de armas; ii) un proceso de selección de imágenes cercanas al escenario real; y iii) dos opciones de aumento de datos durante la fase de entrenamiento. Cuando se consideró tanto la selección de imágenes como el aumento de dato en la creación del modelo se obtuvo un 81.33 % de mAP. A pesar del esempeño prometedor en este escenario complicado, es necesaria la obtención de imágenes específicas de este dominio.

Palabras clave: Detección de armas, detección de objetos, aumento de datos.

An Automatic Model for Firearms Detection in Public Transport

Abstract. Armed robbery on public transport is a common problem in large cities. Detecting these events promptly is important for quick police response. In this paper we proposed a methodology to detect firearms in images captured from video surveillance in public transport. The proposed methodology include: i) the use of an existing architecture based on neural networks to build a firearms detection's model; ii) a process of selecting images close to the real scenario; and iii) two options for data augmentation during the training phase. When considering both the selection of images and the increase in data for building an automatic firearms detection model we reach 81.33 % of mAP. Despite promising performance in this challenging scenario we believe domain-specific images are required.

Keywords: Firearms detection, object detection, data augmentation.

1. Introducción

En los últimos años, en México se ha experimentado un aumento gradual de violencia, delincuencia e inseguridad. Según el Índice Global de Paz 2020, México se ubica en el lugar 137 de seguridad y es considerado uno de los países más peligrosos del mundo [10]. La percepción de la población no es diferente a los índices globales. El 68.1 % de la población mexicana de 18 años o más considera que, en términos de delincuencia, vivir en su ciudad es inseguro [7].

Entre 2012 y 2019, se estima que en México hubo un total de 17.9 millones de delitos donde la víctima estuvo presente, de los cuales en un 42.9 % de los casos hubo un arma involucrada. Y en el 31 % de los casos, se trató de un arma de fuego [8, 9]. Datos anuales (del 2010 al 2019) del INEGI [9, 7] muestran que el delito asalto en transporte público o en la calle ha sido el delito de mayor incidencia en México.

El Área Metropolitana del Valle de México (que comprende Ciudad de México y municipios de Hidalgo, Tlaxcala y Estado de México) tiene una tasa de incidencia de más del doble que en el resto del país [9] en este delito. Afortunadamente, existen algunas acciones que se han tomado para aminorar los asaltos en transporte púbico o en la calle.

El organismo encargado de llevar a cabo esas acciones en la Ciudad de México es el Centro de Comando, Control, Cómputo, Comunicaciones y Contacto Ciudadano (o C5). El C5 realiza sus funciones de asistencia a la comunidad a través del vídeo monitoreo y de llamadas telefónicas [3].

Adicionalmente, una de las acciones implementadas por el gobierno de la Ciudad de México es la instalación de más de 15,000 cámaras de video-vigilancia con botones de auxilio, y más de 2000 cámaras que operan en el Sistema de Transporte Colectivo Metro.

También, en 2019 comenzó la implementación del proyecto de monitoreo integral y seguridad del transporte público vía GPS en la Ciudad de México [4]. Este programa consiste en instalar GPS, botón de pánico, contador de pasajeros y videocámaras.

Sin embargo, dado que el operador del transporte público es quien debe presionar el botón de pánico para que la cámara de la unidad sea enlazada directamente al C5, se pone en riesgo al operador.

Una posible alternativa a la activación manual de los botones de pánico es un sistema de monitoreo automático que detecte armas en el video capturado en las unidades de transporte e informe al C5 cuando se requiera atención específica. En este proyecto, nos enfocamos en la construcción de un modelo que permita identificar un arma en una imagen capturada por las cámaras de video-vigilancia.

La identificación de objetos (e.g., armas) en imágenes es una de las tareas más estudiadas y complejas en visión computacional [11]. Ayudado por el resurgimiento de las redes neuronales y la capacidad de cómputo actual, la arquitectura que más se ha utilizado recientemente en este problema son las redes neuronales convolucionales (CNN por sus siglas en inglés) [2, 11].

De manera general, el objetivo de la detección de objetos es que dado un conjunto predeterminado de objetos (o clases), éstos se puedan ubicar y etiquetar mediante una caja delimitadora (bounding box).

La literatura existente en el área se enfoca en el diseño o modificación-adaptación de redes neuronales profundas que funcionen en escenario donde existen un número grande de objetos a detectar o que sean más eficientes en la detección de objetos particulares [20, 2, 11, 18].

En este artículo, nos enfocamos en la detección de un tipo de objeto, i.e. arma de fuego. Particularmente, en la identificación de armas de fuego en imágenes obtenidas de videos del transporte público. Este problema trae consigo retos importantes: el sistema deberá ser robusto a i) las condiciones de captura de las imágenes, ii) entornos no controlados, y iii) ruido en las imágenes. Pues si bien existen cámaras dentro del transporte público, estas cámaras no tienen resoluciones altas, el ángulo de la toma no es controlado, puede obstruirse la cámara y generar cambios de contraste, entre otros problemas.

El objetivo principal de este trabajo es construir un modelo automático de detección de armas en imágenes dentro del transporte público. Nos enfocamos en la adecuación de la calidad de los datos de entrenamiento para que éstos datos sean lo más cercano al escenario de interés (armas usadas en asaltos en el transporte público).

Para ello se propone una metodología que incorpora: i) la utilización de una arquitectura existente basada en redes neuronales para la construcción del modelo de detección de armas, específicamente YOLO[15, 1]; ii) un proceso de selección de imágenes cercanas al escenario real; y iii) dos opciones de aumento de datos en el entrenamiento del modelo. La evaluación experimental muestran que la selección cuidadosa de un conjunto de imágenes de armas y el aumento de datos tanto interno como externo obtienen un desempeño por arriba del 81 % de mAP.

2. Trabajo relacionado

Agrupamos el trabajo relacionado en: i) aquellos que realizan modificaciones o adaptaciones de una arquitectura existente para el problema de clasificación de armas en imágenes [13], y ii) aquellos que utilizan aumento de datos como técnica para mejorar el desempeño del modelo generado [16, 6].

En [13] los autores proponen un sistema de detección de armas en videos con el objetivo de disminuir falsos positivos. Una de las contribuciones de este trabajo fue la construcción de conjuntos de datos. Los autores construyeron 5 conjuntos de datos, 3 de los cuales contienen únicamente dos clases: imágenes con armas e imágenes sin armas.

El mejor resultado (cero falsos positivos y una medida F1 de 0.91) lo obtuvieron usando un enfoque de región para buscar los objetos en las imágenes y la arquitectura Faster Region based-CNN (or Faster R-CNN) sobre el conjunto de datos de 3000 imágenes con 2 clases. Nosotros hacemos uso de este mismo corpus de 3000 imágenes (al que en la Sección 3.2 llamamos Corpus 1).

De los sistemas que hacen uso de técnicas de aumento de datos se encuentra, por un lado, el desarrollado por [16] que busca detectar armas cortas en videos de cámaras de seguridad. El autor construyó una base de datos de imágenes compuesta de más de 17 mil imágenes dividida en dos clases: imágenes con armas y sin armas (el conjunto de datos está balanceado). Se hizo un aumento manual de los datos utilizando rotaciones, re-dimensionamiento y volteo de la imagen.

Para el entrenamiento del modelo utilizaron dos arquitecturas de redes neuronales: VGG (Very Deep Convolutional Networks for Large-Scale Image Recognition) [17] y ZF Net [19]. Los mejores resultados los obtuvieron con VGG con una exactitud de 90 % utilizando imágenes en escala de grises para el entrenamiento. El uso de ejemplos generados por técnicas de aumento de datos resultó benéfico considerando el alto número de imágenes utilizado.

Por otro lado, en [6] realizaron un sistema para detectar pistolas en imágenes. Para el entrenamiento del modelo usaron la arquitectura YOLO v2 [15] y SSD (Single Shot Detection) [12]. Adicionalmente, los autores trabajaron con conjuntos de datos existentes (uno de los cuales es el mismo usado en [13]) más un conjunto de datos artificial que consiste en imágenes creadas a partir de técnicas de aumento de datos. El mejor resultado lo obtuvieron usando YOLO v2 sin aumento de datos, llegando a un mAP de 0.71.

Del análisis de estos trabajos previos podemos resaltar que el corpus de 3000 imágenes introducido por [13] y usado en [13, 16] es un buen punto de referencia. Usar este corpus nos permitirá realizar algunas comparaciones directas con estos trabajos. Por otro lado, el uso de técnicas de aumento de datos fue útil para el caso donde hay un número muy grande de imágenes iniciales.

No tanto para el caso donde se usan aproximadamente 3000 imágenes como origen; sin embargo, esta estrategia es prometedora. Finalmente, los dos últimos trabajos hacen uso de la arquitectura YOLO en sus métodos; mientras [16] lo usa como parte del filtrado de las imágenes a analizar, pues se entrena para identificar personas; [6] hace uso directo de YOLO para la identificación de armas, este enfoque también es prometedor.

A diferencia de estas investigaciones previas, nuestra metodología propone una etapa de selección de imágenes (analizar la composición del conjunto de datos para conservar imágenes lo más parecidas al problema a resolver) en combinación con dos tipos de aumento de datos bajo una arquitectura ampliamente usada en la detección de objetos.

3. Metodología propuesta

La metodología propuesta contempla el uso de un corpus que sea útil para el problema de la detección de armas a partir de imágenes obtenidas dentro del transporte público.

Por un lado, utilizamos una arquitectura basada en redes neuronales convolucionales, particularmente YOLO (que se describe en la Sección 3.1) para la construcción del modelo de detección de armas. Por otro lado, nuestra metodología enfatiza la importancia de los datos para entrenar un modelo. Por lo cual es relevante considerar una etapa de selección de imágenes para la generación de un nuevo corpus.

Finalmente, se propone la utilización de dos tipos de aumento de datos para enriquecer el conjunto de imágenes recolectadas en el paso anterior. Los dos tipos de aumento de datos (externo e interno) se describen en la Sección 3.3. La metodología que incorpora todas las etapas listadas se muestra en la Figura 1.

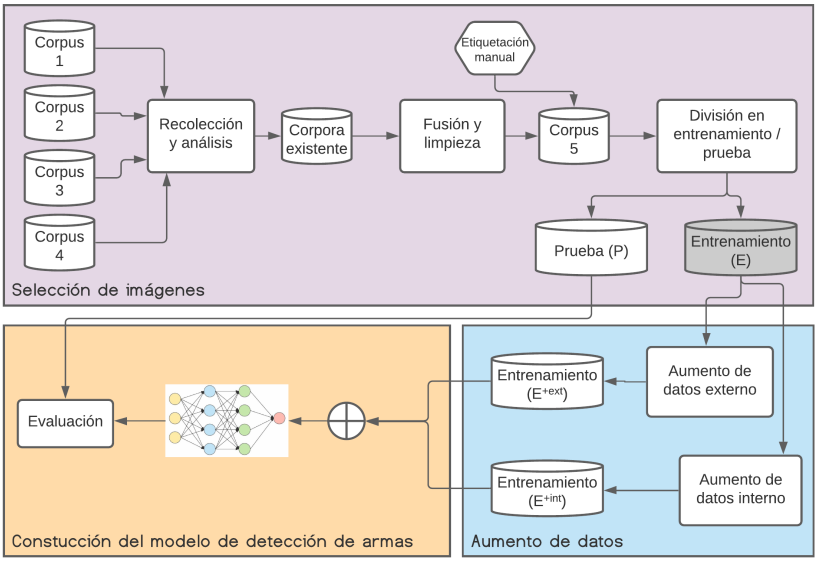


Fig. 1. Esquema general de la metodología propuesta.

3.1. Construcción del modelo de detección de armas

En la Figura 1 (esquina inferior izquierda) se muestran el esquema del proceso de construcción del modelo de detección de armas. De forma general, dado un conjunto de imágenes etiquetadas con armas, se entrena un modelo usando la arquitectura YOLO para posteriormente evaluar el desempeño del modelo usando un conjunto nunca antes visto de imágenes de prueba. A continuación, se describe la arquitectura utilizada.

YOLO: You only look once. YOLO [14] es un sistema de código abierto implementado para hacer detección de objetos en tiempo real. Hace uso de su red neuronal convolucional Darknet, a la que en contraste con otras arquitecturas, se le introducen los ejemplos sólo una vez. Esta característica hace que las detecciones se realicen de manera muy rápida. Además esta arquitectura es capaz de realizar varias detecciones de objetos en una sola imagen.

La red neuronal divide la entrada en regiones y después predice los cuadros delimitadores y probabilidades para cada región. Se optó por utilizar YOLO debido a que en el trabajo de [6], fue este modelo el que dio los mejores resultados, además de que la versión 4 de YOLO se encuentra muy bien calificada dentro del reto del COCO Dataset, siendo el mejor en cuanto velocidad de detección en tiempo real y desempeño [1].

Formato de etiquetado de YOLO. YOLO requiere un archivo de texto por cada elemento del corpus donde se incluirán las etiquetas. La estructura de este archivo contiene x, y, $width$ y $height$ correspondientes a las coordenadas donde se halla el objeto:

```
<clase del objeto>
    <x>,<y>,
    <width>,<height>
```

En la mayoría de casos de asaltos el delincuente lleva el arma en la mano; por lo tanto, en el etiquetado manual que se realiza en nuestra metodología se decidió tomar en cuenta *mano y arma*.

3.2. Selección de imágenes

Dado que no existe un corpus con las características específicas para el problema abordado en este trabajo (detección de armas en el transporte público), la metodología propuesta contempla la recolección y análisis, más la fusión y limpieza de datos. Estos procesos se describen a continuación y se ilustran en el primer renglón de la Figura 1.

Recolección y análisis de corpora existente Los corpora recolectados consisten en imágenes con alguna de las clases de interés: el objeto arma. Estos conjuntos de datos son públicos y han sido usados por la comunidad en trabajos previos.

- Corpus 1[1]. Consiste en 3000 imágenes de armas no etiquetadas. Aproximadamente la mitad son imágenes donde el arma se muestra en tamaños muy grandes, además presentan muchos ángulos repetitivos y sencillos (hacia la izquierda o derecha).
- Corpus 2[2]. Cuenta con 376 imágenes etiquetadas con una única clase (i.e., arma). Este corpus contiene muchos ejemplos de armas largas y armas que se muestran en la totalidad de la imagen. Sólo tiene 160 ejemplos de personas portando armas.
- Corpus 3[3]. Cuenta con 7829 imágenes etiquetadas en tres clases (armas grandes, armas cortas y fuego). Solamente 603 imágenes pertenecen a la clase armas cortas (que son el tipo de armas que nos interesa). De los 603 ejemplos de armas cortas, solo 146 presentan personas portando armas, el resto son solo armas.
- Corpus 4[4]. Cuenta con 333 imágenes no etiquetadas. Este corpus contiene una mayoría de ejemplos de personas con arma (241 imágenes).

Algunos ejemplos de imágenes de estos corpora (particularmente del Corpus 1) se muestra en la Figura 2a. Como puede observarse, los conjuntos de datos descritos son poco aptos para nuestro problema debido a que en el escenario de asaltos en transporte público las armas aparecen en ángulos complicados, el objeto de interés es pequeño y normalmente las imágenes tienen una resolución muy baja (ver Figura 2e para comparación). Ninguna de estas características aparecen en la mayoría de las imágenes incluidas en los corpora descritos.

[1] Obtenido de: https://sci2s.ugr.es/weapons-detection
[2] Obtenido de: https://www.kaggle.com/rogkesavan000/gun-dataset
[3] Obtenido de: https://www.kaggle.com/atulyakumar98/fire-and-gun-dataset
[4] Obtenido de https://www.kaggle.com/issaisasank/guns-objet-detection



(a) Imágenes en los corpora existentes.



(b) Imágenes en el Corpus 5.



(c) Imágenes ejemplo del proceso de aumento de datos externo.



(d) Imágenes ejemplo del proceso de aumento de datos interno (implementadas en YOLO).



(e) Ejemplos de imágenes reales obtenidas de la video vigilancia del transporte público.

Fig. 2. Ejemplos de los imágenes en las diferentes etapas de la metodología presentada.

Fusión y limpieza de corpora Para atender las limitaciones del corpora existente, se optó por eliminar: todas las imágenes donde el arma se presenta en ángulos sencillos o repetitivos, armas mostradas en una gran parte de la imagen (ejemplos de éstas se pueden observar en la Figura 2a).

Se dio prioridad a los ejemplos donde el arma tuviera algún portador (arma en mano) y armas en posiciones no triviales. Como resultado de esta limpieza se formó el *Corpus 5*[5] que contiene un total de 2044 imágenes (algunos ejemplos de imágenes en este corpus se puede observar en la Figura 2b).

[5] El Corpus 5 y el código fuente utilizado en este artículo está disponible en https://github.com/eduardo1311/Detector-de-armas-con-YOLO

Para utilizar el Corpus 5 en la construcción de un modelo de detección de armas es necesario etiquetar las armas presentes en cada imagen.

Por lo tanto, el siguiente paso es hacer el etiquetado manual. Esto es, se ubicó el arma que aparece en la imagen dentro de una caja delimitadora (bounding box). Para este proceso se usó el formato de etiquetado YOLO (ver Sección 3.1). Si el arma está siendo portada, se consideró parte de la mano y el arma en el etiquetado.

3.3. Aumento de datos

Consecuentemente, después del proceso de selección de imágenes, el conjunto resultante de ejemplos es pequeño (2044 en total). Por lo tanto, nuestra metodología propone la utilización de técnicas de aumento de datos. Como puede observarse en los ejemplos de la Figura 2b el conjunto de imágenes resultante en el Corpus 5, aunque son más parecidas al escenario real (asaltos) aún distan de las imágenes obtenidas en el transporte público (ver Figura 2e).

Así, el objetivo principal del aumento de datos es: i) incorporar variaciones a las imágenes tanto en posición como en calidad. Para atender a la variación de la posición se propone el tipo de aumento que denominamos *externo*; para atender a la variación de la calidad de las imágenes, se propone el tipo de aumento que denominamos *interno*.

En esta etapa se genera un conjunto de imágenes que contienen, dependiendo del tipo de aumento de datos utilizado, las imágenes originales más las adicionales (a estos conjuntos les nombramos E^{+ext} y E^{+int}, respectivamente).

Aumento de datos externo. Consta de rotaciones de 45 grados a cada elemento del conjunto de entrenamiento. Se obtienen 7 ejemplos nuevos por cada imagen. Un ejemplo de este tipo de aumento se puede observar en la Figura 2c.

Aumento de datos interno. YOLO implementa módulos de aumento de datos como parte del procesamiento que realiza antes de introducir los ejemplos a la red neuronal. A continuación, se describen las variantes de aumento proporcionado por YOLO que se tomaron en cuenta. Ejemplos de este tipo de aumento se puede observar en la Figura 2d.

- Saturación: Aleatoriamente cambia la saturación de la imagen.
- Exposición: Aleatoriamente cambia el brillo de la imagen.
- Matiz: Aleatoriamente cambia el matiz (color) de la imagen.
- Ruido Gaussiano: Añade ruido Gaussiano a la imagen.
- Flip: Invierte la imagen horizontal o verticalmente.
- Mosaico (Mejora añadida en YOLOv4): Combina varias imágenes en una sola.
- Difuminación (Mejora añadida en YOLOv4): Aplica un desenfoque aleatorio donde se borrará el fondo a excepción del objeto.

Tabla 1. Experimentos propuestos. Se marca con una x cuando se considera el aumento de datos externo o interno según lo indique el nombre de la columna.

Experimento	Aumento externo	Aumento interno
yolov2	-	-
yolov2-int	-	x
yolov2-ext	x	-
yolov2-int-ext	x	x
yolov4	-	-
yolov4-int	-	x
yolov4-ext	x	-
yolov4-int-ext	x	x

4. Evaluación experimental

Para la realización de los experimentos realizamos una división fija de las imágenes en el Corpus 5 de forma que se permita una comparación directa entre los diferentes modelos de detección de armas evaluados.

Así, se obtienen dos subconjuntos: entrenamiento (E) y prueba (P) (como se ilustra en el esquema general del método propuesto en la Figura 1). Para esta división se asignó el 75 % de ejemplos al conjunto de entrenamiento (1537 imágenes) y el 25 % a el conjunto de prueba (507 imágenes).

Se utilizó Colab de Google para hacer el entrenamiento ya que esta plataforma permite tener acceso remoto a equipos con GPU con un límite de 12 horas [5]. Para cada uno de los experimentos, se fijó el tamaño de lote en 64, la tasa de aprendizaje en 0.001, y se realizaron 10000 iteraciones.

Todos los experimentos fueron evaluados con un nivel de confianza de IoU (i.e., la intersección de la unión de las cajas delimitadoras reales y la detectada) de 50 % para medir el mAP (mean Avarage Precision, que mide el área bajo la curva de la relación precisión-recuerdo).

Con el objetivo de explorar la utilidad de las técnicas de aumento de datos utilizadas en la metodología presentada se proponen experimentos con las cuatro combinaciones generadas de usar o no usar el aumento de datos externos y el aumento de datos interno. La lista completa de experimentos realizados se muestra en la Tabla 1.

4.1. Resultados y discusión

En la Figura 3 se muestra una gráfica comparativa de los 8 experimentos realizados. Las líneas punteadas corresponden a los modelos entrenados con YOLOv2 y las líneas continuas a los entrenados con YOLOv4. En esta gráfica se puede notar que sí existe una brecha entre las versiones 2 y 4 de YOLO.

La versión 2 no supera el 75 % de mAP en ninguna de las variantes, mientras el comportamiento de la cuarta versión de YOLO es por arriba de ese rango. Por otra parte, observamos que al considerar el aumento de datos se obtienen mejores resultados que no usarlo.

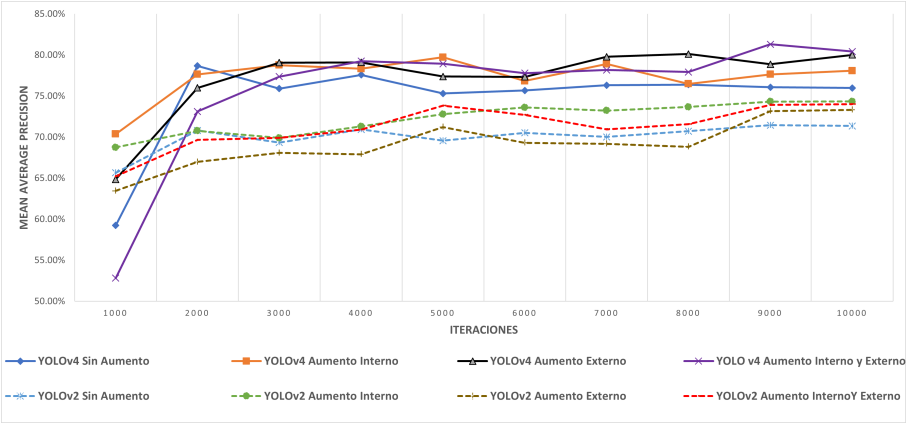


Fig. 3. Gráfica comparativa de resultados obtenidos. Se reporta la métrica de desempeño mAP. Las líneas punteadas corresponden al modelo entrenado sobre YOLOv2, y las líneas sólidas corresponden a desempeño de YOLOv4.

Tabla 2. Mejores resultados por cada experimento. La columna iteración indica el momento en el que el modelo obtuvo su mejor desempeño.

Experimento	iteración	mAP
yolov2	9000	71.44
yolov2-int	10000	74.33
yolov2-ext	10000	73.32
yolov2-int-ext	10000	74.00
yolov4	4000	77.58
yolov4-int	5000	79.71
yolov4-ext	8000	80.12
yolov4-int-ext	9000	**81.30**

Se obtuvieron los mejores resultados con YOLOv4 con aumento de datos interno y externo (yolov4-int-ext). Así mismo, el mejor resultado del modelo generado con YOLOv2 sin aumento de datos (yolov2) es el peor en comparación con los mejores resultados de los demás modelos, esto lo podemos notar en la Tabla 2.

De esta tabla se puede notar que la mayoría de los experimentos llegan a su punto máximo en las últimas iteraciones donde se disminuye la tasa de aprendizaje. También podemos notar que entre la línea de base (yolov2) y el experimento con el mejor rendimiento (yolov4-int-ext) hay casi un 10 % de diferencia.

4.2. Ajuste de los modelos

Para mejorar el desempeño de los modelos, se propusieron experimentos adicionales sobre los parámetros de la arquitectura de la red neural. En estos experimentos se modifica la tasa de aprendizaje (para considerar además de 0.001 a 0.01 y 0.0001) y el tamaño de lote (además de 64 considerar también 32 y 128).



(a) Falsos positivos.



(b) Falsos negativos.



(c) Verdaderos positivos.

Fig. 4. Ejemplos de imágenes en el conjunto de prueba y el resultado de la evaluación cualitativa. Bajo cada imagen se muestra la resolución en pixeles.

Para la realización de estos experimentos, se usó la configuración de datos del experimento yolo4-ext (i.e., solo aumento de datos externo) debido a que ese fue el modelo con el rendimiento más parecido al de los mejores resultados, pero que el tiempo de entrenamiento es la mitad.

Sin embargo, ninguna de estas modificaciones representó mejora alguna. El mejor resultado obtenido fue de 74 % de mAP.

4.3. Evaluación cualitativa

Con la finalidad de evaluar de forma cualitativa el modelo de detección de armas generado en los experimentos previos se eligió de forma manual una muestra del conjunto de prueba (`P`). Esta muestra consistió de 30 ejemplos de imágenes que consideramos eran más cercanas al escenario real (asaltos), ya que es este escenario en donde se pretende que funcione este sistema propuesto.

Para este análisis se usaron las métricas de evaluación tradicionales para sistemas de aprendizaje supervisado; esto es, falsos positivos, falsos negativos y verdaderos positivos.

Se aplicó el modelo yolov4-int-ext al conjunto de imágenes seleccionadas. De las 30 imágenes, el modelo detectó 19 verdaderos positivos, 13 falsos negativos y 5 falsos positivos.

De aquí se obtiene una precisión de 79 %, un recuerdo de 59 % y una medida-F de 0.629. En la Figura 4 se pueden observar tres ejemplos de imágenes con falsos positivos (se identificó un objeto diferente como arma), falsos negativos (no se identificaron armas donde había un arma), y verdaderos positivos (se identificaron armas de forma correcta).

Al analizar los resultados de las 30 imágenes y la Figura 4 se observa que las predicciones tienden a ser erróneas cuando la resolución de las imágenes es baja (por ejemplo la imagen de la izquierda y derecha en la Figura 4a y todas las imágenes en la Figura 4b).

También se pueden ver fallas en casos donde es demasiado complicado encontrar el arma incluso para el ojo humano (por ejemplo en las imágenes de la Figura 4b). Por otra parte, cuando el arma se presenta con fondos contrastantes a su alrededor, es más probable que se haga una detección correcta (por ejemplo las imagen de la izquierda y centro en la Figura 4c), pero si el fondo es oscuro o con un tono de color similar al del arma, será más fácil que el modelo se equivoque (ver las imágenes de la izquierda y la derecha de la Figura 4b).

5. Conclusiones

En este artículo se presentó una metodología para generar un modelo automático de detección de armas de fuego en imágenes obtenidas en el transporte público. La metodología propuesta consiste de tres partes, primero un proceso de selección de imágenes de corpora existente donde las imágenes seleccionadas se parezcan a nuestro escenario real.

Luego, la opción de utilizar dos tipos de técnicas de aumentos de datos para incorporar variaciones a las imágenes tanto en posición como en calidad. Finalmente, la metodología usa una arquitectura de redes neuronales convolucionales utilizada en trabajos previos para la detección de armas: YOLO.

Para determinar la utilidad de realizar una selección de datos y posterior automento de datos en un modelo de detección de armas, se propusieron y realizaron 8 experimentos con dos de las arquitecturas más utilizadas en la detección de objetos: YOLOv2 y YOLOv4. El mejor desempeño se obtuvo en el modelo entrenado con YOLOv4 y usando aumento de datos externo e interno, obteniendo un 81.33 % de mAP.

Después del análisis de los resultados se pudo observar que la calidad de las imágenes usado para entrenar un modelo es determinante para la correcta detección de armas de fuego. Por lo tanto, la limpieza de corpora ha ayudado a obtener mejores modelos en la detección de armas aplicada a ejemplos de asaltos reales, aunque esto disminuye considerablemente el tamaño del corpus.

Aumentar los datos de manera artificial tanto interna como externamente nos ha ayudado a ampliar la variabilidad de imágenes en el conjunto de datos usados para el entrenamiento y ha contribuido a mejorar el desempeño del modelo generado. Sin embargo, esto no resuelve todas las problemáticas del escenario real. Por ejemplo, el modelo tiende a equivocarse cuando la resolución de las imágenes es pequeña o cuando no existen suficiente contraste entre el arma y el fondo donde ésta aparece, dos de los aspectos que ocurren mucho en el escenario real.

Sin embargo, los resultados obtenidos en la evaluación del modelo son prometedores a pesar de haber utilizado una base de datos que contiene en su mayoría ejemplos diferentes a los que hay en el escenario real (como se pudo ver en la Figura 2). Estos resultados nos dan pauta a considerar la replicación de la metodología propuesta con un conjunto inicial de datos obtenidos en la vida real (tomados directamente de los videos que obtiene el C5, por ejemplo).

Como trabajo futuro se planea realizar un etiquetado manual diferente al realizado en este trabajo. Pues en la evaluación cualitativa se observan casos que cuando el modelo comete falsos positivos, es porque identifica una mano en una posición común a la forma de portar un arma que es usada para asaltar. Así mismo, se realizará una comparación exhaustiva con otras arquitecturas usadas en el estado del arte.

Agradecimientos. Los autores y autora agradecen a la Universidad Autónoma Metropolitana Unidad Cuajimalpa por el apoyo otorgado durante la realización de este trabajo.

Referencias

1. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M.: Yolov4: Optimal speed and accuracy of object detection (2020) doi: 10.48550/arXiv.2004.10934
2. Dhillon, A., Verma, G. K.: Convolutional neural network: A review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, vol. 9, no. 2, pp. 85–112 (2020) doi: 10.1007/s13748-019-00203-0
3. Gobierno de la Ciudad de México: El C5 de la CDMX (2021)
4. Gobierno de la Ciudad de México: Transporte público más seguro (2021)
5. Google Colaboraty: ¿Qué es colaboratory? (2021)
6. Gutiérrez-Lancho, C.: Detección de armas en vídeos mediante técnicas de deep learning. Master's thesis (2019)
7. INEGI: Encuesta nacional de seguridad pública urbana, pp. 1 (2020)
8. INEGI: Encuesta nacional de victimización y percepción sobre seguridad pública, pp. 38 (2020)
9. INEGI: Tasa de incidencia delictiva por entidad federativa de ocurrencia por cada cien mil habitantes (2021)
10. Institute for Economics and Peace: Global peace index 2020 briefing (2020)
11. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: A survey. International Journal of Computer Vision, vol. 128, no. 2, pp. 261–318 (2020) doi: 10.1007/s11263-019-01247-4
12. 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, pp. 21–37 (2016) doi: 10.1007/978-3-319-46448-0_2

13. Olmos, R., Tabik, S., Herrera, F.: Automatic handgun detection alarm in videos using deep learning. Neurocomputing, vol. 275, pp. 66–72 (2018) doi: 10.1016/j.neucom.2017.05.012
14. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016) doi: 10.1109/CVPR.2016.91
15. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263–7271 (2017) doi: 10.1109/CVPR.2017.690
16. Romero-Mogrovejo, D. O.: Desarrollo de un sistema de detección de armas de fuego cortas en el monitoreo de videos de cámaras de seguridad. Master's thesis (2018)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014) doi: 10.48550/arXiv.1409.1556
18. Verma, G. K., Dhillon, A.: A handheld gun detection using faster r-cnn deep learning. In: Proceedings of the 7th International Conference on Computer and Communication Technology, pp. 84–88 (2017) doi: 10.1145/3154979.3154988
19. Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833 (2014) doi: 10.1007/978-3-319-10590-1_53
20. Zhao, Z. Q., Zheng, P., Xu, S., Wu, X.: Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232 (2019) doi: 10.1109/TNNLS.2018.2876865

ISSN 1870-4069

Comparación del rendimiento del algoritmo basada en forrajeo de bacterias con Matlab y Octave

Adrian García-López, Betania Hernández-Ocaña, Oscar Chávez-Bosquez, José Hernández-Torruco, Juana Canul-Reich

Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencias y Tecnologías de la Información.
México

192H13002@alumno.ujat.mx,
{betania.hernandez, oscar.chavez, jose.hernandezt,
juana.canul}@ujat.mx

Resumen. Existen metaheurísticas bio-inspiradas en la naturaleza basadas en bacterias que emulan su comportamiento de manera computacional, tal es el caso del Algoritmo de Optimización Basado en el Forrajeo de Bacterias Modificado con Dos Nados (TS-MBFOA, por sus siglas en ingles). Este algoritmo genera un conjunto de soluciones óptimas aproximadas a problemas de optimización complejos en un tiempo razonable. TS-MBFOA está codificado de manera modular en el lenguaje de programación M de Matlab, el cual es un software comercial cuyo alto costo no es accesible para todo usuario. Existen alternativas de uso para ejecutar código del lenguaje M, tal es el caso de GNU Octave, un proyecto de software libre de gran utilidad en el campo del cálculo numérico. En este artículo se hace una comparación del rendimiento del algoritmo TS-MFOA implementado y ejecutado en Matlab y Octave. Ambos entornos son usados para ejecutar a TS-MBFOA y dar solución al problema de prueba conocido como el resorte de tensión/compresión, con el objetivo de determinar las ventajas y desventajas de estas dos plataformas y comprobar si el lenguaje de programación influye en el rendimiento del algoritmo en el proceso de optimización.

Palabras clave: Forrajeo de bacterias, metaheurísticas, Matlab, Octave, optimización.

Performance Comparison of the Bacterial Foraging Algorithm Under Matlab and Octave

Abstract. There are bio-inspired metaheuristics in nature based on bacteria that computationally emulate their behavior; such is the case of the Modified Bacterial Foraging Optimization Algorithm with two swims (TS-MBFOA). This algorithm generates a set of approximately optimal solutions to complex optimization problems in a reasonable time. TS-MBFOA is implemented in a

modular way in the Matlab M programming language, commercial software with high cost not accessible to all users. There are alternatives to execute code of the M language; such is the case of GNU Octave, a free software project of great utility in numerical calculation. This article compares the performance of the TS-MFOA algorithm implemented and executed in Matlab and Octave. Both environments are used to run TS-MBFOA and solve the test problem known as the tension/compression spring to determine the advantages and disadvantages of these two platforms and check if the programming language influences performance.

Keywords: Bacterial foraging, metaheuristic, Matlab, Octave, optimization.

1. Introducción

Una gran clase de problemas lógicos que surgen de situaciones del mundo real pueden formularse como problemas de optimización, ya sea para maximizar o minimizar las variables de diseño de una función objetivo y, por lo tanto, se pueden describir como una búsqueda de la mejor solución. Existen una serie de problemas reales de optimización de difícil solución que requieren encontrar el camino más corto para su solución y una secuencia óptima de sus procesos de trabajo.

Estos problemas suelen requerir agrupamientos, ordenaciones o asignaciones de un conjunto discreto de objetos que satisfagan ciertas restricciones. Para resolver un problema de optimización se necesita asistirse por programación matemática, pero al momento de su implementación pueden ser complicado o confuso [8].

Existen técnicas que ofrecen una mejor solución que la programación matemática, estas se llaman metaheurísticas o algoritmos bio-inspirados y son estrategias para diseñar o mejorar los procedimientos matemáticos orientados a obtener un alto rendimiento [14]. Los algoritmos bio-inspirados tienen dos sub-grupos que se distinguen de acuerdo con el tipo de fenómeno natural en el que se basan: los Algoritmos Evolutivos (AE) que basan su funcionamiento en la teoría de la evolución de las especies y la supervivencia del más apto [4].

Por otro lado, los Algoritmos de Inteligencia Colectiva (AIC) que toman comportamientos de seres vivos como: parvadas de aves, bancos de peces, cúmulos de insectos, hormigas, bacterias, entre otros [2], y estas interactúan en su ambiente donde surgen comportamientos cooperativos que permiten a estos seres simples, de manera individual resolver problemáticas complejas de manera conjunta.

Uno de los AIC más novedoso es el Algoritmo de optimización basado en el forrajeo de bacterias (BFOA, por sus siglas en inglés), este simula de manera computacional el proceso completo de forrajeo de la bacteria E. Coli y fue propuesto por Passino [13]. Su funcionamiento consiste en cuatro pasos: quimiotáxis (movimientos de nados-giros), agrupamiento, reproducción y eliminación-dispersión. Durante estas etapas las bacterias se enfrentan a varias problemáticas en la búsqueda de sus alimentos [10].

El BFOA requiere de la calibración de muchos parámetros y su costo computacional es alto debido al grupo de ciclos anidados que permiten su funcionamiento.

En la propuesta MBFOA, los autores hacen una disminución de parámetros del algoritmo y se aplica un mecanismo para el manejo de las restricciones [11]. Otra propuesta del algoritmo es el Improved MBFOA donde la modificación implementa un mecanismo de sesgo para crear la población inicial, dos operadores de nado, tamaños de paso dinámico y el buscador local [7].

Una de las más recientes modificaciones y la que es utilizada en este artículo es el TS-MBFOA donde se intercalan dos nados en el proceso quimiotáxico. El primero es el nado original a excepción del tamaño de paso, el cual se propone sea aleatorio, y el segundo nado incluye el operador de mutación usado en los algoritmos evolutivos, la inclusión de ambos con el objetivo de mejorar la capacidad de exploración y explotación del algoritmo [5].

El TS-MBFOA es una metaheurística que permite tener resultados competitivos y favorables al solucionar Problemas de Optimización Numérica con Restricciones (PONR). Esta propuesta recientemente se ha utilizado con éxito para resolver problemas de diseño ingenieril, como el conocido Resorte de Tensión / Compresión [5] y la generación de menús nutritivos [6].

Dicho algoritmo posee sus propios parámetros que corresponden a cada uno de sus procesos, los cuales son independientes de las variables del problema a resolver y deben ser calibrados o ajustados por el usuario final de tal forma que permitan un mejor rendimiento del algoritmo.

Una de las características generales de las metaheurísticas es que algunas están codificadas en Matlab, un software comercial de alto costo con difícil acceso para algunos miembros de la comunidad científica. Existe una alternativa de solución para los algoritmos codificados en Matlab, este es GNU Octave, un entorno de programación equivalente pero de licencia libre.

Por esta razón, la implementación del TS-MBFOA se ejecuta en Matlab y GNU Octave para hacer la comparación de estos lenguajes de programación y determinar las ventajas de tiempos de ejecución y la calidad del conjunto de resultados que se obtienen. De esta manera, los resultados obtenidos se validan usando estadísticas básicas como mejor y peor valor, media, mediana y desviación estándar.

2. Two-Swim Modified Bacterial Foraging Optimization Algorithm (TS-MBFOA)

El TS-MBFOA es un algoritmo derivado de MBFOA propuesto para resolver PONR [5], en el cual una bacteria i representa una solución potencial y se denota como $\theta^i(j, G)$, donde j es el ciclo quimiotáxico y G es el ciclo generacional. Una generación consta de un proceso quimiotáxico, agrupamiento, reproducción y eliminación-dispersión.

En el proceso de quimiotáxis dos nados se intercalan, en cada ciclo solo un nado de explotación o exploración es realizado. El proceso comienza con el nado de explotación (nado clásico). Sin embargo, una bacteria no necesariamente intercalará exploración y explotación en los nados, ya que si la nueva posición de un nado dado, $\theta^i(j+1, G)$ tiene una mejor aptitud (basado en las reglas de factibilidad) que la posición original

$\theta^i(j, G)$, otro nado en la misma dirección se llevará a cabo en el siguiente ciclo. De lo contrario, un nuevo giro será calculado. El proceso se detiene después de N_c intentos.

El nado de exploración usa la mutación entre bacterias y es calculado con la Ecuación 1:

$$\theta^i(j+1, G) = \theta^i(j, G) + (\beta)(\theta^r_1(j, G) - \theta^r_2(j, G)), \tag{1}$$

donde $\theta^r_1(j, G)$ y $\theta^r_2(j, G)$ son dos bacterias diferentes seleccionadas aleatoriamente de la población. β es un parámetro definido por el usuario utilizado en el operador de agrupamiento el cual define la cercanía de la nueva posición de una bacteria con respecto a la posición de la mejor bacteria de la población, en este operador, β es un parámetro de control positivo para escalar los diferentes vectores en (0,1], es decir, escalas de la zona donde una bacteria puede moverse. El nado de explotación es calculado con el Ecuación 2:

$$\theta^i(j+1, G) = \theta^i(j, G) + C(i, G)\phi(i), \tag{2}$$

donde $\phi(i)$ se calcula con el operador de giro original de BFOA definido en la Ecuación 3:

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}}, \tag{3}$$

$\Delta(i)^T$ es un vector aleatorio generado con elementos dentro de un intervalo $[-1, 1]$. $C(i, G)$ es el tamaño de paso aleatorio de cada bacteria actualizado con la Ecuación 4:

$$C(i, G) = R * \Theta(i), \tag{4}$$

donde $\Theta(i)$ es un vector generado de forma aleatoria de tamaño n con elementos dentro del rango de cada variable de decisión: $[U_k, L_k]$, $k = 1, ...n$, y R es un parámetro definido por el usuario para escalar el tamaño de paso, este valor debe ser cercana a cero (por ejemplo 5.00e-04). La inicial $C(i, 0)$ se genera utilizando $\theta(i)$. Este tamaño de paso aleatorio permite que las bacterias se puedan mover en diferentes direcciones dentro del espacio de búsqueda y evita la convergencia prematura como se sugiere en [9].

En el ciclo medio del proceso quimiotáxico es aplicado el operador de agrupamiento con la Ecuación 5, donde β es un parámetro positivo definido por el usuario entre (0,1):

$$\theta^i(j+1, G) = \theta^i(j, G) + \beta(\theta^B(G) - \theta^i(j, G)), \tag{5}$$

donde $\theta^i(j+1, G)$ es la nueva posición de la bacteria i, $\theta^B(G)$ es la actual posición de la mejor bacteria generacional y β es un parámetro llamado factor de escalamiento, el cual regula qué tan cerca estará la bacteria i de la mejor bacteria θ^B. Sin embargo, si una solución viola el límite de las variables de decisión, una nueva solución de x_i es generada aleatoriamente entre los límites inferior y superior $L_i \leq x_i \leq U_i$ de las variables de decisión.

En la reproducción se ordenan las bacterias con base en la técnica de manejo de restricciones, eliminando a las peores bacterias $S_b - S_r$ y duplicando a las mejores cada cierto número de ciclos, definido por el usuario con el parámetro $RepCycle$. En la eliminación-dispersión se elimina a la peor bacteria de la población $\theta^w(j, G)$ (basado en las reglas de factibilidad) y se genera una nueva aleatoriamente.

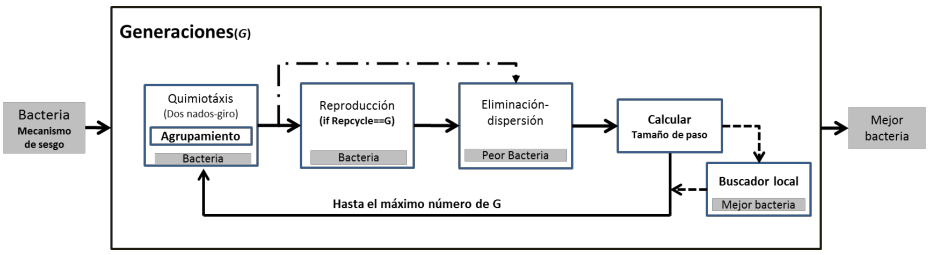


Fig. 1. Procesos generales de TS-MBFOA.

Algorithm 1: Pseudocódigo de TS-MBFOA.

Crear una población inicial de bacterias aleatorias $\theta^i(j,0)\ \forall i,\ i = 1,\ldots,S_b$
Evaluar $f(\theta^i(j,0))\ \forall i,\ i = 1,\ldots,S_b$
for *G=1 to GMAX* **do**
  for *i=1 to* S_b **do**
    for *j=1 to* N_c **do**
      En el proceso quimiotáxico intercalar los nados propuestos con las Ecuaciones 1 y 2. Aplicar el operador de agrupamiento con la Ecuación 5 usando β para la bacteria $\theta^i(j,G)$
    end
  end
  if $G\ mod\ RepCycle == 0$ **then**
    Realizar el proceso de reproducción ordenando la población de acuerdo a las reglas de factibilidad y eliminar a S_r peores bacterias y duplicar el resto de bacterias $S_b - S_r$.
  end
  Realizar el proceso de eliminación-dispersión eliminando a la peor bacteria $\theta^w(j,G)$ de la población actual considerando la técnica de manejo de restricciones.
  Actualizar el vector de tamaño de paso usando la Ecuación 4.
end

Fig. 2. Pseudocódigo del TS-MBFOA.

Aunque en su propuesta original de TS-MBFOA se utiliza un mecanismo de sesgo para generar la población inicial aleatoria y un buscador local, en este artículo no se hace uso de dicho mecanismo para consumir menos costo computacional.

La estructura del TS-MBFOA se presenta en la Figura 1. El pseudocódigo de TS-MBFOA es presentado en la Figura 2.

3. Materiales y método

Para la ejecución de la implementación del TS-MBFOA en Matlab y GNU Octave, a continuación, se describen cada uno de ellos, mostrando sus características principales como lenguajes de programación de alto nivel.

3.1. Matlab

Matlab no es sólo un entorno de desarrollo, también un lenguaje de programación. Matlab integra el cálculo, la visualización y la programación en un ambiente de desarrollo de alto nivel y básicamente permite resolver muchos problemas computacionales, específicamente aquellos que involucren vectores y matrices, en un

tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran.

Matlab se utiliza ampliamente en: cálculos numéricos, desarrollo de algoritmos, modelado, simulación y prueba de prototipos, análisis de datos, exploración y visualización, graficación de datos con fines científicos o de ingeniería y desarrollo de aplicaciones que requieran de una interfaz gráfica de usuario (GUI, Graphical User Interface).

En el ámbito académico y de investigación, es la herramienta estándar para los cursos introductorios y avanzados de matemáticas, ingeniería e investigación. También es la herramienta usada para el análisis, investigación y desarrollo de nuevos productos tecnológicos [3].

Matlab está disponible en sistemas operativos como: Unix, Windows, macOS y GNU/Linux, pero también cuenta con desventajas como: software comercial de alto costo, no se puede ampliar, problemas eventuales de velocidad y poca distribución de ejecutables.

3.2. GNU Octave

GNU Octave es un entorno matemático para realizar cálculos numéricos, el cual es completamente gratuito basado en la filosofía GNU (Sistema operativo libre). Octave comparte características con Matlab donde se puede destacar que ambos ofrecen un intérprete permitiendo ejecutar órdenes en modo interactivo [1].

Entre las funcionalidades más importantes que incorpora Octave pueden mencionarse: elevada capacidad para resolver problemas del Álgebra Lineal, cálculo raíces de ecuaciones no lineales, integración numérica de funciones, manipulación de polinomios, integración de ecuaciones diferenciales, incorporación de cajas de herramientas, Se puede ejecutar en plataformas Unix y Windows, Puede cargar archivos con funciones de Matlab de extensión .m.

Octave soluciona algunos de los inconvenientes técnicos de Matlab, aunque no tiene módulos para el desarrollo de interfaz gráfica, algo visto como una desventaja. Pero cuenta con características especiales como: un buen analizador sintáctico y acepta estructuras complejas.

Aunque el intérprete tiene problemas de estabilidad como el de Matlab se solucionan más rápido puesto que el tiempo de publicación de nuevas versiones es más corto. Es además es muy fácil de extender en C++ y permite un acceso directo a su maquinaria interna y a sus bibliotecas para utilizarlas en cualquier programa en C++ [12]. A continuación, en la Tabla 1 se muestra un comparativo de características de Matlab y GNU Octave.

En la codificación existente se ha utilizado el paradigma de programación modular, con el fin de separar cada uno de los procesos del algoritmo, que básicamente son 4 procesos, además de tener una programación clara de cada uno de ellos usando métodos como funciones y procedimientos con y sin retorno de información.

En la plataforma Github se encuentra de manera gratuita la codificación del TS-MBFOA para resolver el problema del resorte de tensión/compresión en lenguaje M, la cual puede consultarse y descargarse a través del link https://github.com/garcialopez/TS-MBFOA-OCTAVEvsMATLAB.

Tabla 1. Características comparativas de Matlab y GNU Octave.

Característica	**GNU Octave**	**Matlab**
Ejecuta lenguaje M	Sí	Sí
Multiplataforma	Sí	Sí
Simulador	Sí	Sí
Soporte de programación	No	Sí
Entorno visual	Sí	Sí
Desarrollo de interfaces gráficas	No	Sí
Software libre	Sí	No

4. Resultados

El algoritmo TS-MBFOA es ejecutado en los entornos Matlab y Octave de manera independiente en 30 ejecuciones. El problema de prueba es el PONR conocido como Resorte de Tensión/Compresión donde se busca minimizar el peso de un resorte (ver Figura 3) sujeto a restricciones de desviación mínima, tensión de corte, frecuencia de oleada, límites sobre el diámetro exterior, esto sobre variables de diseño.

El diseño de la función para ser procesada en el algoritmo TS-MBFOA cuenta con las siguientes variables de diseño: Diámetro del cable d(x_1), Diámetro del rollo D(x_2), y Número de rollos involucrados N(x_3). Formalmente, el problema puede expresarse de la siguiente manera:

$$(x_3 + 2)x_2x_1^2. \tag{6}$$

Sujeto a:

$$\begin{aligned} &g_1(X) = 1 - ((x_2^3x_3)/(71785x_1^4)) \leq 0,\\ &g_2(X) = ((4x_2^2 - x_1x_2)/12566(x_2x_1^3 - x_1^4))) + (1/5108x_1^2) - 1 \leq 0,\\ &g_3(X) = 1 - (140.45x_1/x_2^2x_3) \leq 0,\\ &g_4(X) = (x_2 + x_1/1.5) - 1 \leq 0, \end{aligned} \tag{7}$$

donde:

$$\begin{aligned} &0.05 \leq x_1 \leq 2,\\ &0.25 \leq x_2 \leq 1.3,\\ &2 \leq x_3 \leq 15. \end{aligned} \tag{8}$$

El código en lenguaje M del TS-MBFOA fue ejecutado en una computadora con las características: 4GB RAM, procesador de 2.3Ghz y sistema operativo Windows de 64bit; el entorno Matlab R2018b bajo la licencia 40693035 y el GNU Octave versión 6.

El TS-MBFOA se configuró con 30 ejecuciones independientes y la calibración de los parámetros fue: bacterias (S_b) = 50, tamaño de paso (R) = 0.001, factor de escalamiento (β) = 1.9, ciclo quimiotáxico (N_c) = 40, bacterias a reproducir (S_r) = 1, frecuencias de reproducción ($RepCycle$) = 100 y número de evaluaciones = 30,000.

Los resultados obtenidos a la minimización del problema se presenta en las Tablas 2 y 3.

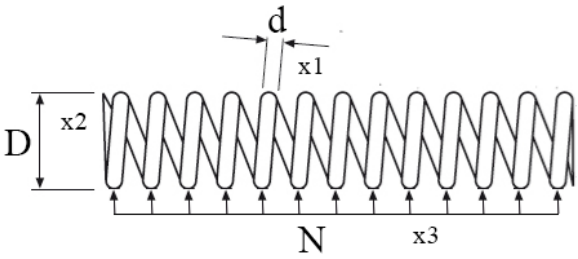


Fig. 3. Resorte de Tensión/Compresión.

Tabla 2. Resultados de TS-MBFOA ejecutado en Matlab para el problema del resorte.

Ejecución	x_1	x_2	x_3	**Función Objetivo**
1	0.051789696	0.359076172	11.1687731	0.012682902
2	0.051408726	0.350006599	11.69993827	0.012672682
3	0.051323133	0.347948886	11.83711678	0.012681989
4	0.05169103	0.356454463	11.31628262	0.012682867
5	0.051551231	0.352828141	11.54267337	0.012698303
6	0.050968603	0.339503268	12.38304812	0.012685288
7	0.051656821	0.355695836	11.35835672	0.012679063
8	0.051116893	0.343009175	12.15497521	0.012686556
9	0.052276436	0.370927952	10.52020387	0.012691499
10	0.051539769	0.352875798	11.53039952	0.012682867
11	0.052027796	0.364794473	10.83824442	0.012677241
12	0.052621549	0.379460676	10.07431931	0.012686934
13	0.051119373	0.343141925	12.14543596	0.012684144
14	0.051382235	0.349344232	11.74137683	0.012673886
15	0.051859118	0.360791058	11.06753926	0.012679433
16	0.051595774	0.354339953	11.44481574	0.012682452
17	0.051174813	0.344460045	12.0487119	0.012673247
18	0.051311125	0.347679802	11.84412994	0.012672672
19	0.05275138	0.382662497	9.923899948	0.012697022
20	0.051110004	0.342691795	12.17608431	0.012690298
21	0.051225268	0.345619307	11.97511887	0.012674242
22	0.052603802	0.379041853	10.0937464	0.012684761
23	0.051642013	0.355479518	11.37752278	0.012682259
24	0.05193514	0.362645388	10.97988345	0.012696253
25	0.051694988	0.356852268	11.28483768	0.012668978
26	0.052128581	0.367359078	10.69530642	0.012673185
27	0.051164621	0.344221262	12.0646512	0.01267378
28	0.052598746	0.378968032	10.1001054	0.01268652
29	0.051280044	0.346939455	11.89501968	0.0126768
30	0.05186574	0.36094871	11.06059164	0.012681467

La comparación de los resultados obtenidos de las 30 ejecuciones independientes del algoritmo determina que ambas ejecuciones encuentran resultados óptimos y

Tabla 3. Resultados de TS-MBFOA ejecutado en Octave para el problema del resorte.

Ejecución	x_1	x_2	x_3	**Función Objetivo**
1	0.051236884	0.345914002	11.95354752	0.012671213
2	0.05167156	0.356096572	11.34346759	0.012686436
3	0.051163989	0.343987178	12.08848084	0.012686307
4	0.051828296	0.360063954	11.1034556	0.012673581
5	0.051562802	0.353639641	11.48479737	0.012678807
6	0.05210742	0.366577403	10.7485808	0.012688979
7	0.051626443	0.355093061	11.3907663	0.012673367
8	0.051757885	0.358271685	11.2029732	0.01267177
9	0.052283211	0.370773217	10.52561293	0.012694975
10	0.051687167	0.356496828	11.31148198	0.012677906
11	0.05179902	0.359258975	11.14740871	0.012673334
12	0.051452182	0.350926475	11.64531196	0.012676729
13	0.051909759	0.361838903	11.00347399	0.01267864
14	0.051414867	0.349886947	11.71274694	0.012683223
15	0.051435782	0.35051317	11.66883685	0.012675544
16	0.051457705	0.351081024	11.63494457	0.012675397
17	0.05176511	0.358435112	11.20035225	0.012678572
18	0.051909904	0.361894529	11.00566901	0.012682801
19	0.051645052	0.355580688	11.37507666	0.012685041
20	0.050965892	0.339442726	12.38383121	0.012682368
21	0.051179829	0.344543074	12.04574631	0.01267611
22	0.05204823	0.365353585	10.80359713	0.012672354
23	0.052494637	0.376357715	10.23368654	0.01268785
24	0.05213442	0.36725143	10.71944589	0.012696405
25	0.050833354	0.336478217	12.58991746	0.012685492
26	0.051955975	0.362936516	10.95593073	0.012693175
27	0.051849943	0.360503843	11.07555731	0.012672627
28	0.05242227	0.374463276	10.33320982	0.012691618
29	0.051170897	0.344192341	12.07537644	0.012685491
30	0.050757896	0.33464182	12.71671054	0.012688146

Tabla 4. Resultados de ejecución de GNU Octave y Matlab.

Herramienta	**GNU Octave**	**Matlab**
Modificación del código	No	No
Interfaz gráfica	No	Sí
Ejecuciones	30	30
Tiempo en segundos	$9,120$	78

aproximados al mejor valor conocido dentro la literatura especializada para el problema de optimización utilizado, el cual es 0.0126.

Cabe mencionar que en ambos lenguajes de programación se obtiene las estadísticas básicas sin la modificación de código fuente del algoritmo codificado en scripts m. Sin embargo, una de las desventajas de ser ejecutada en Octave es el tiempo de ejecución (ver Tabla 4) y que no tiene la capacidad para desarrollar una interfaz de usuario.

En la Tabla 5 se presenta un comparativo de los resultados de las estadísticas básicas

Tabla 5. Comparación de estadísticas básicas del algoritmo TS-MBFOA en Matlab y Octave.

Criterios	TS-MBFOA en Matlab	TS-MBFOA en Octave
Mejor	0.012668978283364	0.0126712133127548
Media	0.012681986380528	0.01268147532895791
Mediana	0.012682355678653	0.01268058712340608
Desviación estándar	7.659412902293769e-06	7.427784107212482e-06
Peor	0.012698303467978	0.01269640528719985

del TS-MBFOA al ser ejecutado en los dos entornos de desarrollo. Los resultados de ambas ejecuciones son generados con los mismos parámetros.

De acuerdo a los valores de la Tabla 5, TS-MBFOA es un algoritmo que mejores resultados obtiene al ser ejecutado en ambas plataformas. Matlab tiene mejor racha de tiempo cumpliendo las 30 ejecuciones en 1 minuto con 18 segundos. Al ejecutar TS-MBFOA en GNU Octave el tiempo de ejecución de 30 ejecuciones independientes fue de 2 horas con 53 minutos, tiempo significativo que implica un atraso para la obtención de los resultados, pero con la ventaja de ser lenguaje de programación libre.

Los resultados se obtuvieron con la misma configuración sin la necesidad de hacer modificaciones en el código, además de obtener resultados óptimos aproximados como lo muestra la media y desviación estándar de la Tabla 5.

En general, los resultados de la ejecución de GNU Octave son competitivos, pero se requiere de un mayor tiempo de ejecución.

De acuerdo a la prueba no paramétrica de *Wilcoxon signed rank test* con un 95 % de confianza, no existe una diferencia significativa entre los resultados de TS-MBFOA ejecutado en Matlab y Octave con un p-value de 0.68916, tomando como valores para la comparación la mejor solución obtenida de cada una de las 30 ejecuciones independientes encontrada por el TS-MBFOA (ver Tablas 2 y 3).

5. Conclusiones

En este artículo el algoritmo basado en el forrajeo de bacterias TS-MBFOA fue implementado en lenguaje M y ejecutado en los entornos de desarrollo Matlab y Octave con el objetivo de comparar el rendimiento del algoritmo en un problema de optimización númerica con restricciones de diseño ingenieríl conocido como el Resorte de tensión/compresión. Además de dar a conocer a la comunidad científica las ventajas y desventajas de usar estos entornos en algoritmos bio-inspirados.

El TS-MBFOA se ejecutó en ambos entornos sin la necesidad de modificar el código fuente del algoritmo. 30 ejecuciones independientes fueron realizadas, donde la principal desventaja de Octave fue el tiempo de ejecución con una diferencia de 2 horas y 53 minutos en comparación con Matlab con tan solo uso 1 minuto con 18 segundos.

A diferencia de Matlab, Octave es un software libre y ejecuta cualquier código generado en scripts .m, excepto programación de interfaces gráficas. En los experimentos realizados, Matlab obtuvo soluciones mucho más rápido, lo cual es una gran ventaja para la toma de decisiones del usuario final ante la solución de problemas complejos de la vida real.

La calidad de los resultados, obtenidos por ambas implementaciones, son competitivas.

De acuerdo a lo presentado en las tablas de estadísticas básicas y la prueba no paramétrica de *Wilconxon Signed Rank Test*, no existe una diferencia significativa entre los resultados de TS-MBFOA ejecutado en Matlab y Octave.

La comunidad científica puede hacer uso de Matlab y Octave en la implementación y ejecución de algoritmos bio-inspirados para la solución de problemas complejos. La codificación en lenguaje M es aceptable en ambos entornos, los resultados obtenidos tienen la misma calidad, sin embargo, el tiempo de ejecución es más elevado en Octave, aunque, Matlab es un software comercial bajo licencia costosa.

Como trabajo futuro, se espera implementar a TS-MBFOA en lenguage Java, el cual es Open source, y analizar el rendimiento del algoritmo en el software que ejecute dicho lenguaje. Además, de probar el algoritmo en un conjunto de pruebas más amplio para obtener conjeturas o inferencias más profundas del rendimiento de este algoritmo en diversos lenguajes de programación.

Referencias

1. Companioni-Guerra, A., Cuesta-Llanes, E., Hernández-Blanco, Y., Orovio-Cobo, V., Días-Ramos, S.: Entorno integrado para el trabajo con GNU/Octave. Revista Cubana de Ciencias Informáticas, vol. 6, no. 4, pp. 1–8 (2012)
2. Engelbrecht, A. P.: Fundamentals of computational swarm intelligence. John Wiley & Sons (2005)
3. Esqueda-Elizondo, J. J.: Matlab e interfaces gráficas. CONATEC 2002: Instituto Tecnológico de Ciudad Madero (2002)
4. Fogel, D. B.: An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks, vol. 5, no. 1, pp. 3–14 (1994) doi: 10.1109/72.265956
5. Hernández-Ocaña, B., Pozos-Parra, P., Mezura-Montes, E., Portilla-Flores, E. A., Vega-Alvarado, E., Calva-Yáñez, M. B.: Two-swim operators in the modified bacterial foraging algorithm for the optimal synthesis of four-bar mechanisms. Computational Intelligence and Neuroscience, no. 17, pp. 1–18 (2016) doi: 10.1155/2016/4525294
6. Hernández-Ocaña, B., Chávez-Bosquez, O., Hernández-Torruco, J., Canul-Reich, J., Pozos-Parra, P.: Bacterial foraging optimization algorithm for menu planning. IEEE Access, vol. 6, pp. 8619–8629 (2018) doi: 10.1109/ACCESS.2018.2794198
7. Hernández-Ocana, B., Pozos-Parra, P., Mezura-Montes, E.: Improved modified bacterial foraging optimization algorithm to solve constrained numerical optimization problems. Applied Mathematics and Information Sciences, vol. 10, no. 2, pp. 607–622 (2016) doi: 10.18576/amis/100220
8. Herrera, F.: Introducción a los algoritmos metaheurísticos. Ciencias de la Computación e IA (2006)
9. Kasaiezadeh, A., Khajepour, A., Waslander, S. L.: Spiral bacterial foraging optimization method: Algorithm, evaluation and convergence analysis. Engineering Optimization, vol. 46, no. 4, pp. 439–464 (2014) doi: 10.1080/0305215X.2013.776550
10. Mezura-Montes, E., Cetina-Domínguez, O., Hernández-Ocaña, B.: Nuevas heurísticas inspiradas en la naturaleza para optimización numérica (2010)
11. Mezura-Montes, E., Hernández-Ocaña, B.: Bacterial foraging for engineering design problems: Preliminary results. In: Memorias del 4o Congreso Nacional de Computación Evolutiva (COMCEV'2008) (2008)
12. Nogueras, G. B.: Octave: Una alternativa real a Matlab a coste cero. Fluid Dynamics Group, UPM, pp. 1–7 (2007)

13. Passino, K. M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, vol. 22, no. 3, pp. 52–67 (2002) doi: 10.1109/MCS.2002.1004010
14. Suarez, A. O.: Una aproximación a la heurística y metaheurísticas. INGE@ UAN-Tendencias en la Ingeniería, vol. 1, no. 2 (2013)

ISSN 1870-4069

COMPOSAIN: Aplicación de un algoritmo genético para el diseño de contrapuntos musicales sin restricción de integridad de figuras

Natan Vilchis-Tavera, Adriana Lara

Instituto Politécnico Nacional,
Escuela Superior de Física y Matemáticas,
México

nvilchist1300@alumno.ipn.mx,
alaral@ipn.mx

Resumen. Durante las últimas décadas los algoritmos evolutivos han generado con éxito soluciones más allá de la intuición humana; es por esto que resulta interesante aplicarlos en ámbitos donde la creatividad y genialidad juegan un papel importante. En este trabajo se presenta la aplicación de un algoritmo genético operando sobre melodías musicales. El problema particular que se aborda es el de generar un contrapunto musical adecuado para una cierta melodía existente. A diferencia de otros trabajos disponibles en la literatura, nuestra propuesta maneja una línea melódica diversa, es decir que permite codificar las notas con libertad en la duración de los sonidos y silencios. Abordamos aquí la tarea de la creación de frases musicales como un problema de optimización, y extrayendo características basadas en teoría musical diseñamos criterios matemáticos que funcionan como guía para el algoritmo genético. Nuestra propuesta fue probada en cuatro melodías, arrojando conclusiones interesantes y rutas prometedoras para futura investigación. Consideramos que este tipo de enfoque puede ser útil como ayuda para la composición musical tradicional asistida por un algoritmo inteligente. Las melodías resultantes de este trabajo pueden consultarse en los enlaces dados en el texto.

Palabras clave: Contrapunto musical, algoritmo genético, composición automática, optimización, composición musical.

COMPOSAIN: A genetic Algorithm Application for the Design of Musical Counterpoints without Figures Length Limitation

Abstract. Over the past decades, evolutionary algorithms have successfully generated solutions beyond human intuition; This is why it is interesting to apply them in areas where creativity and genius play an essential role. In this work, the application of a genetic algorithm operating on musical melodies is presented.

The tackled problem here is the design of a suitable musical it counterpoint for a particular existing melody. Unlike other previous works, our proposal handles a diverse melodic line. It allows the notes to be freely encoded regarding the length of the sounds and silences. We approach the task of creating musical phrases as an optimization problem. By extracting characteristics based on musical theory, we design mathematical criteria that guide the genetic algorithm. Our proposal was tested on four melodies, yielding interesting conclusions and promising routes for future research. We believe that this type of approach can be helpful as an aid to traditional music composition assisted by an intelligent algorithm. The melodies resulting from this work are available by the links given in the text.

Keywords: Musical counterpoint, genetic algorithm, automatic composition, optimization, musical composition.

1. Introducción

Los algoritmos genéticos son una herramienta poderosa que ha tenido éxito en diversas áreas del quehacer humano. En particular, en la generación automática de melodías, podemos encontrar sistemas de improvisación de jazz como GenJam [2, 14] o los sistemas propuesto por Dragan [12] y Garay [6], que generan melodías musicales a manera de primera voz. Estos trabajos han abierto exitosamente la discusión sobre si la creatividad musical es exclusiva del ser humano.

El presente trabajo propone dar al usuario un sistema que tiene como base un algoritmo genético para encontrar la mejor melodía que actúe como segunda voz para una melodía dada llamada *cantus firmus* [3] usando las reglas de contrapunto; es decir, COMPOSAIN permite encontrar la mejor segunda voz de contrapunto tal que se escuche agradable al interpretarse simultáneamente con la melodía dada por el usuario.

El concepto de segunda voz es esencial en la composición y la producción musical, pues para hacer ensambles con diversos instrumentos, es necesario que cada melodía pueda acompañar de la mejor manera a una melodía principal ya establecida, sin ser necesariamente la misma en estructura. Esta es la base de la polifonía.

Un pilar de la teoría musical del contrapunto son las reglas (del contrapunto) de Fux [10]. Existen propuestas, basadas en estas reglas, para generar automáticamente un contrapunto, pero en la literatura esto se reporta únicamente para el trabajo de *cantus firmus* que contienen únicamente notas redondas (𝅝) (e.g., Optimuse [8], Foox [15], Counterpoint Composer [19] y EVOC [18]).

En contraste, en el presente trabajo, consideramos las reglas de Fux en un sentido más amplio y abordamos el problema de generar un contrapunto musical general a partir de un *cantus firmus* rítmicamente libre; es decir, considerando que el *cantus firmus* pueda tener figuras musicales de diferente duración, por ejemplo 𝅗𝅥, ♩, ♩., entre otras y no solamente notas redondas como se ha hecho de forma clásica en la literatura. Para hacer esto posible, hemos definido una serie de criterios matemáticos para adaptar las reglas de Fux como funciones para guiar la búsqueda en un algoritmo genético.

Uno de los retos a la hora de desarrollar este trabajo fue diseñar una función objetivo capaz de producir una melodía que actúe como contrapunto a partir de la información

Tabla 1. Valores de noteToNumbered.

Nota musical	C	$C\sharp$	D	$D\sharp$	E	F	$F\sharp$	G	$G\sharp$	A	$A\sharp$	B
Nota musical numerada	0	1	2	3	4	5	6	7	8	9	10	11

de calidad sonora de *cantus firmus* (i.e., armonía, intervalos musicales, rango de la solución, escala deseada, entre otros).

En la siguiente sección, presentamos conceptos necesarios para entender la Sección 3, donde presentamos la creación del contrapunto como un problema de optimización; el enfoque algorítmico usado se encuentra en la Sección 4. Los resultados se presentan en la Sección 5 (y archivos complementarios). Finalmente, exponemos las conclusiones y recomendaciones para posibles caminos de investigación como trabajo futuro.

2. Conceptos básicos

En esta sección, establecemos la terminología y conceptos necesarios para comprender el resto de este trabajo.

Definición 1 (Nota musical) Una *nota musical* w será un elemento del conjunto W definido por $W := \{C, C\sharp, D, D\sharp, E, F, F\sharp, G, G\sharp, A, A\sharp, B\}$. La correspondiente *nota musical numerada* de w está dada por la función:

$$noteToNumbered : W \to \mathbb{Z}.$$

Descrita en la tabla 1.

Debido a la necesidad de diferenciar entre dos notas musicales de diferente altura[1], usamos la *Notación Científica de Tono* definida de la siguiente manera.

Definición 2 Una nota musical $w_{(i)}$ se considera en *Notación Científica de Tono (NCT)* cuando el subíndice $i \in \mathbb{Z}$ denota la octava correspondiente de w. El conjunto $\mathbb{SPN}$ de notas en NCT es:

$$\mathbb{SPN} := \{\ldots, C_0, C\sharp_0, D_0, D\sharp_0, E_0, F_0, F\sharp_0, G_0, G\sharp_0, A_0, A\sharp_0, B_0, C_1, \ldots\}.$$

Un ejemplo de la relación existente entre una nota musical en NCT y la partitura musical se muestra en la Figura 1, para ello se ha mostrado una pequeña melodía con notas de diferente duración, además se muestra la relación de las notas musicales y las teclas en el piano.

Con estas definiciones, es posible relacionar cada nota en $\mathbb{SPN}$ con un número entero para facilitar la gestión por computadora. Es por ello que se propone la siguiente definición, basada en el protocolo MIDI [1].

[1] La altura musical es la cualidad del sonido que determina si el sonido es grave o agudo [11].

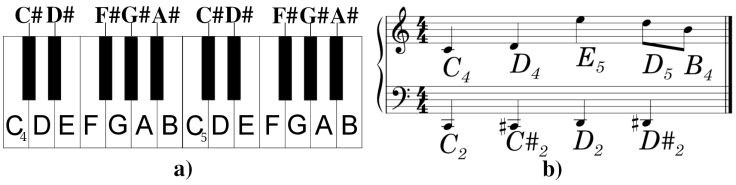


Fig. 1. a) Notas musicales en el piano. b) Ejemplo de correspondencia entre notas musicales en NCT y la partitura.

Definición 3 (Nota MIDI) Sea $w_{(i)} \in \mathbb{SPN}$, la *nota MIDI* correspondiente a $w_{(i)}$ está determinada por la función $SPNtoMIDI : \mathbb{SPN} \to \mathbb{Z}$, donde:

$$SPNtoMIDI(w_{(i)}) := noteToNumbered(w) + 12(i+1).$$

Definición 4 (Transformar nota MIDI a una *nota musical numerada*) Sea x una nota MIDI, la nota musical numerada correspondiente a x está determinada por la función $MIDIToNumbered : \mathbb{Z} \to \mathbb{Z}$, donde:

$$MIDIToNumbered(x) = x \ mod \ 12.$$

Para ejemplificar estas definiciones tomemos la nota musical A_0. La nota MIDI correspondiente es 21 ($noteToNumbered(A) + 12(0+1) = 9 + 12 = 21$). Ahora bien, la nota musical numerada correspondiente a esta nota MIDI es 9 ($21 \bmod 12 = 9$), la cual corresponde a la nota musical A de la Tabla 1.

Definición 5 (Melodía) El vector $x = (x_1, \ldots, x_n) \in \mathbb{Z}^n$ se dice ser una *melodía*, si cada x_i es una nota MIDI, para $i = 1, \ldots, n$.

Definición 6 (Escala musical numerada) Un subconjunto $S \subseteq W$ se nombra como escala musical. La *escala musical numerada* correspondiente a S se define como el conjunto:

$$nToN_{(S)} = \{z \in \mathbb{Z} \mid \ z = noteToNumbered(n), \text{ para cada } n \in S \ \}.$$

La escala musical numerada es útil porque el usuario puede decidir qué escala utilizar para el contrapunto. Para ejemplificar esta definición tomemos la escala de C mayor[2], con ayuda de la Definición 6 se obtiene el conjunto de notas musicales numeradas $\{0, 2, 4, 5, 7, 9, 11\}$. Para definir la función objetivo se utiliza la siguiente proporción, que se deriva de una media para poder evaluar determinadas características como el movimiento de la melodía y la proporción de intervalos musicales en la misma. La *proporción S-W-N* se describe a continuación.

Definición 7 (Proporción S-W-N) Sea $s \in \mathbb{N} \cup \{0\}, w \in [0,1], n \in \mathbb{N}$. La proporción S-W-N está dada por:

$$H(s,w,n) = \begin{cases} 1 & \text{si } \ s \leq w \cdot n, \\ -1 & \text{en caso contrario.} \end{cases}$$

[2] La escala de C mayor es el subconjunto de notas musicales $\{C, D, E, F, G, A, B\}$.

3. Planteamiento del problema

El problema de optimización se describe a continuación:

$$\max_{x \in \mathbb{Z}^n} f(a, b, c, d, v, w, x, y, S').$$

Sujeto a :

- $a, b \in \mathbb{SPN}$. Nota más baja y más alta correspondiente al contrapunto.
- $c, d \in \mathbb{SPN}$. Primera y la última nota deseada para el contrapunto.
- $v \in \mathbb{R}^3, v_i \geq 0, \sum_{i=1}^{3} v_i = 1$. Vector de pesos de movimiento opuesto v_1, movimiento oblicuo v_2 y movimiento directo v_3 entre el *cantus firmus* y el contrapunto.
- $w \in \mathbb{R}^{13}, w_i \geq 0, \sum_{i=1}^{13} w_i = 1$, es el vector de pesos de los intervalos musicales entre el *cantus firmus* y el contrapunto. Los pesos $w_1,\ w_2,\ dots,\ w_{13}$, son: perfecta unísona, segunda menor, segunda mayor, tercera menor, tercera mayor, cuarta perfecta, tritono, quinta perfecta, sexta menor, sexta mayor, séptima menor, séptima mayor, y octava perfecta, respectivamente.
- $y \in \mathbb{Z}^n$, es el vector de notas MIDI, *cantus firmus*.
- $S' = nToN_{(S)}, S \subseteq W$, es la escala numerada que representa la escala musical deseada para el contrapunto.

La función objetivo propuesta se describe mediante:

$$\begin{aligned} f(a, b, c, d, v, w, x, y, S') = {} & h_1(x, a, b) + h_2(x, c) + h_3(x, d) \\ & + h_4(x, y, v) + h_5(x, y, w) \\ & + h_6(x, y, S'), \end{aligned}$$

donde cada término h_i es una función para evaluar una característica musical particular deseada descrita por:

1. h_1: Esta función clasifica el cromosoma basándose en si cada una de sus notas está dentro de un rango establecido por el usuario; esta idea surgió de las reglas de Fux [10]. La función h_1 se ha adaptado para trabajar con melodías de diferentes longitudes.
   La idea principal de la función h_1 es calificar las notas de la melodía y que las notas de la melodía no tengan saltos tan precipitados. Para ello, deben estar dentro de un intervalo determinado por el usuario. Con esto, el usuario es libre de decidir qué rango de notas (graves o agudos) deberán estar en la solución del contrapunto; esto es particularmente útil si desea obtener el contrapunto para una flauta dulce o un bajo:

$$h_1(x, a, b) = \sum_{i=1}^{n} g_1(x_i, a, b), \text{ con:}$$

$$g_1(x_i, a, b) = \begin{cases} 1, \text{ if } SPNToMIDI(a) \leq x_i \leq SPNToMIDI(b), \\ -1, \text{ en otro caso.} \end{cases}$$

2. h_2 y h_3: Estas funciones clasifican un cromosoma en función de si comienza y termina con las notas deseadas por el usuario. Este criterio se toma de las reglas del contrapunto clásico [9, 16] y se adapta para ajustarse a las notas deseadas por el usuario. Ha sido posible que el usuario elija la nota inicial y final del contrapunto (notas clásicas de octava) para proporcionar al músico libertad y un mejor control del contrapunto final:

$$h_2(x, c) = \begin{cases} 1, \text{ si } x_1 = NTIToMIDI(c), \\ -1, \text{ en otro caso.} \end{cases}$$

$$h_3(x, d) = \begin{cases} 1, \text{ si } x_n = NTIToMIDI(d), \\ -1, \text{ en otro caso.} \end{cases}$$

3. h_4: Esta función contempla movimientos oblicuos, directos y opuestos, estas ideas fueron tomadas por las propuestas de Fux [10] y adaptadas para que se puedan trabajar con los pesos elegidos por el usuario. Para la puntuación, la proporción *S-W-N* verifica los movimientos directos, opuestos y oblicuos desde la primera nota hasta la i-ésima nota:

$$h_4(x, y, v) = \sum_{i=1}^{n-1} g_2(i, x_{(i+1)} - x_i, y_{(i+1)} - y_i, v), \text{ con:}$$

$$g_2(i, \Delta x, \Delta y, v) = \begin{cases} H\left(\sum\limits_{\substack{j=1 \\ [y_{(j+1)}-y_j][x_{(j+1)}-x_j]<0}}^{i} 1, v_1, n\right), \text{ si } \Delta x \Delta y < 0, \\ H\left(\sum\limits_{\substack{j=1 \\ [y_{(j+1)}-y_j][x_{(j+1)}-x_j]=0}}^{i} 1, v_2, n\right), \text{ si } \Delta x \Delta y = 0, \\ H\left(\sum\limits_{\substack{j=1 \\ [y_{(j+1)}-y_j][x_{(j+1)}-x_j]>0}}^{i} 1, v_3, n\right), \text{ si } \Delta x \Delta y > 0. \end{cases}$$

4. h_5: Esta función califica el posible contrapunto en función de la proporción de intervalos musicales que actúan entre el *cantus firmus* y el contrapunto. Esta idea se deriva clásicamente de las reglas de Fux [10], porque debe estar presente una proporción adecuada a lo largo de la melodía del contrapunto:

$$h_5(x, y, w) = \sum_{i=1}^{n} g_3\left(i, |x_i - y_i|, w\right), \text{ con:}$$

Algoritmo 1 Algoritmo genético.

1: Inicializar P una población aleatoria de n individuos.
2: establecer $generation \leftarrow 0$
3: **while** $generation < maxGenerations$ **do**
4: Hacer *selección* de la población P.
5: Hacer *cruza* de los individuos seleccionados con una probabilidad p_c.
6: Hacer *mutación* para cada individuo con una probabilidad p_m.
7: Realizar *elitismo*.
8: Hacer $generation \leftarrow generation + 1$
9: **end while**

$$g_3(i, \Delta x, w) = \begin{cases} H\left(\sum\limits_{\substack{j=1 \\ x_j - y_j = 0}}^{i} 1, w_1, n\right), & \text{si } \Delta x = 0, \\ H\left(\sum\limits_{\substack{j=1 \\ |x_j - y_j| \bmod 12 = 0}}^{i} 1, w_{13}, n\right), & \text{si } \Delta x \bmod 12 = 0, \Delta x \neq 0, \\ H\left(\sum\limits_{\substack{j=1 \\ |x_j - y_j| \bmod 12 = \Delta x \bmod 12}}^{i} 1, w_{(1+[\Delta x \bmod 12])}, n\right), & \text{en otro caso.} \end{cases}$$

5. h_6: Esta función califica el cromosoma basándose en si cada una de sus notas está dentro de una escala dada. Se prefieren consonancias sobre las disonancias. Derivado de las reglas de Fux y la teoría del contrapunto [9, 10]:

$$h_6(x, y, S') = \sum_{i=1}^{n} g_4\left(MIDIToNumbered(x_i), MIDIToNumbered(y_i), S'\right), \text{ con:}$$

$$g_4(p, q, S') = \begin{cases} 1 \text{ si } (p \in S' \wedge q \in S') \vee (p \notin S' \wedge q \notin S'), \\ -1 \text{ en otro caso.} \end{cases}$$

4. Algoritmo propuesto

El algoritmo 1 describe el algoritmo genético propuesto utilizado en este trabajo, donde:

– Los individuos son melodías (definición 5) del tamaño del *cantus firmus*.
– Selección se hizo por torneo con tres rondas como se describe en [4].
– Cruza de un punto (dos hijos por cada par de padres).
– Elitismo se aplica de manera estándar.
– Mutación sólo afecta a una sola nota del individuo.

El proceso para obtener el contrapunto usando COMPOSAIN dado el *cantus firmus* se ilustra en la Figura 2, la entrada consiste de un archivo MIDI [17] conteniendo el *cantus firmus*.

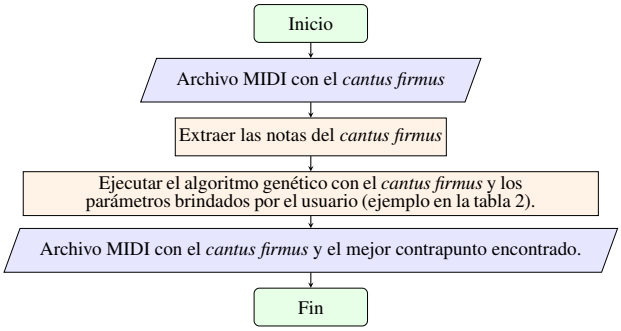


Fig. 2. Esquema general de COMPOSAIN.

Cabe señalar que se ha considerado que la función objetivo f permita definir la aptitud para los contrapuntos durante la búsqueda evolutiva. La aptitud considera el valor de la función objetivo para el *cantus firmus* y el contrapunto dado. Los individuos fueron representados como vectores de números, para esto nos inspiramos en el proyecto Foox [15] y el algoritmo genético desarrollado por Dragan [12].

La valoración del cromosoma según la cantidad que actúa en los intervalos musicales es una idea que proviene de [12], que ha sido adaptada para que a partir de un vector de pesos proporcionado por el usuario, sea posible determinar la proporción deseada en cada uno de los los intervalos musicales presentes en toda la melodía del contrapunto.

También calificamos el cromosoma usando un sistema de recompensa y castigo como se utiliza en *Foox* [15] (es decir, la función objetivo asigna una puntuación positiva (+1) si cumple con ciertas características; de lo contrario, asigna una puntuación negativa (-1)), para este caso, tal idea ha sido utilizada plenamente en la función objetivo de COMPOSAIN; por ejemplo, para la función h_1 de la función objetivo, se contará un punto positivo (+1) si la nota x_i está dentro del intervalo deseado por el usuario, mientras que si la nota x_i está fuera del intervalo, se contabilizará un punto negativo (-1) en la función objetivo, de esta forma, la función objetivo es una suma de puntuaciones donde el mejor contrapunto es el que tiene la puntuación más alta.

5. Resultados

El algoritmo propuesto se implementó en el lenguaje C++ y se ejecutó en un procesador Intel Core i7-4720HQ, con 16 GB en RAM. Presentamos aquí los resultado obtenidos para cuatro melodías elegidas como *cantus firmus*. Para efectos de que los resultados puedan ser replicados, el generador de números pseudoaleatorios utilizado en los operadores del algoritmo genético fue Mersenne twister [13]. A continuación se muestra una visualización del proceso de optimización para cada uno de ellos, junto con observaciones relevantes sobre los resultados[3].

[3] Para escuchar los resultados considerados en este trabajo, el lector puede acceder a https: //natanvilchis.org/composain/

Tabla 2. Parámetros utilizados para *Andante grazioso*.

Parámetro	**Valor**
(n, G, p_c, p_m)	$(100, 50000, 0{,}99, 0{,}07)$
(a, b, c, d)	(E_2, D_4, A_3, A_2)
v	$\left(\frac{6}{35}\ \frac{6}{35}\ \frac{23}{35}\right)^T$
w	$(0, \frac{1}{36}, 0, \frac{20}{36}, \frac{8}{36}, \frac{1}{36}, 0, \frac{1}{36}, 0, \frac{1}{36}, \frac{1}{36}, \frac{1}{36}, \frac{2}{36})^T$
S'	$\{1, 2, 4, 6, 8, 9, 11\}$

Para cada uno de los resultados se mostrará el gráfico de 50 ejecuciones independientes del algoritmo, con la finalidad de medir el desempeño del mismo con diferentes semillas aleatorias.

5.1. Andante Grazioso de W. A. Mozart (1756-1791)

En la figura 3 se muestra el proceso de maximización para *Andante Grazioso*, el valor óptimo para los parámetros proporcionados en la tabla 2 resultó de 145, que se obtiene usando la semilla 7827 para 50,000 generaciones. La solución de contrapunto resultante suena muy similar al contrapunto original compuesto por humanos.

Cabe destacar que el valor de la función objetivo del contrapunto realizado por Mozart es de 145, el mismo que el obtenido por COMPOSAIN. Por otro lado, en la melodía obtenida se controlan los saltos que realiza el contrapunto, es decir, da como resultado una melodía tranquila que no ensombrece el *cantus firmus* cuando escuchan simultáneamente. Finalmente, existe un movimiento contrario en las dos últimas notas del contrapunto que resalta la última parte del *cantus firmus* porque el contrapunto termina en notas graves.

Los parámetros utilizados se mencionan en la tabla 2. Como se muestra en la figura 3, la mediana de los datos y el máximo están en 145; además, hay una mayor cantidad de datos con un valor de aptitud en torno a 145. Se concluye con esto que la mayoría de las ejecuciones en las 50,000 generaciones obtienen una puntuación de 145.

5.2. If You're Happy and You Know It

El proceso de maximización de *If You're Happy and You Know It* se muestra en la figura 4, donde la mayoría de las ejecuciones permanecen en 189 para el valor de la función y solo hay dos valores atípicos. El valor óptimo para los parámetros proporcionados en la tabla 3 es 189, que se obtiene usando la semilla 11 para 25,000 generaciones.

El contrapunto óptimo obtenido realiza una melodía que juega entre notas bajas y altas; por ejemplo, para las últimas ocho notas, el contrapunto tiene dos partes: una parte con notas más altas y otra con notas más bajas, lo que permite tocar octavas con *cantus firmus*.

5.3. Oda a la alegría de L. Van Beethoven (1770-1827)

El proceso de maximización de *Ode to Joy* se muestra en la figura 5.

Tabla 3. Parámetros utilizados para *If you're happy and you know it*.

Parámetro	Valor
(n, G, p_c, p_m)	$(100, 25000, 0{,}99, 0{,}05)$
(a, b, c, d)	(C_4, B_6, C_4, C_6)
v	$\left(\frac{11}{48}\ \frac{26}{48}\ \frac{11}{48}\right)^T$
w	$(0, 0{,}027777, 0, 0{,}555561, 0{,}222222, 0{,}027777, 0, 0{,}027777,$ $0, 0{,}027777, 0{,}027777, 0{,}027777, 0{,}055555)^T$
S'	{0, 2, 4, 5, 7, 9, 11 }

Tabla 4. Parámetros utilizados para *Ode to Joy*.

Parámetro	Valor
(n, G, p_c, p_m)	$(100, 50000, 0{,}99, 0{,}05)$
(a, b, c, d)	(E_2, D_4, A_3, C_3)
v	$\left(0{,}171429\ 0{,}171429\ 0{,}657142\right)^T$
w	$(0, 0{,}0277778, 0, 0{,}5555555, 0{,}2222222, 0{,}0277778, 0,$ $0{,}0277778, 0, 0{,}0277778, 0{,}0277778, 0{,}0277778, 0{,}0555555)^T$
S'	{0, 2, 4, 5, 7, 9, 11 }

Tabla 5. Parámetros utilizados para *The first Noel*.

Parámetros	Valores
(n, G, p_c, p_m)	$(100, 15000, 0{,}99, 0{,}05)$
(a, b, c, d)	(E_2, D_4, A_3, D_3)
v	$\left(0{,}171429\ 0{,}171429\ 0{,}657142\right)^T$
w	$(0, 0{,}0277778, 0, 0{,}5555555, 0{,}2222222, 0{,}0277778, 0,$ $0{,}0277778, 0, 0{,}0277778, 0{,}0277778, 0{,}0277778, 0{,}0555555)^T$
S'	{1, 2, 4, 6, 7, 9, 11 }

El valor óptimo para los parámetros proporcionados en la tabla 4 es de 233, que se obtiene usando la semilla 2892 para 50,000 generaciones. El contrapunto resultante ejecuta una melodía que juega entre notas bajas y altas; por ejemplo, para las últimas ocho notas, el contrapunto tiene dos partes: una parte con notas más altas y otra con notas más bajas que permiten tocar octavas con *cantus firmus*.

Vale la pena notar que la mayoría de las ejecuciones mostradas en la figura 5 obtienen 233 para el valor de la función y solo hay un valor atípico. De esta forma, se ha comprobado que la mayoría de ejecuciones alcanzaron el valor óptimo.

5.4. The First Noel

El proceso de maximización de *The First Noel* se muestra en la figura 6, donde la mayoría de las ejecuciones llegan a 187 como valor de la función objetivo y solo hay dos valores atípicos. El valor óptimo para los parámetros proporcionados en la tabla 5 es 187, que se obtiene usando la semilla 28 para 15,000 generaciones.

Este caso de prueba nos muestra resultados interesantes. Uno de ellos es que determinadas notas graves están presentes a lo largo de la melodía, por lo que el contrapunto podría adaptarse a instrumentos como un bajo.

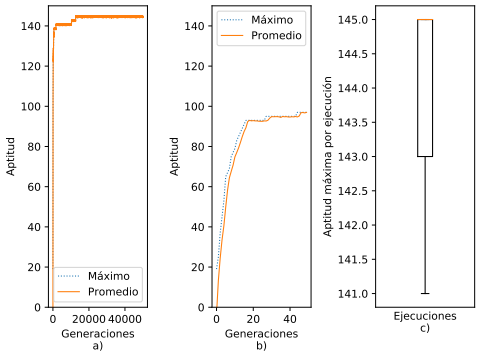


Fig. 3. *Andante grazioso* a) Aptitud máxima y aptitud media de la población por generación. b) Detalle del desempeño del algoritmo en las primeras iteraciones. c) Resultados de 50 ejecuciones independientes.

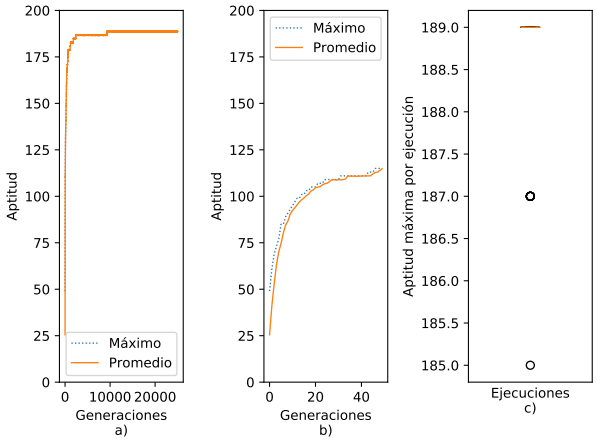


Fig. 4. *If you're happy and you know it* a) Aptitud máxima y aptitud media de la población por generación. b) Detalle del desempeño del algoritmo en las primeras iteraciones. c) Resultados de 50 ejecuciones independientes.

Por ejemplo, para el compás 10 (aproximadamente en el medio de la melodía), la ejecución del bajo es más acentuada que el contrapunto. Por otro lado, la forma en que el contrapunto y el *cantus firmus* realzan toda la canción es agradable de escuchar.

A continuación se muestran las tablas con los parámetros utilizados para cada pieza, donde n es el número de individuos, p_c es la probabilidad de cruza, p_m es la probabilidad de mutación y G es el número de generaciones.

6. Conclusiones y trabajo a futuro

En este trabajo abordamos el problema de generar un contrapunto musical, a partir de un determinado *cantus firmus*.

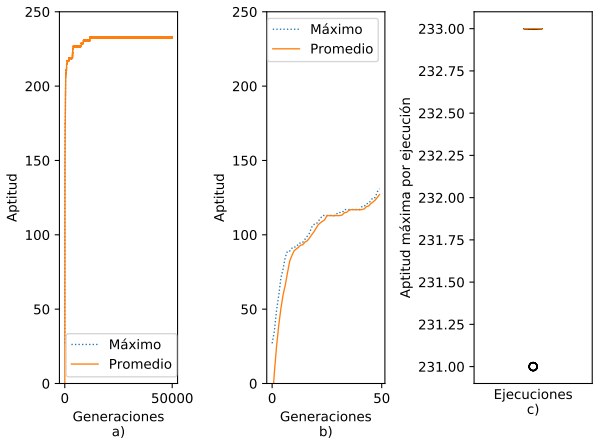


Fig. 5. *Ode to Joy* a) Aptitud máxima y aptitud media de la población por generación. b) Detalle del desempeño del algoritmo en las primeras iteraciones. c) Resultados de 50 ejecuciones independientes.

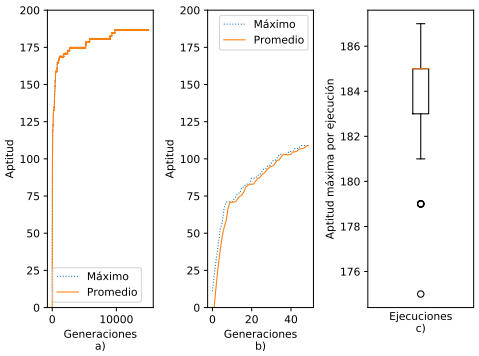


Fig. 6. *The first Noel* a) Aptitud máxima y aptitud media de la población por generación. b) Detalle del desempeño del algoritmo en las primeras iteraciones. c) Resultados de 50 ejecuciones independientes.

A diferencia de otros se consideró una figura musical de duración variable. Basados en teoría musical, planteamos algunos criterios matemáticos para adaptar las reglas de Fux como funciones para guiar la búsqueda en un algoritmo genético y generar automáticamente una melodía que pudiera escucharse en consonancia con la melodía de un usuario.

Se presentaron los resultados obtenidos para cuatro melodías de prueba. Proporcionamos un enlace web a videos para evaluar los resultados audibles donde mostramos que este trabajo tiene potencial como ayuda en la composición musical. Los resultados obtenidos son consistentes con lo que esperábamos en términos del diseño de la melodía: la nota inicial y final está de acuerdo con los parámetros y la proporción de movimientos e intervalos musicales aparece a lo largo de la melodía.

Además, mencionamos que aparece una estructura creativa como solución del algoritmo genético. Una situación interesante es que los contrapuntos obtenidos combinan notas graves y agudas resultando en una solución más versátil. En otros casos se logró igualar resultados de contrapuntos originalmente creados por el humano.

Observamos a través de las ejecuciones independientes que se obtuvieron contrapuntos ligeramente diferentes. Esto tiene sentido ya que es posible a partir de una melodía hacer adornos ligeros sin alterar la estructura principal. Como trabajo adicional, es posible extender la función objetivo propuesta para considerar otros aspectos musicales, como la estructura, los silencios y la diversificación de notas entre el contrapunto (es decir, cuando el contrapunto puede tener un número de notas diferente al del cantus firmus).

La integración de las emociones en el contrapunto (tranquilidad, tristeza, entre otras) es también un camino de investigación prometedor para dar una mayor calidad a los resultados. Para este caso, existen diferentes estudios de las emociones en la música (por ejemplo, la importancia relativa de seis variables [5], o modelo circumplex de emociones [7]), que permite conocer las características específicas que deben tener las obras en para lograr la emoción deseada de la mejor manera. También continuamos tabajando con el planteamiento del problema con un enfoque de optimización multiobjetivo.

Agradecimientos. Proyecto IPN-SIP 20211847, Natan Vilchis agradece el apoyo de la beca Conacyt para estudios de maestría.

Referencias

1. Association of Musical Electronics Industry AMEI and MIDI Manufacturers Association MMA: Universal MIDI packet (UMP) format and MIDI 2.0 protocol, pp. 1–82 (2020)
2. Biles, J. A.: Improvizing with genetic algorithms: Genjam. In: Proceedings of Evolutionary Computer Music, pp. 137–169 (2007) doi: 10.1007/978-1-84628-600-1_7
3. Bloxam, M. J.: Cantus firmus. Oxford University Press (2001) doi: 10.1093/gmo/9781561592630.article.04795
4. Eiben, A. E., Smith, J. E.: Introduction to evolutionary computing. vol. 53, pp. 52,57,84,89 (2003) doi: 10.1007/978-3-662-44874-8
5. Farnsworth, P. R.: The social psychology of music (1958)
6. Garay-Acevedo, A.: Fugue composition with counterpoint melody generation using genetic algorithms. In: Proceedings of International Symposium on Computer Music Modeling and Retrieval, vol. 3310, pp. 96–106 (2005) doi: 10.1007/978-3-540-31807-1_7
7. Gerber, A. J., Posner, J., Gorman, D., Colibazzi, T., Yu, S., Wang, Z., Kangarlu, A., Zhu, H., Russell, J., Peterson, B. S.: An affective circumplex model of neural systems subserving valence, arousal, and cognitive overlay during the appraisal of emotional faces. Neuropsychologia, vol. 46, no. 8, pp. 2129–2139 (2008) doi: 10.1016/j.neuropsychologia.2008.02.032
8. Herremans, D., Sörensen, K.: Composing fifth species counterpoint music with a variable neighborhood search algorithm. Expert systems with applications, vol. 40, no. 16, pp. 6427–6437 (2013) doi: doi.org/10.1016/j.eswa.2013.05.071
9. Jeppesen, K.: Counterpoint: The polyphonic vocal style of the sixteenth century (2013)
10. Mann, A.: The study of counterpoint. vol. 1943 (1971)

11. Martín, I. C.: Iconografía musical infantil. ICONO 14, Revista de Comunicación y Tecnologías Emergentes, vol. 4, no. 1, pp. 1–24 (2006)
12. Matić, D.: A genetic algorithm for composing music. Yugoslav Journal of Operations Research, vol. 20, no. 1 (2010)
13. Matsumoto, M., Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation, vol. 8, no. 1, pp. 3–30 (1998) doi: 10.1145/272991.272995
14. Nam, Y. W., Kim, Y. H.: Automatic jazz melody composition through a learning-based genetic algorithm. In: Proceedings of International Conference on Computational Intelligence in Music, Sound, Art and Design, pp. 217–233 (2019) doi: 10.1007/978-3-030-16667-0_15
15. Nicholas, T.: Foox (2012) https://github.com/ntoll/foox
16. Randall, R.: Class Notes for Counterpoint. pp. 1–3 (2007)
17. Sapp, C. S.: C++ library for parsing standard midi files (2021) https://midifile.sapp.org/
18. Weale, T., Seitzer, J.: Evoc: A music generating system using genetic algorithms. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 1383–1384 (2003)
19. Weaver, J. L.: Counterpoint composer (2021) https://counterpointservice.cfapps.io/

ISSN 1870-4069

Estudio sobre sistemas tutores inteligentes basados en gamificación

Aldo Cisneros, Yasmín Hernández, Alicia Martínez,
Javier Ortiz, Hugo Estrada

TecNM, Centro Nacional de Investigación y Desarrollo Tecnológico,
Cuernavaca, Morelos,
México

{m20ce058, yasmin.hp, alicia.mr, javier.oh,
hugo.ee}@cenidet.tecnm.mx

Resumen. Los Sistemas Tutores Inteligentes representan una de las aplicaciones más exitosas de la inteligencia artificial, debido a su capacidad de adaptación y respuesta a las necesidades particulares de los estudiantes. En los últimos años, se han integrado diversos elementos, tales como las dinámicas de juego, con el fin de tener una interacción más personal y motivante. La gamificación es el uso de elementos de juegos en contextos no lúdicos. La aplicación de la gamificación en la educación es un campo de investigación muy prometedor debido a su capacidad para comprometer a los estudiantes, lo que a su vez permite que aprendan, refuercen el conocimiento, y desarrollen habilidades importantes como la resolución de problemas, la colaboración y la comunicación. Se presenta un análisis de la literatura sobre Sistemas Tutores Inteligentes con gamificación y se propone un Sistema Tutor Inteligente lúdico para enseñar matemáticas a niños de educación básica.

Palabras clave: Ambientes inteligentes de aprendizaje, gamificación en educación, juegos educativos, sistemas tutores inteligentes.

Study on Intelligent Tutor Systems based on Gamification

Abstract. Intelligent tutoring systems are a successful application of artificial intelligence, due to their ability to adapt and respond to needs of students. Recently, several elements have been integrated, such as game elements, to have a more personal and motivating interaction. Gamification is the use of game elements in non-playful contexts. The application of gamification in education is a very promising field of research due to its ability to engage students, which in turn allows students to learn, reinforce knowledge, and develop important skills such as problem solving, collaboration, and communication. An analysis of the literature on intelligent tutoring systems with gamification is presented, also an intelligent tutoring system based on gamification to teach mathematics to elementary school children is proposed.

Keywords: Educational games, gamification in education, intelligent learning environments, intelligent tutoring systems.

1. Introducción

La educación es uno de los sectores con más impacto en el desarrollo de la sociedad, por lo que el desarrollo de tecnologías que apoyen el proceso de enseñanza-aprendizaje recibe mucha atención. Desde que inició la era de la computación se han desarrollado diferentes herramientas de apoyo a la educación; un ejemplo son los Sistemas Tutores Inteligentes, estos sistemas tienen la capacidad de adaptarse y dar respuesta a las necesidades particulares de los estudiantes.

Los Sistemas Tutores Inteligentes (STI) son programas educativos que imitan los patrones de comportamiento de los tutores humanos [23]. Con el objeto de hacer los STI más precisos y adaptivos, se han integrado diversos elementos tales como el reconocimiento de emociones, la personalidad, así como diferentes estados cognitivos como la motivación y la autoeficacia. Estos elementos permiten una interacción más personal y motivante con dichos sistemas.

En los últimos años se han incluido las dinámicas de juego por su probada capacidad para motivar a los estudiantes. La gamificación es una técnica de aprendizaje que traslada los elementos de los juegos al ámbito educativo con el objetivo de conseguir mejores resultados. Su función principal es motivar y comprometer al estudiante con el juego, a través de retos y misiones, con la expectativa de obtener recompensas.

La gamificación se define como el uso de elementos de juegos en contextos no lúdicos [5], tales como: puntos, insignias, niveles, trofeos, retos y misiones, entre otros. La aplicación de la gamificación en la educación es un campo de investigación de reciente nacimiento y promete resultados importantes, ya que ha mostrado capacidad para motivar y enganchar a los estudiantes, lo que deriva en el aprendizaje de estos. A través de la enseñanza lúdica, además de interiorizar el conocimiento, los estudiantes pueden desarrollar habilidades importantes como el pensamiento crítico, la resolución de problemas, la colaboración y la comunicación.

En el sistema educativo mexicano, y particularmente en los niveles básicos, hay diversas materias que tienen un bajo nivel de aprovechamiento por parte de los estudiantes. Tal es el caso de matemáticas, en donde México presenta un rezago importante con respecto al resto de los países que integran la OCDE [1]. Por esta razón, se propone el desarrollo de un Sistema Tutor Inteligente para enseñar matemáticas a estudiantes de 6º de primaria. La planeación educativa para este STI se basa en los libros de texto gratuitos, complementada con el conocimiento de profesores expertos en pedagogía con experiencia enseñando a niños de este nivel educativo.

Adicionalmente, en el diseño del STI se incluyen técnicas de gamificación con el fin de dar una enseñanza motivadora además de adaptiva. En este artículo se hace un análisis de la literatura sobre Sistemas Tutores Inteligentes y Ambientes Inteligentes de Aprendizaje con gamificación, y se presenta una propuesta para desarrollar un Sistema Tutor Inteligente con gamificación para enseñar matemáticas a niños de educación básica. Se presenta el trabajo en curso y resultados preliminares en el diseño y desarrollo del STI.

El resto del artículo se organiza de la siguiente manera: las secciones 2 y 3 describen los STI y la gamificación, respectivamente. La sección 4 hace una revisión de investigaciones sobre el uso de la gamificación con los ambientes de aprendizaje. La sección 5 presenta una propuesta para un STI con gamificación para matemáticas de nivel básico. Finalmente, la sección 6 presenta las conclusiones y el trabajo futuro.

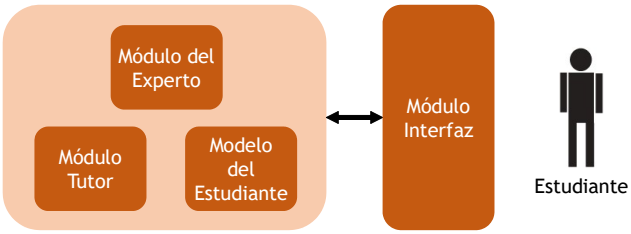


Fig. 1. Arquitectura básica de un Sistema Tutor Inteligente [23].

2. Sistemas tutores inteligentes

Los Sistemas Tutores Inteligentes, han resultado un buen aliado para la educación, y surgen como una alternativa a la enseñanza asistida por computadora. Es un sistema de software que utiliza técnicas de inteligencia artificial para decidir qué y cómo enseñar con base en las características y necesidades de los estudiantes. Los STI simulan los patrones de comportamiento de un tutor humano por lo que son capaces de adaptarse a cada una de las necesidades de los estudiantes.

La principal ventaja de los STI radica en la flexibilidad con la que muestran los contenidos personalizados, ya que no presenta información estática, sino que presentan el conocimiento de manera personalizada y dinámica. Esto se logra recopilando, analizando y comprendiendo cada uno de los datos que se generan a través de la interacción uno a uno entre el estudiante y el STI [19].

La inteligencia de los STI se representa a través de los módulos que lo componen: modelo del estudiante, módulo tutor, módulo experto y módulo interfaz. Estos componentes interactúan entre sí para cumplir distintas funciones con el fin de presentar una enseñanza adaptativa [23]. En la Fig. 1 se muestra la arquitectura básica de un STI.

El modelo del estudiante es el componente más importante de STI debido a que almacena conocimiento sobre el estudiante que sirve para adaptar la enseñanza. La importancia del modelo del estudiante radica principalmente en el conocimiento sobre el estudiante que proporcionará al Módulo Tutor a fin de personalizar el proceso de enseñanza-aprendizaje [19]. El modelo del estudiante es una representación del estudiante, ya que refleja las experiencias de aprendizaje y el conocimiento del dominio que ha adquirido.

De manera tradicional, el modelo del estudiante almacena información sobre el conocimiento y el desempeño del estudiante. Pero, ya se han integrado otros elementos tales como: personalidad, estilo de aprendizaje, preferencias, así como las emociones, con el fin de proporcionar una enseñanza adecuada para cada estudiante particular. De acuerdo con Prentzas [15] el modelo del estudiante puede almacenar cuatro elementos principales: parámetros de interacción, conocimiento del dominio y características del estudiante.

Las características del estudiante y el conocimiento del dominio son aspectos que influyen directamente en el proceso de enseñanza, mientras que, los parámetros de interacción influyen de forma indirecta.

Los parámetros de interacción forman la base del modelo tomando en cuenta la interacción del estudiante con el STI, es decir, avance en el curso, conceptos a los que

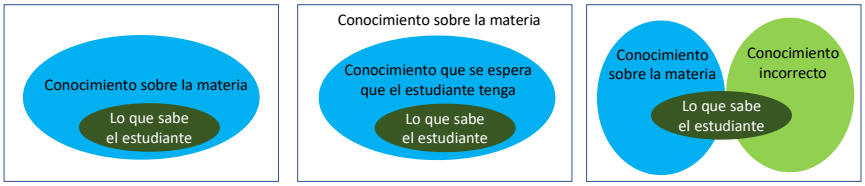


Fig. 2. Tipos de modelos del estudiante. Izquierda: modelo sobrepuesto, centro: modelo diferencial y derecha: modelo de perturbación [17].

accedió, ayuda que solicitó, respuestas y calificaciones en los ejercicios. En las características del estudiante se incluyen las preferencias sobre los medios, nivel de conocimiento del dominio, nivel de concentración, experiencia en el uso de STI, y nivel de detalle deseado del contenido. El conocimiento del dominio se representa a través de estereotipos que denotan diferentes niveles de conocimiento del dominio.

Por otro lado, Cataldi [3] menciona que el modelo del estudiante incluye las siguientes características: estilos de aprendizaje, perfil psicológico y estado de conocimientos. Los estilos de aprendizaje se basan en la Teoría de Felder y Silverman [8]. Se conforma una base de datos que integra las características de cada uno de los estilos de aprendizaje y los métodos de selección del material que considera las características de cada estudiante.

El perfil psicológico se basa en la Teoría de Inteligencias Múltiples de Gardner [10] y se categoriza y actualiza de acuerdo con la interacción del estudiante con el STI. En el estado de conocimientos se encuentra el mapa de los conocimientos que tiene el estudiante sobre la materia, mismo que se actualiza conforme el estudiante avanza en las actividades.

En la literatura se identifican cuatro tipos principales de modelos del estudiante: modelo de superposición (*overlay model*), modelo diferencial (*differential model*), modelo de perturbación (*perturbation model*) y modelo centrado en restricciones (*constraints-based model*) [17].

El modelo de superposición establece que el conocimiento del estudiante es un subconjunto propio del conocimiento del experto y asume que todas las diferencias entre el comportamiento de los estudiantes y el comportamiento del tutor se interpretan como falta de conocimiento del estudiante.En el modelo diferencial se divide el conocimiento del estudiante en dos tipos: el conocimiento que debería tener y el que no se puede esperar que tenga, es decir, reconoce y trata de representar el conocimiento del estudiante con las diferencias entre el estudiante y el tutor.

El modelo de perturbación considera que el conocimiento del estudiante puede ser potencialmente diferente en calidad y cantidad al del experto. En el modelo centrado en restricciones el conocimiento del estudiante se interpreta como una lista de restricciones que han violado el proceso de resolución del problema [17]. En la Fig. 2 se presenta el modelo del estudiante sobrepuesto, diferencial y de perturbación.

La función principal del módulo tutor de un STI es definir, representar, regular y aplicar la mejor estrategia de enseñanza pedagógica, con el objetivo de alcanzar las metas de aprendizaje [13]. El módulo tutor interactúa de manera constante con el modelo del estudiante, ya que analiza la información del estudiante actual para poder

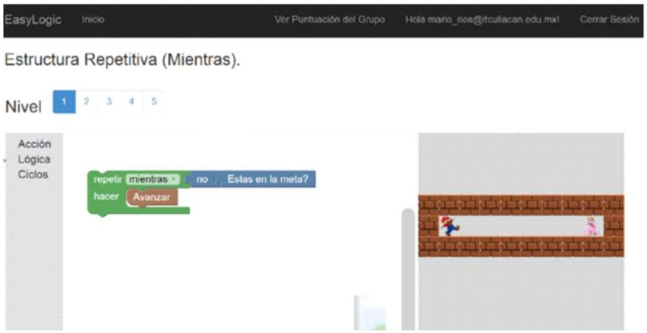


Fig. 3. Interfaz de *EasyLogic* [24].

establecer el material adecuado de acuerdo con su estado. Esto es posible gracias a que integra el conocimiento sobre diferentes estrategias pedagógicas.

De acuerdo con Molina y colegas [14], el módulo tutor consta de tres submódulos: protocolos pedagógicos, planificador de lección y analizador de perfil. Los protocolos pedagógicos, consisten en diferentes estrategias y técnicas de enseñanza que se aplican de acuerdo con el perfil del estudiante. El planificador de lección organiza los contenidos que se presentarán al estudiante.

El analizador de perfil es quien analiza las características del estudiante para seleccionar la estrategia pedagógica más conveniente. El módulo experto es quien contiene el conocimiento sobre el dominio y proporciona la información que se presentará al estudiante de una manera adecuada para que el estudiante pueda adquirir las habilidades y conocimiento de la materia.

El módulo experto también debe cumplir con tres aspectos importantes: i) servir como fuente de conocimiento para ser presentado al estudiante, lo que incluye la generación de preguntas, explicaciones y respuestas, ii) proporcionar un estándar para evaluar el desempeño del estudiante, y iii) tener la capacidad de detectar errores sistemáticos y la brecha en el conocimiento del estudiante y la posible causa de ello.

De acuerdo con Cataldi y colegas [2], el módulo experto consta de conocimiento y elementos didácticos. El conocimiento se presenta a través de conceptos, preguntas, ejercicios, problemas y sus relaciones. Los elementos didácticos se componen de imágenes, videos, sonidos, es decir material multimedia que se requiere para facilitar al estudiante la apropiación de conocimiento en la sesión pedagógica.

3. Gamificación en la educación

Los juegos como herramientas de aprendizaje representan un enfoque prometedor debido a sus habilidades para enseñar y reforzar conocimiento, así como para desarrollar habilidades importantes como la resolución de problemas, el pensamiento crítico, la comunicación y la colaboración. Además, tienen una fuerte capacidad de motivar y de enganchar a las personas en la interacción. El propósito principal de los juegos serios no es únicamente la diversión, si no que busca aprendizaje, capacitación, desarrollo de habilidades, rehabilitación, entre otros fines.

Tabla 1. Principios de diseño de gamificación en la educación [6].

Principios de Diseño		Mecánica de juego
Metas	Metas específicas, claras, moderadamente difíciles, inmediatas	
Desafíos y misiones	Tareas de aprendizaje claras, concretas y procesables con complejidad creciente	
Personalización	Experiencias personalizadas, dificultad adaptativa; desafíos que se adaptan perfectamente al nivel de habilidad del jugador, aumentando la dificultad a medida que la habilidad del jugador aumente	
Avance	Avance visible hacia dominar el tema y la tarea	Puntos, barras de avance, niveles, bienes/dinero virtual
Retroalimentación	Retroalimentación inmediata o acortar los ciclos de retroalimentación; recompensas inmediatas en lugar de vagos beneficios a largo plazo	
Competencia y cooperación/ciclos de participación social		Insignias, tablas de clasificación, niveles, avatares
Calificación acumulada		Puntos
Estado visible	Reputación, credibilidad y reconocimiento social	Puntos, insignias, tablas de clasificación, niveles, avatares
Acceso/Desbloqueo de contenido		
Libertad de elegir	Múltiples rutas hacia el éxito, permitir a los estudiantes elegir sus propios subobjetivos dentro de la tarea más amplia	
Libertad para fallar	Riesgo bajo de presentación, múltiples intentos	
Narración		Avatares
Nuevas identidades y/o roles		Avatares
Integración		
Restricción de tiempo		Reloj de cuenta regresiva

Tabla 2. Mecánicas de juego [9].

Mecánica de juego	Descripción
Puntos	Se asigna un valor cuantitativo a determinadas acciones y se van acumulando a medida que se realizan
Niveles	Se define una serie de niveles que el usuario debe ir superando para llegar al siguiente
Premios	A medida que se consiguen diferentes objetivos se van entregando premios a modo de "colección"
Regalos	Bienes que se dan al jugador o jugadores de forma gratuitita al conseguir un objetivo
Clasificaciones	Clasificar a los usuarios en función de puntos u objetivos logrados, destacando los mejores en una lista
Desafíos	Competencias entre los usuarios, el ganador obtiene los puntos o el premio.
Misiones o retos	Resolver o superar un reto u objetivo, ya sea solo o en equipo

Tabla 3. Dinámicas de juego [9].

Dinámica de juego	Descripción
Recompensas	Obtener un beneficio merecido
Estatus	Establecerse en un nivel jerárquico social valorado
Logro	Superación o satisfacción personal
Competencia	Intentar ser mejor que los demás

Los juegos serios se han aplicado en diferentes sectores e industrias con mucho éxito. Por ejemplo, en salud y educación podemos encontrar muchos investigaciones para apoyar a los niños con problemas de aprendizaje [4]. La gamificación en la educación propone el uso elementos de pensamiento y diseño de juegos en el diseño de ambientes de aprendizaje para mejorar el compromiso y la motivación de los estudiantes. La gamificación se define como el uso de elementos de diseño de juegos en contextos no lúdicos [5]. Se trata de un campo de investigación nuevo.

La gamificación difiere de los juegos educativos o juegos serios en que los juegos educativos implican el diseño de juegos completos con fines no relacionados con el entretenimiento, mientras que las aplicaciones con gamificación solamente incluyen elementos de los juegos [6].

La gamificación ha tenido una rápida adopción en diferentes sectores por su potencial para influir en el comportamiento de los usuarios. Por ejemplo, plataformas como *Stackoverflow.com*, *codeacademy.com* y *khanacademy.org*, entre otras, utilizan elementos de juego para atraer y motivar a los usuarios [6].

De este interés creciente, se han identificado los principios de diseño para la aplicación de la gamificación en la educación [6]. En la Tabla 1 se presentan los principios de diseño y, en su caso, la mecánica del juego que se suele utilizar para implementarlos. Algunos de los principios de diseño son fundamentales y siempre están presentes en los sistemas educativos, pero es posible que deban adaptarse para ajustarse

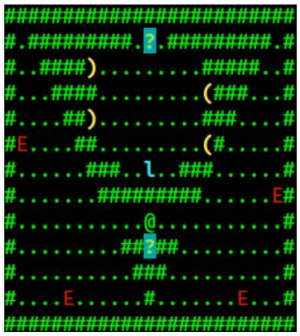


Fig. 4. Interfaz de PLMan [12].

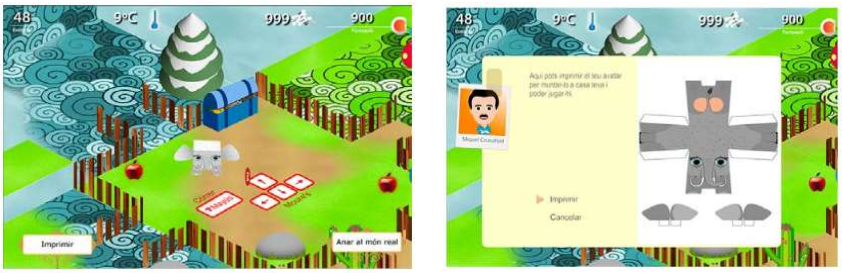


Fig. 5. Interfaz del juego digital Miquel Crusafont [7].

al paradigma de la gamificación. Por ejemplo, la retroalimentación debe ser inmediata o con ciclos más cortos. Algunos principios se han utilizados muy poco, y algunos son nuevos elementos de diseño tomados de los videojuegos [6].

Por otro lado, se identifica que los elementos de la gamificación se clasifican en mecánicas y dinámicas. Las técnicas mecánicas consisten en la forma de recompensar al usuario en función de los objetivos alcanzados, mientras que las técnicas dinámicas hacen referencia a la motivación del propio usuario para jugar y seguir adelante en la consecución de sus objetivos [9]. En la Tabla 2 se presentan las técnicas mecánicas más populares, mientras que en la Tabla 3 se presentan las técnicas dinámicas más utilizadas.

4. Ambientes de aprendizaje con gamificación

Los STI han implementado la gamificación como una parte fundamental para mejorar la interacción con el usuario, motivándolo a finalizar con cada una de las actividades y sobre todo cambiar de manera positiva el estilo de enseñanza-aprendizaje. Son muchas las investigaciones y prácticas realizadas por empresas o investigadores que han utilizado la gamificación en STI, a continuación, se citan algunas, seleccionadas por su carácter innovador y de la enseñanza.

EasyLogic es un ambiente de aprendizaje que integra reconocimiento afectivo con técnicas de gamificación. Este ambiente se ocupa de enseñar lógica algorítmica y



Fig. 6. Diseño preliminar del STI con gamificación.

programación a estudiantes de ingeniería. *EasyLogic* tiene un sistema de puntos en donde el estudiante gana puntos cada vez que lleva a cabo un ejercicio correctamente, y cada cierto número de puntos recibe un trofeo.

También lleva una tabla de clasificaciones en donde se muestran los estudiantes con los mayores puntajes. El desempeño de *EasyLogic* se evaluó comparándolo con el desempeño del mismo sistema sin el reconocimiento de emociones y sin los elementos de gamificación. Los resultados y las pruebas estadísticas mostraron que el aprendizaje de los estudiantes es mejor si se considera el estado afectivo del estudiante y si es motivado por medio de la gamificación [24]. En la Fig. 3 se muestra la interfaz de *EasyLogic*.

PLMan es un sistema de aprendizaje automatizado y personalizado con gamificación. Este sistema apoya dos cursos de primer año de Ingeniería Informática y de Ingeniería Multimedia y tiene como objetivo introducir a los estudiantes en el campo de la lógica. Los estudiantes aprenden el lenguaje de programación Prolog, programando la inteligencia artificial de un personaje similar a Pac-Man y utilizando una base de conocimientos en Prolog.

Para pasar cada nivel del juego, el personaje debe atravesar diferentes laberintos comiéndose los puntos y evitando a todos los enemigos [12]. En la Fig. 4 se muestra un ejemplo de uno de los laberintos de *PLMan*. El juego digital *Miquel Crusafont* se centra en descubrir a dicho personaje y explicar su trabajo en paleontología.

El área básica de estudio del juego es el conocimiento del medio natural y social, aunque también incluye otras áreas relacionadas, como matemáticas. el juego está estructurado en 14 niveles, y se basa en secuencias temporales, de modo que cada bloque puede contener eventos que ocurren en diferentes espacios y tiene una serie de objetivos que los jugadores deben superar. En cada nivel aumenta la dificultad de los retos [7]. En la Fig. 5 se muestra la interfaz de este juego.

SQL-Tutor es un STI maduro que enseña SQL y cuya eficacia ha sido probada en múltiples estudios. Los autores desarrollaron una versión de *SQL-Tutor* con gamificación con el fin de conocer los efectos de la gamificación en el aprendizaje. En esta versión lúdica incluyeron metas, evaluación y desafíos, estos son tres categorías de elementos del juego de la Teoría del Aprendizaje Lúdico [11]. Estos elementos se implementaron a través de 13 tipos de insignias divididas en tres categorías: básica,

clásica y élite. Los autores encontraron que las insignias pueden aumentar positivamente el rendimiento de los estudiantes en el STI. También encontraron evidencia de que el establecimiento de objetivos, los desafíos y los comportamientos de autoevaluación implementados como insignias afectan de manera indirecta y significativa los resultados del aprendizaje a través del tiempo dedicado a la tarea como mediador [20]. La plataforma tecnológica CER promueve el diálogo y el intercambio de experiencias entre docentes de educación técnica.

Esta plataforma cuenta con espacios de interacción tales como: foros y banco de recursos, lo que permite a los usuarios interactuar a través de valoraciones y comentarios a las contribuciones de otros usuarios. Con el fin de motivar un mayor involucramiento en los usuarios, se integraron técnicas de gamificación, tales como insignias, puntos, barras de avance y tablas de líderes. Los autores exponen que con la gamificación aumentó la participación y el intercambio entre los usuarios [21].

La gamificación se ha aplicado en diversos ambientes de aprendizaje basados en la web, tales como *Code Academy, Khan Academy y Stack Overflow,* entre otros. *Khan Academy* es una plataforma educativa que brinda aprendizaje personalizado a los estudiantes. Se trata de una organización sin fines de lucro que ofrece una variedad de cursos para diversas materias, desde matemáticas, ciencias, gramática, economía, hasta música e historia mundial, y está disponible en más de 36 idiomas. Esta plataforma de aprendizaje lúdica utiliza un sistema de calificación basado en recompensas en el que los estudiantes reciben puntos de experiencia por completar los módulos del curso [16].

5. STI con gamificación para matemáticas

Es ampliamente reconocido que los estudiantes perciben la materia de matemáticas como un área de difícil comprensión. Adicionalmente, en muchas escuelas públicas de nivel básico, la atención de los maestros a los estudiantes es insuficiente debido a que la cantidad de estudiantes por maestro es muy alta por lo que resulta muy difícil para los maestros dar un seguimiento puntual al aprendizaje de los estudiantes.

Por otro lado, las diversas evaluaciones al sistema educativo muestran un rezago en matemáticas, ciencias y compresión lectora, por lo que es necesario desarrollar herramientas tecnológicas que apoyen a combatir dicho rezago. Para apoyar a solucionar este problema proponemos el desarrollo de un sistema tutor inteligente dirigido a enseñar matemáticas a estudiantes de sexto año de educación básica.

Acorde con la naturaleza de los STI, el objetivo es brindar a los estudiantes una enseñanza personalizada que se base en sus necesidades de conocimiento particulares. Además, consideramos que uno de los problemas de las herramientas tecnológicas educativas tradicionales es que no invitan al estudiante a interactuar, por lo que el estudiante las percibe como una carga académica y no como una ayuda en el aprendizaje.

Por esta razón, el diseño de los comportamientos del STI estará basado en técnicas de gamificación, con el objeto de atraer y mantener la atención del estudiante en las actividades de aprendizaje. Un componente importante del STI es el modelo de juego que contiene las reglas y elementos juego que guiarán la interacción del estudiante con el escenario de juego. Con base en las propuestas de Dicheva y colegas [6], y de Werbach y Hunter [22], el modelo de juego se compone de mecánicas, dinámicas y

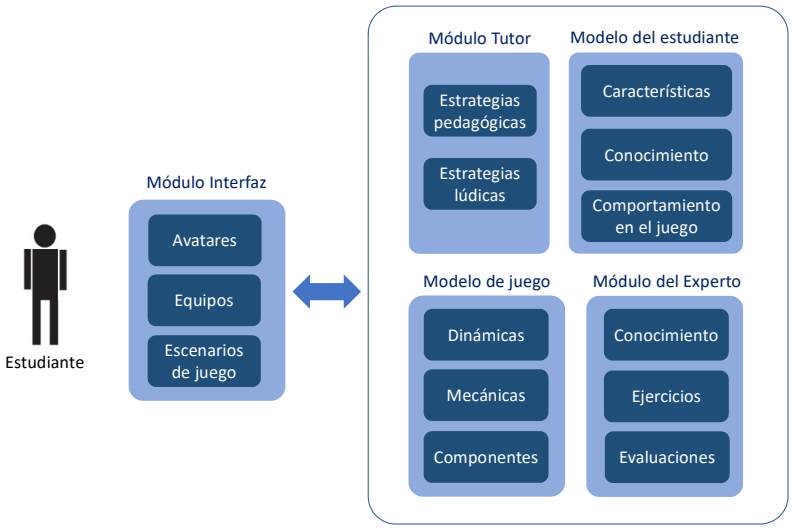


Fig. 7. Arquitectura del STI con gamificación.

componentes. Los componentes son los recursos que se utilizan para diseñar la actividad lúdica, tales como: avatares, puntos, insignias, niveles, misiones, barras de avance y tablas de clasificación. Las mecánicas hacen referencia a las reglas de la actividad lúdica y que establecen cómo recompensar a los estudiantes en función de su desempeño y de los objetivos. Estas pueden ser de varios tipos: competición, colección, cooperación, equilibrio o desafíos. Las dinámicas hacen referencia a la motivación del estudiante para jugar y seguir adelante en la consecución de sus objetivos. Son contextos en los que se desarrolla la actividad y pueden ser narrativa, emociones, avance, restricciones y relaciones sociales.

Al integrar gamificación al STI, estamos agregando módulos y modelos que permitan lograr los objetivos de aprendizaje y la implementación de las dinámicas y mecánicas de juego. En la Fig. 6 se muestra la arquitectura que se propone. Como puede apreciarse, los elementos que la componen son: módulo tutor, módulo experto, modelo del estudiante, módulo interfaz, y el modelo de juego.

El diseño inicial del STI con gamificación considera que los estudiantes llevarán a cabo misiones en una isla y serán recompensados con diferentes insignias y una puntuación acumulada. Se tendrán diferentes niveles que corresponden a la dificultad de las misiones y del material que se esté enseñando. Se tendrán barras de avance y una tabla de clasificaciones.

Se desea motivar tanto la competición como la colaboración, por lo que además se incluirá equipos virtuales para resolver las misiones. También se está trabajando en el diseño de un personaje que acompañe al estudiante en las misiones. En la Fig. 7 se muestra el diseño inicial del STI con gamificación. El módulo tutor establece las actividades pedagógicas y lúdicas que se presentarán al estudiante con base en el modelo del estudiante, es decir, con base en sus características, conocimiento y desempeño.

El módulo experto contiene el conocimiento que se enseña al estudiante, en la forma de explicaciones, ejercicios y material de evaluación. El modelo de estudiante es la representación del estado del estudiante, contiene sus características particulares, el conocimiento que tiene sobre matemáticas, y el desempeño y comportamiento en el juego. El módulo interfaz es el encargado de decidir cuál es la mejor manera de presentar el material educativo al estudiante. El modelo de juego es el responsable de procesar el desempeño de los estudiantes en las actividades para determinar las acciones en el juego, así como recompensas, insignias o castigos que se otorgarán al estudiante.

En el diseño del currículo y del material educativo intervienen expertos en pedagogía y con experiencia en dar clases a estudiantes de educación primaria. La base principal para los contenidos educativos son los lineamientos de la Secretaría de Educación Pública de México, a través de la *Guía para el maestro de educación básica para primaria* y con la ayuda de los expertos en pedagogía [18].

Para el desarrollo de los modelos inteligentes de cada uno de los componentes del STI, estamos evaluando diversas técnicas de inteligencia artificial. Principalmente, estamos trabajando con modelos robustos que nos permitan trabajar con incertidumbre, ya que uno de los problemas a que nos enfrentamos en la construcción de la representación del conocimiento y de la motivación del estudiante es la falta de certeza. Por esta razón se están evaluando modelos preliminares basados en redes bayesianas y en redes de decisión. Con base en nuestra experiencia previa, las redes bayesianas y las redes de decisión son mecanismos robustos ante la incertidumbre.

6. Conclusiones y trabajo futuro

A lo largo de varias décadas, los sistemas tutores inteligentes han demostrado ser una herramienta útil para ayudar a los estudiantes de diferentes niveles educativos a aprender diversas materias. Esto se debe a su habilidad para reconocer las necesidades específicas de cada estudiante y de responder a las mismas de manera adecuada. Sin embargo, los STI son un área abierta de investigación y la comunidad científica continúa buscando maneras de hacer la enseñanza cada vez más precisa y motivante.

La gamificación, definida como la integración de elementos de juego en contextos no lúdicos, ha mostrado ser una herramienta que proporciona experiencias que permiten que los estudiantes se motiven y se comprometan con las actividades, lo que a su vez impacta positivamente en el aprendizaje. En este artículo se presenta un análisis de la literatura sobre herramientas educativas que usan gamificación, se propone un STI para matemáticas de 6º grado de educación primaria y se presenta la arquitectura y diseño iniciales del STI.

Referencias

1. Backhoff Escudero, E.: México en PISA 2018. https://educacion.nexos.com.mx/mexico-en-pisa-2018/ (2019)
2. Cataldi, Z., Calvo, P., Salgueiro, F. A., Lage. F. J.: Diseño de sistemas tutores inteligentes con tecnología de agentes: Los agentes docentes en el módulo tutor. Revista Eletrônica Sistemas de Informação, vol. 6, no. 1 (2007) doi: 10.21529/resi.2007.0601007
3. Cataldi, Z., Lage, F. J.: Modelado del estudiante en sistemas tutores inteligentes. TE&ET

Revista Iberoamericana de Tecnología en Educación y Educacion en Tecnología, vol. 5, pp. 29–38 (2010)
4. Cornejo, R., Martínez, F., Álvarez, V. C., Barraza, C.,. Cibrian, F. L., Martínez-García, A. I., Tentori, M.: Serious games for basic learning mechanisms: Reinforcing mexican children's gross motor skills and attention. Personal and Ubiquitous Computing, vol. 25, no. 2, pp. 0375–390 (2021) doi: 10.1007/s00779-021-01529-0
5. Deterding, S., Dixon, D., Khaledet, D., Nacke, L. E.: From game design elements to gamefulness: Defining "gamification. MindTrek '11, In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, pp. 9–15 (2011) doi: 10.1145/2181037.2181040
6. Dicheva, D., Dichev, C., Agre, G., Angelova, G.: Gamification in education: A systematic mapping study. Educational Technology Society, vol. 18, no. 3, pp. 75–88 (2015)
7. Eguia-Gómez, J. L., Contreras, R., Solano, L.: Juegos digitales desde el punto de vista de los profesores. Una experiencia didáctica en aulas primaria catalanas. Education in the Knowledge Society, vol. 16, no. 2, pp. 31–48 (2015) doi: 10.14201/eks2015161
8. Felder, R. M., Silverman, L. K.: Learning and teaching styles and libraries. J. Engineering Education, vol. 78, pp. 674–681 (1988)
9. Gaitan, V.: Gamificación: El aprendizaje divertido. https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/#:~:text=La Gamificación es unatécnica, concretas%2C entre otros muchos objetivos, last accessed 2021/04/03
10. Gardner, H.: Inteligencias múltiples, la teoría en la práctica (2018) doi: 10.1016/j. quaint.2017.09.030
11. Landers, R., Armstrong, M. B., Collmus, A. B.: How to use game elements to enhance learning: Applications of the theory of gamified learning. Serious Games and Edutainment Applications, vol. 2, pp. 1–702 (2017) doi: 10.1007/978-3-319-51645-5
12. Llorens-Largo, F., Gallego-Durán, F. J., Villagrá-Arnedo, C. J., Compañ-Rosique, P., Satorre-Cuerda, R., Molin-Carmona, R.: Gamification of the learning process: Lessons learned. IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 11, no. 4, pp. 227–234 (2016) doi: 10.1109/RITA.2016.2619138
13. Martínez, M., Flores, M., Cruz. R. G., Soberanes, A.: Propuesta de arquitectura para un sistema tutorial inteligente móvil. Revista Iberoamericana de las Ciencias Computacionales e Informática, vol. 8, no. 16 (2009)
14. Molina-Hurtatiz, Y. E., Pascuas-Rengifo, Y. S., Millan-Rojas, E. E.: Sistemas tutores inteligentes como apoyo en el proceso de aprendizaje. Procedia Computer Science, vol. 6, no. 1 (2015) doi: 10.14483/udistrital.jour.redes.2015.1.a02
15. Prentzas, J., Hatzilygeroudis, I., Garofalakis, J.: A web-based intelligent tutoring system using hybrid rules as its representational basis. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds) Intelligent Tutoring Systems. ITS 2002. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 2363 (2002) doi: 10.1007/3-540-47987-2_16
16. Van Roy, R., Deterding, S., Zaman, B.: Uses and gratifcations of initiating use of gamifed learning platforms. In: CHI EA '18: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–6 (2018) doi: 10.1145/3170427.3188458
17. Salisu-Muhammad, S., Abdullahi-Baffa, B., Shehu, A.: Artificial intelligence approaches in student modeling: Half decade review (2010-2015). IJCSN - International Journal of Computer Science and Network, vol. 5, no. 5 (2016)
18. SEP: Programa de estudios 2011, sexto grado (2011)
19. Suárez-Granados, J. J., Rodríguez del Rey, Y. A., Pérez-Fernández, A. C.: Metodología para desarrollar un sistema tutor inteligente basado en la web, para estudiantes de ingeniería. Revista Universidad y Sociedad, vol. 8, no. 4 (2016)
20. Tahir, F., Mitrovic, A., Sotardi, V.: Investigating the effects of gamifying SQL-Tutor. In: Virtual confernece: 28th International Conference on Computers in Education, pp. 416–425 (2020)

21. Valda-Sanchez, F., Arteaga-Rivero, C.: Diseño e implementación de una estrategia de gamificacion en una plataforma virtual de educación. Fides et Ratio-Revista de Difusión cultural y científica de la Universidad La Salle, vol. 9, no. 9, pp. 65–80 (2015)
22. Werbach, K., Hunter, D.: For the win: How game thinking can revolutionize your businessno title. Wharton Digital Press (2012)
23. Woolf, B. P.: Building intelligent interactive tutors student-centered strategies for revolutionizing e-learning (2010)
24. Zatarain-Cabada, R.: Reconocimiento afectivo y gamificación aplicados al aprendizaje de lógica algorítmica y programación. Revista Electrónica Investigación Educativa, vol. 20, no. 3 (2018) doi: 10.24320/redie.2018.20.3.1636

ISSN 1870-4069

Análisis de rendimiento de particionamiento por medoides y medianas

María Beatriz Bernábe Loranca[1], Jonathan Ramos Flores[1], Rogelio González Velázquez[1], Gerardo Martínez Guzmán[1], Juan Carlos Duque Cardona[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, México

[2] Universidad EAFIT, Departamento de Ciencias, Colombia

beatriz.bernabe@gmail.com

Resumen. Este trabajo consiste en reportar la experiencia computacional del algoritmo PAM (Particionamiento Alrededor de los Medoides) y de la P-Mediana cuando se contrastan sus propiedades para establecer si existe semejanza algorítmica. Entonces, bajo la conjetura de que ambos algoritmos son equivalentes en cuanto a la función objetivo y la configuración de los grupos que generan, el objetivo de este trabajo se centra en comparar los óptimos y centroides (medoides y medianas) para ambas propuestas. El experimento que permite apoyar el supuesto de equivalencia entre PAM y la P-Mediana se apoya del conjunto de unidades geográficas de la Zona Metropolitana del Valle de Toluca. Una vez que se procesaron los datos para distintos tamaños en los grupos, se registró el tiempo, la calidad de las soluciones y la estructura de los grupos. Se concluyó que PAM y la P-Mediana proporcionan resultados consistentes a la hipótesis planteada.

Palabras clave: Combinatorio, PAM, LINGO, óptimo, ramificación y acotamiento.

Medoid and Median Partitioning Performance Analysis

Abstract. This work consists of reporting the computational experience of the PAM (Partitioning Around Medoids) algorithm and the P-Median when contrasting their properties to establish whether algorithmic similarity exists. Then, under the conjecture that both algorithms are equivalent in terms of the objective function and the configuration of the groups they generate, the objective of this work focuses on comparing the optima and centroids (medoids and medians) for both proposals. The experiment that allows supporting the assumption of equivalence between PAM and P-Median is based on the set of geographic units of the Metropolitan Zone of Toluca Valley. Once the data were processed for different sizes in the groups, the time, the quality of the solutions

and the structure of the groups were recorded. It was concluded that PAM and P-median provide consistent results to the hypothesis.

Keywords: Combinatorial, PAM, LINGO, optimal, branch and bound.

1. Introducción

Es importante destacar la complejidad computacional que implica la solución del problema de Particionamiento [1]. Este problema es considerado como un problema combinatorio NP-duro que exige que su solución sea generada a partir de la utilización de métodos no exactos de optimización.

En términos generales, las metodologías para la solución de este problema pueden ser divididas en métodos exactos y aproximados [2]. Dado que PAM y P-mediada pueden ser resueltos bajo procedimientos exactos, este trabajo centra su atención en analizar su desempeño en tres direcciones: 1) equivalencia en el valor de la función objetivo, 2) comparación de los centroides y 3) costo computacional [3].

La evidencia empírica que se aporta se basa en los 469 AGEBs (Áreas Geoestadísticas Básicas) de la Zona Metropolitana del Valle de Toluca (ZMVT).

El trabajo se encuentra estructurado como sigue; la presente introducción como sección 1. La sección 2 aborda los preliminares de este manuscrito. La sección 3 se centra en los aspectos generales relacionados con el particionamiento y de k-medoides. En la sección 4 se expone el modelo de la P-Mediana y PAM. La sección 5 se ocupa de la experiencia computacional y finalmente se presentan las conclusiones.

2. Preliminares

En términos generales, el agrupamiento se divide en dos tipos:

a) Métodos de particionamiento, entendidos como algoritmos que dividen el conjunto de datos en k grupos, donde el número entero k debe ser especificado por el usuario. El algoritmo debe satisfacer las condiciones de partición clásica que se estudia en conjuntos discretos. Normalmente el usuario ejecuta el algoritmo para un rango de valores k y evaluar distintas pruebas.
b) Métodos jerárquicos, los cuales son algoritmos aglomerativos que producen una jerarquía completa de agrupaciones del conjunto de datos. Esta clase de algoritmos no se abordan en este artículo.

De acuerdo a la descripción anterior, estamos colocados en el primer tipo debido a la evidente pertenencia de PAM a la categoría de clasificación por particiones. Sin embargo, aunque la P-Mediana no puede ni debe ser etiquetado como un algoritmo de particionamiento, muchos autores han implementado la P-Mediana usando particionamiento cuando el algoritmo de ramificación y acotamiento que usan optimizadores comerciales, se ven limitados para resolver la P-Mediana.

Incluso, un problema de particionamiento geográfico se puede modelar como un problema de la P-Mediana.

Es importante señalar que el presente documento se ocupa de obtener particiones ejemplares y exactas ya que las aproximaciones metaheurísticas no se incluyen en este análisis debido a su naturaleza aleatoria. En este aspecto, es oportuno comentar la complejidad computacional del particionamiento. El problema de particionamiento consiste en hallar la partición $P \in P_k$ que maximiza la inercia inter-clases $B(P)$, es decir, maximizar la intensidad de la separación entre los centros de los grupos. La complejidad computacional de este problema es del tipo $NP - hard$ [2].

El tamaño combinatorio de este problema puede verse con el siguiente ejemplo. Si denotamos por $S(n,k)$ y B_n el número de particiones de Ω en k clases no vacías y el número total de particiones de Ω respectivamente, entonces por ejemplo $S(60,2) \approx 0.58 \times 10^{18}, S(60,5) \approx 0.72 \times 10^{40}, S(100,5) \approx 0.66 \times 10^{68}$, mientras que $B_{10} = 115975, B_{15} \approx 0.14 \times 10^{10}$ y $B_{40} \approx 0.16 \times 10^{36}$.

En un problema de particionamiento donde Ω tenga 100 objetos y el número de clases sea $k = 5$, si existiera sobre la Tierra un computador tan veloz que fuese capaz de calcular $B(P)$ para cada una de las particiones P de Ω en un tiempo de 10^{-10} segundos en busca de un máximo global, al computador le tomaría algo más de 2×10^{48} siglos en completar el análisis de todas las particiones del problema [2].

2.1. Particionamiento K-medias: inicios y generalidades

A pesar de que no se compara ni analiza a k-medias en este artículo, para exponer la situación en términos razonablemente amplios, se describe de breve manera a K-medias dada su popularidad, relevancia y características básicas que otros algoritmos han tomado y respetado pero que a la vez han mejorado para rebasar las debilidades.

Es decir, se han planteado propuestas similares, pero bajo distintas formas e hipótesis e incluso mejorando sus aspectos teóricos, algorítmicos, así como modificaciones del método (por ejemplo, particionamiento sobre medoides).

Con la publicación de Sokal 1963, el análisis clúster surge como una importante área de estudio [4]. Con Lerman 1970, se inicia la publicación de un compendio de libros y artículos que sentaron las bases de esta área de estudio, expuestos cronológicamente que además profundizan en la naturaleza matemática del problema del agrupamiento [5, 6, 7].

Por otra parte, la publicación de Hartigan 1975 impacta en el mundo académico por su estudio en aspectos computacionales de los algoritmos, lo cual implicó que los problemas y métodos básicos del agrupamiento se hicieron ampliamente conocidos por la comunidad científica [8].

El algoritmo k medias trata de minimizar el criterio clásico de distancia de cada una de las observaciones de la muestra a la media de su grupo al cuadrado. Dicho algoritmo presenta una serie de extensiones y generalizaciones entre las que destacan el caso borroso- o fuzzy-, el de máxima verosimilitud, y aquellos criterios basados en convexidad, entre otros.

Uno de los principales enfoques de las técnicas de agrupamiento se basa en el criterio de la suma de los cuadrados de la varianza conocido como k-medias. Entre las desventajas de K medias se tienen. 1) Encontrar el valor K es una tarea difícil, 2) No es efectivo cuando se usa con clúster global, 3) Si se seleccionan diferentes particiones iniciales, puede variar el resultado de los clústeres, 4) El algoritmo no maneja clústeres de diferente tamaño y densidad.

3. Particionamiento

Los algoritmos de partición han sido durante mucho tiempo algoritmos de clustering, clasificación o agrupamiento por particiones muy populares antes de la aparición de la minería de datos.

Informalmente, el particionamiento se entiende como sigue: Dado un conjunto D de n objetos en un espacio d-dimensional y un parámetro de entrada k, un algoritmo de partición organiza los objetos en k clústeres de tal manera que la desviación total de cada uno de ellos respecto a su centro de clúster o a una distribución de clústeres se minimiza. La desviación de un punto puede calcularse de forma diferente en distintos algoritmos y se denomina más comúnmente función de similitud.

3.1. K-medoids

Dentro de los algoritmos de particionamiento, destaca el particionamiento por medoides (medoids). A diferencia del algoritmo k-means, el método k-medoids utiliza el objeto más céntrico (medoids) de un clúster para ser el centro del clúster en lugar de tomar el valor medio de los objetos de un clúster. Por ello, el método k-medoids es menos sensible al ruido y a los valores atípicos. Sin embargo, esto implica un mayor costo computacional. El siguiente algoritmo describe los aspectos generales de los algoritmos k-medoides:

3.1.1 Algoritmo de reubicación iterativa (RI)

El algoritmo de reubicación iterativa generalizada se compone de los siguientes pasos:

Entrada: El número de clúster k, y una base de datos que contiene n objetos.

Salida: Un conjunto de k clústeres que minimiza una función de criterio E.

Método:

1) elegir arbitrariamente k centros/distribuciones como solución inicial;
2) repetir;
3) (re)calcular la pertenencia de los objetos según la solución actual;
4) actualizar algunos/todos los centros/distribuciones de los clústeres según a las nuevas pertenencias de los objetos;
5) hasta que (no haya cambios en E);

El estado de inicialización k-medoids y variantes es igual a RI en el sentido de la selección aleatoria de los k objetos son el centro de los conglomerados (grupos). De las propiedades de k-means, el clustering k-medoids toma el procedimiento de asignar un objeto a su centro más cercano, lo cual se resuelve en el paso 4 del algoritmo RI haciendo que el paso 3 sea redundante, por tanto, K-medoids dista tanto de RI como de k-medias en que como máximo se cambiará un centro en el paso 4 para cada iteración.

Este cambio de centro debe dar lugar a una disminución de la función criterio, que suele ser la función de error al cuadrado en la minimización de distancias entre objetos y centroides y utiliza implícitamente una medida de distancia [9].

Para desarrollar el paso 4, un algoritmo de k-medoides como PAM es considerado [3], entonces se itera a través de todos los k centros de clúster e intenta reemplazar cada uno de ellos con uno de los otros $(n-k)$ objetos. Para cada uno de estos reemplazos, si la función de error cuadrado E disminuye el reemplazo tendrá lugar haciendo que se produzca la siguiente iteración del algoritmo RI.

Sin embargo, si no se encuentra tal reemplazo después de pasar por todos los k clústeres, no habrá ningún cambio en E, entonces el algoritmo termina con un óptimo local. Como PAM intenta reemplazar cada uno de los k centros de clúster con uno de los $(n-k)$ objetos y cada uno de estos intentos resulta en $(n-k)$ operaciones para calcular E, la complejidad total de PAM en una iteración es $O(k(n-k)^2)$.

Para valores grandes de n, este cálculo se vuelve muy costoso [9]. Debido a su complejidad, PAM funciona eficazmente para conjuntos de datos pequeños, pero no se adapta bien a conjuntos de datos grandes. Esta afirmación justifica que se hayan elegido pruebas de tamaño pequeño para el análisis que se persigue en el presente trabajo.

Por otro lado, para tratar conjuntos de datos más grandes, otras estrategias deben ser desarrolladas y en un trabajo futuro se abordarán. Por ejemplo, el problema de grandes cantidades de datos en particionamiento por medoides, fue expuesto por Kaufman y Rousseeuw y desarrollaron un método basado en el muestreo, llamado CLARA (Clustering LARge Applications) [3].

En lugar de tomar en consideración todo el conjunto de datos, se elige una pequeña porción de los datos reales como representante de los mismos. Los medoides se eligen a partir de esta muestra utilizando PAM y la disimilitud media se calcula utilizando todo el conjunto de datos. Si se calcula un nuevo conjunto de medoides que da una disimilitud menor que la mejor solución anterior, se sustituye la mejor solución por el nuevo conjunto de medoides.

3.2. PAM: Aspectos generales y algoritmo

Una evaluación al algoritmo PAM puede plantearse de distintas maneras, sin embargo, la experiencia dicta que no conocer los detalles algorítmicos ni involucrarse directamente con la implementación conducirá probablemente a dudosos resultados, por tanto, el código Java que se programó un par de años atrás se evaluó y adicionalmente, se ha dado revisada literatura clásica sobre K-medoides, por tanto, se describe a modo de resumen en este apartado.

3.2.1 Alcances de K-medoides

El algoritmo K-medoids se utiliza para encontrar medoides en un clúster que es el punto central de un clúster. K-Medoids es más robusto en comparación con K-Means, ya que en K-Medoids encuentra a k como objeto representativo para minimizar la suma de las disimilitudes de los objetos de datos, mientras que K-Means utiliza la suma de las distancias euclidianas al cuadrado para los objetos de datos.

Partiendo de las bondades de K-Medoids que se basa en técnicas de representación de objetos para reducir los inconvenientes del algoritmo K-Means, se utilizó en este manuscrito para los objetivos planteados [10]. Los medoides son el objeto de datos del clúster que se encuentra más centrados, a veces conocidos en un sentido amplio como centroides. Los medoides se seleccionan aleatoriamente de los objetos de datos K_y para

formar el clúster K_y y los demás objetos de datos se colocan cerca de los medoides en un clúster. Luego se procesan todos los objetos de datos del clúster para encontrar nuevos Medoids de manera repetida para representar el nuevo clúster de mejor manera.

Después de encontrar los nuevos medoides se vinculan todos los objetos de datos al clúster. La ubicación de los medoides cambia en consecuencia con cada iteración. Así se forman k_y clústeres que representan n objetos de datos (en este epígrafe, k_y es la notación que se respetó de los autores Mark y Katherine [10]).

3.2.2 Algoritmo K-Medoides

Entrada: K_y: el número de clústeres, : un conjunto de datos que contiene n objetos.

Salida: Un conjunto de k_y clústeres.

Algoritmo:

- Seleccionar aleatoriamente k_y como Medoides para n puntos de datos.
- Encuentre los Medoids más cercanos calculando la distancia entre los puntos de datos n y los Medoids k y asignar los objetos de datos.
    - Para cada Medoids m y cada punto de datos o asociado a m hacer.
    - Intercambiar m y calcular el coste total de la configuración

Seleccionar el Medoids con el menor coste de la configuración.

- Si no hay cambios en las asignaciones, repetir los pasos 2 y 3 alternativamente.

4. P-Mediana

La P-Mediana consiste en que dado un conjunto de puntos (o ubicación de consumidores) y una matriz de distancias (o costos) entre todos y cada uno de los puntos, elegir p puntos (o ubicación de las instalaciones) con el propósito de minimizar la suma de las distancias de todos los puntos al punto elegido más cercano.

El desarrollo del planteamiento del problema P-Mediana tuvo lugar en los años 60´s se le puede atribuir a Hakimi en el caso discreto [11]

En 1970 ReValle and Swain presentaron la primera formulación de programación entera para el problema de la P-Mediana citado por [12]. En general el problema de la P-Mediana puede ser expresado matemáticamente como un problema de optimización discreta.

Una amplia cobertura de los métodos para atender el problema de la P-Mediana se encuentra en Reese donde se analiza también un conjunto de artículos sobre el P-Mediana que es clasificado como uno de los principales problemas tipo LAP (Problema de Ubicación Asignación, Location-Allocation Problem) [13].

Este problema determina la ubicación de las instalaciones y asigna los puntos de demanda a una instalación. El modelo de la P-Mediana puede establecerse en términos de grafos como: Sea $G = (V, E)$ un grafo no dirigido donde V es el conjunto de n vértices y E es el conjunto de aristas con un peso asociado que puede ser la distancia entre los vértices $d_{ij} = d(v_i, v_j)$ para toda $i, j = 1, \dots, n$ de acuerdo a determinada métrica, con las distancias se forma una matriz simétrica , encontrar $V_p \subseteq V$ tal que $|V_p| = p$, donde p puede ser o bien variable o fijo, y que la suma de las distancias más

cortas de los vértices en $\{V - V_p\}$ a su vértice más cercano en V_p se reduce al mínimo. Dicho en esta forma el PMP es un combinatorio y pertenece a la clase NP-Duro probado por Kariv y Hakimi en 1979 [14].

4.1. Modelo de la P-Mediana

El problema de la P-Mediana considera la siguiente situación. Se requiere particionar un conjunto finito de objetos en exactamente p grupos. Cada uno de dichos grupos estará caracterizado por uno de los objetos, que es seleccionado como la mediana del grupo, y el subconjunto de objetos asignado a dicha mediana.

Para cada par de objetos se especifica una distancia y se requiere minimizar la suma de distancias entre los objetos y las medianas a las que están asignados.

Sea $N = \{1, \ldots, n\}$ el conjunto de objetos. Para cada par $(i, j), i \in N, j \in N$, d_{ij} denota la distancia (similitud) entre los objetos i y j. Dado el número p, que denota el número de grupos, se necesita particionar el conjunto N en p subconjuntos disjuntos, es decir, $N = \bigcup_{k=1}^{p} N_k$ y $N_r \cap N_s = \emptyset$, para todo $r, s \in \{1, \ldots, p\}, r \neq s$.

A continuación, se considera el siguiente modelo de programación matemática para el problema. Se definen las siguientes variables de decisión:

$$y_i = \begin{cases} 1, & \textit{si el objeto i es selccionado como mediana,} \\ 0, & \textit{en otro caso.} \end{cases}$$

y

$$x_{ij} = \begin{cases} 1, & \textit{si el objeto j se asigna a la mediana, i,} \\ 0, & \textit{en otro caso.} \end{cases}$$

El problema se puede modelar de la siguiente manera:

$$\min \quad Z = \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ij} \quad , \tag{1}$$

$$\text{sujeto a} \quad \sum_{i \in N} x_{ij} = 1 \qquad j \in N \quad , \tag{2}$$

$$\sum_{i \in N} y_i = p \quad , \tag{3}$$

$$x_{ij} \leq y_i \qquad i \in N, j \in N \quad , \tag{4}$$

$$x_{ij} \leq \{0,1\} \quad i \in N, j \in N \quad , \tag{5}$$

$$y_i \leq \{0,1\} \qquad i \in N \, . \tag{6}$$

Las restricciones (2) aseguran que cada objeto se asigna a una de las medianas. La restricción (3) asegura que se seleccionan p objetos como medianas. Finalmente, las

Tabla 1. Pruebas de rendimiento con los centros de cada grupo.

	P-Mediana con Ramificación – Acotamiento (Lingo)			**PAM**		
K	Costo	Tiempo (Segundos)	Medianas	Costo	Tiempo (Segundos)	Medoides
4	11.16330	1383.21	72, 95, 132, 283	11.16329	0	252, 72, 132, 283
8	5.995200	1941.34	1, 21, 29, 46, 70, 90, 113, 314	5.9951	3	21, 196, 1, 314, 307, 46, 113, 90
12	3.934600	2547.67	27, 46, 63, 82, 91, 111, 128, 142, 155, 257, 267, 322	3.9392	60	296, 123, 149, 129, 27, 63, 181, 91, 267, 257, 108, 142
24	1.721400	2165.05	7, 36, 47, 50, 51, 54, 63, 75, 87, 90, 102, 111, 114, 116, 119, 135, 159, 186, 195, 210, 264, 272, 307, 311	1.7219	75	116, 186, 132, 329, 20, 142, 63, 50, 215, 210, 250, 119, 135, 307, 36, 159, 53, 264, 272, 7, 311, 114, 102, 111,

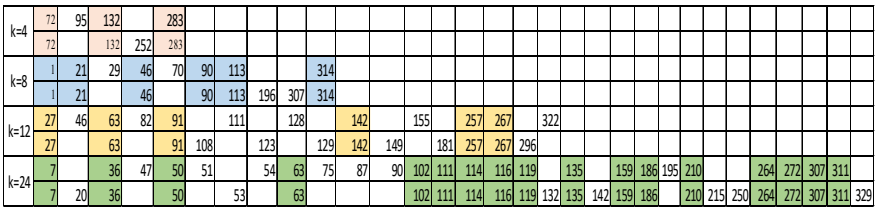


Fig. 1. Medianas vs Medoides.

restricciones (4) a (6) aseguran que los objetos solo puedan ser asignados a las medianas seleccionadas [13].

4.2. Modelo PAM

PAM es un método no jerárquico donde el conjunto de datos es particionado en un número previamente especificado de k conglomerados (grupos), luego iterativamente se asignan las observaciones a los conglomerados hasta que se satisface algún criterio de parada (función a optimizar), por ejemplo, que la suma de cuadrados dentro de los conglomerados sea mínima [3].

La función PAM está basada en la búsqueda de k objetos representativos llamados medoids (medoides) entre los objetos del conjunto de datos. Estos medoids son calculados tal que el total de disimilitud de todos los objetos hacia su medoide más cercano es mínima: es decir, la meta es encontrar un subconjunto $\{m_1 \dots m_k\} \subset \{1 \dots n\}$ que minimiza la función objetivo:

$$\sum_{i=1}^{n} \min_{t=1,\ldots,k} d(i, m_t)$$

.

Cada objeto es asignado al conglomerado correspondiente del centroide más cercano. Esto es, el objeto i es colocado en el conglomerado v_i cuando el medoide m_{vi} está más cercano a i que cualquier otro medoide m_w, o:

$$d(i, m_{vi}) \leq d(i, m_w) \text{ para todo } w = 1, \ldots, k$$

Actualmente el algoritmo de PAM consta de 2 pasos:

1.- Construir los centroides iniciales

m_1 es el objeto con el más pequeño $\sum_{i=1}^{n} d(i, m_1)$

.

.

m_k disminuye el objetivo (1) tanto como sea posible

2.- Intercambio

Repetir hasta lograr convergencia

considerar todos los pares de objetos (i, j) con

$i \in \{m_1, \ldots, m_k\}$ y $j \in \{m_1, \ldots, m_k\}$

y hacer el intercambio $i \leftrightarrow j$ para cualquiera que decrezca más el objetivo.

La función objetivo de PAM depende de las disimilitudes entre los objetos por tanto esta función solamente necesita la matriz de distancias como entrada [3].

5. Experiencia computacional

El propósito de este trabajo es revelar que, bajo mismas condiciones, PAM y P-Mediana son modelos equivalentes. Se puede comprobar este supuesto si se comparan los resultados en dos aspectos: costo de la función objetivo y configuración de la partición resultante.

Es conveniente señalar que la idea para determinar la equivalencia entre Pam y la P-Mediana tuvo lugar cuando una reflexión trivial rodeaba un problema de localización y diseño de territorio que bien se podía atender con la, pero del mismo modo con el particionamiento clásico. Por tanto, se evocó a las metodologías más utilizadas en el particionamiento territorial, donde destacan los modelos de localización-asignación ("location-allocation") y los de particionamiento de conjuntos ("set partitioning").

En el caso particular de diseño de territorio, el particionamiento territorial se modela como un problema de la P-Mediana. Sin embargo, en términos amplios, el particionamiento se puede resolver como un problema de la P-Mediana o la P-Mediana usa el particionamiento para su solución incluso cuando los métodos exactos no son una alternativa.

En este punto, se sostiene que, si existen dos algoritmos equivalentes, la decisión de escoger uno de ellos reside en conocer cual algoritmo genera mejor calidad de las soluciones en mejor tiempo de computo.

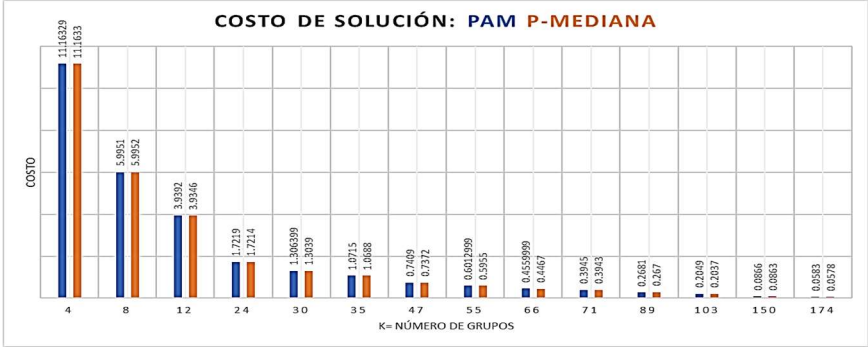


Fig. 2. Gráfica de rendimiento entre PAM y P-Mediana.



Fig. 3. Comparación del costo computacional.

Entonces, desde esa perspectiva y con el propósito de examinar el comportamiento de PAM y la P-Mediana para determinar su equivalencia, se diseñó un básico experimento que contrasta tres aspectos fundamentales y que garantizan de algún modo, la semejanza planteada: 1) función de costo, 2) centroides 3) costo computacional y configuración de los grupos. Informalmente, esta situación puede expresarse considerando que debido a que PAM crea particiones tal que cada grupo de la partición tiene un "centro" llamado medoide, el cual atrae a los objetos que minimizan su distancia hacia el medoide a través de la función de costo, el desafío consiste en retar a PAM con otro algoritmo cuya semántica algorítmica sea similar.

En este escenario, se escogió el modelo de la P-Mediana, el cual posee un método equivalente si se considera la siguiente situación: Se requiere particionar un conjunto finito de objetos en exactamente p grupos. Cada uno de dichos grupos estará caracterizado por uno de los objetos que es seleccionado como la mediana del grupo y el subconjunto de objetos asignado a dicha mediana. Para cada par de objetos se especifica una distancia y se requiere minimizar la suma de distancias entre los objetos y las medianas a las que están asignados.

De acuerdo a lo anterior, se subraya que la finalidad de este trabajo es evaluar 2 algoritmos ampliamente citados en la literatura para medir características semejantes. Dichos algoritmos son formulados también como métodos y se les conoce en términos amplios como de localización-asignación y de clasificación por particiones (PAM y P-Mediana respectivamente).

El experimento fue diseñado para que el costo de computo fuera razonable y de la misma manera el software elegido para programar la P-Mediana pudiera arrojar soluciones sin rebasar las limitaciones de Lingo. Lingo, resuelve la P-Mediana con Branch and Bound y ofrece un valor exacto en el resultado de la función objetivo e implícitamente produce una partición

Por otra parte, PAM proporciona una partición bajo un esquema exhaustivo-combinatorio y se asume que en estas condiciones PAM y P-Mediana son métodos equivalentes en al menos 3 puntos: 1) utilizan en la entrada de datos una matriz de distancias, 2) generan una partición, 2) usan la misma función de costo, 3) cada centro de los grupos que componen la partición tiene la misma meta (básicamente la equivalencia se centra en el punto 2, los demás puntos son tácitos).

5.1 Experimento

En esta subsección se describe la experiencia computacional para determinar la equivalencia entre PAM y P-Mediana.

Las tablas 1-5, concentran los resultados de las pruebas hechas a la P-Mediana con Lingo, y a PAM con un algoritmo de clasificación por particiones combinatorio. Como puede observarse, los resultados no son idénticos, excepto en la tabla 4. Se insiste que la diferencia en la función objetivo es un "épsilon", el cual se justifica en el cálculo numérico por la aproximación o truncamiento.

En la figura 1, hemos identificado las medianas y los medoides para la P-Mediana y PAM respectivamente. Por ejemplo, para k=4 la mediana=95 y el medoide=252 son los únicos centroides que no coinciden.

De la misma manera para k=24, las medianas 47, 51, 54, 75, 87, 90 son distintos a los medoides 20, 53, 132, 142, 215, 250, 329; el resto de los objetos son iguales. En la gráfica de la figura 2, se distingue la "igualdad" de los óptimos para PAM y la P-Mediana de 4, 8, 12, 24, 30, 35, 47, 55, 66, 71, 89, 103, 150, 174.

En la figura 3 se observa que el costo computacional de PAM es mejor hasta k=9 grupos con respecto a Ramificación y Acotamiento para la P-Mediana en Lingo.

En la tabla 3 se obtuvieron 2 casos de las pruebas realizadas. Se observa que para la instancia de k=4 y k=8, PAM produce un valor óptimo menor a la P-Mediana de Lingo con una diferencia de -0.00001 y -0.00010 respectivamente. Como puede verse para k=12 y k=24, Lingo minimiza mejor el objetivo que PAM (la diferencia sigue siendo una épsilon).

El comportamiento de PAM vs. P-Mediana en Lingo cuya diferencia en la distancia de los óptimos es casi despreciable, se aprecia en la siguiente tabla 4, la cual es una extensión de la tabla 3.

En la tabla 5 se distingue que tanto PAM como la P-Mediana tienen el mismo costo objetivo. En la figura 4, se ha ampliado la escala para que sea evidente la diferencia minúscula de los costos objetivos para los algoritmos en cuestión. A pesar de ello, se mantiene la conjetura de la equivalencia para PAM y P-Mediana. A continuación, la tabla 6 revela el registro de la prueba que alcanzó exactamente el mismo óptimo e incluso los mismos grupos con sus respectivos centros para ambos algoritmos.

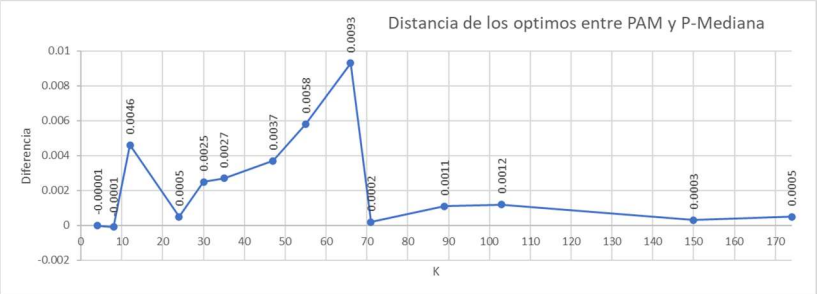


Fig. 4. Limites superior e inferior de los costos para PAM y P-Mediana.

Tabla 5. Comparación entre los dos costos de solución idéntica para k=5.

PAM	Lingo	PAM = Lingo	Resultado
6.88E-02 (0.06879999)	6.88E-02 (0.0688)	0. 068 = 0.068	Iguales

Tabla 6. Grupos con sus respectivos centros.

		PAM		P-Mediana	
K	Clúster	Medoide	Elementos del grupo	Mediana	Elementos del grupo
5	1	3	1,2,4.	3	1,2,4,5.
	2	28	15,19,22,23,26,27.	14	16,17,20,21,24.
	3	14	16,17,20,21,24.	18	11,13.
	4	18	11,13.	25	6,7,8,9,10,12,29,30.
	5	25	5,6,7,8,9,10,12,29,30.	28	15,19,22,23,26,27.

Después de analizar distintas particiones de la P-Mediana en Lingo y PAM, se tienen conclusiones que quizá muchos investigadores han experimentado, por ejemplo, la incapacidad de Lingo para generar soluciones debido a la gran cantidad de variables que genera el algoritmo implícito de Ramificación y Acotamiento.

En este trabajo, la instancia de la ZMVT es de 469 objetos y para que Lingo pudiera competir justamente con PAM, se observó que solo soporta hasta 369 objetos.

6. Conclusiones

Sabemos que un problema es de optimización combinatoria (OC) si en ellos las variables de decisión formadas por vectores X, sólo admiten valores enteros y su espacio de soluciones factibles Ω está formado por permutaciones o subconjuntos de números naturales. PAM pertenece a esta categoría combinatoria.

Por otra parte, la P-Mediana puede ser resuelta con algún software optimizador como Lingo, cuya implementación responde al algoritmo Branch and Bound, el cual produce una solución exacta. No obstante, las restricciones de los optimizadores para resolver la P-Mediana se enfrentan al número de variables que se generan por el algoritmo programado del modelo, que en este caso es entero-binario.

Un dilema que los investigadores tienen en problemas de esta magnitud, consiste en optar por dejar a un lado el software de optimización y programar el modelo de la P-Mediana con algún software de propósito general. Comúnmente la P-Mediana es implementada con las características del particionamiento clásico, siempre que no sea necesario incorporar métodos heurísticos para conseguir una solución aproximada.

En la literatura se encuentran distintas maneras de expresar tanto el particionamiento como la P-Mediana, incluso con distinta notación, pero todas las descripciones tienen la misma intención: crear grupos con un centroide tal que la distancia a cualquier otro objeto del conjunto de datos sea mínima.

Se ha insistido que los centroides en la P-Mediana son las medianas, y tal concepto no debe confundirse con la mediana estadística, dado que la P-Mediana es la generalización de la mediana conocida comúnmente como una medida estadística descriptiva, es decir, la mediana geométrica de un conjunto discreto de puntos o datos de una muestra en un espacio euclídeo, es el punto (centro) que minimiza la suma de las distancias a los puntos de la muestra, lo cual generaliza el concepto de la mediana estadística.

El significado de esta mediana se centra en la propiedad de minimizar la suma de distancias para datos unidimensionales y además proporciona una medida de tendencia central en dimensiones superiores.

Es posible afirmar que la mediana geométrica es un estimador de localización en estadística, incluso también es un indicador estándar en la resolución del problema de localización de instalaciones, donde modela el problema de localizar una instalación para minimizar el costo del transporte [15].

Formalmente para un conjunto dado de *m* puntos $x_1, x_2, \dots x_m$ con cada $x_i \in R^n$ la mediana geométrica se define como $Argmin \sum_{i=1}^{n} \|x_i - y\|_2$, donde argmin significa el valor del argumento $y$que minimiza la suma. En este caso, es el punto y desde donde la suma de todas las distancias euclidianas a x_i es mínima.

Esta afirmación justifica que la mediana geométrica sea la parte medular del modelo de la P-Mediana y también se incorpora como el medoide en el particionamiento sumado a las restricciones; esto es, una partición de un conjunto *A* está formada por los subconjuntos $A_1, A_2, \dots A_n$los cuales deben cumplir:

- Que la unión de todos los subconjuntos sea igual al conjunto dado $A_1 \cup A_2 \cup \dots \cup A_n = A$.
- Que todos los subconjuntos sean disjuntos entre sí y que ningún subconjunto sea vacío.

El particionamiento para un conjunto de datos se plantea matemáticamente como la P-Mediana y viceversa. Ello implica que ambos modelos comparten la función objetivo, excepto que los centros de los grupos son medoides para el particionamiento PAM. Los medoides son centros de grupos y el investigador elige a su criterio la mejor manera de colocar los medoides en el centro de los clústeres, y de acuerdo a la definición de mediana geométrica, ese centro concuerda con el concepto de mediana.

Revisando los costos de los métodos presentados, estos valores son iguales con una discrepancia muy pobre que obedece a la precisión de las cifras significativas aritméticas en el truncamiento y aproximación. Esta situación se puede resolver en PAM con algún método numérico como Newton-Fourier, Bisección o Punto fijo, por

ejemplo, caso contrario, en Lingo desconocemos la manera en que aproximan o truncan los decimales en la función de costo, ahí reside la diferencia del épsilon entre los valores de las funciones objetivo de PAM y P-Mediana en Lingo.

Sin embargo, las normas IEEE en el estándar 754, dicta que los softwares comerciales definan 3 formatos de punto flotante (32, 64, y 128 bits), por tanto, se puede arreglar el problema de aproximación en PAM y de este modo se conseguiría los valores objetivos sean exactamente iguales [16].

Respecto a la diferencia de los centros de los grupos, es más sencillo explicar la situación porque PAM en este artículo ofrece sólo una solución, lo cual no significa que sea la única, es posible que se generasen más óptimos. En otras palabras, en un algoritmo combinatorio como PAM, cuando en el código se comparan soluciones y estas son iguales, el algoritmo escoge cualquiera de ellas porque todas las soluciones óptimas generadas son "igual de buenas", entonces una de ellas coincidirá totalmente con la que produce Lingo.

Referencias

1. Díaz-García, J. A., Bernabé-Loranca, M. B., Luna-Reyes, D. E., Olivares-Benítez, E., Martínez-Flores, J. L.: Relajación lagrangeana para el problema de particionamiento de áreas geográficas. Revista de Matemática: Teoría y Aplicaciones, vol. 19, no. 2, pp. 169–181 (2012) doi: 10.15517/rmta.v19i2.1332
2. Piza-Volio, E., Murillo-Fernández, Á., Trejos-Zelaya, J.: Nuevas técnicas de particionamiento en clasificación automática. Revista de Matemática Teoría y Aplicaciones, vol. 6, no. 1, pp. 51–66 (2009)
3. Kaufman, L., Rousseeuw, P. K.: Clustering by means of medoids. In: Y. Dodge, 405–416 (1987)
4. Sokal, R. R., Sneath, P. H. A.: Principles of numerical taxonomy. W.H. Freeman and Company; 1st edition, pp. 1–359 (1963)
5. Gower, J. C.: Les bases de la classification automatique. Journal of the Royal Statistical Society Series, vol. 20, no. 2, pp. 195–197 (1971) doi: 10.2307/2346469
6. Jardine, N., Sibson, R.: Mathematical taxonomy. Universidad de California, pp. 1–286 (1971)
7. Anderberg, M.: Cluster analysis for applications. Academic Press (1973)
8. Hartigan, J.: Cluster algorithms. John Wiley & Sons (1975)
9. Han, J., Kamber, M., Tung, A.: Spatial clustering methods in data mining: A survey. Geographic Data Mining and Knowledge Discovery, pp. 188–217 (2001)
10. Van der Lear, M., Pollar, K., Bryan, J.: A new partitioning around medoids algorithm. Journal of Statistical Computation and Simulation, vol. 73, no. 8, pp. 575–584 (2003) doi: 10.1080/0094965031000136012
11. Hakimi, S.: Optimum location of switching centers and the absolute centers and medians of a graph. Operations Research, vol. 12, no. 3, pp. 450–459 (1964) doi: 10.1287/opre.12.3.450
12. Church, R.: Cobra: A new formulation of the classic P-Median location problem. Annals of Operation Research, pp. 103–120 (2003)
13. Reese, J.: Solution methods for the P-Median problem: An annotated bibliography. Networks an International Journal, vol. 48, no. 3, pp. 125–142 (2006) doi: 10.1002/net.20128
14. Jiang, H., Zhang, X., Li, M.: Backbone of the P-Median problem. Lecture Notes in Computer Science, vol. 4839, pp. 699–704 (2007) doi: 10.1007/978-3-540-76928-6_78
15. Wolf, G.: Foundations of location analysis. International Journal of Geographical Information Science, pp. 577–578 (2012)

16. https://ieeexplore.ieee.org/document/8766229

ISSN 1870-4069

Enfoques y técnicas de análisis en estudios experimentales para neuromarketing, una revisión de la literatura

Pamela Stephanie Rivera Pérez, Leslie Imelda Callejas Dorantes, Alejandro del Rey Torres Rodriguez

Instituto Tecnológico Superior de Misantla, México

{tephanrp, lesliecallejasdorantes}@gmail.com, artorresr@itsm.edu.mx

Resumen. En la literatura sobre el Neuromarketing se encuentran una diversidad de métodos y enfoques en los estudios experimentales que se realizan para indagar comportamientos humanos frente a estímulos de compra. Esto ha provocado el uso de diferentes técnicas para procesar datos siguiendo varios enfoques o propósitos en la investigación neurocientífica de usuarios o clientes. La finalidad de este trabajo es presentar una revisión cuantitativa de la literatura sobre Neuromarketing. Los resultados abarcan que la herramienta más utilizada fue el electroencefalograma (EEG), el tipo de procesamiento más común fue Event-related potential (ERP) y el enfoque más implementado fue el comercial. Este trabajo está dirigido a la tarea de ubicar áreas de oportunidad en la investigación experimental neurocientífica, aplicada en la investigación de mercados.

Palabras clave: Neuromarketing, experimentos, herramientas, técnicas, enfoques.

Approaches and Analysis Techniques in Experimental Studies for Neuromarketing, a Review of the Literature

Abstract. In the literature on Neuromarketing there are a diversity of methods and approaches in the experimental studies that are carried out to investigate human behavior in the face of purchase stimuli. This has led to the use of different techniques to process data following various approaches or purposes in the neuroscientific research of users or clients. The purpose of this work is to present a quantitative review of the literature on Neuromarketing. The results include that the most used tool was the electroencephalogram (EEG), the most common type of processing was Event-related potential (ERP) and the most widely implemented approach was the commercial one. This work is aimed at the task of locating areas of opportunity in experimental neuroscientific research, applied in market research.

Keywords: Neuromarketing, experiments, tools, techniques, approaches.

1. Introducción

El Neuromarketing es definido por [1] como una medición cerebral para poder estudiar detalladamente cómo los clientes y/o consumidores responden ante un estímulo, apoyándose en las neurociencias y utilizando técnicas como el electroencefalograma, la resonancia magnética, la topografía del estado estacionario, entre otras. [2] sostienen que el Neuromarketing tiene como finalidad el análisis cuantitativo de los mercados y las emociones, guiando al cliente durante el proceso de compra y logrando un mayor impacto que el marketing tradicional.

Con respecto a sus orígenes, [3] sostiene que la evolución del marketing al neuromarketing comenzó durante la década de 1990 (conocida como "la década del cerebro") y permitió descubrir aspectos del marketing que habían permanecido ocultos por años. Afirmaciones de este tipo han despertado el interés de los especialistas en marketing. En este tipo de estudios se suele elegir un área específica con la cual trabajar, como "Willigness to pay", Branding, o publicidad.

Esto indica el tipo de estímulo al que se le debe exponer a los participantes del experimento necesario para el estudio. Este estímulo puede presentarse a través de algún dispositivo multimedia, o de manera tangible mediante la interacción directa del sujeto con el producto, un ejemplo son la presentación de logos de marcas en el monitor de una PC o el testeo de productos comestibles. Ante esto se planifican instrucciones que guían al participante a experimentar el estímulo de tal manera que se le predisponga a crear un criterio hacia el producto evaluado.

Cuando esto sucede, se le debe dar un seguimiento al participante por medio de la medición fisiológica seleccionada para el estudio. Y finalmente, mediante el procesamiento de los datos obtenidos con la medición se da pie a una interpretación sobre los resultados. Este análisis puede ser una fuente importante de información útil al tomar decisiones estratégicas sobre la publicidad de un producto, embalaje o contenido.

Estudios con métodos como los descritos anteriormente, se han vuelto un gran fenómeno dentro de las bases de datos científicas que es digno de ser estudiado, pues el tipo de marketing utilizado no es el único aspecto diverso que caracteriza a los estudios de Neuromarketing. En este trabajo se detectan tres de ellos: enfoque, herramientas y técnicas. Se piensa que estos tres son los aspectos que mejor reflejan la variabilidad que existe dentro de los estudios de esta disciplina y también que su análisis refleja el estado actual del estudio del Neuromarketing.

Con respecto a las herramientas utilizadas, es claro que no en todos los estudios se aplican las mismas mediciones fisiológicas, pues al día de hoy existe una enorme gama de este tipo de herramientas, que se pueden utilizar por separado o al mismo tiempo. En este estudio, algunos de los trabajos mencionados utilizan varias herramientas al mismo tiempo, sin embargo, la herramienta más utilizada se halló que fue el EEG por separado. De igual manera, existen estudios que además de combinar herramientas propias del Neuromarketing, realizan combinaciones con herramientas del marketing convencional como los cuestionarios o las entrevistas.

Así, también existe una gran variabilidad de técnicas para procesamientos de datos obtenidos de mediciones fisiológicas, los cuales se aplican dependiendo de la herramienta elegida en el estudio. Este es el caso de la técnica mayormente encontrada en este trabajo, el ERP, que corresponde a el procesamiento de señales EEG.

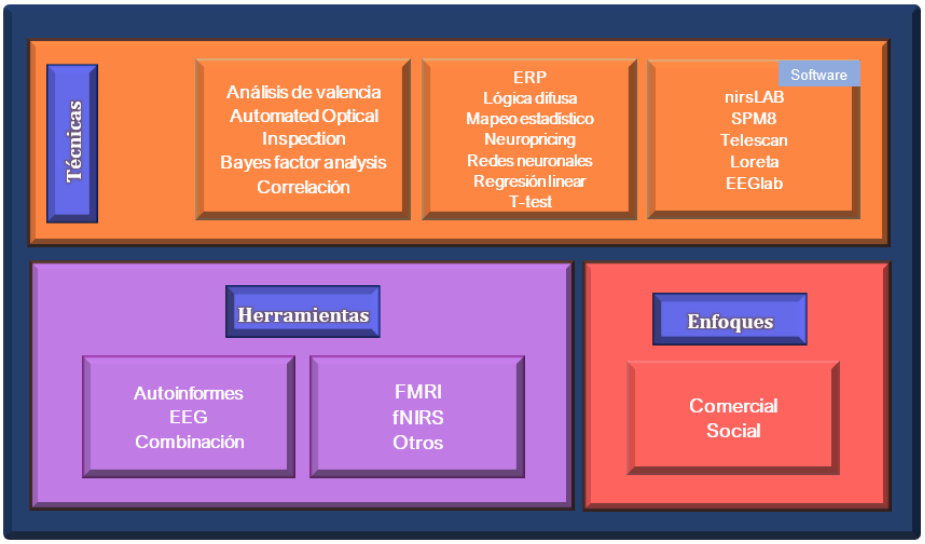


Fig. 1. Gráfico de clasificaciones del artículo.

Por último, en esta revisión se considera que un buen aspecto a considerar como causante de la variabilidad en los estudios de Neuromarketing es el enfoque.

Ya que, aunque el Neuromarketing da la apariencia de existir únicamente para cumplir con objetivos lucrativos, no en todos los casos es así, pues una interesante cantidad de investigaciones abordan enfoques sociales, en los que ya sea, se evalúa el impacto de acciones de marketing o se usan técnicas de Neuromarketing para promover hábitos pro-sociales.

Esto es importante porque el Neuromarketing puede ir más allá de los objetivos relacionados con el aumento de ingresos financieros y aportar más a los motivos pro-sociales. Así, aunque en este trabajo se halló evidencia de la existencia de trabajos de investigación con enfoques Sociales, la mayoría de estos son de tipo Comercial.

2. Métodos y materiales

A través de este estudio se pretende concentrar en un documento el progreso logrado de los investigadores de Neuromarketing en aspectos como herramientas y técnicas de procesamiento de datos fisiológicos y los distintos enfoques que se les han dado a los trabajos realizados.

Para esto se planteó una pregunta de investigación en la cual se ven reflejadas las principales motivaciones que llevaron a los autores a realizar este trabajo, esta es: ¿Cuáles son los enfoques, herramientas y técnicas que los trabajos de investigación de Neuromarketing implementan en la actualidad y por qué optan por esas alternativas?

De esta manera, en el actual estudio se tomó como objetivo: Detectar cuales son las tendencias que los estudios de Neuromarketing siguen en cuanto a herramientas, técnicas y enfoques, y porque son elegidos estos.

Es relevante saber cuáles son las metodologías que eligen los investigadores de Neuromarketing y cuál es el enfoque que le desean dar, ya que de esta manera se puede analizar la dirección que las investigaciones siguen, de igual manera, puede hallarse un área de oportunidad sobre lo que hasta el momento no se ha explorado.

Es por eso que adentrarse en los estudios de investigación de la ciencia y detectar cual es el rumbo que está tomando es una cuestión de la cual la permanencia de futuras investigaciones depende. De esta manera, se analizaron trabajos de investigación realizados en los últimos diez años de diferentes bases de datos científicas entre las cuales se encuentran ScienceDirect, Springer Link, ACM y Library Whiley.

Se procuró que los trabajos incluidos en este artículo fuesen publicados por revistas que forman parte del JCR como Human Brain Maping, Journal of Business Research y Physiology & Behavior. Así mismo, se trató de que la mayoría de los trabajos incluidos fuesen artículos del tipo Journal, o en algunos casos capítulos de libros, excluyendo así los informes o presentaciones. Entre los trabajos incluidos la mayoría de ellos son artículos de investigación y un pequeño grupo son revisiones bibliográficas.

A fin de cumplir con el objetivo planteado, los artículos incluidos se distribuyeron en tres clasificaciones que son correspondientes a los tres aspectos que en este artículo se consideran características representativas de la variabilidad en los estudios de Neuromarketing. En la Figura 1, se muestra la clasificación por aspectos de los artículos incluidos. Se puede apreciar que el tipo de técnicas utilizadas es sumamente variable, casi podría decirse que por cada estudio se cuenta con una técnica distinta, esto a excepción de unos cuantos trabajos que coinciden en una técnica aplicada.

Así mismo, no en todos los estudios se diseñó y desarrolló una herramienta, ya que en algunos estudios se optó por aplicar una librería o software previamente programado, los programas implementados en los estudios de este trabajo se visualizan en la esquina superior derecha de la Figura 1 bajo la denominación “software”. En cambio, en el caso de herramientas, se realizó una clasificación de cinco tipos de ellas, donde “combinaciónçorresponde a trabajos en los que se utilizaron más de una técnica, de esta manera, las subclasificaciones son mono herramienta y multi herramienta.

Y en la clasificación Enfoques, solo se crearon dos subclasificaciones: sociales y enfoques comerciales. De esta manera, en base a la clasificación ilustrada en la Figura 1, se creó un subtema especial para mencionar artículos de la recopilación que se hizo en este estudio y que corresponden con características dignas de ser mencionadas en el subtema.

3. Herramientas de evaluación

El Neuromarketing va más allá de los métodos tradicionales de investigación de mercados que se basan en informes verbales y de introspección, ya que a través de las técnicas que se implementan en este, se trata de interpretar el comportamiento de los consumidores; como se dice en [37] se trata de descifrar la caja negra (el cerebro) para cumplir con fines mercadológicos.

Por ejemplo, en [6] dedica un trabajo únicamente a las técnicas de Neuromarketing existentes, entre las cuales se menciona al EEG. Una de las técnicas preferidas del Neuromarketing según [22].

Tabla 1. Herramientas de evaluación y rango de precios.

Herramienta	Rango de precio (USD)
EEG	199 - 1699+
Eye Tracking	300 - 600+
SGR	73 - 140+

Es cierto que muchos estudios han implementado el uso de las nuevas técnicas de investigación de mercados y que se ha dado evidencia de que estas llegan a proporcionar resultados más certeros que las técnicas convencionales, sin embargo, tal indicio nos da a entender que se debe invitar a dejarlas en desuso, pues existen muchos trabajos que utilizan los resultados obtenidos de las técnicas convencionales para reforzar los resultados de las técnicas de análisis fisiológico.

Un estudio que se apoyó del uso de mediciones fisiológicas y mediciones convencionales fue [26] investigó si las herramientas de Neuromarketing hacen un aporte valioso a las investigaciones convencionales, realizaron algunas pruebas evaluando la preferencia de los consumidores ante marcas de cola.

Se analizó el proceso de toma de decisiones midiendo las diferencias en la actividad de la corteza prefrontal en los experimentos con Functional near-infrared spectroscopy (fNIRS) apoyado de cuestionarios en los que los participantes expresaban verbalmente su opinión.

De igual manera, en [17] examinaron los efectos de la ansiedad matemática y el género en las respuestas cerebrales a las elecciones del consumidor. Así mismo, hay estudios que les es suficiente utilizar únicamente los cuestionarios, esto demuestra que no se deben descartar de manera errática los métodos tradicionales pues en situaciones específicas resulta ser más rentable realizar experimentos con autoinformes y no con técnicas de análisis fisiológico.

Por ejemplo [12] proponen que los consumidores que están expuestos a ciertas condiciones de iluminación, están más inclinados a comprar productos hedónicos en lugar de utilitarios. También, existen muchos estudios en los que se combinan diferentes técnicas de medición fisiológica, un ejemplo es [13] donde se examinó la eficacia de los estilos narrativos en publicidad con respecto a términos de cambio de actitud y retención, aplicaron distintas mediciones fisiológicas, las cuales fueron Eye Tracking y Skin Galvanic Respose (SGR).

Un ejemplo en el que se hace uso de técnicas fuera de las herramientas de exploración psicológica es [10] realizaron una investigación para destacar la postura actual del marketing digital y para detectar si hay investigaciones que se han enfocado en el análisis de búsqueda por voz.

Así mismo, [31] utiliza las tecnologías digitales para diseñar un marco basado en el análisis de costos de transacción en el que el marketing contribuye al éxito de las (Data Business Platforms). Abordando los costos, en la Tabla 1 se realiza una comparación entre los precios de las herramientas mencionadas en esta sección. En el caso de [32, 4, 15, 20] realizaron revisiones bibliográficas en las que se revisaron trabajos con aplicación de técnicas innovadoras como estas.

4. Técnicas de procesamiento

Muchos trabajos científicos prefieren desarrollar herramientas computaciones para la realización del procesamiento de los datos obtenidos en sus experimentos, una buena razón para decidir esto es el ajuste que se puede lograr diseñando tal, así como los costos. En [30] investigaron cuales son los procesos neuronales o psicológicos involucrados en el Willingness to Pay (WTP) en productos varios.

donde las señales EEG obtenidas por los experimentos se procesaron evaluando la asimetría prefrontal, utilizando una librería de MATLAB llamada EEGLAB. También [38] exploró diferentes técnicas de la teoría estadística por medio de registro de actividad EEG y algunos de los procedimientos utilizando el software LORETA, una herramienta pre-producida.

Los softwares pre-producidos no son únicos para el EEG, un software muy conocido es SPM, que proporciona un esquema de análisis de neuroimágenes, es una distribución libre, pero con derechos de autor.

En [8] se hizo uso de este software para el procesamiento de Functional magnetic resonance imaging (fMRI) obtenidas de la exploración de las respuestas neuronales ante descripciones de productos del tipo funcionales y del tipo experienciales en anuncios publicitarios. Este mismo software, SPM, fue utilizado en [28].

En el caso de [35] tuvieron por objetivo evaluar si las marcas, en oposición a los productos sin nombre, están asociadas con actitudes positivas implícitas. Esto los llevo a realizar experimentos dónde las señales EEG fueron analizadas con Event-Related Potentials (ERP).

Otro ejemplo es [36] en el que se analizó el impacto que los anuncios de T.V. interactivos tienen sobre la atención de los espectadores y por lo que se descubrió la utilidad que tiene el pico de frecuencia P3a en la implementación de Neuromarketing. También se aplicó el ERP análisis en [9] que estudió la similitud percibida entre diferentes productos alimenticios, por lo que probó dos medidas neurometrías potenciales, la amplitud de los potenciales de respuesta evocada (ERP) y la potencia de las oscilaciones gamma.

Así mismo, en [39] analizaron el efecto del contenido atractivo en el comportamiento de clics de los consumidores y exploraron cómo el zumbido temporal, identificado como la P3 del ERP y el Late positive potential (LPP), indican el compromiso del consumidor.

También existen estudios que están interesados en medir la pertinencia del uso de dos tipos de mediciones como [27] investigaron las posibles oportunidades que ofrecen los métodos explícitos y no explícitos de manera correlacional para comprender las expectativas de los consumidores de vino al registrar medidas indicativas de sus emociones. Y analizaron las señales por medio del análisis de valencia.

También en [5] se analizó la correlación de métodos convencionales y fisiológicos, de tal manera que aplicaron EEG para evaluar una serie de comerciales de televisión. Así mismo, otros estudios utilizaron procedimientos diferentes como [24] que creó un algoritmo de clasificación para medir lo que nos gusta y lo que no nos gusta, a través de una prueba de electroencefalograma (EEG), utilizando redes neuronales.

En [21] utilizaron señales EEG para sus estudios y procesaron la señal por medio de la regresión lineal. En cambio, en [7] utilizaron fMRI y procesaron los datos por medio del mapeo estadístico paramétrico.

5. Enfoques

Los objetivos del Neuromarketing no son forzosamente con miras a lucrar con los hallazgos obtenidos en las investigaciones, pues el estudio de los efectos de las neurociencias y sobre todo de las técnicas de Neuromarketing puede ser una clave para la conservación de una sociedad con salud física y mental. [29] específicamente se dedicó a analizar la influencia de esta ciencia en la percepción que tienen las personas sobre sí mismas.

También [25] realizan experimentos con fMRI para evaluar el comportamiento de infantes frente a la invitación de ingerir alimentos con poco valor nutricional, así como alimentos nutritivos. Este estudio invita a que las marcas generen una mayor conciencia pues hasta el día de hoy difunden campañas publicitarias dirigidas a los infantes sin ningún filtro.

Otro trabajo es el de [19], donde por medio del análisis de electroencefalogramas (EEG) se estudió la correlación existente entre actitudes pro sociales y el aumento de la disposición a pagar. Estos hallazgos pueden despertar el interés de distintas compañías por volverse en una marca con responsabilidad social o por lanzar productos con causas sociales. Así mismo, en [14] Se realizó un tipo de estudio que puede conducir a una mejor comprensión del comportamiento de los consumidores y facilitar una comercialización exitosa de las energías renovables.

Si bien es cierto, que existen estudios como los recientemente mencionados, que apoyan causas sociales, también existen otros estudios que se dedican a alcanzar objetivos puramente comerciales. Como es el caso de [16] que estudió la relación entre los procesos mentales y corporales para aumentar el conocimiento actual sobre la capacidad de los mensajes subliminales a fin de influir en las selecciones de los consumidores. O [11] que estudió el efecto del color de fondo en las preferencias de publicidad de los clientes.

De igual manera, sucede con [23] en donde se realizan pruebas para examinar las preferencias de los consumidores ante la modificación de empaques de imágenes, textos o colores. En [18] se analizan los procesos neuronales involucrados en la toma de decisiones frente a juegos de economía.

Claramente en estos se estudia el comportamiento humano a fin de asegurar el éxito de los esfuerzos mercadológicos aplicados, lo cual hasta cierto punto es válido porque a través del éxito de la implementación de nuevas técnicas estas se validan y consolidan, sin embargo, es cierto que una regulación de estas técnicas es necesaria por el bien de la sociedad.

6. Implicaciones éticas

Como adición a la existencia de trabajos de investigación con enfoques sin fines de lucro, también es cierto que a la fecha se ha trabajado duramente en promover la

inclusión de un criterio ético en los estudios y en desarrollar recomendaciones precisas y prácticas.

Por ejemplo, en [22] se hace mención de algunas de las preocupaciones éticas que se tienen sobre la aplicación de Neuromarketing, y se plantean soluciones a estas como lo son formular políticas y establecer comités de ética, y participar en aprendizaje y prácticas responsables.

De igual manera, se sustenta que el neuromarketing ha contribuido a profundizar el conocimiento que se tiene sobre el comportamiento humano. Este podría ser el caso del estudio [33] en el que se realiza una revisión de trabajos que se caracterizan por tener resultados útiles para comprender el comportamiento de las personas frente a productos de consumo alimenticio.

Es cierto que es necesario preocuparse por la dirección que las innovaciones tecnológicas siguen, sin embargo, los estudios emergentes que trabajan arduamente en la difusión de prácticas éticas, demuestran que, así como las técnicas de Neuromarketing se vuelvan más sofisticadas cada vez, también las regulaciones éticas y legales se vuelvan más precisas y transparentes. Un ejemplo de estudios que trabajan en esto es [34].

Otros trabajos como [37] que remarcan la importancia de asegurar el respeto a la privacidad y la inclusión del consentimiento informado por parte de los participantes de experimentos de Neuromarketing. Así mismo, se insta a los autores de trabajos de Neuromarketing a presentar sus hallazgos en términos más legibles de tal manera que la sociedad pueda tener un criterio más certero sobre las implicaciones de las investigaciones.

Sin dudarlo las implicaciones éticas en los estudios de Neuromarketing son retos que los trabajados actuales y futuros enfrentan y seguirán enfrentando. Pues las practicas éticas son indispensables para asegurar un futuro para el Neuromarketing. Así mismo, los derechos de los participantes implicados en los estudios de Neuromarketing son un tema que no puede dejarse desapercibido y que merece un manejo cuidadoso y transparente.

7. Discusión

Así como fue mencionado anteriormente, el tipo de técnicas de procesamiento de señales que se implementan en los estudios son sumamente variables, esto se comprueba a través de la muestra de estudios elegidos, en los que prácticamente se halla una técnica diferente para cada estudio.

Es por ello que se decidió utilizar una gran cantidad de artículos en esta sección para así presentar el comportamiento que los estudios de Neuromarketing manifiestan, que como se puede notar es muy diverso. De esta manera en la Figura 2, se puede ver que casi el 50 % de los artículos incluidos en este trabajo se introdujeron en la sección de Técnicas, esto a fin de visualizar la variedad de las técnicas.

Algo muy diferente sucedió con las herramientas, donde se usó una cantidad un poco más grande de artículos que en las técnicas, pero con muchas menos variantes, pues solo se hallaron 6 subclasificaciones en esta clasificación.

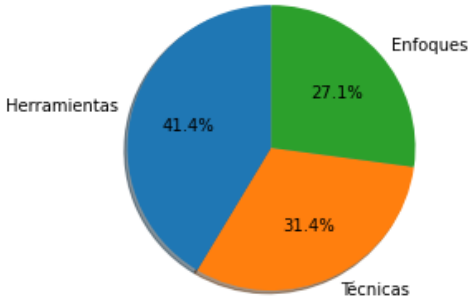


Fig. 2. Clasificación de investigaciones.

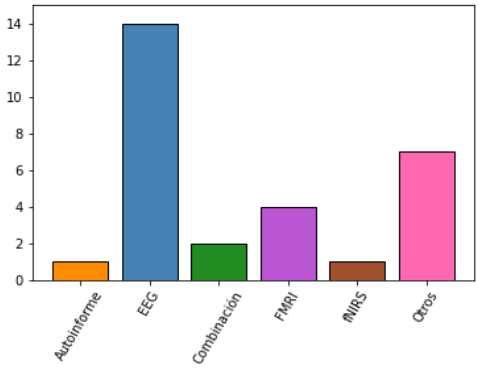


Fig. 3. Herramientas de Neuromarketing.

Estas fueron Autoinformes, EEG, Combinación, FMRI, fNIRS y Otros, donde en la clase combinación se incluyeron estudios que hicieron uso de varias herramientas para las evaluaciones; y en la clase Otros se trata de estudios en los que no usaron técnicas como los autoinformes o el seguimiento de bioseñales, pero si se realizaron Herramientas como el procesamiento del lenguaje natural.

De esta manera, en la Figura 3, se puede visualizar como el preferido definitivo es el EEG con un total de 14 estudios que lo implementaron, lo que es prácticamente la misma cantidad de estudios que utilizan el resto de las herramientas detectadas en este trabajo juntas. La notable supremacía del EEG se puede deber a que, a diferencia de las demás herramientas, esta es relativamente económica, portable, y con una muy buena definición que la hace ser bastante potente.

En el caso de las técnicas de procesamiento de datos, se puede mencionar que existen trabajos de investigación en los que se diseña e implementa el procesamiento, de tal manera que el modelo se adapta a los datos y a la metodología de los experimentos. Sin embargo, es notable que no todos los estudios tienden a proceder de esta manera, pues hay quienes optan por hacer uso de software prediseñado.

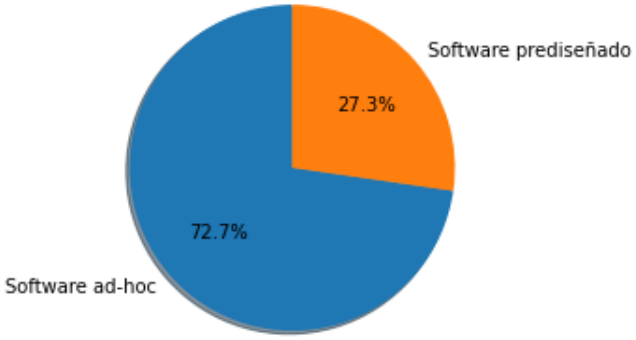


Fig. 4. Tipos de procesamiento.

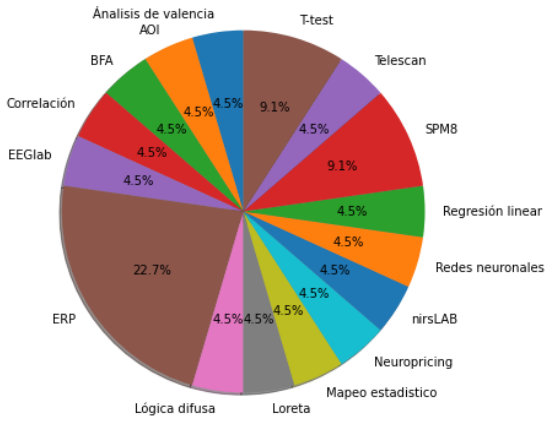


Fig. 5. Técnicas de procesamiento.

Como se muestra en la Figura 4, la mayoría de los artículos incluidos en este trabajo hacen uso de software ad hoc, es decir, que construyen desde cero un sistema que se adapte a las exigencias de la investigación, esto se puede deber a que, para realizar su programación se puede hacer uso de software libre y a que existe un mejor ajuste entre los datos y el modelo de procesamiento.

Con respecto al análisis de la cantidad de estudios que pertenecen a cada técnica de procesamiento, en la Figura 5 se puede apreciar cómo es que el ERP (Event-related potential), es la técnica preferida en estudios de Neuromarketing, lo cual sucede así pues esta técnica se utiliza para procesar datos EEG, que son obtenidos por la herramienta que se encuentra en la mayoría de los estudios incluidos en este trabajo.

Sin embargo, también es muy utilizada porque por medio de esta se puede analizar el estado mental en una ventana de tiempo en la que se incluye la exposición que el participante del experimento tiene ante el estímulo de marketing.

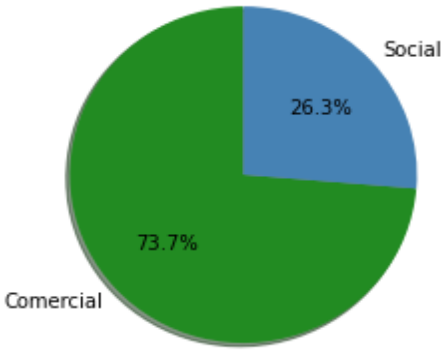


Fig. 6. Tipos de enfoques.

De igual manera, en el caso de los enfoques estudiados en este trabajo, como ya se mencionó fueron el comercial y el social, y a través de la muestra de investigaciones que se manejó es evidente que la mayoría de los trabajos de investigación del área de Neuromarketing tienden a tener un enfoque con fines de lucro, en este caso, como se visualiza en la Figura 6, el 73.7 % de los trabajos incluidos en la sección de Enfoques tiene un carácter Comercial, mientras que el 26.3 % de los trabajos son con enfoque Social.

Esto es completamente entendible pues las empresas desean incrementar sus ingresos, y mientras a través de las técnicas de Neuromarketing esto pueda ser posible, habrá muchos más trabajos con enfoques como estos.

Sin embargo, al día de hoy con la cantidad de procedimientos de Neuromarketing que se están aplicando en campañas publicitarias, imagen de las marcas y embalaje, se comienza a hacer necesario el dar un correcto seguimiento al crecimiento de estos proyectos y la implementación de medidas que resguarden la salud mental y física de los individuos que forman parte de la sociedad.

8. Conclusión

Es claro que existe una gran diversidad no solo en los tipos de estudios de Neuromarketing sino en las técnicas utilizadas, el tipo de procesamiento elegido y los enfoques manejados. Evidencia de esto es el análisis realizado en este trabajo, que, de igual manera, permite responder a la pregunta de investigación planteada, ya que se descubrió cual fue la alternativa más elegida en cuanto a Herramientas, Técnicas y Enfoques. Con respecto a las técnicas de Neuromarketing empleadas, los estudios incluidos en este trabajo, prefirieron el electroencefalograma (EEG), seguido de las técnicas varias, llamadas ‘Otros’ y del FMRI.

Ahora bien, los resultados de las Técnicas halladas en este trabajo, evidencian que la elección de Técnicas y Herramientas esta fuertemente ligada, ya que la técnica mayormente hallada fue el ERP, utilizada para procesar datos EEG.

De igual manera, una de las técnicas que ocupan el segundo lugar, el software SPM8, se usa para procesar datos de FMRI, una de las Herramientas más halladas en esta revisión; y en el caso del t-test, es una Técnica adaptable a distintos tipos de datos, por lo que no es extraño verle en ese lugar.

Como se puede ver, el tipo de datos que se usan en un estudio experimental es un gran factor que influye en la elección de la Técnica, sin embargo, como se ve en la Tabla1, los costos pueden llegar a afectar dicha elección. Por último, con respecto a los enfoques hallados, es bastante claro que la mayoría de trabajos maneja un enfoque comercial, y que el enfoque menos empleado es el social.

Lo cual es comprensible por las consecuencias positivas que los hallazgos producen en la economía, sin embargo, es importante promover en todos estos estudios, la inclusión de buenas prácticas como la aprobación de los comités de ética, el consentimiento informado por parte de los participantes de experimentos y la divulgación científica en términos más coloquiales.

Para futuras investigaciones se recomienda la realización de una detallada investigación sobre las herramientas de Neuromarketing que se desean implementar en los experimentos, así como la comprobación de que exista un verdadero ajuste entre los objetivos del estudio y las cualidades de tal herramienta. De esta manera será posible conocer el tipo de datos que se manejaran y podrá determinarse la metodología que se aplicará en los experimentos para saber con exactitud el tipo de análisis que se necesita.

Es evidente, que los fines lucrativos son los que más atraen a las investigaciones a implementar el Neuromarketing, sin embargo, como se mencionó, es importante dar un detallado seguimiento legal y ético de las actividades a las que los participantes de los experimentos estarán expuestos, así mismo, se alienta a que los investigadores de Neuromarketing trabajen en estudios en los que se analice el estado de la sociedad ante la utilización de herramientas de Neuromarketing y de igual manera, se utilicen estas para promover practicas positivas para la sociedad.

Referencias

1. Armstrong, G., Kotler, P.: Marketing: An introduction. Person Educación (2013)
2. Crescitelli, E., Colaferro, A. C. : A Contribuição do neuromarketing para o estudo do comportamento do consumidor. Brazilian Business Review, vol. 11, no. 3, pp. 130–156 (2014)
3. Braidot, N. P., Annecchini, P. B.: Neuroventas: Conozca cómo funciona el cerebro para vender con inteligencia y resultados exitosos. Editorial Granica (2013)
4. Amado, A., Cortez, P., Rita, P., Moro. S.: Research trends on big data in marketing: A text min-ing and topic modeling based literature analysis. European Research on Management and Business Economics, vol. 24, no. 1, pp. 1–7 (2018) doi: 10.1016/j.iedeen.2017.06.002
5. Bertin, M., Tokumi, R., Yasumatsu, K., Kobayashi, M., Inoue, A.: Application of EEG to TV commercial evaluation. In: International Conferenceon Biomedical Informatics and Technology, Springer, vol. 404, pp. 277–282 (2013) doi: 10.1007/978-3-642-54121-6_24
6. Boz, H., Arslan, A., Koc, E.: Neuromarketing aspect of tourısm pricing psychology. Tourism Management Perspectives, vol. 23, pp. 119–128 (2017) doi: 10.1016/j.tmp.2017.06.002
7. Casado-Aranda, L. A., Van der Laan, L. N., Sánchez-Fernández, J.: Neural correlates of gender congruence in audiovisual commercials for gender-targeted products: An fMRI study. Human Brain Mapping, vol. 39, no. 11, pp. 4360–4372 (2018) doi: 10.1002/hbm.24276

8. Couwenberg, L. E.,. Boksem, M. A, Dietvorst, R. C., Worm, L., Verbeke, W. J., Smidts. A.: Neural responses to functional and experiential ad appeals: Explaining ad effectiveness. International Journal of Research in Marketing, vol. 34, no. 2, 355–366 (2017) doi: 10.1016/j.ijresmar.2016.10.005
9. Domracheva, M., Kulikova, S.: EEG correlates of perceived food product similarity in a cross-modal taste-visual task. Food Quality and Preference, vol. 85, pp. 103980 (2020) doi: 10.1016/j.foodqual.2020.103980
10. Dumitriu, D., Popescu, M. A.: Artificial intelligence solutions for digital marketing. Procedia Manufacturing, vol.46, pp. 630–636 (2020) doi: 10.1016/j.promfg.2020.03.090
11. Golnar-Nik, P., Farashi, S., Safari, M. S.: The application of EEG power for the prediction and interpretation of consumer decision-making: A neuromarketing study. Physiology & Behavior, vol. 207, pp. 90–98 (2019) doi: 10.1016/j.physbeh.2019.04.025
12. Guido, G., Piper, L., Prete, M. I., Mileti, A., Trisolini. C. M.: Effects of blue lighting in ambient and mobile settings on the intention to buy hedonic and utilitarian products. Psychology & Marketing, vol. 34, no. 2, pp. 215–226 (2017) doi:10.1002/mar.20984
13. Hamelin, N., Thaichon, P., Abraham, C., Driver, N., Lipscombe, J., Pillai, J.: Storytelling, the scale of persuasion and retention: A neuromarketing approach. Journal of Retailing and Consumer Services, vol. 55, pp. 102099 (2020) doi: 10.1016/j.jretconser.2020.102099
14. Herbes, C., Friege, C., Baldo, D., Mueller, K. M.: Willingness to pay lip service? Applying a neuroscience-based method to WTP for green electricity. Energy Policy, vol. 87, pp. 562–572 (2015) doi: 10.1016/j.enpol.2015.10.001
15. Herhausen, D., Miočević, D., Morgan, R. E., Kleijnen. M. H.: The digital marketing capabilities gap. Industrial Marketing Management, vol. 90, pp. 276–290 (2020) doi: 10.1016/j.indmarman.2020.07.022
16. Hsu, L., Chen, Y. J.: Neuromarketing, subliminal advertising, and hotel selection: An EEG study. Australasian Marketing Journal, vol. 28, no. 4, pp. 200–208 (2020) doi: 10.1016/j.ausmj.2020.04.009
17. Jones, W. J., Childers, T. L., Jiang, Y.: The shopping brain: Math anxiety modulates brain re-sponses to buying decisions. Biological Psychology, vol. 89, no. 1, pp. 201–213 (2012) doi: 10.1016/j.biopsycho.2011.10.011
18. Lee, V. K., Harris, L. T.: How social cognition can inform social decision making. Frontiers in Neuroscience, vol. 7, pp. 259 (2013) doi: 10.3389/fnins.2013.00259
19. Lee, E. J.: Empathy can increase customer equity related to pro-social brands. Journal of Business Research, vol. 69, no. 9, pp. 3748–3754 (2016) doi: 10.1016/j.jbusres.2015.05.018
20. Leeflang. P.: Paving the way for distinguished marketing. International Journal of Research in Marketing, vol. 28, no. 2, pp. 76–88 (2011) doi: 10.1016/j.ijresmar.2011.02.004
21. Li, H., Wang, D., Chen, J., Luo, X., Li, J., Xing, X.: Preservice fatigue screening for constructionworkers through wearable EEG based signal spectral analysis. Automation in Construction, vol. 106, pp. 102851 (2019) doi: 10.1016/j.autcon.2019.102851
22. Lim, W. M.: Demystifying neuromarketing. Journal of Business Research, vol. 91, pp. 205–220 (2018) doi: 10.1016/j.jbusres.2018.05.036
23. García-Madariaga, J., López, M. F. B., Burgos, I. M., Virto, N. R.: Do isolated packaging variables influence consumers' attention and preferences? Physiology & behavior, vol. 200, pp. 96–103 (2019) doi: 10.1016/j.physbeh.2018.04.030
24. Márquez, B. Y., Palencia, J. S. M., Garza, A. A., Alvarado, K. R.: Neural network slgorithm to measure the human emotion. In: Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence, pp. 1–5 (2018) doi: 10.1145/3302425.3302441
25. Masterson, T. D., Stein, W. M., Beidler, E., Bermudez, M., English, L. K., Keller, K. L.: Brain response to food brands correlates with increased intake from branded meals in children: An

fMRI study. Brain Imaging and Behavior, vol. 13, no. 4, pp. 1035–1048 (2019) doi: 10.1007/s11682-018-9919-8
26. Meyerding, S. G., Mehlhose, C. M.: Can neuromarketing add value to the traditional marketing research? An exemplary experiment with functional near-infrared spectroscopy (fNIRS). Journal of Business Research, vol. 107, pp. 172–185 (2020) doi: 10.1016/j.jbusres.2018.10.052
27. Mignani, C., Matteo, V., Bailetti, L. I., Bonfini, M., Cavicchi, A.: Emotion in the glass: Aninnovative study to understand unconscious reactionsin wine tasting. Case Studies in the Wine Industry, pp. 101–114 (2019) doi: 10.1016/B978-0-08-100944-4.00008-2
28. Mostafa, M. M.: Brain processing of vocalsounds in advertising: A functional magnetic resonanceimaging (fMRI) study. Expert Systems with Applications, vol. 39, no. 15, pp. 12114–12122 (2012) doi: 10.1016/j.eswa.2012.04.003
29. O'Connor, C.: Science in society: Neuroscience and lay understandings of self and identity. Neuroimaging Personality, Social Cognition, and Character, pp. 371–382 (2016) doi: 10.1016/B978-0-12-800935-2.00020-8
30. Ramsøy, T. Z., Skov, M., Christensen, M. K., Stahlhut, C.: Frontal brain asymmetry and willingness to pay. Frontiers in Neuroscience, vol. 12, pp. 138 (2018) doi: /10.3389/fnins.2018.00138
31. Rangaswamy, A., Moch, N., Felten, C., Van-Bruggen, G., Wieringa, J. E., Wirtz, J.: The role of marketing in digital business platforms. Journal of Interactive Marketing, vol. 51, pp. 72–90 (2020) doi: 10.1016/j.intmar.2020.04.006
32. Shah, D., Murthi, B. P. S: Marketing in a data-driven digital world: Implications for the role and scopeof marketing. Journal of Business Research, vol. 125, pp. 772–779 (2020) doi: 10.1016/j.jbusres.2020.06.062
33. Songsamoe, S., Saengwong-ngam, R., Koomhin, P., Matan, N.: Understanding consumer physiological and emotional responses to food products using Electroencephalography (EEG). Trends in Food Science & Technology, vol. 93, pp. 167–173 (2019) doi: 10.1016/j.tifs.2019.09.018
34. Stanton, S. J., Sinnott-Armstrong, W., Huettel, S. A: Neuromarketing: Ethical implications of its use and potential misuse. Journal of Business Ethics, vol. 144, no. 4, pp. 799–811 (2017) doi: 10.1007/s10551-016-3059-0
35. Thomas, A., Hammer, A., Beibst, G., Münte, T. F.: Un estudio ERP de productos de marca y sin nombre. Neurociencia de BMC, vol. 14, no. 1, pp. 1–9 (2013)
36. Treleaven-Hassard, S., Gold, J., Bellman, S., Schweda, A., Ciorciari, J., Critchley, C., Varan, D.: Using the P3a to gauge automatic attention to interactive television advertising. Journal of Economic Psychology, vol. 31, no. 5, pp. 777–784 (2010) doi: 10.1016/j.joep.2010.03.007
37. Ulman, Y. I., Cakar, T., Yildiz, G.: Ethical issues in neuromarketing: I consume, therefore I am! Science and Engineering Ethics, vol. 21, no. 5, pp. 1271–1284 (2015) doi: 10.1007/s11948-014-9581-5
38. Vecchiato, G., Fallani, F. D. V., Astolfi, L., Toppi, J., Cincotti, F., Mattia, D., Salinari, S., Babiloni, F.: The issue of multiple univariate comparisonsin the context of neuroelectric brain mapping: An application in a neuromarketing experiment. Journal of Neuroscience Methods, vol. 191, no. 2, pp. 283–289 (2010) doi: 10.1016/j.jneumeth.2010.07.009
39. Zhang, J., Yun, J. H., Lee, E. J.: Brain buzzfor facebook? Neural indicators of SNS content engagement. Journal of Business Research, vol. 130, pp. 444–452 (2020) doi: 10.1016/j.jbusres.2020.01.029

ISSN 1870-4069

Sistema de detección de intrusos en el CAN bus de automóviles utilizando una red neuronal multicapa

Juan Carlos Venegas-Segura[1], Ponciano Jorge Escamilla-Ambrosio[1], Alberto Jorge Rosales-Silva[2], Floriberto Ortiz-Rodríguez[2], Narciso Muñoz-Aguirre[3]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, México

[2] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, México

[3] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Azcapotzalco, México

Karlvsk16@gmail.com,{pescamillaa,arosaless, flortiz, nmunoz}@ipn.mx

Resumen. La constante evolución tecnológica en los vehículos ha traído consigo un sin número de ventajas permitiéndole brindar mejores servicios. No obstante, se ha demostrado en múltiples investigaciones que los vehículos son susceptibles a sufrir ataques de ciberseguridad. En consecuencia, en este trabajo se propone un sistema de detección de intrusos en el CAN bus de un automóvil utilizando una red neuronal multicapa. Se realizó un proceso de experimentación para encontrar la estructura de la red neuronal que realice de manera satisfactoria la tarea en cuestión. Esto incluyó experimentar con el número de capas ocultas y el número de neuronas en dichas capas. La red neuronal resultante se encarga de detectar tráfico malicioso en el CAN bus, permitiendo identificar datos libres de ataque y datos con algún tipo de ataque. Los ataques que se consideran en este trabajo corresponden a los ataques tipo denegación de servicio, suplantación de identidad e inyección de datos aleatorios. Se demuestra de manera experimental que, una vez encontrada la estructura adecuada y entrenada la red neuronal propuesta, ésta es capaz de identificar los ataques antes mencionados.

Palabras clave: Ciberseguridad, sistema de detección de intrusos (IDS), red neuronal multicapa, CAN bus, electronic control unit (ECU).

Intrusion Detection System in the CAN bus of Automobiles Using a Multilayer Neural Network

Abstract. The constant technological evolution in vehicles has brought with it several advantages that allow it to provide better services. However, it has been shown in multiple investigations that these vehicles are susceptible to

cybersecurity attacks. Consequently, this work proposes an intrusion detection system in the CAN bus of automobiles using a multilayer neural network that is responsible of protecting this type of networks. An experimental process was carried out in order to find the structure of the neural network to obtain satisfactory results for the task at hand. This included experimentation with the number of hidden layers and the number of neurons on such layers. The neural network found detects malicious traffic on the CAN bus, allowing to discriminate free-attack data and data with some type of attack. The attacks considered in this work correspond to denial of service, impersonation and fuzzy. It is shown that, once the neural network is trained, it can identify the attacks previously mentioned.

Keywords: Cybersecurity, intrusion detection system (IDS), multilayer neural network, controller area network (CAN), electronic control unit (ECU).

1. Introducción

El uso de tecnología en los vehículos ha mejorado muchos procesos en los cuales los dispositivos mecánicos utilizados tenían cierta desventaja debido al tiempo de respuesta que necesitaban o incluso al desgaste que sufrían, lo que en ocasiones provocaba algunas descomposturas.

Con la intervención de la electrónica y la introducción de computadoras en los vehículos, que en este contexto se conocen como ECU (Electronic Control Unit, en inglés), se logró un avance tecnológico en la industria automotriz. La aparición del CAN bus (del inglés, Controller Area Network) permitió reducir la cantidad de cableado que se requería para mantener conectado cada uno de los dispositivos dentro de un vehículo.

Hoy en día, el CAN bus se ha convertido en el principal protocolo de comunicación en cualquier vehículo. Sin embargo, múltiples investigaciones como las realizadas por los hackers Charli Miller y Chris Valasek [1], demuestran que tienen una gran cantidad de vulnerabilidades, permitiéndole a un atacante ocasionar algún mal funcionamiento o incluso tomar el control de algún sistema del vehículo.

En este trabajo se presenta una investigación relacionada a los problemas de ciberseguridad que se tienen en el CAN bus de los automóviles actuales. Se presenta una solución que utiliza una red neuronal multicapa para la detección de tres tipos de ataque, además del estado sin ataque. Los ataques que se consideran en este trabajo corresponden a los ataques tipo denegación de servicio (DoS, Denial of Service, en inglés), suplantación de identidad (impersonation, en inglés) e inyección de datos aleatorios (fuzzy attack). El presente trabajo se encuentra estructurado de la siguiente manera.

En la Sección 2 se presenta un resumen de los trabajos relacionados. En la Sección 3 y Sección 4 se presentan algunas consideraciones técnicas que fueron tomadas en el desarrollo de este trabajo.

Finalmente, en la Sección 5 se presenta el proceso que se siguió para encontrar una estructura adecuada de una red neuronal que permita dar solución al problema planteado, donde se determinó el número de capas ocultas, así como el número de neuronas que tendrá cada una de ellas.

2. Trabajos relacionados

Hoy en día, existen muchos trabajos de investigación que proponen un sistema de seguridad para la protección de la red de un vehículo. Muchas de estas investigaciones se enfocan en el intercambio de información a través del CAN bus, en especial la frecuencia con la que cada computadora en un vehículo (ECU) envía mensajes a través de éste. Se ha demostrado que cada ECU envía mensajes de forma periódica y, por lo tanto, esta característica se verá alterada durante un ataque.

Centrándose en trabajos que proponen un sistema de detección de intrusos (IDS, Intrusion Detection System, en inglés) usando herramientas del área de aprendizaje de maquina (Machine Learning, en inglés) como son las redes neuronales, se encuentra el trabajo realizado por Hyung Ming Song et al. [2] quienes proponen un IDS basado en una red convolucional profunda (Deep Convolutional Neural Network, en inglés) reportando una exactitud del 99%. Min-Joo Kang et al. [3] proponen un IDS usando una red neuronal profunda (Deep Neural Network, en inglés) reportando una exactitud del 98 %. Por su parte Hyung Ming Song et al. [4] proponen un IDS basado en una Red GAN (Generative Adversarial Networks, en inglés) y reportan una exactitud del 98 %.

A diferencia de los trabajos anteriores, en la presente investigación se realiza un proceso de experimentación en el cual se determina el número de capas ocultas y el número de neuronas que tendrá cada una de ellas. Esto se realiza con el objetivo de encontrar la estructura de una red neuronal lo más simple posible con el fin de reducir la complejidad de la RN, pero que resuelva el problema en cuestión sin la necesidad de emplear una red neuronal compleja como las reportadas en la literatura. Asimismo, se utilizan los datos de entrada en un formato de ventana deslizante lo que permitirá llevar a cabo un proceso de clasificación más rápido y en tiempo real.

3. CAN bus

El CAN bus [16] es una representación de un bus serial diseñado en 1986 por la compañía alemana Robert Bosh, para su uso en el desarrollo industrial y su aplicación en vehículos. El protocolo CAN está basado en una topología bus para la transmisión de mensajes en tiempo real y entornos distribuidos y que ofrece una gestión a la comunicación entre múltiples unidades de control.

El protocolo CAN soporta frecuencias de datos de hasta de 125 kbps, con una distancia de hasta 40m, el cual provee una mayor capacidad de transferencia de datos y rango extendido de transmisión, Además, el CAN puede soportar tasas de transferencia de hasta 1 Mbps.

El protocolo CAN contempla cuatro tipos de mensajes: mensaje con datos (data frame), mensaje de solicitud remota (remote frame), mensaje de error (error frame) y mensaje de sobrecarga (overload frame).

La arquitectura del CAN bus permite llevar a cabo el proceso de comunicación entre las diferentes unidades de control, permitiéndoles identificar a cada nodo (a cada ECU) solo aquellos mensajes que le pudieran servir.

Desafortunadamente, el protocolo CAN fue diseñado sin contar con algún sistema de protección, por lo que cada mensaje que circula por el mismo es tomado como verídico. En consecuencia, si algún mensaje falso es procesado por alguna ECU, se

llevará a cabo alguna acción que pudiera provocar alguna avería, un mal funcionamiento o incluso alguna situación catastrófica**.**

4. Redes neuronales

En general no existe una sola definición de red neuronal, y ésta puede variar de acuerdo con el texto o al artículo consultado. Sin embargo, a grandes rasgos se puede decir que una red neuronal es un modelo computacional compuesto de unidades procesadoras que se encuentran interconectadas entre sí. Estos sistemas buscan emular el comportamiento del cerebro humano con el fin de solucionar múltiples problemas, uno de ellos es el de clasificación.

En el presente trabajo se plantea dar solución a los problemas de ciberseguridad en el CAN bus que tienen los vehículos hoy en día. Como se ha mencionado antes, es posible que algún agente atacante pueda realizar diversos ataques a una red vehicular permitiéndoles llevar a cabo diferentes tareas.

Se plantea utilizar una red neuronal multicapa que permita dar solución al problema antes planteado. El proceso para desarrollar la red neuronal será a través de experimentos que permitan encontrar las principales características que conforman a la red neuronal: optimizadores, datos de entrada, número de capas ocultas, número de neuronas en cada capa intermedia, entre otros**.**

4.1. Arquitectura de una red neuronal

Se denomina arquitectura a la topología o estructura de una red neuronal. En una red neuronal artificial los nodos se conectan por medio de sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. En general los nodos (neuronas) se suelen agrupar en unidades estructurales que se denominan capas. El conjunto de una o más capas constituye la red neuronal.

Se distinguen tres tipos de capas: de entrada, de salida y ocultas. Una capa de entrada, también denominada sensorial, está compuesta por neuronas que reciben datos o señales procedentes del entorno. Una capa de salida se compone de neuronas que proporcionan la respuesta de la red neuronal. Las capas ocultas [17] son internas a la red y no tienen contacto directo con el entorno exterior.

4.2. Red neuronal multicapa

Una de las primeras redes que aparecieron a lo largo de la historia fue el perceptrón, la unidad básica desde donde nacería y se potenciarían las redes neuronales artificiales. El perceptrón es la forma más simple de una red neuronal usada para la clasificación de un tipo especial de patrones, los linealmente separables (es decir, patrones que se encuentran a ambos lados de un hiperplano). Básicamente, consiste en una neurona con pesos sinápticos y umbral ajustable.

La red neuronal multicapa surge como respuesta a las limitaciones que poseía un perceptrón simple, ya que con este solo se podían discriminar patrones que pudieran ser separados por un hiperplano - una recta. La forma de solventar estas limitaciones

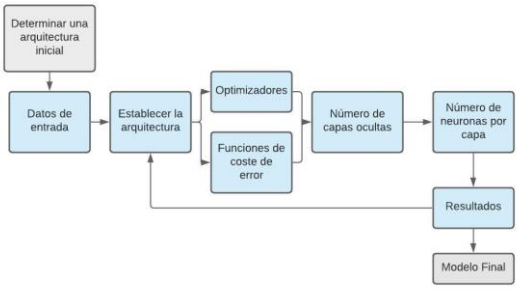


Fig. 1. Diagrama del proceso seguido para el desarrollo de la red neuronal.

fue a través de la inclusión de capas ocultas, permitiendo solucionar problemas que hasta ese momento no eran linealmente separables.

El perceptrón multicapa o MLP (Multi-Layer Perceptron, en inglés) se suele entrenar por medio de un algoritmo de retropropagación de errores o BP (Back Propagation, en inglés) de ahí que dicha arquitectura se conozca también bajo el nombre de red de retropropagación.

La popularidad de la arquitectura MLP se debe al hecho de que un MLP con una única capa oculta puede aproximar cualquier función continua en un intervalo hasta el nivel deseado, y que proporciona una base sólida al campo de las redes neuronales.

Los modelos con "profundidad" (decenas o cientos de capas) como las redes neuronales profundas han demostrado que pueden dar solución a un gran número de tareas, sin embargo, debido a su alta complejidad y altos requerimientos computacionales pueden ser demasiado difíciles de entrenar.

5. Desarrollo de la estructura de la red neuronal para ciberseguridad en el CAN bus

Definir la estructura de una red neuronal puede ser una tarea laboriosa debido a la falta de un protocolo o alguna estrategia establecida, la cual especifique que estructura se debe utilizar para cada problema bajo consideración (como el número de capas ocultas y el número de neuronas en cada capa).

En muchas ocasiones, esto se puede deducir mediante experimentos de prueba y error o incluso a través de la experiencia que pudiera tener el programador.

Para validar la estructura de la red neuronal que se utilizó en este trabajo, se realizaron una serie de experimentos, en los cuales se modificaron algunas características propias de la red neuronal**.**

En la Figura 1 se muestran las etapas seguidas para el desarrollo de la red neuronal propuesta en este trabajo. El proceso de validación de la red neuronal que se siguió consiste en realizar experimentos con el fin de obtener una arquitectura adecuada que proporcione buenos resultados. En este caso se modificó el número de capas, el número de neuronas, las funciones de coste de error y los optimizadores que intervienen en la arquitectura de una red neuronal**.**

Tabla 1. Número de mensajes que contiene los conjuntos de datos.

Conjunto de datos	No. de mensajes
Libre de ataques	2,369,868
Ataque tipo DoS	656,579
Ataque tipo Fuzzy	591,990
Ataque tipo Impersonation	995,472

En primera instancia, se determinó una arquitectura inicial con una sola capa oculta y sus correspondientes capas de entrada y de salida. Posteriormente se establecieron los conjuntos de datos que se utilizarían para alimentar a la red neuronal, así como la estructura que tendrían esto.

Una vez formada la red neuronal inicial ya con los datos de entrada establecidos, se comenzó a experimentar con las funciones de coste de error, así como con los optimizadores que forman parte de una red neuronal. En la literatura se han documentado muchas funciones, sin embargo, solo se eligieron las que son más comunes.

A continuación, se realizaron una serie de experimentos para determinar el número de capas ocultas y la cantidad de neuronas que tendría cada capa. En este paso se debe tomar en cuenta que entre más capas ocultas tenga la red neuronal mayor será su complejidad y como consecuencia se necesitarán más recursos computacionales.

Al comparar los resultados obtenidos con los casos anteriores, se determinará si hubo alguna mejora, y en caso de obtener mejores resultados, el proceso se hará de forma iterada hasta que los valores comiencen a converger a un valor. Cada una de las etapas que se siguieron proporcionó una arquitectura lo más sencilla posible con la cual se puede dar solución al problema planteado en este trabajo.

5.1. Conjunto de datos

Los datos utilizados durante este trabajo fueron recolectados por el laboratorio de investigación "Hacking and Countermeasure Research Lab (HCRL)" [13] y se encuentran disponibles para propósitos de investigación. Estos datos corresponden a sesiones de conducción de un automóvil realizados durante aproximadamente 40 minutos.

Durante estas pruebas se obtuvieron datos correspondientes a un tráfico en el CAN bus cuando no hay un ataque presente y cuando existen tres tipos de ataques: 1) DoS (acrónimo del inglés Denail of Service), 2) Fuzzy y 3) Impersonation (suplantación de identidad). La Tabla 1 muestra la cantidad de mensajes que se tiene por cada conjunto de datos.

Un DoS [15] es un ataque en el que se envía una gran cantidad de paquetes hacia una única ubicación. Esto generalmente conduce a una lentitud extrema o incluso a una parada completa del sistema. En un vehículo, un atacante podría inundar un componente importante. Por ejemplo, en el sistema de control electrónico del acelerador (ETC, del inglés Electronic Throttle Control) podría causar un mal funcionamiento y provocar un acelerador atascado o inoperativo.



Fig. 2. Estructura de un mensaje CAN.

ID_1 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_2 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_3 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_4 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_5 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_6 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_7 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_8 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8
ID_9 DF_1 DF_2 DF_3 DF_4 DF_5 DF_6 DF_7 DF_8

Fig. 3. Estructura de las tramas que se presentan a la red neuronal.

En un ataque tipo Impersonation [14], el atacante busca engañar al sistema haciéndose pasar por una entidad fiable. Cuando se da este tipo de ataque en un vehículo se busca tener acceso a un sistema específico, por ejemplo, el sistema GPS. Este tipo de ataque permite que un atacante falsifique su posición geográfica haciendo que otros crean que el vehículo está en otra posición.

En un ataque tipo Fuzzy [15], el atacante envía mensajes engañando a los sistemas de control (en cualquier orden o aleatorio) utilizando datos de forma constante. Como resultado de ello, todos los nodos (ECU) de la red reciben muchos mensajes funcionales y se puede provocar un mal funcionamiento de la red. La Figura 2 muestra la estructura de un mensaje CAN, donde se pueden distinguir algunos campos importantes como el inicio de trama (SOF, del inglés Start Of Frame), un identificador ID del mensaje, un bit de "Solicitud de transmisión remota" (RTR) y un área llamada "campo de datos" (Data Field, en inglés).

Para el proceso de entrenamiento de la red neuronal, los datos fueron divididos en dos conjuntos: conjunto de entrenamiento y conjunto de prueba. La distribución que se tomó corresponde a un 80% para el conjunto de entrenamiento y un 20% para el conjunto de prueba. Para establecer los datos que se usaron en la capa de entrada de la RN, se formó una trama que consiste en nueve mensajes consecutivos.

Sin embargo, no todos los campos de un mensaje en el CAN bus son considerados, en este caso cada mensaje está conformado por el Arbitration ID (referido simplemente como ID) y el campo de datos (Data Field). El identificador ID es un elemento importante dentro de la red, ya que, a través de este las ECU son capaces de identificar solo aquellos mensajes que les pudiera servir. La Figura 3 muestra la estructura de la trama que se presenta a la entrada de la red neuronal.

5.2. Pruebas de validación de la red neuronal

Para determinar la arquitectura de la red neuronal que mejores resultados proporcione, se realizaron una serie de experimentos cuyo fin fue determinar las

Tabla 2. Resultados obtenidos durante las pruebas para una red neuronal con una sola capa oculta.

Optimizador	No. de épocas	No. de capas	No. de neuronas (capa oculta)	Exactitud (Entrenamiento)	Exactitud (Validación)
SGD	200	1	40	0.6515	0.6507
Adam	200	1	40	0.9099	0.9019
Adagrad	200	1	40	0.7021	0.7109

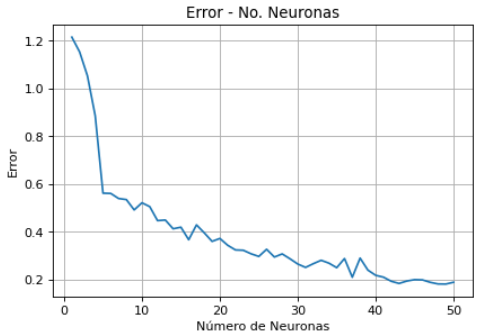


Fig. 4. Gráfica de error-número de neuronas en la capa oculta de la red neuronal propuesta.

funciones de coste de error, optimizadores, el número de capas y el número de neuronas, del modelo de red neuronal final.

De acuerdo con lo anterior, se armó una red neuronal tradicional con una capa de entrada (compuesta por 81 neuronas), una capa intermedia u oculta (con *n* neuronas), y una capa de salida (compuesta por 4 neuronas). Uno de los objetivos de estas pruebas consistió en identificar el número de capas ocultas que tendrá la red neuronal, así como el número de neuronas que tendrá cada una de ellas.

En primera instancia, se comenzó a realizar pruebas con los optimizadores y las funciones de coste de error. Los cuales se muestran a continuación.

Optimizadores:

– Adam

– Adagrad

– SGD

Funciones de Coste de error:

– Binary cross entropy

– Sparse categorical cross entropy

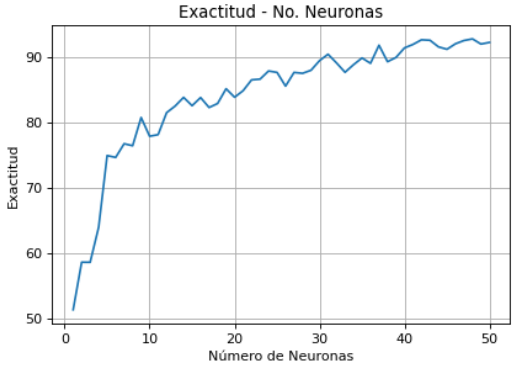


Fig. 5. Gráfica de exactitud-número de neuronas en la capa oculta de la red neuronal propuesta.

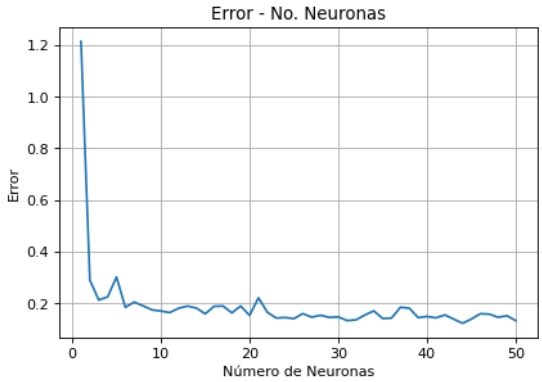


Fig. 6. Grafica de error-número de neuronas para la red neuronal con dos capas ocultas.

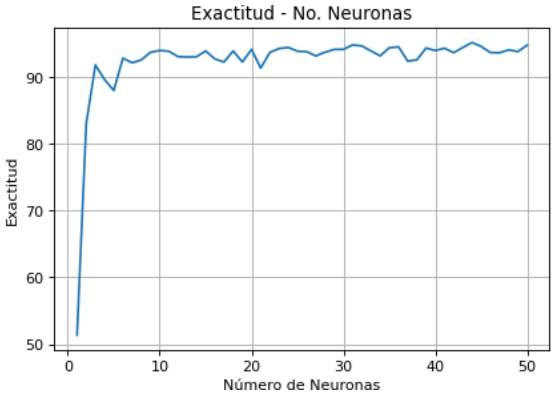


Fig. 7. Grafica de exactitud-número de neuronas para la red neuronal con dos capas ocultas.

Para la selección de la función de coste de error se comenzó con pruebas en las cuales se utilizó la función “Binary cross entropy” para realizar clasificación con solo 2 clases. Posteriormente se añadieron más clases para obtener una arquitectura multi-clase; es decir, datos que pertenecen tanto a un estado normal como a alguno de los tres ataques

Tabla 3. Resultados obtenidos durante las pruebas para una red neuronal con 2 capas ocultas.

Optimizador	**No. de épocas**	**No. de capas**	**No. de neuronas (1ª, 2ª capa)**	**Exactitud (Entrenamiento)**	**Exactitud (Validación)**
SGD	200	2	40,30	0.6036	0.6003
Adam	200	2	40,30	0.9257	0.9247
Adagrad	200	2	40,30	0.7165	0.7065

Tabla 4. Resultados obtenidos con diferente número de capas ocultas.

Capas Ocultas	No. de épocas	No. de neuronas (1ª, 2ª, …, 5ª capa)	Exactitud (Entrenamiento)	Exactitud (Validación)
1	200	40	0.9099	0.9019
2	200	40,30	0.9257	0.9247
3	200	40,30,28	0.9514	0.9379
4	200	40,30,28,26	0.9532	0.9404
5	200	40,30,28,26,24	0.9505	0.9315

considerados. Para este último, se utilizó la función de coste "Sparse categorical cross entropy". De forma simultánea, se experimentó con los optimizadores para determinar cuál de ellos se comportaba mejor.

Cada uno de los experimentos realizados arrojó resultados diferentes, en los cuales se pudo identificar las funciones que mejor se adaptaban a la arquitectura de la red neuronal, siendo en este caso el optimizador "Adam" y la función de coste "Sparse categorical cross entropy". En la Tabla 2 se muestran los resultados obtenidos durante esta prueba.

El número de neurona que tendría la capa oculta se determinó a través de experimentos en los cuales se entrenó la red neuronal desde una sola neurona hasta un total de 50. En la Figura 4 se presenta la gráfica correspondiente a la evolución del error al ir aumentando el número de neuronas en la capa intermedia, y en la Figura 5 se presenta la evolución del valor de exactitud obtenido al ir aumentando el número de neuronas en la capa intermedia.

En ambas figuras se puede observar que el error y la exactitud convergen aproximadamente después de tener 30 neuronas en la capa intermedia. En consecuencia, se optó por utilizar un total de 30 neuronas para estas primeras pruebas.

El siguiente paso fue realizar experimentos en el cual se modificó el número de capas que tendría la red neuronal. Los experimentos realizados fueron similares al que se hizo con una sola capa oculta, es decir, el número de neuronas en la segunda capa oculta se fue incrementando de una en una hasta encontrar convergencia en error y exactitud. Los resultados obtenidos durante estas pruebas se observan en las Figuras 6 y 7. En dichas figuras se puede apreciar la evolución del error y de la exactitud al ir incrementando el número de neuronas.

Tabla 5. Resultados obtenidos para una red neuronal bi-clase.

Caso	No. de épocas	No. de capas ocultas	No. de neuronas (1ª, 2ª y 3ª capa)	Exactitud (Entrenamiento)	Exactitud (Validación)
1	200	3	40,30,28	0.9999	0.9995
2	200	3	40,30,28	0.9989	0.9951
3	200	3	40,30,28	0.9623	0.9500

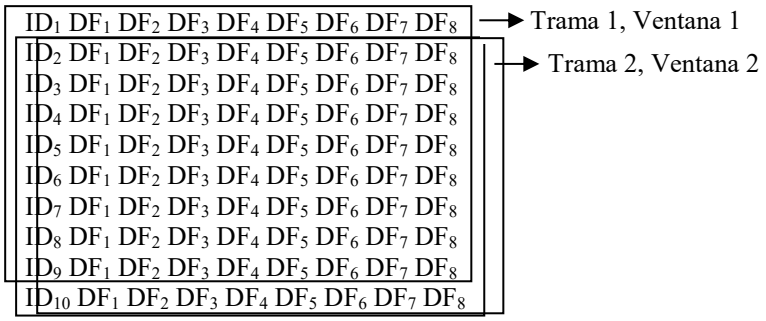


Fig. 8. Esboce de las tramas que se forman al tomar en cuenta las ventanas deslizantes.

De acuerdo con los resultados mostrados en las gráficas se puede observar que se requieren un total de 20 neuronas para la segunda capa intermedia. En la Tabla 3, se muestran los resultados obtenidos con una red neuronal que tiene dos capas ocultas. De estas pruebas, se observa que hay una mejora en el valor de exactitud obtenido en la red neuronal. Se obtiene un 92 % de exactitud comparado con el valor obtenido para una red neuronal con una sola capa oculta (90 %).

El siguiente paso fue realizar pruebas en las cuales se aumentaron el número de capas ocultas y posteriormente el número de neuronas para cada caso.

El proceso para determinar el número de neuronas que tendría la tercera, cuarta y quinta capa se realizó de forma similar a la descrita en la red con una sola capa. Los resultados obtenidos en estas pruebas se muestran en la Tabla 4.

De la tabla anterior se puede concluir que el número de capas ocultas adecuado para resolver el problema en cuestión es de 3, debido a que el valor de exactitud se mantiene o disminuye después de este número (al agregar más capas intermedias). De esta forma es posible descartar una arquitectura con un número mayor de capas ocultas con el fin de tener una arquitectura lo más ligera posible.

5.3. Pruebas con redes neuronales bi-clase

Las siguientes pruebas consistieron en entrenar la red neuronal de tal forma que solo se obtuviera un clasificador bi-clase. Es decir, una red que clasifique datos libres de ataque o datos de algún tipo de ataque. Los clasificadores bi-clase considerados son:

1. Datos libre de ataque – Datos de ataque DoS
2. Datos libre de ataque – Datos de ataque Fuzzy

Tabla 6. Resultados obtenidos para una red neuronal con ventanas deslizantes.

Caso	No. de épocas	No. de capas ocultas	No. de neuronas (1ª capa, 2ª capa, 3ª capa)	Exactitud (Entrenamiento)	Exactitud (Validación)
4 clases	200	3	30,20,18	0.9602	0.9540
3 clases	200	3	30,20,18	0.9971	0.9972
DOS – Free	200	3	30,20,18	0.9999	0.9998
Fuzzy – Free	200	3	30,20,18	0.9983	0.9976
Impersonation – Free	200	3	30,20,18	0.9731	0.9604

Tabla 7. Resultados obtenidos en otros trabajos relacionados.

Articulo	Exactitud	Precisión	Estructura
[2] DCNN	99 % (DOS,Spoofing, Fuzzy)	99 %	3 filtros (varias capas convolucionales)
[3] DNN	98 % (Inyección de mensajes)	98 %	11 capas ocultas
[4] GAN	98 % (DOS,Spoofing, Fuzzy)	98 %	2 RN (convolucional, profunda)
Propuesta	99 % (DOS,Impersonation, Fuzzy)	99 %	3 capas ocultas

3. Datos libre de ataque – Datos de ataque Impersonation

Los resultados que se obtuvieron se observan en la Tabla 5. En dicha tabla se puede observar que hay una mejora en el valor de exactitud para los casos en que hay una red neuronal bi-clase.

5.4. Pruebas con ventanas deslizantes

Finalmente, se realizaron pruebas en las cuales los datos que se usan en la capa de entrada son tratados en un formato de ventana deslizante. En este caso, cada trama tiene la característica que el mansaje más antiguo se descarta de la trama y el nuevo mensaje que llega al CAN bus se añade a la misma trama.

En este caso, a diferencia de las pruebas anteriores donde se presenta una trama de 9 mensajes en la que se tiene que esperar para tener la cantidad necesaria que formará la trama, la red neuronal que procesa el patrón de entrada en un formato de ventana deslizante tiene la capacidad de formar una trama vez que llega un nuevo mensaje.

Este formato puede ser utilizado para llevar a cabo experimentos en tiempo real ya que no hay necesidad de esperar a tener los 9 mensajes, es decir, cuando un mensaje nuevo llega al CAN bus la trama se actualizará con este nuevo mensaje y el más antiguo

se descarta. Además, se puede observar una mejora en el valor de exactitud al utilizar los datos de entrada como ventanas deslizantes. En la Figura 8 se presenta un esboce de las tramas que se forman al tomar en cuenta las ventanas deslizantes. Tomando en cuenta la estructura de los datos en un formato de ventana deslizante, se realizaron experimentos en los cuales se utilizó dicho formato. Los resultados obtenidos durante las pruebas se muestran en Tabla 6.

En la misma Tabla se puede observar que una red neuronal con ventanas deslizantes proporciona mejores valores de exactitud comparado con los experimentos anteriores. Además, las redes neuronales bi-clase tienen mejor comportamiento que cuando se tiene una red multiclase. Una red neuronal que posee datos de ataque tipo Impersonation otorga valores de exactitud más bajos. Por tal motivo, se puede decir que los datos de este tipo de ataque poseen una complejidad mayor y en consecuencia se puede proponer una arquitectura diferente para este caso.

6. Conclusiones y trabajo a futuro

Desarrollar una red neuronal puede ser una tarea difícil y los resultados pueden variar en cada proyecto. En este trabajo se ha propuesto un IDS basado en técnicas de aprendizaje automático para detectar datos anormales en el CAN bus de un automóvil, diferenciando de cuando se observa un comportamiento normal.

El MLP propuesto puede discriminar datos normales y de 3 tipos de ataques. Esta red neuronal tiene un buen desempeño para detectar datos libres de ataque, DoS, Fuzzy e Impersonation. Se puede observar que es posible obtener resultados adecuados para los casos en que se utiliza una red neuronal bi-clase y una red neuronal con ventanas deslizante.

Los mejores resultados obtenidos se presentan cuando se utiliza este tipo de clasificador y una ventana deslizante en la configuración de los patrones de entrada que se le presentan a la red neuronal, con una estructura de 3 capas ocultas.

Las exactitudes obtenidas en este caso son del 99 % para una red neuronal bi-clase para los casos de ataque DoS y Spoofing, 97 % para un ataque tipo Fuzzy y del 98 % para una red neuronal multiclase.

También podemos observar que se obtienen buenos resultados de exactitud sin la necesidad de contar con una red neuronal compleja como las redes neuronales convolucionales o las redes profundas. La Tabla 7 muestra los valores de exactitud que se obtuvieron en los trabajos relacionados, así como los resultados obtenidos en este trabajo. El trabajo que faltaría realizar consiste en realizar pruebas en tiempo real para un caso simulado e incluso a futuro realizarlo en un vehículo real.

Agradecimientos. Los autores agradecen el apoyo de CONACYT, y al IPN bajo los proyectos SIP-1999 y SIP-20210039

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: W2000: A modelling notation for complex web applications. Mendes, E., Mosley, N. (Eds): Web Engineering, Springer, pp. 335–364 (2006) doi: 10.1007/3-540-28218-1_11

2. Miller, C., Valasek, C.: A survey of remote automotive attack surfaces. black hat USA, no. 94 (2014)
3. Song, H., M., Woo, J., Kim, H. K.: In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications, vol. 21, pp. 100–198 (2020) doi: 10.1016/j.vehcom.2019.100198
4. Kang, M. J., Kang, J. W.: Intrusion detection system using deep neural network for in-vehicle network security. PloS one, vol. 11, no. 6, (2016) doi: 10.1371/journal. pone.0155781
5. Seo, E., Song, H., M., Kim, H., K.: Gids Gan based intrusion detection system for in-vehicle network. In: Proceedings of 16th Annual Conference on Privacy, Security and Trust IEEE, pp. 1–6 (2018) doi: 10.1109/PST.2018.8514157
6. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: 2016 IEEE International Conference on Data Science and Advanced Analytics, pp. 130–139 IEEE (2016) doi: 10.1109/DSAA. 2016.20
7. Ilakkiya, B., Vanitha, V.: A survey on engine control unit. vol. 2, no. 3, pp. 21–24 (2016)
8. Rizvi, S., Willet, J., Perino, D., Marasco, S., Condo, C.: A threat to vehicular cyber security and the urgency for correction. Procedia Computer Science, vol. 114, pp. 100–105 (2017) doi: 10.1016/j.procs.2017.09.021
9. Ogle, A.: Automotive controller area network systems: Driver convenience or cyber security threats? Utica College ProQuest Dissertations Publishing (2017)
10. Markovitz, M., Wool, A.: Field classification, modeling and anomaly detection in unknown CAN bus networks. Vehicular Communications, vol. 9, pp. 43–52. (2017) doi: 10.1016/ j.vehcom.2017.02.005
11. Li, H., Zhao, L., Juliato, M., Ahmed, S., Sastry, M. R., Yang, L. L.: Poster intrusion detection system for in-vehicle networks using sensor correlation and inte-gration. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 2531–2533 (2017) doi: 10.1145/3133956.3138843
12. Miller, C., Valasek, C.: A survey of remote automotive attack surfaces. Black hat USA, (2014)
13. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA (2015)
14. Hacking and countermeasure research lab. (https://ocslab.hksecurity.net/Datasets)
15. Kosmanos, D., Pappas, A., Maglaras, L., Moschoyiannis, S., Aparicio-Navarro, F. J., Argyriou, A., Janicke, H.: A novel intrusion detection system against spoofing attacks in connected electric vehicles. Array, vol. 5, (2020) doi: 10.1016/j.array.2019.100013
16. Alshammari, A., Zohdy, M. A., Debnath, D., Corser, G.: Classification approach for intrusion detection in vehicle systems. Wireless Engineering and Technology, vol. 9, no. 4, pp. 79–94 (2018) doi: 10.4236/wet.2018.94007
17. Sanchez-Vela, L. G., Molano-Clemente, M. J., Fabela-Gallegos, M. J., Martinez-Madrid, M., Hernandez-Jimenez, J. R., Vazquez-Vega, D.: Revisión documental del protocolo can como herramienta de comunicación y aplicación en vehículos. Publicación Técnica, no. 474 (2016)
18. Matich, D. J.: Redes neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional (2001)

ISSN 1870-4069

Previsión de ventas basado en un modelo híbrido de pronóstico gris y un modelo de red neuronal artificial

Kennya Jeamileth Trochez Aguilar, Rodolfo Eleazar Pérez Loaiza, Perfecto Malaquías Quintero Flores

Tecnológico Nacional de México,
Instituto Tecnológico de Apizaco,
Departamento de Estudios de Posgrado e Investigación,
México

{kwow1820, reploaiza, parafait}@gmail.com

Resumen. Los pronósticos de ventas son muy importantes para el sector minorista, ya que los pronósticos de ventas precisos ayudan a aumentar en gran medida los beneficios finales de estas. Las técnicas de planificación logística basadas en algoritmos heurísticos han sido favorables en comparativa a los métodos tradicionales de pronóstico, especialmente en los casos donde no hay datos históricos o bien, estos históricos son difíciles de interpretar. En esta propuesta se investiga la volatilidad de la demanda y otros factores desde fluctuaciones hasta cuellos de botella problemáticos para los ejecutivos de pymes. Debido a lo difícil de obtener un pronóstico preciso, se propone un método de pronóstico híbrido que fusiona una red neuronal artificial (RNA) y el modelo de predicción gris (GM (1,1)), donde el algoritmo obtiene las variables influyentes a través del análisis de correlación. Después de emplear la predicción gris y la RNA para predecir, se introduce la idea del MSE y se comparan los datos de ventas reales con el pronóstico gris junto con el resultado de la predicción de la red. Finalmente, la viabilidad y precisión de la predicción del algoritmo se verifican a través de los datos reales de las ventas. Los resultados experimentales muestran que, en la predicción de datos de ventas, el algoritmo tiene un error bajo comparado con otros métodos estadísticos y su rendimiento de predicción es mejor que el modelo ARIMA, que el modelo manual y que el modelo de series temporales.

Palabras clave: Red neuronal artificial, modelo gris, previsión de ventas, análisis de correlación, ARIMA, series temporales.

Sales Forecast Based on a Hybrid Gray Forecast Model and an Artificial Neural Network Model

Abstract. Sales forecasts are very important to the retail industry; as accurate sales forecasts help to greatly increase the bottom line of sales. Logistics planning techniques based on heuristic algorithms have been favorable compared to traditional forecasting methods, especially in cases where there are no historical data or where historical data are difficult to interpret. This proposal investigates demand volatility and other factors ranging from fluctuations to problematic

bottlenecks for SME executives. Due to the difficulty of obtaining an accurate forecast, a hybrid forecasting method is proposed that merges an artificial neural network (ANN) and the gray prediction model (GM (1,1)), where the algorithm obtains the influencing variables through correlation analysis. After using the gray prediction and ANN to predict, the idea of the MSE is introduced and the actual sales data is compared to the gray forecast along with the network prediction result. Finally, the feasibility and accuracy of the algorithm's prediction are verified through actual sales data. The experimental results show that, in the prediction of sales data, the algorithm has a low error compared to other statistical methods and its prediction performance is better than the ARIMA model, the manual model and the time series model.

Keywords: Artificial neural network, gray model, sales forecast, correlation analysis, ARIMA, time series.

1. Introducción

Hoy en día la ciencia de datos, junto con su análisis y las técnicas de aprendizaje automático y aprendizaje profundo están creciendo a nivel potencial y las empresas ahora buscan técnicas capaces de examinar, comprender e interpretar la información útil y ayudarlos a impulsar y mejorar la toma de decisiones para brindar un mejor servicio y una interpretación de la información más eficiente, mientras que el aprendizaje automático tiene sus orígenes en la informática como una de las ramas de la inteligencia artificial y uno de los componentes tecnológicos más importantes de la ciencia de datos.

Por otra parte, los pronósticos precisos son de gran importancia para el sector comercial, estos no sólo pueden proporcionar una base científica para la toma de decisiones comerciales y el trabajo de planificación de la empresa, sino que afectan directamente el rendimiento del mercado de productos de la empresa y las ganancias de ventas finales, además de que también ayudan a los comerciantes a ajustar de manera flexible sus sistemas de planificación y a mejorar la velocidad de respuesta a los cambios en la demanda, con el fin de obtener una ventaja competitiva y mayores beneficios económicos.

Los pronósticos erróneos dan lugar a productos fuera de venta o stocks innecesarios, lo que afectará directamente el crecimiento del comercio, e incluso provocará pérdidas significativas para el mismo. En el sector comercial minorista, hay muchos factores que afectan el desempeño de las ventas, como el comportamiento del cliente, los cambios estacionales y festivos, la inversión en publicidad de productos, las actividades de promoción comercial a gran escala e incluso algunas políticas económicas y comerciales. Estos factores de influencia provocan incertidumbre en la demanda de productos y presentan inconsistencias en la precisión de los pronósticos de ventas.

Actualmente, los métodos de pronóstico se dividen principalmente en dos tipos: uno se basa en la teoría matemática y estadística de métodos clásicos, como el suavizado exponencial, ARIMA, modelo de regresión o modelo de suavizado exponencial triple, el otro es un algoritmo heurístico moderno que utiliza una rama de la inteligencia artificial, como las redes neuronales artificiales (RNA). En realidad, la relación entre las ventas y los factores que influyen es bastante complicada, y los datos de ventas suelen ser asimétricos y cambian mucho.

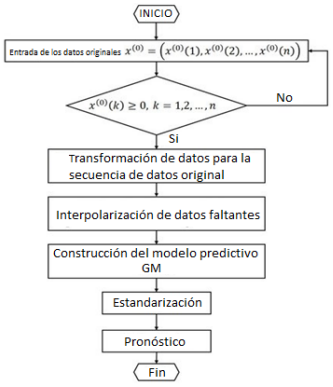


Fig. 1. Diagrama de flujo del funcionamiento del algoritmo de predicción gris (GM).

En los métodos clásicos, el suavizado exponencial, el modelo de regresión y el algoritmo ARIMA se basan en los modelos de Box y Jenkins, y estos dos métodos utilizan métodos lineales para el modelado de series de tiempo [2], pero el modelo lineal no puede capturar parte de la realidad, por lo que los problemas de ciclo asimétrico en la serie de tiempo son susceptibles a la irregularidad y variabilidad de los datos, por lo tanto es difícil obtener resultados de pronóstico precisos y no siempre es el método más apto para los pronósticos de ventas. Por el contrario, los algoritmos de redes neuronales generalmente pueden resolver eficazmente estos problemas y obtener resultados de predicción relativamente buenos [1,3].

Pero vale la pena señalar que la precisión de las predicciones de estos algoritmos depende en gran medida de sus objetivos experimentales, áreas de aplicación y si tienen una gran cantidad de datos históricos reales. En el caso de una cantidad limitada de datos históricos, es difícil obtener resultados de predicción ideales. En resumen, los algoritmos y los modelos de redes neuronales no siempre son ideales para obtener resultados satisfactorios al predecir ventas con datos limitados o bien con un excedente de datos anómalos, es por ello que en este trabajo se presenta una propuesta de una red neuronal que se adapta a casos donde el volumen de información útil es limitado.

2. Trabajos relacionados

En sus inicios, el empleo de las redes neuronales artificiales (RNA) para realizar pronósticos fue llevado a cabo por primera vez en el año 1964, cuando Hu empleó una RNA lineal adaptable de Widrow para el pronóstico del clima (Zhang, Patuwo & Hu, 1998). Pero no fue sino hasta 1986 cuando Rumelhart, Hinton & Williams (1986) introdujeron el algoritmo de retropropagación, que las RNA empezaron su verdadero desarrollo. Dos años más tarde Werbos (1988) reportó que la RNA entrenada por retropropagación "superó el desempeño de los métodos estadísticos tradicionales tales como los procedimientos de regresión y Box-jenkins en varios casos".

En los primeros años del siglo XXI, las RNAs se popularizaron en el pronóstico de series de tiempo en diversas áreas que incluyen finanzas, generación de energía, medicina, recursos del agua y ciencias ambientales, entre otras (Maier & Dandy, 2000).

Otros estudios realizados por Smith & Gupta (2000) y Zhang (2004), mostraron aplicaciones de las RNA en problemas de investigación de operaciones, negocios y pronósticos de mercado.

En cuanto a investigaciones de estadísticas, existen diversas investigaciones sobre algoritmos de pronóstico de ventas, Sun [20] introdujo la aplicación de Extreme Learning Machine (ELM) en el pronóstico de ventas, y luego propusieron el algoritmo ELM extendido (EELM). Muchos experimentos han demostrado que el algoritmo es más estable. Hsu [8] propone el Modelo Gris (GM), que puede obtener mejores resultados de predicción en el caso de datos limitados.

3. Método teórico

Para resolver el problema presentado, esta investigación propone un algoritmo híbrido, que utiliza una combinación de algoritmo de modelo gris GM (1,1) y algoritmo de red neuronal y considera de manera integral la mayoría de los factores que pudieran influir en el resultado de las ventas totales reales.

En el proceso de previsión de las ventas, el algoritmo híbrido de este articulo resuelve el problema de la precisión de predicción inestable de los modelos alternos, esto gracias a la implementación de una red neuronal artificial. En el caso de un gran volumen de datos atípicos, o como cuando la cantidad de datos es limitada, los resultados de predicción son más fiables a diferencia de otros métodos. La prueba final muestra que la precisión de predicción de este algoritmo inteligente híbrido es mejor que la de ARIMA, series de tiempo y GM generalizado.

3.1. Modelo gris GM (1,1)

El modelo de pronóstico gris es un método de pronóstico que establece un modelo matemático y hace predicciones a través de una pequeña cantidad de información incompleta. La teoría del sistema gris es una teoría que estudia y resuelve el análisis, modelado, pronóstico, toma de decisiones y control del sistema gris.

La predicción gris es la predicción realizada en el sistema gris. En la actualidad, algunos métodos de predicción de uso común (como el análisis de regresión, etc.) requieren una muestra grande. Si la muestra es pequeña, a menudo causa grandes errores e invalida el objetivo de predicción. El modelo de pronóstico gris requiere menos información de modelado, operación conveniente y alta precisión de modelado. Es una herramienta eficaz para tratar con problemas de pronóstico de muestras pequeñas.

Suponga que la secuencia de datos original, $x^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n)\}$, al definir la ecuación 1:

$$x^1(k) = \sum_{i=1}^{k} x^{(0)}(i), k = 1,2,3,\ldots,n, \quad (1)$$

Obtenemos una nueva serie $x^{(1)}(k)$: $x^{(1)}(k) = x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n)$. Para algunos procesos, $x^{(1)}(k)$ es la solución de la ecuación 2 diferencial ordinaria formada en blanco:

$$\frac{dx^{(1)}}{dt} + \alpha x^{(1)} = u, \quad (2)$$

donde α y T son el número gris, es decir, los parámetros colgantes, que se calcula por el método de mínimos cuadrados. La ecuación (1) se llama GM (1, 1). La solución de (1) es:

$$x^{(i)}(k+1) = \left[x^{(0)}(1) - \frac{u}{a}\right] e^{-ak} + \frac{u}{a}, \tag{3}$$

La ecuación (2) se llama función de respuesta en el tiempo. Para $k \geq 2$, se llama fórmula de predicción, denotada por la ecuación 4:

$$x^{(\hat{0})}(k) = x^{(\hat{1})}(k+1) - x^{(\hat{1})}(k). \tag{4}$$

De lo anterior se puede ver que la esencia del modelo de predicción gris es generar mediante la acumulación de la secuencia de datos original una nueva generación, debilitar la influencia de factores de perturbación aleatorios, encontrar la ley de desarrollo de las cosas, generar una secuencia de datos con fuerte regularidad para posteriormente predecir el desarrollo futuro de la secuencia, porque el proceso se simula usando la curva exponencial.

Se debe mencionar que debido a que el modelo gris se genera acumulando secuencias de datos para fortalecer las leyes internas de los datos, en el caso de datos limitados, los resultados de la predicción son más precisos, ya que el método no requiere un estudio de grandes volúmenes de información. Además, es un modelo estable y adaptable ante la carencia de datos estadísticos, por lo que el algoritmo híbrido que se propone en este trabajo puede obtener resultados más precisos de predicción que otros algoritmos cuando la cantidad de datos es limitada.

3.2. Red neuronal artificial

Las redes neuronales artificiales (RNA), en el campo del aprendizaje automático, son una imitación de las redes neuronales biológicas, especialmente del modelo matemático o modelo de cálculo de la estructura y función del cerebro, utilizado para estimar o aproximar la función. En la mayoría de los casos, la red neuronal artificial puede cambiar su estructura interna sobre la base de información externa, es un sistema adaptativo y, en términos sencillos, tiene una función de aprendizaje. La red neuronal moderna es una herramienta de modelado de datos estadísticos no lineales.

Las RNAs generalmente se optimizan mediante un método de aprendizaje basado en matemáticas estadísticas, por lo que también es una práctica aplicación de la estadística. Por otro lado, en el campo de la inteligencia artificial, podemos tomar decisiones sobre la obtención de información mediante la aplicación de estadísticas matemáticas, es decir, a través de métodos estadísticos, las redes neuronales artificiales pueden tener capacidad de toma de decisiones y capacidad de juicio lógico similar a los humanos. Este método brinda una ventaja sobre el procesamiento de información y toma de decisiones.

3.3. Red neuronal Backpropagation (BPP)

Las redes BP tienen un método de entrenamiento supervisado. A la red se le presenta parejas de patrones, un patrón de entrada emparejado con un patrón de salida deseada. Por cada ejecución los pesos son ajustados de forma que disminuya el error entre la

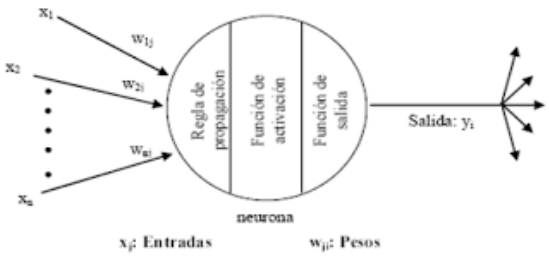


Fig. 2. Modelo del funcionamiento de una red neuronal artificial.

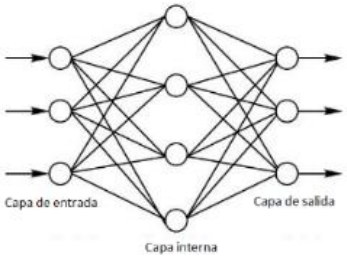


Fig. 3. Modelo arquitectónico de una red neuronal backpropagation (BP).

salida deseada y la respuesta de la red. Este algoritmo de aprendizaje conlleva una fase de propagación hacia adelante y otra fase de propagación hacia atrás. Ambas fases se realizan por cada patrón presentado en la sesión de entrenamiento.

3.3.1 Propagación hacia adelante

Esta fase de propagación hacia adelante se inicia cuando se presenta un patrón en la capa de entrada de la red. Cada unidad de la entrada se identifica con un elemento del vector patrón de entrada. Las unidades de entrada toman el valor de su correspondiente elemento del patrón de entrada y se calcula el valor de activación o nivel de salida de la primera capa.

Posteriormente, las demás capas realizarán la fase de propagación hacia adelante, el cual determina el nivel de activación de las otras capas. La unidad procesadora *j* obtiene la cantidad S_j según la siguiente ecuación:

$$S_j = \sum i \; a_i \; w_{ji}, \tag{5}$$

Y genera la salida o nivel de activación empleando la siguiente ecuación:

$$salida = f(S_j). \tag{6}$$

La función *f* es una función umbral genérica, entre las que cabe destacar la función Sigmoide o la función Hiperbólica. El valor de la salida de la unidad *j* es enviado o transmitido a lo largo de todas las conexiones de salida de dicha unidad.

Cabe mencionar que las unidades procesadoras de la capa de entrada no realizan ninguna operación de cálculo con sus entradas, ni operaciones con funciones umbrales, sólo asumen su salida como el valor del correspondiente elemento del vector de entrada.

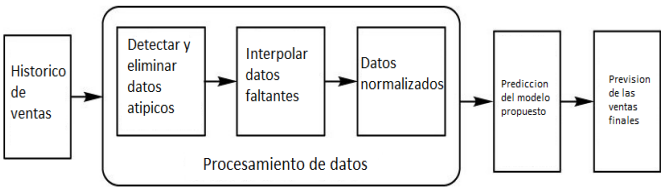


Fig. 4. Esquema del modelo híbrido propuesto.

Por otro lado, algunas redes BP utilizan unidades llamadas bias, en inglés, como parte de cualquiera de las capas ocultas y de la capa de salida.

Estas unidades presentan constantemente un nivel de activación de valor 1. Además, esta unidad está conectada a todas las unidades de la capa superior y los pesos asociados a dichas conexiones son ajustables en el proceso de entrenamiento. El uso de ésta unidad tiene un doble objetivo; mejorar las propiedades de convergencia de la red y ofrecer un nuevo efecto umbral sobre la unidad que opera.

3.3.1 Propagación hacia atrás

Una vez finalizada la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. Los cálculos de las modificaciones de todos los pesos de las conexiones empiezan por la capa de salida y continúa hacia atrás a través de todas las capas de la red hasta la capa de entrada. Dentro de los tipos de ajustes de pesos se pueden clasificar dos grupos; ajuste de unidades procesadoras de la capa de salida y ajuste de unidades procesadoras de las capas ocultas.

Ajuste de Pesos de la Capa de Salida: El ajuste de estos pesos es relativamente sencillo, debido a que existe y se conoce el valor deseado para cada una de las unidades de la capa de salida. Cada unidad de la capa de salida produce un número real como salida y se compara con el valor deseado especificado en el patrón del conjunto de entrenamiento. A partir del resultado de la comparación se calcula un valor de error δj, según la siguiente ecuación para cada unidad de la capa de salida, siendo t_j el valor de salida deseado y *f* la derivada de la función umbral *f:*

$$\delta_j = \left(t_j \quad a_j \right) f\,(S_j), \tag{7}$$

Ajuste de Pesos de las Capas Ocultas: Éstas capas no tienen un vector de salidas deseadas y por tanto no se puede seguir el método de propagación de error mencionado en el caso de las unidades procesadoras de la capa de salida. El valor de error calculado para este tipo de unidades procesadoras se obtiene a partir de la siguiente ecuación:

$$\delta_j = [\sum k\delta k \;\; wk_j] f\,(S_j), \tag{8}$$

El ajuste de los pesos asociados a las conexiones se realiza a partir del valor generalizado de cada unidad de proceso. Cada peso es ajustado según la siguiente ecuación conocida como la regla generalizada:

$$\delta w_{ji} = \eta \delta_j \, a_i. \tag{9}$$

La variable η es el coeficiente de aprendizaje. Este coeficiente, normalmente oscila entre 0.25 y 0.75, mismo que refleja el grado de aprendizaje de la red.

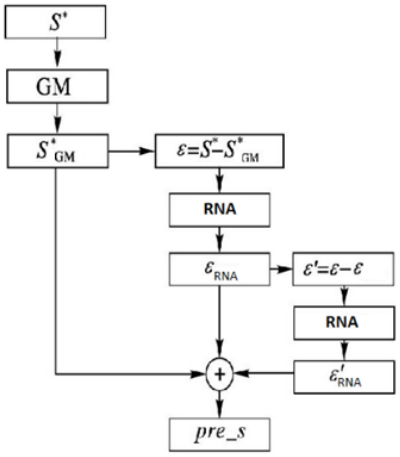


Fig. 5. Esquema del algoritmo predictivo GM.

Algoritmo 1: Algoritmo de la red neuronal Backpropagation

```
1: Algoritmo BP (red datos, η) {
2:   {wij}← Inicializar;
3:   Mientras ¬ Convergencia(red) Hacer {
4:     e← seleccionarDatos(datos);
5:       {yk} ← Forward(e);
6:       {dk} ← salidasDeseadas(e);
7:       Para cada nk ∈ CAPA (red, k) Hacer {
8:         δk = (dk-yk) f' (netk);
9:         }
10:        Para j= k -1 Hasta 1 Hacer {
11:             Para nj ∈ Capa (red, j) Hacer {
12:                 δj = f' (netj)∑j+1 δk +1wj(j+1);
13:              }
14:        }
15:        Para j = k hasta 1 Hacer {
16:            w(j-1) j = w(k-1) j + η δj y(j-1);
17:         }
18:   red ← ActualizarRed({wij});
19:   }
20:  Devolver red;
21: }
```

En el algoritmo 1 se aplica el algoritmo para cada conjunto de entrenamiento (línea 4). Se realizan las iteraciones necesarias hasta que el error baje de un umbral determinado. En la siguiente fase de propagación hacia adelante se obtiene el cálculo de la salida de la red (y_k) en la línea 5, y después se obtiene el cálculo de las salidas deseadas de la última capa (línea 6). En la siguiente etapa comienza la propagación hacia atrás.

Primero se obtiene el cálculo de los δ de la capa en función de la siguiente capa (línea 11). Finalmente se lleva a cabo la actualización de pesos de todas las capas (línea 16).

3.4. Algoritmo híbrido propuesto para la previsión de ventas en una tienda de abarrotes

Este artículo propone que el algoritmo híbrido se compone principalmente del análisis, tratamiento y predicción de datos. El tratamiento de datos incluye la detección y eliminación de valores atípicos de datos, la interpolación, reemplazo de valores perdidos y la normalización de datos. Como el esquema del modelo presentado en la figura 4.

3.5. Procesamiento de datos

El pronóstico de ventas se ve afectado por muchos factores. Los datos originales utilizados en el pronóstico tienen ruido de datos, es decir, hay muchos datos extremos o anormales, y el ruido de datos provocará una gran desviación en el pronóstico final. El pre-procesamiento de datos puede hacer que el algoritmo de aprendizaje supervisado obtenga buenos resultados de predicción.

Por lo tanto, en la investigación de este artículo, los datos serán procesados para reducir el ruido de los datos, con el fin de obtener buenos resultados de predicción. El día *i* proporcionó aquí los datos de ventas expresados como $S_i(i = 1,2, ..., n)$, donde *n* representa el número total de días.

3.6. Detectar y eliminar datos atípicos

Los datos atípicos son datos que se desvían de la mayoría de los datos del conjunto de datos. En el proceso de predicción de datos, los valores atípicos deben eliminarse y tratarse como valores perdidos. Basado en el análisis de muchos datos históricos de ventas, este artículo cree que cuando el conjunto de datos de ventas es $S= \{S_1, S_2, ..., S_n\}$, S_i satisface las condiciones:

$$abs\big(S_1 - media(S)\big) > n * estándar(S). \quad (10)$$

Es decir, encuentra que los datos de ventas S_I son datos anormales. Entre ellos: *media (S)* representa el valor promedio de la secuencia *S*; *estándar* (*S*) representa la desviación estándar de la secuencia *S*; *abs* $(\cdot)$ representa la operación de valor absoluto.

3.7. Interpolar datos faltantes

En la mayoría de las fuentes de datos originales reales, inevitablemente habrá datos faltantes, y los datos faltantes deben interpolarse para garantizar la integridad de los datos. En este artículo, el valor promedio de los cuatro elementos de datos adyacentes antes y después de los datos faltantes se utiliza para reemplazar el valor perdido.

3.8. Estandarización de datos

La estandarización de los datos puede aumentar la velocidad del cálculo de los datos y, mediante la ésta, los datos con diferentes dimensiones se pueden cambiar a la misma dimensión. En este documento, el método de estandarización de puntaje z se utiliza

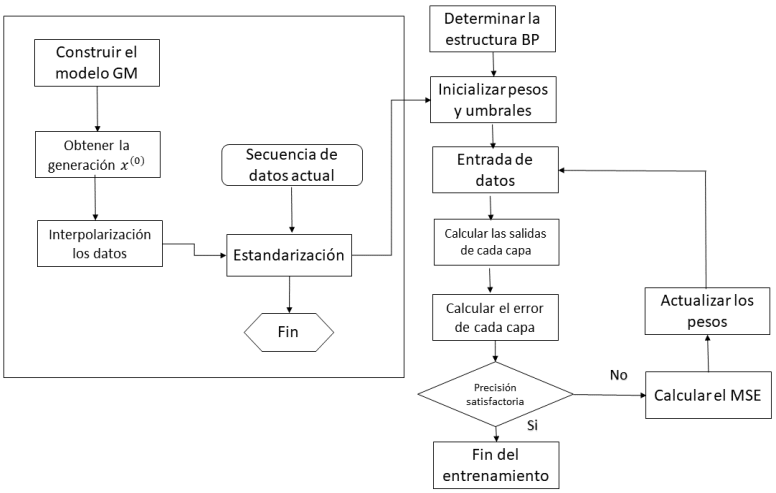


Fig. 6. Diagrama de flujo del algoritmo final. En él se muestra el funcionamiento de la fusión de la red neuronal con el modelo predictivo gris.

Tabla 1. Comparación del rendimiento de cada algoritmo en dos pruebas extras

Modelo	MSE Prueba 1	MSE Prueba 2
ARIMA	14.6 %	13.1%
Series de tiempo	19.9%	17.4%
GM estadístico (generalizado)	13.0%	10.5%
Red neuronal DGM	11.5%	9.7%

para estandarizar la entrada y la salida. El proceso de estandarización se expresa de la siguiente manera:

$$S_j = media(S) + S'_j \,.\, estandar(S). \tag{11}$$

3.9 Predicción del modelo propuesto

Son muchos los factores que inciden en las ventas. Para simplificar el modelo y predecir de forma rápida y eficaz, por lo tanto, estos factores deben ser analizados, a fin de distinguir entre los factores de influencia principales y los factores de influencia secundarios. En los métodos de análisis estadístico matemático, se pueden utilizar muchos métodos de análisis para esto, como el análisis de regresión, sin embargo, estos métodos tienen las siguientes deficiencias:

– La muestra debe obedecer a una determinada distribución de probabilidad típica, existe una relación lineal entre los datos de cada factor y el sistema, y los factores son independientes entre sí.

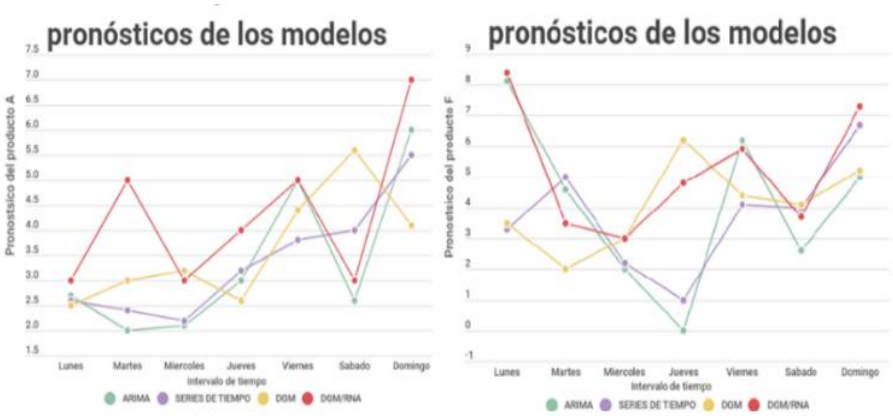


Fig. 7. Gráficas comparativas de los pronósticos obtenidos.

- El análisis requiere una gran cantidad de datos como base y la cantidad de cálculo es grande.
- Puede haber discrepancias entre los resultados cualitativos y cuantitativos.

La literatura [10] y otros modelos que combinan algoritmos lineales y no lineales utilizan principalmente algoritmos lineales para la predicción principal y algoritmos no lineales para la predicción residual. La fórmula de cálculo es la siguiente:

$$S*=S*_{lineal}+(S*-S*_{lineal})no\ lineal+\varepsilon_{1,} \tag{12}$$

$$\hat{S}^{*}=S*_{lineal}+(S*-S*_{lineal})\ no\ lineal, \tag{13}$$

dónde: S^* es la secuencia de ventas procesada; S_{lineal}^* es el resultado de la predicción del algoritmo lineal; $(S^*\text{-}S_{lineal}^*)_{no\ lineal}$ es el resultado del cálculo del algoritmo no lineal; $(S^*\text{-}(S^*\text{-}S_{lineal}^*)_{no\ lineal})$ como ε de la fórmula (14) $_1$ factores se añadió e influye en los resultados de las ventas, las ventas de predicción no lineal; S^* es el resultado de la predicción de todo el modelo predictivo y ε_1 es el residuo de predicción de todo el algoritmo.

En la figura 5, ε_{RNA} representa el resultado de usar el algoritmo RNA para predecir ε; ε ′ es el residuo de la predicción RNA de ε. El resultado final de la predicción de la prueba experimental es:

$$pre_s=S*_{GM}+\varepsilon_{RNA}+\varepsilon'_{RNA}, \tag{14}$$

dónde ε'_{RNA} es el resultado de la predicción del ε'residual que la RNA predice nuevamente. En el proceso de pronóstico real, el modelo gris se usa principalmente para predecir datos históricos de ventas: el primer paso la red neuronal se usa para analizar los residuos de ventas pronosticados por el modelo gris, luego, los factores que tienen un mayor impacto sobre las ventas se suman a la predicción. Posteriormente la RNA realiza la predicción sobre el residual entre el volumen de ventas real y el resultado de la predicción como está ilustrado de la figura 6.

3.10 Cálculo del MSE

MSE básicamente mide el error cuadrado promedio de las predicciones. Para cada punto se calcula la diferencia cuadrada entre las predicciones y el objetivo y luego promedia esos valores. Para ello, se suman todos los errores y se dividen entre el número total de puntos. *N* representa el número total de puntos dando lugar a la fórmula del Error Cuadrático Medio. Dicho cálculo se denota por la siguiente ecuación:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i), \tag{15}$$

donde y_i es el resultado real esperado y $\hat{y}_i$ es la predicción del modelo.

4. Resultados experimentales

En este artículo, el proceso de pronóstico se utilizó para predecir los datos de ventas, combinado con las características de datos de ventas limitados, gran ruido de datos y grandes fluctuaciones de datos, el algoritmo GM y el modelo de la RNA es utilizado para realizar predicciones de datos de ventas de productos.

Dentro de la hibridación de estos algoritmos, el modelo gris puede obtener resultados de predicción mejores y estables cuando la cantidad de datos es limitada, por otra parte, la red neuronal tiene un buen rendimiento en la predicción de secuencias no lineales. Los datos originales utilizados en la investigación de este artículo provienen de los datos de ventas reales de productos de una tienda de abarrotes.

Se puede ver en la Tabla 1 que el porcentaje del MSE generado por el algoritmo híbrido propuesto es de aproximadamente 11.5% cuando se utiliza una muestra pequeña del conjunto de datos para la predicción. El rendimiento del algoritmo es mejor que los otros modelos empleados por el cliente, debido a que estos dos últimos requieren un mayor volumen de datos para su análisis, mientras que el algoritmo híbrido es más estable y la precisión de la predicción es relativamente alta; se puede ver que la precisión y la estabilidad de la predicción del algoritmo propuesto mejora a comparación de los otros tres algoritmos.

De igual manera, el pronóstico fue satisfactorio al tener un MSE relativamente bajo como se muestra en la figura 7. La predicción residual secundaria del algoritmo inteligente híbrido puede ser muy eficaz, mejora la precisión y estabilidad de los resultados finales del pronóstico en gran medida y brinda altas perspectivas de su aplicación en los pronósticos de ventas reales.

5. Conclusiones y trabajo a futuro

En un pequeño comercio de abarrotes (pymes), la previsión de ventas es una parte importante de la gestión del suministro de productos y minoristas, lo que tiene un gran impacto en la mejora del beneficio final de la pyme. Aunque ya se emplean muchos modelos para hacer previsiones de ventas de múltiples productos, la cantidad limitada de datos y las grandes fluctuaciones en los datos generales son siempre los factores que limitan la aplicación de estos modelos a las previsiones de ventas totales.

Este artículo propone un algoritmo híbrido, que combina el algoritmo GM y una red neuronal. Incluye principalmente dos partes: tratamiento de datos y predicción de datos. En el proceso de predicción de datos, el algoritmo GM se utiliza para predecir la secuencia mientras que la RNA predice el residuo de la predicción del modelo gris y posteriormente agrega factores que tienen un gran impacto en las ventas.

En el proceso para la obtención del pronóstico, debido a las fluctuaciones relativamente grandes en las ventas, se deben considerar múltiples opciones que brinden un pronóstico certero, o que por lo menos, tengan un mínimo margen de error. A través del pronóstico de dos productos de la pyme y las ventas de las tiendas, se concluye que el algoritmo inteligente híbrido propuesto en esta investigación tiene un margen de error relativamente pequeño, pero cuenta con una mayor precisión, a pesar de haber empleado una muestra limitada de datos. La estabilidad de predicción del algoritmo en este documento es alta y se puede generalizar su aplicación a varios sectores.

También se debe mencionar que, debido a la pandemia, hoy en día es más costoso y complicado obtener un pronóstico muy certero, debido a la volatilidad y anomalías presentes por la reciente situación.

Referencias

1. Adya, A. M., Collopy, F.: How effective are neural networks at forecasting and prediction? A review and evaluation. Journal of Forecasting, vol. 17, no. 5-6, pp. 481–495 (1998)
2. Chase, C.: Ways to improve sales forecasts. The Journal of Business Forecasting Methods & Systems, tomo 12, no, 3, pp. 15–17 (1993)
3. Stubbings, P., Virginas, B., Owusu, G., Voudouris, C.: Modular neural networks for recursive collborative forecasting in the service chain. Knowledge-Based Systems, vol. 21, no. 6, pp. 450-457 (2008)
4. Deng, J. L.: Grey system theory, Publishing House of Huazhong University of Science and Technology, Wuhan, China (2002)
5. Li, S., Cheng, Y.: Deterministic fuzzy time series model for forescasting enrollments. Computers and Mathematics with Applications, vol. 53, no. 12, pp. 1904–19020 (2007)
6. Zhang, G.: Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, vol. 50, pp. 159–175 (2003) doi: 10.1016/S0925-2312(01)00702-0
7. Wong, W. K., Guo, Z.: a hybrid intelligent model for médium-term sales forecasting in fashion retail supply chains using extreme lerning machine and harmony search algorithm. International Journal of Production Economics, vol. 128, no. 2, pp. 614–624 (2010) doi: 10.1016/ j.ijpe.2010.07.008
8. Hsu, C. C., Chen, C. Y.: Applications of improved grey prediction model for power demand forecasting. Energy Conversion and Management, vol. 44, no. 14, pp. 2241-2249 (2003)
9. Xie, N., M., Liu, S. F.: Discrete GM 1, 1 and mechanism of grey forecasting model. Xitong Gongcheng Lilun yu Shijian, System Engineering Theory and Practice, vol. 25, no. 1, pp. 93–99 (2005)
10. Zhang, D., Zhang, X.: Study on forecasting the stock market trend based on stochastic analysis method. International Journal of Business and Management, vol. 4, no. 6, pp. 163–170 (2009)
11. Xie, N. M., Liu, S. F.: Research on discrete grey model and its predictive result. Journal Systemy Enregy, pp 520–523 (2006)

12. Xie, N. M., Liu, S. F.: Discrete grey forecasting model and its optimization. Applied Mathematical Modelling, vol. 33, no. 2, pp. 1173–1186 (2009) doi: 10.1016/j.apm.2008.01.011
13. Li, D. C., Chang, C. J., Chen, W. C., Chen, C. C.: An extended grey forecasting model for omnidirectional forecasting considering data gap difference. Applied Mathematical Modelling, vol. 35, no. 10, pp. 5051–5058 (2011) doi: 10.1016/j.apm.2011.04.006
14. Lactermacher, G., Fuller, J. D.: Backpropagation in time-series forecasting. Journal of Forecasting, vol. 14, no. 4, pp. 381–393 (1995) doi: 10.1002/for.3980140405

ISSN 1870-4069

Modelado de un sistema de inferencia difusa aplicado al monitoreo de emisiones contaminantes de tráfico de flotillas de repartición de productos

Melina Ramos Sánchez

Tecnológico Nacional de México / Instituto Tecnológico de Apizaco,
Departamento de Estudios de Posgrado e Investigación,
México

melina.rschz@gmail.com

Resumen. La contaminación en el aire fue responsable de aproximadamente 4,2 millones de muertes en todo el mundo en 2016, siendo las emisiones de los vehículos de carretera la principal fuente de contaminación atmosférica en las zonas urbanas [1]. En los últimos años, la calidad del aire a nivel mundial ha decaído de manera exponencial afectando la salud humana y el medio ambiente. En este contexto, la redacción del presente artículo tiene como objetivo modelar un sistema de inferencia difusa aplicado al monitoreo de emisiones contaminantes de tráfico de flotillas de repartición de productos, para apoyar a la toma de decisiones sobre la operación de los vehículos, ayudando así, a potenciar el ahorro de combustible y la reducción de emisiones contaminantes. El monitoreo particular de emisiones de gases contaminantes de cada vehículo permitirá la reducción de estos gases y al mismo tiempo se apoyará a garantizar el derecho a un medio ambiente sano para la sociedad reduciendo costos ambientales.

Palabras clave: Lógica difusa, emisiones contaminantes, tráfico de flotillas, monitoreo.

Modeling of a Fuzzy Inference System Applied to the Monitoring and Control of Polluting Emissions From Product Distribution Fleets

Abstract. Air pollution was responsible for approximately 4.2 million deaths worldwide in 2016, with emissions from road vehicles the main source of air pollution in urban areas [1]. In recent years, air quality worldwide has declined exponentially, affecting human health and the environment. In this context, the writing of this article aims to model a fuzzy inference system applied to the monitoring of polluting emissions from the traffic of product distribution fleets, to support decision-making on the operation of vehicles, thus helping, to promote fuel savings and the reduction of polluting emissions. The particular monitoring of pollutant gas emissions of each vehicle will allow the reduction of these gases and at the same time it will support to guarantee the right to a healthy environment for society reducing environmental costs.

Keywords: Fuzzy logic, polluting emissions, fleet traffic, monitoring.

1. Introducción

Los vehículos propulsados por motores de combustión contribuyen significativamente a la contaminación del aire relacionada con el tráfico, esto incluye una amplia variedad de productos químicos tóxicos, que incluyen carbón negro (BC), hidrocarburos aromáticos policíclicos (HAP), monóxido de carbono, óxidos de nitrógeno, óxidos de azufre y material particulado (PM). Esta contaminación se ha relacionado con múltiples resultados adversos para la salud humana que incluyen enfermedades respiratorias y cardiovasculares [2].

Según la Organización Mundial de la Salud (OMS), la contaminación del aire exterior es responsable de aproximadamente 4,2 millones de muertes en todo el mundo. La cardiopatía isquémica y los accidentes cerebrovasculares (58%), la enfermedad pulmonar obstructiva crónica y las infecciones agudas de las vías respiratorias inferiores (18%) y el cáncer de pulmón (6%) son las principales causas de muerte prematura (OMS, 2018) [3].

Por lo general las emisiones de una sola unidad, de cualquiera de estos vehículos motorizados son muy bajas comparadas con las emisiones de una chimenea industrial. Sin embargo, debido a la cantidad de vehículos en circulación, en muchas ciudades representan la fuente principal de contaminación. Siendo así la circulación de flotillas de vehículos de repartición de productos una de las principales causas de la contaminación ambiental, ya que se define como la actividad diaria más "contaminante" que se realiza comúnmente [4].

De esta manera es necesario implementar nuevas técnicas y desarrollos tecnológicos que contribuyan a mitigar los problemas antes planteados, es por ello que surgió el modelado de un sistema de inferencia difusa, para a través de datos generales obtenidos de un dispositivo OBD II (On boarding dispositive) conectado a los vehículos, conocer el grado de afección que tienen en el medio ambiente los gases contaminantes que estos emiten.

En las siguientes secciones se desarrolla paso a paso el modelado del sistema difuso para llegar al objetivo, en la sección 2, se describen los actuales trabajos relacionados al tema de control y monitoreo de vehículos con inferencia difusa, también se analizaron artículos enfocados a la aplicación de técnicas difusas como solución a los problemas de contaminación en el aire y finalmente estudios realizados sobre el exceso de emisiones contaminantes en el aire proveniente de vehículos y transporte carretero.

En la sección 3 se describe de manera detallada la metodología aplicada para lograr la propuesta de solución planteada, en la sección 4 se mencionan los resultados obtenidos con el modelado del sistema difuso y finalmente en la sección 5 se muestran las conclusiones y posibles trabajos futuros de la investigación.

2. Trabajos relacionados

En relación al tema de inferencia difusa aplicado al control y monitoreo de vehículos, existen diversas propuestas de solución. En [5] utilizando el método de inferencia difusa de Mandani que hace uso de las reglas Si X Entonces Y, si predicado entonces conclusión, se desarrolla un algoritmo de inferencia difuso para controlar la velocidad y detectar los obstáculos que se presentan en el camino y así evitar que el automóvil

choque cuando el conductor sufre cualquier alteración en su cuerpo. Así mismo, en [6] se presenta un sistema basado en un modelo lingüístico de decisión desarrollado utilizando sistemas inteligentes (lógica difusa) que obtiene una evaluación de la eficiencia en la conducción, tomando como patrón el comportamiento de los conductores en la realización de una conducción eficiente o no eficiente.

También, en [7] se presenta una aplicación de un sistema de inferencia difusa basado en una red adaptable (ANFIS) para diagnosticar previamente problemas incipientes de la red subyacente en el vehículo que posiblemente podrían causar más fallas.

En otro contexto, se analizaron algunos trabajos referentes al tema de inferencia difusa como solución a los problemas de gases contaminantes y niveles de contaminación en el aire. En [8] la investigación presenta un sistema de monitoreo y control sobre el índice de calidad del aire interior del medio ambiente basado en un controlador de lógica difusa para identificar y clasificar expresado en cuatro categorías: excelente, bueno, malo y peor.

Por otro lado, en [9] se describe un estudio utilizando datos disponibles de contaminantes clave para predecir su estado futuro mediante modelos de series de tiempo analizados con un sistema de inferencia neuro difuso adaptativo. De esta manera, en [10] con el fin de mejorar la precisión de la predicción en tiempo real de la concentración de contaminantes del aire, el artículo propone una máquina de aprendizaje extremo ponderada neuro difusa autoadaptativa basada en un método combinado de predicción de la concentración de contaminantes atmosféricos.

Para concluir esta sección, se describe el análisis de algunas propuestas de solución, enfocadas a la problemática del exceso de emisiones contaminantes en el aire proveniente de vehículos y transporte carretero. En [11] el documento presenta tasas de emisión de gases de efecto invernadero que ayudan a pronosticar y modelar las emisiones de estos gases en vehículos ligeros.

De otra manera, en [12] se obtuvieron resultados netos sobre la teoría de que las emisiones de tráfico aumentan a causa de la aceleración y desaceleración de los vehículos, esta hipótesis se comprobó con base en un modelo de emisión instantánea integrado con un modelo de simulación de tráfico microscópico. Los resultados muestran que la gestión activa de la velocidad no tiene un impacto significativo en las emisiones contaminantes.

Con este enfoque en [13] se realizó una revisión sistemática de las principales variables de tráfico utilizadas en la modelización de emisiones y calidad del aire. Los datos meteorológicos también influyen en los resultados. Las conclusiones de esta investigación mostraron que la lista de variables de tráfico clave propuesta apunta al flujo por tipo de vehículo como la variable más importante.

3. Preliminares

En los últimos años la inteligencia artificial se ha empleado en problemas cada vez más complejos, los sistemas difusos se han posicionado como una de las herramientas más útiles para modelar este tipo de problemas [15].

3.1. OBD II

La fuente de la información para este análisis proviene del dispositivo OBD II (On Board Diagnostics) el cual es un sistema de diagnóstico vehicular, incorporado al vehículo que tiene la función de controlar y monitorear al motor [14]. Conectado a una flotilla de autos tipo Van, con modelos del 2010 al 2020, con características estándar, realizando el análisis de 20 camionetas componentes de la flotilla.

3.2. Lógica difusa

El concepto de Lógica difusa fue expuesto por Lofti A. Zadeh, profesor de la Universidad de California en Berkeley, en 1965 en el artículo "Fuzzy sets", quien inconforme con los conjuntos clásicos lo presentó como una forma de procesar información permitiendo pertenencias parciales a los conjuntos [16].

Las personas son capaces de tomar decisiones basadas en información imprecisa. Los conjuntos difusos proveen una herramienta para representar esta información, y poder así también construir un sistema de ayuda para la toma de decisiones.

3.3. Sistemas difusos

Un sistema basado en reglas difusas (SBRD) está compuesto por:

- Base de conocimientos
- Base de reglas
- Base de predicados
- Base de datos
- Motor de inferencia
- Ejecuta las reglas de inferencia para llegar a una conclusión.

En este tipo de sistemas se debe tener entradas y salidas de acuerdo a los predicados y reglas de inferencia usados [6]. Un sistema de inferencia difusa define una correspondencia no lineal entre una o varias variables de entrada y una variable de salida; esto proporciona una base desde la cual pueden tomarse decisiones o definir patrones.

3.4. Método mamdani

El método de Mamdani es el más usado en aplicaciones, basado en la siguiente regla:

If x1 is A and x2 is B
then u1 is D

donde x1 y x2 son las variables de entrada: estado del convertidor catalítico, rendimiento del sensor de oxígeno, nivel de aceite y rendimiento de gasolina, A y B son funciones de pertenencia de entrada en este caso referidas al estado de las variables (excelente, constante, ineficaz, insuficiente, suficiente, excesivo), u1 es el monitoreo final de las emisiones contaminantes (nivel de contaminación), D es la función de pertenencia de la salida. En este trabajo de investigación se emplea la función de

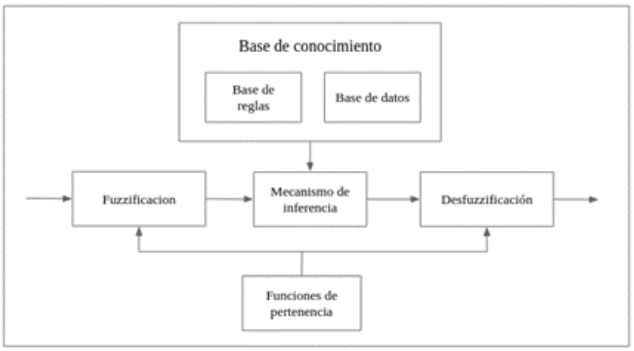


Fig. 1. Modelo del sistema de inferencia difusa [Elaboración propia].

pertenencia triangular por su facilidad computacional, y AND como operador lógico difuso.

La primera parte de la sentencia "IF x1 is A AND x2 is B" es el antecedente y la restante es el consecuente. Un ejemplo es IF estado del convertidor catalítico is ineficaz and rendimiento del sensor de oxigenos is pobre THEN nivel de contaminación is excesivo.

El método de Mamdani utiliza un conjunto de reglas difusas "SI-ENTONCES" (IF-THEN). Toma como entrada los valores de la fuzzificación y se aplican a los antecedentes de las reglas difusas. Si una regla tiene múltiples antecedentes, se utiliza el operador AND u OR para obtener un único número que represente el resultado de la evaluación. Este número se aplica al consecuente [15].

En un sistema difuso tipo Mamdani se distinguen las siguientes partes:

Fuzzificador

La entrada del sistema de lógica difusa tipo Mamdani, es un valor numérico proveniente de un dispositivo OBD II, como se mencionó antes, para que este valor pueda ser procesado por el sistema difuso se necesita convertir en un "lenguaje" que el mecanismo de inferencia pueda procesar.

Esta es la función del fuzzificador, que toma los valores numéricos provenientes del exterior y los convierte en valores difusos que pueden ser procesados por el mecanismo de inferencia [16].

Base de reglas difusas

La base de reglas es la manera que tiene el sistema difuso de guardar el conocimiento lingüístico que le permite resolver el problema para el cual ha sido diseñado [17]. Estas reglas son del tipo IF-THEN. En la figura 1 se muestra el modelo del sistema de inferencia difusa con el que se trabaja esta investigación.

Mecanismo de inferencia

Una vez teniendo los resultados de la función de pertenencia arrojados por el fuzzificador, los mismos deben ser procesados para generar una salida difusa. La tarea

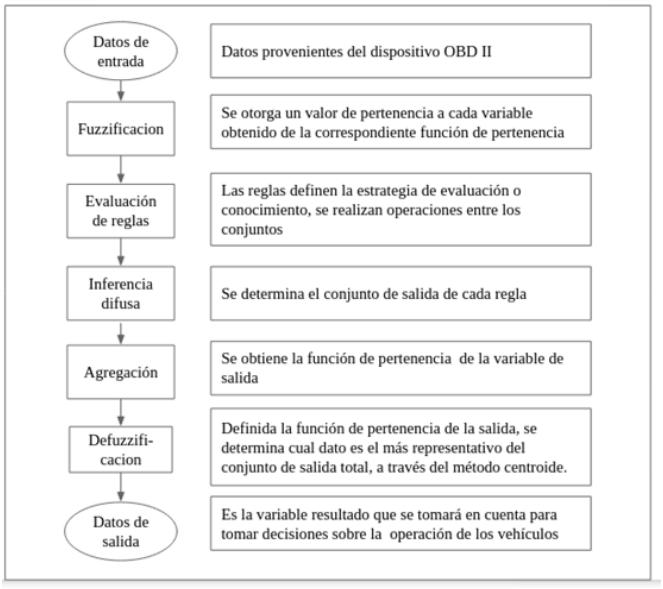


Fig. 2. Diagrama de flujo del proceso de inferencia [Elaboración propia].

del sistema de inferencia es tomar los niveles de pertenencia y apoyado en la base de reglas generar la salida del sistema difuso [18].

4. Metodología

La implementación del modelo, se ha elaborado mediante un motor de inferencia difuso implementado a tal efecto en el lenguaje de programación Python, utilizando la biblioteca Inferfuzzy dedicada a implementar Sistemas de Inferencia Difusa, en conjunto con desarrollos en el software GNU Octave, un lenguaje de programación matemático de licencia libre.

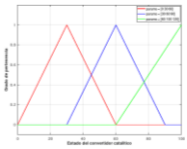
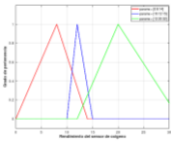
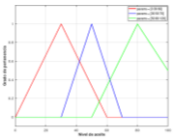
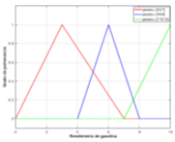
Para la realización de esta investigación se consultaron diversas fuentes, inicialmente para seleccionar las variables de importancia en la emisión de gases contaminantes después, la base de datos fue obtenida de la bibliografía del libro "CO2 Catcher: A Platform for Monitoring of Vehicular Pollution in Smart Cities" escrito por J. Oliveira en el 2017, de la editorial IEEE First Summer School on Smart Cities (S3C), los cuales son datos de prueba provenientes de un dispositivo OBD II.

A continuación, en la figura 2 se muestra el algoritmo en forma de diagrama de flujo que describe cada paso realizado para la implementación del modelo.

4.1. Variables lingüísticas de entrada

En la tabla 1 se describen las variables lingüísticas de entrada para el sistema de inferencia difusa, el valor asignado y la función de pertenencia correspondiente a cada una.

Tabla 1. Descripción de variables lingüísticas de entrada.

Variable	Valor	Función de pertenencia
Estado del convertidor catalítico (edoCC)	{excelente, constante, ineficaz}	{□excelente, □constante,  □ineficaz}
Rendimiento del sensor de oxigeno (renSO)	{pobre, ideal, rico}	{□pobre, □ideal, □rico} 
Nivel de aceite (nivelAceite)	{insuficiente, suficiente, excesivo}	{□insuficiente, □suficiente,  □excesivo}
Rendimiento de gasolina (renGas)	{bajo, deseable, alto}	{□bajo, □deseable, □alto} 

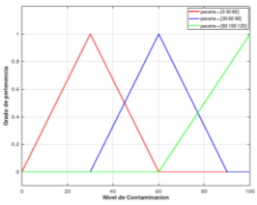
Con ayuda del dispositivo OBD II el cual mediante un escáner detecta los códigos que marcan fallas en los distintos sensores que tiene el vehiculo podremos conocer el estado del convertidor catalítico, el rendimiento del sensor de oxígeno, el nivel de aceite con el que cuenta el vehículo y el rendimiento de gasolina que tiene, ya que, si alguno de estos elementos se encuentra en mal estado o en falla, las emisiones del vehículo se alteran [19]. A continuación, se describe la importancia de las variables antes descritas.

Estado del convertidor catalítico

Actualmente los vehículos de combustión interna vienen equipados con un elemento llamado convertidor catalítico, el cual es uno de los componentes más importantes para el control de las emisiones de gases contaminantes resultantes de la combustión producida por los motores.

Su principal función es transformar los gases contaminantes resultantes de la quema de combustible en otros gases más amigables con el medio ambiente [20].

Tabla 2. Descripción de la variable lingüística de salida.

Variable	Valor	Función de pertenencia
Nivel de contaminación (nivelContaminacion)	{bajo, medio, excesivo}	{□bajo, □medio, □excesivo} 

El convertidor catalítico utiliza elementos químicos como el platino y rodio para disminuir las emisiones de NOx, el estado del catalizador del vehículo, se evalúa en una escala de 0 a 100, como excelente, constante o ineficaz.

Rendimiento del sensor de oxigeno

Todos los autos producidos después de 1980, cuentan con un sensor de oxígeno. Éste forma parte del sistema de control de emisiones. La meta del sensor es ayudar al motor a funcionar lo más eficiente posible, al igual que producir la menor cantidad de emisiones permisibles.

Un motor quema combustible en la presencia de oxígeno si la proporción tanto de aire como de gasolina es equitativa la mezcla es ideal, para la quema total, se necesitan de 14.7 partículas de aire por 1 de combustible.

La computadora del motor vigila esta mezcla para determinar si es ideal, rica o pobre (entrada de mayor o menor cantidad de partículas de oxígeno por partícula de combustible) [21].

Cuando el sensor de oxígeno falla, la computadora del vehículo ya no puede detectar el rango de aire/combustible, por lo que termina adivinando cuál es. Con esto el auto se desempeña pobremente, las emisiones de gases contaminantes son excesivas y el uso de combustible es más de lo que requiere.

Los estados de evaluación de esta variable corresponden a pobre, ideal o rico, tomando como se mencionó antes, una entrada ideal de 14.7:1 partículas, en otro caso si se cuenta con una entrada de aire menor en la combinación, entonces se tiene un excedente de combustible esto sería una mezcla rica. Finalmente, si se tiene más aire y menos combustible, entonces se tiene un exceso de oxígeno, lo que es una mezcla pobre.

Nivel de aceite

La principal función del aceite en los motores es fungir como lubricante. Generalmente se espera que disminuyan la fricción, el desgaste y el calor en los componentes de la maquinaria.

Uno de los aspectos más importantes de un lubricante es su capacidad para influir en las emisiones que contaminan el ambiente [22].

Tabla 3. Muestra de la base de reglas difusas.

Num.	Antecedente	Consecuente
R1	IF edoCC is excelente AND renSO is pobre	THEN nivelContaminacion is medio
R2	IF edoCC is excelente AND renSO is normal	THEN nivelContaminacion is bajo
R3	IF edoCC is ineficaz AND renSO is rico	THEN nivelContaminacion is excesivo
R4	IF edoCC is constante AND nivelAceite is insuficiente	THEN nivelContaminacion is medio
R5	IF edoCC is constante AND nivelAceite is suficiente	THEN nivelContaminacion is bajo
R6	IF edoCC is constante AND nivelAceite is excesivo	THEN nivelContaminacion is medio
R7	IF edoCC is ineficaz AND renGas is bajo	THEN nivelContaminacion is excesivo
R8	IF edoCC is ineficaz AND renGas is deseable	THEN nivelContaminacion is medio
R9	IF edoCC is ineficaz AND renGas is alto	THEN nivelContaminacion is medio
R7	IF edoCC is excelente AND nivelAceite is insuficiente	THEN nivelContaminacion is medio
R8	IF edoCC is excelente AND nivelAceite is suficiente	THEN nivelContaminacion is bajo
R9	IF edoCC is excelente AND nivelAceite is excesivo	THEN nivelContaminacion is medio
R10	IF edoCC is excelente AND renSO is rico	THEN nivelContaminacion is medio

Cuando el nivel de aceite es el adecuado para el motor, puede ayudar a reducir algunos de los contaminantes dañinos que se arrojan al medio ambiente y al mismo tiempo cubrir con las necesidades que la máquina requiere. Insuficiente, suficiente y excesivo son las etiquetas lingüísticas para esta variable en una escala de 0 a 100, siendo el 100 un nivel de aceite excesivo.

Rendimiento de gasolina

En todos los casos las emisiones de CO2 son directamente proporcionales al consumo de combustible [23]. Sabiendo esto y tomando en cuenta que los vehículos tipo Van, componentes de una flotilla dedicada a la repartición de productos, son la muestra para esta investigación y consumen gasolina y no otro tipo de combustible, el rendimiento deseable es de 9.4 km por litro, evaluando un rendimiento menor como bajo y un rendimiento mayor como alto.

4.2. Variable lingüística de salida

La variable lingüística de salida es quien nos dará el resultado que estamos esperando conocer para poder tomar decisiones exitosas referentes a la disminución de emisiones contaminantes del automóvil, para esto finalmente se requiere el nivel de

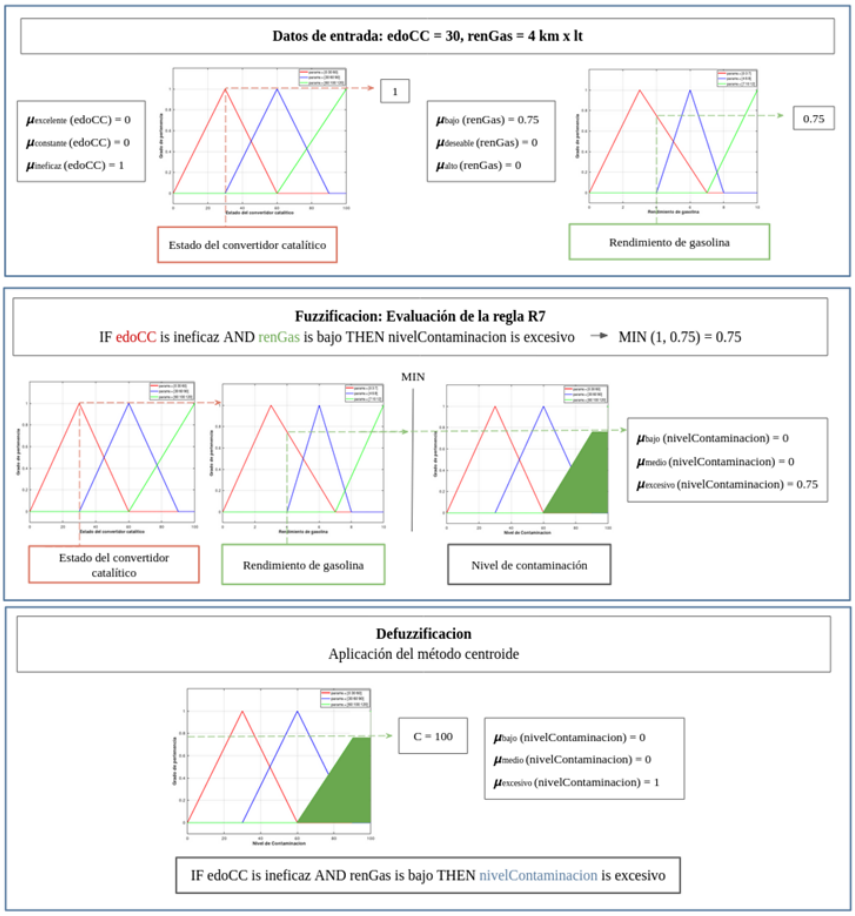


Fig. 3. Proceso de evaluación para la regla R7 [Elaboración propia].

contaminación que se está presentando evaluado con los valores lingüísticos, bajo, medio y excesivo, como se describe en la tabla 2.

4.3. Base de reglas difusas

En la tabla 3, se describen parte de las reglas difusas generadas para alimentar el sistema de inferencia difusa, este proceso se realizó con todas las combinaciones posibles entre variables, obteniendo inicialmente 84 reglas difusas distintas.

4.4. Inferencia difusa

En la figura 3 se muestra de manera gráfica el proceso para generar la salida del sistema difuso basado en la regla 7 (R7) descrita en la tabla 3. Tomando como indicios los siguientes valores para las variables involucradas: edoCC = 30 y renGas = 4km x lt.

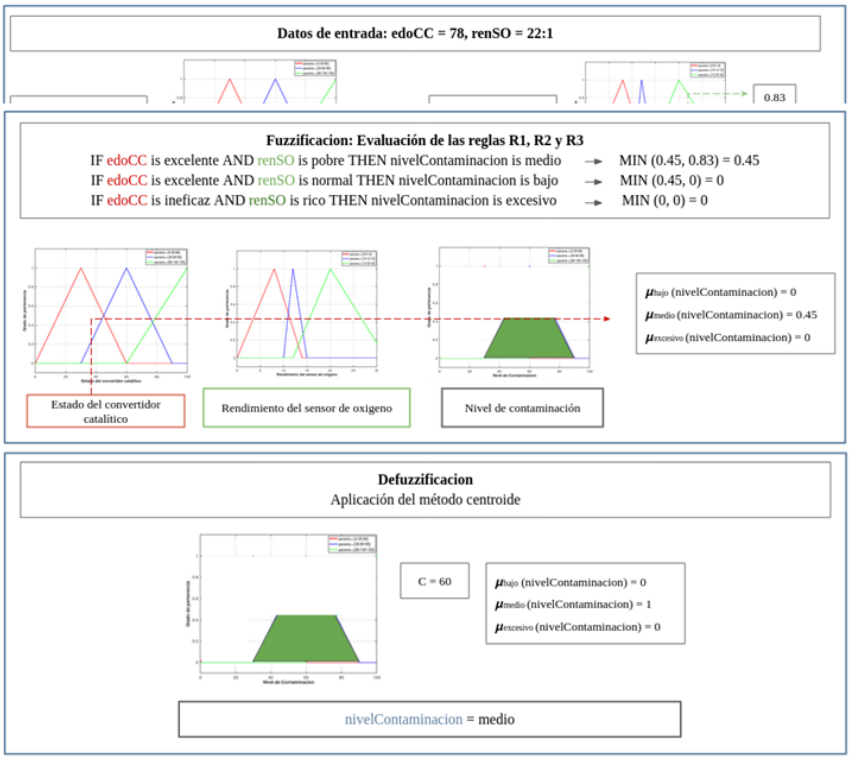


Fig. 4. Proceso de inferencia difusa con las reglas R1, R2 y R3 [Elaboración propia]

Con los predicados correctos la base de conocimiento infiere de manera exitosa el nivel de contaminación que un vehículo está arrojando, dando la pauta para la toma de decisiones beneficiosas respecto a la operación de los vehículos.

5. Experimentos y resultados

Para analizar detalladamente el comportamiento de las variables se realizó el proceso de inferencia difusa, con diferentes funciones de pertenencia además de la triangular, que fue el punto de partida de la investigación, tales como la función de pertenencia trapezoidal y gaussiana, todos los procesos tomando como base las reglas R1, R2 y R3 de la tabla 3 mostrada antes, obteniendo variaciones muy pequeñas, a continuación, se muestran los resultados gráficamente en la figura 4. utilizando función de pertenencia triangular.

Finalmente, los resultados obtenidos muestran que podemos conocer a partir de la aplicación de un sistema de inferencia difuso, información relevante para combatir los niveles de contaminación en el aire actuales, esto a través de las emisiones contaminantes del tráfico de vehículos, conociendo después de una evaluación el nivel de contaminación con el que trabajan.

6. Conclusiones y trabajo a futuro

Tras la finalización del trabajo planteado, se puede destacar que los objetivos establecidos inicialmente han sido cumplidos satisfactoriamente, ya que se ha conseguido crear un prototipo que evalúa con el modelado de un sistema de inferencia difusa el nivel de emisiones contaminantes que arroja un automóvil dedicado a la repartición de productos.

Todo ello ha sido posible, mediante la implementación de un modelo matemático basado en una técnica de inteligencia artificial, lógica difusa, que permite proporcionar información relevante para tomar decisiones desde la perspectiva de operación de los vehículos. Dicho modelo, hace uso de determinadas reglas que evitan la contradicción y repetición de las mismas.

Por otro lado, desde el punto de vista del trabajo futuro, uno de los objetivos más importantes que se pretende alcanzar a partir del modelo creado, consiste en refinar los resultados obtenidos mediante la aplicación de técnicas de inteligencia artificial y realizar la implementación con datos reales provenientes de una flotilla de vehículos trabajando en tiempo real, que permitan lograr un aumento en la eficacia del modelo y sus resultados, estudiando las posibles reglas sobrantes y generando un mayor rendimiento en la aplicación.

Referencias

1. Harrison, R. M., Van, Vu T., Jafar, H. Shi, Z.: More mileage in reducing urban air pollution from road traffic. ScienceDirect Environment International, vol. 149 (2021) doi: 10.1016/j.envint.2020.106329
2. Hilpert, M., Johnson, M., Kioumourtzoglou, M., Domingo-Relloso, A., Peters, A., Mora, B., Hernández, D., Ross, J., Chillrud, S.: A new approach for inferring traffic-related air pollution: Use of radar-calibrated crowd-sourced traffic data. ScienceDirect Environment International, vol. 127, pp. 142–159 (2019) doi: 10.1016/j.envint.2019.03.026
3. Pinto, J. A., Kumar, P., Alonso, M. F., Lemker, W., Pedruzzi, R., Soares, F., Moreira, D. M., Toledo. T.: Atmospheric pollution research. Brazil, Elsevier B.V (2019)
4. Khreis. H.: Traffic, air pollution, and health. Barcelona: ISGlobal (2020)
5. Mejía, F. Arellano, A. Vallejo, G. Villa. E.: Control de velocidad de un automóvil empleando una técnica de inteligencia artificial. Revista Infociencia, vol. 10 (2016) doi: 10.24133/infociencia.v10i1.974
6. Villeta, M., Lahera, T., Merino, S., Zato, J. G., Naranjo, J. E., Jiménez, F.: Modelo para la conducción eficiente y sostenible basado en lógica borrosa. Revista Iberoamericana de Automática e Informática industrial, vol. 9, no. 3 (2012) doi: 10.1016/j.riai.2012.05.009
7. Suwatthikul, J., McMurran, R., Jones R. P.: In-vehicle network level fault diagnostics using fuzzy inference systems. Applied Soft Computing, vol. 11, no. 4, pp. 3709–3719 (2011) doi: 10.1016/j.asoc.2011.02.001
8. Dionova, B. W., Mohammed, M. N., Al-Zubaidi, S., Yusuf, E.: Environment indoor air quality assessment using fuzzy inference system. ICT Express, vol. 6, no. 3, pp. 185–194 (2019) doi: 10.1016/j.icte.2020.05.007
9. Zeinalnezhad, M., Chofreh, A. G., Goni, F. A., Kleme, J. J.: Air pollution prediction using semi-experimental regression model and adaptive neuro-fuzzy inference system. Journal of Cleaner Production, vol. 261 (2020) doi: 10.1016/j.jclepro.2020.121218

10. Li, Y., Jiang, P., She, Q., Lin, G.: Research on air pollutant concentration prediction method based on self-adaptive neuro-fuzzy weighted extreme learning machine. Environmental Pollutionvol, vol. 241, pp. 1115–1127 (2018) doi: 10.1016/j.envpol.2018.05.072
11. Iankov, I., Taylor, M. A. P., Scrafton D.: Forecasting greenhouse gas emissions performance of the future Australian light vehicle traffic fleet. Transportation Research Part A: Policy and Practice, vol. 99, pp. 125–146 (2017) doi: 10.1016/j.tra.2017.03.011
12. Panis, L. I., Broekx, S., Liu R.: Modelling instantaneous traffic emission and the influence of traffic speed limits. Science of The Total Environment, vol. 371, vol. 1-2, pp. 270–285 (2006) doi: 10.1016/j.scitotenv.2006.08.017
13. Pinto, J. A., Kumar., P., Alonso., M., F. Andreão., W. L., Pedruzzi, R., Soares dos Santos., F., Martins-Moreira, D., Toledo de Almeida Albuquerque, T.: Traffic data in air quality modeling: A review of key variables, improvements in results, open problems and challenges in current research. Atmospheric Pollution Research, vol. 11, no. 3, pp. 454-468 (2020) doi: 10.1016/j.apr.2019.11.018
14. Oliveira J., Lemos, J., Vieira, E., Silva, I., Abrantes, I., Barros, D., Costa, D. G.: CO2 catcher a platform for monitoring of vehicular pollution in smart cities. IEEE First Summer School on Smart Cities (S3C), Natal, pp. 37–42 (2017) doi: 10.1109/S3C.2017.8501380
15. Ross, T. J.: Fuzzy logic engineering applications. Tercera Edición, University of New Mexico, USA: Wiley (2010)
16. Zadeh, L. A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, vol. 1, no. 1, pp. 3–28 (1978) doi: 10.1016/0165-0114(78)90029-5
17. Kaufmann, A., Gil-Aluja, J.: Introducción de la teoría de los subconjuntos borrosos a la gestión de las empresas, Editorial Milladoiro, España (1993) doi: 10.23857/fipca ec.v6i1.348
18. Zimmermann, H. J.: Fuzzy sets and its applications. Publications Kluwer Nijhoff, Dordrecht, Southern Holland (1985)
19. CONUEE: Comision nacional para el uso eficiente de la energía (2005)
20. Auto avance. Funcionamiento del Convertidor Catalítico (2013) https://www.autoavance .co/blog-tecnico-automotriz/174-funcionamiento-del-convertidor-catalitico/
21. Blancarte, J.: Auto cosmos ¿Cómo funciona el sensor de oxígeno en un auto? (2013) http://especiales.ve.autocosmos.com/tipsyconsejos/noticias/2013/02/09/como-funciona-el-sensor-de-oxigeno-en-un-auto
22. Esquivel, P.: Machinery Lubrication. vol. 4 (2017)
23. Rocenro, A.: Emisiones de CO2 (2009) https://www.auto10.com/reportajes/emisiones-de-co2-que-contamina-mas-un-gasolina-o-un-diesel/588

ISSN 1870-4069

Modelo operador lineal en toma de decisiones: sensibilidad a ganancias y pérdidas

Laurent Avila-Chauvet, Diana Mejía Cruz

Instituto Tecnológico de Sonora,
Departamento de Psicología
México

diana.mejia@itson.edu.mx

Resumen. El Iowa Glambling Task (IGT) ha sido utilizado para evaluar la toma de decisiones en participantes con daño ventromedial en la corteza prefrontal y el desarrollo de los procesos de toma de decisiones. El estudio propone un modelo de agente que integra el Operador Lineal como regla de aprendizaje y parámetros de sensibilidad a las pérdidas y ganancias para simular la toma de decisiones el IGT. Asimismo, busca comparar la ejecución de los jóvenes adultos que participaron en el estudio, con la ejecución de los agentes simulados a partir de la manipulación del factor de memoria, la sensibilidad a las pérdidas y sensibilidad a las ganancias. Los resultados muestran que conforme aumenta la sensibilidad a las ganancias y decrece la sensibilidad a las pérdidas, aumenta la preferencia de los agentes por las alternativas subóptimas. Asimismo, cuando la sensibilidad a las pérdidas es mayor, conforme aumenta el factor de memoria, aumenta la preferencia por las alternativas óptimas. A pesar de que los resultados de los agentes y los participantes son cualitativamente similares los participantes mostraron mayor aversión al riesgo que los agentes. Concluimos la importancia de evaluar los parámetros del modelo operador lineal en diferentes muestras clínicas y comunitarias.

Palabras clave: Modelo operador lineal, toma de decisiones, agentes, sensibilidad a las pérdidas, sensibilidad a las ganancias.

Lineal Operator Model in Decision Making: Sensitivity to Gains and Losses

Abstract. The task Iowa Gambling Task (IGT) has been used to evaluate decision-making in individuals with ventromedial impairment in the prefrontal cortex and the development of decision-making processes. This study proposed an agent model that integrates the Lineal Operator as a learning rule, gains, and loss sensitivity parameters to simulate IGT decision making. Likewise, it was compared the performance of young adults who participated in the study, with the performance of the simulated agents based on the manipulation of the memory factor, losses, and gains sensitivity. The results show that as gains sensitivity increases and losses sensitivity decreases, agents' preference for suboptimal alternatives increases. Likewise, when the losses sensitivity is greater, as the memory factor increases, the preference for optimal alternatives increases. Despite the fact that the results of the agents and the participants are

qualitatively similar, the participants showed greater aversion to risk than the agents. We conclude the importance of evaluating the parameters of the lineal operator model in different clinical and community samples.

Keywords: Lineal operator model, decision making, agents, sensitivity to losses, sensitivity to gain.

1. Introducción

La elección racional puede tener distintos usos o significados en función de la disciplina y/o área de estudio. Kacelnick [7] señala tres tipos de racionalidad: 1) *Filosófica/Psicológica*, tomar decisiones en función de ideas y/o creencias racionales; 2) *Económica*, maximizar la cantidad de recursos, elegir aquellas alternativas que generan mayores ganancias a un menor costo; y 3) *Biológica,* elección de aquellas estrategias que permiten la supervivencia de la especie y/o aumentar la eficacia biológica. En situaciones de riesgo donde la probabilidad de ganancia y perdida de las alternativas es conocida, se ha observado que algunos individuos tienden a desviarse de la elección económicamente racional y elegir las alternativas desventajosas o subóptimas, es decir, aquella que generan menores ganancias a largo plazo.

El Iowa Glambling Task (IGT) es una tarea comportamental que permite evaluar la toma de decisiones en *situaciones de incertidumbre*, en las cuales los participantes exploran las propiedades de las alternativas; y posteriormente *situaciones de riesgo*, en las cuales los participantes conocen las propiedades de las alternativas.

Esta tarea ha sido utilizada para evaluar la toma de decisiones en participantes con daño ventromedial en la corteza prefrontal [1].

Así mismo, ha permitido entender el proceso de toma de decisiones y evaluar el desarrollo en los procesos de decisión durante la adolescencia y el declive en la toma de decisiones después de los 60 años [4]. Algunos de los modelos cognitivos y computacionales que se han utilizado para entender y simular la toma de decisiones en el IGT son: La *Valencia de Expectativa* (EV) y el *Aprendizaje de Valencia Prospectiva* (PVL). Los parámetros de estos modelos capturan distintos procesos psicológicos como la motivación, memoria y consistencia de la respuesta [5, 6, 10].

Una versión hibrida de estos modelos (PVL-Delta) que utiliza el *Operador Lineal* como regla de aprendizaje, ha demostrado simular de forma más efectiva la toma de decisiones en el IGT, que sus versiones aisladas. Este último integra parámetros relacionados con la partición del espacio y la influencia selectiva [9].

A pesar de que estos modelos describen y/o simulan el proceso de toma de decisiones en el IGT, modelos bayesianos que contemplan la sensibilidad a las ganancias, la sensibilidad a las pérdidas y la memoria de pagos, han demostrado ser más eficaces para entender el proceso de toma de decisiones y sus implicaciones clínicas [11].

El presente estudio propone un modelo de agente más simple, que integra el *Operador Lineal* como regla de aprendizaje, parámetros de sensibilidad a las pérdidas y sensibilidad a las ganancias y la regla de decisión *Igualación* para simular la toma de decisiones el IGT. Asimismo, el estudio propone comparar la ejecución de agentes simulados con la ejecución de jóvenes adultos a partir de la manipulación del factor de memoria y la sensibilidad a las pérdidas y ganancias.

2. Método

2.1. Participantes y diseño

Se utilizó un diseño transversal y una estrategia de muestreo en bola de nieve. A cada participante se le asignó un numero de ID para garantizar su anonimato. Antes de comenzar, los participantes aceptaron un consentimiento informado en el cual se les señalaba que su participación era voluntaria y que los datos recolectados únicamente serían utilizados para fines de investigación.

Los datos se recolectaron en un servidor privado al cual los investigadores principales tenían acceso. Un total de 59 participantes completaron este estudio (70.3% mujeres y 29.7% hombres). El promedio de edad de los hombres fue de 24.27 años (DS = 10.31), y mujeres de 24.27 años (DS = 10.31). Los años de nivel educativo para hombre fue de 17.73 años (DS = 11.10), y mujeres de 15.23 años (DS = 3.12).

2.2. Instrumentos

IOWA Gambling Task: Los participantes fueron evaluados utilizando una versión computarizada en línea del IGT. Se le indico a los participantes que en cada ensayo debían elegir una carta de uno de los cuatro mazos marcados como A, B, C y D; y que debían tratar de ganar la mayor cantidad de dinero posible y evitar perder tanto como fuera posible. Así mismo se les indico que:

> "En la parte superior del mazo elegido aparecerá con verde la cantidad de dinero ganada y con rojo la cantidad de dinero pérdida (i. e. $100 – $50 = $50). En la parte superior izquierda se mostrará el dinero acumulado."

Al finalizar cada ensayo, los participantes acumulaban ganancias y/o pérdidas en función de las propiedades de cada una de las alternativas (ver, Tabla 1). Los mazos A y B (alternativas *subóptimas*) generaban pérdidas a largo plazo, mientras que los mazos C y D (alternativas *óptimas*) generaban ganancias a largo plazo. La tarea finalizaba cuando elegían un total de 100 cartas.

2.3. Procedimiento

Todos los participantes fueron reclutados a través de una plataforma en línea. El estudio fue llevado a cabo durante 2 meses (31 de julio 2020 a 29 de septiembre 2020). Los participantes aceptaron voluntariamente ser parte del estudio a través del consentimiento informado. Los procedimientos cumplieron con la declaración de Helsinki sobre la participación en investigación con humanos: 1) Confidencialidad y privacidad de los datos; 2) Consentimiento informado; 3) Publicación y registros de investigación; 4) Diseminación de resultados.

Los participantes podían solicitar mayor información a través de los correos electrónicos del laboratorio.

2.4. Modelo operador lineal

Los Modelos Basados en Agentes (MBA) son modelos computacionales que abstraen y/o simulan algunos de los elementos y relaciones que componen un sistema real. Generalmente están representados por matrices que contienen: 1) Agentes (objetos que realizan una acción) con distintas reglas de comportamiento, aprendizaje y elección; y 2) Objetos con distintas propiedades con las cuales pueden interactuar los agentes.

El modelo propuesto pretende simular y/o predecir la preferencia por de cada una de las alternativas en el IGT. Así mismo asume que:

a) El ambiente está compuesto únicamente por 4 mazos de cartas que difieren en la magnitud de ganancias, la magnitud de pérdidas y probabilidad de pérdidas. La tarea termina cuando el agente elige 100 cartas.
b) Los agentes pueden en cada ensayo elegir uno de los cuatro mazos. En cada ensayo se calculan las ganancias netas (ganancias - pérdidas) ponderadas por la sensibilidad del agente a las ganancias y/o las pérdidas.
c) Al inicio de cada simulación el valor que le atribuye el agente a cada mazo es el mismo. A medida que el agente adquiere experiencia con cada uno de los mazos, su valor aumenta o decrementa en función del valor del mazo en el presente y el valor del mazo en el pasado.
d) La probabilidad de que un agente elija un mazo depende del valor de cada uno de los mazos. Aquellos mazos con mayor valor, tienen mayor probabilidad de ser elegidos, en contraste con los mazos de menor valor (*Igualación*).

Se considera que estos supuestos, por un lado, abstraen los elementos más relevantes del IGT, y, por otro lado, describen la forma en la que los agentes le asignan valor a las alternativas y eligen las alternativas. A continuación, se describen los parámetros y reglas del modelo.

a) El ambiente está compuesto por cuatro mazos que difieren en la Magnitud de Ganancias (G), Magnitud de Pérdidas (P) y Probabilidad de pérdidas (Pp). Con el fin de normalizar el valor de la magnitud de las ganancias y pérdidas de la tarea original, se dividieron los valores entre el valor máximo de cada uno de los mazos.
b) Cada unidad temporal (t) corresponde a un ensayo o elección. La simulación finaliza cuando las unidades temporales (t) son iguales al total de unidades temporales programadas (Ts).
c) Los agentes utilizan la regla de decisión *Igualación* para elegir uno de los mazos en una unida temporal. Por ejemplo, la probabilidad de que el agente elija el mazo "A" en una unidad temporal futura ($Pra(t+1)$), depende del valor del mazo "A" ($Va(t)$) dividido entre la suma del valor de todos los mazos ($Va(t) + Vb(t) + Vc(t) + Vd(t)$). Ecuación 1:

$$Pra(t+1) = Va(t) \,/\, (Va(t) + Vb(t) + Vc(t) + Vd(t)). \quad (1)$$

d) Cuando los agentes eligen un mazo, se calcula las ganancias netas (Gn) en la unidad temporal presente. Las Ganancias (G) son ponderadas por la sensibilidad a las ganancias (Gs), posteriormente se sustraen las pérdidas (P)

Tabla 1. Parámetros de simulaciones del modelo.

Parámetros	Símbolos	Valores			
Unidades temporales	*Ts*	100			
Magnitud de Ganancias	*G*	A	B	C	D
		100	100	50	50
Magnitud de Pérdidas	*P*	A	B	C	D
		1250	250	250	50
Probabilidad Pérdidas	*Pp*	A	B	C	D
		0.1	0.5	0.1	0.5
Sensibilidad Ganancias	*Gs*	0.1 - 1			
Sensibilidad Pérdidas	*Ps*	0.1 - 1			
Factor Memoria	*x*	0.1 – 0.9			

ponderadas por la sensibilidad a las pérdidas (*Ps*), en caso de haberlas. Ecuación 2:

$$Gn(\mathrm{t}) = G(\mathrm{t})\,Gs - P(\mathrm{t})\,Ps. \tag{2}$$

e) Una vez calculadas las ganancias netas, se actualiza el valor atribuido por el agente al mazo elegido a partir de la regla de aprendizaje *Operado lineal*. Por ejemplo, el valor del mazo "A" (*Va* (t)) dependerá de su valor en una unidad temporal pasada (*Va*(t-1)) ponderado por el factor de memoria (*x*), más las ganancias netas (*Gn*) ponderadas por el valor residual del factor de memoria (1-*x*). A medida que aumenta el factor de memoria, los agentes tenderán a ponderar en mayor medida el valor de las alternativas en el pasado, Ecuación 3:

$$Va(\mathrm{t}) = x\,Va(\mathrm{t}\text{-}1) + (1\text{-}x)\,Gn. \tag{3}$$

2.5. Simulaciones

En cada simulación, un agente individual era expuesto al IGT de la misma forma en la que fueron expuestos los participantes humanos. Con el objetivo de probar cual interacción de parámetros (*x*, *Gs*, *Ps*) resulta en una mayor y menor proporción de elecciones optimas, se simularon 50 agentes para cada una de las 900 combinaciones que diferían en el factor de memoria, sensibilidad a las pérdidas y sensibilidad a las ganancias, en intervalos de 0.1.

Posteriormente se simularon 50 agentes con los valores que resultaban en una mayor y menor proporción de elecciones optimas y se analizó su preferencia por cada uno de los mazos y el cambio de preferencia a lo largo de la tarea.

Finalmente, se eligió la combinación de parámetros que resultaba en la proporción de elecciones optimas de los participantes y se simulo 2 grupos de 30 agentes, con el objetivo de contrastar su ejecución con la de los participantes.

3. Resultados

En las Figuras 1a y 1b se muestra la proporción de elecciones optimas en función de la interacción de los parámetros. Cada punto corresponde al promedio de 50

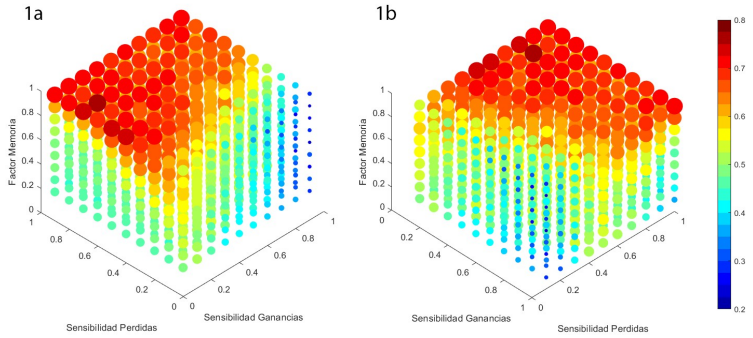


Fig. 1. Proporción de elecciones optimas en función de la interacción de los diferentes parámetros (factor de memoria, sensibilidad a las ganancias, sensibilidad a las pérdidas) en intervalos de 0.1. Cada punto corresponde al promedio de 50 agentes.

agentes individuales, puntos más grandes y rojos, señalan mayor proporción de elección de las alternativas optimas (C y D).

La combinación de parámetros que mostró una mayor preferencia por las alternativas optimas fue $x = 0.9$, $Gs = 0.2$ y $Ps = 0.8$, mientras que la combinación de parámetros que muestra una mayor preferencia por las alternativas subóptimas fue $x = 0.8$, $Gs = 1$ y $Ps = 0.1$. Como tendencias generales se puede observar que: 1) conforme aumenta la sensibilidad a las ganancias y decrece la sensibilidad a las pérdidas, aumenta la preferencia de los agentes por las alternativas subóptimas y 2) cuando la sensibilidad a las pérdidas es mayor a 0.3, conforme aumenta el factor de memoria, aumenta la preferencia por las alternativas óptimas.

En las Figuras 2a se muestra la proporción de elección de cada uno de los mazos por agente y su promedio. Para el grupo A se utilizó la interacción de parámetros que desemboco en una mayor preferencia por las alternativas optimas, mientras que para el grupo B se utilizó la interacción de parámetros de desemboco en una menor preferencia por las alternativas óptimas (ver, Figura 1).

La figura 2b muestra la preferencia por las alternativas optimas a lo largo de 5 bloques de 20 elecciones o cartas. Los resultados señalan que la preferencia del grupo A por las alternativas optimas tendió a incrementar, mientras que para el grupo B tendió a decrementar. Esto sugiere que los agentes aprendieron las propiedades de las alternativas y que modificaron su preferencia en función de la experiencia con la tarea. Estos resultados son similares a lo observado en humanos [1].

Al igual que en la figura 2, en la Figura 3 se muestra la proporción de elección de cada uno de los mazos y la proporción de elecciones optimas a lo largo de 5 bloques de 20 elecciones. En la parte inferior se muestra la ejecución de los participantes, mientras que en la parte superior se muestra la ejecución de 30 agentes por grupo. Los participantes se dividieron en 2 grupos en función de la media de la proporción de elecciones óptimas.

El grupo A está compuesto aquellos participantes que tuvieron una preferencia superior o igual a la media, mientras que el grupo B está compuesto aquellos participantes que tuvieron una preferencia menor a la media. En el caso de los agentes, para el grupo A se utilizaron los parámetros que desembocaron en la media de elecciones óptimas de los participantes del grupo A ($x = 0.7$, $Gs = 0.5$ y $Ps = 0.2$.),

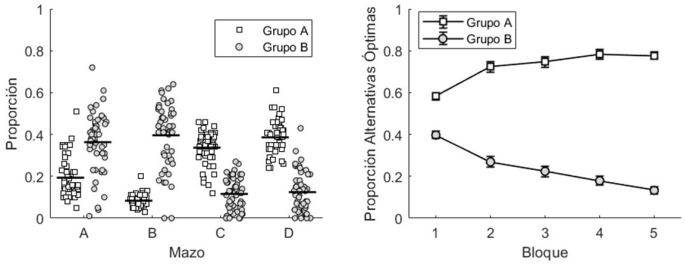


Fig. 2. Proporción de elección de cada uno de los mazos por agente y su promedio. Mientras que del lado derecho se muestra el promedio de la preferencia por las alternativas óptimas a lo largo de 5 bloques de 20 elecciones o cartas.

mientras que para el grupo B se utilizaron los parámetros que desembocaron en la media de elecciones óptimas de los participantes del grupo B (x = 0.3, Gs = 0.9 y Ps = 0.1.).

Al igual que los participantes, la preferencia por las alternativas óptimas o subóptimas aumenta en cada bloque. A pesar de que los resultados son cualitativamente similares entre los agentes y los participantes, es importante notar que los agentes y participantes del grupo B difieren en su preferencia por los mazos subóptimos. Los participantes eligieron el mazo con menor probabilidad (A) de castigo, mientras que los agentes eligieron el mazo con mayor probabilidad de castigo (B).

4. Discusión

El estudio propone un modelo de agente que simula la ejecución en el IGT a partir del *Operador lineal* (regla de aprendizaje), *Igualación* (regla de decisión) y dos parámetros asociados a la sensibilidad a las ganancias y pérdidas. Así mismo simula cualitativamente la preferencia de los participantes por las alternativas óptimas y/o subóptimas.

En relación al modelo de agente, los resultados señalan que conforme aumenta la sensibilidad a las ganancias y decrece la sensibilidad a las pérdidas, los agentes muestran mayor preferencia por las alternativas subóptimas. Bechara, et al. [1] hipotetizan que la preferencia de los participantes con daño ventromedial en la corteza prefrontal por las alternativas subóptimas puede deberse a la alta sensibilidad a las ganancias inmediatas y la insensibilidad a los castigos.

Así mismo, se ha reportado que los participantes con daño ventromedial en la corteza prefrontal muestran menor grado de respuestas anticipatorias de conductancia de la piel ante las alternativas con mayor probabilidad de pérdidas [2, 3]. Los resultados del modelo son consistentes en relación a la sensibilidad a las pérdidas de los participantes con daño ventromedial en la corteza prefrontal.

Por otro lado, cuando la sensibilidad a las pérdidas y el factor de memoria fueron mayores, los agentes mostraron preferencia por las alternativas óptimas. Este resultado es consistente con la hipótesis de Bechara, et al. [1] en la cual señala que los participantes con daño ventromedial en la corteza prefrontal muestran preferencia por

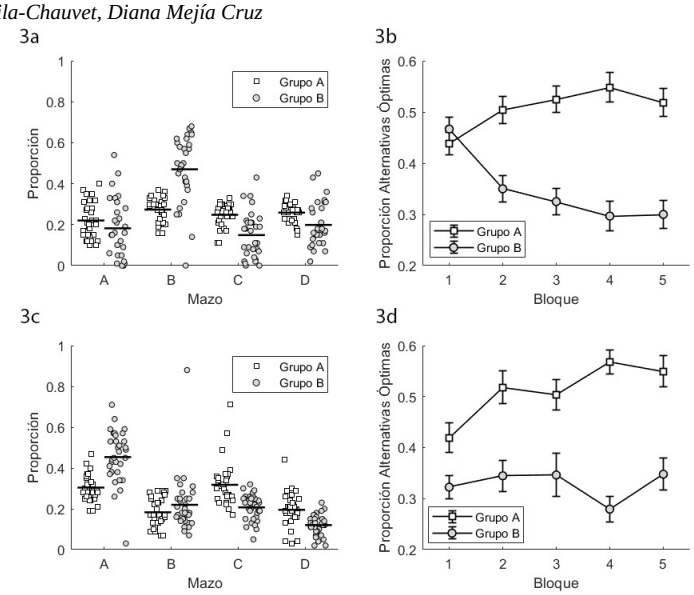


Fig. 3. Proporción de elecciones para cada uno de los mazos y la proporción de elecciones óptimas en cada uno de los bloques. En la parte superior se muestra la ejecución de los agentes, mientras que en la parte inferior se muestra la ejecución de los participantes. Los participantes se dividieron en 2 grupos, grupo "A" superior a la media y grupo "B" inferior a la media.

las alternativas subóptimas debido a que su comportamiento está guiado por las consecuencias inmediatas.

A medida que aumenta el factor de aprendizaje en los agentes, estos tienden a ponderan menos las consecuencias inmediatas y ponderar más las ganancias obtenidas a lo largo del tiempo. En relación a la simulación de la ejecución de los participantes, se tomaron los valores de los parámetros que desembocaron en la media de la proporción de elecciones óptimas de los participantes que tendieron a elegir las alternativas óptimas (Grupo A) y subóptimas (Grupo B).

El factor de memoria y la sensibilidad a las pérdidas fue mayor para el Grupo A, mientras que la sensibilidad a las ganancias fue mayor para el Grupo B. Estos resultados son consistentes con las hipótesis de Bechara, et al. [1] sobre los factores que afectan la elección de las alternativas óptimas.

Finalmente, los participantes mostraron mayor preferencia por las alternativas con menor frecuencia de castigos, en contraste con los agentes. Esto sugiere que los participantes están dispuestos a tomar riesgos de manera ocasional con el objetivo de aumentar sus ganancias, mientras que los agentes evitan las pérdidas de mayor magnitud.

En conclusión, se muestra un modelo de agente relativamente *simple*, capaz de simular la toma de decisiones en el IGT. Así mismo, el modelo simula cualitativamente la ejecución de jóvenes adultos a partir de la manipulación del factor de memoria y los parámetros de sensibilidad a las pérdidas y ganancias.

En el futuro se espera evaluar la flexibilidad cognitiva, la sensibilidad a las ganancias y la sensibilidad a las pérdidas en muestras clínicas y comunitarias con el

objetivo de alimentar al modelo y simular la preferencia por las alternativas óptimas. Se espera que los conjuntos de estos factores puedan fungir como un buen predictor de la toma de decisiones y comportamientos asociados a conductas de riesgo.

Referencias

1. Bechara, A., Damasio, A. R., Damasio, H., Anderson, S. W.: Insensitivity to future consequences following damage to human prefrontal cortex. Cognition, vol. 50, pp. 7–15 (1994) doi: 10.1016/0010-0277(94)90018-3
2. Bechara, A., Damasio, H., Tranel, D., Damasio, A. R.: Deciding advantageously before knowing the advantageous strategy. Science, vol. 275, pp. 1293–1295 (1997) doi: 10.11 26/science.275.5304.1293
3. Bechara, A., Tranel, D., Damasio, H., Damasio, A. R.: Failure to respond autonomically to anticipated future outcomes following damage to prefrontal cortex. Cerebral cortex, vol. 6, pp. 215–225 (1996) doi: 10.1093/cercor/6.2.215
4. Beitz, K. M., Salthouse, T. A., Davis, H. P.: Performance on the Iowa Gambling task from 5 to 89 years of age. Journal of Experimental Psychology: General, vol. 143, no. 4, pp. 1677–16689 (2014)
5. de Visser, L., Homberg, J., Mitsogiannis, M., Zeeb, F., Rivalan, M., Fitoussi, A.: Dellu-Hagedorn, F.: Rodent versions of the Iowa Gambling task: Opportunities and challenges for the understanding of decision-making. Frontiers in neuroscience, vol. 5, no. 109 (2011) doi: 10.3389/fnins.2011.00109/full
6. Horstmann, A., Villringer, A., Neumann, J.: Iowa Gambling task: There is more to consider than long-term outcome. Using a linear equation model to disentangle the impact of outcome and frequency of gains and losses. Frontiers in Neuroscience, vol. 6, no. 61 (2012) doi: 10.3389/fnins.2012.00061/full
7. Kacelnik, A.: Meanings of rationality. Rational animals, pp. 87-106 (2006)
8. Steingroever, H., Wetzels, R., Wagenmakers, E. J.: Validating the PVL-Delta model for the Iowa Gambling task. Frontiers in psychology, vol. 4, no. 898 (2013) doi: 10.3389/ fpsyg.2013.00898/full
9. Steingroever, H., Wetzels, R., Wagenmakers, E. J.: Absolute performance of reinforcement-learning models for the Iowa Gambling task. Decision, vol. 1, no. 3, pp. 161-183 (2014) doi: 10.1037/dec0000005
10. van den Bos, R., Lasthuis, W., Den Heijer, E., van der Harst, J., Spruijt, B.: Toward a rodent model of the Iowa Gambling task. Behavior research methods, vol. 38, no. 3, pp. 470–478 (2006) doi: 10.3758/BF03192801
11. Wetzels, R., Vandekerckhove, J., Tuerlinckx, F., Wagenmakers, E. J.: Bayesian parameter estimation in the expectancy Valence model of the Iowa Gambling task. Journal of Mathematical Psychology, vol. 54, no. 1, pp. 14–27 (2010) doi: 10.1016/j.jmp.20 08.12.001

ISSN 1870-4069

Generador de problemas de ecuaciones no lineales con gamificación

Hugo Pablo Leyva[1], Rafaela Blanca Silva López[2]

[1] Universidad Autónoma Metropolitana Azcapotzalco, México

[2] Universidad Autónoma Metropolitana Lerma, México

hpl@correo.azc.uam.mx, r.silva@correo.ler.uam.mx

Resumen. Uno de los problemas más laboriosos y de alguna manera tediosos, de cada periodo escolar de un curso de métodos numéricos, es la generación de problemas para cada uno de los temas. Se requieren diferentes problemas para utilizarse como ejemplos de clase, tareas o exámenes. Por lo tanto, se requiere contar con una amplia gama de problemas distintos para cada periodo escolar para evitar que los estudiantes tengan las respuestas. Los problemas generados deben contar con su solución, además de poderlos resolver en un tiempo equivalente al que se da para resolver el examen. El objetivo del presente trabajo es presentar un generador de problemas para el tema de raíces de ecuaciones no lineales. El conjunto de programas elaborados permite la generación de una gran cantidad de problemas con su solución, que un estudiante puede resolver en el tiempo que se da para realizar los exámenes. Uno de los enfoques con los cuales se puede dar un curso es el de gamificación, que esencialmente trata de usar un ambiente de juego en ámbitos tan diversos como la educación, o la salud, entre otros. Los problemas que se generan en el presente trabajo se enfocan en ecuaciones cuyas raíces tienen una interpretación en un ambiente gamificado. Se presentan los algoritmos para la elaboración de los programas, la puesta en marcha y resultados en cuanto a la generación de problemas usados en el tema de raíces de ecuaciones no lineales para el curso de Métodos Numéricos en Ingeniería de la UAM Azcapotzalco.

Palabras clave: E-learning, gamificación, aprendizaje basado en problemas, métodos numéricos, ecuaciones no lineales, algoritmos inteligentes.

Problem Generator of Nonlinear Equations with Gamification

Abstract. One of the most laborious and somehow tedious problems, of each school period of a course of numerical methods, is the generation of problems for each of the topics. Different problems are required to be used as examples of class, assignments, or exams. Therefore, a wide range of different problems are required for each school term to prevent students from having the answers. The problems generated must have their solution, in addition to being able to solve them in a time equivalent to that given to solve the exam. The objective of this work is to present a problem generator for the root theme of nonlinear equations.

The set of elaborate programs allows the generation of a lot of problems with their solution, which a student can solve in the time given to take the exams. One of the approaches with which a course can be given is gamification, which essentially tries to use a gaming environment in areas as diverse as education, or health, among others. The problems generated in this work focus on equations whose roots have an interpretation in a gamified environment. Algorithms for program development, commissioning and results are presented in terms of the generation of problems used in the topic of roots of nonlinear equations for the course of Numerical Methods in Engineering of the UAM Azcapotzalco.

Keywords: E-learning, gamification, problem-based learning, numerical methods, nonlinear equations, intelligent algorithms.

1. Introducción

Una de las labores docentes que demandan más tiempo es la generación de problemas para un curso como el de Métodos Numéricos. Hay que cuidar detalles como la complejidad que podría llevar a un tiempo largo de solución por parte de los estudiantes. Es necesario, contar con la solución del problema para que dado el caso se proporcione retroalimentación al estudiante si la solicitase. Se debe contemplar la elaboración de problemas que se utilizarán como ejemplos en clase, otros distintos para usarse en las tareas y algunos más para los exámenes. Esto evita que los estudiantes puedan copiarse con facilidad.

El aprendizaje basado en problemas puede ser usado en cursos de ingeniería y ciencias como lo es el de Métodos numéricos con excelentes resultados. Precisamente para este curso, se requiere una cantidad enorme de problemas. Otra manera de hacer más llevadera la clase de un curso como el de métodos numéricos es la gamificación [1]. Al darle un aire de juego a las actividades de la clase permite atrapar el interés del alumnado [2].

El e-learning, se ha convertido en una modalidad de conducción del proceso de enseñanza y aprendizaje vital en los tiempos actuales de la pandemia. Este permite atender a un mayor número de estudiantes empleando un LMS (Learning Management System). Este tipo de herramientas, tienen la virtud de contar con facilidades para gestionar una batería de problemas, tal que se pueden aplicar muchas versiones de un examen sin que estos se repitan. Para esto requieren de igual manera una cantidad considerable de problemas.

Uno de los temas más relevantes de un curso de Métodos numéricos es el de raíces de ecuaciones no lineales. Este tema se encuentra con mucha frecuencia en otros cursos de ingeniería en las Instituciones de Educación Superior (IES). Para contar con el número suficiente de problemas, se podrían copiar de algún libro o página web, sin embargo, sería difícil encontrar suficientes problemas para cambiarlos en cada ciclo escolar.

Otra dificultad radica en que si se descubre cual es la fuente de la que se tomaron los problemas, los estudiantes podrán tener acceso a ellos de un curso a otro y perderán su validez como instrumentos de evaluación.

Por lo anterior, se proponen el desarrollo de algoritmos inteligentes y el desarrollo de programas para la generación de problemas tales que: se cuente con un gran número

de problemas (miles) para evitar que se compartan por los estudiantes, se tenga la solución para dar retroalimentación a los alumnos, se generen en poco tiempo, se puedan usar para ejercicios, tareas o exámenes, así como para alimentar la batería de problemas que requiere un LMS para aplicar exámenes sin repetición de los problemas.

Las características de los programas contemplan:

- Generar problemas para el tema de raíces de ecuaciones no lineales de la asignatura de Métodos Numéricos en Ingeniería, junto con su solución.
- La cantidad de problemas es variable.
- Los parámetros de solución (máximo de iteraciones, cifras significativas, etc.) se pueden configurar.
- Sólo se generan problemas que se resuelvan en pocas iteraciones.
- Se apliquen filtros adicionales para generar problemas no tan laboriosos para los estudiantes.
- Las soluciones de los problemas se pueden interpretar en el contexto de un problema gamificado.
- Se genera también si se desea, un archivo que corresponde a una batería de problemas que tiene el formato para ser usada en un LMS.

Para cumplir con estas características y restricciones se diseña una serie de algoritmos inteligentes que satisfacen todas las necesidades planteadas.

2. Trabajos relacionados

Los autores que abordan el concepto de gamificación, consideran que es una táctica utilizada para motivar a las personas en actividades ordinarias [3], usar el juego en un contexto no relacionado con el mismo para cambiar el comportamiento de las personas [4]. La gamificación [5] es un motivador en el aprendizaje del alumno, de hecho aumenta el interés y la motivación para abordar un tema específico [2]. En [6] se preparan ejercicios en Excel para usarlos en un enfoque de aula invertida.

Los autores se enfocan más en cómo implementar su curso que del desarrollo de ejercicios. En [7] preparan problemas a partir de ejercicios de libros de texto que tienen un enfoque de aprendizaje adaptativo orientado a la trayectoria de presentación del material de un curso de métodos numéricos.

En [8] se proponen ejercicios diseñados por expertos que son validados con software numérico para desarrollar material de enseñanza. Los problemas son resueltos con Maple o Matlab. Se enfocan más en el desarrollo del material y su uso para la enseñanza. En [9] diseñaron ejercicios usando el software de Mathematica con objetos de aprendizaje para el curso de Métodos Numéricos.

En [10] usan GeoGebra que es un sitio web dónde se cuentan con recursos para enseñanza de las matemáticas y ciencias. Los ejercicios se emplean en un curso de métodos numéricos montado en el LMS Moodle. En [11] usan la herramienta web e-status para generar problemas numéricos. Los problemas generados son de estadística, más que de métodos numéricos.

En [12] en lugar de emplear un software que genere los problemas, se van a un enfoque más simple que es usar de un repositorio los problemas ya hechos. Los problemas son de ingeniería química, son una colección de 10 problemas numéricos.

Se trata de problemas típicos de los ingenieros químicos, cuya solución es numérica. Se analiza la solución de estos con varios softwares.

En [13] se usa OptiA, este software se emplea para resolver problemas de programación matemática, que son problemas de optimización. No se pretende obtener un generador de problemas. No se identifica algún trabajo en el que se propongan algoritmos inteligentes ni se desarrolle una aplicación que genere los problemas con sus soluciones satisfaciendo las características y restricciones mencionadas. Los programas propuestos en este trabajo cubren estas deficiencias de manera efectiva.

3. Algoritmos inteligentes y programas para la generación de problemas

Se diseñaron algoritmos que satisfacen las restricciones establecidas para la generación de problemas de raíces de ecuaciones no lineales. Contemplando la estrategia didáctica de gamificación en la enseñanza de las ecuaciones generadas. Se analiza un caso de uso gamificado en el que se utilizan ecuaciones no lineales.

Caso de uso: Para cobrar la herencia.

A tu difunto tío, le encantaban las matemáticas y los acertijos. Como dudaba de que todos sus sobrinos fueran merecedores de su gran fortuna, les dejo un acertijo en su testamento. Al momento de leerlo el albacea menciona la condición que debe cubrir el heredero que se quedará con toda la fortuna. "Mi fortuna será para aquel de mis sobrinos que se presente a reclamar la herencia en la fecha y hora señaladas, en las oficinas del albacea. Solo aquel que llame a la puerta el día y a la hora correcta podrá entrar. Se descalificará a cualquiera que pretenda quedarse a esperar ante la puerta". La fecha y la hora están dadas por las soluciones del siguiente par de ecuaciones:

$$a_2 f(x) + a_1 x^n + a_0 = 0.$$

$$b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0 = 0.$$

"El que se presente a reclamar la herencia deberá de entregar por escrito, la solución junto el razonamiento que empleó para hallarla. En el caso de que 2 o más herederos se presenten a la hora y fecha correctas, el ganador será aquel que tenga la solución más precisa acorde a los requerimientos de los valores a encontrar y con el menor número posible de pasos. En el supuesto de empate, el albacea les proporcionara más problemas a resolver en ese momento. Ganará el que resuelva todos en las mismas condiciones que el problema que los llevo hasta ahí".

Si tú fueras uno de los sobrinos ¿Cuál debe ser la metodología más adecuada para resolver la ecuación y así poder reclamar la herencia?

En este problema f(x) es una ecuación no lineal que puede ser alguna de las siguientes: $\mathrm{sen}(x)$, $\cos(x)$, $\mathrm{senh}(x)$, $\cosh(x)$, $\mathrm{tgh}(x)$, $\mathrm{ctgh}(x)$, $\mathrm{sech}(x)$, $\mathrm{csch}(x)$, e^x, e^{-x}, e^{x^2}, e^{-x^2}, $ln(x)$, $log(x)$, $\sqrt{x}$, $\sqrt[3]{x}$, $\frac{1}{x}$.

Las constantes son: $a_2, a_1, a_0, n, b_4, b_3, b_2, b_1, b_0$.

El grado del término polinomial n varia de 1 a 4.

Los coeficientes de la ecuación no lineal y el grado del término polinomial se generan aleatoriamente.

Los coeficientes del polinomio se obtienen al aproximar por la serie de Taylor a un polinomio de grado 4 la ecuación no lineal. Las raíces reales de este polinomio se usan para generar aproximaciones a las raíces de la ecuación no lineal. El polinomio se resuelve analíticamente con la solución general de la ecuación de tercer o cuarto grado según se requiera [14]. Solo se consideran ecuaciones no lineales cuyo polinomio de Taylor tengan raíces reales.

Las raíces tanto del polinomio como de la ecuación no lineal se deben poderse interpretar ya sea como fechas u horas. Para esto se toman las primeras seis cifras de las raíces considerando el punto decimal y se interpretan de la siguiente manera:

- Si las 2 primeras cifras delante del punto decimal están entre 1 y 23 se puede tratar ya sea de una fecha o una hora.
- Si estas cifras están entre 24 y 31 se puede tratar de una fecha.
- Si son 0 se puede tratar de una hora.
- Si las segundas 2 cifras después del punto decimal están entre 01 y 12 se puede tratar ya sea de una fecha o una hora.
- Si estas cifras están entre 13 y 59 se puede tratar de una hora.
- Si son 00 se puede tratar de una hora.
- Si las terceras 2 cifras están entre 00 y 59 se puede tratar ya sea de una fecha o una hora.
- Si estas cifras están entre 60 y 99 se puede tratar de una fecha.
- Las fechas tienen el formato día/mes/año.
- Los días varían de 1 a 31.
- Los meses varían de 01 a 12.
- Los años que se manejan con 2 cifras varían de 00 a 99.
- Las horas tienen el formato HH:MM:SS.
- Las horas varían de 0 a 23.
- Los minutos varían de 00 a 59.
- Los segundos varían de 00 a 59.

Al procesar las 6 primeras cifras de cada raíz se deben de tener combinaciones validas conforme a las reglas anteriores. También se deben considerar el número de días que tiene cada mes, inclusive en año bisiesto.

Se descartan las raíces negativas y complejas.

Es posible que una raíz se pueda interpretar tanto como fecha u hora.

En la figura 1 se tiene el diagrama de flujo del algoritmo que genera las ecuaciones no lineales y polinomios.

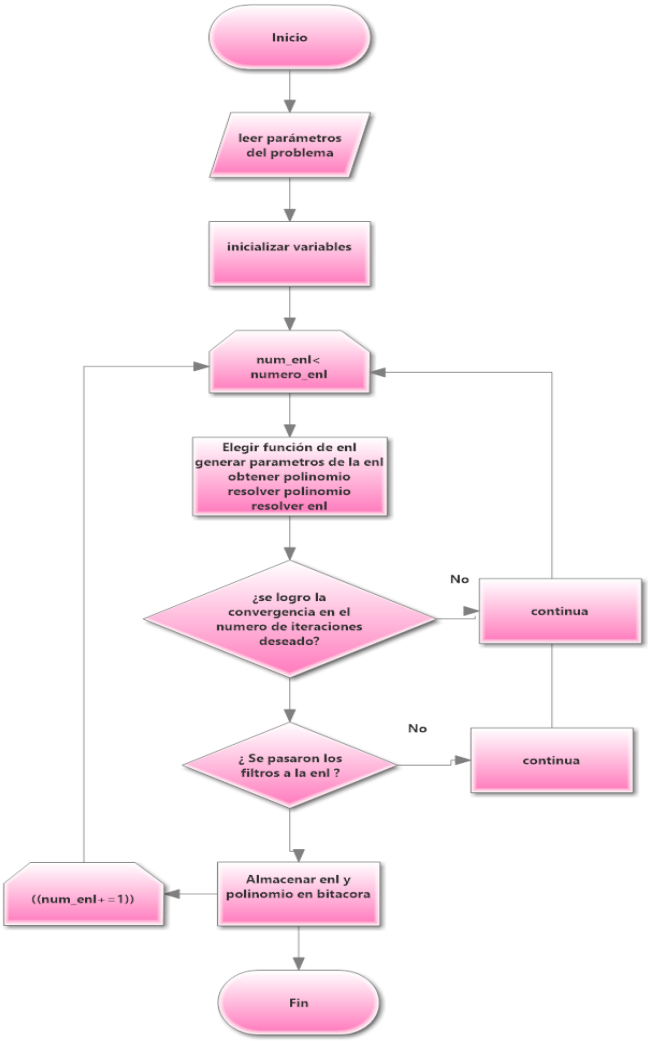


Fig. 1. Algoritmo para generar ecuaciones no lineales y polinomios.

La ecuación no lineal se resuelve usando el método de Newton Rapshon modificado [15]. Se prefiere este método por su velocidad de convergencia y por ser inmune a problemas de convergencia que se pueden presentar en caso de tener raíces dobles.

Entre los parámetros que se proporcionan están:

- El número de problemas a generar.
- El nombre del directorio donde se almacenan los problemas junto con su solución y las bitácoras con el listado final de problemas generados.
- El límite del rango de valores para el coeficiente a_2 que se generara aleatoriamente.

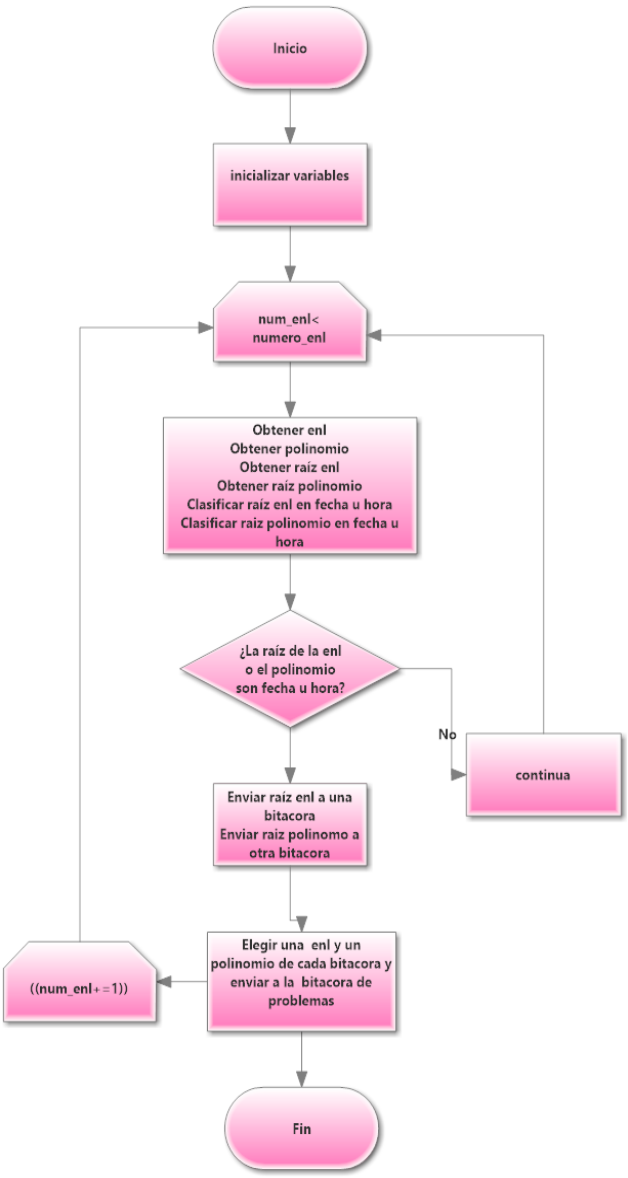


Fig. 2. Algoritmo para generar parejas de ecuaciones no lineales y polinomios.

- El límite del rango de valores para el coeficiente a_1 que se generara aleatoriamente.
- El límite del rango de valores para el coeficiente a_0 que se generara aleatoriamente.

- El número de cifras significativas con el que se desea encontrar las raíces.
- El número máximo de iteraciones que se usara para el método de Newton Rapshon modificado.
- El número de cifras que se usaran para hacer los cálculos.

Los filtros que se mencionan son:

- Los coeficientes a_2, a_1, a_0 deben de ser distintos a cero para que se pueda considerar una ecuación no lineal con el formato que se usa.
- El polinomio de Taylor asociado a la ecuación no lineal debe tener al menos una raíz real.

En la figura 2 se tiene el diagrama de flujo del algoritmo para obtener las parejas de ecuaciones no lineales y polinomios cuyas raíces se pueden interpretar como fechas u horas, que se requieren para el problema gamificado.

El algoritmo de la figura 2 contempla las siguientes restricciones:

- Se descartan las raíces negativas. Se consideran solo las primeras 6 cifras de la raíz considerando el punto decimal.
- Se verifica que las seis cifras cumplan el formato de fechas u horas mencionado antes.
- Se clasifica en raíces que se interpretan solo como fechas, solo como horas o ambiguas.
- Se forman parejas de una ecuación no lineal y un polinomio que tengan una raíz que se interpreta como fecha y otra como hora. También se considera que solo una raíz se defina como fecha u hora y la otra que pueda ser ambigua.

Los programas se elaboraron en los lenguajes Korn Shell, C, Latex, HTML. Como se usó el sistema operativo Linux se emplearon Shell scripts en Korn Shell para la generación de las ecuaciones y manipulación de los archivos generados con la lista de problemas, así como con su solución. La solución de los polinomios y las ecuaciones no lineales se programaron en lenguaje C. El programa genera páginas Web con la lista de problemas y otra con las soluciones. Estas páginas están en HTML.

Para generar el listado de las ecuaciones en formato algebraico se usó Latex. Se utilizan librerías de Linux para obtener las ecuaciones como imágenes en una página web. Para realizar los cálculos en los Shell scripts de Korn Shell se usó la herramienta BC, la cual es una calculadora que viene con el sistema operativo. En esta se especifica la cantidad de cifras que se emplean al hacer cálculos. Las funciones que generan valores aleatorios se hicieron con facilidades que proporciona Korn Shell.

3.1. Pruebas

Se hicieron 10 corridas con los siguientes parámetros cada una:

- Numero de ecuaciones a generar 1000.
- Directorio donde almacenan los problemas generados T21I_# donde el # vario de 0 a 9.
- El límite del rango de valores para el coeficiente a_2 es 100.
- El límite del rango de valores para el coeficiente a_1 es 10.

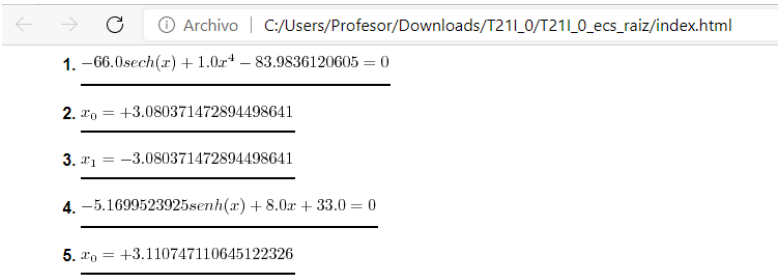


Fig. 3. Listado de ecuaciones no lineales generadas con su solución.

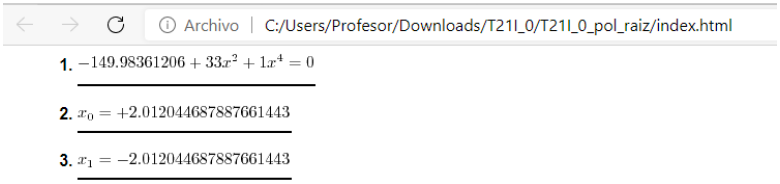


Fig. 4. Listado de polinomios generados con su solución.

- El límite del rango de valores para el coeficiente a_0 es 100.
- El número de cifras significativas para encontrar las raíces es 8.
- El número máximo de iteraciones es 10.
- El número de cifras para hacer los cálculos es 10.

Para realizar las corridas se hizo con el siguiente comando en una terminal de Linux:

$ for ((i=0; ((i<10)) ; ((i+=1)))) do nohup time generaecuacionesaleatorias.ksh 1000 T21I_0$i 100 10 100 8 10 10 & done El ciclo for del comando anterior se ejecutará un total de 10 veces. El nohup junto con el carácter & en el comando anterior permiten que el programa se quede en ejecución aun si la terminal llegara a cerrarse. El time devolverá el tiempo de ejecución de cada corrida del programa. Las 10 corridas se hacen concurrentemente.

El primer argumento son el número de problemas a generar en este caso 1000. El segundo argumento es el nombre de cada directorio donde se almacenarán los problemas generados. En este caso serán T21I_0, T21I_1, … T21I_9. El tercer argumento es límite del rango de valores aleatorios que se generan para el término que corresponde a alguna de las funciones que se emplean en este programa. En este caso 100. El cuarto argumento es límite del rango de valores aleatorios que se generan para el termino polinomial.

En este caso 10. El quinto argumento es límite del rango de valores aleatorios que se generan para el termino independiente. En este caso 100. El sexto argumento son las cifras significativas con las que encontrarán las raíces de las ecuaciones no lineales. En este caso 8. El séptimo argumento es el máximo de iteraciones con las que se realizara el método numérico que resuelve la ecuación no lineal. En este caso 10. El octavo argumento son las cifras con las que el comando BC realizara los cálculos para generar las ecuaciones no lineales o polinomios. En este caso 10. Se muestra a continuación parte de las páginas web generadas con la lista de problemas y sus soluciones.

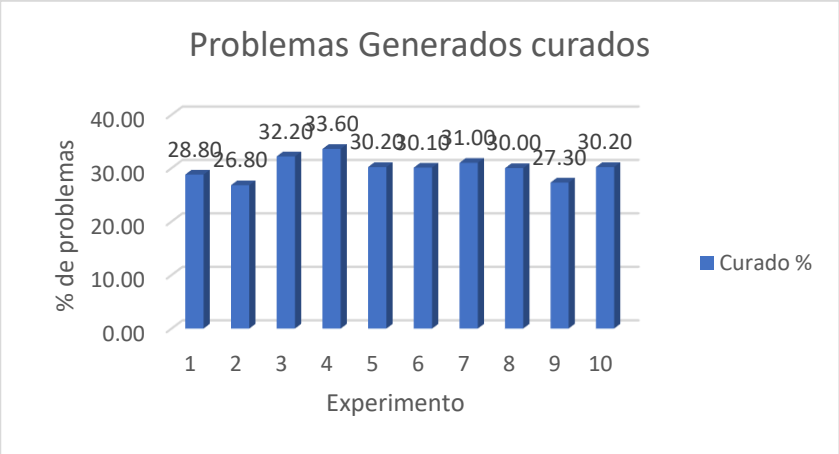


Fig. 5. Porcentaje de problemas generados curados.

Tabla 1. Características del servidor y tiempo de corrida.

Sistema operativo	CPU	Memoria RAM	Disco duro	Tiempo de corrida
Linux CentOS 6.7	24 Intel	268 Gb	7 Tb	30 min
Linux CentOS 7.8	2 Intel	3 Gb	214 Gb	3 hr 30 min

De los 1000 problemas generados, sólo el 30% aproximadamente, son interpretados adecuadamente en el contexto de gamificación del caso propuesto.

Se entiende por problemas curados aquellos cuya solución se puede interpretar como fechas u hora. En la Tabla 1 se muestran las características de 2 servidores donde se hicieron las pruebas con su tiempo de ejecución. Un ejemplo de un problema generado por el programa es: Para cobrar la herencia.

A tu difunto tío, le encantaban las matemáticas y los acertijos. Como dudaba de que todos sus sobrinos fueran merecedores de su gran fortuna, les dejo un acertijo en su testamento. Al momento de leerlo el albacea menciona la condición que debe cubrir el heredero que se quedará con toda la fortuna. "Mi fortuna será para aquel de mis sobrinos que se presente a reclamar la herencia en la fecha y hora señaladas, en las oficinas del albacea. Solo aquel que llame a la puerta el día y a la hora correcta podrá entrar. Se descalificará a cualquiera que pretenda quedarse a esperar ante la puerta". La fecha y la hora están dadas por las soluciones del siguiente par de ecuaciones:

$$33e^{x} + 9x^{4} - 34.4176330566 = 0$$

$$-1.291666666x^{4} + 0x^{3} + 22.4110412597x^{2} + 0x - 97 = 0$$

"El que se presente a reclamar la herencia deberá de entregar por escrito, la solución junto el razonamiento que empleó para hallarla. En el caso de que 2 o más herederos se presenten a la hora y fecha correctas, el ganador será aquel que tenga la solución más precisa acorde a los requerimientos de los valores a encontrar y con el menor número

posible de pasos. En el supuesto de empate, el albacea les proporcionara más problemas a resolver en ese momento. Ganará el que resuelva todos en las mismas condiciones que el problema que los llevo hasta ahí".

Si tú fueras uno de los sobrinos ¿Cuál debe ser la metodología más adecuada para resolver la ecuación y así poder reclamar la herencia?

La ecuación no lineal tiene 2 raíces:

x_0=+0.04206064203937895374

x_1=-1.295971852465515717

Solo se considera la raíz positiva. Tomando las primeras cifras tenemos 0.0420. La primera cifra es un cero por lo cual es una hora. Serían las 0:04:20, o sea las cero horas con cuatro minutos y veinte segundos.

El polinomio tiene 4 raíces:

x_0=+2.876013313714593789

x_1=-2.876013313714593789

x_2=+3.013143039440491222

x_3=-3.013143039440491222

Solo se consideran las raíces positivas. Tomando las primeras cifras tenemos 2.8760 y 3.0131. La primera raíz se descarta porque tenemos 87 como las segundas cifras, esta no se puede considerar ni una fecha, ni una hora. De la segunda puede ser una fecha o una hora. Como hora sería 03:01:31 que sería las tres horas con un minuto y treinta y un segundos. Como fecha sería 03/01/31 que sería el tres de enero de 2031.

Como la raíz de la ecuación no lineal se interpreta como una hora, la raíz del polinomio se interpretará como una fecha. Por todo lo anterior hay que tocar la puerta del albacea el 03/01/31 a las 0:04:20.

4. Conclusiones y trabajo a futuro

Se observa que se genera una gran cantidad de problemas (miles) en poco tiempo. De estos sólo un porcentaje pequeño alrededor del 30 % son válidos para el contexto de gamificación del caso de uso utilizado. Aun así, la cantidad de problemas son suficientes para usarlos como ejercicios en clase, tareas y exámenes, sin que se repitan y puedan ser compartidos por los estudiantes.

Dependiendo de la metodología para impartir el curso es suficiente con un solo caso de uso gamificado, ya que se usan los problemas generados en un solo tema del curso, y se puede dejar como tarea un solo problema. De esta forma el alumno no se verá abrumado por tener que resolver todos los problemas generados.

Los demás problemas pueden sugerirse que los resuelva como practica para que se prepare para el examen. El tiempo de solución de los problemas, cumple con la restricción del tiempo asignado para realizar el examen, ya que a lo mucho se tomará 10 iteraciones para ser resueltos por un método numérico como es el de Newton-Rapshon modificado.

Si no se considera un contexto de gamificación, se cuenta entonces con una gran cantidad problemas para el curso (100% de problemas generados). El tiempo de ejecución para generar unos 1000 problemas es razonable, aunque no se cuenten con muchos recursos de cómputo por parte del servidor.

El algoritmo que se usó para generar los problemas para un uso gamificado, redujo el total a un 30%, ya que deben de cumplir los requerimientos del caso de uso. Como trabajo futuro se pretende adaptar el algoritmo para generar las raíces que ya cumplan con los requisitos del problema gamificado, y con estas generar los problemas con las ecuaciones requeridas.

Es probable que la dificultad que se puede tener con este enfoque es que las ecuaciones resultantes tarden más de las iteraciones requeridas para que el problema se pueda resolver en el tiempo destinado a un examen.

Se analizarán otros casos de uso para problemas gamificados tales que, las raíces de las ecuaciones no lineales se puedan interpretar de acuerdo con los requerimientos establecidos en el caso de uso. Por ejemplo, que las raíces sean: edades, coordenadas GPS, pesos, estaturas, entre otros.

Estos casos de uso se podrán usar para darle variedad a los problemas gamificados que se dejarán como tarea al alumno para llamar su atención y no aburrirlo con un solo ejemplo de caso de uso. También se permitirá cambiar el caso de uso de un trimestre a otro. Por último, se podrá dejar diferentes casos de uso a cada estudiante.

Referencias

1. Genç, Z., Tinmaz, H.: The perception on fundamentals of online courses a case on prospective instructional designers. European Journal of Contemporary Education, vol. 15, no. 1, pp. 163–172 (2016) doi: 10.13187/ejced.2016.15.163
2. Simões, J., Díaz, R., Vilas, A.: A social gamification framework for a K-6 learning platform. Computers in Human Behavior, vol. 29, no. 2, pp. 345–353 (2013) doi: 10.1016/j.chb.2012.06.007
3. Deterding, S., Dixon, D., Khaled, R., Nacke, L. E.: From game design elements to gamefulness: Defining gamification. In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environ, pp. 9–15 (2011) doi: 10.1145/2181037.2181040
4. Aristizábal, J. H., Colorado, H., Gutiérrez, H.: El juego como una estrategia didáctica para desarrollar el pensamiento numérico en las cuatro operaciones básicas. Sophia, vol. 12, no. 1, pp. 117–125 (2016)
5. Nah, F. F. H., Telaprolu, V. R., Rallapalli, S., Venkata, P. R.: Gamification of education using computer games. In. Yamamoto, S. (eds) Human Interface and the Management of Information, Information and Ineraction for Learning, Culture, Collaboration and Business, HIMI 2013, Lecture Notes in Computer Sciences, vol. 8018, no. 3, pp. 99–107 (2013) doi: 10.1007/978-3-642-39226-9_12
6. Pramita, M., Sukmawati, R. A., Sari, D. P.: The implementation of flipped classroom assisted by learning management system for numerical method courses. vol. 274, pp. 158–162 (2018) doi: 10.2991/iccite-18.2018.36
7. Clark, R. M., Kaw, A.: Adaptive learning in a numerical methods course for engineers: Evaluation in blended and flipped classrooms. Computer Applications in Engineering Education, vol. 28, no. 1, pp. 62–79 (2020) doi: 10.1002/cae.22175
8. Handayani, A., Herman, D. T., Fatimah, S.: Developing teaching material software assisted for numerical methods. Journal of Physics: Conference Series, International Conference on Mathematics and Science Education, vol. 895, no. 1 (2017) doi: 10.1088/1742-6596/895/1/012068

9. Caligaris, M., Rodríguez, G., Favieri, A., Laugero, L.: Developing mathematical abilities using learning objects when learning numerical methods. ICERI2017 Proc., vol. 1, pp. 6018–6026 (2017) doi: 10.21125/iceri.2017.1569
10. Becerra-Romero, A., Díaz-Rodríguez, M., González-Estrada, O. A.: Development of a virtual learning environment for the subject numerical methods under Moodle. J. Journal of Physics: Conference Series, vol. 1161, no. 1 (2019) doi: 10.1088/1742-6596/1161/1/012010
11. Alastrué, G., Antonio, J.: Problemas numéricos con corrección automática: qué se puede y qué se debe hacer. In: V Jornadas para la Enseñanza y Aprendizaje de la Estadística y la Investigación Operativa, pp. 35–52 (2014)
12. Cutlip M. B.: A collection of 10 numerical problems in chemical engineering solved by various mathematical software packages. Computer Applications in Engineering Education, vol. 6, no. 3, pp. 169–180 (1998) doi: 10.1002/(SICI)1099-0542(1998)6:3<169: AID-CAE6>3.0.CO;2-B
13. Doležal, J., Fidler, J.: Numerical solution of dynamic optimization problems using parametrization and Op^{ti}A software. vol. 78, NO. 2-3, pp. 111–121 (1996) doi: 10.1016/ 0096-3003(96)00002-1
14. Perry, R. H., Green, D. W., Maloney, J. O.: Perry manual del ingeniero químico. 3ra ed, México, DF: McGraw-Hill (1992)
15. Burden, R. L., Faires, D. J., Burden, A. M.: Análisis Numérico. 10a ed, México, D.F.: Cengage Learning Editores, S.A. de C.V (2017)

ISSN 1870-4069

Modelando y optimizando el problema de selección de portafolio de múltiples objetivos y parámetros difusos triangulares

Luis Mario Velasco Ocejo, Héctor Joaquín Fraire Huacuja,
Fausto Antonio Balderas Jaramillo

Tecnológico Nacional de México,
Instituto Tecnológico de Ciudad Madero,
México

luisvelasco.dev@gmail.com, {hector.fh,
fausto.bj}@cdmadero.tecnm.mx

Resumen. En este artículo se aborda el problema de optimización de la selección de portafolio considerando parámetros difusos triangulares. En la literatura de trabajos relacionados, esta variante del problema no ha sido abordada. En este trabajo se proponen un nuevo modelo matemático del problema y dos algoritmos de solución basados en la metodología del algoritmo genético basado en no dominancia. Uno de estos algoritmos utiliza el estimador de densidad Desviación de Dispersión Espacial, para mejorar la distribución de soluciones en los frentes de Pareto. Para evaluar el desempeño de los algoritmos propuestos, se realizaron una serie de experimentos con un conjunto de instancias de distintos tamaños. Las conclusiones de la evaluación experimental fueron validadas con la aplicación de dos pruebas no paramétricas de hipótesis: Wilcoxon y Friedman. Los resultados de la evaluación de dichas pruebas se reportan en el presente artículo.

Palabras clave: Parámetros difusos triangulares, frente de Pareto, desviación de dispersión espacial.

Modeling and Optimizing the Multi-Objective Portfolio Selection Problem with Triangular Fuzzy Numbers

Abstract. In this paper, the Multi-objective Portfolio Selection Optimization Problem with Triangular Fuzzy Parameters is approached. This variant of the problem has not been approached in related literature. In this work, a new mathematical model and two non-domination methodology-based genetic algorithms are proposed for solving the problem. One of them uses the Spatial Spread Deviation density estimator in order to increase the distribution of the solutions in Pareto fronts. In order to test the performance of the proposed algorithms, a series of experiments with different sized instances was carried out. Experiment results were confirmed by applying two non-parametric hypothesis tests: Wilcoxon and Friedman. The results of those tests are reported in this paper.

Keywords: Triangular fuzzy parameters, Pareto front, spatial spread deviation.

1. Introducción

A lo largo de la historia, el proceso de toma de decisiones ha representado una actividad crucial para todas las entidades del ramo organizacional, debido a que la elección de una buena o mala decisión impactará directamente en sus recursos y en el curso de acciones a tomar en el futuro.

Por ello, la toma de decisiones debe estar siempre influenciada por la información obtenida del análisis de los recursos y objetivos de la organización en cuestión. Sin embargo, dado que el proceso de toma de decisiones depende enteramente de dicha información, resulta especialmente complejo tomar una decisión cuando la información disponible es ambigua, imprecisa o incierta.

El problema de optimización de selección de portafolio es un problema de toma de decisiones al que las organizaciones suelen enfrentarse cotidianamente. El problema consiste en seleccionar un conjunto de proyectos (llamado portafolio) de entre todos los proyectos disponibles cuyo costo pueda ser solventado con los recursos disponibles y que maximice los beneficios que su realización le representan a la empresa.

Dado que en muchos casos la información disponible acerca de los costos y beneficios sólo se conoce parcial o aproximadamente, se han ideado diversos mecanismos para modelarla, siendo uno de ellos el uso de conjuntos difusos [1].

En el presente trabajo, se aborda el modelado de la incertidumbre presente en los parámetros del problema de optimización de selección de portafolio mediante números difusos triangulares.

2. Trabajos relacionados

Existe una gran variedad de mecanismos para abordar la incertidumbre de los parámetros del problema multiobjetivo de selección de portafolios con parámetros estáticos, en [2] por ejemplo, se proponen mecanismos estadísticos que permiten reducir la incertidumbre y modelar el proceso de toma de decisiones, lo cual requiere de un proceso de análisis para determinar las distribuciones de probabilidad adecuadas y verificar las dependencias estadísticas presentes, lo que puede resultar especialmente complejo para determinadas instancias del problema.

En [1] se aborda únicamente el modelado de las restricciones del problema mediante números difusos, lo cual evita la necesidad de verificar las dependencias estadísticas y la selección de distribuciones de probabilidad.

En otros trabajos, suele abordarse la incertidumbre presente únicamente en las restricciones [3] o únicamente en los beneficios [4] que representan los proyectos.

Por otro lado, otros trabajos suelen abordar una variante distinta del problema en la que tanto las restricciones como las funciones objetivo (beneficios) son modelados mediante números difusos, con la distinción particular del tipo de números difusos empleado para modelarlos.

En [5] por ejemplo, la imprecisión es modelada mediante números difusos del tipo intervalo y propone su resolución mediante el algoritmo I-NSGA-II, una variante del algoritmo NSGA-II adaptado para funcionar con parámetros difusos de dicho tipo.

Derivado del análisis de los trabajos relacionados antes mencionados, se proponen como aportaciones principales del presente trabajo el modelado del problema

empleando números difusos triangulares y su resolución mediante un nuevo algoritmo: Tri-NSGA-II.

De igual manera, se propone la modificación del esquema original del algoritmo NSGA-II para incorporar el estimador de densidad SSD, produciendo la variante Tri-NSGA-II-SSD del algoritmo.

3. Teoría de números difusos

Este capítulo contiene los conceptos principales de la teoría de números difusos abordados en el presente artículo.

3.1. Conjuntos difusos

Sea *X* un conjunto de elementos denotados de forma genérica como *x*, entonces un conjunto difuso *A* definido sobre *X* es un conjunto de pares ordenados: $A = \{\mu_A(x) | \ x \in X\}$. Donde $\mu_A(x)$ es conocido como función (o grado) de pertenencia (o membresía) de *x* en *A*, cuyo propósito es mapear a *X* en el espacio de membresía *M* [6].

Números difusos generalizados. Un número difuso generalizado, denotado comúnmente como $\tilde{A}$, es un subconjunto cualquiera del conjunto de números reales $\mathbb{R}$ cuya función de membresía $\mu_{\tilde{A}}(x)$ satisface las siguientes condiciones [7]:

1. $\mu_{\tilde{A}}(x)$ es un mapeo continuo de $\mathbb{R}$ en el intervalo [0, 1].
2. $\mu_{\tilde{A}}(x) = 0, -\infty < x < a.$
3. $\mu_{\tilde{A}}(x) = L(x)$, es estrictamente creciente para [a, b].
4. $\mu_{\tilde{A}}(x) = w$, para $b < x < a.$
5. $\mu_{\tilde{A}}(x) = R(x)$, es estrictamente decreciente para [α, β].
6. $\mu_{\tilde{A}}(x) = 0$, para $\beta < x < \infty.$

donde $0 < w < 1, a, b, \alpha, \beta$ son números reales.

Un número difuso generalizado *A* se denota como: $\tilde{A} = (a, b, \alpha, \beta, w)_{LR}$. Cuando *w=0*, el número se denota como $\tilde{A} = (a, b, \alpha, \beta)_{LR}$. Cuando *a=b* y *L(x)* y *R(x)* son líneas rectas, *A* es un número difuso triangular y se denota como $\tilde{A} = (a, \alpha, \beta)$. Una función de membresía triangular se define como:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x \le a_{\min} \\ \frac{x - a_{min}}{\beta - a_{min}}, & x \in (a_{min}, \beta) \\ \frac{a_{min} - x}{a_{min} - \beta}, & x \in (\beta, a_{min}) \\ 0, & x \ge a_{\max} \end{cases}, \quad (1)$$

3.2 Adición de números difusos triangulares

Dados dos números difusos triangulares $\tilde{A}_1 = (a_1, \alpha_1, \beta_1)$ y $\tilde{A}_2 = (a_2, \alpha_2, \beta_2)$, entonces [8]:

$$\tilde{A}_1 + \tilde{A}_2 = (a_1 + a_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2), \quad (2)$$

3.3 Integración de media graduada

La integración de Media Graduada [9] (*Graded Mean Integration,* GMI por sus siglas en inglés) es un método de normalización de números difusos (usualmente llamado método de *defuzzificación*) empleado para facilitar su comparación y establecer una relación de orden entre ellos. El número difuso mayor será aquel que cuente con más alto valor de GMI.

El valor de GMI de un número difuso $\tilde{A}$, denotado como $P(\tilde{A})$ se obtiene mediante la siguiente fórmula:

$$P(\tilde{A}) = \frac{\int_0^w h\left(\frac{L^{-1}(h)+R^{-1}(h)}{2}\right)dh}{\int_0^w hdh}, \quad (3)$$

En el caso particular de los números difusos triangulares, la ecuación (3) puede simplificarse de la siguiente manera:

$$P(\tilde{A}) = \frac{6a+\beta-\alpha}{6}. \quad (4)$$

3.4 Relación de orden entre números difusos triangulares

Dados dos números triangulares $\tilde{A}_1$ y $\tilde{A}_2$:

- $\tilde{A}_1 > \tilde{A}_2$ si y solo si $P(\tilde{A}_1) > P(\tilde{A}_2)$.
- $\tilde{A}_1 < \tilde{A}_2$ si y solo si $P(\tilde{A}_1) < P(\tilde{A}_2)$.
- $\tilde{A}_1 = \tilde{A}_2$ si y solo si $P(\tilde{A}_1) = P(\tilde{A}_2)$.

3.5 Dominancia

Con base en la definición de dominancia [10], es posible establecer una relación de dominancia entre dos vectores difusos: $X = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n\}$ y $Y = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_n\}$, donde $\tilde{x}_i$ y $\tilde{y}_i$ son números difusos triangulares. Se dice que *X* domina a *Y* si y solo si $\tilde{x}_i \geq \tilde{y}_i \; \forall \; i \in \{1,2, \dots, n\}$ y $\tilde{x}_i > \tilde{y}_i$ para al menos un valor de $i \in \{1,2, \dots, n\}$ [11].

4. Modelo matemático del problema multiobjetivo de selección de portafolio con parámetros difusos triangulares

Sean *n* el número de proyectos a considerar, $\tilde{C}$ el presupuesto total disponible, *O* el número de objetivos, $\tilde{c}_i$ el costo del proyecto *i*, $\tilde{b}_{ij}$ el beneficio obtenido de la realización del proyecto *i* en el objetivo *j*, *K* y *M* el número de áreas y regiones a considerar respectivamente, $\tilde{A}_k^{min}$ y $\tilde{A}_k^{max}$ los límites inferior y superior del presupuesto asignado al área *k*, $\tilde{R}_m^{min}$ y $\tilde{R}_m^{max}$ los límites inferior y superior del presupuesto asignado a la región *m*.

Las variables a_i, b_i son valores que representan el área y la región a las que pertenece el proyecto *i*. $x = (x_1, x_2, \dots, x_n)$ es un vector binario que especifica los proyectos incluidos en un determinado portafolio. Si $x_i = 1$, el proyecto *i* se encuentra dentro del portafolio, en caso contrario, el proyecto i no está incluido en él. Con los elementos antes mencionados, es posible modelar el problema de la siguiente manera:

$$Max\ Z = (\widetilde{z_1}, \widetilde{z_2}, \dots, \widetilde{z_n}), \tag{5}$$

donde:

$$\widetilde{z_j} = \sum_{i=1}^{n} \tilde{b}_{ij} x_i\,, \qquad j = 1,2,\dots,O, \tag{6}$$

Sujeto a:

$$\sum_{i=1}^{n} \tilde{c}_i x_i \le \tilde{C}, \tag{7}$$

$$\tilde{A}_k^{min} \le \sum_{i=1,a_i=k}^{n} \tilde{c}_i x_i \le \tilde{A}_k^{max}, \qquad k = 1,2,\dots,K, \tag{8}$$

$$\tilde{R}_m^{min} \le \sum_{i=1,b_i=m}^{n} \tilde{c}_i x_i \le \tilde{R}_m^{max}, \qquad m = 1,2,\dots,M, \tag{9}$$

$$x_i \in \{0,1\}, i = 1,2,\dots,n. \tag{10}$$

En la formulación antes mencionada, todas las variables denotadas con tilde (~) representan números difusos triangulares.

La función objetivo maximiza las contribuciones a cada uno de los objetivos (5). El valor de los objetivos se calcula mediante la adición de la contribución que cada uno de los proyectos presentes en el portafolio representa para cada objetivo (6). La restricción de presupuesto (7) asegura que la suma de los costos de inversión de los proyectos seleccionados no exceda el presupuesto total.

De manera similar, las restricciones de área (8) y región (9) aseguran que la suma de los costos de los proyectos pertenecientes a determinada área y región se encuentren dentro de los límites de presupuesto establecidos. Las soluciones al problema serán vectores binarios de tamaño *n* que representarán los proyectos incluidos dentro del portafolio (10).

5. Algoritmo Tri-NSGA-II

En esta sección se describe el diseño del algoritmo principal propuesto en el presente trabajo: Tri-NSGA-II. Este algoritmo es una adaptación del algoritmo clásico NSGA-II [12] para trabajar con parámetros difusos triangulares. Al igual que en el algoritmo original, Tri-NSGA-II presenta un enfoque generacional para abordar el problema de selección de portafolio.

El algoritmo requiere como entrada una instancia del problema que contiene los valores difusos de cada uno de los parámetros mencionados en la sección anterior y su salida consiste en una aproximación al frente de Pareto de dicha instancia.

5.1. Representación de soluciones y evaluación de factibilidad

Como se mencionó anteriormente, las soluciones al problema de selección de portafolio pueden representarse mediante un vector binario, cuyos valores representan la inclusión o ausencia de cada uno de los proyectos en el portafolio en cuestión, por lo que el espacio de posibles soluciones para un problema de *n* proyectos es $S = \{0,1\}^n$.

En la Fig. 1 puede observarse un ejemplo de solución candidata para determinada instancia del problema. Una solución candidata deber ser evaluada para determinar un

vector de objetivos difusos y su factibilidad se debe verificar para comprobar que satisface las restricciones del modelo matemático del problema. Los parámetros difusos triangulares son utilizados en ambos procesos y, para compararlos, será necesario el cálculo de GMI de cada uno de ellos.

5.2. Operador de cruza uniforme

Es un operador clásico de cruza en el que se seleccionan dos elementos de la población para generar descendencia. En él, los genes de cada uno de las soluciones padre tienen la misma probabilidad de ser transferidos a su descendencia [13, 14].

5.3. Operador de mutación: *one bit flip*

Es un método clásico de mutación de soluciones con representación binaria. Consiste en seleccionar de manera aleatoria una posición de una solución y cambiar su valor binario (de 0 a 1 o viceversa).

5.4. Generación de la población inicial

La población inicial del algoritmo Tri-NSGA-II comprende un número previamente definido de portafolios, generados de manera aleatoria. Cada solución generada es evaluada para verificar su factibilidad. La población inicial deberá estar conformada únicamente por soluciones factibles.

5.5. Ordenamiento de la población

El ordenamiento de la población constituye una parte fundamental en el funcionamiento del algoritmo Tri-NSGA-II. Las soluciones que conforman la población son ordenadas en dos etapas: la etapa elitista, denominada Ordenamiento por No Dominancia (en inglés: *Non-dominated Sorting*) que asegura la permanencia de las mejores soluciones, y la etapa de diversificación, conocida como Ordenamiento por Distancia de Amontonamiento (o *Crowding Distance Sorting*, CD *Sorting* por sus siglas en inglés) que busca asegurar que las soluciones sean suficientemente distintas entre sí.

5.6. Ordenamiento por no dominancia

El ordenamiento por no dominancia es un método de ordenamiento que tiene como propósito preservar la calidad de los portafolios presentes en la población, su funcionamiento puede dividirse en dos etapas, la primera de ellas consiste en la generación de un grupo de soluciones, denominado “frente de soluciones no dominadas”. Para determinar las soluciones que integran dicho frente, los vectores de objetivos de cada una de las soluciones de la población serán comparados entre sí para verificar la relación de dominancia que existe entre ellas. Aquellas soluciones que no sean dominadas por ninguna otra solución formarán parte del frente.

La segunda etapa del proceso de ordenamiento consiste en la generación de frentes subsecuentes con el conjunto de portafolios restante. Para ello, cada uno de los

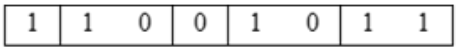


Fig. 1. Ejemplo de solución candidata.

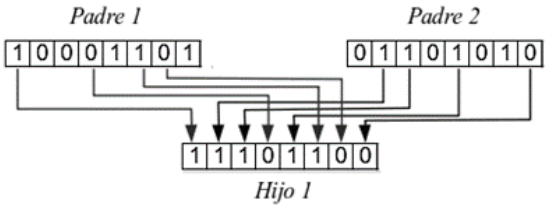


Fig. 2. Ejemplo de cruza uniforme.



Fig. 3. Ejemplo de mutación one bit flip.

portafolios presentes en el frente recién creado es explorado y el contador de dominancia de cada uno de los portafolios dominados por ellos se decrementa en 1, en caso de que el contador de alguno de ellos llegue a 0, será agregado al nuevo frente.

5.7. Ordenamiento por distancia de amontonamiento

La Distancia de Amontonamiento (*Crowding Distance*, CD por sus siglas en inglés) de una solución es un valor estimado de la densidad de soluciones que la rodean [12]. Este tipo de ordenamiento consiste en explorar cada uno de los frentes generados en la etapa anterior, calculando el valor de la métrica CD para cada uno de los portafolios contenidos en ellos.

Posteriormente, los portafolios de cada frente serán reorganizados de acuerdo a sus respectivos valores de CD. Para calcular el valor de la métrica es necesario ordenar las soluciones de cada frente por sus valores objetivo, posteriormente, se deberá asignar ∞ como valor de CD de la mayor y menor solución respectivamente. El valor de CD del resto de las soluciones puede calcularse mediante la fórmula:

$$d_{Ij}m = d_{Ij}m + \frac{f_{m^{j+1}}^{I^m} - f_{m^{j-1}}^{I^m}}{f_m^{max} - f_m^{min}}, \tag{11}$$

donde d es el valor de la métrica, I es la posición de la solución dentro de la población, j es la posición de la solución con respecto a las demás soluciones del frente f es el valor objetivo, m es el objetivo actual. El proceso se realiza con cada uno de los m objetivos, la sumatoria de dichos valores representará el valor de CD de dicha solución. Una vez calculada la métrica para cada uno de los portafolios, el frente deberá ser ordenado en forma decreciente.

6. Algoritmo Tri-NSGA-II-SSD

Tri-NSGA-II-SSD es una adaptación del algoritmo Tri-NSGA-II que incorpora la Desviación de Dispersión Espacial (*Spatial Spread Deviation*, SSD por sus siglas en inglés) como estimador de densidad. En la presente sección se describe el funcionamiento de dicho estimador dentro del algoritmo.

6.1. Desviación de dispersión espacial (SSD)

La Desviación de Dispersión Espacial es un estimador de densidad cuyo propósito de incorporación a Tri-NSGA-II es que las soluciones de los frentes no se encuentren separadas entre sí por márgenes muy amplios [15]. De manera similar al ordenamiento por CD, los valores de los objetivos de las soluciones de cada frente son normalizadas para posteriormente calcular su respectivo valor de SSD con las fórmulas siguientes:

$$temp1(i) = \frac{1}{n-1}\sqrt{\sum_{j=1}^{n}\left(D(i,j) - (D_{max} - D_{min})\right)^2} \quad \forall \quad i \neq j, \tag{12}$$

$$temp2(i) = \sum_{j \in K} \frac{D_{max} - D_{min}}{D(i,j)}. \tag{13}$$

La Fig. 4 muestra ambas etapas del algoritmo Tri-NSGA-II.

$$SSD(i) = SSD_0(i) + temp1(i) + temp2(i), \tag{14}$$

donde *D(i,j)* es la distancia de la solución *i* a la solución *j*. D_{max} y D_{min} son la mayor y menor distancia entre las soluciones, respectivamente. *K* es el número de *k* vecinos más cercanos a la solución *i* y SSD_0 es el valor inicial de SSD de la solución *i*, el cual, en caso de ser uno de los valores extremos del frente de objetivos normalizados deberá tomar el valor de $-\infty$.

7. Resultados experimentales

Con el propósito de evaluar el funcionamiento y desempeño de los algoritmos propuestos en el presente trabajo se realizaron dos experimentos, los cuales se describen en la presente sección.

7.1. Descripción de instancias

La experimentación se realizó usando un conjunto de instancias del problema de selección de portafolio previamente procesadas con el propósito de convertir sus parámetros a números difusos triangulares. Dichas instancias contienen la siguiente información:

- Presupuesto total (número triangular).
- Número de objetivos.
- Número de áreas.
- Límites de presupuesto para cada área (uno por línea) (números triangulares).

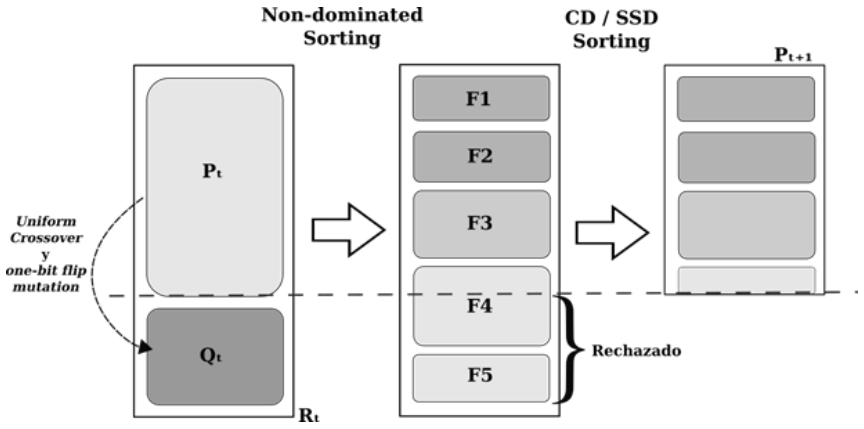


Fig. 4. Diagrama general de Tri-NSGA-II.

- Número de regiones.
- Límites de presupuesto para cada región (uno por línea) (números triangulares).
- Número de proyectos.
- Descripción de proyectos (uno por línea), incluye: costo (número triangular), área, región, aportación a objetivos (números triangulares).

Las instancias están identificadas con un nombre en el formato o*n*p*m*_*id*Tri, donde *n* representa al número de objetivos y *m* al número de proyectos e *id* es un número que identifica a la instancia.

7.2. Condiciones de experimentación

Los experimentos descritos en el presente trabajo fueron realizados en un equipo con procesador AMD A8-6410 *Quad core* a 2.4GHz, 6GB RAM 1600MHz. Los parámetros de configuración de los algoritmos se describen a continuación:

- Tamaño de la población: 50.
- Número de iteraciones: 100.
- Porcentaje de población a cruzar 20%.
- Porcentaje de población a mutar 50%.

7.3. Experimento 1: validación de funcionamiento

El propósito del primero de los experimentos realizados es validar el correcto funcionamiento de los algoritmos implementados. El experimento consistió en resolver mediante ambos algoritmos la instancia biobjetivo *o2p20_0Tri* (de la cual se conoce el frente óptimo) y comparar los frentes aproximados obtenidos por los algoritmos Tri-NSGA-II y Tri-NSGA-II-SSD.

En la Fig. 5, es posible observar que los frentes aproximados producidos por ambas variantes del algoritmo son relativamente cercanos al frente óptimo. De

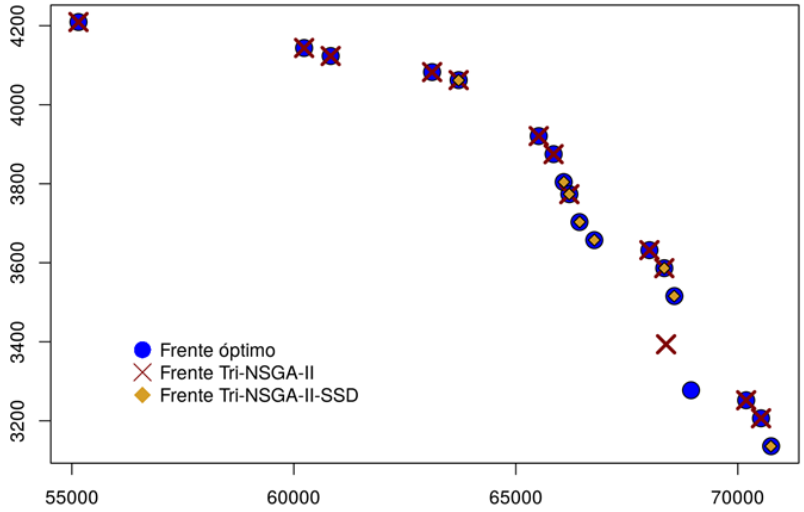


Fig. 5. Comparación de frentes para *o2p20_Tri* (valores normalizados con GMI).

igual manera, es posible observar que el frente de la variante con SSD incluye soluciones no contempladas en el de la variante con CD.

7.4. Experimento 2: evaluación de desempeño

Con el propósito de evaluar el desempeño de los algoritmos propuestos, se realizó un experimento que consistió en la resolución de un conjunto de 10 instancias de distintos tamaños, de 2, 4 y 9 objetivos respectivamente. Cada uno de los algoritmos fue ejecutado un total de 30 veces por instancia. Para cada una de las instancias se generó un conjunto de referencia, conformado por el conjunto de soluciones no dominadas de la combinación de los 30 frentes obtenidos.

Dicho frente de referencia fue utilizado como aproximación al frente óptimo de cada instancia para el cálculo de las métricas de hipervolúmen (*HyperVolume*) y dispersión generalizada (*Generalized Spread*). Posteriormente, se calcularon la mediana (M_e) y el rango intercuartílico (*IQR*) de los valores de dichas métricas, debido a que el rango intercuartílico y la mediana son menos sensibles a valores extremos en comparación con la media y la desviación estándar.

Con el propósito de validar los resultados obtenidos, se les aplicó una prueba de Wilcoxon con un nivel de confianza de 95%. En las tablas 5 y 6 pueden observarse los resultados de las métricas por cada instancia. Aquellos resultados cuyo resultado es significativo (de acuerdo a la prueba) se encuentran resaltados.

En la Tabla 5 puede observarse que Tri-NSGA-II es mejor con diferencia significativa en la resolución de las 10 instancias de prueba para la métrica de *HyperVolume*. Por otro lado, en la Tabla 6 es posible observar que Tri-NSGA-II-SSD tiene mejor desempeño en 7 de 10 instancias, en todas ellas con diferencia significativa. De igual manera, se realizó la prueba de Friedman con los valores obtenidos para ambas métricas.

Tabla 1. Resultados de evaluación con métrica *HyperVolume*.

HyperVolume				
	Tri-NSGA-II		**Tri-NSGA-II-SSD**	
Instancia	M_e	**IQR**	M_e	**IQR**
o2p20_Tri	0.238811441	0.044851738	0.163710911	0.069198901
o4p25_0Tri	0.264973213	0.05253274	0.20611812	0.096375359
o4p25_1Tri	0.209264293	0.079859039	0.146343058	0.100503483
o4p25_2Tri	0.259096812	0.066920563	0.209574394	0.121363497
o4p25_3Tri	0.172032511	0.03686516	0.161246173	0.083199129
o4p25_4Tri	0.26040734	0.097399762	0.213590777	0.122254636
o4p25_5Tri	0.227411752	0.064154667	0.180578485	0.087828932
o9p100_0Tri	0.072317856	0.078978551	0.059251828	0.066776339
o9p100_1Tri	0.081862723	0.079925586	0.051945505	0.041187613
o9p100_2Tri	0.124196526	0.092999414	0.084485714	0.0633435

Tabla 2. Resultados de evaluación con métrica *Generalized Spread*.

Generalized Spread				
	Tri-NSGA-II		**Tri-NSGA-II-SSD**	
Instancia	M_e	**IQR**	M_e	**IQR**
o2p20_Tri	0.95787788	0.45810173	0.71617801	0.1342707
o4p25_0Tri	0.41706173	0.06932411	0.34831437	0.06777055
o4p25_1Tri	0.403559	0.10486043	0.32747673	0.07611996
o4p25_2Tri	0.35834772	0.08948966	0.31741646	0.06983148
o4p25_3Tri	0.43841931	0.09865111	0.33124594	0.08006976
o4p25_4Tri	0.41285987	0.08332076	0.30090141	0.05669353
o4p25_5Tri	0.33274883	0.05214539	0.30352366	0.04080783
o9p100_0Tri	0.19171175	0.03487414	0.19084287	0.0275921
o9p100_1Tri	0.18635716	0.04270058	0.19014306	0.03099422
o9p100_2Tri	0.18869284	0.03584193	0.19065208	0.03091227

Tabla 3. Rankings de Friedman para ambos algoritmos con las métricas *HyperVolume* y *Generalized Spread.*

HyperVolume **(p-value: $4.43753\mathrm{E}^{-11}$)**		***Generalized Spread (p-value:*** $6.84371\mathrm{E}^{-11}$**)**	
Algoritmo	**Ranking**	**Algoritmo**	**Ranking**
Tri-NSGA-II	1.3066666666666682	Tri-NSGA-II	1.7366666666666628
Tri-NSGA-II-SSD	1.693333333333331	Tri-NSGA-II-SSD	1.2633333333333354

8. Conclusiones y trabajos futuros

En este trabajo se abordó el problema de optimización de la selección de portafolio considerando parámetros difusos triangulares. En la literatura revisada no existen antecedentes para abordar dicha variante del problema. Se propuso un nuevo modelo matemático y dos algoritmos basados en la metodología clásica del algoritmo NSGA-II, denominados Tri-NSGA-II y Tri-NSGA-II-SSD. Se realizaron pruebas experimentales sobre un conjunto de instancias de distintos tamaños y se les aplicaron dos pruebas no paramétricas (Wilcoxon y Friedman) como mecanismo de validación.

El análisis del desempeño de los algoritmos permitió observar que para la métrica de hipervolúmen, el algoritmo Tri-NSGA-II logró mejores resultados en las 10 instancias de prueba con diferencia significativa y que, por otro lado, el algoritmo Tri-NSGA-II-SSD obtuvo mejores resultados para la métrica de dispersión generalizada en 7 de 10 instancias con diferencia significativa.

Ambos resultados fueron validados mediante la aplicación de la prueba de Friedman, en donde puede observarse que para la métrica de hipervolúmen Tri-NSGA-II tiene menor *ranking* y que para la métrica de dispersión generalizada, la variante con SSD tiene un *ranking* menor.

Como trabajo futuro se contempla la implementación de variantes de estado estable de los algoritmos propuestos, así como realizar la evaluación de su desempeño considerando calidad y eficiencia.

Referencias

1. Perez, F., Gomez. T.: Multiobjective project portfolio selection with fuzzy constraints. Annals of Operations Research, vol. 245, no. 1-2, pp. 7–29 (2016) doi:10.1007/s10479-014-1556-z
2. Medaglia, A., L.: A Multiobjective evolutionary approach for linearly constrained project selection under uncertainty. European Journal of Operational Research, vol. 179, no. 3, pp. 869–894 (2007) doi: 10.1016/j.ejor.2005.03.068
3. Hua, K., Liu, B.: Project scheduling problem with mixed uncertainty of randomness and fuzziness. European Journal of Operational Research, vol. 183, no. 1, pp. 135–47 (2007) doi: 10.1016/j.ejor.2006.09.055
4. Kuchta, D.: Use of fuzzy numbers in project risk criticality assessment. International Journal of Project Management, vol. 19, no. 5, pp. 305–310 (2001) doi: 10.1016/S0263-7863 (00)00022-3
5. Balderas-Jaramillo F. A.: Modelando la impresición del problema de cartera de proyectos con filosofía gris (tesis para obtener el grado de doctor en ciencias de la computación). Instituto Tecnológico de Tijuana (2018)
6. Zadeh, L. A.: Fuzzy sets. Information and control, vol. 8, no. 3, pp. 338–353 (1965) doi:10. 1016/S0019-9958(65)90241-X
7. Vahidi, J., Rezvani S.: Arithmetic operations on trapezoidal fuzzy numbers. Journal of Nonlinear Analysis and Application, pp. 1–8 (2013) doi:10.5899/2013/jnaa-00111
8. Kumar, V.: Fuzzy genetic algorithms & fuzzy zdt test functions (tesis para obtener el grado de maestro en ciencias en matemáticas y computación). Indian Institute of Technology, Kharahpur (2010)
9. Chen, H. S., Hsieh, C. H.: Representation, ranking, distance and similarity of l-r type fuzzy number and application. Australian Journal of Intelligent Processing Systems, vol. 6, no. 4 (2000)
10. Deb, K.: Multi-objective optimization using evolutionary algorithms. 1st ed, John Wiley & Sons, pp. 3–34 (2001) doi: 10.1007/978-0-85729-652-8_1
11. Yao, S.: A multi-objective dynamic scheduling approach using multiple attribute decision making in semiconductor manufacturing. International Journal of Production Economics, vol. 130, no. 1, pp. 125–33 (2011) doi: 10.1016/j.ijpe.2010.12.014
12. Deb, K.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation. vol. 6, no. 2, pp. 182–197 (2002) doi:10.1109/4235.996017

13. Ackley, D. H.: A connectionist machine for genetic hillclimbing. Springer, US (1987)
14. Gilbert, S.: Uniform crossover in genetic algorithms. In: Proceedings of the 3rd In: Conference on Genetic Algorithms, US (1989)
15. Santiago, A. Dorronsoro, B.: A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME. Information Sciences, vol. 471, pp. 233–251 (2019) doi: 10.1016/j.ins.2018.09.005

ISSN 1870-4069

Tratamiento de señales electroencefalográficas de imaginación motora con fines de clasificación

Alexis Edmundo Gallegos Acosta, María Dolores Torres Soto, Aurora Torres Soto, Eunice Esther Ponce de León Sentí

Benemérita Universidad Autónoma de Aguascalientes, México

alexisedm@gmail.com, {mdtorres, atorres, eponce}@correo.uaa.mx

Resumen. En este artículo se presenta un análisis de la influencia del tratamiento de señales electroencefalográficas en el desempeño de un clasificador de máquina de soporte vectorial (SVM), cuyo objetivo fue la distinción de señales de imaginación motora ("abrir" y "cerrar" la mano derecha). El tratamiento consistió en el cálculo de la transformada rápida de Fourier (FFT) y la normalización Z-score. Esto debido a que las señales crudas poseen la suma de toda la actividad cerebral que incluye actividades fisiológicas como el movimiento ocular, cardiaca, la respiración, entre otros. Así, el objetivo de la selección de una técnica de tratamiento de datos permite la extracción de características que facilitan su posterior clasificación o evaluación. Las señales con imaginación motora que conforman el conjunto de datos utilizado para este artículo se obtuvieron por medio del dispositivo BCI Emotiv EPOC+ de 14 canales. Como resultado se presenta que el conjunto de datos transformado por FFT y normalizado obtiene mejor exactitud y precisión, a la vez que mantiene un buen equilibro en la distinción de las clases mencionadas.

Palabras clave: Señales electroencefalográficas, aprendizaje automático, transformada de Fourier, imaginación motora.

Electroencephalographic Signal Processing of Motor Imagery for Classification Purposes

Abstract. This paper presents an analysis of the influence of the electroencephalographic signal processing on the performance of a support vector machine (SVM) classifier, whose objective was the distinction of motor imagery signals ("opening" and "closing" the right hand). The treatment consisted of fast Fourier transform (FFT) computation and normalization Z-score. This is because the raw signals have the sum of all the brain activity that includes physiological activities such as eye movement, cardiac, breathing, among others. In this way, the purpose of the selection of a data processing technique allows the extraction of characteristics that facilitate their subsequent classification or evaluation. The motor imagery signals that make up the data set used for this article were obtained by means of the 14-channel BCI Emotiv EPOC+ device. As a result, the FFT-transformed and normalized data set obtains better accuracy and precision, while maintaining a good balance in the distinction of the mentioned classes.

Keywords: Electroencephalographic signals, machine learning, Fourier transform, normalization.

1. Introducción

El electroencefalograma (EEG) es un dispositivo desarrollado con el objetivo de registrar la actividad eléctrica del cerebro y que, en la actualidad, es utilizado de forma rutinaria para el diagnóstico de patologías y trastornos asociados al cerebro [1, 2]. La señal capturada por un dispositivo EEG es compleja debido a que se constituye de la suma de la actividad de diferentes poblaciones neuronales incluyendo actividades fisiológicas como el movimiento ocular, la respiración, la actividad cardiaca, entre otras [2, 3].

Las señales EEG poseen distintas propiedades como forma, amplitud, contenido frecuencial, etc., y de ellas se puede extraer información útil sobre el estado mental asociado a actividades motoras, sensoriales, estadías del sueño o diferentes enfermedades mentales [4].

En medicina el análisis de patrones en señales EEG se realiza por medio de un proceso de reconocimiento visual que requiere de entrenamiento para distinguir patrones y su asociación con diferentes enfermedades convirtiéndose así, en un proceso subjetivo [4]. Como consecuencia, se han desarrollado nuevas tecnologías y técnicas de procesamiento matemático que buscan mejorar la interpretación del comportamiento de las señales y que, a su vez, faciliten su análisis y sistematización [2, 4, 5].

A la fecha existen diferentes trabajos en la literatura que tienen el objetivo de proponer modelos que permitan el análisis de señales EEG con diferentes propósitos como la detección de enfermedades, detección de movimientos, detección de reacciones de acuerdo con distintos estímulos, control de dispositivos, etc. Por ejemplo, en [6] se muestra la planeación de una interfaz cerebro-computadora BCI que permite la detección de emociones.

Por su parte, [5] busca la detección de movimientos oculares, musculares y cognitivos. Mientras que [4] estudia diferentes técnicas de procesamiento de señales para la detección de ritmos cerebrales y crisis epilépticas. Además, existen proyectos que buscan convertir las señales EEG, en conjunto con sistemas BCI, en canales alternativos de comunicación y control como lo realizado en [7], identificando señales de imaginación motora; es decir, el análisis de actividades neuronales asociadas a movimiento mental sin necesidad de la intervención física de los nervios y músculos del cuerpo.

El presente artículo tiene como objetivo analizar la influencia de tratamiento de señales EEG con imaginación motora (“abrir” y “cerrar” la mano derecha), en el desempeño de un modelo de máquina de soporte vectorial (SVM); es decir, un clasificador que tiene el propósito de distinguir señales EEG con la intención de abrir la mano de las señales con la intención de cerrar la mano derecha.

Dicho tratamiento de datos se realizó por medio del cálculo de la transformada rápida de Fourier (FFT), una de las herramientas de extracción de características más populares en el análisis de series temporales [5, 8].

Como siguiente paso, se hace uso de la normalización Z-score, que permite a los algoritmos de aprendizaje automático encontrar tendencias evitando la existencia de diferencias drásticas entre las características analizadas. De manera que los datos tomen una misma escala y sean igualmente importantes al momento de realizar una clasificación mediante un algoritmo computacional.

Como resultado se tiene que, como se menciona en [9], la normalización de los datos es imprescindible para llevar a cabo la clasificación de objetos. Además, de que en conjunto con el cálculo de la transformada de Fourier permiten obtener incluso, mejores resultados en el desempeño de un clasificador como las máquinas de soporte vectorial.

A continuación, se abordan los conceptos importantes para este trabajo como la imaginación motora, las señales electroencefalográficas, la transformada de Fourier y el aprendizaje automático. Después, se procede a exponer los materiales y métodos seguidos para lograr el objetivo de artículo para, finalmente, presentar los resultados obtenidos, así como las conclusiones a las que se llegaron.

2. Conceptos relacionados

2.1. Imaginación motora

La imaginación motora (motor imagery), intención de movimiento o práctica mental del movimiento, ha sido fundamental en avances científicos bajo el contexto de rendimiento deportivo y rehabilitación [10, 11]. Se trata de un proceso cognitivo-perceptivo que consiste en la realización mental de movimientos que no van acompañados de ningún tipo de actividad muscular o periférica [12].

De acuerdo con Mokienko et al en [12], se distinguen dos tipos de imaginación motora: las formas visuales y las formas cinestésicas de imágenes motrices. En la forma visual, el sujeto produce una imagen de su movimiento simulando verse a sí mismo en tercera persona; mientras que, en la segunda forma, el sujeto genera la sensación cinestésica del movimiento.

La práctica de imaginación motora permite el desarrollo motor y aprendizaje de habilidades motrices evidenciando incluso, ganancia de fuerza en grupos musculares específicos. Desde el punto de vista médico, ha tenido resultados positivos en la rehabilitación en pacientes con condiciones neuromusculares.

Por ejemplo, pacientes con accidentes cerebrovasculares en los que la imaginación motora funciona como una "puerta trasera" al sistema motor y a la rehabilitación en todas las etapas de ictus al mantener el uso de impulsos voluntarios [13], además de que se ha demostrado que la imaginación motora activa zonas del cerebro que también están implicadas en la preparación y ejecución del movimiento físico [12].

Las ciencias computacionales también han estudiado la imaginación motora para desarrollo de interfaces cerebro-computadora (BCI) las cuales, han ganado una amplia atención en los últimos años. Los sistemas BCI basados en imaginación motora más importantes, son aquellos que involucran aplicaciones en tiempo real como la escritura sin contacto, brazos protésicos, sistemas de realidad virtual, control de dispositivos, etc. En este tipo de sistemas la clasificación de los datos de imaginación motora es el núcleo de este tipo de sistemas, dando como resultado el diseño de diferentes marcos de trabajo para mejorar el procesamiento de este tipo de datos [7, 14].

2.2. Señales electroencefalográficas

La electroencefalografía es una técnica que permite el registro de la actividad cerebral (señales EEG). "Las señales registradas corresponden a una mezcla de

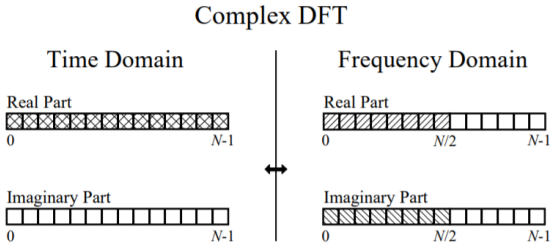


Fig. 1. Transformada discreta de Fourier [17].

información cerebral, así como artefactos fisiológicos (movimientos oculares, musculares, latido cardiaco, sudor) o técnicos (líneas de alimentación, desconexión de electrodos)" [15].

La captura de señales EEG es una técnica no invasiva que, en medicina es utilizada para la búsqueda de patrones anormales que identifiquen alguna patología. Esto se debe a que la actividad cerebral forma un patrón reconocible [16].

Para la elaboración de este artículo, se utilizó el dispositivo EMOTIV Epoc+ con tasa de muestreo de 128Hz por medio de 14 canales (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) distribuidos en el cuero cabelludo bajo el estándar 10-20, el cual es un método internacional que describe la ubicación de los electrodos de un dispositivo de electroencefalograma [17].

2.3. Transformada rápida de Fourier (FFT)

La transformada de Fourier (FT) es una de las herramientas más poderosas en el análisis de señales y crucial en muchas aplicaciones como la computación científica y la ciencia de datos [18]. El análisis de Fourier permite expresar una función como la suma de sus componentes periódicas, es decir, descompone la función en una serie de funciones trigonométricas más simples [18], [19], que, al sumarse tendrían la forma exacta de la señal original [20].

Existe la posibilidad de analizar señales cuya representación es exacta por medio de una suma finita de N senoides. Por el contrario, existen otras con las cuales solo es posible realizar una suma parcial para obtener una aproximación adecuada cuyo error tiende a cero al considerar más términos o frecuencias [20]. Es decir, dada una señal periódica con periodo *T,* su aproximación está dada por la suma parcial de la serie de Fourier de *f(t),* como se muestra la ecuación (1) [20]:

$$f(t) \cong a_0 + \sum_{n=1}^{N} a_n \cos(n\omega_0 t) + \sum_{n=1}^{N} b_n \sin(n\omega_0 t) \quad (1)$$

,

donde a_0, a_n y b_n son los coeficientes trigonométricos de Fourier, mientras que ω_0 es la frecuencia.

El cálculo de la transformada de Fourier con gran cantidad de muestreos se vuelve complejo dada la gran cantidad de operaciones a realizar provocando que el tiempo requerido también incremente [21]. Para ello se puede aplicar el algoritmo de la transformada rápida de Fourier (FFT) o transformada discreta de Fourier (DFT).

Sin embargo, cabe aclarar que tanto la FFT como la DFT son utilizadas en la literatura de forma indiferente, a pesar de ello, el FFT es un algoritmo que calcula la transformada discreta de Fourier [18].

La transformada discreta de Fourier realiza el análisis espectral de una señal, trasladándose del espacio de tiempo discreto al espacio de frecuencias donde es posible obtener información de las componentes frecuenciales predominantes de una señal [22].

La DFT transforma la señal del dominio del tiempo de N puntos en dos señales nuevas (parte real e imaginaria) en el dominio de la frecuencia de N puntos [17] (ver Fig. 1). Este nuevo dominio de la señal descompone la señal en su correspondiente sinusoidal de diferentes frecuencias [23].

La transformada rápida de Fourier (FFT) para una señal discreta x_n, donde k es cada valor discreto de la señal, está dada por la ecuación (2) [23]:

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n] \quad con\ k = 0, 1, \dots, N-1 \tag{2}$$

La transformada de Fourier presenta características importantes como la linealidad, esto es, posee la propiedad distributiva respecto de la suma. Por otra parte, se trata de una función simétrica conjugada lo que, beneficia el cálculo de la magnitud de los puntos en un periodo completo, pues solo es necesario calcular los *N/2+1* primeros puntos. Finalmente, si la señal sufre una rotación en un ángulo determinado, su transformada también se verá afectada por una rotación del mismo ángulo [24].

2.4. Aprendizaje automático

El aprendizaje automático o machine learning (ML) es una rama de la inteligencia artificial encargada del desarrollo de técnicas que permiten a las computadoras aprender [25]. Es decir, es un área que dedica su estudio a la construcción de programas computacionales que tengan la habilidad de mejorar su desempeño en alguna tarea específica gracias a la experiencia [26, 27].

Un programa informático aprende de un conjunto de tareas, si su medida de desempeño mejora con la experiencia [27]. Es decir, se dice que un sistema aprende de forma automática si, para resolver un problema, es capaz de tomar decisiones con base en la experiencia acumulada, incluso se busca que estos sistemas sean capaces de trabajar con una gran cantidad de datos de entrada, los cuales pueden contener ruido e inconsistencias [28].

Los algoritmos de ML son capaces de encontrar patrones a partir de datos previamente suministrados, lo que permite la creación de modelos de predicción y toma de decisiones [29].

Así, la calidad y la cantidad de la información con la que se entrena este tipo de algoritmos es crucial, pues se trata de métodos matemáticos sofisticados y métodos de búsqueda en grandes escalas que requieren de estructuras de representación del conocimiento adecuada, para agilizar la identificación de hechos relevantes [29, 30].

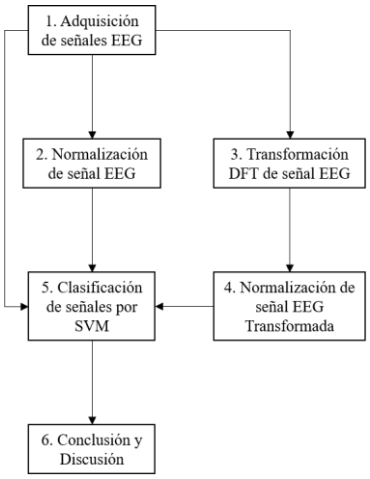


Fig. 2. Metodología.

Nilsson, en su libro *Introduction to Machine Learning* [31], enlista los requerimientos que conforman un sistema de aprendizaje automático [26, 31]:

- Selección del tipo de entrenamiento.
- Definición de la función objetivo a aprender.
- Definición de la representación de la función objetivo.
- Selección del algoritmo de aprendizaje.

En la literatura se pueden encontrar diversos algoritmos de aprendizaje automático como: los árboles de decisión, las máquinas de soporte vectorial, máxima entropía, naive Bayes, etc. [26]. Para este trabajo se implementó un modelo de máquina de soporte vectorial que permite modelar señales EEG con imaginación motora de abrir y cerrar la mano derecha.

3. Materiales y métodos

En la Fig. 2 se muestra la metodología que se siguió para la elaboración de este artículo. Este procedimiento tiene el objetivo de analizar el desempeño de un modelo de máquina de soporte vectorial (SVM) al entrenarse con cuatro bases de datos. La primera de ellas es la base de datos crudos de señales EEG con imaginación motora; la segunda, es el mismo conjunto de datos, pero tratados por medio de una normalización Z-score.

Por su parte, el tercer conjunto de datos corresponde a la señal cruda procesada por medio de la transformada rápida de Fourier (FFT) y la última corresponde a una base de datos normalizada por medio de la normalización Z-score de los datos transformados anteriormente por la FFT.

La implementación del algoritmo de máquina de soporte vectorial, así como la implementación y representación de la transformada rápida de Fourier se realizaron con las librerías disponibles de Python.

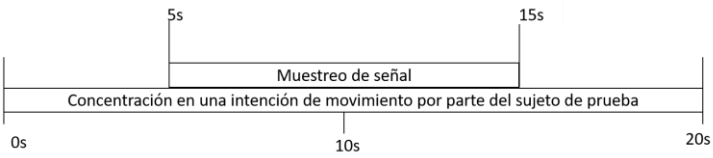


Fig. 3. Protocolo de captura de señales EEG con imaginación motora.

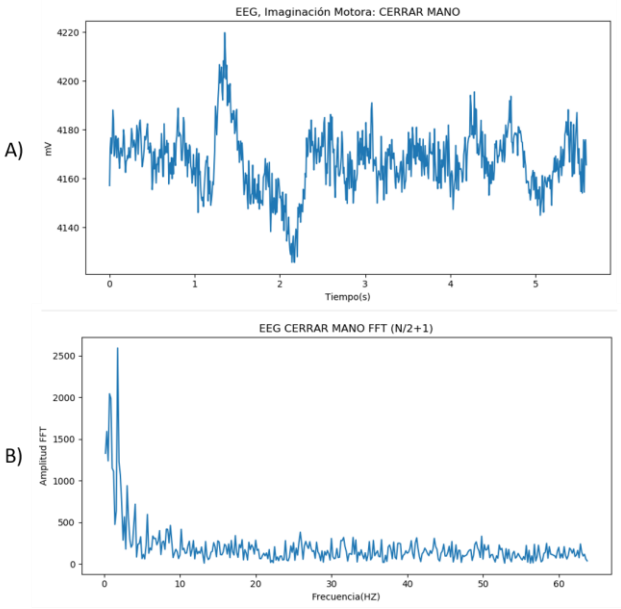


Fig. 4. (A) Señal en crudo. (B) Señal transformada FFT.

El primer paso de metodología consta de la adquisición de las señales EEG en crudo. Este paso de la metodología se realizó empleando el dispositivo de EEG Emotiv EPOC+ de 14 canales (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) con una tasa de muestreo de 128 muestras por segundo. Además, se contó con el software de captura Cykit [32], de acceso libre, que permitió el registro de las señales de imaginación motora correspondientes a abrir y cerrar la mano derecha en archivos csv para su tratamiento posterior.

Para realizar el muestreo de las señales EEG con imaginación motora, se siguió un protocolo de captura basado en el presentado por Paredes et al en [17]. Como se muestra en la Fig.3, la captura consta de 20 segundos en los que el sujeto de prueba se concentra en una de las intenciones solicitadas (abrir o cerrar la mano derecha). La captura de la señal inicia cinco segundos después de que el sujeto de prueba inició su concentración y se detiene cinco segundos antes de finalizar la prueba.

De esta manera es posible tener mayor seguridad de que se captura la intención o imaginación del movimiento de la mano. Una vez terminada la captura de imaginación motora, se cuenta con un archivo .csv con los registros de los 14 canales del dispositivo de Emotiv. Para este caso, la base de datos constó de 1,283 registros obtenidos a

conveniencia de un solo sujeto de prueba, de los cuales, 566 corresponden a la imaginación motora de abrir la mano derecha y 717 registros corresponden a cerrar la mano, teniendo dos conjuntos disjuntos de señales EEG.

El segundo paso fue trasladar una copia de la base de datos cruda a una escala común debido a la magnitud de los valores que toma cada vector de las 14 canales. De manera paralela, se aplicó una transformada discreta de Fourier como tercer paso, descrita en el apartado 2.3. En la Fig.4 se expone la gráfica de la señal en crudo del canal AF3 (gráfica A) correspondiente a la imaginación de movimiento de abrir la mano.

Por su parte, la gráfica B muestra la aplicación de la transformada rápida de Fourier para calcular la transformada discreta de Fourier. Como se puede observar se aprovecha la propiedad de simetría ya comentada en el apartado 2.3. Al igual que el conjunto de datos en crudo, el conjunto de datos tratado con la transformada rápida de Fourier fue trasladado a una escala común por medio de la normalización Z-score que corresponden al cuarto paso de la metodología de la Fig. 2.

En este punto, se cuenta con cuatro conjuntos de datos: las señales en crudo, las señales en crudo normalizadas, señales transformadas a Fourier y señales transformadas normalizadas. Dichos conjuntos de datos fueron utilizados para entrenar y validar un modelo de máquina de soporte vectorial (SVM) para así, analizar el desempeño de este trabajando con cada uno de los conjuntos de datos.

El desempeño del modelo se midió mediante las métricas de desempeño tradicionales: exactitud, precisión, sensibilidad, especificidad y puntuación F1 la cual, resume la precisión y la sensibilidad de los modelos en una sola métrica, siendo de mayor utilidad cuando existe un desbalance entre las clases [33].

La Tabla 1, muestra los parámetros utilizados para construir cada uno de los modelos SVM probados cada conjunto de datos. Los valores de Kernel, Gamma y C fueron definidos de acuerdo con los resultados observados en [34].

Cada modelo fue replicado 30 veces para analizar su comportamiento. En siguiente apartado se exponen los resultados obtenidos.

4. Resultados

Como ya se ha mencionado, se entrenaron y validaron los cuatro modelos de SVM. Cada modelo fue replicado 30 veces para valorar su desempeño. Así mismo, cada réplica seleccionó su propio conjunto de datos para entrenamiento y validación en porcentajes de 80% y 20% respectivamente. Como medidas de desempeño se utilizaron las métricas de exactitud, precisión, sensibilidad, especificidad y puntuación F1.

En la Fig. 5 se muestra el comportamiento de la exactitud que presentaron los modelos de SVM. Los datos en crudo y los datos tratados con transformada de Fourier presentan valores muy similares rondando entre el 40% y 65%.

En cambio, los conjuntos de datos normalizados presentan una mejor exactitud. Sin embargo, en este caso los datos transformados y normalizados son claramente superiores al mantenerse cerca del 100% durante el proceso de replicado. Esto quiere decir, que estos modelos pueden identificar de manera efectiva señales tanto de “abrir” como de “cerrar” la mano.

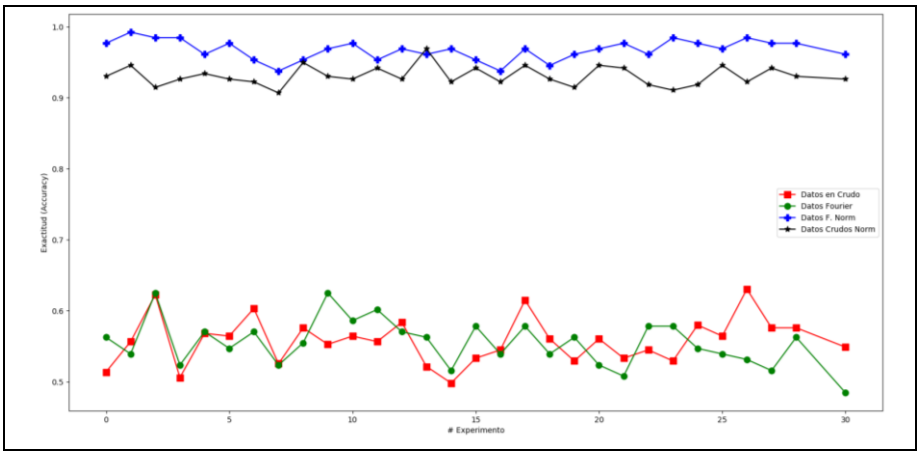


Fig. 5. Exactitud (Accuracy).

Tabla 1. Parámetros del modelo SVM.

Kernel	Valor Gamma	Valor C
Rbf	0.04	200

La métrica siguiente es la precisión que, en este caso mide la proporción de señales correspondientes a "abrir la mano" que fueron correctamente clasificadas. Como se puede observar en la Fig.6, el comportamiento es similar al de la Fig.5, mostrando que los datos normalizados y transformados normalizados presentan una mejor distinción de la imaginación motora de abrir la mano.

La Fig.7 muestra el desempeño de los modelos SVM según la métrica de sensibilidad que indica la capacidad del modelo para distinguir las señales de "abrir la mano" de las señales de "cerrar la mano". La gráfica de la Fig. 7 muestra que los modelos que trabajaron con datos crudos y datos transformados son perfectamente capaces de clasificar señales con la intención de abrir la mano, incluso, empatan con un 100% durante todo el proceso de replicado por lo que se aprecia una sola línea.

Al igual que en el caso de la exactitud, las réplicas del modelo SVM que trabajaron con el conjunto de datos transformados y normalizados tienen un mejor comportamiento que el modelo que trabajó con los datos crudos normalizados llegando incluso, a igualar a los modelos con datos crudos y transformados en algunas de las réplicas.

La especificidad mide la capacidad del modelo de distinguir las señales de "cerrar la mano" de las señales de "abrir la mano". A diferencia del desempeño obtenido en la métrica de sensibilidad, los modelos que trabajaron con datos crudos y normalizados son, prácticamente incapaces de reconocer una señal con la intención de cerrar la mano (ver Fig. 8). En cuanto a los modelos SVM que trabajaron con los conjuntos normalizados mantienen un comportamiento similar al caso anterior, lo que significa que existe un equilibrio entre su capacidad de distinguir una señal de la otra.

Finalmente, se analiza en la Fig. 9 la métrica de la puntuación F1. Aquí se puede observar que una vez más los modelos que trabajaron con conjuntos normalizados

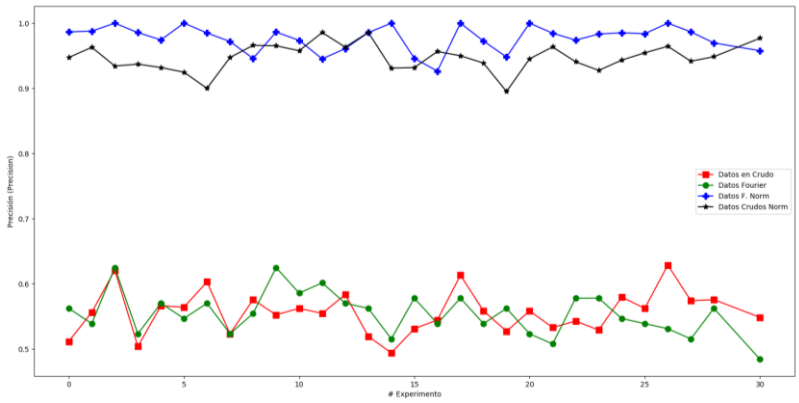


Fig. 6. Precisión (Precision).

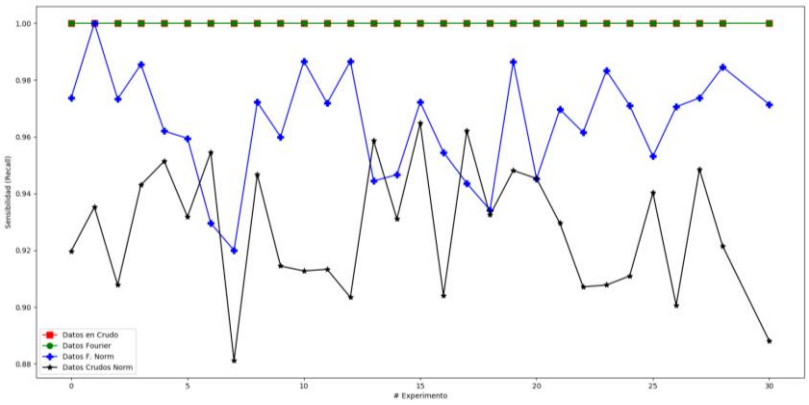


Fig. 7. Sensibilidad (Recall).

tienen un muy buen nivel de precisión y sensibilidad, siendo superior el modelo con datos transformados y normalizados.

5. Conclusiones y trabajo a futuro

Frente a lo expuesto en este artículo, se concluye que la calidad de los datos con los que se alimenta un algoritmo de aprendizaje automático es de alta importancia, pues permite agilizar su análisis y procesamiento de datos.

Es por ello, que elegir un buen algoritmo de extracción de características, como la transformada de Fourier, es clave para obtener mejores resultados en el desempeño de los algoritmos de clasificación. En este caso la FFT resulta conveniente pues asume que los datos son localmente estacionarios y los datos utilizados de las señales, aunque no lo son, corresponden a muestras obtenidas en un tiempo finito específico. Asimismo, la utilización de algún método de normalización como la Z-score es imprescindible pues evita la existencia de sesgo en la clasificación de objetos al trasladar cada una de las características a una escala común. De esta manera, como se muestra en el apartado de

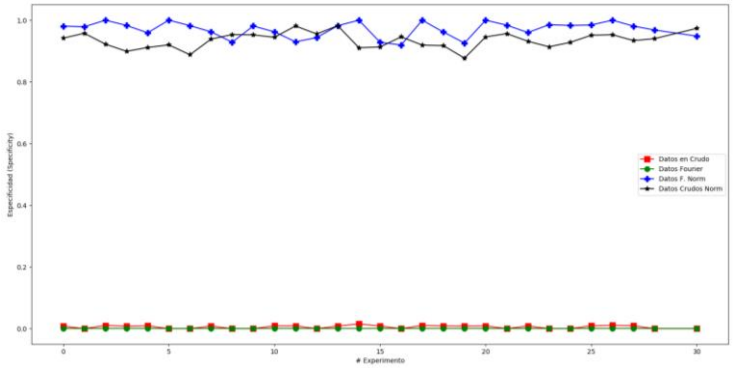


Fig. 8. Especificidad (Specificity).

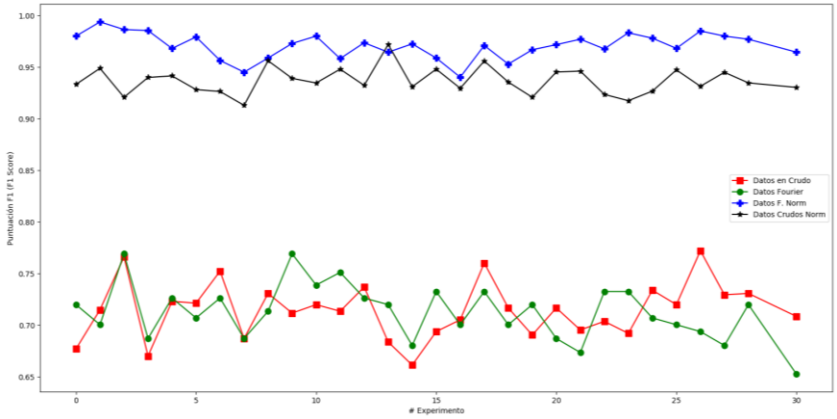


Fig. 9. Puntuación F1 (F1 Score).

resultados, ambas herramientas favorecen un alto desempeño por parte del modelo de máquina de soporte vectorial en cada una de las métricas de evaluación utilizadas.

Como trabajo a futuro se tiene la construcción de una base de datos mayor con el muestreo de una mayor cantidad de sujetos de prueba. Además de experimentar con más herramientas de extracción de características como la transformada Wavelet y dimensión fractal. Adicionalmente, se pretende probar su desempeño en algoritmos de aprendizaje automático como las redes neuronales artificiales (RNA), random forest (RF), naive Bayes, entre otros.

Referencias

1. Espinosa, L.: Electroencefalografía inalámbrica: Una mirada actual y propuesta de sistema portátil. Universidad Técnica Federico Santa María, Chile, 2016, http://profesores.elo.utfsm.cl/~agv/elo323/2s15/projects/reports/LeonardoEspinosa_WirelessEEG.pdf
2. Urrestarazu. E., Iriarte, J.: Mathematical analyses in the study of electroencephalographic

signals, Rev. Neurol., vol. 41, no. 5, pp. 423–434 (2005) https://dadun.unav.edu/handle/10171/22783

3. 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. Revista Mexicana de Ingeniería Biomédica, vol. 34, no. 1, pp. 23–39 (2013) doi: 10.35537/10915/32602
4. Bermúdez-Cicchino, A. N.: Técnicas de procesamiento de EEG para detección de eventos. Universidad Nacional de La Plata (2013) doi: 10.35537/10915/32602
5. Gómez-Figueroa, L. J.: Análisis de señales EEG para detección de eventos oculares, musculares y cognitivos. Universidad Politécnica de Madrid, España (2016)
6. Aparicio-García, R. S., Juárez-Gracia, G., Álvarez-Cedillo, J. A., Sandoval- Gutiérrez., J., Tovar-Corona, B.: Método de extracción y selección de características para la implementación de una interfaz cerebro-computadora en detección de emociones. Dyna New Technol, vol. 4, no. 1, pp, 14 (2017) doi: 10.6036/nt7829
7. Sreeja, S. R., Rabha, J., Nagarjuna, K. Y., Samanta, D., Mitra, P., Sarma, M.: Motor imagery EEG signal processing and classification using machine learning approach. In: Proceedings of International Conference on New Trends in Computing Sciences (2017) doi: 10.1109/ICTCS.2017.15
8. Martín-Sánchez, J. L.: Aportaciones al diseño de interfaces persona-máquina utilizando señales EEG. Universidad de Alcalá, Alcalá (2017)
9. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery DMKD´03 (2003) doi: 10.1145/882082.882086
10. 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., vol. 66, no. 05, (2018) doi: 10.33588/rn.6605.2017394
11. MacIntyre, T. E., Madan, C. R., Moran, A. P., Collet, C., Guillot, A.: Motor imagery, performance and motor rehabilitation. Progress in Brain Research, vol. 240, S. Marcora and M. Sarkar, Eds. Elsevier, pp. 141–159 (2018) doi: 10.1016/bs.pbr.2018.09.010
12. Mokienko, O. A., Chernikova, L. A., Frolov, A. A., Bobrov, P. D.: Motor imagery and its practical application, Neurosci. Behav. Physiol., vol. 44, no. 5 (2014) doi: 10.1007/s11055-014-9937-y
13. Dickstein, R., Deutsch, J. E.: Motor imagery in physical therapist practice, Phys. Ther., vol. 87, no. 7, pp. 942–953, (2007) doi: 10.2522/ptj.20060331
14. Kumar, S., Sharma, A., Mamun, K., Tsunoda, T.: A deep learning approach for motor imagery EEG signal classification. In: 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (2016) doi: 10.1109/APWC-on-CSE.2016.017
15. Santillan-Guzman, A., Ramírez-Uriarte, H. Z., Oliveros-Oliveros, J. J.: Morin Castillo, M. M., Ramírez-Díaz, H.: Interfaz gráfica intuitiva para el procesamiento de señales EEG. In: Memorias DEl Congr. Nac. Ing. Biomédica, vol. 5, no. 1, pp. 146–149 (2018) doi: 10.24254/CNIB.18.20
16. Tombesi, S.: Análisis de electroencefalograma con transformada de Fourier. (2012)
17. Zea-Paredes, D. A., Arévalo-Villacrés, J. M.:Sistema de detección de la intención de dos movimientos de la mano a través del procesamiento de señales EEG. Universidad de Cuenca, Cuenca, Ecuador (2020)
18. MacLeod, C.: Fourier transforms with scipy.fft python signal processing. Real Python (2020) https://realpython.com/python-scipy-fft/
19. Scipy, Fourier transforms (scipy.fft), *Scipy.org* (2021) https://docs.scipy.org/doc/scipy/reference/tutorial/fft.html.

20. Urquiza, N.: Aplicación de las series de Fourier a un circuito excitado por una señal periódica. Bahía Blanca, Argentina, (2011) http://lcr.uns.edu.ar/fvc/NotasDe Aplicacion/FVC-NicolásUrquiza.pdf.
21. Supervielle-Brouqués, F. G.: Digitalización del sonido. Bahía Blanca, Argentina (2011) http://lcr.uns.edu.ar/fvc/NotasDeAplicacion/FVC-Gaston Supervielle.pdf.
22. Revel-Luna, D.: Transformada discreta de fourier: Análisis de señales en tiempo real (2020)
23. 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, vol. 39, no. 22, pp. 36–48 (2018) https://www.revistaespacios.com/a18v39n22/18392236.html
24. Araujo, C. I.: Transformada discreta de Fourier propiedades, aplicaciones, ejemplos. lifeder.com (2019) https://www.lifeder.com/transformada-discreta-de-fourier/
25. Sancho-Caparrini, F.: Introducción al aprendizaje automático. Sevilla, España (2017). http://www.cs.us.es/~fsancho/?e=75
26. Godoy-Viera, Á. F.:Técnicas de aprendizaje de máquina utilizadas para la minería de texto. Investigación Bibliotecológica Archivonomía, Bibliotecología e Información, vol. 31, no. 71 (2017) doi: 10.22201/iibi.0187358xp.2017.71.57812
27. Mitchell, T. M.: Machine learning, 1st ed. McGraw-Hill/Engineering/Math (1997)
28. Moreno, A., Armengol, E., Béjar, J., Belanche, L., Cortés, U., Gavaldá, R., Gimeno, J. M., López, B., Martín, M., Sánchez, M.: Aprendizaje automático. Cataluña, España, Ediciones UPC (1994) chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://upcommons. upc.edu/bitstream/handle/2099.3/36157/9788483019962.pdf
29. Bastidos, M. X.: ¿Qué es el aprendizaje de máquina y en qué casos debería usarlo en su negocio? (2020) https://reconoserid.com/que-es-el-aprendizaje-de-maquina-y-en-que-casos-deberia-usarlo-en-su-negocio/
30. Nilsson, N. J.: Introduction to Machine Learning. Stanford University (1998)
31. Cymaticorps, Cykit. Cymaticorp, (2017) https://github.com/CymatiCorp/CyKit
32. Barrios-Arce, J. I.: La matriz de confusión y sus métricas, Health Big Data (2019) https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/ (2021)
33. Gallegos-Acosta, A. E., Torres-Soto, M. D., Torres-Soto, A., Ponce de León-Sentí, E. E.: Contrastación de algoritmos de aprendizaje automático para la clasificación de señales EEG. Res. Comput. Sci., vol. 148, no. 8, pp. 515–525 (2021) https://www.rcs.cic.ipn.mx/ 2020_149_8/Contrastacion de algoritmos de aprendizaje automatico para la clasificacion de senales EEG.pdf

ISSN 1870-4069

Identificación de potenciales “Hotspots” usando métodos de clasificación

Humberto Reyes San Pedro, Humberto Cervantes Maceda,
Abel García Nájera

Universidad Autónoma Metropolitana,
Posgrado Ciencias y Tecnologías de la Información,
México

{hreyes7, hcm}@xanum.uam.mx, agarcian@cua.uam.mx

Resumen. El desarrollo continuo del software es una realidad en la actualidad. Los cambios en el software pueden deberse al desarrollo de nuevas funcionalidades, la corrección de defectos o la reparación de algunas decisiones de diseño o de desarrollo, convenientes a corto plazo, pero que causan problemas más adelante. Esto último se conoce como pago de la deuda técnica del sistema. La gestión de la deuda técnica es una actividad fundamental en el desarrollo, y en este trabajo se propone utilizar modelos de software que permitan identificar de manera eficiente los archivos propensos al cambio y propensos al error, a los que llamamos “Hotspots” que son fuente de deuda técnica. Este estudio examina la efectividad de nuestro modelo basado en un algoritmo de clasificación, en métricas de software de cada archivo del sistema y en el manejo de los valores de las métricas, con el fin de encontrar una combinación de ellas que mejor nos ayuden a determinar Hotspots.

Palabras clave: Deuda técnica, hotspots, métricas de software, clasificadores.

Potential "Hotspots" Identification Using Classification Methods

Abstract. Continuous software development is a reality these days. Changes in software may be due to the development of new functionalities, the correction of defects or the repair of some design or development decisions, convenient in the short term, but that later causes problems. These decisions are known as the technical debt of the system. Its management is a fundamental activity in development, and in this work we propose the use of software models that support the identification of change prone and error prone files, which we call "Hotspots", which are sources of technical debt. This study examines the effectiveness of our model based on classification algorithms, on software metrics for each file in the system, and on managing the values of the metrics to find a combination of them that best helps us determine Hotspots.

Keywords: Technical debt, hotspots, software metrics, classifiers.

1. Introducción

En 1992 fue introducida la metáfora de Deuda Técnica (DT) desde un punto de vista financiero, para reconocer los posibles efectos negativos a largo plazo y de largo alcance que tienen ciertas decisiones que se toman durante el desarrollo [2]. La DT es generalizada, afecta a todos los aspectos de la ingeniería del software: desde el levantamiento de requerimientos, diseño arquitectónico elegido y hasta como es implementado. El término "deuda técnica" definido por Howard Cunningham en 1992, no es nuevo y tampoco los conceptos que cubre.

Durante 35 años los ingenieros de software lo han conocido con otros nombres: mantenimiento de software, evolución, envejecimiento, decadencia, reingeniería, sostenibilidad, entre otros. Pero el progreso ha sido poco sistemático, el tema no se considera atractivo y rara vez es enseñado en las escuelas hasta la fecha. La deuda técnica arquitectónica (DTA) se presenta al hacer decisiones de diseño que pueden afectar a los atributos de calidad tanto internos como externos, atentando en primera instancia a los atributos de mantenibilidad y de modificabilidad, con lo cual posteriormente se comprometerán las entregas dado que se tiene un código difícil de modificar.

Nuestro estudio se centra en anticiparse a estos problemas que se pueden presentar y prevenirlos con mantenimientos correctivos al software. La anticipación de la que hablamos se enfoca en detectar aquellos módulos del software (archivos) que sean propensos al cambio o propensos al error y que, dentro del desarrollo de este artículo, denominaremos como Hotspots (puntos calientes). Identificar estos Hotspots es de suma importancia para saber en dónde enfocar recursos para reducir la DT y tener un producto de software de mejor calidad y mucho más mantenible.

Algunos estudios en la literatura han abordado la evaluación de los archivos propensos al cambio o al error, varios de ellos utilizando algoritmos de aprendizaje automático (AA) y algunos otros de métodos estadísticos, como regresión logística. En este estudio se propone el uso de métodos cualitativos y la validación de los mismos, considerando métricas tanto internas como externas del código apoyados de algoritmos de clasificación, por su capacidad de aprender de los datos de prueba, para determinar qué combinación de métricas nos ayudan a detectar con mayor certeza los Hotspots.

Cada método se estará validando sobre once proyectos de código abierto desarrollados en lenguaje Java de diferentes tamaños. Además de proponer diferentes normalizaciones a las métricas para ajustar los valores medidos en diferentes escalas respecto a una escala común.

Las métricas son obtenidas de herramientas de análisis de código, disponibles en el mercado, como: Archinaut[1] (ARCH), Designite[2] (DES), Succinct Code Counter[3] (SCC), DV8[4] (DV8) y SonarQube[5] (SQ). Estas herramientas toman como entrada código fuente, código compilado e historial de las aplicaciones generado a través del uso de los sistemas de control de versiones como Git y de sistemas de control de

[1] https://dl.acm.org/doi/pdf/10.1145/3387906.3388633

[2] http://www.designite-tools.com/

[3] https://boyter.org/posts/sloc-cloc-code/

[4] https://www.archdia.net/products-and-services/

[5] https://www.sonarqube.org

incidentes los cuales ofrecen información muy valiosa que se traduce en métricas a utilizar.

De los proyectos que se seleccionaron, se hizo un corte en la versión de cada uno de ellos en la que se consideraron mil commits asociados a incidentes registrados, a lo que llamaremos "bug commit". Como resultado obtuvimos métricas estructurales e históricas, algunas de ellas repetidas entre las diferentes herramientas. De un total de treinta y seis métricas obtenidas, se seleccionaron para nuestro estudio dieciséis de ellas (entre estructurales e históricas) y una más para validación de resultados.

Se aplicaron métodos de clasificación exhaustivos y de AA para buscar en el espacio de soluciones la mejor combinación de métricas que nos clasifique los archivos Hotspot y No Hotspot. Los clasificadores AA se han aplicado con éxito en diferentes dominios de la ingeniería de software, tales como predicción de defectos o la predicción de la capacidad de mantenimiento [6], así que, los tomaremos en este estudio también. El reto es encontrar un conjunto reducido de métricas para detectar Hotspots en las aplicaciones de manera temprana.

Para abordar este reto se realizó el presente estudio empírico buscando dar respuesta a las siguientes preguntas de investigación: P1) ¿Existen combinaciones de métricas que puedan ofrecer mejores resultados en la detección de Hotspots en las aplicaciones de software? P2) ¿Las métricas para detectar Hotspots de una aplicación pueden funcionar para determinar Hotspots en otras aplicaciones? P3) ¿Es posible determinar qué algoritmos resulta mejor para la detección de Hotspots entre un método de clasificación y un método base (exhaustivo)?

El resto del artículo está organizado como sigue: Sección 2 resumen de trabajos previos relacionados. Sección 3 antecedentes de la investigación. Sección 4 explica la metodología de la investigación. Sección 5 presenta el análisis de resultados y se da respuesta a las preguntas de investigación. Sección 6 presenta las conclusiones del estudio y trabajos futuros sugeridos.

2. Trabajos relacionados

Se han desarrollado varios enfoques en la búsqueda de la detección de archivos propensos al cambio o al error en los sistemas utilizando combinaciones de métricas, como los siguientes: Hilton et al. [3] aplican un método que utiliza métricas internas OO y externas como tasa de defectos y tiempo que lleva ejecutar las funciones por archivo, solo que son obtenidas una vez que el sistema es liberado en producción.

Usando el clasificador de Bosques aleatorios y métricas mencionadas obtiene una predicción del 75% de los Hotspots, con un 75% de exactitud para aplicaciones diferentes a las que fue entrenado el algoritmo. Concluyen que entre más datos tenga el clasificador, mejorarán la exactitud del algoritmo. La métrica con mejor rendimiento fue: WMC (Weighted Methods per Class). Bansal et al. [4] plantearon un método de predicción del cambio, "en proyectos" y "entre proyectos" a través de: métricas y características de distribución de conjuntos de métricas para definir reglas de selección de conjuntos de datos para entrenamiento y prueba, utilizando el algoritmo C4.5 (Generador de árboles de decisión) para dicho fin.

Enfatizando que los conjuntos de datos seleccionados son de suma importancia para la clasificación. Para la clasificación de los archivos, utilizaron 3 clasificadores:

AdaBoost, LogitBoost y Bagging, sobre 3 aplicaciones de código abierto, donde Bagging tuvo los mejores resultados con un poco más del 65% de exactitud.

Al-Khiaty et al. [5] plantearon un método que utiliza métricas estructurales y evolutivas, estas últimas las obtienen del historial de versiones de las aplicaciones. El clasificador utilizado para la detección de archivos propensos al cambio fue redes Aductivas: Método grupal de manejo de datos (GMDH). Concluyen que su método mejora la precisión de clasificar que los métodos estadísticos. Hicieron experimentos con: métricas estructurales, evolutivas y combinadas. Los mejores resultados fueron con métricas combinadas.

Las métricas con mejor rendimiento fueron: edad y frecuencia de cambio (BCO), ocurrencia de cambio (FCH, LCH, CHO), tamaño (LCOM). Además de hacer un experimento más en el que tomaban las métricas de salida sugeridas por el clasificador en una siguiente iteración como entradas, alcanzando mejores resultados con un 83.58% de exactitud. Malhotra et al. [6] propusieron una comparación de 10 clasificadores de AA junto con un método de regresión logística para la predicción de archivos propensos al cambio.

Utilizaron tres aplicaciones de código abierto con dos versiones de cada una de ellas junto con sus métricas estructurales OO de cada archivo. Concluyen que el algoritmo con mejor rendimiento en las tres aplicaciones es máquinas de soporte vectorial (SVM) y que hay cinco clasificadores mejor evaluados que el método de regresión logística. Además de que las métricas que representan buenos indicadores de cambio son: CBO (Coupling between Objects), SLOC (Source Line of Code) y WMC (Weighted Methods for Class).

Los clasificadores de machine learning tienen una precisión del 64% al 80%. Bura et al. [9] utilizaron un método óptimo de predicción del cambio, a partir del manejo de control de versiones, Diagramas de Clases y Diagramas UML con los cuales se calculan cinco métricas: duración de ejecución, información de tiempo de ejecución, dependencia de clases, regularidad y popularidad. Las dos primeras se calculan directamente del código, la dependencia de clases con diagramas de clases y UML, la de regularidad con la extracción frecuente de conjuntos de elementos con algoritmo ABC y la de popularidad se calcula a partir de la regularidad tomándola como entrada (con fórmulas).

La clasificación la realiza aplicando algoritmo ID3: árbol de decisión. El método lo aplican sobre dos aplicaciones de código abierto. Y los resultados revelan que las clases con valores de métricas altas, son más propensas a cambios. Yanb et al. [10] proponen un modelo de predicción de clases propensas al cambio. Esto a través de un esquema no etiquetado (sin control de versiones) utilizando un algoritmo de clasificación no supervisado de última generación llamado CLAMI y haciendo el comparativo con una variación del mismo llamado CLAMI+.

El método se probó para catorce aplicaciones de código abierto, mostrando que los métodos no supervisados obtienen buenos resultados comparados con: LogitBoost, Multilayer perceptron, Radial basis function network, Support vector machine y K-means. Utilizan cinco métricas OO: WMC, DIT, NOC, RFC, LCOM. Su método lo compararon: con el método usando datos históricos del mismo proyecto (within a project) y con el método usando datos históricos de otros proyectos (cross a project), concluyendo que los métodos no supervisados dan resultados similares o los mejoran a los supervisados.

Garba [11] Propone un método que usa la entropía como medida efectiva de la degradación, considerándola como la tendencia del software a volverse más complejo y desorganizado por los cambios los que lo vuelve más difícil y costoso de mantener.

Valores bajos de entropía indica madurez del software. Se calcula el índice de madurez del software, el cual es una métrica que proporciona un indicador de la estabilidad software en función de los cambios que se producen para cada versión del producto. Determinando el modelo de costo constructivo (COCOMO) para sugerir la inversión que se tiene que hacer en aquellos archivos que tengan altos niveles de entropía.

3. Antecedentes de la investigación

En este trabajo de investigación entenderemos como Hotspots a la pieza o piezas de software que son propensas al cambio o al error como lo consideraron en [6][7] y [8]. En la mayoría de los desarrollos podemos identificar una pieza de software como un archivo, que puede representar una clase para lenguajes OO o puede representar un archivo cabecera para el lenguaje C. En el desarrollo moderno, es común el uso de dos herramientas que permiten tener disciplina en el desarrollo: los sistemas de control de versiones (SCV) y los sistemas de control de incidentes (SCI).

Generalmente, los cambios a realizar en el sistema se registran en el SCI como un incidente (puede ser una nueva funcionalidad o una corrección de defecto) y una vez aceptados se realizan en el código que es cambiado y que es versionado en el SCV mediante un "commit". Cuando los equipos tienen una buena disciplina de desarrollo, asocian los commits que realizan a los incidentes registrados en el SCI. Esta práctica permite identificar qué commits están asociados a incidentes relacionados con defectos (bugs) y, más específicamente, qué archivos son modificados para corregir bugs.

En este artículo, haremos referencia a estos commits relacionados con corrección de bugs como "bug commits". Cuando un mismo archivo aparece frecuentemente en "bug commits", se considera que es un Hotspot ya que es propenso a cambio y el hecho de que tenga que ser cambiado frecuentemente cuando se corrigen defectos permite pensar que es dentro de él que se originan estos defectos. El problema de la identificación de hotspots mediante esta métrica es que solo puede obtenerse de manera tardía.

En esta investigación buscamos identificar conjuntos de métricas de código que ayuden a detectar Hotspots de manera temprana. Estas métricas son obtenidas a partir de diversas herramientas de análisis de código, las cuales ofrecen por archivo, métricas estructurales tales como: tamaño, complejidad, presencia de "malos olores" de código o de diseño, dependencias, entre otras; todas ellas pueden ser obtenidas desde la primera versión del sistema. Otras herramientas toman como entrada códigos fuente, algunas otras el código compilado y algunas más consideran también información proveniente de los SCV y SCI.

En este estudio utilizamos para generar métricas por archivos las siguientes herramientas: Archinaut (ARCH), Designite (DES), Succinct Code Counter (SCC), DV8 (DV8) y SonarQube (SQ). Esta selección de herramientas es la misma que se realizó en [8] ya que se busca poder comparar nuestros resultados con los de ese estudio. ARCH toma varias fuentes de datos como entrada, incluida la información de dependencia generada por Depends y el historial de SCV de un proyecto. DES toma el

Tabla 1. Métricas seleccionadas para el estudio.

Id Métrica	Clave de Métrica	Descripción
M01	Des_Design Smells	Diseño oloroso
M02	SCC_CLOC	Tamaño en Lineas de Código Fisico (LOC)
M03	SCC_COMPLEXITY	Complejidad
M04	ARCH_Dependent Partners	Número de archivos que dependen de este archivo (fan-in)
M05	ARCH_Depends on Partners	Número de archivos de los que depende este archivo (fan-out)
M06	ARCH_Total Dependendencies	Fan in + Fan out
M07	ARCH_CoChange Partners (h)	Número de archivos que cambiaron al mismo tiempo que el archivo en uno o más "commits"
M08	DV8_TotalIssues	Número total de problemas de DV8 para los archivos
M09	DV8_Clique	Número de componentes fuertemente conectado en las que participa este archivo
M10	DV8_Crossing(h)	Número de cruces en los que participa este archivo
M11	DV8_ModularityViolation(h)	Número de violaciones de modularidad en los que participa este archivo
M12	DV8_PackageCycle	Número de ciclos de paquetes en los que participa este archivo
M13	DV8_UnhealthyInheritance(h)	Número de jerarquías de herencias en mal estado en los que participa este archivo
M14	DV8_UnstableInterface	Número de interfaces inestables en los que participa este archivo
M15	DV8_PresentinIssues	¿En cuántos de 6 problemas este archivo está presente?
M16	SQ_Issues	Número de problemas de SonarQube
M17	DV8_TargetChangeCount	Número de veces que aparece un archivo en un commit asociado con error ("bug commit")

código fuente como entrada e identifica los olores en los niveles de implementación, diseño y arquitectura. SCC es una herramienta para medir el tamaño y la complejidad del código.

DV8 realiza el cálculo del nivel de desacoplamiento, el costo de propagación, la identificación de raíces arquitectónicas y la identificación de anti patrones de diseño. SQ es una herramienta de revisión automática de código de grado industrial ampliamente utilizada y reconocida que define un amplio conjunto de reglas de calidad [8].

Dentro de las diferentes métricas que ofrecen las herramientas descritas previamente, simplificando aquellas que describen la misma información, la que se comprueba que son más fiable y las que se ofrecen a nivel archivo, tenemos la Tabla 1 en la que se listan las métricas utilizadas en nuestra investigación. Las columnas de la Tabla 1 son: Id Métrica, Clave de Métrica (que incluye como prefijo la herramienta de la que se obtuvo) y una descripción.

Además de que se marca el Identificador de la métrica con una "(h)" al final para indicar que esa métrica es obtenida del historial que proporcionan los SCV y SCI, y que pueden ser obtenidas desde los primeros "bug commit". Las primeras dieciséis métricas serán las que utilizaremos para nuestro estudio, las cuales nos ofrecen características por archivo de tipo: estructural como tamaño por número de líneas de código (CLOC) o complejidad (COMPLEXITY) y de ciclos como archivos que dependen de uno (fan-

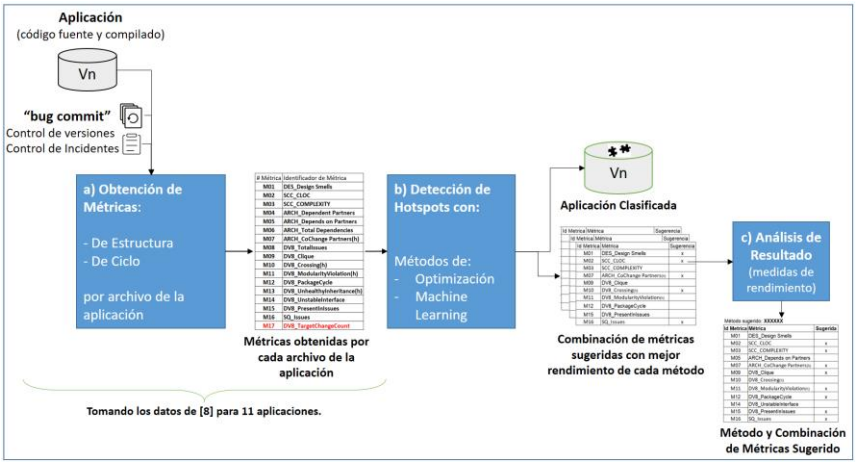


Fig. 1. Diagrama de la metodología aplicada en la detección de Hotspots.

in) o paquetes que dependen de otro (Package Cycle). La métrica DV8 Target ChangeCount (TCC) es especialmente importante ya que nos indica el número de "bug commits" que ha tenido un archivo. Por cada aplicación de las once disponibles se hizo un análisis con el diagrama de cajas y bigotes para la métrica TCC y una vez analizados los datos se determinó seleccionar como Hotspots todos los archivos cuyo valor en esta métrica fuera igual o mayor al Q3 (cuartil 3). El resto de los archivos se consideran como No Hotspots. Como podemos ver en la Tabla 2 se identificaron como Hotspots el 5.3% del total de los archivos que están presentes en nuestra investigación.

4. Metodología de la investigación

La metodología que se aplicó en este trabajo está representada en el flujo de la Figura 1. y cómo podemos observar consta de tres etapas: a) Obtención de métricas. A partir de la versión actual del sistema y de los registros de los "bug commit", se obtienen las métricas de cada una de los archivos que componen el sistema: dieciséis para ser evaluadas y una que sirve de referencia para comprobar la efectividad de la clasificación del método (TCC). Se retoma la extracción de métricas que realizaron [8] para las once aplicaciones que se usaron en esta investigación, por lo que este paso ya está realizado.

b) Detección de Hotspots a través de métodos de clasificación. Esta etapa recibe como entrada las métricas de cada archivo de la aplicación a analizar y entrega como resultado los archivos de la aplicación clasificados como Hotspots y los resultados de la evaluación de cada uno de los métodos de clasificación aplicados, con las variantes en los conjuntos de datos por la normalización de los mismos. c) Análisis de Resultados. Se toma como entrada los resultados de los seis métodos aplicados para la clasificación, se analizan para determinar cuál de ellos es el que ofrece los resultados más altos en las medidas de rendimiento que regularmente se utilizan para evaluar los clasificadores como en [4, 5, 6] y que se explican más adelante, principalmente: sensibilidad,

Tabla 2. Proyectos de software para la investigación.

	Entrenamiento									Prueba		
Aplicación	**tez**	**tajo**	**knox**	**nifi**	**oozie**	**sentry**	**flume**	**deltaspike**	**hbase**	**broke-j**	**amqp-0x**	**Totales**
Achivos Totales	839	1557	863	4112	812	561	441	742	1786	1865	561	**14,139**
Archivos Hotspots (TCC)	90	78	23	123	98	12	58	28	129	92	12	**743**
												5.3%

Combinación	{	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0** }
Núm. de Métrica		M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12	M13	M14	M15	M16

Fig. 2. Representación binaria de una combinación de métrica.

especificidad y el área bajo la curva (AUC), las cuales reflejan la capacidad de detectar Hotspots.

4.1. Proyectos de software para entrenamiento y prueba

Para la investigación se consideró un conjunto de once proyectos de software OO con las siguientes características: están desarrollados en Java, son mantenidos por "The Apache Software Foundation", son de código abierto, y los mensajes de commit del SCV están vinculados con incidencias en el SCI.

Se consideraron 1,000 commits asociados a incidencias (TCC). Se dividieron los conjuntos de datos siguiendo la regla que se aplicó en [5] con una variación de 5% para favorecer a los datos de entrenamiento, quedando en 75:25, con nueve aplicaciones para entrenamiento y dos para el entrenamiento (broker-j y ampq-0x) para comprobar el resultado del método sugerido como el mejor predictor de Hotspots.

En cada una de las aplicaciones se tiene definidos cuáles y cuántos son los archivos que se están marcando como Hotspots, en base a la medida TCC, que nos servirán para validar el resultado de la clasificación que se obtenga de cada uno de los métodos. La Tabla 3 muestra la lista de proyectos, así como la cantidad de archivos totales y la cantidad de archivos considerados como Hotspots.

A través de la matriz de confusión de cada experimento obtuvimos las medidas de rendimiento que nos permitieron determinar el método que mejor identifica los Hotspots. Se clasificaron un total de 14,139 archivos, de los cuales 743 (5.3%) son catalogados como Hotspots (con ayuda de la métrica TCC, previamente comentada).

4.1. Normalización de los datos

La normalización de los conjuntos de datos, como ya lo mencionamos, es el ajuste a los valores medidos en diferentes escalas respecto a una escala común. Se utilizó esta técnica estadística de normalización de datos para explorar si es posible encontrar resultados diferentes y más favorable debido a que analizando los valores entre las métricas de la Tabla 2 vemos que son muy diversos. Para buscar la homogeneidad de los datos se optó por aplicar: a) Normalización por Máximo valor en cada métrica. Por métrica de los datos se encuentra el valor máximo y se realiza el cociente entre el valor

de cada archivo en esa métrica y el valor máximo obtenido para la misma métrica. De esta manera tendremos valores entre 0 y 1.

Esto evita que una métrica con un valor muy grande influya más que otras métricas donde sus valores sean más bajos. b) Normalización por Máximo valor de CLOC (M02), Se eligió número de líneas de código (CLOC) dado que es sabido que a archivos grandes mayor riesgo de presentar errores, por lo que al normalizar el resto de las métricas con base en esta se provoca que todas estén a un mismo nivel.

En este caso se busca que los datos estén homogeneizados con respecto al tamaño del archivo más grande de la aplicación que se vaya a evaluar, de esta forma se propone que ahora todos los archivos tienen valores de métricas homogéneos y puedan ser comparables. Esta normalización también remueve las correlaciones que existen entre métricas como complejidad y tamaño.

4.2. Métodos propuestos para clasificar Hotspots

Como recordaremos, una de nuestras preguntas de investigación es encontrar si existe una combinación de métricas que nos ayuden a clasificar los Hotspots de un sistema. Para ello, se consideraron dos métodos para clasificar: Exhaustivo (Exh) el cual nos permitió evaluar todas las combinaciones de métricas combinado con tres tratamientos de los conjuntos de datos (sin normalizar, normalizar por máximo por métrica y normalizar por máximo CLOC) y Bayes Ingenuo (BI) con función de probabilidades condicionadas para datos continuos (Gaussiana) combinado con los mismos tres tratamientos de datos que el Exh, por lo que en total se tuvieron seis métodos a evaluar.

Para cada método: i) Se implementó en lenguaje C, ii) Para cada una de las aplicaciones se evaluó cada una de las combinaciones de métricas posibles para determinar qué combinación nos da las mejores medidas de rendimiento al ser evaluada, iii) Se propone una representación binaria de la combinación que se estará evaluando (Fig. 2), donde un 1 indica que la métrica se utiliza y un 0 que no.

Como podemos ver en el ejemplo de la Fig. 2, se estaría evaluando una combinación de métricas que utiliza las métricas: M01, M03, M06, M08, M09, M11, M12, M14 y M15 presentadas en la Tabla 1, iv) Dado que son 16 métricas tendremos 2^16 (65,635) combinaciones posibles, v) Las medidas de rendimiento (obtenidas a partir de la matriz de confusión) que se utilizaron y que se usan más frecuentemente para la evaluación de los métodos de clasificación son: Exactitud (proporción de predicciones correctas y total de predicciones), Precisión (proporción entre los positivos reales predichos y todos los casos positivos), Sensibilidad (fracción de verdaderos positivos), Especificidad (fracción de verdaderos negativos), Medida-F (valor único ponderado entre precisión y exactitud) y Análisis ROC (área bajo la curva –AUC-, representación gráfica de Sensibilidad vs. 1-Especificidad), vi) Los resultados de cada experimento estarán dados por los elementos de la matriz de confusión: TP (verdaderos positivos –es Hotspots y se clasifica como Hotspot-), TN (verdaderos negativos –No es Hotspots y se clasifica como No Hotspot-), FP (falsos positivos –No es Hotspots y se clasifica como Hotspot-) y FN (falsos negativos –es Hotspots y se clasifica como No Hotspot-) con los cuales se podrán calcular las medidas de rendimiento del experimento i-ésimo.

Con las medidas de rendimiento serán con las que estaremos evaluando el resultado del método de clasificación más apto. A continuación, discutiremos el detalle para cada uno de los dos métodos.

4.2.1. Método exhaustivo (Exh)

Para este método se aplicó: Por cada combinación de métricas se evaluó con la función de evaluación (FE) a todos los archivos a para obtener los Hotspots Propuestos (HPP). Los HPP serán los archivos que resulten con valor más alto en FE descrita en la Ecu. 1. Lo anterior se logra ordenando de manera descendente todos los archivos de la aplicación que han sido evaluados con la *FEa* y se tomará el mismo número de archivos que tenemos establecidos como Hotspots en la Tabla 2 por aplicación.

Con ello se procede a calcular la matriz de confusión y sus medidas de rendimiento. Donde Valor(a, m) es el valor de la métrica m para el archivo a, *CM(m)* es el valor de la representación binaria de la combinación que se está evaluando, teniendo como posibles valores un 1 o un 0. Para este método se realizarán tres experimentos: **a) Exh (sNorm)** sin normalizar, no se le hace nada a los valores de las métricas de cada base de datos, **b) Exh (nMax)**, se Normalizan los datos por valor Máximo en cada métrica. Se homogeniza la base de datos con valores entre 0 y 1, **c) Exh (nCLOC)**, es Normalizando los datos por Máximo Número de Líneas (métrica: M02):

$$FEa = \sum_{m=1}^{16} (Valor(a,m) * CM(m)) \quad (1)$$

.

4.2.2. Método bayes ingenuo (BI)

Para este método y tomando en consideración la recomendación de [10] sobre los clasificadores supervisados, se aplicó: El clasificador supervisado de BI en su versión para datos continuos que utiliza la distribución Gaussiana (Ec. 3) para calcular las probabilidades condicionadas *p(Mn|Cj)* de la Ec. 2, donde: *Cj* son las clases que serán asignadas a cada archivo (*C1*=Hotspots y *C0*=No Hotspot), *Mn* son los valores de las diferentes métricas que se están evaluando del archivo que se está clasificando, μ_{Cj} es la media de la clase *Cj* y σ^2_{Cj} es la desviación estándar de la clase *Cj*. Para cada archivo se realizará la evaluación de la probabilidad condicional por clase: *p(Cj|M1,...,Mn)* y se clasificó como Hotspot cuando *p(C1|M1,...,Mn)* > *p(C0|M1,...,Mn)*, en otro caso se clasificó como No Hotspot:

$$p(Cj|M1,...,Mn) = p(Cj) \prod_{i=1}^{n} p(Mi|Cj) \quad (2)$$

,

$$p(Mi|Cj) = \frac{1}{\sqrt{2\pi\sigma^2_{Cj}}} * e^{-(Mi - \overline{\mu_{Cj}})2 /(2\sigma^2_{Cj})} \quad (3)$$

.

Se evaluó cada combinación de las métricas (2^16) para determinar la combinación que tenga la mejor medida de rendimiento. Se aplicó a los conjuntos de datos de entrenamiento. Para las nueve aplicaciones de entrenamiento originalmente definidas en la Tabla 3, se dividirán para tener la porción de datos de entrenamiento y de pruebas para BI: se toma una de las aplicaciones como de prueba y las otras ocho se utilizan

para entrenamiento y así sucesivamente se va variando la aplicación de prueba, de esta forma nos resultan nueve experimentos los cuales ofrece la variedad que se busca para validar el clasificador de BI. Para este método se realizaron tres experimentos: **a) BI (sin Norm)**, no se normaliza la base de datos. **b) BI (nMAX)**, se normalizan los datos por máximo valor de cada métrica, **c) BI (nCLOC)**, se normalizan los datos por Máximo Número de Líneas (métrica: M02).

5. Análisis de resultados

En esta sección se describen los resultados obtenidos de los dos métodos considerados, así como una comparación de ellos para posteriormente saber cuál es el que mejor identifica los Hotspots y con cuál combinación de métricas. Se recapitulan los elementos que estuvieron presentes en cada uno de los experimentos: a) dieciséis métricas para cualquier base de datos y una más que sirve para clasificar los HPR, b) once aplicaciones, nueve para entrenamiento, dos para validar los resultados y dar respuesta a la P2 de la investigación, c) seis métodos de clasificación a validar y d) seis medidas de rendimiento que se usarán para evaluar los resultados.

5.1. Resumen de resultados

Se considera la medida de rendimiento AUC que se introdujo anteriormente como el indicador más representativo para seleccionar el método de clasificación, dado que esta medida engloba la capacidad del clasificador en detectar los Hotspots como los No Hotspots (ponderación de Sensibilidad y Especificidad). Al ser los conjuntos de datos tan desbalanceados (5.3% Hotspots y 94.7 No Hotspots), ambas clases nos interesan y AUC representa el balance de ambas clasificaciones.

Los valores de AUC van entre 0.0 y 1.0, donde 0.0 es que no clasifica nada y 1.0 es la clasificación perfecta de todos los datos. Los clasificadores son evaluados a través de AUC con los siguientes rangos: [0.0, 0.50) es como lanzar una moneda, [0.50, 0.60) malo, [0.6, 0.75) regular, [0.75, 0.90) bueno, [0.90, 0.97) muy bueno y [0.97, 1.00) excelente.

En la Tabla 3, mostramos el resumen de los resultados obtenidos en cada experimento, identificando la mejor combinación de métricas en cada uno de ellos y podemos comentar lo siguiente: **i)** Los dos métodos que más destacan por sus resultados en las diferentes medidas de rendimiento son: Exh (nMAX) y BI (nMAX), **ii)** Exh (nMAX) obtiene los mejores resultados en cuatro de las seis medidas de rendimiento que se están considerando, pero los resultados en Sensibilidad (capacidad de detectar Hotspots) está en un 53.7%.

iii) BI (nMAX) tiene a su favor dos de las seis medidas de rendimiento que se están considerando, donde en Sensibilidad tiene un 81.6% y una Especificidad de 87.8% lo que sugiere un balance entre ambas clasificaciones (Hotspots y No Hotspots) lo cual se refleja en el AUC con un valor de 84.7%, iv)

El método que utiliza el menor número de métricas es el de BI (nMAX) el cual solo utiliza tres de ellas: SCC_CLOC, DV8_Crossing(h) y DV8_ModularityViolation(h), revisando las métricas tenemos que una es el tamaño de los archivos (CLOC) y las otras dos se basan en el historial de las aplicaciones obtenidas de los SCV; DV8_ Modularity

Tabla 3. Mejores resultados de cada método evaluado con aplicaciones de entrenamiento.

		Medidas de rendimiento					
Método	**Métricas sugeridas**	**Exactitud**	**Precisión**	**Sensibilidad**	**Especificidad**	**Medida-F**	**AUC**
Exh (sNorm)	M01,M08,M09,M10,M11, M13,M14,M15,M16	0.936	0.475	0.476	0.965	0.619	0.721
Exh (nMAX)	M01,M02,M03,M05,M10, M11,M14,M15,M16	0.944	0.535	0.537	0.969	0.670	0.753
Exh (nCLOC)	M02,M08,M09,M11,M12, M15,M16	0.936	0.467	0.467	0.965	0.608	0.716
BI (sNorm)	M01,M02,M03,M07,M09, M10,M11,M12,M15,M16	0.897	0.362	0.584	0.919	0.478	0.751
BI (nMAX)	M02,M10,M11	0.873	0.323	0.816	0.878	0.458	0.847
BI (nCLOC)	M01,M03,M05,M07,M11, M12,M13,M14,M15,M16	0.884	0.292	0.560	0.905	0.410	0.733

Tabla 4. Resultados obtenidos con aplicaciones de prueba.

		Medidas de rendimiento					
Método	**Métricas sugeridas**	**Exactitud**	**Precisión**	**Sensibilidad**	**Especificidad**	**Medida-F**	**AUC**
Exh (sNorm)	M01,M08,M09,M10,M11, M13,M14,M15,M16	0.936	0.475	0.476	0.965	0.619	0.721
Exh (nMAX)	M01,M02,M03,M05,M10, M11,M14,M15,M16	0.944	0.535	0.537	0.969	0.670	0.753
Exh (nCLOC)	M02,M08,M09,M11,M12, M15,M16	0.936	0.467	0.467	0.965	0.608	0.716
BI (sNorm)	M01,M02,M03,M07,M09, M10,M11,M12,M15,M16	0.897	0.362	0.584	0.919	0.478	0.751
BI (nMAX)	M02,M10,M11	0.873	0.323	0.816	0.878	0.458	0.847
BI (nCLOC)	M01,M03,M05,M07,M11, M12,M13,M14,M15,M16	0.884	0.292	0.560	0.905	0.410	0.733

Violation(h) en [8] es una de las métricas que en ese estudio de correlación de métricas se detecta como una de las mejores predictora de Hotspots.

En la Tabla 3 columna Métricas sugeridas, mostramos el resumen de las métricas sugeridas en cada método y de donde podemos comentar lo siguiente: i) La métrica que está presente en todas las combinaciones de métricas de los seis métodos es DV8 Modularity Violation(h) (número de violaciones de modularidad en las que participa este archivo), lo cual reafirma a la conclusión realizada por [8], que esta métrica es una buena predictora de Hotspots.

ii) Dos métricas que participan en cinco de los seis métodos, estás son: DV8_PresentInIssues (en cuantos errores de las 5 herramientas está presente este archivo) y SQ_Issues (se engloban: "malos olores" de código, errores, vulnerabilidades y Hotspots de seguridad) [8]. iii) Dos de las métricas propuestas no fueron utilizadas en ninguna de las seis combinaciones sugeridas por los métodos, estas son: ARCH_DependentPartners (número de archivos que dependen de él –FanIn-) y ARCH_TotalDependencies (FanIn + FanOut). iv) El método que propone la combinación con el menor número de métricas es BI(nMAX), las cuales son: SCC_CLOC, DV8_Crossing(h) y DV8_ModularityViolation(h).

Como lo mostramos en la Tabla 1, se tienen dos aplicaciones de prueba para validar los resultados con los datos de entrenamiento. Se ejecutaron las mejores soluciones de

cada uno de los métodos utilizados y ver el comportamiento con otras aplicaciones que no han participado aún en el proceso de entrenamiento. Los resultados se presentan en la Tabla 4 y obtenemos resultados similares a los obtenidos con los datos de entrenamiento, donde los métodos de Exh (nMAX) y BI (nMAX) nos dan los mejores resultados en AUC con valores de 86.5 y 95.3%.

5.2. Respuesta a las preguntas de investigación

Con los resultados obtenidos daremos paso a dar respuesta a nuestras preguntas de investigación e ir obteniendo conclusiones de nuestra investigación.

P1. ¿Existen combinaciones de métricas que puedan ofrecer mejores resultados en la detección de Hotspots en las aplicaciones de software? **R1**. Con base a los resultados que podemos ver en la Tabla 3, con resultados desde 71.6% y hasta 84.7% en AUC, lo cual indica que tenemos una buena clasificación de Hotspots. Teniendo el mejor resultado de sensibilidad y AUC con el método BI (nMAX) y una combinación de 3 métricas (M02, M10 y M11 mostradas en la Tabla 3 columna Métricas sugeridas), favoreciendo tanto la detección de Hotspots (sensibilidad con un 81.7%) como los No Hotspots (especificidad con un 87.8%) obteniendo una AUC de 84.7%.

P2. ¿Las métricas para detectar Hotspots de una aplicación pueden funcionar para determinar Hotspots en otras? **R2**. Como podemos ver en la Tabla 4, para ambas aplicaciones se obtienen muy buenos resultados en exactitud con cualquiera de los métodos. También podemos ver que ambas aplicaciones de prueba, los dos métodos que más destacan son: Exh (nMAX) y BI (nMAX) lo que va acorde con el resultado que se obtuvo con las aplicaciones de entrenamiento. El AUC que ofrece BI (nMAX) es del 86.5% para broker-j (una aplicación mediana) y del 95.3% para amqp-0x (una aplicación pequeña), por lo que podemos recomendar el método de BI (nMAX) junto con la combinación de métricas sugerida (M02, M10 y M11) para intentar predecir Hotspots en otras aplicaciones.

P3. ¿Es posible determinar qué algoritmos resulta mejor para la detección de Hotspots entre un método de clasificación y un método base (exhaustivo)? **R3**. Con base a los resultados obtenidos en este estudio podemos señalar que el método de Bayes Ingenuo con los diferentes tratamientos de datos casi siempre estuvo por arriba en los resultados del AUC con respecto al método Exhaustivo, por lo que podemos proponerlo como un buen clasificador de Hotspots.

6. Conclusiones y trabajo futuro

Como pudimos darnos cuenta, la clasificación de los archivos de Hotspots en una aplicación es de suma importancia al tratar de disminuir su DT, por lo que proponemos los resultados obtenidos en esta investigación como una opción para poder ayudar en la detección de Hotspots, con una algoritmos de AA, como lo es BI, acompañado de una normalización de datos para homogeneizarlos y el uso de tres métricas que resultaron bastante reveladoras la información que representan y que permite al algoritmo trabajar para clasificar con un porcentaje bastante aceptable del 83.5% y del 95.3% de AUC para aplicaciones totalmente ajenas al proceso previo de entrenamiento. Se podría hacer una detección temprana de Hotspots, si los hay, después de los primeros

"bug commit". Cabe señalar que conforme se vaya dotando al algoritmo de mayor información de entrenamiento de las tres métricas propuestas se podrían mejorar los resultados de clasificación. Los métodos exhaustivos no estuvieron tan lejos de ofrecer también buenos resultados, solo que el conjunto de métricas que ocupan es mayor a la combinación que se encontró con BI, pero estos también podrían ser una opción.

También podemos comentar que los resultados de [4, 5, 6] están por debajo en medidas de rendimiento como la exactitud o el AUC, lo que nos hace pensar que nuestros resultados son prometedores. Al igual que en la mayoría de los estudios previamente realizados y revisados, la métrica que está presente en la mayoría de ellos, al igual que en el nuestro, es el número de líneas de código de cada archivo (CLOC), dado que a mayor número de líneas mayor la probabilidad de incurrir en errores.

El trabajo futuro que se visualiza, es: trabajar en un método sistemático para la obtención de las métricas con mayor facilidad, continuar creciendo la base de entrenamiento del algoritmo para su aprendizaje y una posible mejora de la detección de Hotspots y probar el método con aplicaciones no necesariamente con las mismas características base que se tomaron para esta investigación y validar el comportamiento del método.

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: A modelling notation for complex web applications. In: Mendes, E., Mosley, N. (Eds) Web Engineering, Springer, pp. 335–364 (2006)
2. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based Web Engineering: An approach based on Standards. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (Eds.): Web engineering: Modelling and implementing web applications. Springer, pp. 157–191 (2008) doi: 10.1007/978-1-84628-923-1_7
3. Sánchez-Santamaria, M., García-García, L. A.: La ingeniería web: Desarrollo de aplicaciones web de alta calidad. HYPATIA (2011)
4. Vera, P., Pons, C., González, C., Giulianelli, D., Zodríguez, R.: Metodología de modelado de aplicaciones web móviles basada en componentes. In: XV Workshop de Investigadores en Ciencias de la Computación, pp. 451–455 (2013)
5. Moroni, N., Señas, P.: Uso de grafos para el modelado de experiencias educativas colaborativas basadas en la web. In: XIII Congreso Argentino de Ciencias de la Computación, pp. 1134–1146 (2007) http://sedici.unlp.edu.ar/handle/10915/22913
6. Bouchrika, I., Ait-Oubelli, L., Rabir, A.: Mockup-based navigational diagram for the development of interactive web applications. In: Proceedings of the International Conference on Information Systems and Design of Communication, pp. 27–32 (2013) doi: 10.1145/2503859.2503864
7. Lipschutz, S.: Sets and basic operations on sets: Set theory and related topics. McGraw-Hill, pp. 5–6 (1998)
8. Lipschutz, S., Lars Lipson, M.: Set Theory Discrete Mathematics. McGraw-Hill, pp. 1–22 (2007)

ISSN 1870-4069

Plataforma digital para la seguridad escolar mediante el control de acceso automatizado basado en el reconocimiento facial

Cynthia B. Perez[1], Jesus R. Villavicencio[1], Jessica Beltran[2], Roberto Limon-Ulloa[1], Perla Duran[1], Samuel Torres[1], Bryan García[1]

[1] Instituto Tecnológico de Sonora (ITSON), México

[2] Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI-IPN), México

{cynthia.perez, roberto.limon, jesus.villavicencio162447, perla.duran204151, samuel.torres202413, bryana.garcia}@potros.itson.edu.mx, jbeltranm@ipn.mx

Resumen. Las condiciones de inseguridad pública ameritan que se desarrollen sistemas orientados a brindar seguridad dentro de instituciones educativas. Se pueden utilizar diferentes enfoques tecnológicos orientados a apoyar la seguridad, entre ellos, el uso de sistemas de monitoreo automático de personas para verificar que solo ingresan a los campus quienes tienen previa autorización. Una forma no intrusiva de abordar este problema es mediante el reconocimiento facial con técnicas de aprendizaje automático. En este trabajo, se describe el desarrollo de una plataforma digital para el control de acceso automático y notificación de alertas a los guardias de seguridad de una institución educativa. Se realizaron experimentos con la base de datos LFW [28] y con imágenes de estudiantes y profesores de una institución educativa para conocer y comparar la efectividad del reconocimiento facial. Así mismo, se utilizaron características extraídas con redes neuronales profundas y algoritmos de clasificación para identificar a la persona. Los resultados indican la viabilidad de implementar esta plataforma *in situ* y proporcionan direcciones para el trabajo futuro.

Palabras clave: Reconocimiento Facial, universidad inteligente, openface, seguridad escolar.

A Digital Platform for School Security through Automated Access Control based on Facial Recognition

Abstract. Public unsafe conditions justify the development of systems that help to provide security inside educational institutions. There can be different technological approaches to support security, including automatic systems to monitor that only authorized people can enter campus. A non intrusive way to address this problem, is through facial recognition with machine learning techniques. In this work, is described the development of a digital platform to control access automatically and to send alerts to security guards from an educational institution. Experiments were conducted using the data set LFW [28] and with images from students and professors from the

educational institute to know and compare the effectiveness of facial recognition. Thus, features extracted with deep learning and classification algorithms to identify persons were used. The results show the viability to implement the platform *in situ* and provide directions for future work.

Keywords: Facial recognition, smart campus, openface, school surveillance system.

1. Introducción

Hoy en día, el tema de seguridad pública urbana es una de las principales preocupaciones que tiene la sociedad alrededor del mundo, principalmente en aquellos países en vías de desarrollo. En México, de acuerdo con el Instituto Nacional de Estadística y Geografía (INEGI), en el año 2020, el 67.8% de la población mayores de 18 años considera que vivir en su ciudad es inseguro [1]. En ese sentido, debido a la situación que se vive en torno a la seguridad en el país, la ocurrencia de incidentes relacionados con la seguridad no es ajena a las instituciones educativas.

Es por ello que particularmente en las universidades, existe un gran interés por incorporar tecnologías de la información para facilitar la administración y seguridad de los campus. Para apoyar estas iniciativas, se han propuesto diferentes sistemas automatizados para el control de acceso en organizaciones públicas y/o privadas basadas en comunicación de campo cercano (Near Field Communication, NFC) [8], identificación por radio frecuencia (Radio-frequency Identification, RFID) [2-4], sistemas basados en iris y sistemas basados en reconocimiento facial [5-7].

El reconocimiento facial es una de las técnicas más populares para incorporar a los sistemas de seguridad por su facilidad de uso, su naturaleza sin contacto y que no es una técnica intrusiva ni agresiva [1,4,6,7]. De acuerdo con Victor Skinner [9], la Universidad de San Francisco hace uso de un sistema comercial basado en reconocimiento facial llamado iOmniscient [1] para administrar el acceso de los estudiantes a los dormitorios de la universidad, donde el objetivo es contar con una forma de controlar el acceso a las residencias universitarias que no sea intrusivo y que no requiera que los estudiantes se detengan en su acceso al campus.

Mediante este sistema de control de acceso basado en reconocimiento facial, la universidad puede tener un historial de los ingresos a las residencias e identificar a los visitantes no autorizados que necesitan registrarse. Por otro lado, la escuela de desarrollo infantil de la Universidad de Seattle y la academia católica St. Therese utilizan un sistema de control de acceso basado en reconocimiento facial llamado SAFR[2] donde el personal debe sonreír a la cámara para que el sistema les permita el acceso abriendo automáticamente la puerta de entrada, de lo contrario, deben pasar a una oficina a registrarse. Así mismo, este sistema se utiliza para monitorear pasillos y áreas comunes permitiendo activar alertas de sonido, envío de mensajes y llamar a la policía en caso que así lo requieran [10].

En China, la Universidad Normal de Beijing hace uso de sistemas de reconocimiento facial en todos los dormitorios para el control de acceso donde los estudiantes tienen tres opciones para activar el sistema de reconocimiento facial como pasar su credencial

[1] https://iomni.ai/
[2] https://es.safr.com/

(ID) del campus, decir su nombre o ingresar los últimos cuatro dígitos de la contraseña de su credencial para que se les permita el acceso al edificio de dormitorios [11]. Así mismo, la Universidad de Pekín en China desarrolló un sistema de reconocimiento facial para identificar tanto a estudiantes como personal académico y administrativo cuando ingresan al campus.

En este caso, la universidad acondicionó el acceso instalando torniquetes electrónicos con lectores de credencial y tabletas con cámaras incorporadas para que el estudiante o el personal tengan la opción de ingresar a la institución por medio de su credencial o por medio del sistema de reconocimiento facial [12]. De esta manera, el control de acceso en instituciones educativas mediante reconocimiento facial se está considerando como una herramienta tecnológica que ofrece una solución eficiente, precisa, de acceso rápido e imparcial ayudando a las instituciones a mejorar la seguridad del campus permitiendo el acceso a solo personas autorizadas.

Sin embargo, implementar de forma eficiente este tipo de sistemas a gran escala es desafiante y representa un costo que muchas veces las instituciones educativas no pueden costear [13]. Es por ello, que este documento describe el desarrollo de una plataforma digital para el control de acceso por medio de reconocimiento facial basado en aprendizaje profundo que permita identificar personas autorizadas y no autorizadas en los diferentes puntos de entrada/salida de una institución educativa utilizando librerías de código abierto como OpenFace [14].

El resto del artículo está organizado como sigue. En la Sección 2, se describe el problema de reconocimiento facial basado en aprendizaje profundo. En la Sección 3 se presenta el diseño e implementación de la plataforma digital. La Sección 4 presenta la experimentación y resultados obtenidos. Finalmente, las conclusiones y trabajo futuro son presentados en la Sección 5.

2. Reconocimiento facial basado en aprendizaje profundo

El reconocimiento facial es un problema clásico de la visión por computadora que hasta la fecha continúa siendo de gran interés para la comunidad científica, así como también para la sociedad a través de aplicaciones comerciales. La Figura 1 presenta el proceso que se sigue para llevar a cabo el reconocimiento facial de forma automatizada de tal forma que primero es necesario que el sistema reciba como entrada una imagen (offline/online); posteriormente, se lleva a cabo el proceso de detección y alineación del rostro para que inicie el proceso de extracción de características y a su vez, el sistema compare mediante el proceso de correspondencia/clasificador, las características extraídas de la imagen de entrada con el conjunto de características de los rostros que se encuentran en la base de datos. Finalmente, se evalúa el rendimiento del algoritmo mediante diferentes métricas y se muestra la imagen con los datos de la persona que ha sido reconocida.

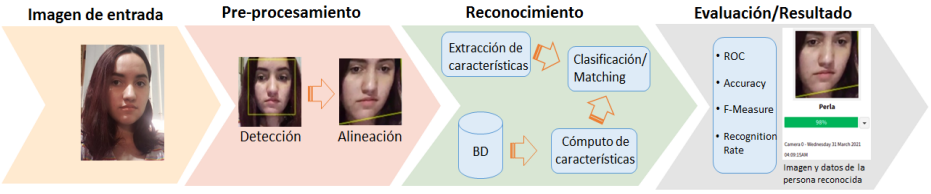


Fig. 1. Esquema tradicional del reconocimiento facial.

Los algoritmos de reconocimiento facial utilizados en aplicaciones reales hacen uso de cámaras donde las imágenes pueden tener variaciones como cambios de iluminación, baja resolución, movimientos del rostro, oclusiones, difuminación en la calidad de la imagen, entre otros. Es por ello que la mayoría de los sistemas de reconocimiento facial trabajan adecuadamente en entornos controlados como el de la iluminación y un posicionamiento frontal del rostro.

En este tipo de variaciones, los algoritmos descriptivos han mostrado ser eficaces. Algunas de las técnicas descriptivas más populares se encuentran SIFT [15], LBP [16] y HOG [17]. Pérez y Olague propusieron operadores descriptivos evolucionados llamados $RDGP_2$ que mejoraron en un 26.7% el rendimiento de los algoritmos de reconocimiento de objetos por imágenes de SIFT, SURF y GLOH [18]. Es por ello, que existe un interés en abordar el reconocimiento facial bajo este tipo de condiciones llamadas no restrictivas en donde el uso de técnicas de aprendizaje profundo ha ganado popularidad ya que ofrece mayor robustez contra las diferentes variaciones que pueden afectar el proceso de reconocimiento [19].

En ese sentido, las técnicas descriptivas han sido utilizadas en algoritmos de aprendizaje profundo como las redes neuronales convolutivas en un entorno de IoT [13,20-21], demostrando recientemente una mejora significativa en la precisión del reconocimiento facial, como por ejemplo el algoritmo DeepFace de Facebook [22], el FaceNet de Google [23], COCOLoss [24] y ArcFace [25]. En este trabajo, se utilizó la versión de código abierto de FaceNet llamada OpenFace [14] para llevar a cabo el proceso de reconocimiento facial bajo un enfoque de aprendizaje profundo.

OpenFace es uno de los algoritmos de reconocimiento facial de código abierto más preciso que combina la técnica Triplet Loss de Deepface [22] con una red neuronal convolucional profunda basada en la red de FaceNet [23]. Openface se entrenó con 500k imágenes combinando dos bases de datos utilizadas para reconocimiento facial, CASIA-WebFace [26] y FaceScrub [27] obteniendo un modelo de red llamado *nn4.small2* con un menor número de parámetros para la base de datos utilizada. Para la etapa de detección del rostro se puede utilizar HOG (Histogram of Oriented Gradient) y SVM (Support Vector Machine) de DLib, o bien, Haar cascade de OpenCV.

De acuerdo con Amos et. al. [14], HOG y SVM obtienen mejor precisión que el detector de OpenCV en la etapa de detección. Una vez que el modelo de red es obtenido y se obtienen las características de los rostros, se utilizan técnicas de clasificación para completar la tarea de reconocimiento. Finalmente, para evaluar el rendimiento de OpenFace, Amos et. al. utilizaron la base de datos LFW [28] con el modelo de red *nn4.small2.v1* obteniendo una precisión de 0.9292± 0.0134.

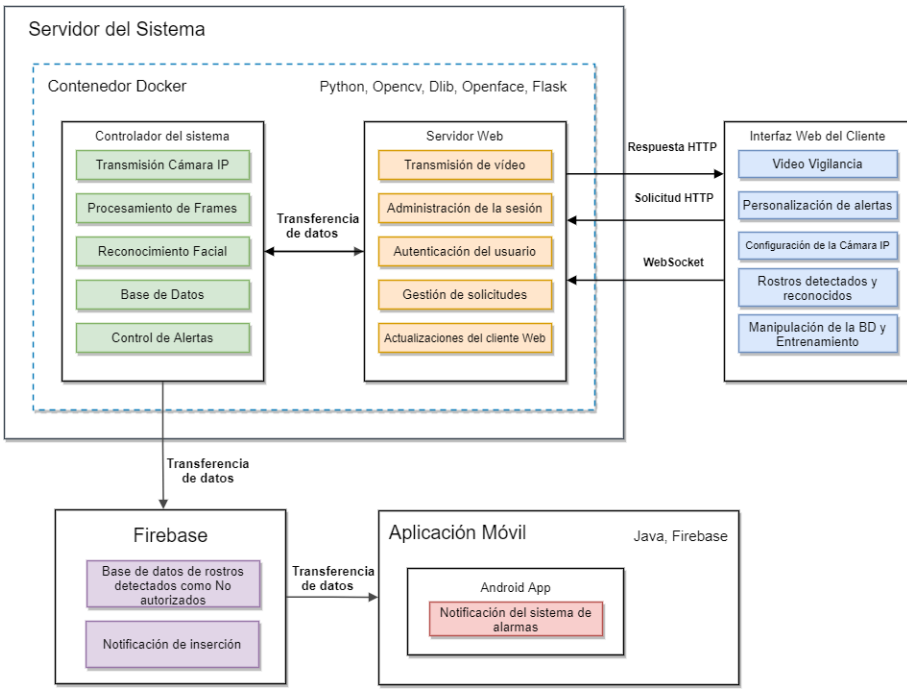


Fig. 2. Diseño de la arquitectura de la plataforma para el control de acceso universitario.

La etapa de clasificación permite que el sistema pueda identificar quién es quién no solamente saber si la persona se encuentra o se parece a alguien almacenada en la base datos. En ese sentido, existen diferentes clasificadores que han sido utilizados en el reconocimiento facial como la máquina de vectores de soporte (SVM, por sus siglas en inglés), Naive Bayes y Árboles de decisión (Decision Trees) por mencionar algunos.

El algoritmo SVM, es un algoritmo de aprendizaje supervisado que se utiliza en problemas de clasificación; éste encuentra los parámetros óptimos de un hiperplano separador que maximiza el margen entre los objetos de entrenamiento. Se utiliza el truco del kernel, que consiste en mapear las entradas, que no son linealmente separables en el espacio de entrada, a un espacio de alta dimensión. El kernel lineal entre dos objetos $\boldsymbol{x}$ y $\boldsymbol{x}'$ se define como $Klineal = \boldsymbol{x} \cdot \boldsymbol{x}'$, mientras que el kernel radial, o Gaussiano, se define como $Kradial = e\text{^}(-(\|x - x\text{^}' \|\text{^}2/(2\sigma\text{^}2))$). Adicionalmente, para incluir regularización, se agrega el parámetro C en el proceso de optimización [30].

El algoritmo probabilístico de clasificación Bayes ingenuo o *Naive Bayes* en inglés, se basa en el teorema de Bayes y supone que las características de los objetos son independientes. En la etapa de entrenamiento, se calculan las probabilidades a *priori* y se encuentran los parámetros que maximizan la verosimilitud dadas las características obtenidas para los objetos de cada clase. Se obtiene la probabilidad *posteriori* de que un objeto evaluado pertenece a una clase con base a sus características [31]. Por otro lado, los modelos de árboles de decisión para clasificación consisten en un conjunto de

reglas para estratificar el espacio de predicción en subgrupos más pequeños y homogéneos [32].

3. Descripción de la plataforma digital

En esta sección, presentamos la descripción de la plataforma digital para el control de acceso en un campus universitario mediante reconocimiento facial. El objetivo es detectar a las personas autorizadas/no autorizadas que desean ingresar al campus de forma automática por medio de reconocimiento facial.

Cuando una persona no autorizada desea entrar a la institución, se manda automáticamente una alerta a una aplicación móvil diseñada para los guardias de seguridad del campus quienes están capacitados para tomar la acción correspondiente; por ejemplo, tomar sus datos y permitirle la entrada temporalmente, registrarlo en el sistema, o bien, no permitirle la entrada.

El diseño de la plataforma se basó en la propuesta de Brandon Joffe [29], en donde considera un servidor dedicado al procesamiento de reconocimiento facial. Para ello, se usa una comunicación basada en web que permite la interacción con el usuario y manipulación de la plataforma. Adicionalmente, se desarrolló una aplicación móvil para el manejo de alertas, la cual se comunica con el servidor central por medio de Firebase[3], obteniendo de esta manera, información sobre el proceso de reconocimiento facial que se enviará como alerta a la aplicación móvil, ver Figura 2.

La plataforma está diseñada para establecer comunicación con cámaras IP por medio de las cuales se obtendrá el rostro de las personas. Una vez que se obtiene la información de la(s) cámara(s) se procede a realizar el proceso de reconocimiento y automáticamente se envía la alerta a la aplicación móvil cuando se identifica una persona no autorizada por la institución, ver Figura 3. En este caso, el objeto IPCamera transmite el video directamente desde una cámara IP para que sea procesado y transmitirlo al cliente web.

Cada cámara tiene su propio objeto MotionDetector y FaceDetector que son utilizados por otros procesos para llevar a cabo el reconocimiento facial. En ese sentido, el proceso de reconocimiento está basado en el algoritmo OpenFace usando el modelo de red neuronal pre-entrenado *nn4.small2.v1.t7* y el clasificador LinearSVM.

4. Experimentación y resultados

Esta sección describe la experimentación llevada a cabo para la etapa de clasificación del proceso de reconocimiento facial descrito en la Sección 2 y se presenta la implementación de una plataforma digital.

3 https://firebase.google.com/

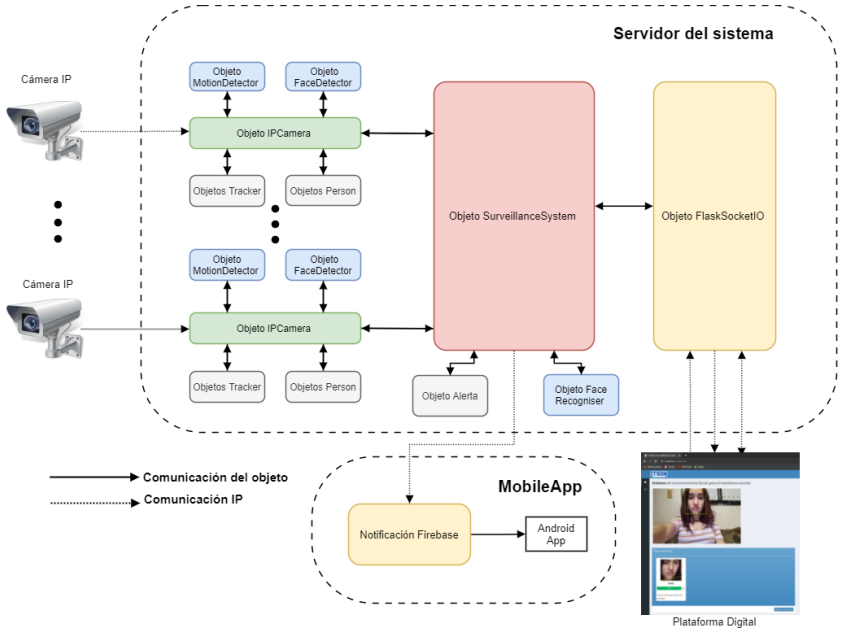


Fig. 3. Funcionamiento de la plataforma para el control de acceso y envío de alertas.

4.1. Clasificación de características para el reconocimiento facial

Para llevar a cabo la comparación de clasificadores se utilizó una computadora Lenovo, Intel Core i5 con 8GB de RAM y sistema operativo Linux Ubuntu kernel 5.3.0-40. En ese sentido, para los experimentos se utilizó la base de datos LFW [28] la cual es ampliamente utilizada en el reconocimiento facial. Esta base de datos cuenta con 5749 clases (personas) y 13233 fotos.

Se observó que la mayoría de las clases cuentan con una sola foto lo cual para este tipo de experimentación es insuficiente. 5591 clases tienen entre 1 y 10 fotos; 96 clases tienen entre 10 y 20 fotos; 28 clases tienen entre 20 y 30 fotos; 15 clases tienen entre 30 y 40 fotos; 7 clases tienen entre 40 y 50; 4 clases tienen entre 50 y 60 fotos y el resto de clases más de 60 fotos por clase.

Considerando esto, se decidió utilizar el grupo de 28 clases que contiene un total de 653 fotos y el grupo de 4 clases que contiene un total de 212 fotos. Los resultados de este experimento se pueden observar en las Figuras 4 y 5.

Para llevar a cabo el entrenamiento y la prueba de cada uno de los clasificadores, se dividió la base de datos en un 80% para entrenamiento y un 20% para pruebas para cada uno de los grupos (28 y 4 clases). Como se puede observar en la Tabla 1, LinearSVM y GaussianNB son mejores que RadialSVM y DecisionTrees en ambos grupos de clases. Es por ello, que se decidió utilizar LinearSVM como clasificador en el proceso de reconocimiento facial.

Adicionalmente, se evaluó el sistema de reconocimiento de la plataforma utilizando una base de datos de la institución con 91 fotos de 3 estudiantes y 1 profesor (4 clases)

Tabla 1. Comparación del rendimiento de 4 clasificadores utilizados en el reconocimiento facial sobre la base de datos LFW.

Clasificador	28 clases		4 clases	
	% Reconocimiento	F-Measure	% Reconocimiento	F-Measure
LinearSVM	96.15	0.9804	100	1
RadialSVM	95.38	0.9764	100	1
DecisionTrees	70.77	0.8288	92.86	0.96295
GaussianNB	96.15	0.9804	100	1

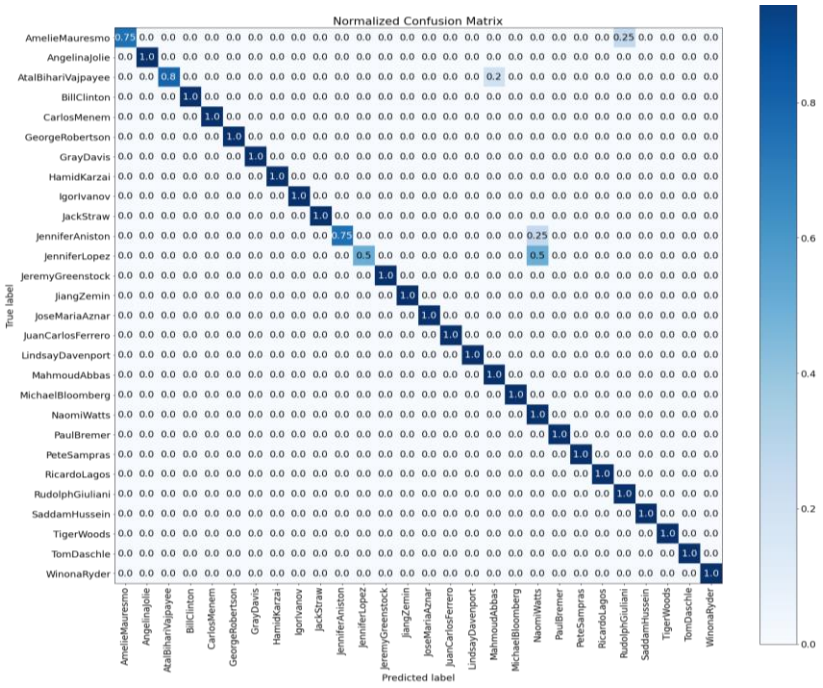


Fig. 4. Matriz de confusión normalizada para 28 clases de la base de datos LFW [28] utilizando Linear SVM.

obteniendo como resultado un porcentaje de reconocimiento del 95.24% y una F-Measure de 0.9978, ver Figura 6.

4.2. Plataforma digital para el control de acceso universitario

La plataforma se implementó en una computadora Lenovo, Intel Core i5 con 8GB de RAM y sistema operativo Linux Ubuntu kernel 5.3.0-40 y se utilizó la aplicación IP WebCam en un teléfono móvil con sistema operativo Android para usar la cámara del

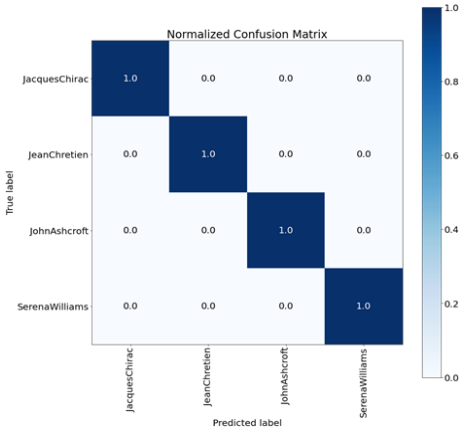


Fig. 5. Matriz de confusión normalizada para 4 clases de la base de datos LFW [28] utilizando LinearSVM.

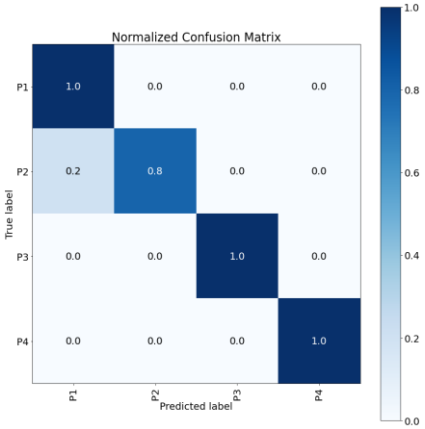


Fig. 6. Matriz de confusión normalizada utilizando la base de datos de la Institución (4 clases).

teléfono como una cámara IP. En el caso de la aplicación móvil, ésta fue desarrollada con Java en Android Studio en un Huawei Nova 3 y probada en un Samsung S9.

Por otro lado, la plataforma cuenta con la opción de agregar personas al sistema en tiempo real. Si esto sucede, se debe de re-entrenar el clasificador en la misma plataforma para que la persona sea reconocida como persona autorizada en el futuro.

En ese sentido, la Figura 7 presenta un ejemplo de la plataforma mostrando una persona autorizada por el sistema, en donde se despliegan datos como nombre de la persona y el porcentaje de confianza indicando con ello, la certeza del reconocimiento

Por otro lado, la Figura 8 presenta un ejemplo de la plataforma cuando una persona no autorizada se detecta, si esto sucede, se envía automáticamente una alerta a la aplicación móvil, ver Figura 9b. De lo contrario, cuando el sistema reconoce a una persona autorizada la aplicación móvil se presenta sin cambios, ver Figura 9a.

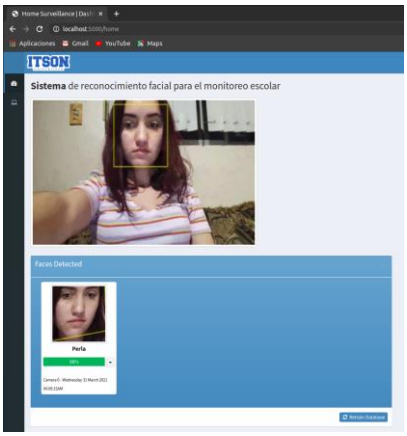


Fig. 7. Ejemplo de la plataforma reconociendo a una persona autorizada por el sistema.

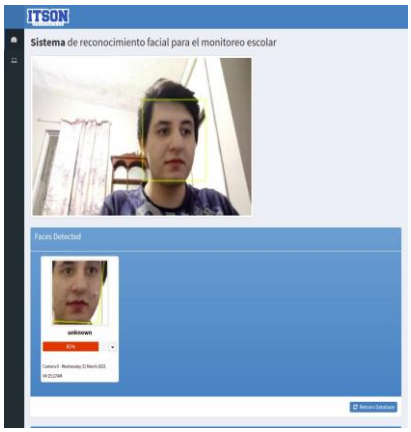


Fig. 8. Ejemplo de la plataforma mostrando a una persona NO autorizada.

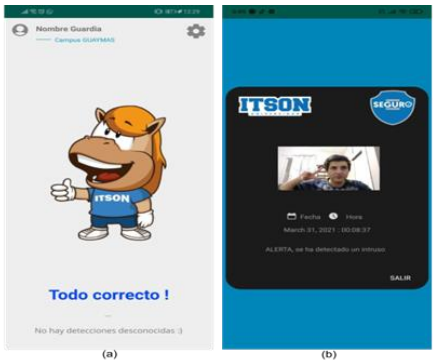


Fig. 9. Alerta enviada de la plataforma a la aplicación móvil. (a) Persona autorizada (b)

5. Conclusiones y trabajo futuro

Este trabajo presenta el diseño e implementación de una plataforma digital para el control de acceso del personal académico, administrativo y estudiantil a un campus universitario mediante reconocimiento facial. La plataforma digital fue implementada en un equipo local, sin embargo, se tiene programado migrarla a un servidor institucional con cuatro cámaras IP HikVision H.265+ instaladas en los accesos de entrada al campus.

Esta plataforma permite ingresar nuevas personas al sistema y cuenta con el envío de alertas a una aplicación móvil de forma automática cuando una persona no autorizada intenta ingresar a la institución. Para la etapa de reconocimiento facial se utilizó el algoritmo de código abierto OpenFace con el modelo de red neuronal pre-entrenado *nn4.small2.v1.t7* y el clasificador LinearSVM.

Para esto, se presenta una comparación de 4 clasificadores utilizando la base de datos LFW con 28 y 4 clases mostrando un rendimiento superior al 95%. Adicionalmente, se evaluó la plataforma con 4 clases utilizando imágenes del personal académico y estudiantil mostrando un porcentaje de reconocimiento del 95.24% y un valor de F-Measure de 0.9978.

En ese sentido, como trabajo futuro se tiene pensado implementar la plataforma digital *in situ*, y para ello será necesario preparar una base de datos institucional y proponer un esquema de reconocimiento a gran escala ya que la institución cuenta con alrededor de 21,818 estudiantes y 2,820 trabajadores, lo que representa alrededor de 24,638 clases para el sistema de reconocimiento facial.

Agradecimientos. Este trabajo fue financiado por el programa para el desarrollo profesional docente (PRODEP) 2019.

Referencias

1. INEGI. Departamento de comunicación social. Encuesta Nacional de Seguridad Pública Urbana (ENSU) México. https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2020/ensu/ ensu2020_10.pdf (2021)
2. Gonzalez-Garcia, C., Meana-Llorián, D., Pelayo G-Bustelo, B. C., Cueva-Lovelle, J. M., Garcia-Fernandez, N.: Midgar Detection of people through computer vision in the internet of things scenarios to improve the security in Smart Cities, Smart Towns, and Smart Homes. Future Generation Computer Systems, Elsevier, vol. 76, pp. 301–313 (2017) doi: 10.1016/j.future.20 16.12.033
3. Dong, X., Kong, X., Zhang, F., Chen, Z., Kang, J.: On campus a mobile platform towards a smart campus. SpringerPlus, pp. 1–9 (2016) doi: 10.1186/s40064-016-2608- 4
4. Kumar, P. M., Gandhi, U., Varatharajan, R., Manogaran, G., Jidhesh, R., Vadivel, T.: Intelligent face recognition and navigation system using neural learning for smart security in internet of things. Cluster Computing, no. 22, pp. 7733–7744 (2019) doi: 10.1007/s10586-017-1323-4
5. Rameswari, R., Kumar, S. N., Aananth, M. A., Deepak, C.: Automated access control system using face recognition. Materials Today Proceedings. vol. 45, pp. 1251–1256 (2020) doi: 10.1016/j.matpr.2020.04.664
6. Lee, H., Park, S. H., Yoo, J. H., Jung, S. H., Huh, J. H.: Face recognition at a distance for a stand-alone access control system. Sensors, vol. 20, pp. 1–18 (2020) doi: 10.3390/s20030785
7. Tisse, C. L., Martin, L., Torres, L., Robert, M.: Person identification technique using human iris recognition. In: Proceedings of. Vision Interface Canadian Image Processing and Pattern Recognition Society (CIPPRS), 15th International Conference on Vision Interface. pp. 294–299 (2002)
8. Bueno-Delgado, M. V., Pavón-Marino, P., De-Gea-Garcia, A., Dolon-Garcia, A.: The smart university experience: An NFC-based ubiquitous environment. In: Proceedings of the IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 799–804 (2012) doi: 10.1109/IMIS.2012.110
9. Skinner, V.: Universitiy of San Francisco using facial recognition to track dorm activity. EAG. News.org (2014). https://www.eagnews.org/2014/11/university-of-san-francisco-using-facial-recognition-to-track-dorm-activity/

10. Mikkelsen, D.: Two Seattle schools among first to use facial recognition software in the US. K5 News (2018). https://www.king5.com/article/news/education/two-seattle-schools-amongfirst-to-use-facial-recognition-software-in-us/281-609937626
11. Xiaofei, Du.: Facial recognition system introduced to Beiging Normal University dorms. People's Daily Online (2017) http://en.people.cn/n3/2017/0516/c90000-9216437.html
12. Gan, N.: Want to get into Peking University? Just show your face. Inkstone news (2018). https://www.inkstonenews.com/tech/peking-university-installs-facial-recognition-system-students-and-staff-campus-gate/article/2152893
13. Oh, S. H., Kim, G. W., Lim, K. S.: Compact deep learned feature-based face recognition for visual internet of things. The Journal of Supercomputing, vol. 74, pp. 6729–6741 (2018) doi: 10.1007/s11227-017-2198-0
14. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: Openface: A general-purpose face recognition library with mobile applications. Technical Report, CMU-CS-16-118, CMU School of Computer Science (2016)
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, vol. 60, pp. 91–110 (2004) doi: 10.1023/B:VISI.0000029664.99615.94
16. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 28, pp. 2037–2041 (2006) doi: 10.1109/TPAMI.2006.244
17. Déniz, O., Bueno, G., Salido, J., De la Torre, F.: Face recognition using histograms of oriented gradients. Pattern Recognition Letters, vol. 32, pp. 1598–1603 (2011) doi: 10.1016/j.patrec.2 011.01.004
18. Perez, C. B., Olague, G.: Learning invariant region descriptor operators with genetic programming and the f-measure. In: Proceedings of the 19th International Conference on Pattern Recognition, pp. 1–4 (2008) doi: 10.1109/ICPR.2008.4761178
19. Guo, G., Zhang, N.: A survey on deep learning based face recognition. Computer vision and image understanding, 189, pp-1–47 (2019) doi:10.1016/j.cviu.2019.102805
20. Yang, S., Luo, P., Loy, C. C., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3676–3684 (2015)
21. Farfade, S. S., Saberian, M. J., Li, L. J.: Multi-view face detection using deep convolutional neural networks. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, pp. 643–650 (2015) doi: 10.1145/2671188.2749408
22. Parkhi, O. Vedaldi, M., Zisserman A.: Deep face recognition. In British Machine Vision Conference, pp. 1–12 (2015)
23. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815–823 (2015)
24. Liu, Y., Li, H., Wang, X.: Rethinking feature discrimination and polymerization for large-scale recognition. arXiv preprint arXiv (2017) doi: 10.48550/arXiv.1710.00870
25. Deng, J., Guo, J., Xue, N., Zafeiriou, Arcface S.: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
26. Yi, D., Lei, Z., Liao, S., Li, Stan. Z.: Learning face representation from scratch arXiv preprint arXiv. (2014) doi: 10.48550/arXiv.1411.7923
27. Ng. H. W., Stefan, W.: A data-driven approach to cleaning large face datasets. IEEE International Conference on Image Processing, pp. 343–347 (2014) doi: 10.1109/ICIP.2014.7025068
28. Huang, G., B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild A database for studying face recognition in unconstrained environments. Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Erik Learned-Miller and Andras Ferencz and Frédéric Jurie (2008)

29. Joffe, B.: Home Surveillance with facial recognition. https://github.com/BrandonJoffe/home_ surveillance#features
30. Muller, K. R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE transactions on neural networks, pp. 181–201 (2001)
31. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification, 2nd edition, Jhon Wiley & Sons, Inc (2001)
32. James, G., Witten, D., Hastie, T., Tibshirani, R.: An introduction to statistical learning. Ed. Springer, 112, pp. 3–7 (2013) doi: 10.1007/978-1-0716-1418-1.pdf

ISSN 1870-4069

Recopilación y análisis de datos para la implementación de un sistema que ayude a la predicción de intenciones de robo en personas

Juan Carlos Sánchez Navarro, Blanca Estela Pedroza Méndez, José Federico Ramírez Cruz, Rubén Xochitiotzi Cuatecontzi, Abel Martínez Gorospe

Instituto Tecnológico Nacional de México / Campus Apizaco, Tlaxcala, México

{m19371377, blanca.pm, federico.rc, m19371381, m19371369}@apizaco.tecnm.mx

Resumen. Con el paso de los años, los robos a casa habitación, empresas y establecimientos, han ido en incremento, por lo que muchas instituciones buscan contrarrestar esa situación, ya que esto afecta a la integridad de las personas que son víctimas. Es evidente que no es fácil detectar las intenciones negativas de un individuo o de una persona que desea cometer un robo, sin embargo, existen trabajos que están enfocados a desarrollar sistemas que ayuden a este fin. En este artículo se analiza el cómo recopilar características para predecir intenciones de robo en personas, traspasando el conocimiento que algunos expertos en el tema han obtenido de manera empírica o psicológica. Con base al enfoque planteado, se puede generar información, a través de patrones obtenidos de las expresiones faciales de un sujeto y predecir emociones durante un intervalo de tiempo. Los resultados obtenidos muestran que al momento de cometer un robo las personas sienten miedo, sorpresa y por alguna razón felicidad, pero casi nunca sufren un disgusto. A partir de esta información se podría conocer la probabilidad de que algún individuo tenga intenciones de cometer un robo. Así mismo, a partir del análisis realizado, se propone una metodología para generar conjuntos de datos, a través de la detección de emociones por medio de micro expresiones, para posteriormente usar un algoritmo de aprendizaje automático no supervisado, como podría ser DBSCAN para predicción y agrupamiento de clústeres para un correspondiente análisis.

Palabras clave: Predicción de robos, reconocimiento facial de emociones, aprendizaje automático supervisado, aprendizaje automático no supervisado.

Data Collection and Analysis for the Implementation of a System for the Prediction of Robbery Intentions in People

Abstract. Over the years, robberies from homes, businesses and establishments have been on the increase, so many institutions seek to counteract this situation, since this affects the integrity of the people who are victims. It is evident that it is not easy to detect the negative intentions of an individual or a person who wishes to commit a robbery, however, there are works that are focused on

developing systems that help to this end. This article analyzes how to collect characteristics to predict robbery intentions in people, transferring the knowledge that some experts on the subject have obtained empirically or psychologically. Based on the proposed approach, the information can be generated through patterns obtained from the facial expressions of a subject and predict emotions during a time interval. The results currently obtained show that at the time of committing a robbery, people feel fear, surprise and for some reason happiness, but they almost never experience an upset. From this information, the probability that an individual has intentions to commit a robbery could be known. Likewise, based on the analysis carried out, a methodology is proposed to generate data sets, through the detection of emotions through micro expressions, to later use the unsupervised machine learning algorithm such as DBSCAN for prediction and grouping of clusters for a corresponding analysis.

Keywords: Theft prediction, facial emotion recognition, supervised machine learning, unsupervised machine learning.

1. Introducción

El saber si una persona tiene intenciones de realizar un acto ilícito es una tarea difícil, y aunque algunos tengan la capacidad de detectar esto, de manera empírica, no muchos cuentan con esta habilidad. Es por ello, que en este artículo se realiza un análisis de diversos algoritmos que pueden ayudar a generar una metodología para realizar un sistema capaz de predecir intenciones de robo en personas con base a sus patrones.

A partir de esta premisa, la idea de un análisis para crear un sistema, ayudaría a la toma de decisiones, con base a la detección de robos, lo cual, significaría una mejora en la calidad de vida, tanto de un consumidor, trabajador y/o dueño de algún establecimiento. Para ello, la pregunta de investigación es: ¿cómo realizar un sistema computacional capaz de procesar datos y transformarlos en información que permita predecir robos antes de ser cometidos o durante la acción?

Para resolver esta pregunta, es imprescindible conocer el área de impacto donde surge esta idea, la cual pertenece al campo de la seguridad pública o privada, donde el obtener soluciones cada vez más inteligentes es de suma importancia para resolver temas subsecuentes, como son los robos.

Por tanto, es importante saber definir que es un robo, y existiendo diferentes definiciones de acuerdo con la cultura y al contexto que aborda este tema, se puede definir como “apropiarse de algo ajeno por medio de la fuerza o por intimidación” [1]. La fuerza o intimidación que se maneja en la definición anterior puede tener graves consecuencias hacía una persona, ya sea una lesión pequeña o incluso la muerte.

Además, es importante conocer algunos otros términos para la comprensión de este artículo. El reconocimiento facial de emociones, es un dominio que estudia la forma de analizar y reconocer los movimientos faciales y los cambios en los rasgos del rostro para con esto hacer una predicción de la emoción de alguna persona [2].

El aprendizaje automático es una rama de la inteligencia artificial, dentro de la cual, se desarrollan técnicas para que las computadoras aprendan con base a la detección de patrones en conjuntos de datos [3].

El artículo está estructurado de la siguiente manera: en la sección 2, se presenta una revisión de diversos trabajos relacionados a la detección de robos, para encontrar

algoritmos que ayuden a la recopilación de datos y a su análisis. En la sección 3, se muestra el análisis propuesto, con una tabla de resultados, que muestra algunas observaciones para la implementación de un sistema para detección de robos. Finalmente, en la sección 4, una propuesta para un sistema capaz de predecir intenciones de robo en personas y avances de recopilación de datos, y en la sección 5, se dan las conclusiones y trabajos futuros.

2. Trabajos relacionados

La idea de usar sistemas por computador para ayudar a la seguridad no es un tema nuevo, dado que existen sistemas y publicaciones científicas antiguas que tenían esta finalidad, por ejemplo, en 1998 en Taiwán predominaban los criminales que chantajeaban a la gente por medio de llamadas. Por esta situación se propuso un sistema capaz de consultar una base de datos que contenía voces de criminales y delimitar resultados, que podrían ser la voz del criminal que estaba al otro lado del teléfono [4].

Entre otros ejemplos, existen sistemas de biometría que sirven para que una persona no reconocida, no pueda tener acceso a una caja fuerte, a un smartphone, entre otros. También existen rastreadores para saber en dónde está un objeto, por ejemplo, un carro, lo cual sirve mucho en dado caso de un robo. Basado en estos sistemas desarrollados para el campo de seguridad, uno de los que impulsa de manera significativa el planteamiento de este artículo, son las cámaras con sistema de detección de rostros.

Este sistema es muy atractivo, pues puede reconocer, a través de rasgos faciales, a una o varias personas y dentro del entorno de la seguridad este reconocimiento permite asegurar el acceso a un lugar u objeto, pero ¿qué pasaría si vamos más lejos? Si en lugar de detectar una persona, detectará si esa persona desea cometer una acción, por ejemplo, un robo. Basado en esa premisa, la gran pregunta realizada es ¿Cómo predecir si una persona quiere cometer un robo?

2.1. ¿Cómo predecir cuándo un individuo tiene intenciones de robar?

Basado en como predecir si una persona tiene intenciones delictivas, se puede realizar la pregunta siguiente: ¿Qué datos se necesitan dar a un ordenador para poder hacer esto posible? Para eso se toman varias contribuciones que se han hecho a lo largo del tiempo como las de [5–8].

Con base a estas, se puede observar que los actos criminales pueden ser individuales o colectivos y que los patrones de comportamiento son de suma importancia para predecir actividades criminales, pero ¿Qué pasa si ahora se genera un enfoque relacionado solamente con atributos de un ladrón? ¿Se pueden utilizar cuestiones psicológicas para la detección de este tipo de criminales?

Para contestar estas preguntas se rescata información útil de un artículo publicado en el 2019 escrito por Yajie Su, Han Lu y Hongzhou Zhang titulado “Modelado de mapeo cualitativo del sentido de seguridad de los delincuentes en casos de robo” [9]. En dicho artículo se utiliza una relevante investigación dentro del campo de la psicología criminalística, donde el trabajo recoge la experiencia y el conocimiento de los investigadores en casos de robo, a través de entrevistas y cuestionarios, y obtiene

los principales factores que afectan la sensación de seguridad interior de los delincuentes, en todo el proceso de comisión de un delito.

Ahora bien, uno de los fundamentos para esa investigación fue la realizada por Maslow, quien propuso que, dentro de la necesidad de una persona interviene la seguridad a nivel de demanda después de las necesidades fisiológicas básicas de los seres humanos. "Una sensación de seguridad es la sensación de fortaleza o debilidad de una persona al lidiar con el miedo y la ansiedad, que se manifiesta principalmente en una sensación de certeza y control" [9].

También existe una clasificación teórica de ladrones, basada en observaciones [10] y se definen en cuatro tipos: el organizado, el desorganizado, el interpersonal y el oportunista. Dentro de los cuales se pueden ver características psicológicas y puntos básicos relacionados con sus emociones.

Conociendo esto, se puede deducir que para detectar si un individuo tiene intenciones de robar, se pueden examinar patrones psicológicos, y aun mejor, patrones emocionales de la(s) persona(s) para generar un sistema capaz de alarmar cuando esto suceda.

2.2. Reconocimiento de emociones

Para crear un sistema capaz de reconocer emociones con el propósito anterior, primero se necesita saber cómo recolectar los patrones emocionales, para detectar personas con estas intenciones de robo. De esta manera, se conoce que existen sistemas computacionales basados en reconocimiento de patrones de voz, reconocimiento de patrones en imágenes (en este caso, secuencias de varias imágenes), reconocimiento de patrones cerebrales, reconocimiento de la frecuencia cardiaca, entre algunos otros, que se utilizan para el reconocimiento de emociones.

Pero bien, teniendo en cuenta que la mayoría de los lugares de aplicación en donde tendría utilidad un sistema capaz de predecir intenciones de robo en personas, ya cuentan con sistemas de cámaras, en este artículo nos enfocamos en el reconocimiento con dichas herramientas, por lo tanto, una de las mejores opciones es enfocarse en el reconocimiento de imágenes.

Ya definido lo anterior, ahora se hará el seguimiento con base en cómo se clasifican las emociones, tomando en cuenta la investigación de Paul Ekman, considerado uno de los mejores expertos del mundo en el análisis del comportamiento no verbal y la detección del engaño. Ekman demostró que Darwin estaba en lo cierto en 1969 con el libro llamado "La expresión de las emociones en los animales y el hombre", y en 1978 presentó junto a su compañero Wallace Friese el FACS (Facial Action Coding System), un sistema de codificación que recoge todos los movimientos expresivos del rostro, en unidades de acción (AU) [2].

En la actualidad, este sistema se utiliza en la comunidad científica internacional. Las siete emociones básicas universales que él propuso son; la sorpresa, la tristeza, el desprecio, el miedo, la ira, la alegría y el asco, y cada una tiene sus propias unidades de acción, características que facilitan su reconocimiento. Esa investigación ayudó a la creación de varios sistemas de reconocimiento de emociones [11–13].

Basándose en estas emociones principales, se entiende que se pueden utilizar para el reconocimiento de emociones en imágenes [13–16]. También se ha observado en varias investigaciones científicas que, para entrenar un modelo capaz de clasificar las

emociones dentro del campo de la visión artificial, exactamente dentro del reconocimiento de imágenes, el ocupar una red neuronal convolucional es el método ideal para recoger patrones que definen una emoción dentro de una imagen y una forma eficiente de extraer emociones de una persona es enfocarse en sus expresiones faciales, además de que esto necesita un poder de cómputo intermedio [16].

3. Elección de algoritmo clasificador

Ahora bien, para desarrollar un sistema que pueda realizar la acción que se menciona al comienzo de este artículo, se necesita utilizar algún algoritmo de aprendizaje automático para poder predecir no solo las emociones del ladrón, sino también, clasificar en grupos los conjuntos de emociones al momento de cometer un robo o al tener la intención de realizar esta acción, para ello existen diversas fuentes que ayudan a la comprensión de los métodos de aprendizaje automático como los que se presentan en [17], dónde se obtiene la clasificación de estos métodos, la cual es: aprendizaje supervisado y aprendizaje no supervisado. Entonces ¿Cuál es el tipo de aprendizaje automático adecuado para un sistema con la habilidad que se ha mencionado a lo largo de este artículo?

Aprendizaje supervisado. Los algoritmos del aprendizaje supervisado se basan en instancias (propiedades o características) que son etiquetadas para poder hacer una predicción sobre futuras instancias [18, 19].

Aprendizaje no supervisado. Los algoritmos de aprendizaje no supervisado utilizan instancias sin etiquetar para detectar patrones entre las instancias y clasificarlos para obtener predicciones [19].

Conociendo la finalidad que tiene cada uno de los tipos de aprendizaje, se puede deducir que, el algoritmo más adecuado para crear un sistema capaz de detectar intenciones de robo en personas es el aprendizaje no supervisado, puesto que aunque ya existen algunas clasificaciones para los ladrones [10], si se realiza el enfoque en las emociones que se pueden observar en expresiones faciales, podrían existir algunos grupos y/o subgrupos que no se especifican y pueden ser un diferenciador para el sistema. Además de que al existir instancias sin etiquetar el sistema podría dar grupos para ser evaluados.

Y ahora, ¿Qué algoritmo de aprendizaje no supervisado es más conveniente? Para descubrir aquello, se observan algoritmos de aprendizaje no supervisado que podrían ser candidatos óptimos para el desarrollo de un sistema como el que se ha mencionado a lo largo de este artículo.

K-medias. El algoritmo K-medias es muy utilizado puesto que es extremadamente sencillo de implementar y eficaz computacionalmente hablando, contra otros algoritmos de agrupamiento. Este algoritmo tiene la meta de particionar un conjunto de n-observaciones en k grupos, en el que cada observación pertenece al grupo con valor medio más cercano.

Debido a la forma tan rápida y variada que maneja este algoritmo, es utilizado para la agrupación de documentos [20], optimización de clústeres basados en imágenes [21], análisis de conglomerados [22], además de diversas aplicaciones que van desde reconocimiento en el área educativa, hasta el ámbito empresarial para grupos segmentados [23–29].

Tabla 1. Comparación de cuál algoritmo tiene un mejor potencial para ser elegido clasificador para el sistema detector de intenciones de robo.

Algoritmo	Definición	¿Optimo?, ¿Por qué?
Aprendizaje supervisado	Son técnicas para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento se componen de pares de objetos (generalmente vectores): un componente del par son los datos de entrada y el otro, los resultados deseados.	No, ya que no se conoce con certeza los valores y grupos deseados.
K-medias	Este algoritmo tiene el objetivo de dividir un conjunto de n observaciones en k grupos en los que cada observación pertenece al grupo con el valor medio más cercano.	Sé podría utilizar, pero, aunque es sencillo de implementar no reconoce el ruido y esto podría ocasionar fallas si no se controlan los datos.
Clustering jerárquico	Los algoritmos de agrupación jerárquica se basan en la creación de una jerarquía de agrupaciones donde cada nodo es una agrupación, que consta de agrupaciones de sus nodos secundarios.	No, puede ser un problema muy grande si aumentamos la cantidad de datos aumentando demasiado la complejidad.
DBSCAN	Este algoritmo puede encontrar cualquier forma de agrupación arbitraria sin verse afectado por el ruido. Además de eso, puedes encontrar la parte más densa de las muestras mediante centroides, dejando de lado las partes menos conocidas o ruidosas.	Sí, ya que se necesita crear k-grupos y eliminar posibles ruidos, dada la complejidad de los datos.

Clustering Jerárquico. Este algoritmo, al pertenecer al concepto de aprendizaje no supervisado sigue estando sin etiquetar previamente y solo se cuenta con variables que no son dependientes (características). A partir de esto, los algoritmos de clústeres jerárquicos se basan en la creación de una jerarquía de clústeres donde cada nodo es un clúster, que consta de clústeres de sus nodos hijos.

Como ventajas tiene que a diferencia de K-medias, en este no es necesario definir el número de clústeres que se van a formar; esto debido a que el algoritmo puede obtener el número de clústeres directamente. Además, se puede observar que es fácil de implementar y el denograma es muy útil para la interpretación de los datos.

También existen desventajas como que no es apropiado para conjuntos de datos muy grandes debido a que la complejidad aumenta y podría ser muy costoso computacionalmente hablando, y podemos agregar que podría ser difícil determinar el número correcto de clústeres por medio de un denograma.

Este algoritmo se utiliza para el procesamiento paralelo en memorias de computadora [30], segmentar información en datos de manera coherente y análisis de datos [31–33], mejorar la eficiencia de grupos de datos para su correspondiente análisis [34], sistemas de búsqueda [35] y análisis de valores atípicos [36].

DBSCAN. El algoritmo de K-medias es un método para agrupar datos de una forma no supervisada, pero ¿Qué pasa cuando se cuenta con clústeres de forma arbitraria, o

clústeres dentro de clústeres? Esta técnica y otras tradicionales podrían no compartir suficiente similitud de sus elementos en el mismo clúster, o no pertenecer a ninguno. En problemas donde fallan algoritmos como k-medias por los factores ya mencionados, se puede utilizar el agrupamiento espacial basado en densidad de aplicaciones con ruido (DBSCAN), debido a que este algoritmo puede encontrar cualquier forma arbitraria de clúster sin verse afectado por el ruido.

Además de que puede encontrar la parte más densa de las muestras por medio de centroides, haciendo a un lado partes menos densas o ruidosas. La noción de densidad puede verse como: el número de puntos dentro de un radio específico. Este algoritmo tiene múltiples aplicaciones, debido a que también puede detectar muestras que no pertenecen a ningún clúster y son solamente ruido, como en aplicaciones de estabilización en el campo de visión artificial [37], métodos de agrupación más novedosos [38–40], análisis de volúmenes grandes de datos [41], resoluciones complejas [42], entre muchas otras para análisis complejos de clustering.

Después de analizar estos algoritmos se llega a la conclusión que se muestra en la tabla 1.

4. Propuesta y avances

Con base al análisis anterior, se llegó a la conclusión de que uno de los patrones a analizar, que más predomina en un ladrón en tiempo real, son sus emociones, que pueden ser analizadas por un computador, esto con base a estudios psicológicos y criminalísticos antes mencionados. Además, se detectó que un aparato muy famoso en la actualidad son las cámaras de seguridad que cada vez son más usadas por negocios y casa habitación, las cuales se plantean como un método no invasivo y nos ayudan a un reconocimiento de emociones con base a las micro expresiones faciales, en este caso se basa en las siete emociones universales que de igual manera se menciona anteriormente.

Para recopilar los patrones se utiliza el reconocimiento facial de emociones que, para este sistema, se desarrolla con apoyo de la biblioteca gratuita de código abierto para el procesamiento de imágenes en tiempo real la cual es OpenCV2, la biblioteca de aprendizaje automático llamada Tensorflow y el lenguaje de programación Python, En la fig. 1. se pueden observar algunas imágenes cuando está trabajando el algoritmo.

Ahora bien, esos patrones actualmente se están recogiendo de videos de sujetos cometiendo algún robo, los cuales se analizan antes y durante dicha acción, y se calcula el porcentaje de tiempo de cada emoción principal (enojado, disgustado, temeroso, feliz, neutral, triste y sorprendido) durante cada video.

Por ejemplo: si durante un video donde el total de veces del conjunto de todas las emociones analizadas da un total de 100 y en ese mismo video se obtiene que 72 veces se encontró que el sujeto estuvo temeroso, entonces se calcula el promedio que tuvo esa emoción contra el total de todas que fue 100, y se obtiene el valor de 0.72 que este proceso se repite hasta hacerlo con cada emoción, para almacenar estas características en un CSV (delimitado por comas). En la Fig. 2. Se pueden observar algunos datos que ha encontrado al analizar las emociones con base al rostro de individuos al cometer un robo.

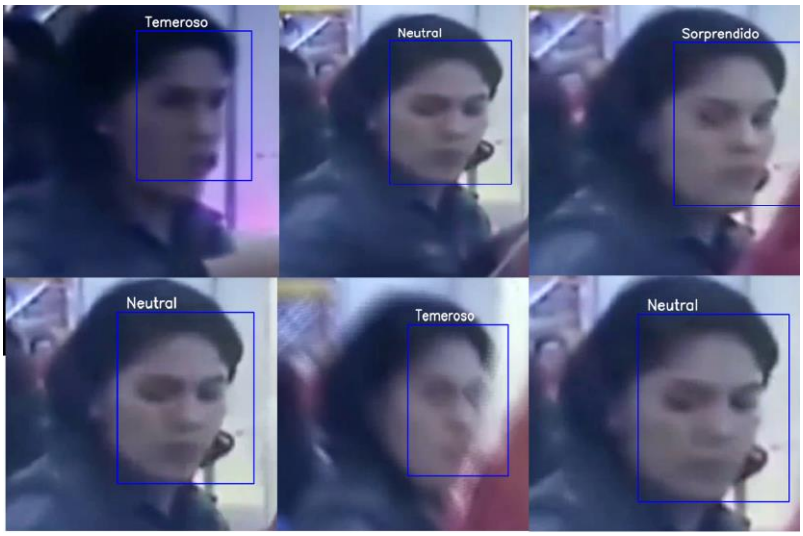


Fig. 1. Reconocimiento de emociones con base en micro expresiones faciales en tiempo real, al momento de recopilar características de una persona robando.

Dentro de la fig. 2. haciendo su correspondiente análisis, se observa que; las personas al momento de llevar a cabo un robo presentan un miedo, en la mayoría de los casos con un concreto toque de satisfacción (por alguna razón que podría ser al obtener lo que ellos querían), y en algunos otros casos miedo con vergüenza. En casos muy concretos toques de enojo y estado neutral.

Además, se observa que la emoción que menos predomina es el disgusto. Actualmente se cuenta con 100 datos que ya han sido analizados por expertos en el campo de la psicología para comparar los resultados obtenidos en cada video contra lo que ellos piensan que pasa emocionalmente por la mente del ladrón.

Pero para pasarlos por el clasificador DBSCAN (el cual por el análisis mostrado en la sección 4 se propone utilizar) y compararlo contra otros clasificadores, se necesitan más datos para lograr un desempeño notable a ser evaluado, por eso actualmente se siguen recopilando datos.

Y, ahora bien, teniendo un conjunto de datos más grande y analizándolo con el algoritmo DBSCAN, se podrían observar los clústeres que este formaría, para ser analizados dentro del campo de la psicología criminalística. Además, como principalmente se pensó, el poder hacer que un computador prediga con cierta exactitud si una persona tiene intenciones de robar en tiempo real o por un video ya grabado.

5. Conclusiones y trabajo a futuro

La creación de un sistema capaz de predecir si un individuo tiene intenciones de cometer un robo es una tarea difícil, por lo que un análisis como el que se realiza en este artículo es importante para tener resultados óptimos. Se observa que, con esta metodología planteada se tendría un medio no invasivo como lo es el reconocimiento facial de emociones a través de secuencias de imágenes que, pueden ser obtenidas

	0	1	2	3	4	5	6
0	4.178273	0.000000	72.144847	13.927577	3.899721	4.735376	1.114206
1	8.391608	0.000000	35.664336	27.272727	15.384615	11.188811	2.097902
2	1.650165	0.000000	39.273927	39.603960	9.240924	6.930693	3.300330
3	0.488998	0.000000	42.909535	30.562347	19.804401	5.990220	0.244499
4	2.685950	0.000000	45.247934	13.016529	8.884298	29.958678	0.206612
5	0.000000	0.000000	95.597484	0.000000	3.773585	0.000000	0.628931
6	14.000000	0.000000	30.000000	30.000000	12.000000	12.000000	2.000000
7	0.000000	0.000000	27.702703	6.756757	38.851351	26.689189	0.000000
8	0.387597	0.000000	39.631783	3.294574	35.658915	12.209302	8.817829
9	5.042017	0.420168	49.159664	26.890756	3.781513	3.781513	10.924370
10	1.657459	0.000000	59.668508	11.602210	7.734807	3.314917	16.022099
11	1.255230	0.000000	35.564854	43.096234	2.510460	7.112971	10.460251
12	9.691630	0.000000	61.674009	24.229075	2.202643	0.881057	1.321586
13	0.000000	0.000000	41.176471	12.745098	14.705882	31.372549	0.000000
14	2.064897	0.000000	44.100295	7.227139	6.342183	11.061947	29.203540
15	3.482587	0.000000	22.885572	6.965174	27.860697	38.805970	0.000000
16	2.040816	0.000000	33.163265	26.275510	11.734694	18.622449	8.163265
17	3.500000	0.000000	47.750000	2.750000	4.500000	41.000000	0.500000
18	4.687500	0.000000	53.125000	26.562500	9.375000	6.250000	0.000000
19	0.543478	0.000000	57.608696	5.978261	22.826087	10.869565	2.173913
20	0.000000	0.000000	45.531915	31.489362	5.957447	16.170213	0.851064
21	4.545455	0.000000	59.090909	9.090909	13.636364	13.636364	0.000000
22	3.896104	0.000000	42.424242	30.303030	6.493506	13.852814	3.030303
23	0.608519	0.000000	58.215010	22.312373	6.085193	11.561866	1.217039
24	4.564315	0.000000	39.834025	47.302905	3.319502	4.149378	0.829876
25	0.000000	0.000000	33.450704	54.577465	11.619718	0.352113	0.000000
26	0.943396	0.000000	28.301887	35.849057	23.584906	9.433962	1.886792
27	1.282051	0.000000	28.205128	57.692308	3.846154	6.410256	2.564103
28	0.865266	0.000000	24.351051	32.385661	27.194067	13.597033	1.606922
29	0.788955	0.000000	82.840237	11.045365	3.550296	1.183432	0.591716
30	0.000000	0.000000	45.302013	26.174497	16.442953	11.409396	0.671141
31	0.000000	0.000000	30.000000	23.636364	12.727273	32.727273	0.909091

Fig. 2. Muestra de datos obtenidos con base al análisis de emociones por medio de la detección de micro expresiones faciales, donde la columna 0 es enojado, la 1 es disgustado, la 2 es temeroso, la 3 es feliz, la 4 es neutral, la 5 es triste y la 6 es sorprendido.

fácilmente en cámaras de seguridad o cualquier otra, con una resolución aceptable para reconocer estos patrones en un intervalo de tiempo que deberá ser probado, el cuál como resultado detectaría un acto delictivo como el robo.

Además, se puede concluir que, el buen análisis de un algoritmo de clasificación eficiente para la detección de intenciones de robo en personas, el cual por medio del estudio mostrado en este artículo es DBSCAN, en combinación a un conjunto de datos de entrenamiento apropiado, puede presentar resultados interesantes en el campo de la criminología y la psicología para un estudio correspondiente.

Finalmente, el análisis para encontrar una metodología adecuada para la creación de un sistema para la predicción de intenciones de robo en personas ha servido para realizar un sistema de estas capacidades, el cual ya está en fase de desarrollo. Y también se espera que este artículo puede servir de inspiración para que más gente involucrada en el ámbito científico y tecnológico se involucre en este tema, que es importante para el campo de la seguridad para cada individuo y/o institución.

Agradecimientos. Se le agradece al Consejo Nacional de Ciencia y Tecnología (CONACyT) y al Tecnológico Nacional de México Campus Apizaco, por el apoyo brindado para la realización de esta investigación.

Referencias

1. Cook, P. J.: Robbery violence. The journal of criminal law and criminal, vol. 78, no. 2, pp. 357–376 (1987) doi:10.2307/1143453
2. Alvarez, V. M., Velazquez, R., Gutierrez, S., Enriquez-Zarate, J.: A method for facial emotion recognition based on interest points. International Conference on Research in

Intelligent and Computing in Engineering RICE, pp. 1–4 (2018) doi:10.1109/RICE.2018.8509055
3. Ahmad, M. A., Teredesai, A., Eckert, C.: Interpretable machine learning in healthcare. 2018 IEEE International Conference on Healthcare Informatics (ICHI), pp. 447–447 (2018) doi:10.1109/ICHI.2018.00095
4. Shih, C. H., Liu, Yen, C. S., Lee. C. H., Liao, J. H., Jeng, B. S.: A fast shrinking suspicious criminal system from the voice. In: Proceedings IEEE 32nd Annual 1998 International Carnahan Conference on Security Technology, pp. 158–162 (1998) doi:10.1109/CCST.1998.723782
5. Sathyadevan, S., Devan, M. S., Surya Gangadharan, S.: Crime analysis and prediction using data mining. In: First International Conference on Networks & Soft Computing ICNSC´14, pp. 406–412 (2014) doi:10.1109/CNSC.2014.6906719
6. Isah, H., Neagu, D., Trundle, P.: Bipartite network model for inferring hidden ties in crime data. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '15, pp. 994–1001 (2015) doi:10.1145/2808797.2808842
7. Babakura, A., Sulaiman, M. N., Yusuf, M. A.: Improved method of classification algorithms for crime prediction. In: Porceedings of 2014 International Symposium on Biometrics Security Technologies, ISBAST, pp. 250–255 (2015) doi:10.1109/ISBAST.2014.7013130
8. Mahmud, N., Zinnah, K. I., Rahman, Y. A., Ahmed, N.: CRIMECAST: A crime prediction and strategy direction service. In: 19th International Conference on Computer and Information Technology (ICCIT), pp. 414–418 (2017) doi:10.1109/ICCITECHN.2016.7860234
9. Su, Y., Lu, H., Zhang, H.: Qualitative mapping modeling of criminals' sense of security in theft cases. In: Proceedings of IEEE International Conference Big Data, pp. 4337–4341 (2019) doi:10.1109/BigData47090.2019.9005471
10. Fox, B. H., Jennings, W. G., Farrington, D. P.: Bringing psychopathy into developmental and life-course criminology theories and research. Journal of Criminal Justice, vol. 43, no. 4, pp. 274–289 (2015) doi: 10.1016/j.jcrimjus.2015.06.003
11. Ko, K. E., Sim, K. B.: Development of a facial emotion recognition method based on combining AAM with DBN. International Conference on Cyberworlds, pp. 87–91 (2010) doi:10.1109/CW.2010.65
12. Shojaeilangari, S., Yau, W. Y., Nandakumar, K., Li, J., Teoh, E. K.: Robust representation and recognition of facial emotions using extreme sparse learning. IEEE Transactions on Image Processing, vol. 24, no. 7, pp. 2140–2152 (2015) doi:10.1109/TIP.2015.2416634
13. Abanoz, H., Çataltepe, Z.: Emotion recognition on static images using deep transfer learning and ensembling. 26th Signal Processing and Communications Applications Conference, pp. 1–4 (2018) doi: 10.1109/SIU.2018.8404346
14. Zhang, L., Zhang, B.: Non-invasive multi-disease classification via facial image analysis using a convolutional neural network. International Conference on Wavelet Analysis and Pattern Recognition ICWAPR'18, pp. 66–71 (2018) doi:10.1109/ICWAPR.2018.8521262
15. Ricardo, D., Salazar, C., Lanzarini, L.: Aplicación de detección de objetos y algoritmo de Viola Jones para el desarrollo de una base de datos en pacientes de Alzheimer. In: 14th Iberian Conference on Information Systems and Technologies CISTI, pp. 19–22 (2019)
16. Sergeeva, A. D., Savin, A. V., Sablina, V. A., Melnik, O. V.: Emotion recognition from micro-expressions: Search for the face and eyes. In: 8th Mediterranean Conference on Embedded Computing MECO´19, pp. 1–4 (2019) doi:10.1109/MECO.2019.8760029
17. Ray, S.: A Quick review of machine learning algorithms. International Conference on Machine Learning, Big Data, Cloud and Parallel Computing COMITCon´19, pp. 35–39 (2019) doi:10.1109/COMITCon.2019.8862451
18. Fidler Dennis, J., Arnroth, L.: Supervised learning techniques. vol. 18, pp. 1164–1184 (2015)

19. Dalal, K. R.: Analysing the role of supervised and unsupervised machine learning in IoT. International Conference on Electronics and Sustainable Communication Systems ICESC´20, pp. 75–79 (2020) doi:10.1109/ICESC48915.2020.9155761
20. Singh, V. K., Tiwari, N., Garg, S.: Document clustering using K-means, heuristic K-means and fuzzy C-means. In: Proceedings of International Conference on Computational Intelligence and Communication, pp. 297–301 (2011) doi:10.1109/CICN.2011.62
21. Chen, X., Miao, P., Bu, Q.: Image segmentation algorithm based on particle swarm optimization with K-means optimization. In: Proceedings of IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 156–159 (2019) doi:10.110 9/ICPICS47731.2019.8942442
22. Hacker, T.: Cluster analysis for the cloud parallel competitive fitness and parallel K-means for large dataset analysis. In: 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, pp. 177–184 (2012) doi: 10.1109/CloudCom.20 12.6427553
23. Shah, S., Singh, M.: Comparison of a time efficient modified K-mean algorithm with K-mean and K-medoid algorithm. In: 2012 International Conference on Communication Systems and Network Technologies, pp. 435–437 (2012) doi:10.1109/CSNT.2012.100
24. Lan, X., Li, Q., Zheng, Y.: Density K-means: A new algorithm for centers initialization for K-means. In: Proceedings of 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 958–961 (2015) doi:10.1109/ICSESS.2015.7339213
25. Wang, X., Jiao, Y., Fei, S.: Estimation of clusters number and initial centers of K-means algorithm using watershed method. In: 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), pp. 505–508 (2016) doi:10.1109/DCABES.2015.132
26. Qi, J., Yu, Y., Wang, L., Liu, J.: K-means: An effective and efficient k-means clustering algorithm. In: Proceedings of IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications, pp. 242–249 (2016) doi:10.1109/BDCloud-SocialCom-SustainCom.2016.46
27. Shi, G., Gao, B., Zhang, L.: The optimized K-means algorithms for improving randomly-initialed midpoints. In: Proceedings of 2nd International Conference on Measurement, Information and Control, vol. 2, pp. 1212–1216 (2013) doi:10.1109/MIC.2013.6758177
28. Saboori, E., Parsazad, S.: Improving the K-means algorithm using improved downhill simplex search. In: Proceedings of 2nd International Conference on Software Technology and Engineering, vol. 2, pp. V2-350–V32-354 (2010) doi:10.1109/ICSTE.2010.5608792
29. Sebayang, F. A., Lydia, M. S., Nasution, B. B.: Optimization on purity K-means using variant distance measure. In: Proceedings of 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), pp. 143–147 (2020) doi:10.1109/MECnIT48290.2020.9166600
30. Shimizu, T., Sakurai, K.: Comprehensive data tree by actor messaging for incremental hierarchical clustering. In: 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 1, pp. 801–802 (2018) doi:10.1109/COMPSAC.2018.00127
31. Pradip, K. G., Patil, D. R.: Summarization of sentences using fuzzy and hierarchical clustering approach. Symposium on Colossal Data Analysis and Networking (CDAN), pp. 1–7 (2016) doi:10.1109/CDAN.2016.7570964
32. Rong, Y., Liu, Y.: Staged text clustering algorithm based on K-means and hierarchical agglomeration clustering. In: Proceedings of IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 124–127 (2020) doi:10.1109/IC AICA50127.2020.9182394
33. Park, S., Park, Y. B.: Photovoltaic power data analysis using hierarchical clustering. In: International Conference on Information Networking (ICOIN), pp. 727–731 (2018) doi:10.1109/ICOIN.2018.8343214

34. Chen, T. S., Tsai, T. H., Chen, Y. T., Lin, C. C., Chen, R. C., Li, S. Y., Chen, H. Y.: A combined K-means and hierarchical clustering method for improving the clustering efficiency of microarray. In: International Symposium on Intelligent Signal Processing and Communication Systems, pp. 405–408 (2005) doi:/10.1109/ispacs.2005.1595432
35. Zhang, Y., Feng, B.: A co-occurrence based hierarchical method for clustering web search results. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 407–410 (2008) doi:10.1109/WIIAT.2008.35
36. Rahman, M. A., Rahman, M. M., Ul-Husna, A., Haque-Mollah, M. N.: Robust hierarchical clustering for metabolomics data analysis in presence of cell-wise and case-wise outliers. In: Proceedings of International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), pp. 1–4 (2018) doi:10.1109/IC4ME2.2018.8465616
37. Yu, H., Zhang, W.: Stabilizing System. pp. 1297–1301 (2013)
38. Zhu, Q., Tang, X., Liu, Z.: Revised DBSCAN clustering algorithm based on dual grid. Chinese Control And Decision Conference (CCDC), pp. 3461–3466 (2020) doi:10.1109/CCDC49329.2020.9163926
39. Chen, M., Gao, X. D., Li, H. F.: Parallel DBSCAN with priority R-tree. In: Proceedings of International Conference on Information Management and Engineering, no. 3, pp. 508–511 (2010) doi:10.1109/ICIME.2010.5477926
40. Smiti, A., Elouedi, Z.: DBSCAN-GM An improved clustering method based on Gaussian means and DBSCAN techniques. In: IEEE 16th International Conference on Intelligent Engineering Systems (INES), pp. 573–578 (2012) doi:10.1109/INES.2012.6249802
41. Ohadi, N., Kamandi, A., Shabankhah, M., Fatemi, S. M., Hosseini, S. M., Mahmoudi, A.: SW-DBSCAN: A grid-based DBSCAN algorithm for large datasets. In: 6th International Conference on Web Research (ICWR), pp. 139–145 (2020) doi:10.1109/ICWR49608.2020.9122313
42. Çelik, M., Dadaşer-Çelik, F., Dokuz, A. Ş.: Anomaly detection in temperature data using DBSCAN algorithm. In: International Symposium on Innovations in Intelligent Systems and Applications, pp. 91–95 (2011) doi:10.1109/INISTA.2011.5946052

ISSN 1870-4069

Diseño y desarrollo de un robot de servicio con un control de velocidad PID

Rosario Baltazar, Martín Sánchez, Miguel Ángel Casillas,
Liliana Sánchez, Arnulfo Alanis

Tecnológico Nacional de México,
Instituto Tecnológico de León,
México

{rosario.baltazar, 15241180, liliana.sanchez,
miguel.casillas}@leon.tecnm.mx,
alanis@tectijuana.edu.mx

Resumen. En este trabajo se muestra el diseño y desarrollo de un Robot de Servicio con velocidad controlada a través de un Control PID (Proporcional Integral Derivativo) para ser aplicado en terapias para niños con capacidades diferentes o adultos mayores con Alzheimer. El robot se diseñó con base en el microcontrolador ESP 32 y se aprovechó la capacidad de su doble núcleo para que en uno de los núcleos se realizara el control de la rueda izquierda y en el otro núcleo se programó el control de la rueda derecha. El robot también cuenta con diferentes sensores, entre ellos un acelerómetro que permite conocer la orientación del robot y su desplazamiento. Se cuenta con un encoder para conocer la velocidad del robot y lograr el control a través de las revoluciones por minuto RPM. También se incluye un sensor GPS para tener la localización del robot en tiempo real. Finalmente se agregaron 6 sensores ultrasónicos para evitar colisiones. Una de las grandes ventajas es que todo el diseño del hardware y los programas son de elaboración propia, es resultado del trabajo de los miembros de grupo de ambientes inteligentes, esto permitirá en un futuro, agregar más propiedades al robot.

Palabras clave: Control PID, ESP 32, acelerómetro, encoder, GPS.

Design and Development of a Service Robot with PID Velocity Control

Abstract. The design and development of a Service Robot with controlled speed through a PID Control to be applied in therapies for children with different capacities or to the elderly people with Alzheimer's is presented. The robot was designed based on the ESP 32 microcontroller and the capacity of its dual core was used so that the left wheel was controlled in one of the cores and the right wheel control was programmed in the other core, as well. The robot shown in this work also has different sensors, including an accelerometer that allows knowing the orientation of the robot and its displacement. An encoder was implemented to know the speed of the robot and achieve control through the RPM. A GPS sensor is also included to have the location of the robot in real time and thus facilitate its

control. Finally 6 ultrasonic sensors were added to avoid collisions. One of the great advantages is that all the hardware design and the programs are self-made, it is the result of the work of the members of the intelligent environments group, this will allow in the future to add more properties to the robot.

Keywords: PID Control, ESP 32, acelerometer, encoder, GPS.

1. Introducción

Desde hace algún tiempo se ha investigado el efecto que tienen los robots en las personas. Un ejemplo de ello es la investigación acerca del uso de robots de servicio para la asistencia a personas de la tercera edad como lo hizo Ariño [1] en sus tesis.

Paro es un robot con forma de foca que se ha usado para terapias con adultos mayores en el trabajo [2] muestra los resultados de los efectos psicológicos y sociales de la actividad asistida por robot durante un año en personas mayores.

Por otro lado, se ha investigado que los niños autistas son pacientes más receptivos cuando las teorías de aprendizaje o terapias son implementadas vía tecnológica como lo han revelado estudios previos; autores como [3, 4], evidencian la presencia de un claro interés en robots y en dispositivos electrónicos de diversa naturaleza.

Otros autores muestran terapias asistidas por robots para niños con autismo, que buscan provocar un patrón de comportamiento repetitivo y secuencial para desarrollar alguna habilidad y obtener retroalimentación del comportamiento en el infante [5]. Diehl [6], hace un extenso estudio de los robots y su uso en niños con autismo desde el punto de vista clínico.

También se ha desarrollado investigación con respecto a los robots sociales como herramientas para ayudar durante la terapia y la educación de niños con discapacidades intelectuales, sin embargo, hay pocos estudios que muestren el uso por largos periodos de tiempo, sin embargo, en [7] se analizó el impacto de una exposición prolongada a robots de asistencia social en un contexto educativo con estudiantes de secundaria con discapacidad intelectual y autismo. En donde los participantes respondieron positivamente ante el uso de los robots en las escuelas.

Por está razón nace la idea de crear un robot de servicio simple y de bajo costo como primera aproximación al desarrollo de un robot que permita una interacción con niños con capacidades diferentes.

Este robot es para lograr posteriormente agregar más módulos, en los cuales ya se está trabajando por el equipo de investigación, que permitan realizar terapias y otras actividades con los niños, teniendo un código propio y un hardware propio.

Este trabajo se centró principalmente en el diseño del control de la velocidad de los motores del robot, esto debido a que la velocidad de un robot de servicio o de un robot que interactúa con niños debe ser adecuada, si la velocidad es muy alta puede lastimar a los niños o personas que están interactuando con el robot. Además se están aprovechando las características del doble núcleo del ESP 32, para que la base del control recaiga en un dispositivo de muy bajo costo y fácil programación.

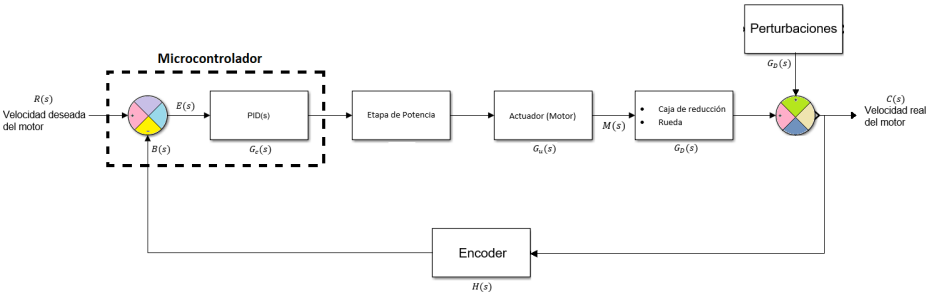


Fig. 1. Sistema de control de lazo cerrado para la velocidad del motor A.

2. Diseño del robot

En esta sección se muestran los componentes del diseño del robot y sus características.

2.1. Sensores

Para lograr que el robot tenga una cierta autonomía para su desplazamiento, se requieren diferentes sensores que permitan conocer su ubicación en tiempo real, por lo que se agregaron los siguientes componentes:

Acelerómetro. Este sensor es capaz de medir la vibración y aceleración de una estructura, en sus tres ejes conocidos en inglés como Roll, Pitch y Yaw. Este sensor es necesario para conocer la orientación del robot, es decir, si se desplaza o incluso si se cae.

Se utilizó como acelerómetro el circuito MPU-6050, este módulo tiene también un giroscopio de tres ejes con el que podemos medir velocidad angular y un acelerómetro también de 3 ejes con el que medimos los componentes x, y y z de la aceleración.

Para programarlo se definieron como variables la aceleración en g y la aceleración angular en °/s. Se obtuvieron los valores raw, se trabajo en el escalamiento de las lecturas y se aplicó un filtro. Finalmente se probó en conjunto con el ESP 32 físicamente.

Encoder. Se utilizó un encoder rotativo incremental óptico para calcular las rpm, el cual se acopló al motor y a las ruedas por medio de una banda dentada. El encoder se encarga de enviar las pulsaciones generadas por las ruedas al microcontrolador a través de interrupciones para calcular las RPM. Esto se programó para comprobar que el motor giraba exactamente 30 vueltas emitiendo 64 pulsos por cada revolución y cada pulso enviado por el encoder se efectuaba aproximadamente cada 5.6°.

Sensor de Posición GPS. En el robot se utilizó un Sistema de Posicionamiento Global (GPS; en inglés, Global Positioning System), modelo NEO-6. Estos elementos disponen de interface de comunicación UART Universal Asynchronous Receiver-Transmitter, en español:

Transmisor-Receptor Asíncrono Universal, SPI del inglés Serial Peripheral Interface o en español bus de interfaz de periféricos, Inter-Integrated Circuit (I2C) y USB (en inglés: Universal Serial Bus) o Bus Universal en Serie.

```
/*Declaración de variables PID*/
unsigned long lastTime;
double Input, Output, Setpoint;
double errSum, lastErr;
double Kp = 1, Ki = 0.6, Kd = 0.08;
int SampleTime = 1000;
double error = 0;
/*Variables para calculo de RPM*/
int in1 = 32;
int in2 = 33;
int motor_ = 16;
volatile int motor = 0;
volatile int rpm = 0;
#define channel      0
#define frecuencia   10000
#define resolucion   8
unsigned long tAnterior;
```

Fig. 2. Código para la declaración de variables.

```
void loop() {
  ledcWrite (channel, Output);          //Función
       cycle
  delay(1000);
  rpm = (motor*60/64)*1000/(millis()-tAnterior);
  tAnterior = millis();                 //Se igua
       cada segundo
  Input = (double)rpm;                  //Igualam
  unsigned long now = millis();         //se Igua
  double timeChange = double (now - lastTime);//T
  /*Calcula todos los errores*/
  double error = Setpoint - Input;
  errSum += (error * timeChange);
  double dErr = (error - lastErr) / timeChange;
  /*Calculamos la función de salida del PID*/
  Output = Kp * error + Ki * errSum + Kd * dErr;
  /*Guardamos el valor de algunas variables para
  lastErr = error;                      //Igualam
  lastTime = now;                       //Igualam
  motor = 0;                            //Se rein
}
/*Función de la interripción*/
void motor_sen()
{
  motor++;                              //Sumator
}
```

Fig. 3. Código para el cálculo del error del control PID en el ESP32.

Soportan los protocolos NMEA (National Marine Electronics Association) y RTCM que son protocolos de Internet. Además, se proporciona la localización, la fecha, la velocidad y la latitud entre otros parámetros. Se realizó la conexión para que los parámetros de ubicación se enviaran a la plataforma de IoT para transformar los datos recibidos en coordenadas en un mapa y así tener la localización del robot en tiempo real.

Sensor ultrasónico. Finalmente se usaron seis sensores ultrasónicos HC-SR04, con la finalidad de detectar obstáculos y conocer la distancia hacia ellos, se agregó un led rojo para avisar que el robot tiene un obstáculo a menos de 30 cm.

Actuadores. Este modelo del robot solo se desplaza en el plano, de tal manera que se le acoplaron un par de llantas a los motores, así mismo se incorporó una rueda que no tiene ningún control (rueda loca).

Para mover las ruedas se seleccionaron motores de corriente continua y se realizó el cálculo para conocer el par dependiendo del peso que iba a cargar nuestro robot, después de realizar el cálculo se determinó el uso de dos motoreductores metálicos 37D57L. Para lograr la conexión de estos motores con el microcontrolador se utilizaron dos puentes H de 2A.

3. Control

3.1. Controlador PID básico

Para lograr una velocidad adecuada en el desplazamiento del robot se desarrollo el diseño de un controlador PID básico como primera opción, capaz de controlar la velocidad del motor y evitar las variaciones y compensar las perturbaciones en el sistema como se muestra en la figura 1. El sistema de lazo cerrado de la figura 1 se compone de los siguientes elementos:



Fig. 4. Derivative kick. Recuperado de: [8].

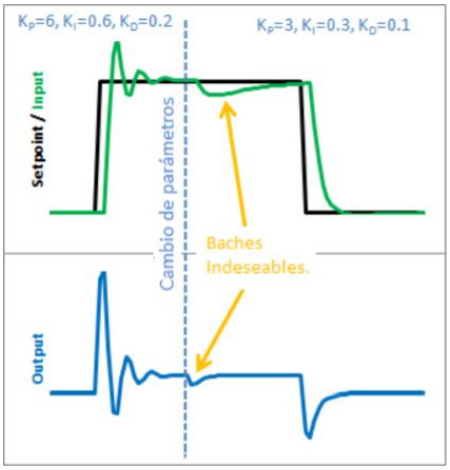


Fig. 5. Cambio de sintonización. Recuperado de: [8].

- $R(s)$ - Variable de entrada.
- $E(s)$ - Error.
- $B(s)$ - Valor medido por el sensor.
- $G_C(s)$ - Controlador.
- $G_v(s)$ - Elemento final de control.
- $M(s)$ - Variable manipulada.
- $G_P(s)$ - Planta.
- $G_D(s)$ - Perturbaciones.
- $H(s)$ - Sensor.
- $C(s)$ - Variable controlada.

Cada una de estas partes del sistema de control de lazo cerrado se denomina por una variable ya definida por la teoría de control. Este sistema de control comienza cuando se ingresa una variable de entrada $R(s)$ al sistema, en este caso la variable de entrada es "velocidad deseada del motor", después se hace un cálculo del error el cual se calcula de la siguiente manera: $E(s) = R(s) - B(s)$, es decir, el valor deseado menos el valor medido por el sensor $H(s)$, después de calcular el error, la señal entra al bloque del controlador $G_C(s)$, en este caso un controlador PID, la señal de salida del controlador PID ingresa por el bloque de la etapa de potencia para después ingresar al bloque del actuador $G_v(s)$ para reestablecer la variable controlada en su valor de referencia, después de esto la variable manipulada $M(s)$ entra en un punto de suma y se suma con las perturbaciones externas del sistema $G_D(s)$, al final del sistema de control se encuentra la variable controlada o de salida $C(s)$, la cual se retroalimenta hacia el sensor para calcular nuevamente el error del sistema.

```
30 void loop() {
31   ledcWrite (channel, Output);          //Función
         cycle
32   delay(1000);
33   rpm = (motor*60/64)*1000/(millis()-tAnterior);
34   tAnterior = millis();                 //Se igua
         cada segundo
35   Input = (double)rpm;                  //Igualam
36   unsigned long now = millis();         //se Igua
37   double timeChange = double (now - lastTime);//T
38   /*Calcula todos los errores*/
39   double error = Setpoint - Input;
40   errSum += (error * timeChange);
41   double dErr = (error - lastErr) / timeChange;
42   /*Calculamos la función de salida del PID*/
43   Output = Kp * error + Ki * errSum + Kd * dErr;
44   /*Guardamos el valor de algunas variables para
45   lastErr = error;                      //Igualam
46   lastTime = now;                       //Igualam
47   motor = 0;                            //Se rein
48 }
49 /*Función de la interripción*/
50 void motor_sen()
51 {
52   motor++;                              //Sumator
53 }
```

Fig. 6. Código para el cálculo del error del control PID con tiempos regulares en el ESP32.

```
52 /*Función para los tunnings de los valores KP, KI, KD*/
53 void SetTunings(double Kp_, double Ki_, double Kd_)
54 {
55   double SampleTimeInSec = ((double)SampleTime) / 1000;
56   Kp = Kp_;
57   Ki = Ki_ * SampleTimeInSec;
58   Kd = Kd_ / SampleTimeInSec;
59 }
60 /*Función para colocar el tiempo de muestreo*///Función que te
61 //permite cambiar a un nuevo tiempo de muestreo mientras el program
62 void SetSampleTime(int NewSampleTime)
63 {
64   if (NewSampleTime > 0)
65   {
66
67     double ratio = (double)NewSampleTime / (double)SampleTime;
68     Ki *= ratio;
69     Kd /= ratio;
70     SampleTime = (unsigned long)NewSampleTime;
71   }
72 }
73 /*Función de la interrupción*/
74 void motor_sen()
75 {
76   motor++;
77 }
```

Fig. 7. Código para el cálculo del error del control PID con tiempos regulares segunda parte en el ESP32.

De la documentación existente sobre sistemas de control, podemos destacar la ecuación 1, esta ecuación muestra la representación matemática de los 3 controladores que componen al controlador PID:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t)dt + K_p T_d \frac{d\,e(t)}{dt}, \tag{1}$$

donde u(t) es la salida del controlador y entrada de control al proceso, e(t) es el error de salida, K_p es la ganancia proporcional, T_i es la constante de tiempo integral y T_d es la constante de tiempo derivativa.

3.2. Código de programación del controlador PID básico

Este controlador PID se programó en el IDE de Arduino y se bajó al microcontrolador ESP 32. En el ESP 32 se tienen dos núcleos y en el robot se tienen dos ruedas, entonces se hizo uso de esa propiedad y se hizo un controlador en cada uno de los núcleos.

El programa se presenta a continuación y muestra la programación de un controlador PID básico, en el cual primeramente se definen las variables para el controlador, para el PWM y para las interrupciones del encoder.

Las revoluciones por minuto (RPM) son calculadas por medio de una ecuación que se programó en el código y que incluye los pulsos generados, los pulsos por revolución del encoder y algunas variables del tiempo.

Como primer código de programación del controlador PID tenemos en la siguiente imagen 2 el código para la declaración de variables: Posteriormente, el programa genera los cálculos para determinar los errores y la salida del controlador PID.

```
void loop() {
  ledcWrite (channel, Output);  //Función para escribir el PWM la cual
  unsigned long now = millis(); //se iguala la variable now(actual) a
  int timeChange = (now - lastTime);//Tiempo de cambio entre cálculos
  //Determina si hay que ejecutar el PID o retornar de la función//
  if (timeChange>= SampleTime)
  {
    rpm = (motor*60/64)*1000/(millis()-tAnterior);  //60 = 60s(1min),
    tAnterior = millis();             //Se iguala tiempo anterior a millis
    Input = (double)rpm;              //Igualamos las RPM de salida a la e
    /*Calcula todos los errores*/
    double error = Setpoint - Input;
    errSum += error;
    double dInput = (Input - lastInput);//Resta del Input actual menos
    /*Calculamos la función de salida del PID*/
    Output = Kp * error + Ki * errSum - Kd * dInput;
    /*Guardamos el valor de algunas variables para el próximo recalcul
    lastInput = Input;                //Igualamos el Input anterior al Input
    lastTime = now;                   //Igualamos el último tiempo de cálcul
    motor = 0;                        //Se reinicia el contador de los pulso
  }
}
/*Función para los tunnings de los valores KP, KI, KD*/
void SetTunings(double Kp_, double Ki_, double Kd_)
{
  double SampleTimeInSec = ((double)SampleTime) / 1000;
  Kp = Kp_;
  Ki = Ki_ * SampleTimeInSec;
  Kd = Kd_ / SampleTimeInSec;
}
```

Fig. 8. Código para el cálculo del error del control PID con derivative kick en el ESP32.

```
/*Función para colocar el tiempo de muestreo*/

//Función que permite cambiar al nuevo tiempo de muestreo mientras
void SetSampleTime(int NewSampleTime)
{
  if (NewSampleTime > 0)
  {

    double ratio = (double)NewSampleTime / (double)SampleTime;
    Ki *= ratio;
    Kd /= ratio;
    SampleTime = (unsigned long)NewSampleTime;
  }
}
/*Función de la interrupción*/
void motor_sen()
{
  motor++;                          //Sumatoria de los pulsos
}
```

Fig. 9. Código para el cálculo del error del control PID con derivative kick segunda parte en el ESP32.

Por último, la salida del controlador la ingresa en la función **ledcWrite()** para generar la señal PWM del motor. Esto se repite para la otra rueda. En la figura 3 está el código para el cálculo del error del controlador PID.

3.3. Código de programación del controlador PID con tiempos regulares

Para la segunda versión del código de programación del controlador PID se aseguró que la función que ejecuta el PID lo haga regularmente.

Se basa en un tiempo de ejecución predeterminado, el PID decide si debe hacer cálculos o retornar a la función. Una vez que el PID se ejecuta a intervalos regulares, los cálculos correspondientes a la parte de la derivada e integral se simplifican.

A continuación, se muestra la versión 2 del código de programación del controlador PID. Figura 6 y 7. En el código mostrado en la Figura 7 se puede observar como varía el error actual tomando como base el error anterior y como se modifica el tiempo de muestreo. La función que se encarga de calcular el PID será evaluada a tiempos regulares y el tiempo de muestreo es constante.

3.4. Código de programación del controlador PID eliminando el fenómeno "Derivative Kick"

La modificación que se presenta a continuación cambiará levemente el termino derivativo con el objetivo de eliminar el fenómeno "Derivative Kick", el cual consiste en sobrepicos que se generan en la señal de salida.

Este fenómeno, se produce por variaciones rápidas en la señal de referencia que es del tipo función escalón, debido a la presencia del término derivativo en la acción de control, la señal de control $u(t)$ contendrá una función impulso ya que, en el momento el cambio finito de la referencia, la derivada del error se hace infinita.

```
void loop() {
  ledcWrite (channel, Output);//Función para escribir el PWM
  unsigned long now = millis(); //se Iguala la variable now(a
  int timeChange = (now - lastTime);//Tiempo de cambio entre
  //Determina si hay que ejecutar el PID o retornar de la fun
  if (timeChange >= SampleTime)
  {
     rpm = (motor*60/64)*1000/(millis()-tAnterior);  //60 = 6
     tAnterior = millis();      //Se iguala el tiempo anterior
          segundo
     Input = (double)rpm;       //Igualamos las RPM de salida a
     /*Calcula todos los errores*/
     double error = Setpoint - Input;
     ITerm += (Ki * error);
     double dInput = (Input - lastInput);
     /*Calculamos la función de salida del PID*/
     Output = Kp * error + ITerm - Kd * dInput;
     /*Guardamos el valor de algunas variables para el próximo
     lastInput = Input;         //Igualamos el Input anterior al
     lastTime = now;            //Igualamos el último tiempo de c
     motor = 0;                 //Se reinicia el contador de los
  }
}
```

Fig. 10. Código para el cálculo del error del control PID eliminando cambios de sintonización en el ESP32.

```
/*Función para los tunnings de los valores KP, KI, KD*/
void SetTunings(double Kp_, double Ki_, double Kd_)
{
   double SampleTimeInSec = ((double)SampleTime) / 1000;
   Kp = Kp_;
   Ki = Ki_ * SampleTimeInSec;
   Kd = Kd_ / SampleTimeInSec;
}
/*Función para colocar el tiempo de muestreo*///Función que te
//permite cambiar a un nuevo tiempo de muestreo mientras el progr
void SetSampleTime(int NewSampleTime)
{
   if (NewSampleTime > 0)
   {
      double ratio = (double)NewSampleTime / (double)SampleTime;
      Ki *= ratio;
      Kd /= ratio;
      SampleTime = (unsigned long)NewSampleTime;
   }
}
/*Función de la interrupción*/
void motor_sen()
{
   motor++;                    //Sumatoria de los pulsos
}
```

Fig. 11. Código para el cálculo del error del control PID eliminando cambios de sintonización en el ESP32 segunda parte.

La Figura 4 ilustra el problema, siendo el $\text{error} = \text{setpoint} - \text{entrada}$, cualquier cambio en el error. Esto provoca que se visualice un sobrepico que se puede distinguir fácilmente en la salida y que se puede corregir de una manera sencilla:

$$\frac{d\,\text{Error}}{dt} = \frac{d\,\text{Setpoint}}{dt} - \frac{d\,\text{Input}}{dt}, \tag{2}$$

$$\frac{d\,\text{Error}}{dt} = -\frac{d\,\text{Input}}{dt}. \tag{3}$$

Como podemos observar en las Ecuaciones (2) y (3) la derivada del error es igual a la derivada negativa de la entrada en situaciones donde el setpoint es constante, excepto cuando el setpoint está cambiando, esto acaba siendo de gran ayuda para mejorar el código de programación. Ahora en lugar de añadir $Kd \times \text{error derivado}$, se le restará $Kd \times \text{valor de entrada derivado}$.

Esto se conoce como el uso de “derivada de la medición”. A continuación, se muestra el código de programación. En este código solo se reemplazó la derivada positiva del error con la derivada negativa de la entrada y se sustituye el último valor del error por el último valor que tomó la entrada (lastInput).

3.5. Código de programación del controlador PID eliminando cambios de sintonización

Se puede cambiar la sintonización del PID, mientras el sistema se encuentra en funcionamiento por lo que la señal de salida puede tener la siguiente forma. Como se puede observar en la Figura 5 el cambio de valores de sintonización afecta a nuestro sistema PID, el culpable del bache de la Figura 5 es el término integral; es el único

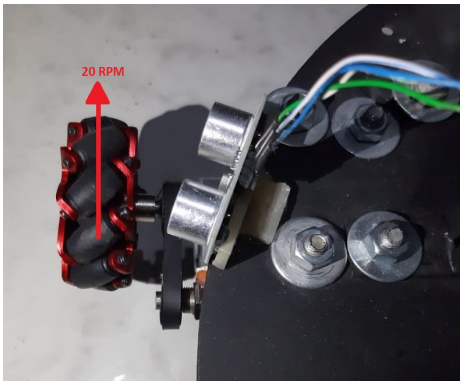


Fig. 12. Rueda izquierda.

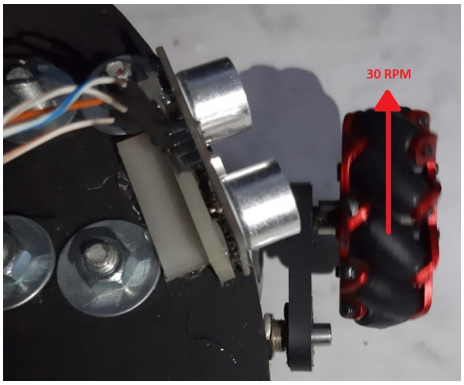


Fig. 13. Rueda derecha.

valor que cambia drásticamente la señal de salida cuando se reajusta el parámetro K_I. Esto sucede debido a la interpretación de la integral:

$$K_I \int e\, dt \approx K_{I_n} \left[e_n + e_{n-1} + ... \right]. \tag{4}$$

La interpretación de 4 funciona muy bien hasta que K_I cambia. Supongamos que en un tiempo X al operario se le ocurre cambiar el valor de K_I, lo que pasaría sería que la suma de todos los errores se multiplicaría con el valor de K_I, esto no es lo que se necesita.

Nosotros solo queremos que al cambiar de sintonización solo afecte a los valores que estén por delante. La solución a este error consiste en reescalar la suma del error, doblando el valor de K_I o cortando la suma de los errores a la mitad. esto quita el bache del término integral, solucionando el problema:

$$K_I \int edt = \int K_I e\, dt, \tag{5}$$

$$K_I \int edt \approx K_{I_n} e_n + K_{I_{n-1}} e_{n-1} + ... \tag{6}$$

En lugar de tener el término K_I fuera de la integral, lo introducimos dentro del cálculo. Hasta este punto no se ha realizado nada extraño, pero en la práctica está acción resulta en una gran diferencia en la función de salida del PID. Ahora se toma el error y se multiplica por el valor de K_I en ese momento, luego se almacena la suma de los diferentes errores multiplicados por la constante K_I.

Esto resulta en una función de salida, suave y sin sobresaltos, con la ventaja de no tener que utilizar matemática adicional para ello. A continuación se muestra el código de programación para eliminar efectos de cambios de sintonización.

4. Resultados

4.1. Resultados de la conexión del GPS

Los resultados del robot se mostrarán por partes, primero se muestra el resultado de la conexión del GPS, se verifica que el GPS establezca la conexión con los satélites, luego se comienzan a recibir datos y se envían al microcontrolador y este enviaba a su vez cada uno de los datos a la plataforma de Ubidots, entonces el microcontrolador envía una alerta por medio de un led verde y lo enciende por 500ms indicando que el envío de datos había sido un éxito.

Cuando los datos se envían satisfactoriamente, el sistema despliega la información por el monitor serial proporcionando fecha, hora, localización, el número de satélites encontrados y adicionalmente la altitud.

Cabe destacar que este tipo de GPS no solo puede brindar estos datos sino también tiene posibilidad de enviar la velocidad en nudos, la rotación en grados y la variación magnética en grados, pero estos datos se omitieron ya que solo se requieren los datos más elementales.

Ya creado y configurado el mapa en la página de Ubidots, cada vez que la página recibe un valor proveniente del microcontrolador, el mapa marca una localización aproximada de donde se encuentra el robot.

4.2. Resultados del control de velocidad

Cálculo de las revoluciones del motor DC. Como en el monitor serial del IDE de Arduino se registró una velocidad aproximada de 10 RPM, esta velocidad se calcula por medio de la ecuación en el código del cálculo de revoluciones del motor y adicionalmente también se utilizan las interrupciones que el encoder óptico emite cada vez que el motor gira.

La velocidad programada para la rueda derecha fue de 30 RPM y para la rueda izquierda de 20 RPM, en la Figura 12 y Figura 13 se muestran las llantas izquierda y derecha respectivamente con su velocidad de cada una.

Esta claro que no se puede observar el movimiento del robot, por eso en las figuras anteriores se muestra una flecha con la dirección del movimiento de las llantas y la magnitud de la velocidad. La velocidad del motor se encuentra en un intervalo de 9-11 RPM.

Este programa funciona perfectamente cuando sólo se quiere mantener el motor a una velocidad dada pero como la aplicación en este proyecto requiere que el robot tenga un control de la velocidad, para ello fue necesario la implementación de un PID que controlara la velocidad del motor.

4.3. Resultados del controlador PID básico

Después de poner en marcha el controlador PID Básico se revisó el funcionamiento pero seguía teniendo algunas variaciones en la salida, En la Figura 14 se puede observar las RPM de salida del motor utilizando la primera versión del controlador PID.

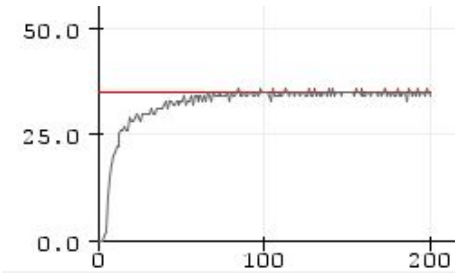


Fig. 14. Velocidad de salida utilizando el controlador PID Básico (PID_V1).

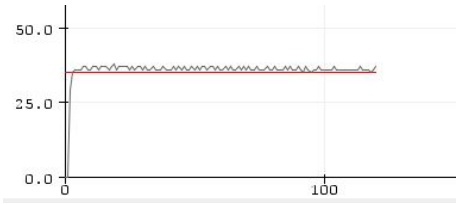


Fig. 15. Velocidad de salida utilizando el controlador PID (PID_V2).

Como se puede observar en la Figura 14, la señal de color negro representa las RPM de salida y estas se mantenían casi iguales al Setpoint (señal roja) el cual se programó de 35 RPM, el sistema se mantenía muy estable ya que no había variaciones muy grandes en la señal de salida de las RPM, pero seguía habiendo la problemática de los intervalos regulares para simplificar los cálculos.

4.4. Resultados del controlador PID con tiempos regulares

Al igual que la versión 1 del controlador PID en esta versión del controlador PID con tiempos regulares se tiene una salida en RPM del motor que se puede obtener a través del Arduino y la cual se puede observar en la Figura 15. Como se puede observar en la Figura 15, la señal de salida de la velocidad en RPM se comporta cada vez mejor, esto se debe a las modificaciones que se le hicieron al primer programa del controlador PID.

En esta figura la señal de salida de la velocidad se mantiene por debajo de la señal Setpoint pero conforme transcurre el tiempo la señal se va sobreponiendo a la señal Setpoint, lo cual refleja que las modificaciones del programa mejoraron el control de la velocidad.

4.5. Resultados del controlador PID eliminando fenómeno Derivative Kick

La versión 3 del controlador PID se encarga de eliminar el fenómeno Derivative Kick. En la Figura 16 se observa que el sistema responde cada vez mejor si se compara con la señal de la Figura 14, ya que la señal de salida de las RPM es más estable y se sobrepone sobre la señal Setpoint por lo tanto, cada vez más se tiene una versión más adecuada del control de la velocidad del robot.

4.6. Resultados del controlador PID eliminando cambios de sintonización

Para visualizar la señal de salida de la versión 4 del controlador PID de velocidad se observa la Figura 17, en ella se muestra la señal de salida de las RPM del motor DC.

Como se observa en la Figura 17, la señal de salida se comporta cada vez más estable ya que, aunque se tienen algunas pequeñas perturbaciones en la señal de salida, el sistema se estabiliza en 35 RPM desde que el controlador se pone en marcha.



Fig. 16. Velocidad de salida utilizando el controlador PID (PID_V3).



Fig. 17. Velocidad de salida utilizando el controlador PID (PID_V4).

4.7. Pruebas del GPS y el acelerómetro

Después de que se desarrolló la placa PCB, se procedió a cargar el programa donde se envían los datos de la localización obtenidos por el GPS y los datos de rotación del robot de servicio.

El programa se encarga de desplegar los datos de los puertos de transmisión y recepción, la dirección IP del internet y los datos proporcionados por el GPS (Satélites, Fecha, Hora, Latitud, Longitud), los datos de Roll y Pitch fueron proporcionados por el acelerómetro.

Cuando se despliega la información anterior el programa esperaba la conexión del GPS, esto debido a que duraba un tiempo en conectarse a los satélites para obtener la localización, después de establecer la conexión con el microcontrolador, el programa establecía conexión con Ubidots y comenzaba a enviar datos cada minuto.

Después de realizar todas las conexiones necesarias para los motores, los sensores ultrasónicos y la PCB procedimos a comprobar el funcionamiento del controlador PID con los sensores ultrasónicos.

Primeramente, se probó la placa PCB para verificar su funcionamiento con el controlador PID, en la Figura 18 se muestra la placa en funcionamiento junto con los módulos L298N y la placa del GPS.

Finalmente se pretende realizar el marketing de este robot. En este momento todos y cada uno de los sectores han tenido que evolucionar con la actualización del marketing al marketing digital, han surgido redes sociales, aplicaciones móviles, que han permitido a muchos organismos posicionar sus productos o servicios con grandes beneficios económicos.

En el giro dela Robótica no es la excepción, esta inventiva: “Robot de servicio con control de velocidad PID”, ha de introducirse también en este mundo comercial virtual, con el fin de cubrir algunas necesidades de personas con capacidades diferentes y adultos mayores, quienes deben ser cuidadosamente tratados.

5. Conclusiones

Se presentó el diseño y desarrollo de un robot de servicio para el cual se desarrollaron los cálculos para lograr el control de cada una de las ruedas. Se agregaron sensores para detección de obstáculos, de la ubicación y para la detección de giro o aceleración.

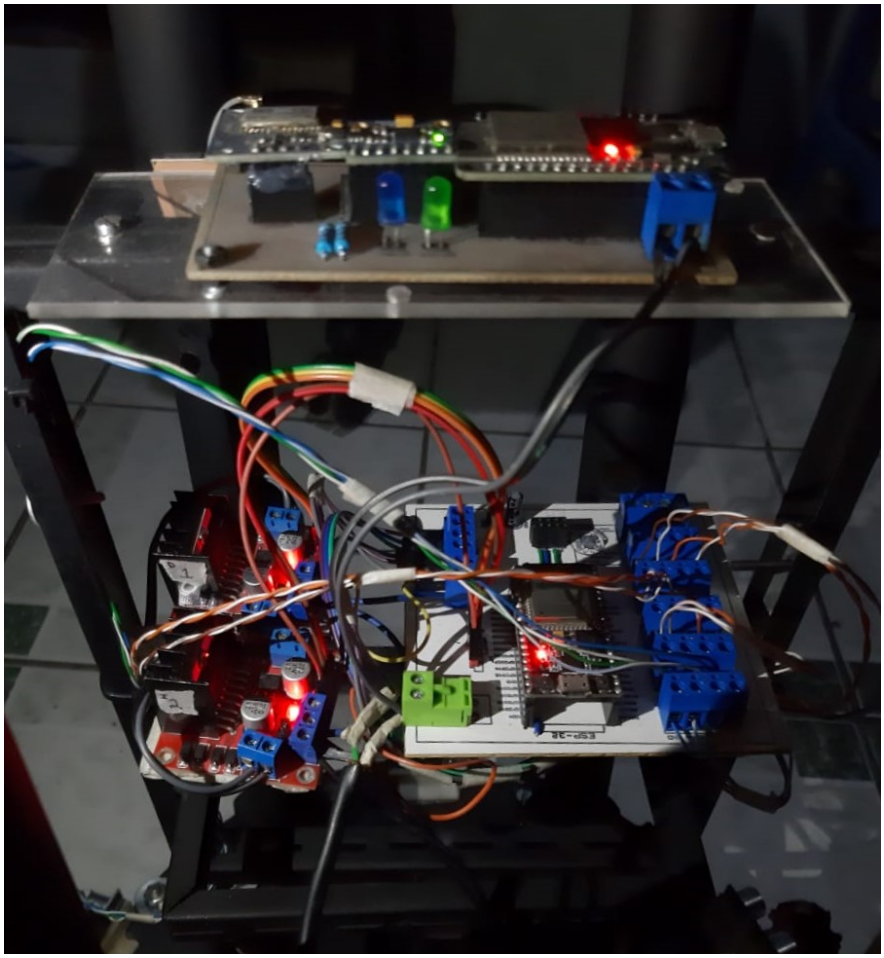


Fig. 18. Placa PCB del controlador PID en funcionamiento.

Se codificaron los encoder para el cálculo de las vueltas y esto convertirlo al control de velocidad. Se diseñó el control de la velocidad por medio de un PID digital que se programó en cada uno de los núcleos del ESP y se muestran los resultados.

La importancia de dar a conocer este producto de innovación tecnológica en esta era digital es que posibilite al segmento de personas con capacidades diferentes y adultos mayores, en cuanto a obtener una herramienta didáctica que promueva desarrollar habilidades y actitudes para mejorar su calidad de vida. Es por ello que ante ésta revolución digital se diseñan e implementan estrategias en medios digitales para su logro.

Agradecimientos. Los autores agradecen al TecNM por el apoyo al Proyecto Integración de agentes inteligentes, asistentes inteligentes e interfaces BCI para la atención de personas con capacidades diferentes.

Referencias

1. Ariño, J.: Desarrollo de un robot de servicio para la asistencia a personas de la tercer edad. Doctoral dissertation, Universitat Politècnica de València (2014)
2. Wada, K., Shibata, T., Saito, T., Sakamoto, K., Tanie, K.: Psychological and social effects of one year robot assisted activity on elderly people at a health service facility for the aged. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 2785–2790 (2005) doi: 10.1109/ROBOT.2005.1570535
3. Dautenhan, K.: Robots as social actors: Aurora and the case of autism. In: Proceedings of the Third International Cognitive Technology Conference, vol. 359, pp. 374 (1999)
4. Robins, B., Dautenhahn, K., Te Boerkhorst, R., Billard, A.: Robots as assistive technology - does appearance matter? In: 13th IEEE International Workshop on Robot and Human Interactive Communication, pp. 277–282 (2004) doi: 10.1109/ROMAN.2004.1374773
5. Ravindra, P., De Silva, S., Tadano, K., Saito, A., Lambacher, S. G., Higashi, M.: Therapeutic-assisted robot for children with autism. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3561–3567 (2009) doi: 10.1109/IROS.2009.5354653
6. Diehl, J. J., Schmitt, L. M., Villano, M., Crowell, C. R.: The clinical use of robots for individuals with autism spectrum disorders: A critical review. Research in Autism Spectrum Disorders, vol. 6, no. 1, pp. 249–262 (2012) doi: 10.1016/j.rasd.2011.05.006
7. Silvera-Tawil, D., Yates, C. R: Socially-assistive robots to enhance learning for secondary students with intellectual disabilities and autism. In: 27th IEEE International Symposium on Robot and Human Interactive Communication, pp. 838–843 (2018) doi: 10.1109/ROMAN.2018.8525743
8. Beauregard, B.: Improving the beginner's pid–introduction (2011) http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/comment-page-1/comments

ISSN 1870-4069

Peligros y ventajas del diseño de instrumentos de medición auxiliado con estrategias de selección de atributos automáticas

Mario E. Marin, Julio C. Ponce, Ángel E. Muñoz, Alejandro Padilla

Universidad Autónoma de Aguascalientes,
Centro de Ciencias Básicas, Aguascalientes,
México

marin@ia.org.mx,{jcponce,aemz,apadilla}@correo.uaa.mx

Resumen. Se presenta una estrategia para ayudar en el diseño de instrumentos de medición en Ciencias Sociales por medio de selección de atributos automática utilizando eliminación recursiva de atributos con validación cruzada y comparándola con el uso de filtrado por correlación y por pruebas univariadas, regresión paso a paso, y eliminación recursiva. Para esto se usan conjuntos de datos producidos por instrumentos de medición prediseñados con rigor metodológico y estadístico, permitiendo generar un nuevo instrumento de medición con menor número de variables, pero conservando el poder predictivo con respecto a alguna variable de interés. Se presentan los peligros y ventajas de este acercamiento, así como un ejemplo utilizando microdatos de la Encuesta Nacional de Uso de Tiempo de INEGI, 2014 y 2019, para generar un nuevo instrumento con menos variables que los conjuntos de microdatos de estas encuestas, pero con poder predictivo mayor con respecto a variables de satisfacción en el uso del tiempo en ámbitos académicos y laborales.

Palabras clave: Aprendizaje máquina, selección de atributos, selección de modelo automatizada, instrumentos de medición, bienestar, uso de tiempo.

Dangers and Advantages of Design of Measurement Instruments Using Automated Feature Selection Tools

Abstract. A strategy is presented to aid in the design of measurement instruments in Social Sciences by means of automated attribute selection using recursive feature elimination with cross-validation and comparing it to other methods such as filter feature selection using correlations and univariate tests, stepwise regression, and recursive elimination. For this, datasets generated by pre-designed instruments with methodological and statistical rigor are used, allowing for the generation of a new measurement instruments with fewer variables that preserve predictive power for some variable of interest. The dangers and advantages of this approach are presented, as well as an example using the National Time Use Survey of INEGI microdata, in its 2014 and 2019 editions, to generate a new instrument with fewer variables than the microdata sets of these surveys, but with better predictive power for variables of satisfaction in the use of time in academic and work settings.

Keywords: Machine learning, attribute selection, automatic model selection, measurement instruments, wellbeing, time use.

1. Introducción

Prácticamente todos los conjuntos de datos utilizados en el contexto de Datos Masivos (*Big Data*) y aprendizaje máquina son obtenidos mediante instrumentos de medición, que son herramientas que permiten asignar numerales a objetos o eventos de acuerdo a ciertas reglas [1], con mayor o menor grado de diseño metodológico. El diseño de instrumentos de medición es una importante disciplina dentro de la práctica de la investigación científica en Ciencias Sociales.

Es un proceso que no puede ser resuelto de manera sencilla y sin un seguimiento riguroso de pasos en una metodología de la investigación asignando arbitrariamente números a fenómenos [2]. Generalmente se recomienda tener en mente un objetivo de estudio antes de comenzar siquiera el primer paso de decidir qué tipo de instrumento de medición se va a diseñar [3], incluso cuando ya se cuentan con datos e indicadores relevantes generados por instrumentos de medición preexistentes [2]. El proceso de diseño de instrumentos de medición se apega al proceso de aplicar el método científico en donde la formulación de las hipótesis precede al diseño de mediciones, experimentos y predicciones.

Por otro lado, el desarrollo de la analítica de *Big Data* y la minería de datos ha creado la posibilidad de generar y analizar grandes cantidades de datos que pueden ser preexistentes a las hipótesis. En la práctica, muchos investigadores y analistas que ejercen la minería de datos proponen hipótesis basadas en patrones encontrados en datos preexistentes a estas o incluso generados en tiempo real [4] y entrenan herramientas de aprendizaje máquina, que juegan el rol de conocimiento inferido [5], para predecir variables en base a datos fuera de la muestra original, y a veces generados por distintos instrumentos de medición.

Existe controversia al respecto de si este tipo de labor investigativa tiene el mismo nivel de validez que con el enfoque mencionado en el párrafo anterior [4], o si este es un enfoque más ingenieril y aplicado. Sin embargo, existen prácticas como la validación de modelos [6] y el preregistro de hipótesis [7] que ayudan en este respecto, y ciertamente las prácticas de analítica de *Big Data* y la minería de datos se han incorporado a la labor investigativa [4], [5], por lo que es importante dar cara tanto a sus ventajas como a sus peligros.

El objetivo del presente trabajo se enfoca, en el cruce de las áreas de minería de datos e investigación, en averiguar si es posible utilizar un proceso de selección de modelo automatizado mediante el análisis de conjuntos de datos masivos preexistentes para auxiliar en el diseño de instrumentos de medición con rigor metodológico. Esto valiéndose de técnicas automatizadas de selección de atributos para así facilitar la creación de instrumentos de medición que sirvan para recopilar nuevos conjuntos de datos, de menor número de dimensiones que el original, pero con similar o mayor poder predictivo respecto a una variable de interés.

El presente texto se organiza así: después de esta introducción en donde se presentan temas, conceptos y objetivos; la sección 2 trata acerca de la definición, objetivos, y tipos de instrumentos de medición y de estrategias de selección de atributos. La sección 3 presenta las desventajas y peligros de utilizar enfoques automáticos de selección de atributos para generar instrumentos de medición basados en conjuntos de datos de alta dimensionalidad que produzcan conjuntos de datos de menor dimensionalidad. La sección 4 propone un proceso de 6 pasos con 6 condiciones que deben ser cumplidas

para evitar la mayor parte del riesgo y desventajas presentadas en la sección anterior. La sección 5 presenta un problema concreto basado en los microdatos de dos ediciones de la Encuesta Nacional de Uso de Tiempo (ENUT) de INEGI. La sección 6 aplica el proceso propuesto en la sección 4 al problema concreto presentado en la sección anterior y presenta los resultados obtenidos. Finalmente, la sección 7 presenta las conclusiones obtenidas del ejercicio de la sección anterior, así como puntos que quedan pendientes para trabajo futuro.

2. Teoría: Diseño de instrumentos de medición y selección de modelos y atributos

Los instrumentos más conocidos que los investigadores usan para recopilar datos acerca de la realidad, y evaluar sus hipótesis, son los de medición. En Ciencias Sociales pueden ser diseñados para medir tanto datos numéricos como nominales, para obtener conocimiento acerca de realidades personales o sociales de la muestra a la que se pretende investigar [8].

La gran mayoría de los conjuntos de datos relevantes en Ciencias Sociales existen porque previo a la recopilación de estos se tuvo un proceso de diseño de instrumento de medición para utilizar éste en la recopilación de los datos. Los instrumentos de medición o diseños de investigación pueden ser encuestas, entrevistas, experimentos, observaciones, recuperación de archivos textuales o de datos, o una combinación de cualquiera de los cinco primeros [3].

El proceso de elección de tipo de instrumento y su diseño es un proceso delicado que debe ser hecho a conciencia, tomando en cuenta tanto el tipo de datos que se quiere recabar como la naturaleza de la muestra. No está dentro de los alcances del presente trabajo entrar en detalles acerca de este procedimiento, pero es importante dejar en claro que el uso de conjuntos de datos preexistentes para la investigación siempre debe tomar en cuenta y alinearse con las características del instrumento original con el cual fueron recabados los datos. Ningún conjunto de datos existe en un vacío metodológico incluso si fueron recabados de manera automatizada.

La selección de modelos es el proceso de elegir un modelo matemático apropiado de entre una clase de modelos [9]. En el caso atendido por este trabajo se estaría hablando de un proceso de selección de modelos automatizado. Una de las herramientas para selección de modelos son las estrategias de selección de atributos que son procedimientos por los cuales se atiende el problema, cada vez más común, de que muchos conjuntos de datos de relevancia investigativa y en los que podrían aplicarse herramientas de aprendizaje máquina tienen una gran cantidad de atributos o dimensiones, lo que dificulta su procesado, disminuye el rendimiento de herramientas de regresión y aprendizaje máquina, y aumenta el uso de recursos computacionales en su procesado [4].

La selección de atributos tiene como objetivo elegir un subconjunto de atributos relevantes de entre los originales, descartando los atributos irrelevantes, redundantes o ruidosos [10]. Existen varios acercamientos de selección de atributos que pueden clasificarse de acuerdo con la disponibilidad de datos etiquetados o por el tipo de estrategia de búsqueda. En el caso que atañe a este trabajo se consideran casos automatizados en donde las etiquetas están disponibles y en donde la predicción es una

tarea de clasificación. Por el tipo de estrategia de búsqueda se hablará de herramientas de filtrado con las cuales los atributos se seleccionan en función de las características de los datos sin utilizar algoritmos de aprendizaje; y los de envoltorio, los cuales utilizan un algoritmo de aprendizaje específico para evaluar la calidad de las características seleccionadas [10]. Asimismo, se verán estrategias de regresión paso a paso. Se eligieron estos métodos por su popularidad en la comunidad de minería de datos.

3. El problema, los peligros y las ventajas

Una tarea común en los contextos de *Big Data* y aprendizaje máquina es la clasificación, en donde se entrena un algoritmo clasificador con un conjunto de datos etiquetado con las clases, y después este clasificador se pone a la tarea de predecir las clases de datos no etiquetados. Muchas veces se dispone de grandes conjuntos de datos de acceso público o de trabajos investigativos anteriores, pero para avanzar en el trabajo investigativo actual se requiere recabar nuevos datos de muestras diferentes de la misma población.

Muchas veces es deseable que estos nuevos conjuntos de datos tengan un menor número de atributos y esto lleva a la pregunta de cómo elegir un subconjunto de atributos que conserve la mayor cantidad de poder predictivo del conjunto de datos originales o que lo supere. A esto se le conoce como el problema de la selección de modelo en minería de datos, y parece un problema para los métodos automatizados de selección de atributos.

En el contexto de minería de datos, la literatura propone soluciones como la regresión paso a paso, o los métodos de selección de atributos por filtrado [11]. Sin embargo, existe también en la literatura una crítica consistente del uso de este tipo de técnicas incluso para predicción de datos dentro de un subconjunto de la misma muestra [4, 12], que advierten de peligros tales como sobreajuste y tendencia a llegar a falsas conclusiones. Recientemente, Smith llevó a cabo una simulación Montecarlo en donde generó conjuntos de datos con entre 10 y 1000 variables explicativas candidatas cuyos valores fueron tomados aleatoriamente de distribuciones normales.

Aleatoriamente se eligieron cinco variables explicativas y en base a estas se creó un valor objetivo. A continuación, se aplicó el método de selección de atributos de regresión paso a paso y se encontró que incluso a partir de conjuntos de datos con 100 variables explicativas candidatas la probabilidad de que una variable seleccionada sea realmente explicativa es prácticamente igual a que sea una variable sin relación real con la variable objetivo.

La situación empeora conforme mayor es el número de variables explicativas candidatas [4], contrario a lo que podemos encontrar en parte de la literatura en *Big Data* y la intuición de algunos analistas en cuanto a que “más datos, es mejor”. Esta crítica se extiende al uso de otras técnicas de selección de atributos más sofisticadas, aunque el mismo Smith reconoce que técnicas como los ensambles de árboles de decisión regularizados y la eliminación recursiva de atributos con validación cruzada tienen mayor éxito [4].

Sin embargo, los principales obstáculos en el uso de estas técnicas son con la teoría estadística. Muchas de las técnicas de selección de atributos sufren del problema de

tener hipótesis post hoc, es decir, intentar probar la hipótesis con el mismo conjunto de datos que la sugiere [13]. Esto es claro en las técnicas de regresión paso a paso o por filtrado, en donde se selecciona un subconjunto de atributos basado en pruebas estadísticas con la misma muestra con la que después se pretende hacer predicciones. Este uso de pruebas univariadas también conlleva violar varios de sus supuestos con su uso en conjunto, además de que éstas suponen hipótesis preespecificadas. Asimismo, la aplicación de estas técnicas puede ser un pretexto para evitar pensar en el problema de selección de modelo como tal [12].

Sin embargo, se puede argumentar que estos problemas son menos relevantes en el caso de predicción con nuevos datos, aunque dependiendo de los objetivos últimos del proyecto investigativo el uso de estrategias de selección de atributos automatizadas podría introducir restricciones en el tipo de conclusiones a las que se puede llegar. Por sí solas, las técnicas de selección de atributos automáticas son insuficientes para asegurar que el modelo sugerido por éstas lleve a un instrumento de medición apropiado para recabar nuevos datos que conserven el poder predictivo respecto a la variable de interés.

Algo de lo que el investigador únicamente se enteraría cuando evaluara el rendimiento del nuevo conjunto de datos. Sin embargo, siempre y cuando no se caiga en la falsa confianza de querer solo "justificar con los números" el nuevo modelo, es posible apoyarse de las herramientas de selección de atributos automatizadas, lo que conlleva las ventajas asociadas a automatizar una parte del proceso de diseño de instrumento de medición alta en investigación teórica en el dominio de la aplicación, si se toman en cuenta algunas condiciones.

4. Una propuesta de solución

Existe un fuerte argumento de que el verdadero "estándar de oro" para la selección de atributos es el conocimiento experto en el dominio de la aplicación [4]. Esto está en línea con la literatura de diseño de instrumentos de medición en donde la operacionalización de las conceptos es un proceso que debe ser llevado a cabo de manera juiciosa, justificando con evidencia previa y teoría la inclusión o exclusión de cada variable que operacionalice un concepto que se pretende medir [2, 4, 14].

Sin embargo, para estos propósitos, las estrategias de selección de modelo automatizadas ya han sido estudiadas y consideradas válidas si se cumplen ciertas condiciones en materia estadística. Además existe evidencia de que la inclusión de ciertos atributos puede no ser derivable de teoría y conocimiento experto y requerir evidencia empírica [15].

Asimismo, existen muchas críticas a la forma en que comúnmente se operacionalizan los conceptos en Ciencias Sociales, pero se reconoce que pese a todos los obstáculos metodológicos y epistemológicos, al final los datos pueden tener correlaciones con comportamientos reales y validez predictiva incluso si la fidelidad de los datos continua siendo un tema de discusión epistemológica o investigativa: a final de cuentas, una teoría es más que una generalización inductiva, más bien debe describir, explicar y predecir y no habiendo una diferencia profunda entre estos tres [14] existe valor en un modelo que tenga poder predictivo práctico siempre y cuando se respeten sus limitaciones estadísticas o inferenciales.

Atender a los pasos del proceso de diseño de instrumentos de medición y a las limitaciones de las herramientas estadísticas es precisamente lo que garantiza el poder predictivo, por lo que existen ciertas condiciones bajo las cuales se pueden utilizar herramientas de selección de atributos automatizadas para el presente objetivo:

1) El conjunto de datos original fue recabado por medio de un instrumento de medición cuyo diseño ya considera la teoría estadística y el conocimiento experto necesario. Esto no exime de estudiar el instrumento y su teoría.
2) Se tienen por lo menos dos muestras distintas para generar el subconjunto de atributos con uno y validar el rendimiento con otro. A falta de dos muestras distintas, un sustituto tolerable, con menor garantía de validez, sería separar en dos subconjuntos una sola muestra con gran cantidad de registros. Dependiendo del atributo a predecir puede ser necesario que se cuente con varios miles o decenas de miles de registros para este propósito.
3) Deben evitarse, en la medida de lo posible, usar estrategias de selección de atributos que violen ampliamente los supuestos de las pruebas estadísticas utilizadas, incluso si sus resultados parecen ser buenos. Esto incluye técnicas de filtrado por pruebas estadísticas y la regresión paso a paso.
4) Se debe tener un conocimiento general del dominio de la aplicación y acceso a conocimiento experto por medio de fuentes bibliográficas o en persona.
5) Se debe tener un conocimiento general del tema de diseño de instrumentos de medición y acceso a conocimiento experto.
6) Se debe contar con el tiempo necesario para hacer modificaciones relevantes al modelo sugerido por los métodos de selección de atributos automáticos.

Cumplidas estas condiciones pueden seguirse los siguientes seis pasos:

1) Asegurarse que los conjuntos de datos ya hayan sido limpiados y preprocesados de manera adecuada, así como probados para poder predictivo, preferentemente con validaciones cruzadas.
2) Aplicar una o varias estrategias de selección de atributos con los parámetros adecuados para obtener el número de atributos que se considera deseable para el nuevo conjunto de datos.
3) Comparar rendimientos de distintas estrategias, validando mediante entrenamiento con la muestra no utilizada para las estrategias de selección de atributos y comparando con el uso de los conjuntos de datos completos.
4) Aplicar conocimiento experto o juicio para elegir una de las estrategias de selección de atributos. Alternativamente, seleccionar el subconjunto de atributos producto de la técnica de eliminación recursiva de atributos con validación cruzada, la cual ha demostrado mayor eficacia y validez [4].
5) Elaborar un instrumento de medición de prueba, ciñéndose al diseño del instrumento original puesto que este impone limitaciones a respetarse.
6) A partir de este momento, considerar los siguientes pasos en el desarrollo del instrumento de medición [3, 8] exclusión de atributos o inclusión de nuevos atributos, u operacionalización de nuevos conceptos que sean necesarios y no estén incluidos en el instrumento original.

5. Un problema concreto

A continuación, se presenta un ejemplo de aplicación relevante al problema. Se tiene interés en crear un conjunto de datos de bajo número de atributos, pero con alto poder predictivo con respecto a variables de presencia/ausencia de satisfacción en el uso del tiempo en ámbitos laborales y académicos. La satisfacción es un componente del constructo del bienestar subjetivo [16] utilizado ampliamente en muchas y variadas áreas de investigación social y económica [17, 18] para medir el bienestar de las personas de acuerdo a su propia percepción.

La satisfacción se puede medir tanto en general, como en dominios específicos, tales como la vida familiar, la situación económica, la vida afectiva, etc. El uso del tiempo es uno de estos posibles dominios y, en concreto, el uso del tiempo en el trabajo o los estudios un par de dominios definidos más específicamente.

Estas variables son relevantes porque están correlacionadas con otras variables de bienestar subjetivo [19] y objetivo [18], así como con comportamientos relevantes de administración del tiempo en ambientes laborales y académicos ligados a condiciones de alto bienestar subjetivo y alto desempeño importantes tanto para individuos como para organizaciones [20, 21].

Si se quisiera desarrollar un instrumento que midiera variables que fueran predictivas respecto a la satisfacción en el uso del tiempo en un tipo de actividades, se tendría que desarrollar una teoría de que tipo de cosas están ligadas a esta variable objetivo, como se obtendrían los datos, que tipos de herramientas se utilizarían, etc. Todo esto es un proceso largo y oneroso que requiere de la participación de equipos multidisciplinarios y la consulta de literatura y estado del arte del área. Aquí se ve la ventaja del acercamiento parcialmente automatizado propuesto en este texto.

Si se pudiera hallar un par de conjuntos de datos, obtenidos de distintas muestras, que incluyan las variables de interés, que hayan sido generados por medio de un mismo instrumento que ya haya seguido el procedimiento metodológico adecuado de diseño de instrumento y recolección de datos, y que dichos conjuntos de datos tengan poder predictivo suficiente para que alguna herramienta clasificadora tenga un nivel de desempeño juzgado satisfactorio, se podría cumplir la primera y segunda condición y se abre la posibilidad de trabajar en cumplir las siguientes cuatro.

6. Una solución concreta, y resultados

Para llevar a cabo la tarea propuesta en la sección anterior, se tomaron los microdatos de la Encuesta Nacional de Uso del Tiempo (ENUT), diseñada y llevada a cabo por el Instituto Nacional de Estadística y Geografía (INEGI) en México, en sus ediciones 2014 y 2019. Una consulta a la documentación de ambas encuestas [22], [23] permite comprobar el buen diseño de ambos instrumentos de medición y que son comparables, por lo que se puede utilizar un conjunto de datos para seleccionar atributos y otro para entrenar.

En este caso se hará la selección de 30 atributos con el conjunto de datos de la ENUT-2014 (439 atributos con 42,118 registros,) y se harán pruebas del poder predictivo con validación cruzada con el conjunto de datos de la ENUT-2019 (432

Tabla 1. Métricas de desempeño, usando validación cruzada estratificada con 5 dobleces, en clasificación para atributos de presencia de satisfacción en el uso del tiempo para actividades académicas y laborales de los conjuntos de datos completos de la ENUT-2014 y ENUT-2019; utilizando 4 diferentes clasificadores.

Año	Atributo	Clasificador	*Accuracy*	*Precision*	*F1 Sc.*	Tiempo C.
2014	Actividades académicas o de estudio	Bosque A.	**0.9495**	**0.9521**	**0.9494**	18.05s
		MSV	0.4977	0.4973	0.4874	46.57s
		Red Neuronal	0.8634	0.8758	0.8612	61.45s
		R. Logística.	0.8952	0.8981	0.8950	**18.78s**
2014	Actividades laborales o económicas	Bosque A.	**0.8149**	**0.8639**	**0.8085**	0.66s
		MSV	0.5032	0.5037	0.4913	367.47s
		Red Neuronal	0.5654	0.6172	0.4745	248.23s
		R. Logística.	0.73772	0.73856	0.73746	**1.63s**
2019	Actividades académicas o de estudio	Bosque A.	**0.9583**	**0.9607**	**0.9583**	32.66s
		MSV	0.9128	0.9136	0.9128	37.65s
		Red Neuronal	0.9376	0.9378	0.9375	171.18s
		R. Logística.	0.9177	0.9182	0.9177	**0.94s**
2019	Actividades laborales o económicas	Bosque A.	**0.8142**	**0.8630**	**0.8077**	130.77s
		MSV	0.7783	0.7896	0.7761	1006.16s
		Red Neuronal	0.7507	0.7514	0.7505	1816.61s
		R. Logística.	0.7368	0.7380	0.7365	**3.51s**

atributos con 71,404 registros) utilizando únicamente los subconjuntos de 30 atributos generados con el conjunto de datos de la ENUT-2014.

En la Tabla 1 se puede ver el poder predictivo de ambos conjuntos de datos completos con respecto a las variables de presencia de satisfacción en el uso del tiempo en el trabajo y en el estudio. Estas son variables binarias. Se aplicó un proceso de submuestreo eliminando aleatoriamente registros de la clase más común hasta que ambas quedaron equilibradas, se eliminaron atributos correspondientes a metadatos y todos los atributos relacionados con bienestar subjetivo dada su alta correlación con las variables de interés y su poco común disponibilidad en el contexto de esta investigación, y se aplicó una validación cruzada estratificada con 5 dobleces.

Con la intención de ver si los cambios hechos a los conjuntos de datos cambian los desempeños de distintos clasificadores, se compararon los siguientes: bosque aleatorio con 1500 árboles y un máximo de 100 niveles, máquinas de soporte vectorial (MSV) con una forma de función de decisión de uno contra uno, red neuronal profunda con 1000 capas ocultas de 241 neuronas, y regresión logística multinomial con el algoritmo de optimización *Broyden–Fletcher–Goldfarb–Shanno*.

Como podemos ver en la Tabla 1, ambos conjuntos de datos tienen alto desempeño en la predicción de las dos variables de interés, por lo que son adecuados para la tarea de seleccionar un subconjunto de atributos aplicando las herramientas automáticas de selección de atributos. No sorprende que el subconjunto de datos de la ENUT-2019 tenga mayor desempeño ya que éste tiene más registros que el otro. Asimismo, podemos ver que el clasificador con el mejor desempeño es claramente el bosque aleatorio, a pesar de que la regresión logística tiene el menor tiempo de cómputo.

Hecho esto, se evaluaron distintas técnicas de selección de atributos, incluyendo algunas ya mencionadas como no recomendables, como filtrado por mayor índice de correlación Pearson, filtrado por pruebas univariadas f-test, y regresión paso a paso

Tabla 2. Tiempos de cómputo de estrategias de selección de atributos.

Estrategia de selección de atributos	Actividades académicas	Actividades laborales
Filtrado por correlación	**00:00:01**	**00:00:01**
Filtrado por pruebas univariadas (f-test)	**00:00:01**	**00:00:01**
Regresión paso a paso hacia adelante	00:21:10	00:37:03
Regresión paso a paso hacia atrás	00:05:42	00:05:39
Eliminación recursiva	05:40:28*	03:51:27*
Eliminación recursiva, validación cruzada	21:40:10*	23:05:21*

Formato: hh:mm:ss *Utilizando cómputo paralelo.

hacia adelante y hacia atrás. Se utilizaron dos que son recomendables por la literatura consultada: eliminación recursiva de atributos y eliminación recursiva de atributos con validación cruzada.

En el caso de las dos últimas técnicas, que son de tipo envoltorio, se incorpora el clasificador de bosque aleatorio dado que fue el que tuvo el mejor desempeño de acuerdo con los datos de la Tabla 1. En la Tabla 2 podemos ver el tiempo de cómputo tomado por cada estrategia de selección de atributos. Todos los cómputos fueron hechos en una computadora con procesador Intel Core i5-9300H a 2.40Ghz y 32GB de RAM. Se utilizó computo paralelo cuando fue posible.

Como se puede ver en la Tabla 2, el tiempo de cómputo de las técnicas de filtrado es prácticamente instantáneo, mientras que la regresión paso a paso toma algunos minutos; sin embargo, utilizar eliminación recursiva toma varias horas, y casi un día para eliminación recursiva con validación cruzada (con k = 5).

El conjunto de datos de la ENUT-2014 con algunos cientos de dimensiones y menos de 50,000 registros es relativamente pequeño comparado con los conjuntos de datos típicos en *Big Data*, por lo que podemos ver porque técnicas de filtrado y la regresión paso a paso son mucho más utilizadas que técnicas con mayor nivel de validez pero que toman mucho más tiempo de cómputo en el contexto de *Big Data*. En el caso de auxiliar en el diseño de instrumentos de medición, este proceso se realiza una sola vez y puede justificarse el uso de estrategias de selección de atributos con varias horas o incluso días de tiempo de cómputo, excepto en casos extremos.

En la Tabla 3 se puede ver una comparación de desempeño para clasificar por presencia de satisfacción en el uso de tiempo en actividades académicas y laborales tanto probando con la misma muestra de la ENUT-2014 como con la de la ENUT-2019. En ambos casos se usa validación cruzada con 5 dobleces. En todos los casos el tiempo de cómputo es menor que utilizando el conjunto de datos completo.

Como podemos ver en la Tabla 3 y la Tabla 1, las diferencias en desempeño en cuanto a las variables de satisfacción en el uso del tiempo en actividades académicas y laborales son pequeñas tanto entrenando con los conjuntos de datos completos como con los subconjuntos sugeridos por las estrategias de selección de atributos.

Asimismo, las diferencias también son pequeñas utilizando los subconjuntos sugeridos tanto entrenando con datos con la misma muestra con que se seleccionaron los atributos como entrenándolos con una muestra diferente, todo lo que da validación al procedimiento. No solo eso, sino que, en todos los casos excepto uno, el desempeño usando la muestra de validación (ENUT-2019) no es menor al obtenido utilizando la muestra original (ENUT-2014) con cualquiera de los cuatro clasificadores probados,

Tabla 3. Comparación de desempeño para clasificación por presencia de satisfacción de uso del tiempo en actividades académicas y laborales usando subconjuntos de entrenamiento obtenidos por distintas estrategias de selección de atributos, distintos clasificadores y entrenando con la misma muestra utilizada para selección de atributos (2014) y con una nueva muestra (2019).

Método de selección	**Clasificador**	**Act. Académicas**		**Act. Laborales**	
		***Accuracy* (2014)**	***Accuracy* (2019)**	***Accuracy* (2014)**	***Accuracy* (2019)**
Filtrado por correlación	Bosque A.	0.9576	0.9628	0.8079	0.8070
	MSV	0.9437	0.9516	0.8099	0.8039
	Red Neuronal	0.9427	0.9276	0.8082	0.8128
	R. Logística.	0.9154	0.9247	0.7790	0.7818
Filtrado, univariado (f-test)	Bosque A.	0.9561	0.9641	0.8102	0.8056
	MSV	0.9382	0.9518	0.8083	0.8051
	Red Neuronal	0.9432	0.9527	0.8113	0.8115
	R. Logística.	0.9191	0.9216	0.7799	0.7767
Regresión paso a paso hacia adelante	Bosque A.	**0.9601**	**0.9642**	**0.8191**	**0.8155**
	MSV	0.8997	0.9292	0.8034	0.7999
	Red Neuronal	0.9439	0.9582	0.8111	0.8144
	R. Logística.	0.9154	0.9246	0.7499	0.7425
Regresión paso a paso hacia atrás	Bosque A.	0.9583	0.9667	0.8159	0.8156
	MSV	0.8925	0.9412	0.8000	0.8019
	Red Neuronal	0.9437	0.9603	0.8051	0.8086
	R. Logística.	0.9202	0.9266	0.7355	0.7346
Eliminación recursiva	Bosque A.	0.9575	0.9619	0.8088	0.8074
	MSV	0.9399	0.9482	0.7927	0.7953
	Red Neuronal	0.8922	0.9532	0.7726	0.7797
	R. Logística.	0.9110	0.9203	0.7348	0.7335
Eliminación recursiva con validación cruzada	Bosque A.	0.9573	0.9639	0.8135	0.8119
	MSV	0.9390	0.9472	0.7936	0.7937
	Red Neuronal	0.9480	0.9514	0.7696	0.7812
	R. Logística.	0.9156	0.9202	0.7339	0.7372

esto es atribuible a la mayor cantidad de registros en la nueva muestra, y que estas son comparables entre sí con el mismo diseño metodológico.

Y la excepción, como veremos más adelante, se trata del único caso en que se aplicó un criterio no automatizado potencialmente mejorable. Sin embargo, mayor desempeño no necesariamente es el único criterio relevante.

El mejor desempeño para predicción de satisfacción en el uso del tiempo en actividades académicas y laborales es visto utilizando regresión paso a paso hacia adelante con bosque aleatorio, aunque el acercamiento para selección de atributos más recomendable por la teoría es eliminación recursiva de atributos con validación cruzada cuya diferencia en desempeño con la regresión paso a paso es mínima y por lo tanto aceptable.

Comparando los valores de la Tabla 3 y la Tabla 1 es claro que, aunque la diferencia en rendimiento entre bosque aleatorio y los otros clasificadores se reduce considerablemente al utilizar conjuntos de datos con menor dimensionalidad, este primero sigue siendo la mejor opción en general dado que solo se dio un caso en donde las máquinas de soporte vectorial la superaron ligeramente. Respecto a la excepción

mencionada de un rendimiento inferior utilizando la muestra de validación de la ENUT-2019 con respecto a utilizar la misma muestra de la ENUT-2014, se da al utilizar bosque aleatorio para predecir satisfacción en el uso del tiempo en actividades laborales con el conjunto de atributos seleccionado por eliminación recursiva con validación cruzada.

Es importante notar que dicho método, a diferencia de los otros, sugiere 30 atributos para predicción de satisfacción en el uso del tiempo en actividades académicas que es el número de atributos que se busca. Sin embargo, la misma estrategia sugiere 181 atributos para predicción de satisfacción en el uso del tiempo en actividades laborales, lo que apunta a que el poder predictivo para esta variable está más distribuido entre las variables explicativas candidatas.

Tanto para poder seguir comparando directamente como por razones prácticas, y pagando un costo pequeño en desempeño por ello, se seleccionaron los 30 atributos para predecir satisfacción en el uso del tiempo en actividades laborales de la siguiente forma: de entre los 181 sugeridos en el caso de la eliminación recursiva con validación cruzada se seleccionaron los 27 atributos que dicha estrategia sugirió tanto para predicción de satisfacción en el uso del tiempo en actividades académicas como laborales y se seleccionaron aleatoriamente 3 atributos más de entre los restantes atributos sugeridos para predicción de satisfacción en el uso del tiempo en actividades laborales y que no fueron sugeridos para predicción de satisfacción en el uso del tiempo en actividades académicas.

Esto es un acercamiento útil, aunque mejorable, para esta situación específica en que el mismo nuevo instrumento de medición se pretende utilizar para recolectar datos para predecir ambas variables, con lo que al final de este trabajo se contaría con 33 atributos sugeridos para dicho instrumento debido a las coincidencias entre las sugerencias para predecir satisfacción en el uso del tiempo en actividades académicas y laborales con 30 atributos.

Una mejor selección de los tres atributos no coincidentes, pero ya respaldada con conocimiento experto, podría mejorar el desempeño en la predicción de satisfacción en el uso del tiempo en actividades laborales utilizando como punto de partida el subconjunto de atributos sugerido por eliminación recursiva de atributos con validación cruzada.

En la Tabla 4 está la lista de atributos que se sugiere que el nuevo instrumento incluya para predecir satisfacción en el uso del tiempo en actividades académicas y laborales, es decir, los datos que se necesitarían de un respondiente para predecir si está satisfecho con su uso de tiempo en actividades académicas y laborales. El trabajo de diseño de un instrumento de medición no acaba en este punto, sino que continua con un análisis de los atributos sugeridos por el procedimiento tanto en el marco teórico como de aplicación del instrumento.

De ser relevante se continua con más pruebas utilizando diferentes clasificadores, descartando algunos atributos e incluyendo otros no contenidos en el conjunto de datos original en base a sugerencias de la literatura o de expertos; asimismo, dependiendo de la naturaleza de los atributos algunos pueden transformarse o agruparse, como por ejemplo en el caso del ejemplo presentado: distintas variables correspondientes a distintas actividades a las que se asignó tiempo pueden englobarse en una sola, por ejemplo, tiempo dedicado a “actividades domésticas”, o “actividades sociales”. Sin embargo, este trabajo ya forma parte de los siguientes pasos del proceso de diseño de

Tabla 4. Subconjunto de atributos de la ENUT [22, 23] con mayor desempeño, producido por eliminación recursiva con validación cruzada, para predicción de satisfacción en el uso del tiempo en actividades académicas y laborales.

Número total de habitaciones en la vivienda	Tiempo dedicado a limpiar o recoger el interior de la vivienda (F.S.)
Número de personas que habitan la vivienda	Tiempo dedicado a doblar/acomodar/guardar ropa (E.S.)
Edad en años	Estado civil: soltero/no soltero
Asiste a la escuela (edad 5 a 24 años)	Tiempo dedicado a hacer deporte o ejercicio físico (E.S.)
Tiempo dedicado a trabajar (E.S.)	Tiempo dedicado a ver televisión sin hacer otra actividad (E.S.)
Tiempo dedicado a dormir (incluye siesta) (E.S.)	Tiempo dedicado a revisar el correo; consultar redes sociales o (E.S.)
Tiempo dedicado a dormir (incluye siesta) (F.S.)	Tiempo dedicado a revisar el correo; consultar redes sociales o chatear (F.S.)
Tiempo dedicado a comer (E.S.)	Vivienda tiene bomba de agua**
Tiempo dedicado a comer (F.S.)	Se dedica exclusivamente a estudiar
Tiempo dedicado a arreglo e higiene personal; ir al baño (E.S.)	Tiempo dedicado a proteger bienes y vivienda (E.S.)
Tiempo dedicado a arreglo e higiene personal; ir al baño (F.S.)	Tiempo dedicado a tomar cursos o clases (F.S.)*
Tiempo dedicado a tomar cursos o clases (E.S.)	Tiempo dedicado a tareas; prácticas escolares o estudiar (F.S.)*
Tiempo dedicado a tareas; prácticas escolares o estudiar (E.S.)	Tiempo dedicado a trasladarse de y hacia la escuela (F.S.)*
Tiempo dedicado a trasladarse de y hacia la escuela (E.S.)	Tiempo dedicado a organizar o repartir los quehaceres del hogar (E.S.)**
Tiempo dedicado a cocinar o preparar alimentos o bebidas (E.S.)	Tiempo dedicado a consultar información, navegar en Internet (E.S.)
Tiempo dedicado a servir comida; recoger; lavar; secar y acomodar trastes (E.S.)	Tiempo dedicado a asistir o participar en actividades/celebraciones religiosas (E.S.)**
Tiempo dedicado a limpiar o recoger el interior de la vivienda (E.S.)	

**Sugerido solo para predicción en el contexto de actividades académicas.*
***Sugerido solo para predicción en el contexto de actividades laborales.*
E.S. = Entre semana. F.S.= Durante el fin de semana.

instrumentos de medición y a partir de este momento requiere de un trabajo metodológico propio.

7. Conclusiones y trabajo a futuro

El presente trabajo ofrece un ejemplo que muestra evidencia de que es posible utilizar el análisis de conjuntos de datos preexistentes para auxiliar en el diseño de instrumentos de medición con rigor metodológico, valiéndose de técnicas automatizadas de selección de atributos utilizadas para selección de modelos.

En la sección 4 se ofrecen seis pasos y seis condiciones que deben cumplirse para poder validar los resultados obtenidos con este acercamiento, principalmente el uso de dos muestras diferentes obtenidas mediante instrumentos de medición con la misma

metodología y el respaldo de una cantidad razonable de conocimiento experto en el dominio de la aplicación y en el diseño de instrumentos de medición.

Se mencionan las ventajas: que las herramientas utilizadas en el contexto de *Big Data* y minería de datos permiten acelerar y facilitar el proceso de diseño de instrumentos de medición, pero se alertan de los peligros y se propone una forma de evitarlos. En base a los resultados en la Tabla 1 y la Tabla 3, se puede ver que los modelos de 30 atributos seleccionados por las estrategias de selección de atributos automatizadas quedan validados al entrenarlos con una muestra diferente, la de la ENUT-2019, con la cual las métricas de *accuracy* no solo son altas sino que en este caso, usando el mejor clasificador, nunca son menores al desempeño obtenido al entrenar con los conjuntos de datos originales completos de 432 y 439 atributos para la ENUT-2019 y ENUT-2014 respectivamente.

Esto es evidencia de que es posible utilizar un instrumento de medición preexistente para diseñar uno de menor número de atributos con similar o mejor poder predictivo para una o varias variables de interés, con las ventajas que esto conlleva a comparación de empezar el diseño desde un principio. La evidencia presentada apunta a que cumplir las seis condiciones y seguir los seis pasos descritos en la sección 4 evita problemas tanto estadísticos como metodológicos para este procedimiento al tiempo que permite automatizar una importante parte del proceso de diseño de instrumentos de medición, lo que es una ventaja grande en el contexto de minería de datos e investigación.

Sin embargo, debe considerarse que este acercamiento no exime al investigador o analista de pensar en y estudiar los conceptos operacionalizados por el instrumento original, y que este procedimiento de automatización necesariamente traslada las bondades, límites y posibles errores del instrumento original al nuevo de maneras que necesitan ser analizadas independientemente durante los subsecuentes pasos de diseño del instrumento de medición que no es agotado por el procedimiento presentado en este trabajo. Atención a las situaciones que se presentan en estos siguientes pasos y un análisis del desempeño obtenido con las muestras recabadas utilizando el nuevo instrumento de medición son precisamente trabajo futuro pendiente.

Referencias

1. Stevens, S. S.: On the theory of scales of measurement. Science, vol. 103, no. 2684, pp. 677 (1946) doi: 10.1126/science.103.2684.677
2. Martínez-Rizo, F. Los indicadores como herramientas para la evaluación de la calidad de los sistemas educativos. Sinéctica, no. 35, pp. 1–17, (2010)
3. Vogt, P. W. Gardner, D. C., Haeffele, L. M.: General introduction, design, sampling, and ethics. In When to use what research design, The Guilford Press, pp. 1–7 (2012)
4. Smith, G.: Step away from stepwise. Journal Big Data, vol. 5, no. 32, pp. 1-12 (2018) doi: 10.1186/s40537-018-0143-6
5. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Magazine, vol. 17, no. 3, pp. 37 (1996) doi: 10.1609/aimag.v17i3.1230
6. Gass S. I., Fu, Eds., M. C.: Validation. In encyclopedia of operations research and management science. Boston, MA: Springer US, pp. 1597 (2013)

7. Fressoli J. M., Arza, V.: Los desafíos que enfrentan las prácticas de ciencia abierta. Ediciones complutenses; Teknokultura, vol. 15, no. 2, pp. 429–448 (2018) doi: 10.5209/tekn.60616
8. Mejía Mejía, E.: Instrumentos de acopio de datos. Técnicas e instrumentos de investigación, 1a edición, Lima, Perú: Centro de Producción Editorial e Imprenta de la Universidad Nacional Mayor de San Marcos, pp. 13–94 (2005)
9. Sammut, C., Webb, G. I.: Model selection. Encyclopedia of machine learning and data mining, Boston, Springer US, pp. 845 (2017)
10. Wang, S., Tang, J., Liu, H., Mining, C., Sammut, G. I.: Feature selection. Encyclopedia of machine learning and data webb, Boston, Springer US, pp. 503–511, (2017) doi: 10.5555/3153490
11. Cios, K. J., Swiniarski, R. W., Pedrycz, W., Kurgan, L. A.: Feature extraction and selection methods. Data Mining: A Knowledge Discovery Approach, Boston, Springer US, pp. 133–33 (2007)
12. Harrel, F.: Problems with stepwise regression. Stata | FAQ, P. W. Vogt. Post Hoc Theorizing, Dictionary of Statistics & Methodology, (1996) 2005https://methods.sagepub.com/reference/dictionary-of-statistics-methodology.
13. Roskam, E. E.: Operationalization, a superfluous concept. Quality and Quantity, vol. 23, no. 3–4, pp. 237–275 (1989) doi: 10.1007/BF00172446
14. Castle, J. L., Doornik, J. A., Hendry, D. F.: Evaluating automatic model selection. Journal of Time Series Econometrics, vol. 3, no. 1 (2011) doi: 10.2202/1941-1928.1097
15. Diener, E.: Subjective well-being. Psychological Bulletin, vol. 95, no. 3, pp. 542–575 (1984) doi: 10.1037/0033-2909.95.3.542
16. Diener, E., Suh, E., M. Lucas, R. E., Smith, H. L.: Subjective well-being: Three decades of progress, Psychological Bulletin, vol. 125. pp. 276–302 (1999)
17. OECD, Why does subjective well-being matter for well-being, en How's Life: Measuring well-being, Paris: OECD Publishing, pp. 266–267 (2011)
18. Boniwell, I.: Satisfaction with time use and its relationship with subjective well-being. The Open University (2006) doi: 10.21954/ou.ro.0000e97
19. Aeon B.: Aguinis, H.: It's about time: New perspectives and insights on time management. Academy of Management Perspectives, vol. 31, no. 4, pp. 309–330 (2017) doi: 10.5465/amp.2016.0166
20. Claessens, B. J. C., van Eerde, W., Rutte, C. G., Roe, R. A.: A review of the time management literature, Emerald insight, Discover Journals, Books & Case Studies, vol. 36, no. 2, pp. 255–276 (2007) doi: 10.1108/00483480710726136
21. INEGI. Encuesta nacional sobre uso del tiempo 2014 (ENUT): Documento metodológico. Aguascalientes: Instituto Nacional de Estadística y Geografía (2015)
22. INEGI. Encuesta nacional sobre uso del tiempo 2019: Diseño Conceptual. Aguascalientes: Instituto Nacional de Estadística y Geografía (2020)

ISSN 1870-4069

Máquina de estado líquido con reservorio de neuronas pulsantes a tiempo discreto para clasificación espacio-temporal

Christian Alba-Cisneros[1], Andrés Espinal[2], Manuel Ornelas-Rodríguez[1], Horacio Rostro-Gonzalez[3], Marco Sotelo-Figueroa[2], Jorge Soria-Alcaraz[2], Héctor Puga-Soberanes[1], Raúl Santiago-Montero[1]

[1] Tecnológico Nacional de México, Instituo Tecnológico de León, México

[2] Universidad de Guanajuato, División de Ciencias Económico Administrativas, México

[3] Universidad de Guanajuato, División de Ingenierías Irapuato-Salamanca, México

{m19240999, manuel.ornelas}@leon.tecnm.mx, {aespinal, hrostro}@ugto.mx

Resumen. La Máquina de Estado Líquido (LSM por sus siglas en inglés -Liquid State Machine-) es un modelo computacional capaz de tratar naturalmente datos espacio-temporales para realizar tareas como clasificación. Básicamente, una LSM genera respuestas descriptivas de los datos espacio-temporales de entrada mediante un reservorio de neuronas pulsantes; el cual no es entrenado y tampoco dependiente de la tarea a realizar. Posteriormente, representaciones de las respuestas son procesadas por una función de aprendizaje simple; que es entrenada y dependiente de la tarea a llevar a cabo. En este trabajo, se propone utilizar un modelo de neurona pulsante relativamente sencillo, de bajo costo computacional y que trabaja a tiempo discreto como unidad de procesamiento del reservorio en una implementación estándar de LSM. La configuración propuesta de LSM es usada para clasificar la base de datos espacio-temporal Poker-DVS. Los resultados presentados y analizados se obtienen a partir de procesos de clasificación para todas las combinaciones a pares de las 4 clases y con el total de clases disponibles en la base de datos.

Palabras clave: Máquina de estado líquido, neurona pulsante, clasificación espacio-temporal, poker-DVS.

Liquid State Machine with Reservoir of Discrete Time Spiking Neurons for Spatio-Temporal Classification

Abstract. The Liquid State Machine (LSM) is a computational model capable of naturally processing spatio-temporal data to perform tasks such as classification.

Basically, a LSM generates descriptive responses of spatio-temporal input data by means of a reservoir of spiking neurons; which is neither trained nor dependent on the task to carry out. Later, the descriptions of the responses are processed by a simple learning function; that is trained and dependent on the task to be performed. In this work, it is proposed to use a relatively simple spiking neuron model, with computationally low-cost and working at discrete time as reservoir's processing unit in a standard LSM implementation. The proposed LSM configuration is used to classify the Poker-DVS spatiotemporal database. The results presented and analyzed are obtained from classification processes for all pairwise combinations of the 4 classes and with the total of classes available in the database.

Keywords: Liquid state machine, spiking neuron, spatio-temporal classification, poker-DVS.

1. Introducción

La Máquina de Estado Líquido (LSM por sus siglas en inglés -Liquid State Machine-) [20] es uno de los modelos fundamentales del Cómputo Reservorio (RC por sus siglas en inglés -Reservoir Computing-), paradigma relativamente nuevo que permite llevar a cabo tareas de reconocimiento de patrones espacio-temporales, como reconocimiento de voz [27, 35], predicción de series de tiempo [25], clasificación de estilos musicales [12], entre otros.

Este paradigma se caracteriza por usar una Red Neuronal Recurrente (RNN por sus siglas en inglés -Recurrent Neural Network-) aleatoria como un reservorio no entrenado del cual se analizan sus respuestas a estímulos de entrada mediante una función externa simple de aprendizaje [33, 16, 17]; por lo tanto, RC provee una forma de aprovechar el poder computacional de las RNNs evitando las dificultades de su entrenamiento [3].

A diferencia de los otros modelos fundamentales del RC, Máquina de Estado de Eco [10] y la Regla de Aprendizaje de Decorrelación de Retropropagación [31], la LSM constituye su reservorio (también llamado líquido) con unidades de procesamiento que tratan naturalmente datos espacio-temporales; estas son neuronas pulsantes [5].

De tal manera, el reservorio de la LSM es una Red Neuronal Pulsante (SNN por sus siglas en inglés -Spiking Neural Network-); las SNNs son consideradas la tercera generación de las Redes Neuronales Artificiales (ANNs por sus siglas en inglés -Artificial Neural Network) [18].

El desempeño de las LSMs se relaciona con sus dos propiedades: separación y aproximación. La propiedad de separación refiere a la capacidad de una LSM para reflejar respuestas distintas para diferentes datos espacio-temporales de entrada y la propiedad de aproximación trata sobre la capacidad de la LSM para distinguir y transformar diferentes estados internos del reservorio en la salida deseada [20, 19].

Diversas investigaciones han enfocado sus esfuerzos en lograr mejoras en el desempeño de las LSM en tareas de reconocimiento de patrones espacio-temporales con base a estas propiedades; por ejemplo: la elección del modelo de neurona pulsante a utilizar en el reservorio [7], la exploración de la arquitectura de la SNN como

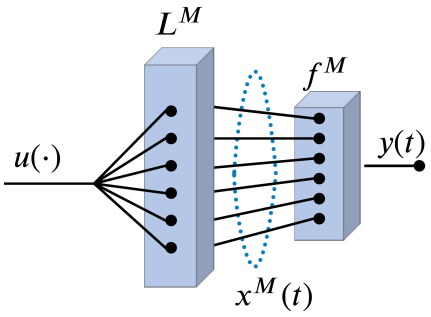


Fig. 1. Esquema de una máquina de estado líquido, la cual transforma flujos de entrada $u(\cdot)$ en flujos de salida $y(\cdot)$, L^M es el líquido, mientras que $x^M(t)$ es el estado líquido en el tiempo t el cual es transformado mediante un mapeo de lectura para generar una salida. $y(t)$ (tomada de [20]).

reservorio [9], el estudio y análisis de la conectividad sináptica del reservorio y sus efectos [11], la exploración y optimización de diferentes arquitecturas de la SNN usada como reservorio [9, 8], diseño de las representaciones de las respuestas descriptivas de la LSM a datos espacio-temporales [24] y reglas de aprendizaje para modificar los pesos de las conexiones sinápticas del reservorio [23, 22].

Un aspecto crucial que impacta ambas propiedades de la LSM es el modelo neuronal usado en el reservorio [7]. El presente trabajo tiene como propósito principal analizar el comportamiento y desempeño del modelo de neurona pulsante Belson-Mazet-Soula [30] (modelo BMS -nombrado así por sus autores [1]-) como unidad de procesamiento del reservorio de una implementación estándar de LSM (es decir, como se propone originalmente en [20]).

La neurona pulsante BMS es un modelo sencillo que trabaja a tiempo discreto y de bajo costo computacional [1], estas características motivan el desarrollo de la presente investigación debido que RNAs densas y con una cantidad significativa de parámetros pueden ser simuladas en equipos de cómputo comerciales; por ejemplo, para toda la experimentación de este trabajo se utilizó una laptop con AMD Ryzen 5 PRO 2500U 2.00 GHz y 8 GB en RAM.

La LSM con reservorio basado en el modelo BMS es probada en tareas de clasificación espacio-temporal usando la base de datos Poker-DVS; con esta base de datos se efectuaron diversas pruebas de clasificación mediante las combinaciones a pares de las 4 diferentes clases (Club, Diamond, Heart, Spade) y usando todas las clases. Los resultados obtenidos son analizados y comparados con algunos reportados en el estado del arte.

El resto del artículo se estrucutra de la siguiente manera. La sección 2 describe la LSM a través de los elementos que la conforman y la manera en que se utilizan en este trabajo. El modelo BMS de neurona pulsante es explicado en la sección 3. La experimentación y resultados obtenidos se muestran en la sección 4. Por último, las conclusiones y trabajo a futuro son comentadas en la sección 5.

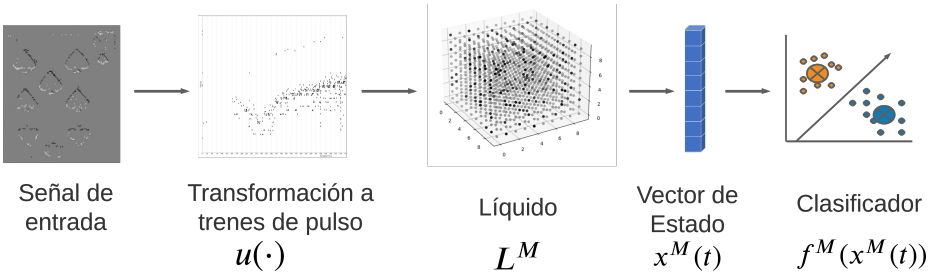


Fig. 2. Descripción extendida de la LSM.

2. Máquina de estado líquido

La LSM [20] es un modelo para realizar tareas de reconocimiento de patrones espacio-temporales, el cual aprovecha el poder de las RNN sin efectuar un entrenamiento [3]. Básicamente, la LSM está compuesta por dos partes [22]: una SNN recurrente L^M que actúa como reservorio (también llamado líquido o filtro) y un mapa de lectura $f^M(x^M(t))$ que procesa las representaciones $x^M(t)$ de las respuestas del reservorio a la señal espacio-temporal entrante $u(s)$ (ver Figura 1).

En otras palabras, una señal entrante $u(s)$ es procesada por el reservorio L^M, el cual actúa como un filtro temporal. De la respuesta del reservorio L^M a la señal entrante $u(s)$ se extrae una representación $x^M(t)$ la cual se denomina como vector de estado (Ec. 1):

$$x^M(t) = (L^M u)(t), \tag{1}$$

$$y(t) = f^M(x^M(t)). \tag{2}$$

En este trabajo, el vector estado $x^M(t)$ es un arreglo binario del tamaño de la cantidad de unidades de procesamiento en el reservorio y su valor es definido para cada componente, como 1 si la neurona correspondiente disparó en el tiempo total de simulación, de lo contrario es 0; a pesar de ser una idea sencilla para definir el vector de estado, se usó de esta forma debido a las características que las neuronas tienen como detectores de coincidencias [14], con el supuesto que, estímulos de entrada de la misma clase activen el mismo conjunto de neuronas y que este difiera lo suficiente para estímulos de entrada de otra clase.

Finalmente, la salida $y(t)$ (Ec. 2) proviene del mapa de lectura $f^M(x^M(t))$ que procesa cada momento (t) el estado líquido actual $x^M(t)$. El vector de estado $x^M(t)$ obtenido del reservorio L^M es clasificado por una función sencilla de aprendizaje $f^M(x^M(t))$; la cual se encarga de realizar la tarea de reconocimiento de patrones.

En este trabajo, la función de aprendizaje, es un clasificador de mínima distancia (ver [4] para una descripción detallada).

La Figura 2 bosqueja de manera general la implementación en este trabajo, una señal obtenida tomada por el sensor DVS del palo de la baraja es transformada a trenes de pulsos los cuales serán la señal de entrada $u(s)$, dicha señal es presentada al reservorio L^M del cual en el tiempo t se recupera un vector de estado $x^M(t)$ (vector binario resultado de la activación -1- o no activación -0- de cada neurona en el reservorio), este

Tabla 1. Distribución de patrones en Poker-DVS.

Clase	No de patrones
Club (C)	30
Diamond (D)	43
Heart (H)	23
Spade (S)	35

vector de estado es presentado a la función f^M (clasificador de mínima distancia) el cual lo clasifica en una de las posibles categorías del problema generando la señal de salida $y(t)$ de la LSM.

3. Modelo BMS de neurona pulsante

El modelo BMS [30] es una discretización derivada del conocido modelo de neurona pulsante de Integración y Disparo [6] (I&F por sus siglas en inglés -Integrate and Fire) la cual proporciona una correspondencia uno a uno entre el potencial de membrana y el estado de activación (representado como un código binario) de la neurona [26]. El modelo BMS está definido por la Ec. 3 y la Ec. 4 que describen el potencial sináptico y el estado de disparo de la i-ésima neurona V_i en el tiempo k; respectivamente:

$$V_i[k] = \gamma V_i[K-1](1 - Z_i[K-1]) + \sum_{j=1}^{N} W_{ij} Z_j[K-1] + I_i^{ext}, \tag{3}$$

$$Z_i[K] = \begin{cases} 1 \text{ si } V \geq \theta, \\ 0 \text{ en otro caso,} \end{cases} \tag{4}$$

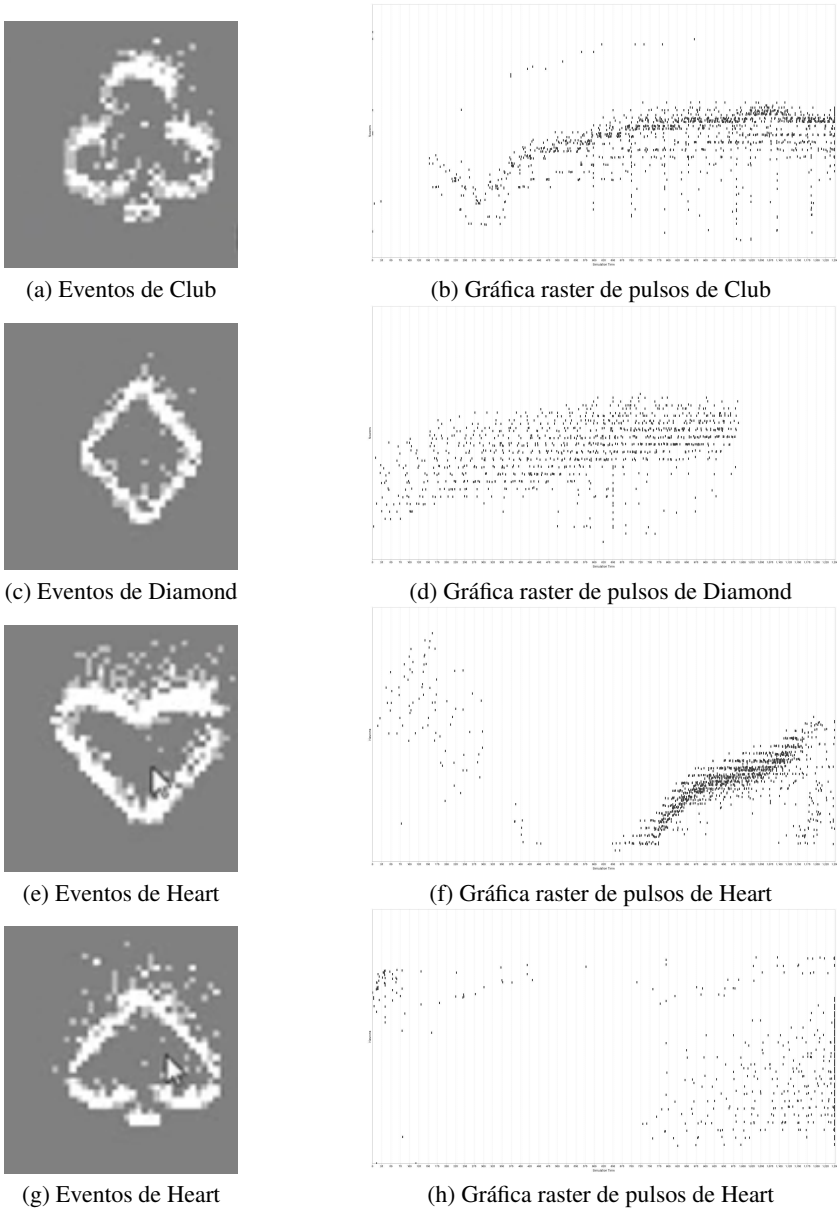
donde $\gamma \in [0, 1]$ define la tasa de disparo, N es el número de neuronas en la SNN y W es la matriz de pesos sinápticos e I_i^{ext} representa un estímulo externo (en este trabajo $I_i^{ext} = 0$). Cuando el potencial sináptico $V_i[k]$ alcanza un umbral θ dado, se produce un pulso (este es determinado en el término $Zi[k]$ de la ecuación(4) y la i-ésima neurona se restablece mediante el término $(1 - Zi[k])$ en la ecuación (3).

4. Experimentos y resultados

Para analizar el rendimiento de una LSM que implementa el modelo de neurona pulsante BMS como unidad de procesamiento de su reservorio, se utilizó la base de datos Poker-DVS [4] [28].

Esta base de datos contiene eventos capturados al pasar rápidamente las cartas de una baraja frente a un Sensor de Visión Dinámica (DVS por sus siglas en inglés -Dynamic Vision Sensor-), la tarea a llevar a cabo es la clasificación de los diferentes palos de la baraja.

[4] Disponible en: http://www2.imse-cnm.csic.es/caviar/POKERDVS.html



(a) Eventos de Club (b) Gráfica raster de pulsos de Club

(c) Eventos de Diamond (d) Gráfica raster de pulsos de Diamond

(e) Eventos de Heart (f) Gráfica raster de pulsos de Heart

(g) Eventos de Heart (h) Gráfica raster de pulsos de Heart

Fig. 3. Palos de la baraja representados por imágenes de flujos de eventos (columna izquierda) y gráficas raster de los trenes de pulsos (columna derecha).

La base de datos Poker-DVS contiene 4 clases (Club, Diamond, Heart y Spade) y un total de 131 patrones caracterizados por flujos de eventos de 32×32 píxeles. En la Tabla 1 se muestran los nombres de las clases, el identificador de clase y la cantidad de patrones en cada una de ellas.

Tabla 2. Valores usados en los parámetros de la LSM.

Parámetros LSM	
Neuronas en capa entrada	1024
Dimensión reservorio	$10 \times 10 \times 15$
Neuronas exitatorias (%) [36]	80 %
Neuronas inhibitorias (%) [36]	20 %
Porcentaje conexión c. entrada-reservorio	0,01 %
Rango de pesos c. entrada-reservorio	$(0{,}001, 1)$
λ	1
Prob. conexión excitatoria-excitatoria [36]	0,3
Prob. conexión excitatoria-inhibitoria [36]	0,2
Prob. conexión inhibitoria-excitatoria [36]	0,4
Prob. conexión inhibitoria-inhibitoria [36]	0,1
Rango de pesos sinápticos reservorio	$(0{,}001, 1)$
Tiempo de simulación	1250 [ms]
Parámetros neurona pulsante BMS	
γ [26, 2]	0,5
θ [26, 2]	1

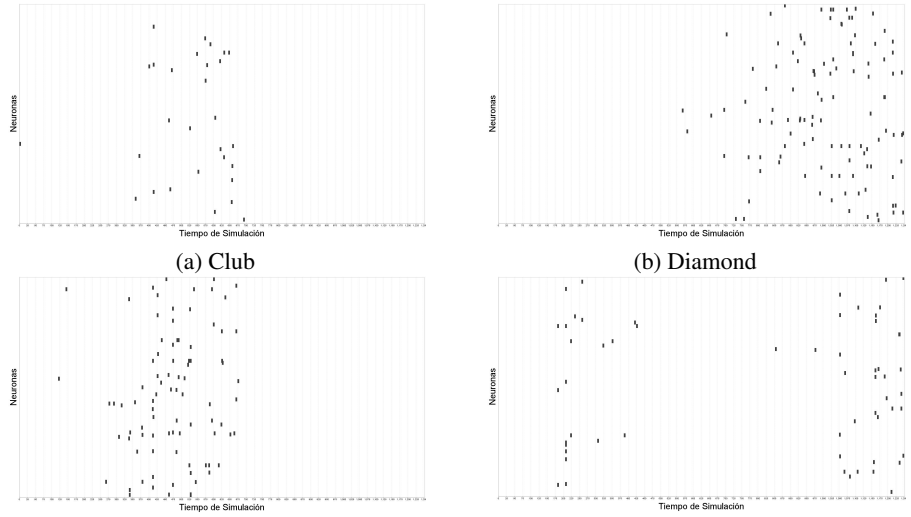
Los flujos de eventos representando cada patrón son interpretados como trenes de pulsos (uno por cada píxel); estos serán las señales de la capa de entrada de la LSM.

En la Figura 3, la columna izquierda muestra imágenes de flujos de eventos capturados por la DVS para cada uno de los diferentes símbolos de la baraja (Club -Figura 3a-, Diamond -Figura 3c-, Heart -Figura 3e- y Spade -Figura 3g-) y la columna derecha muestra la correspondiente gráfica raster de los trenes de pulsos de cada diferente símbolo de la baraja (Club -Figura 3b-, Diamond -Figura 3d-, Heart -Figura 3f- y Spade -Figura 3h-).

Para la experimentación, además de usar la base de datos Poker-DVS con todas sus clases, se generaron a partir de está, conjuntos extra con base a todas las combinaciones de pares de clases; dando un total de 7 conjuntos de experimentación.

Debido a la naturaleza aleatoria en la creación del reservorio en una implementación estándar de LSM, cada conjunto fue sometido a 36 procesos de clasificación, donde para cada uno se usó Validación Cruzada de K Pliegues (KFCV por sus siglas en inglés -K-Folds Cross Validation-) con $K = 10$ [15] generando en cada etapa individual de la validación un nuevo reservorio y calculando la exactitud para el conjunto de prueba obtenido por la LSM.

La Tabla 2 muestra la lista de valores usados para los parámetros de la LSM, el modelo BMS y la ecuación de conectividad (Ec. 3 en [36]), donde los valores de algunos parámetros fueron extraídos de trabajos del estado del arte (referencia a la derecha del nombre del parámetro) y para otros fueron obtenidos empíricamente o de acuerdo a las características de la base de datos Poker-DVS.



(a) Club (b) Diamond

(c) Heart (d) Spade

Fig. 4. Gráficas de raster de la actividad del reservorio para patrones de las diferentes clases de Poker-DVS(Esta actividad neuronal es transformada a vectores de estado para su clasificación).

La Figura 4, muestra la actividad generada por el reservorio en respuesta a patrones de las diferentes clases en la base de datos Poker-DVS. En la Tabla 3 se muestran para cada conjunto experimental de datos el promedio de exactitud de clasificación del conjunto de prueba obtenido al aplicar KFCV en el respectivo proceso de clasificación y también, en la parte inferior, un resumen estadístico por cada base de datos.

La Figura 5 muestra una gráfica de cajas y bigotes a modo de resumen visual de los estadísticos. A partir de los resultados obtenidos, se puede observar que implementaciones estándar de LSMs usando el modelo de neurona pulsante BMS en su reservorio, en general, tienen poca dispersión (ninguno sobrepasa el 0.028) pese a la naturaleza aleatoria de la creación de su reservorio.

Además, el desempeño de la LSM para problemas de clasificación binaria derivados de la base de datos Poker-DVS es para la mayoría de los casos bueno sobrepasando en promedio el $85\,\%$ de exactitud; solo dos subconjuntos que involucran la clase Spade rondan en promedio el $80\,\%$.

El comportamiento de la LSM al trabajar con todas las clases de la base de datos Poker-DVS, resultó en un $76\,\%$ de exactitud en promedio, la más baja en todas las pruebas; lo cual podrá deberse, además del incremento de clases, al conflicto que se puede vislumbrar de la clase Spade con Club y Diamond en los resultados de los subconjuntos biclase.

El desempeño de las LSMs con reservorio basado en el modelo BMS para el caso de la prueba de clasificacón usando las 4 clases se comparó contra diversos trabajos reportados en el estado del arte; a continuación se describen brevemente dichas investigaciones.

En [34] se propone una SNN de 4 capas, la primer capa convierte las señales de entrada en trenes de pulsos, la segunda capa está conformada por neuronas pulsantes de

Tabla 3. Promedios de exactitud de clasificación para los experimentos realizados.

# Exp	C-D	C-H	C-S	D-H	D-S	H-S	Todas
1	0.8900	0.9600	0.7985	0.9500	0.8000	0.9123	0.7449
2	0.8857	0.9675	0.7970	0.9750	0.8162	0.9123	0.7529
3	0.8857	0.9400	0.8485	0.9500	0.8438	0.9046	0.7359
4	0.8757	0.9600	0.7909	0.9583	0.8648	0.9200	0.7315
5	0.8800	0.9475	0.8076	0.9583	0.8295	0.9123	0.7442
6	0.9186	0.9600	0.8152	0.9583	0.8648	0.9123	0.7402
7	0.9000	0.9400	0.7742	0.9667	0.8505	0.9323	0.7909
8	0.8900	0.9550	0.7833	0.9750	0.8086	0.8800	0.7659
9	0.8943	0.9550	0.8242	0.9500	0.7876	0.9323	0.7569
10	0.9086	0.9400	0.7909	0.9333	0.8019	0.9000	0.7406
11	0.9000	0.9800	0.7894	0.9583	0.8019	0.9323	0.7902
12	0.9000	0.9475	0.8045	0.9583	0.7867	0.9446	0.7783
13	0.8800	0.9475	0.8485	0.9417	0.8219	0.9000	0.7322
14	0.9000	0.9475	0.7667	0.9500	0.8457	0.9169	0.7826
15	0.9043	0.9600	0.7727	0.9583	0.7943	0.9123	0.7692
16	0.9186	0.9600	0.7500	0.9500	0.7810	0.9123	0.7572
17	0.9000	0.9400	0.8303	0.9583	0.8724	0.9000	0.7366
18	0.9043	0.9800	0.8076	0.9583	0.7790	0.8923	0.7783
19	0.9000	0.9400	0.7333	0.9500	0.7933	0.9123	0.7783
20	0.9143	0.9600	0.8061	0.9583	0.8362	0.9123	0.7569
21	0.8943	0.9400	0.7894	0.9667	0.8019	0.9246	0.7620
22	0.8857	0.9475	0.8136	0.9583	0.8295	0.8923	0.7529
23	0.9043	0.9475	0.7970	0.9500	0.8171	0.9400	0.7616
24	0.8986	0.9675	0.8242	0.9667	0.8438	0.9200	0.8033
25	0.8857	0.9400	0.7894	0.9500	0.8524	0.8923	0.7909
26	0.9000	0.9675	0.8000	0.9750	0.8362	0.9000	0.7656
27	0.8714	0.9400	0.8061	0.9417	0.8238	0.9200	0.7580
28	0.8900	0.9475	0.8061	0.9583	0.7733	0.9323	0.7529
29	0.9143	0.9475	0.7894	0.9500	0.8305	0.8646	0.7902
30	0.8943	0.9350	0.8242	0.9500	0.8295	0.9000	0.7656
31	0.8943	0.9475	0.8167	0.9500	0.7590	0.9000	0.7779
32	0.9143	0.9475	0.8076	0.9583	0.8295	0.9246	0.7493
33	0.9043	0.9750	0.8061	0.9583	0.8457	0.9200	0.7946
34	0.8514	0.9475	0.8318	0.9417	0.8010	0.9123	0.8029
35	0.8757	0.9600	0.7818	0.9417	0.8305	0.9246	0.7822
36	0.8717	0.9400	0.7818	0.9750	0.8448	0.8800	0.7572
Min	0.8514	0.9350	0.7333	0.9333	0.7590	0.8646	0.7315
Max	0.9186	0.9800	0.8485	0.9750	0.8724	0.9446	0.8033
Med	0.8964	0.9475	0.8023	0.9583	0.8267	0.9123	0.7618
μ	0.8944	0.9524	0.8001	0.9558	0.8202	0.9112	0.7647
σ	0.0147	0.0121	0.0242	0.0101	0.0280	0.0174	0.0204

I&F las cuales se encargan de extraer las características de los trenes de pulsos que son enviadas a la siguiente capa, la tercer capa se encarga de la etapa de aprendizaje donde cada clase está representada por poblaciones de neuronas y finalmente, la cuarta capa es la capa de decisión donde se combinan las salidas de la capa anterior para realizar la claisifcación; la exactitud reportada es de $96{,}0\,\%$.

Los eventos espacio-temporales son caracterizados en [21] mediante descriptores de Histogramas de Eventos Orientados y Patrón Binario Local con los cuales reportan una exactitud del $98\,\%$.

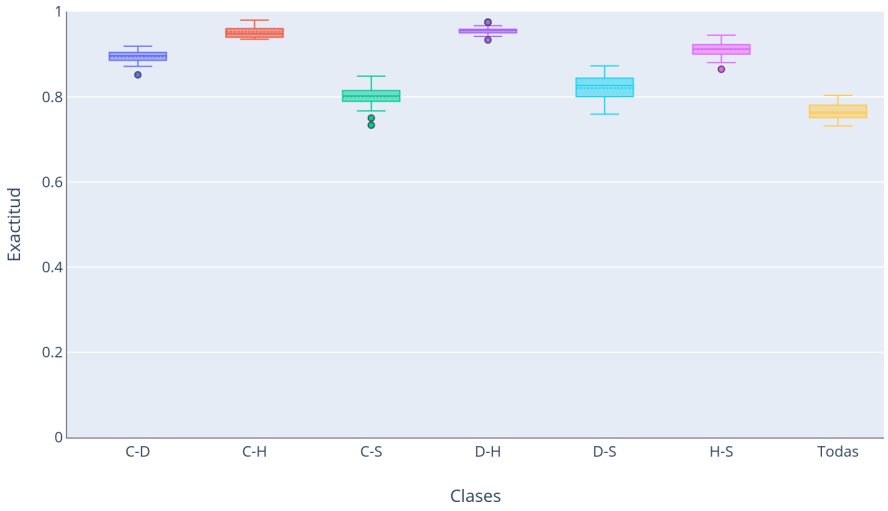


Fig. 5. Boxplot de los resultados.

En [29] se propone una metodología de 3 módulos, el primero se encarga de extraer características binarias de los eventos espacio-temporales, el segundo es un detector de movimiento y el tercero es un ensamble de modelos de aprendizaje máquina basados en una variante de clasificadores Bayesianos semi-ingenuos; se reporta una exactitud de $97\,\%$. Una SNN convolucional se utiliza en [13] reportando una exactitud del $90\,\%$.

Una regla de aprendizaje supervisado para SNNs es propuesta en [32], la cual usa la actividad generada por una SNN con topología arbitraria para construir histogramas y efectuar un proceso de aprendizaje basado en descenso de gradiente estocástico; se reporta una exactitud entre $94\,\%$ y $100\,\%$.

La Figura 6 muestra las exactitudes reportadas en el estado del arte y la obtenida mediante la configuración propuesta en esta investigación (LSM+BMS), se puede observar que las metodologías mejoradas o definidas adhoc logran mejores resultados de exactitud; sin embargo, la configuración propuesta de LSM permite explorar alternativas de mejora que serán comentadas en la sección 5.

5. Conclusiones y trabajo futuro

Hoy en día la cantidad y naturaleza de la información nos llevan cada vez más a utilizar instancias de pruebas espacio-temporales con la exigencia de resolver las cosas al instante. Con el presente trabajo vislumbramos un panorama favorecedor en cuanto al rendimiento de LSMs ya que no requieren de efectuar un proceso de calibración de pesos o de definir un patrón de conexiones sinápticas explícitos; los cuales suelen ser necesarios en la mayoría de implentaciones de ANNS, incluyendo las SNNs, que suelen ser procesos que requieren de expertos humanos, requerir de tiempo y recursos de procesamiento.

El modelo BMS es una neurona pulsante de bajo costo computacional que ha sido usado en trabajos con recursos de hardware limitado (como en aplicaciones robóticas),

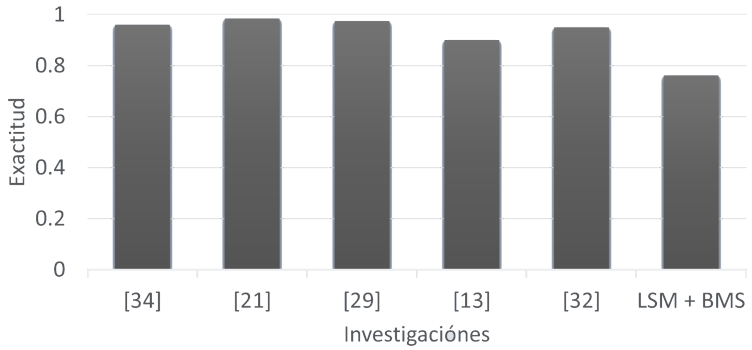


Fig. 6. Comparación de exactitudes obtenidas para Poker-DVS.

estas características hacen que el estudio de su implementación en LSMs sea atractivo para implementaciones en hardware, robótica o simulaciones en equipos de cómputo de recursos limitados.

Los resultados obtenidos en las pruebas a pares de clases de Poker-DVS muestran valores aceptables superando $85\,\%$ de exactitud en la mayoría de las instancias con poca dispersión en la repetición de pruebas a pesar de la aleatoriedad en la creación de los reservorios.

Sin embargo, los resultados obtenidos al resolver el problema de clasificación incluyendo las cuatro clases quedan por debajo de los reportados en el estado del arte; reportando un $76\,\%$ de exactitud.

Lo anterior, sirve de motivación para explorar alternativas para mejorar los resultados obtenidos; esto mediante la exploración de formas de representación de las respuestas descriptivas del reservorio, estructura de la topología del reservorio, calibración de los pesos exógenos y/o endógenos del reservorio, etc.

Aunado con las ventajas descritas previamente, este trabajo sirve como base para seguir explorando nuevas alternativas de bajo costo computacional en el área de las LSMs, permitiendo entender la naturaleza del modelo para así fortalecer sus debilidades.

En un futuro se propone trabajar en la calibración de los pesos y las conexiones sinápticas dentro del líquido mediante técnicas de optimización evolutivas o bionspiradas, con el objetivo de lograr resultados competitivos con el estado del arte. Además de buscar formas de mejorar las propiedades de las LSM que favorezcan a su desempeño y caracterización.

Referencias

1. Cessac, B.: A discrete time neural network model with spiking neurons. Journal of Mathematical Biology, vol. 56, no. 3, pp. 311–345 (2008) doi: 10.1007/s00285-007-0117-3
2. Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E. I., Ornelas-Rodriguez, M., Puga-Soberanes, H., Sotelo-Figueroa, M. A., Melin, P.: Quadrupedal robot locomotion: A biologically inspired approach and its hardware implementation. Computational Intelligence and Neuroscience, vol. 2016 (2016) doi: 10.1155/2016/5615618

3. Ferreira, A. A., Ludermir, T. B., De Aquino, R. R.: An approach to reservoir computing design and training. Expert Systems with Applications, vol. 40, no. 10, pp. 4172–4182 (2013) doi: 10.1016/j.eswa.2013.01.029
4. Friedman, M., Kandel, A.: Introduction to pattern recognition: Statistical, structural, neural and fuzzy logic approaches, vol. 32 (1999)
5. Gerstner, W.: Population dynamics of spiking neurons: Fast transients, asynchronous states, and locking. Neural Computation, vol. 12, no. 1, pp. 43–89 (2000) doi: 10.1162/089976600300015899
6. Gerstner, W., Kistler, W. M.: Spiking neuron models: Single neurons, populations, plasticity. Cambridge University Press (2002)
7. Grzyb, B. J., Chinellato, E., Wojcik, G. M., Kaminski, W. A.: Which model to use for the liquid state machine? In: Proceedings of the International Joint Conference on Neural Networks, pp. 1018–1024 (2009) doi: 10.1109/ijcnn.2009.5178822
8. Hazan, H., Manevitz, L. M.: Topological constraints and robustness in liquid state machines. Expert Systems with Applications, vol. 39, no. 2, pp. 1597–1606 (2012) doi: 10.1016/j.eswa.2011.06.052
9. Hourdakis, E., Trahanias, P.: Use of the separation property to derive liquid state machines with enhanced classification performance. Neurocomputing, vol. 107, pp. 40–48 (2013) doi: 10.1016/j.neucom.2012.07.032
10. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol. 148, no. 34, pp. 13 (2001)
11. Ju, H., Xu, J. X., Chong, E., vanDongen, A. M.: Effects of synaptic connectivity on liquid state machine performance. Neural Networks, vol. 38, pp. 39–51 (2013) doi: 10.1016/j.neunet.2012.11.003
12. Ju, H., Xu, J. X., vanDongen, A. M.: Classification of musical styles using liquid state machines. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1–7 (2010) doi: 10.1109/ijcnn.2010.5596470
13. Kaiser, J., Zimmerer, D., Tieck, J. C. V., Ulbrich, S., Roennau, A., Dillmann, R.: Spiking convolutional deep belief networks. In: International Conference on Artificial Neural Networks, pp. 3–11 (2017) doi: 10.1007/978-3-319-68612-7_1
14. Kamaruzaman, F., Shafie, A. A., Mustafah, Y. M.: Coincidence detection using spiking neurons with application to face recognition. Journal of Applied Mathematics, vol. 2015 (2015) doi: 10.1155/2015/534198
15. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 2, pp. 1137–1145 (1995)
16. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review, vol. 3, no. 3, pp. 127–149 (2009) doi: 10.1016/j.cosrev.2009.03.005
17. Lukoševičius, M., Jaeger, H., Schrauwen, B.: Reservoir computing trends. KI-Künstliche Intelligenz, vol. 26, no. 4, pp. 365–371 (2012) doi: 10.1007/s13218-012-0204-5
18. Maass, W.: Networks of spiking neurons: The third generation of neural network models. Neural networks, vol. 10, no. 9, pp. 1659–1671 (1997) doi: 10.1016/s0893-6080(97)00011-7
19. Maass, W.: Liquid state machines: Motivation, theory, and applications. Computability in context: Computation and logic in the real world, pp. 275–296 (2011) doi: 10.1142/9781848162778_0008
20. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation, vol. 14, no. 11, pp. 2531–2560 (2002) doi: 10.1162/089976602760407955

21. Negri, P.: Shapes characterization on address event representation using histograms of oriented events and an extended LBP approach (2018) doi: 10.48550/ARXIV.1802.03327
22. Norton, D., Ventura, D.: Improving liquid state machines through iterative refinement of the reservoir. Neurocomputing, vol. 73, no. 16–18, pp. 2893–2904 (2010) doi: 10.1016/j.neucom.2010.08.005
23. Paugam-Moisy, H., Martinez, R., Bengio, S.: Delay learning and polychronization for reservoir computing. Neurocomputing, vol. 71, no. 7–9, pp. 1143–1158 (2008) doi: 10.1016/j.neucom.2007.12.027
24. Rhéaume, F., Grenier, D., Bossé, É.: Multistate combination approaches for liquid state machine in supervised spatiotemporal pattern classification. Neurocomputing, vol. 74, no. 17, pp. 2842–2851 (2011)
25. Rosselló, J. L., Alomar, M. L., Morro, A., Oliver, A., Canals, V.: High-density liquid-state machine circuitry for time-series forecasting. International Journal of Neural Systems, vol. 26, no. 5 (2016) doi: 10.1142/s0129065715500367
26. Rostro-Gonzalez, H., Garreau, G., Andreou, A., Georgiou, J., Barron-Zambrano, J. H., Torres-Huitzil, C.: An FPGA-based approach for parameter estimation in spiking neural networks. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 2897–2900 (2012) doi: 10.1109/iscas.2012.6271920
27. Schrauwen, B., D'Haene, M., Verstraeten, D., Van Campenhout, J.: Compact hardware liquid state machines on fpga for real-time speech recognition. Neural networks, vol. 21, no. 2-3, pp. 511–523 (2008) doi: 10.1016/j.neunet.2007.12.009
28. Serrano-Gotarredona, T., Linares-Barranco, B.: Poker-DVS and MNIST-DVS. Their history, how they were made, and other details. Frontiers in Neuroscience, vol. 9, pp. 481 (2015) doi: 10.3389/fnins.2015.00481
29. Shi, C., Li, J., Wang, Y., Luo, G.: Exploiting lightweight statistical learning for event-based vision processing. vol. 6, pp. 19396–19406 (2018) doi: 10.1109/access.2018.2823260
30. Soula, H., Beslon, G., Mazet, O.: Spontaneous dynamics of asymmetric random recurrent spiking neural networks. Neural Computation, vol. 18, no. 1, pp. 60–79 (2006) doi: 10.48550/ARXIV.CS/0411052
31. Steil, J. J.: Backpropagation-decorrelation: Online recurrent learning with O (N) complexity. In: Proceedings of the IEEE International Joint Conference on Neural Networks. vol. 2, pp. 843–848 (2004) doi: 10.1109/ijcnn.2004.1380039
32. Stromatias, E., Soto, M., Serrano-Gotarredona, T., Linares-Barranco, B.: An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. Frontiers in Neuroscience, vol. 11, pp. 350 (2017) doi: 10.3389/fnins.2017.00350
33. Verstraeten, D., Schrauwen, B., d'Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. Neural Networks, vol. 20, no. 3, pp. 391–403 (2007) doi: 10.1016/j.neunet.2007.04.003
34. Wang, T., Shi, C., Zhou, X., Lin, Y., He, J., Gan, P., Li, P., Wang, Y., Liu, L., Wu, N., Luo, G.: CompSNN: A lightweight spiking neural network based on spatiotemporally compressive spike features. Neurocomputing, vol. 425, pp. 96–106 (2021)
35. Zhang, Y., Li, P., Jin, Y., Choe, Y.: A digital liquid state machine with biologically inspired learning and its application to speech recognition. IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 11, pp. 2635–2649 (2015) doi: 10.1109/tnnls.2015.2388544
36. Zhou, Y., Jin, Y., Ding, J.: Evolutionary optimization of liquid state machines for robust learning. In: International Symposium on Neural Networks, pp. 389–398 (2019) doi: 10.1007/978-3-030-22796-8_41

Electronic edition
Available online: http://www.rcs.cic.ipn.mx



http://rcs.cic.ipn.mx



Centro de Investigación
en Computación