

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

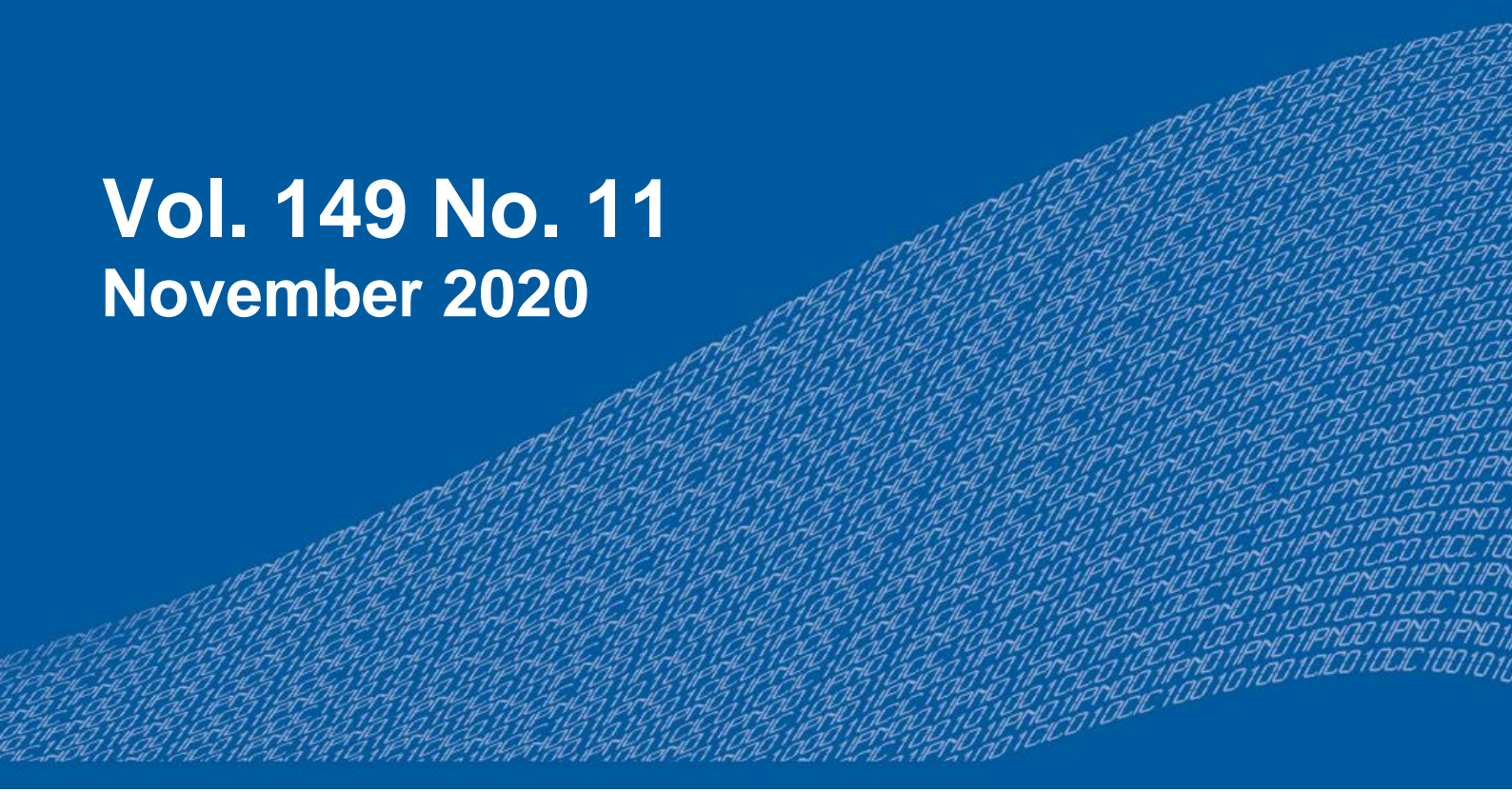


Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

ISSN: 1870-4069

Vol. 149 No. 11
November 2020



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
Olga Kolesnikova, ESCOM-IPN, Mexico
Petros Maragos, Nat. Tech. Univ. of Athens, Greece
Julian Padget, University of Bath, UK
Mateo Valero, UPC, Barcelona, Spain
Rafael Guzmán, Univ. of Guanajuato, Mexico
Juan Manuel Torres Moreno, U. of Avignon, France

Editorial Coordination:

Alejandra Ramos Porras

Research in Computing Science, Año 19, Volumen 149, No. 11, noviembre de 2020, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. <https://www.rcs.cic.ipn.mx>. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de noviembre de 2020.

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

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

Research in Computing Science, year 19, Volume 149, No. 11, November 2020, is published monthly by the Center for Computing Research of IPN.

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

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

Volume 149(11)

Artificial Intelligence and Industry 4.0

Juan Humberto Sossa Azuela
Maria Elena González Bañales
Edgar Gonzalo Cossio Franco (eds.)



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



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2020

ISSN: in process

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

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

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

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

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

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

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

Editorial

In this special issue of the prestigious journal "Research in Computing Science", artificial intelligence and industry 4.0 works are presented, which were carefully selected under the triple review scheme by the technical committee made up of national and international experts in the field through the EasyChair platform.

The selection was based on their originality, impact, contribution and scientific rigor. The acceptance rate was 54% and the topics covered range from machine learning, deep learning, computer vision, big data, IoT, robotics, smart cities, among other topics that directly impact Industry 4.0.

The papers presented in this special issue, as well as their authors, are described below: 1) Frequency of Mind-Wandering in a Sustained Attention to Response Task: A Cognitive Model of Distraction, author Renzo Cuadra, 2) Traffic Prediction Architecture based on Machine Learning approach for Smart Cities, authors Marco Hernandez Inzunza, Luis Hernandez Robles, Miriam Carlos Mancilla & Emmanuel López Neri, 3) Utility-pole, nuts and cross-arm visual detection, for electric connections maintenance robot, authors José Luis Gomes Torres, Saúl Martínez Díaz, Iddaly Iris Méndez Gurrola, Alejandro Israel Barranco Gutierrez & Kyoichi Tatsuno, 4) Proposal for a control, monitoring and assistance system for optimization of energy resources at home, authors Leandro Espino Espino, Yotziri Paloma Pérez Rios & Edgar Gonzalo Cossio Franco, 5) Intelligent model to determine the software development effort through machine learning, authors Edgar Gonzalo Cossio Franco, Mariana Chávez Marcial, Emma Yesenia Rivera Ruíz, Yotziri Paloma Pérez Rios & Marco Julio Franco Mora, 6) Selection of Actions for Navigation of a Mobile Robot Based on Fuzzy Q-Learning, authors Elizabeth López Lozada, Elsa Rubio, Humberto Sossa & Victor H. Ponce Ponce, 7) Attribution of Authorship of Twitter Messages through Automatic Syntactic Analysis, authors Francisco Antonio Castillo Velázquez, José Luis Martínez Godoy, Maria del Consuelo Patricia Torres Falcon, Jonny Paul Zavala de Paz, Adela Becerra Chavez & José Amilcar Rizzo Sierra, 8) A data fusion approach of physical variables measured through a wireless sensor network, authors Thomas A. Arias Pelayo, Guillermo Molero Castillo, Everardo Bárcenas & Rocío Aldeco Pérez, 9) A Physical Variable Data Fusion Approach as part of Ambient Intelligence, authors Julio Muñoz Benítez, Guillermo Molero Castillo, Everardo Bárcenas, Rocío Aldeco Pérez & Alejandro Velázquez Mena, 10) Bio-inspired Computing in Software Testing: a Systematic Literature Review, authors Jethran Gomez San Gariel, Angel Sanchez Garcia & Karen Cortes Verdin, 11) Design and construction of an IoT Humidity Sensor made of PVDF for applications in the Crop Industry, authors Lizbeth Berenice Contreras Cariño & Carlos Omar González Morán, 12) Algorithm for counting models in FNC, authors Pedro Bello López, Meliza Contreras González & Miguel Rodríguez Hernández, 13) Performance evaluation of infrared, ultrasonic and vision sensors for vehicle counting, authors Juan José Lozano Barrón, Osslan Osiris Vergara Villegas, Salvador Noriega Morales & Vianey Guadalupe Cruz Sánchez, 14) On the algebrization of paraconsistent logic CG'3, authors Miguel Pérez Gaspar & Everardo Ismael Bárcenas Patiño, 15) A Systematic Mapping Study of Artificial Intelligence in Software Requirements, authors Alfonso Robles Aguilar, Jorge Octavio Ocharán Hernández & Angel Sanchez

Garcia, 16) Gasoline price detector based in artificial vision and cloud computing, authors Mírcel Karel García Rodríguez, Osslán Osiris Vergara Villegas & Ivón Oristela Benitez Gonzalez, 17) Continuous retraining of an LSTM algorithm with a multi-agent system that implements the BDI model for predicting bicycle trips, authors Ramón A. Briseño, Juan C. López, Rocio Maciel Arellano, Víctor M. Larios, Raul J. Beltrán & J. Antonio Orizaga T., 18) Visual explanation of the prediction of an image classifier, authors Tonantzin Marcayda Guerrero Velázquez & Juan Humberto Sossa Azuela, 19) Autonomous reconstruction of a metric map of a competition track using artificial vision, authors Esther Guadalupe Diaz Sarmientos, Juan Humberto Sossa Azuela & Alberto Petrilli Barceló, 20) Design and development of a remote monitoring system implementing Internet of Things, authors Yehoshua Agilar Molina, Sergio Sosa & Josue Ernesto Galindo Morales, 21) Time2Vec Embedding on a Seq2Seq Recurrent Network for Pedestrian Trajectory Prediction, autore Victor Peñaloza, 22) Methodology to Control a Mobile Robot with Fuzzy Logic, authors Ana Claudia Zenteno Vázquez, Gustavo Trinidad Rubín Linares, María del Carmen Santiago Díaz, Judith Pérez Marcial, Yeiny Romero Hernández & David Luna Ramos, 23) Handcraft and automatic approaches for the recognition of leukemia images, authors Rocio Ochoa Montiel, Leonardo Martínez Ibarra, Humberto Sossa Azuela & Gustavo Olague, 24) Design of an artificial organ with application to nursing in the UAEM valle de chalco university center, authors Maribel Hernandez Arroyo & Marco Alberto Mendoza Pérez, 25) Optimization of the Coefficients of the Predictive Model of the Number of Daily Cases of Coronavirus Covid-19 in Mexico, authors Ana Claudia Zenteno Vázquez, Gustavo Trinidad Rubín Linares, María del Carmen Santiago Díaz, Judith Pérez Marcial, Yeiny Romero Hernández & Antonio Eduardo Álvarez Nuñez, 26) IoT Management Platform using Industria 4.0 techniques for precision agriculture, authors Ali Perez Gomez, Adriana Carolina Sanchez Cahuich & Janeth Jimenez Gomez, 27) IoT system for monitoring and analysis of light intensity in indoor plants, authors Gabriel Elías Chanchí Golondrino, Manuel Alejandro Ospina Alarcón & Wilmar Yesid Campo Muñoz.

Among the countries participating in this special issue are: Mexico, the Netherlands, Japan, Guatemala, Colombia and Cuba, as well as 15 national and international universities such as: University of Groningen, TECNM, UVM, University of Tokyo, UNIVA, IPN, UPQ, UNAM, UV, UAEM, BUAP, UACJ, UDG, UTM, CICESE.

Special thanks are extended to the Computer Research Center of the National Polytechnic Institute, as well as the technical committee and reviewer for their support.

Juan Humberto Sossa Azuela
Maria Elena González Bañales
Edgar Gonzalo Cossio Franco
Guest Editors

November 2020

Table of Contents

	Page
Frequency of Mind-Wandering in a Sustained Attention to Response Task: A Cognitive Model of Distraction	11
<i>Renzo Cuadra</i>	
Traffic Prediction Architecture based on Machine Learning Approach for Smart Cities	23
<i>Marco C. Hernandez Inzunza, Luis Hernandez Robles, Miriam A. Carlos Mancilla, Emmanuel López Neri</i>	
Utility-Pole, Nuts and Cross-Arm Visual Detection for Electric Connections Maintenance Robot	35
<i>José Luis Gomez Torres, Saúl Martínez Díaz, Iris Iddaly Méndez Gurrola, Alejandro Israel Barranco Gutiérrez, Kyoichi Tatsuno</i>	
Propuesta de un sistema de control, monitoreo y asistencia para optimización de recursos energéticos en el hogar.....	49
<i>Leandro Espino Espino, Yotziri Paloma Pérez Rios, Edgar Gonzalo Cossio Franco</i>	
Modelo inteligente para determinar el esfuerzo de desarrollo de software mediante machine learning	63
<i>Edgar Gonzalo Cossio Franco, Mariela Chávez Marcial, Emma Yesenia Rivera Ruiz, Yotziri Paloma Pérez Rios, Marco Julio Franco Mora</i>	
Selección de acciones para la navegación de un robot móvil basada en fuzzy Q-learning	79
<i>Elizabeth López Lozada, Elsa Rubio Espino, Juan Humberto Sossa Azuela, Víctor Hugo Ponce Ponce</i>	
Atribución de autoría de mensajes de Twitter a través del análisis sintáctico automático	91
<i>Francisco Antonio Castillo Velásquez, José Luis Martínez Godoy, María del Consuelo Patricia Torres Falcón, Jonny Paul Zavala De Paz, Adela Becerra Chávez, José Amilcar Rizzo Sierra</i>	
A Data Fusion Approach of Physical Variables Measured through a Wireless Sensor Network	103
<i>Thomas Alejandro Arias Pelayo, Guillermo Molero Castillo, Everardo Bárcenas, Rocío Aldeco Pérez</i>	

A Physical Variable Data Fusion Approach as Basis for the Reasoning Process in Ambient Intelligence	115
<i>Julio Muñoz Benítez, Guillermo Molero Castillo, Everardo Bárcenas, Rocío Aldeco Pérez, Alejandro Velázquez Mena</i>	
Cómputo bioinspirado en la prueba de software: una revisión sistemática de la literatura.....	125
<i>Jethran E. Gómez San Gabriel, Ángel J. Sánchez García, Karen Cortés Verdín</i>	
Diseño y construcción de sensor de humedad IoT hecho de PVDF para aplicaciones en la industria del cultivo	135
<i>Lizbeth Berenice Contreras Cariño, Carlos Omar González Morán</i>	
Algoritmo para el conteo de modelos en FNC.....	147
<i>Pedro Bello López, Guillermo De Ita Luna, Meliza Contreras González, Miguel Rodríguez Hernández</i>	
Evaluación del desempeño de sensores infrarrojo, ultrasónico y visión para el conteo de vehículos.....	159
<i>Juan José Lozano Barrón, Osslan Osiris Vergara Villegas, Salvador Noriega Morales, Vianey Guadalupe Cruz Sánchez</i>	
Sobre la algebrización de la lógica paraconsistente CG ³	169
<i>Miguel Pérez Gaspar, Everardo Bárcenas</i>	
A Systematic Mapping Study of Artificial Intelligence in Software Requirements	179
<i>Alfonso Robles Aguilar, Jorge Octavio Ocharán Hernández, Ángel Juan Sánchez García</i>	
Detector de precios de gasolina basado en visión artificial y computación en la nube	189
<i>Mirsel K. García Rodríguez, Osslan O. Vergara Villegas, Ivón O. Benítez González</i>	
Reentrenamiento continuo de un algoritmo LSTM con un sistema multiagente que implementa el modelo BDI para la predicción de viajes en bicicleta.....	201
<i>Ramon A. Briseño, Juan C. López, Rocío Maciel Arellano, Víctor M. Larios, Raul J. Beltrán, J. Antonio Orizaga T.</i>	
Explicación visual de la predicción de un clasificador de imágenes	211
<i>Tonantzin M. Guerrero Velázquez, Juan Humberto Sossa Azuela</i>	

Reconstrucción autónoma de un mapa métrico de una pista de competencias usando visión artificial	223
<i>Esther Diaz Sarmientos, Humberto Sossa Azuela, Alberto Petrilli Barceló</i>	
Diseño y desarrollo de un sistema de monitoreo remoto implementando internet de las cosas	235
<i>Y. Aguilar Molina, S.A. Sosa Ramírez, J.E. Galindo Morales, A.N. Morales Reyes, R. Gallardo Sánchez, I.V. Villaseñor García</i>	
Time2Vec Embedding on a Seq2Seq Bi-directional LSTM Network for Pedestrian Trajectory Prediction.....	249
<i>Victor Peñaloza</i>	
Metodología para controlar un robot móvil con lógica difusa	261
<i>David Luna, Ana C. Zenteno, María del C. Santiago, Yeiny Romero, Judith Pérez, Gustavo T. Rubín</i>	
Handcraft and Automatic Approaches for the Recognition of Leukemia Images.....	271
<i>Rocio Ochoa Montiel, Leonardo Martínez Ibarra, Humberto Sossa, Gustavo Olague</i>	
Diseño de un órgano artificial con aplicación a la enfermería en el centro universitario UAEM Valle de Chalco.....	281
<i>Maribel Hernández Arroyo, Marco Alberto Mendoza Pérez</i>	
Optimización de los coeficientes del modelo predictivo del número de casos diarios de coronavirus Covid-19 en México	293
<i>Ana C. Zenteno, María del C. Santiago, Yeiny Romero, Judith Pérez, Gustavo T. Rubín, Antonio E. Álvarez</i>	
Plataforma de gestión IoT mediante técnicas de industria 4.0 para agricultura de precisión.....	303
<i>Alí Pérez Gómez, Adriana Carolina Sánchez Cahuich, Janeth Jiménez Gómez</i>	
Sistema IoT para el seguimiento y análisis de la intensidad de luz en plantas de interiores	317
<i>Gabriel Elías Chanchí Golondrino, Manuel Alejandro Ospina Alarcón, Wilmar Yesid Campo Muñoz</i>	

Frequency of Mind-Wandering in a Sustained Attention to Response Task: A Cognitive Model of Distraction

Renzo Cuadra

University of Groningen,
Institute of Artificial Intelligence,
Faculty of Science and Engineering,
Netherlands

`r.m.cuadra.lazarte@student.rug.nl`

Abstract. The present study examines how the frequency of mind wandering varies over time during a semantic sustained attention to response task. A model was built on the ACT-R cognitive architecture in order to replicate the study conducted by McVay & Kane (2009) on human participants, where subjects showed a notable increase in the frequency of mind-wandering over time. The model was able to replicate human behavior, showing a steady increase in the proportion of mind wandering with respect to attending over time. This increase was found to be statistically significant, thus supporting the hypothesis that mind wandering increases over time. Further research is proposed where more complexity is introduced in the model, so that it is able to better approach human behavior with respect to response times.

Keywords: Mind-Wandering, SART, ACT-R, Cognitive Model.

1 Introduction

The experience of getting distracted is common to all human beings. While performing a given task, we may suddenly find ourselves thinking of some other topic, with little or no relation to the task at hand. This experience of distraction is called mind wandering, defined as the process of task-unrelated thinking that is initiated by the mind itself, as opposed to external triggers (Van Vugt, Taatgen, Sackur, Bastian, Borst, & Mehlhorn, 2015). It is a mental state (or a sequence of mental states) that arise freely in the absence of strong constraints on the content of thought (Christoff, Irving, Fox, Spreng & Andrews-Hanna, 2016). Mind-wandering as a mental state is more constrained than dreaming—given that the subject is consciously awake—but less constrained than creative thinking and goal-directed thought, where executive control is fully activated.

In cognitive tasks—commonly measured in education—such as reading comprehension, aptitude tests, intelligence tests and sustained attention tasks, mind-wandering has been found to have negative effects on performance (Mooneyham & Schooler, 2013; Smallwood & Schooler, 2015). Mind-wandering has also been found

to be correlated to increased error rates and variability in reaction times (Bastian & Sackur, 2013). Research on individual differences with respect to mind-wandering has shown a negative relationship between cognitive performance and the frequency of mind-wandering (Robison & Unsworth, 2018). Two theories have been posed in response to this. The control failure theory posits that mind-wandering arises spontaneously due to failure to maintain focus on the task (McVay & Kane, 2012). On the other hand, the lack of motivation theory argues that low-ability participants may feel disengaged from the task and therefore engage more in mind-wandering (Seli, Cheyne, Xu, Purdon & Smilek, 2015).

It has been observed that the rate of mind-wandering does not change linearly with time, so we may not engage in mind-wandering for equal amounts of time, the same number of times during a given cognitive task. Mind-wandering is also more frequent in attention tasks that require less vigilance (McVay & Kane, 2012; Robison & Unsworth, 2018). In an experiment looking at mind-wandering during lectures, it was observed that over time the frequency of mind-wandering increased and the amount of time subjects remained focused on the lecture decreased gradually (Risko, Anderson, Sarwal, Engelhardt, & Kingstone, 2011). This effect of fatigue has been replicated by Stawarczyk & D'Argembeau (2016), who found that greater sleepiness predicted a higher frequency of mind-wandering and lower task performance. In another study, drivers reported engaging more in mind-wandering when they felt tired as opposed to when they felt more alert (Burdett, Charlton & Starkey, 2016).

The present study will focus on mind-wandering during a sustained-attention-to-response-task (SART) based on semantic categories. The task presents the participant with a stream of visual stimuli that can be either a target or a non-target. When a target stimulus is presented, they are required to press a key as soon as possible. When a non-target stimulus is shown, they are required to withhold the key press (Cheyne, Carriere, & Smilek, 2009).

In the semantic SART implemented by McVay & Kane (2009), the participant sees a series of words on the screen. Target and non-target words differ in their semantic category—targets are fruits such as “apple” and non-targets are animals such as “dog”. During the task, the subjects are occasionally prompted by a thought probe asking them whether their recent thoughts were about task-related matters (on-task) or other matters (off-task). Responses to thought probes are considered as self-reports of mind-wandering: participants that report thinking of task-unrelated matters are taken to have been mind-wandering in the recent period.

This type of SART adds some interesting complexity to the simpler version where stimuli are only letters—O's are targets and Q's are non-targets, for example. The semantic SART imposes more cognitive demand on the subject, leading them to search in their memory for the appropriate categorisation of the words that are presented. Thus, the semantic SART typically shows higher mean response times compared to simpler types of SART (McVay & Kane, 2009). Furthermore, there is scarce literature looking into complex SART versions connected to mind-wandering with a computer modelling approach.

Hence, the aim of the study is to investigate how the frequency of mind-wandering varies over time during a semantic sustained-attention-to-response-task. This research question was tested on a computer model of McVay & Kane's (2009) experiment, implemented in the Adaptive Control of Thought-Rational (ACT-R) cognitive

architecture. This makes it possible to measure how the proportion of on-task and off-task thoughts varies across many consecutive SART trials, and to directly compare the model's performance to that of human participants. McVay & Kane (2009) found that, with increased repetition of SART trials, the proportion of task-unrelated thoughts increases while on-task thoughts decrease. Following their results, it is expected that the frequency of mind-wandering will increase over time during the semantic SART performed by the model.

2 Method

A computer model of the semantic SART was implemented in ACT-R, which is a powerful cognitive architecture where cognitive tasks can be modelled by programming a series of if-then statements—called production rules—describing how the cognitive elements interact (Van Vugt et al., 2015). The model for this study was built upon the SART model developed by Van Vugt et al. (2015), which was extended to incorporate the semantic SART that McVay & Kane (2009) conducted on human subjects. A detailed description of the model is offered below. The LISP code for the ACT-R model, as well as the dataset used for analysis, can be found in the following repository: <https://github.com/renzo-cuadra/mind-wandering-model>

2.1 Implementing Mind-Wandering

Mind-wandering was implemented following Van Vugt et al.'s (2015) ACT-R model, where task-unrelated thoughts during SART arise due to competition between two goals: attend and wander. When the activation of the attend goal is higher, the model focuses on performing the task. On the other hand, when the activation of wander is higher, the model starts retrieving memories repetitively from the declarative module, until it retrieves one memory, which reminds it to pay attention to the task. Only then does the mind-wandering process stop. If during mind-wandering the model sees any stimulus on the screen, it immediately presses the “w” key as a default response. This simulates how a distracted person would respond automatically, without evaluating the appropriate response. After giving this default response, the model may retrieve either the attend or wander goal, depending on activation. If the wander goal is activated, the memory retrieval process will start again. If the attend goal is activated, the model will get back to the task.

The production rules involved in this implementation of mind-wandering are shown below in pseudocode. The rule start-wandering begins the memory retrieval process as soon as the model detects that its current goal is wander. Then, the retrieve-memories rule checks that the memory retrieved is not the one that tells the model to pay attention and, if so, continues retrieving other memories. Meanwhile, the remember-to-attend rule activates when the pay-attention memory is retrieved, signaling the model to change its goal from wander to attend.

Finally, the default-response rule fires when the model is in the wander state and the screen changes (that is, a stimulus is shown), after which the model presses the “w” key and retrieves a new goal: either attend or wander, once again.

No individual differences were considered as part of the mind-wandering implementation in this model. The only source of variation in model performance across runs was the stochasticity introduced by model parameters and the randomness in stimuli display and memory retrieval during mind-wandering:

```
production start-wandering
  IF
    goal IS wander
  THEN
    retrieve memory
production retrieve-memories
  IF
    memory retrieved
    AND
    memory NOT pay-attention
  THEN
    Retrieve memory
production remember-to-attend
  IF
    memory retrieved
    AND
    memory IS pay-attention
  THEN
    set goal attend
production default-response
  IF
    goal IS wander
    AND
    screen changed
  THEN
    press key w
    AND
    retrieve goal
```

2.2 Implementing Semantic SART

Semantic information was implemented by incorporating a vocabulary of fifteen animal words and fifteen fruit words in declarative memory, mimicking a subject with perfect knowledge of these words and their semantic categories, which is to be expected from an average adult, such as the participants in McVay & Kane's (2009) experiment. Fruit and animal words were obtained from Battig & Montague's (1969) list of the most common English words in these categories (see Table 1). The added chunks included mappings from strings to objects and from objects to categories.

These mappings provide the model with knowledge of how a string of characters (a word) represents an object, and how that object belongs to a particular semantic category. The below mappings were provided for all fifteen words used in the model's vocabulary:

```
;; Chunks for string-to-object mappings
```

Table 1. List of fifteen most common English words in the fruit and four-footed animal categories from Battig & Montague (1969).

N°	Fruits	Four-footed animals
1	Apple	Dog
2	Orange	Cat
3	Banana	Horse
4	Grape	Lion
5	Pear	Bear
6	Peach	Tiger
7	Strawberry	Cow
8	Kiwi	Elephant
9	Pineapple	Deer
10	Watermelon	Mouse
11	Tomato	Pig
12	Plum	Rat
13	Grapefruit	Giraffe
14	Mango	Squirrel
15	Cherry	Rabbit

```
(map-apple isa word string "apple" meaning apple)
(map-dog isa word string "dog" meaning dog)
;; Chunks for object-to-category mappings
(apple isa object title apple category fruit)
(dog isa object title dog category animal)
```

The process for performing the semantic SART is as follows: when the attend goal is activated and a stimulus is presented, the model reads the text, maps that string onto an object by retrieving the corresponding string-to-object mapping in declarative memory and then categorizes the object by searching the object-to-category mapping.

After that, the model searches for the stimulus-response mapping to evaluate the appropriate response: it presses the “a” key if the object is a fruit and refrains from pressing any key if it is an animal. If, on the other hand, the model is in the wander state and it sees a new stimulus appear on the screen, it will press the “w” key as a default response. Figure 1 offers a graphical representation of this process.

2.3 Implementing Thought Probes

In McVay & Kane’s (2009) design, thought probes were presented as a menu where participants could select various options that represented how related their recent thoughts were to the task. In the model, however, thought probes are represented by an “X” on the screen. If a thought probe is shown and the attend state is active, the model will read it and map it to its proper response, which is to press the “x” key.

If the wander state is on, it will give the default mind-wandering response and press the “w” key. This makes it possible to detect when the model was paying attention or mind-wandering whenever a thought probe is presented. All responses to thought probes were recorded in order to determine the frequency of mind-wandering throughout the experiment.

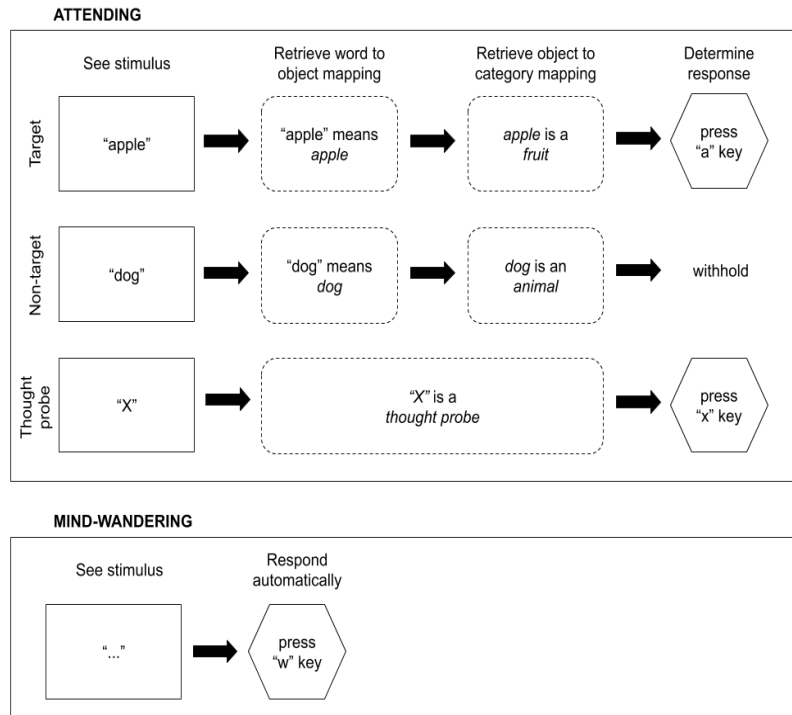


Fig. 1. Graphical depiction of the process followed by the model to respond to stimuli, both in the attending and mind-wandering states.

2.4 Procedure

The model was run 100 times in order to simulate 100 experiment participants. In each run, the model performed four blocks of the semantic SART, each one including 225 trials of one word, for a total of 900 trials per run. Targets (fruit words) were presented in 89% of trials and non-targets (animal words) were presented in 11% of trials, with a random distribution. In each block, a thought probe was presented 120 times (53% of trials), interspersed randomly throughout all trials.

The model's responses to all thought probes were recorded: "x" key presses counted as on-task thoughts and "w" key presses counted as off-task thoughts. In order to measure the frequency of mind-wandering over time, the mean proportion of on-task and off-task responses per block was computed across all participants. Furthermore, response times were recorded to evaluate the model's performance with respect to humans.

The independent variables in the study are the number of blocks of the semantic SART performed, the proportion of target vs. non-target words and the number of thought probes. All three were kept constant for all participants, as explained above. The dependent variable is the proportion of on-task and off-task responses and the response time in each block.

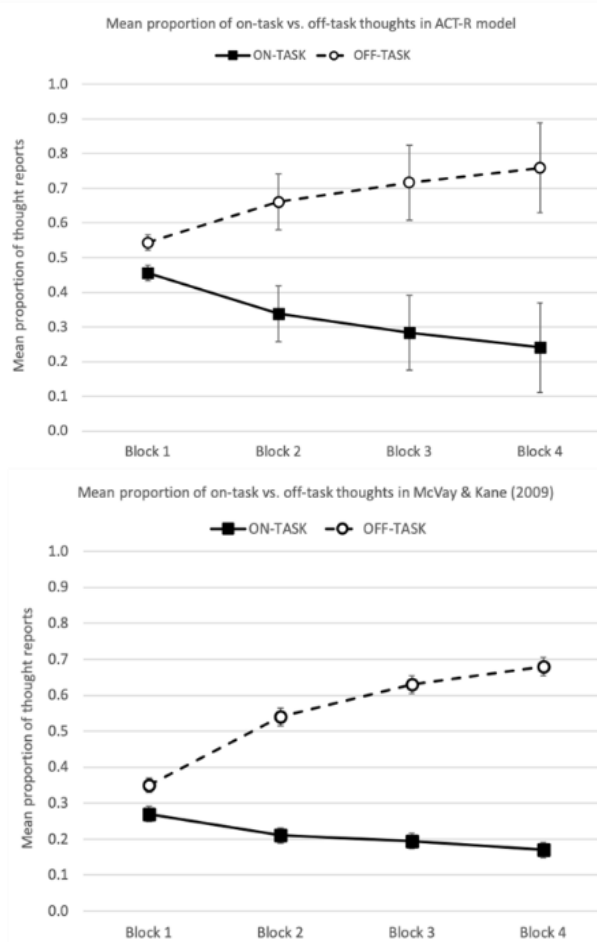


Fig. 2. Mean proportion of on-task vs. off-task thought reports by the model ($N = 100$) compared to human data ($N = 243$). Bars represent standard error. In McVay & Kane (2009), ratios do not add up to one in each block because a third intermediate option for thought probes was available. This was disregarded for purposes of this study.

2.5 Parameter Settings

McVay & Kane's (2009) results show a steep increase in the proportion of mind-wandering with respect to attending over time (see Figure 2). The model assumes that this increase is due to growing activation of the wander goal. In order to achieve this, base-level learning was used with a decay rate of 0.1, to ensure that chunks that were recently activated would be strengthened for a long time.

In addition, spreading activation was used with a value of 2.0 so that activation could be amply spread between associated chunks. The chunks that would bring the model into the attend state were given less activation than their peers. Lastly, all chunks were

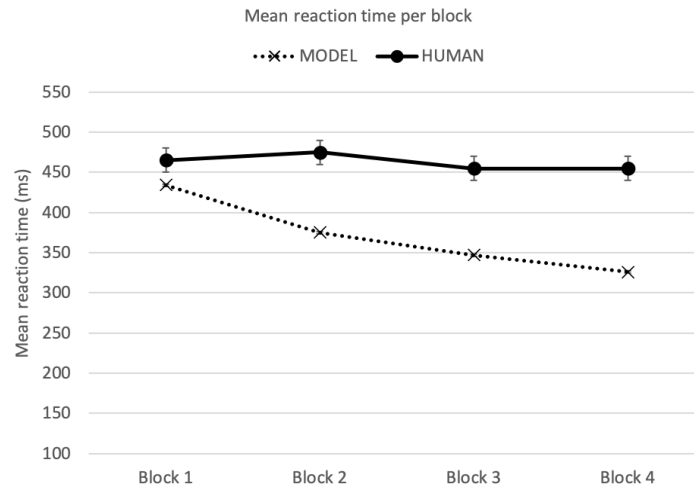


Fig. 3. Mean reaction time in the model compared to the data from McVay & Kane (2009).

initialized with a creation time of 10,000, except for the attend goal chunk, which was given a creation time of zero. In this way, the model starts the task with a similar proportion of mind-wandering versus attending, but then mind-wandering rapidly takes over due to growing activation.

3 Results

The frequency of mind-wandering over time was measured as the proportion of on-task versus off-task thought reports. These can be seen in Figure 2, which compares the ratios obtained by the model to the ratios obtained by McVay & Kane (2009) from human participants. The model visibly approaches the rate of mind-wandering found in human subjects. Off-task thoughts start at an even proportion with on-task thoughts (both near 0.5) in block 1, but then increase steadily up to around 0.75 in block 4. That is to say, the model engages more frequently in mind wandering as time goes by figure 2.

A one-way repeated measures ANOVA was conducted in order to assess the significance of the changes in the proportion of off-task thoughts over time. The proportion of off-task thoughts was found to be significantly different in every block, $F(3, 297) = 531.47, p < 0.05$, generalized $\eta^2 = 0.82$. Post-hoc analyses with a Bonferroni adjustment revealed that all the pairwise differences between blocks were significantly different ($p < 0.05$). This shows that the rate of mind-wandering increases significantly from one block to the next.

Response times (RT) were recorded as an indicator of the model's performance with respect to human participants' in McVay & Kane (2009). Model response times show a decrease from block 1 through block 4, unlike human response times, which remain stable (see Figure 3). While the model's performance improves over time in spite of the growing frequency of mind-wandering, humans do not show such a decisive improvement over time.

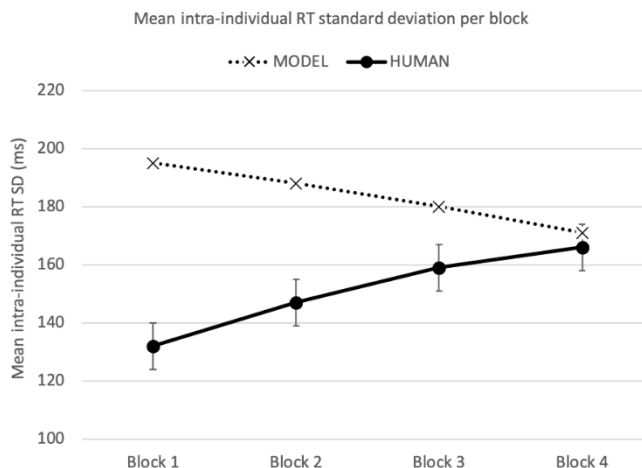


Fig. 4. Mean intra-individual reaction time standard deviation in the model compared to McVay & Kane’s (2009) data.

Performance was further assessed by the mean standard deviation of intra-individual response time per block, which indicates the extent to which response times vary over time for the average participant. In this regard, the model’s behavior differs notably from that of humans: while mean intra-participant SD of response times increases notably over time in humans, in the model it decreases (see Figure 4).

With respect to response time, humans behave much more erratically over time, whereas the model does just the opposite: it shows more consistent behavior from one block to the next.

4 Discussion

The present study examined how the frequency of mind-wandering varies over time during a semantic SART. An ACT-R model was built in order to replicate the study conducted by McVay & Kane (2009) on human participants. Following McVay & Kane’s (2009) findings, it was hypothesized that the frequency of mind-wandering would increase over time.

The model was able to reflect successfully the performance of human participants with respect to the frequency of mind-wandering, showing a steady increase in the proportion of off-task thoughts and a decrease in the proportion of on-task thoughts throughout the experiment. The increase in the proportion of off-task thoughts was found to be statistically significant across all blocks. These results support the hypothesis that the frequency of mind-wandering increases over time during a semantic SART.

These findings provide additional evidence in support of the control failure theory of mind-wandering proposed by Mcvay & Kane (2012). Considering mind-wandering to be the product of a competition between the goal of attending and the goal of retrieving memories, following the model proposed by Van Vugt et al. (2015), renders

a successful model of human performance in a cognitive task. Under this proposal, control failure arises from the memory retrieval goal taking over the attending goal, thus clouding executive control.

This model also provides a plausible mechanism for how attention may decrease over time due to the growing prevalence of mind-wandering, as the memories being retrieved in this process are strengthened by recurrent retrieval cycles. This is in line with the positive correlation of fatigue and mind-wandering frequency, as well as their negative effect on performance, as reported by Risko et al. (2011), Mcvay & Kane (2012), Burdett et al. (2016), Stawarczyk & D'Argembeau (2016) and Robison & Unsworth (2018).

Further assessment of the model was focused on response time. In the model, the reduction in response time is related to the increase in mind-wandering over time. When mind-wandering, the model does not take the time to find the correct response for the given stimulus and just responds automatically, thus reducing mean response time. In human participants, however, response times remain approximately stable across blocks. This implies that the decrease in response time in the model should not necessarily be regarded as a sign of learning, but rather a mere side-effect of the increase in mind-wandering, which is not present in humans.

The model also shows a notable difference with regard to response time variability. Humans tend to be more erratic in their response times as they do more trials—most likely due to fatigue and the growing rate of distraction. However, the model shows more consistent response times across trials. This is also due to the growing frequency of mind-wandering, which leads the model to give more automatic responses more often.

In humans, however, mind-wandering is associated with inconsistent performance, which indicates that the default responses implemented in the model are likely to be an oversimplification of the response mechanism used in human cognition.

These discrepancies indicate that the model is unable to capture the full complexity of human performance. Therefore, suggestions for future research involve developing the model further, while keeping the growing trend in mind-wandering over time. For instance, the default response mechanism in mind-wandering could be extended to accommodate a range of responses, some of which may involve some degree of cognitive processing—thus increasing response time. A more complex system of memory retrieval could be used for mind-wandering, where some memories may be more pervasive than others. The model's vocabulary can also be extended with a larger set of words and categories. Furthermore, thought probes could consider the option range given in McVay & Kane (2009). This would bring the model closer to humans' more variable performance.

Finally, this modelling approach of mind-wandering through competing goals should be applied to other cognitive tasks. It will then be possible to examine how the frequency of mind-wandering and response time vary in those tasks and to compare that performance to the semantic SART.

The utility of studying the mechanisms of mind-wandering is in deepening our understanding of how human cognition and executive control behave in task-oriented scenarios. Applications of this research are especially useful in educational and work environments, where enhancing learning and productivity by maximizing the time spent focusing on the task as opposed to mind-wandering is key to meeting performance

goals. Building cognitive models of mind-wandering in various tasks may reveal which task attributes make participants more or less prone to off-task thoughts, which may aid teachers or managers in designing tasks that facilitate performance. Similarly, studying individual differences in mind-wandering may prove extremely useful in assisting low-performing individuals to improve performance by enhancing task focus. In sum, expanding our insight into mind-wandering could largely advance the way humans learn and work.

References

1. Bastian, M., Sackur, J.: Mind wandering at the fingertips: automatic parsing of subjective states based on response time variability. *Frontiers in Psychology*, 4 (2013)
2. Battig, W.F., Montague, W.E.: Category norms of verbal items in 56 categories a replication and extension of the Connecticut category norms. *Journal of experimental Psychology*, 80(3) (1969)
3. Burdett, B.R., Charlton, S.G., Starkey, N.J.: Not all minds wander equally: The influence of traits, states and road environment factors on self-reported mind wandering during everyday driving. *Accident Analysis and Prevention*, 95, pp. 1–7 (2016)
4. Cheyne, J.A., Carriere, J.S., Smilek, D.: Absent minds and absent agents: Attention-lapse induced alienation of agency. *Consciousness and Cognition*, 18, pp. 481–493 (2009)
5. Christoff, K., Irving, Z.C., Fox, K.C., Spreng, R.N., Andrews-Hanna, J.R.: Mind-wandering as spontaneous thought: a dynamic framework. *Nature Reviews Neuroscience*, 17(11), pp. 718–731 (2016)
6. McVay, J.C., Kane, M.J.: Conducting the train of thought: working memory capacity, goal neglect, and mind wandering in an executive-control task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(1), pp. 196 (2009)
7. Mcvay, J.C., Kane, M.J.: Drifting from slow to “doh!”: Working memory capacity and mind wandering predict extreme reaction times and executive control errors. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 38(3), pp. 525–549 (2012)
8. Mooneyham, B.W., Schooler, J.W.: The costs and benefits of mind-wandering: a review. *Canadian Journal of Experimental Psychology/Revue Canadienne de Psychologie Expérimentale*, 67(1) (2013)
9. Risko, E.F., Anderson, N., Sarwal, A., Engelhardt, M., Kingstone, A.: Everyday attention: Variation in mind wandering and memory in a lecture. *Applied Cognitive Psychology*, 26, pp. 234–242 (2011)
10. Robison, M.K., Unsworth, N.: Cognitive and contextual correlates of spontaneous and deliberate mind-wandering. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 44(1), pp. 85 (2018)
11. Seli, P., Cheyne, J.A., Xu, M., Purdon, C., Smilek, D.: Motivation, intentionality, and mind wandering: Implications for assessments of task-unrelated thought. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41(5), pp. 1417–1425 (2015)
12. Smallwood, J., Schooler, J.W.: The science of mind wandering: empirically navigating the stream of consciousness. *Annual Review of Psychology*, 66, pp. 487–518 (2015)
13. Stawarczyk, D., D'Argembeau, A.: Conjoint influence of mind-wandering and sleepiness on task performance. *Journal of Experimental Psychology: Human Perception and performance*, 42(10), 1587 (2016).
14. Van Vugt, M. K., Taatgen, N.A., Sackur, J., Bastian, M., Borst, J., Mehlhorn, K. Modeling mind-wandering: a tool to better understand distraction. In: *Proceedings of the 13th International Conference on Cognitive Modeling*, pp. 252–257 (2015)

Traffic Prediction Architecture based on Machine Learning Approach for Smart Cities

Marco C. Hernández Inzunza², Luis Hernández Robles³, Miriam A. Carlos Mancilla¹,
Emmanuel López Neri¹

¹ CIIDETEC-Universidad del Valle de México Jalisco,
Mexico

² Universidad Autónoma de Sinaloa,
Departamento de Ciencias en Computación,
Mexico

³ Instituto Tecnológico de Culiacán,
Departamento de Ciencias en Computación,
Mexico

{marcohdes94i, lu97is}@gmail.com,
{miriam.carlos, emmanuel.lopezne}@uvmnet.edu

Abstract. It is common to see more and more people dealing with traffic due to excessive population growth in cities. The traffic has become one of the main topics of interest is used for smart cities approaches. Accordingly, this study presents the development and implementation of a new architecture to predict the traffic flow in a city and the strategy used in this scheme. The proposal considers the use of machine learning, computer vision, deep learning, and neuronal networks to implement the solution. The architecture is composed of four main components; (1) A Machine Algorithm System (*MASY*) that works using pattern recognition of the traffic; (2) A Neuronal Artificial System (*NASY*) helps with the traffic classification; (3) A Web user application (*WeUsAP*) to present the results, and process entry user data and finally, (4) A Car Counting Wizard (*CCW*) video capturing component based on computer vision to create a statistical analysis of vehicles. Consequently, some results and comparatives are presented in order to obtain an analysis of data accuracy.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, Computer Vision, Artificial Neural Networks (ANNs); Vehicular Traffic prediction; Smart Cities; Module architecture.

1 Introduction

In the last decade, cities have shown an accelerated growth of vehicular traffic linked with the increase in population and the subsequent road traffic demand [24] but in same

way multiple traffic control strategies have continued introduced with the similar accelerated advance of technology. Although some of these strategies use a proactive approach, is required that quickly adapts to changes in use of the traffic road. Statistically, a person wastes an average of 42 hours a year driving in the traffic during peak hours [1]. The elements used to design an optimal vehicular traffic condition control are vehicular flow capacity of a road and efficient urban transportation planning. Most accidents are the result of a wrong driving decision that negatively affects the traffic condition. Some of the most common variables to consider for traffic control are the fast population growth factor, public transportation network, accessible routes, infrastructure degradation, among others.

In this proposal, machine learning, deep learning, and computer vision are used. The proposal is interested in traffic classification using an Artificial Neural Network (ANN) and the implementation of an architecture that considers four components is presented. First, A Machine Algorithm System (MAS) for pattern recognition, Second, A Neuronal Artificial System (NAS) for data prediction using a machine learning strategy; Third, A Web User Application (WeUsAP) to communicate the results to users; Finally, a Car Counting Wizard (CCW) based on computer vision to create a statistical analysis of vehicles. The proposal does not require GPS.

This paper is organized as follows. Section 2 presents the proposed architecture for traffic predictions; section 3 discusses results and finally, some conclusions are presented.

2 Related Works

Many proposals work with branches of Artificial intelligence (AI) such as machine learning, deep learning, or data analysis, in one effort to solve problems easily and predict future system characteristics. The machine learning approach helps with the design and implementation of a system that learns automatically. For instance, in [20] a comparison between regression models based on a machine learning approach to predict traffic using a virtual network is presented, in which authors describe better accuracy results using real-life data. While in [3] a traffic prediction method using machine learning is presented, the proposal works with LoRa and a traffic predetermined algorithm. The authors do not present any results or proofs.

In [7] the authors proposed five Machine Learning (ML) algorithms for traffic classification based on IP (Internet Protocol). The focus of this proposal is to demonstrate the benefits of computational performance as a significant metric for traffic classification. In [25] the authors present a traffic prediction method for an application. In this work, a statistical method is used, however, it is not intended to create a new methodology or strategy to improve the traffic, only the results of the application are presented.

Deep Learning is a way to automate a prediction analysis. For instance, a deep learning model for traffic prediction is proposed by [14]. The authors present a model that works with local data introduced manually to the system. In addition, a deep belief network is used for processing. Another work presented in [23] proposed data analytics using a machine learning algorithm to provide an adaptive model for traffic predictions. Despite the proposal presents good predictions for traffic, the model spends a lot of

energy for monitoring and data processing. In [8] it is presented a binary neuronal network algorithm for traffic flow prediction. The prediction is made by short terms intervals, to visualize how the predicted value can affect the making decisions or actions. The algorithm requires to use a GPS (Global Positioning System) after they send information about places, traffic situations, objects, user preferences, among others. The information is stored and then sent to users. Some definitions for Speed and Traffic are well known in the community, however, to avoid ambiguity, in this proposal flow, and density are defined as follows. *Flow* measures how many cars pass a given point on the road at a given time t . *Density* is defined as the number of cars at one point of any road x at time t . In this proposal, the equation 1 is considered to calculate the traffic flow, presented in [2]:

$$Q=K \times V, \tag{1}$$

where Q represents vehicular flow, K is the vehicular density, and V represents the speed. Other applications use the flow and density to calculate better routes, such as Waze [6] and INRIX [1], however, the accuracy of the predictions is provided by the information of users. Then, users can make decisions about alternative routes and report current traffic situations. In order to control the data and process the incoming information of the environment. According to John McCarthy [4], AI is “the science and engineering of making intelligent machines, especially intelligent computer programs”.

For instance, in [9] the authors present an evaluation between flow density and motorway problems using an IA approach, the weather is considered in this proposal. A cross-validation strategy for traffic estimation is presented in [10]. In [11, 12] different strategies for traffic estimation are evaluated, comparisons are made, and the author evaluates the use of a neuronal network.

3 Architecture for Traffic Prediction

The architecture for data prediction proposed is presented in Fig. 1. The architecture does not require GPS to work. The architecture is composed of four main components, MASY, NASY, User App, and the CCW, each one is described in detail.

3.1 Machine Algorithm System (MASY)

MASY is a component designed to predict cars and to obtain an analysis. This module uses a multiple linear regression algorithm optimized with a gradient boost regressor technique for error correction as well-known as a Multiple Linear Regression (MLR) that explains the relationship between one dependent variable and multiple independent variables written in Python.

This technique describes the relationship between days, hours, time, number of cars of every day, and the traffic. MASY works with a preprocessing stage, in which the data is classified in a database made in PostgreSQL. Information from Australian Smart City [13] is used to verify the behavior of car prediction in MASY.

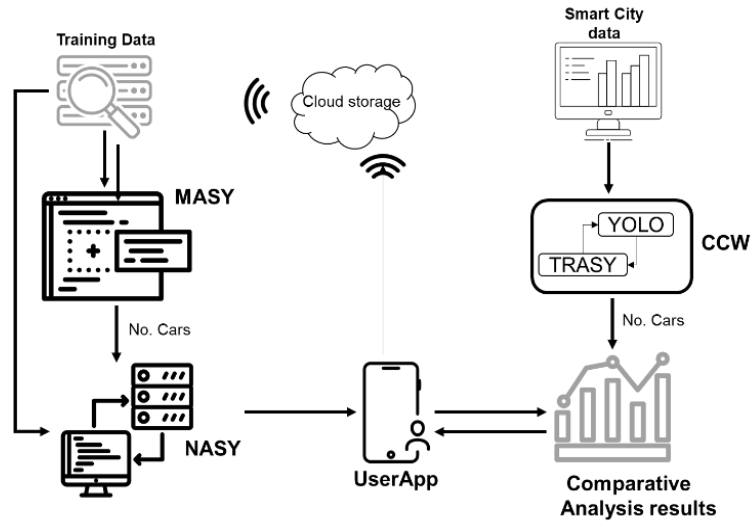


Fig. 1. Proposed architecture for traffic flow prediction.

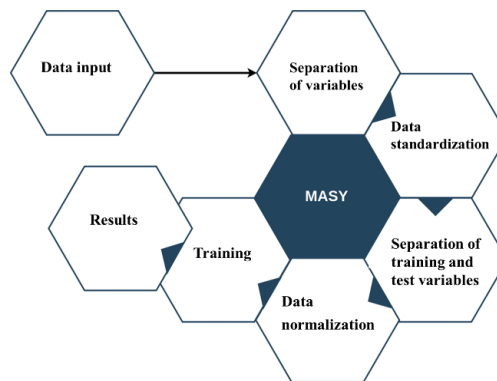


Fig. 2. MAS Y's module tasks.

Additional information is considered for real time estimation, such as flow (cars per time), road conditions, trails, among others. This information is difficult to obtain from people but easy to calculate from additional devices such as cameras, sensors, or monitoring devices.

In this proposal the data was collected through cameras installed during short periods during rush and normal hours, the data stored was used to train MAS Y. The main purpose of MAS Y module is to process the information through a machine learning strategy. An overview of MAS Y's process is presented in Fig. 2. Once MAS Y receives the data, is divided into six tasks, the first one consist of dividing independent and dependent variables; the second task is the standardization of data in which the data is transformed to a normal distribution; In the third task, the training dataset and test are made; In the fourth task, the data is normalized; Fifth task is the data training and finally results are presented.

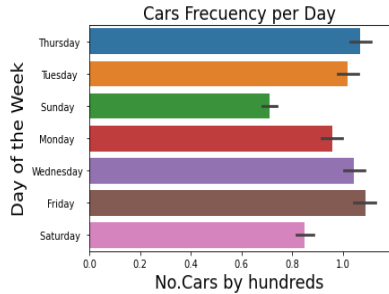


Fig. 3. Number of cars each day for a week and the days of the week.

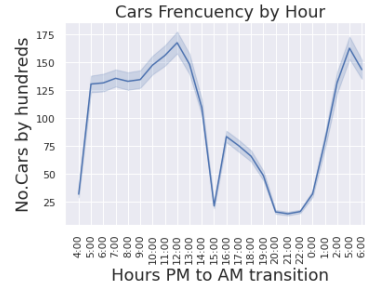


Fig. 4. Number of cars each day for a week and the days of the week.

Traffic is calculated by equation 1. A data visualization analysis of the dataset helps to understand how the city’s dataset results are compared with real data of the city. Fig. 3 shows the calculation of the car behavior of each day for a week.

As we can notice, there was more car traffic on Thursdays and Fridays, followed by Wednesdays, and Tuesdays with a similar number of cars, and Sunday with a minor number of cars in the week. In Fig. 4 the amount of cars is displayed during different hours of the day. The graph represents the average number of cars measured during a week at different times. Once this information is obtained, is sent to NASY.

3.2 Neuronal Artificial System (NASY)

NASY is used for traffic classification. Its objective is to perform a traffic prediction and to inform whether there will be traffic or not based on real estimations. NASY uses traffic flow theory [2], it was trained using 60,000 samples from Australian Smart city data. NASY receives the data from MASY, processes the information using a neural network, and supervised learning to predict the traffic behavior.

The data entries for NASY are speed, flow, and density of cars. The *Speed* is defined as a parameter according to the kind of road and traffic regulations; the density (*D*) is calculated using equation 2; other variables are considered, such as date, day, street type, suburbs, and the location for an accurate prediction. A pre-processing procedure is done equal to MASY:

$$D=N/d, \tag{2}$$

where *N* is the car number, and *d* is the distance of the road, avenue, or trail. An overview of NASY’s process is presented in Fig.5. NASY is composed of seven tasks. MASY and NASY perform the same procedure for the first four tasks. The fifth task is the simulation of the neuronal network; the sixth task performs the data training called ReLU (Rectified Linear Unit) and finally, in the seventh task results are presented.

Fig. 6 shows the ANN representation used in NASY, where green nodes represent the input data, mine while the blue ones present the hidden layers in the neuronal network. NASY uses a deep learning model built using Tensorflow[15] and Keras[16].

NASY is configure using 60 neurons in the input layer and 30 neurons in every two hidden layers. According to [17] these layers are enough to process the information.

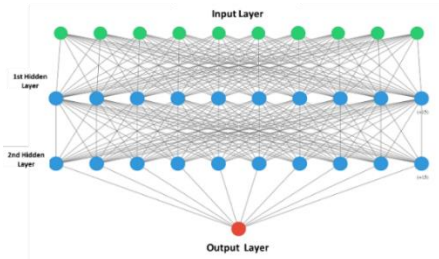


Fig. 5. NASY's process.

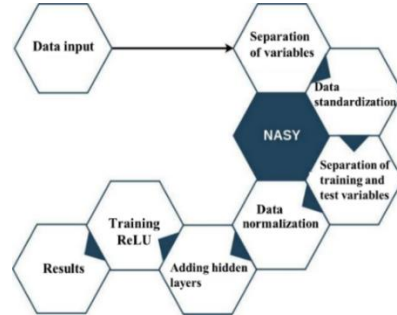


Fig. 6. ANN representation used in NASY.

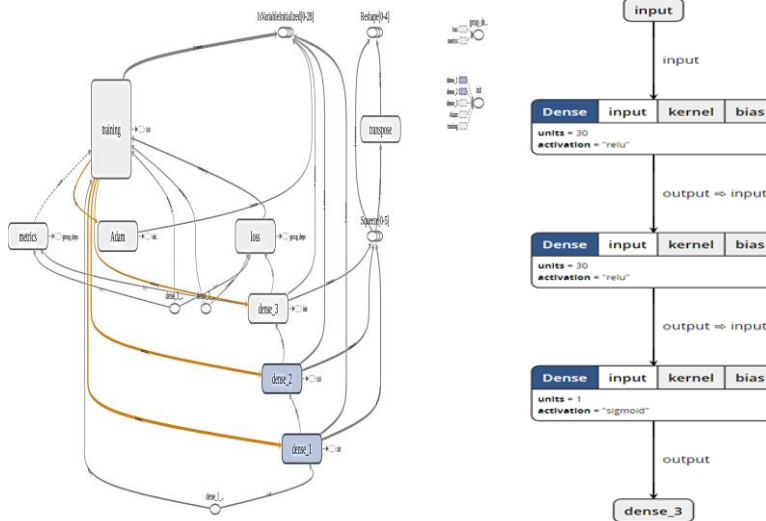


Fig. 7. Learning process of NASY based on Tensorflow and the H5 model visualization of the ANN used in NASY.

In [18, 19] the authors considered a batch size 32, epochs are defined as 100 which is the optimization parameter obtained after finish running the test. In the proposal was implemented an exhaustive grid search where the size of the batch, epochs, and optimization functions was compared. The learning algorithm proposed is based on backpropagation.

The reLU function only takes positives value and has been proved that reLU shows an engaging training behavior in experimental practice presented in [5]. The data stored by NASY are Data, Road_name, Location, Suburb, Speed_Limit, Direction, Time, Avg_speed, Max_speed, number of cars (cars), K for the vehicular flow, day of the week, Maximum vehicular flow (MaxK), Month, Year, and Type of road.

The model obtained from Tensorflow is presented in Fig. 7, shows the NASY learning process and provides an H5 (Hierarchical Data Format) file, designed to store large amounts of data to be used in the user interface explained in the next section.

The image shows a web form for traffic prediction. It has two columns of input fields. The left column contains: Year (2014), Direction (North), Hour (24hrs Format), Street Type (Street), and Average Speed limit km/h. The right column contains: Month (January), Speed Limit km/h (street), Day (Monday), No. Cars, and Max Speed Km/H. A green button labeled 'RESULT' is positioned at the bottom left of the form area.

Fig. 8. Web user platform for prediction of the traffic.

The H5 model is composed of the data entries for the ANN, density (number of connected nodes), *unit* (defined as 30, these are the entries to hidden layers), and finally, the *activation function* (reLU for hidden layers and the sigmoid function for outputs).

3.3 Web User Application (WeUsAP)

The Web platform is the third component in the architecture and is useful to visualize the results obtained from NASY and MASy. The results help us to improve the models and test the models using different values. The platform uses Flask (a micro web framework for Python), a database based on PostgreSQL, and Heroku to enable the developers to build, run, and operate applications entirely in the cloud. The platform works as a simple web application; it has a friendly graphical interface measured by metrics given by Google called *Material Design*. Fig. 8 shows the user interface for the traffic prediction. The users use this application to make a traffic prediction. The required input data are the day, month, year, speed limit, type of road, among others. The information introduced pretend to be known by people who use the platform and is easy to define without having a specialized knowledge of vehicular traffic.

After making a traffic prediction, every user has to answer the next question. Do you think the car prediction was right? The information is stored in the database to make a posterior analysis about making improvements or adjustments to the algorithms for a better car prediction.

3.4 Car Counting Wizard (CCW)

CCW is the component proposed to automate the car counting to compare real data with data predictions by NASY. This component integrates two submodules Tracking System (TRASY) [21]. TRASY is an open code algorithm written in python that uses OpenCV for image manipulation. The main objective of TRASY is counting objects that pass through a green frame. 2) You Only Look Once (YOLO) [22]. YOLO is an open-source project written in C and CUDA for real-time object detection. YOLO classifies images into the types of objects identified, such as motorcycle, bus, cars, trailers, even people, and stores the information in a database.

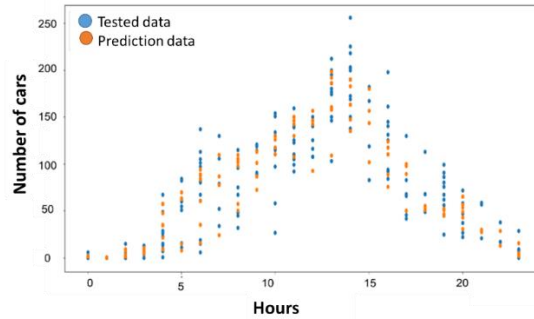


Fig. 9. MASY's prediction results.

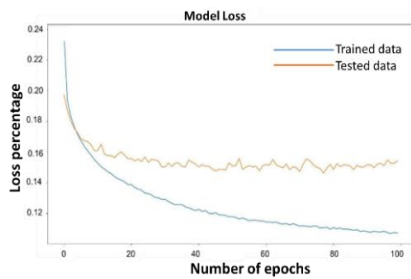


Fig. 10. MASY's prediction results.

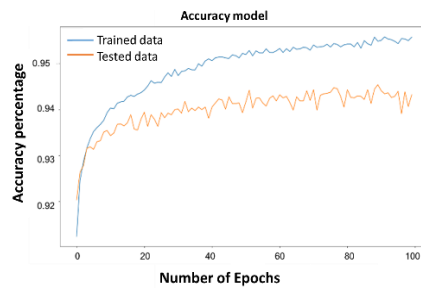


Fig. 11. MASY's model accuracy.

The result information obtained in this module is used to compare the results between the proposed architecture and the obtained by CCW.

4 Results

An original trained data set is represented by DS_i . A subset DS_i' is randomly selected $DS_i' = rand(DS_i)$; The *tested dataset* DS_T is the remaining dataset classified as true or false once the outputs of training are given. Based on the DS_i , MASY is able to predict the average number of cars that could appear in a specific hour of one day. In Fig. 9, the results achieved for traffic predictions made by MASY are presented. Blue dots represent the DS_T , and yellow dots show the predictions. The car prediction presents a 90% of accuracy of a day.

Fig. 10 shows the relationship between the input data (trained data DS_i) and the results obtained. The loss of precision in the results of the proposed model (NASY) is not significant and the model is able to adapt to the missing information to obtain good results. The number of epochs is set to 100.

The prediction is still being stable and maintains the accuracy (Fig. 11), the blue line represents the trained dataset DS_i and yellow line describes the test dataset DS_T . The performance of the algorithm is good and it is capable of adapting to obtain good predictions whether the information is not complete or if the information is loss.

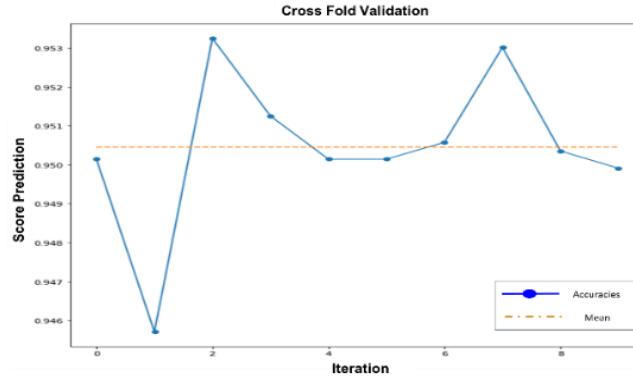


Fig. 11. Architecture cross validation for urban traffic predictions.

After training and testing the model, a cross-validation technique is made in order to observe and analyze the behavior of the architecture. The goal of cross-validation is to verify the model's ability to predict new data (Fig. 12). The testing was configured to 10 iterations. It is possible to observe that as time passes the accuracy of the predictions increase. The architecture can learn from the above information to increase its knowledge and maintain the accuracy of the data.

The stability of the architecture was reached in only 2 iterations. A Hyper-parameter optimization process was applied to the model to ensure the parameters. The parameters were defined as batch size: 25 vs 32, 100 epochs vs 500, and Adam algorithm vs Rmsprop. This test was applied in a grid-search cross-validation. The best parameters according to the test were: batch size": 32, "epochs": 100, "optimizer": "Adam".

Finally, as seen in the related works, there are multiple strategies to address such a complex and highly relevant issue today, such as traffic prediction. However, current proposals are only dedicated to covering a single specific point, that is, counting cars, verifying a certain position or testing the effectiveness of an algorithm. In this proposal, we work with 4 main components. The first is a module for the identification of the count of the cars and what type they are; the second module valid and gather the information of the hour, the day, the traffic zone and components such as the type of circulation. The third module learns the parameters and compares using real data with those obtained in the model to learn and be able to make a more accurate prediction of the amount of traffic per time and day, and kind of road. Finally, the four module is an application that allows users to interact with the application to get more real and achievable data for anyone. These modules present a whole comprehensive system that uses machine learning, a neural network, among other techniques. This is a prototype that can be easily extended to be used in many places of a city, as part of a future work.

5 Conclusion

The proposal of architecture for traffic prediction based on neural networks and strategies such as machine learning, computer vision, and deep learning was made. The

proposed architecture works relatively fast and in a productive manner to obtain promising results if the information used is real.

The neural network is able to obtain a better prediction and to improve the learning during the processing stage with high accuracy. Users can test the algorithm and access the architecture using real information about their location, country, and traffic situation. The web application uses achievable information for any user, and it is easy to know. Finally, a comparison was made between the data obtained from the architecture and those obtained directly from CCW.

The results obtained present that the prediction model closely resembles the counting of cars made by CCW, the approximation was quite good, and the model improves as time passes, due to its learning process. For future implementations, traffic prediction should be improved, and the simplest model should be made. In addition, this proposal will be linked to an urban traffic platform (CiudadelaSIM) to observe and analyze the behavior of vehicles based on real data. For future implementations, it is possible to use a “Low power” device using a raspberry PI. A standalone device can be used to make this implementation useful and easy to supervise. This proposal can use a device to interact with a mobile application and the request to make predictions can be made by a http request, in addition the architecture and the trained data can be exported to any mobile device.

References

1. INRIX: Los Angeles Tops INRIX Global Congestion Ranking (2017)
2. Fernández-Aguilera, R.: Elementos de la teoría del tráfico vehicular. Fondo Editorial. Santiago, Chile, pp. 216 (2011)
3. Sagarika Verma, Sayali Badade: Traffic prediction using machine learning. In: Proceeding of National Conference on Machine Learning (2019)
4. McCarthy, J.: What is Artificial Intelligence?. Stanford University, Computer Science Department, pp. 1–15 (2007)
5. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Gordon, G.J., Dunson, D.B., Dudík, M. (eds.), AISTATS of JMLR proceeding, 15, pp. 315–323 (2011)
6. Waze: <https://www.waze.com/business/#boat1> (2006)
7. Williams, N., Zander, S., Armitage, G.J.: A Preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. Computer Communication ACM SIGCOMM, 36(5), pp. 1–15 (2006)
8. Zheng, W., Lee, D.G., Asce, M., Shi, Q.: Short-term freeway traffic flow prediction: bayesian combined neural network approach. Journal of Transportation Engineering, 132(2), pp. 1–8 (2006)
9. Florio, L., Mussone, L.: Neural-network models for classification and forecasting of freeway traffic flow stability. Control Engineering Practice, 4(2), pp. 153–164 (1996)
10. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence (IJCAI), 2. pp. 1137–1143 (1995)
11. Van Hinsbergen, C.P., Van Lint, J.W.C., Sanders, M.: Short term traffic prediction models. In: 14th World Congress on Intelligent Transport Systems, ITS World Congress, pp. 1–7 (2007)

12. Zheng, W., Lee, D.G., Asce, M., Shi Q.: Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of Transportation Engineering*, 132(2), pp. 1–8 (2006)
13. Kaggle: Your home for data science. <https://www.kaggle.com/> (2018)
14. Yuhan Jia, Jianping Wu, Yiman Du: Traffic speed prediction using deep learning method. In: *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1217–1222 (2016)
15. TensorFlow: <https://www.tensorflow.org/?hl=es> (2018)
16. Keras Documentation: <https://keras.io/> (2018)
17. Panchal, G., Ganatra, A., Kosta, Y., Panchal, D.: Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers. *International Journal of Computer Theory and Engineering*, 3(2), pp. 332–337 (2011)
18. Buitick, L., Louppe, G., Blondel, M., Pedregosa, P., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Glor, J., Holt, B., Varoquax, G.: API design for machine learning software: experiences from the scikit-learn project. In: *Workshop: Languages for Data mining and Machine Learning*, pp. 108–122 (2018)
19. Scikit-learn: Tuning the hyper-parameters of an estimator. Scikit-learn developers, documentation section 3.2. (2019)
20. Nikravesh, A.Y., Ajila, S.A., Lung, C., Ding, W.: Mobile Network Traffic Prediction Using MLP, MLPWD, and SVM. In: *IEEE International Congress on Big Data (Big Data Congress)*, pp. 402–409 (2016)
21. Nikishaev, A.: Traffic counting example based on OpenCV object detection with background subtraction (2018)
22. Redmon, J.: YOLO: Real-Time Object Detection (2018)
23. Morales, F., Ruiz, M., Gifre, L., Contreras, L.M., Lopez, V., Velasco, L.: Virtual network topology adaptability based on data analytics for traffic prediction. In: *IEEE/OSA Journal of Optical Communications and Networking*, 9(1), pp. A35–A45 (2017)
24. Department for Transport: Road traffic forecast. Department for Transport’s National Transport Model (NTM), *Moving Britain Ahead* (2018)
25. Hinsbergen, C., Lint, J.W.C., Sanders, F.: Short term traffic prediction models. In: *14th World Congress on Intelligent Transport Systems* (2007)

Utility-Pole, Nuts and Cross-Arm Visual Detection, for Electric Connections Maintenance Robot

José Luis Gómez Torres¹, Saúl Martínez Díaz¹, Iris Iddaly Méndez Gurrola²,
Alejandro Israel Barranco Gutiérrez³, Kyoichi Tatsuno⁴

¹ TecNM La Paz,
Mexico

² Universidad Autónoma de Ciudad Juárez,
Mexico

³ Cátedras CONACyT-TecNM Celaya,
Mexico

⁴ Meijo University, Nagoya,
Japan

israel.barranco@itcelaya.edu.mx

Abstract. In this work, a computer vision system used by a utility-pole maintenance robot is presented. The objective of the system is that the robot repairs connections between cables and screws on the utility-pole. To accomplish this task, it uses a pair of cameras that detects and locates the utility pole, crosstree, nuts, and screws. The detection is based on color, shape and dimensions analysis, by using conventional RGB cameras. On the other hand, the challenges facing this approach include: utility-pole detection though in the same scene, some objects with the same color (concrete) appear; flexibility in the distance between cameras and utility-pole for detection and localization, about 2-10 meters range and slope estimation of utility-pole and crosstree. The experiments show that it is possible to recognize and locate the interest pieces with satisfactory results and they encourage to expand the current system capabilities.

Keywords: Utility Pole, Maintenance Robot, Object Recognition, Object Location.

1 Introduction

In the last fifty years, developments in robotics have grown exponentially (Cianchetti, 2014), thus these systems are used to improve the life quality of humans in different environments.

In this context, a maintenance robot who repair cable connections on utility-poles (UPs) is been constructed at Tatsuno Laboratory at Meijo University to avoid the risk when electric technicians make this work. As a human, an android needs to recognize the UP, cables, screws, nuts, thus dimensions and locations (Y. He, 2008), (Barranco-Gutierrez, Martínez Díaz, & Gomes Torres, An Approach for Utility Pole Recognition in Real Conditions, 2014) to execute this task. Due to the grown-on algorithms development for processing digital images, it estimated possible that their techniques could perform the mentioned tasks.

In literature, there are many techniques for object recognition, but with the searching it was observed that there are few object recognition techniques outdoors and practically nil in recognizing UP. Concerning to objects location in 3D coordinates, it observed that exist many probabilistic algorithms, but they have different restrictions; so, the use of stereo vision system was selected because it has fewer restrictions to calculate positions in 3D than monocular systems, some experiments were performed using this technique (Barranco-Gutierrez, Martínez Díaz, & Gomes Torres, An Approach for Utility Pole Recognition in Real Conditions, 2014).

This work used Mexican UPs to experiment in thus detection algorithms under solar lighting. The system detects the UP in the range of 15 to 3 meters from camera to the target. This paper shows improvement to the system for example: it works with a Japanese element UP, the crosstree, screws, nuts and the holes (where the screws are inserted) are detected and located. This, using only one set of color stereoscopic vision. In Fig. 1 is shown the prototype of this research.

1.1 Related Work

Although there are many object recognition techniques. There are few concrete proposals to solve the problem of the recognition of Electric Poles due to the specificity of the problem and the challenges involved in solving it. For example, the large number of pole-like objects on the street such as PVC pipes, telephone poles, traffic signs, and cable TV poles. In (Zhang, et al., 2018), the authors proposed using a deep learning-based method for automatically mapping roadside utility poles with crossarms (UPCs) from Google Street View (GSV) images.

The method combines the state-of-the-art DL object detection algorithm (i.e., the RetinaNet object detection algorithm) and a modified brute-force-based line-of-bearing (LOB, a LOB stands for the ray towards the location of the target [UPC at here] from the original location of the sensor [GSV mobile platform]) measurement method to estimate the locations of detected roadside UPCs from GSV. Also, in (Watanabe, 2018) a deep network is used to detect utility poles. The authors found that the electric poles can be detected with an average precision (AP) of 72.2%. Their results demonstrate the operational feasibility of the autonomous electric pole inspection system that implements a deep network-based object detector.

Reference (Y. He, 2008) proposes an approach for automatically inputting handwritten Distribution Facility Drawings (DFD) and their maps into a computer, by using the Facility Management Database (FMD).

The recognition method makes use of external information for drawing/map recognition. It identifies each electric-pole symbol and supports cable symbol on drawings simply by consulting the FMD, but the system requests manual feedback from

the operator in online stage. Reference (Barranco-Gutierrez, Martínez Díaz, & Gomes Torres, An Approach for Utility Pole Recognition in Real Conditions, 2014) presents an algorithm for the recognition of similar electrical poles from an aerial image by detecting the pole shadow. One pole is used as a template (already identified by a human operator) for the algorithm.



Fig. 1. Prototype in Tatsuno Laboratory illustrates the UP, manipulator robot and stereo vision system.

The algorithm includes feature extraction, utility pole position determination and elimination of redundant candidates. This work references the ideas from other works in a similar sense, the papers (Nakajima, 1995) (Cetin, 2009) detail, although the color distribution changes under different lighting conditions, some aspects of its structure turn out to be invariants. In the works (Igor, Amit, & Ehud, 2013) (Berwick & Lee, 1998), it explains that the sky has always been the crucial element in modeling the

background of an outdoor scene. The position of the sun during the day gives a different impact on the sky color. The method presented in (Halawani & Sunar, 2010) is used to compare our method although the acquisition system is different. In other outdoor computer vision applications such as autonomous vehicles. It also presents the challenges of lighting from the sun, the large number of different objects, shadows, reflections, oclusions and the high similarity between some objects (Janai, Güney, Behl, & Geiger, 2020).

2 Proposed Methodology

The training and on-line checking stages use a very similar methodology; the unique difference in the training stage, it consists in the possibility to monitor the results and modify some constants values for tuning the system according to the problem conditions. The procedure is based on three basic aspects of computer vision: Color, shape and dimensions. This is, the implemented techniques are: thresholding on HSV scheme to filter by color, Hu invariants to describe shapes and stereo vision to measure objects dimensions. All of them, combined to detect and locate utility-poles with its crosstree, nuts and screws. Below, the approach steps to detect the utility-pole, are listed:

- I. Stereo image acquisition,
- II. HSV conversion,
- III. Color thresholding,
- IV. Canny edge detector,
- V. Lines detector,
- VI. AND operation,
- VII. Hu invariants, slope and area calculus,
- VIII. Classification stage,
- IX. Stereo matching and utility pole localization.

I. Stereo Image Acquisition

To take photographs in the stereo vision system, it must be calibrated (Barranco-Gutierrez, Martínez-Díaz, & Gomez-Torres, *Visión estereoscópica con Matlab y OpenCV*, 2018). This means that it must to know, the vector between both camera reference technically, to have the fundamental and essential matrices of stereoscopic system as indicate (Zisserman & Hartley, 2003), (Zhang Z. , 2000). After this, we can start training or on-line work. In the image acquisition system, a color image is received from the left and right cameras with resolution $M \times N \times 3$ (M for rows, N for columns and three-color components) with $b = 10$ bits of resolution for each pixel and the stereo image is expressed in both sides $I^s = I^{\text{left}}$ or $I^s = I^{\text{right}}$ in equations in equation (1):

$$I^s(x, y, z) \in \{0 \leq \mathbb{Z} \leq 2^b\}, \quad (1)$$

where x, y, z are indexes that locate to each pixel and \mathbb{Z} is the set of integer numbers. $z = 0$ refers to the image's RED component, $z = 1$ the image's GREEN component and $z = 2$ the image's BLUE component.

II. HSV Conversion

In this step, the program converts the left and right images to HSV space as indicates by equations (2, 3, 4) in order to obtain a color information in the axis H:

$$H(x, y) = \begin{cases} \text{undefined} & \text{if } \max(R, G, B) = \min(R, G, B), \\ \frac{\pi(G - B)}{3(\max(R, G, B) - \min(R, G, B))} & \text{if } \max(R, G, B) = R \text{ and } G \geq B, \\ \frac{\pi(G - B)}{3(\max(R, G, B) - \min(R, G, B))} + 2\pi & \text{if } \max(R, G, B) = R \text{ and } G < B, \\ \frac{\pi(G - B)}{3(\max(R, G, B) - \min(R, G, B))} + \frac{2\pi}{3} & \text{if } \max(R, G, B) = G, \\ \frac{\pi(G - B)}{3(\max(R, G, B) - \min(R, G, B))} + \frac{4\pi}{3} & \text{if } \max(R, G, B) = B, \end{cases} \quad (2)$$

$$S(x, y) = \begin{cases} 0 & \text{if } \max(R, G, B) = 0, \\ 1 - \frac{\min(R, G, B)}{\max(R, G, B)} & \text{if } \max(R, G, B) \neq 0, \end{cases} \quad (3)$$

$$V(x, y) = \max(R, G, B), \quad (4)$$

where $R^s(x, y) = I^s(x, y, 1)$, $G^s(x, y) = I^s(x, y, 2)$, $B^s(x, y) = I^s(x, y, 3)$, the function $\max(R, G, B)$ retrieves the maximum value among its three arguments and in contrast, $\min(R, G, B)$ obtains the minimal.

III. Color Thresholding

Because the H component of the RGB conversion to HSV contains the color information of objects in the image and component V describes the lighting in qualitative terms, a filter with the desired color and lighting is constructed using a range within a Gaussian distribution function as described in (5):

$$BW^s(x, y) = \begin{cases} 1 & \text{if } H^s(x, y) < \mu_h \text{ AND } S^s(x, y) < \mu_s \text{ AND } V^s(x, y) < \mu_v, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

IV. Canny Edge Detector

To find contours from the thresholded $BW^s(x, y)$, the Canny method is used, in order to delimit objects with the same color, (6):

$$C^s(x, y) = \text{Canny}(BW^s(x, y)). \quad (6)$$

V. Lines Detector

Then, in the resulting images, lines between 80 and 100 degrees are detected using Hough transform from the edge image, in order to delimit the utility pole, (7):

$$Ho^s(x, y) = \text{Hough}(BW^s(x, y), \mu_{\text{Hough}(s)}). \quad (7)$$

VI. AND Operation

In order to get a reliability edges, a combination of $C(x, y)$ and $Ho(x, y)$ is calculated, (8):

$$E^s(x, y) = C^s(x, y) \text{ AND } \text{Hough}(BW^s(x, y), \mu_{\text{Hough}(s)}). \quad (8)$$

Because it describes each edge in the image, the process of connected components labeling runs using the iterative method, (9):

$$L_i^s(x, y) = \text{labeling}(E^s(x, y)), \quad (9)$$

where $O_i \cap O_j \in \emptyset \quad \forall j \neq i$ for all $i, j = 1, 2, 3 \dots, LN$.

VII. Hu Invariants, Slope and Area Calculus

Each labeled edge is described using the seven Hu invariants, (10):

$$[\phi_{1,i}^s, \phi_{2,i}^s, \phi_{3,i}^s, \phi_{4,i}^s, \phi_{5,i}^s, \phi_{6,i}^s, \phi_{7,i}^s] = \text{Hu}(L_i^s(x, y)). \quad (10)$$

VIII. Classification Stage

In this work, the classification stage is implemented with 10 neurons in hidden layer Neural Network (NN), trained to identify the shape of utility-pole using Hu invariants equation (11). Its architecture consists in fourteen inputs connected to a “tansig” neuron and a “logsig” function at the output:

$$O_i^s = \text{NN}(\phi_{1,i}^s, \phi_{2,i}^s, \phi_{3,i}^s, \phi_{4,i}^s, \phi_{5,i}^s, \phi_{6,i}^s, \phi_{7,i}^s), \quad (11)$$

where O_i^s has binary values, 0 to indicate that is not a utility pole and 1 to express that is a utility pole edge.

X. Stereo Matching and Utility Pole Localization

The centroid of $L_k^s(x, y)$ for $O_i^s = 1$, is calculated in order estimate the location of utility-pole for $s = left$ and $s = right$ using equation (12):

$$(C_x^s, C_y^s) = centroid(L_k^s(x, y)). \quad (12)$$

Finally, the location of the utility-pole is triangulated using the characteristics of the stereo vision system, (13):

$$(P_x, P_y, P_z) = triangulation(C_x^{left}, C_y^{left}, C_x^{right}, C_y^{right}). \quad (13)$$

Once the pole has been detected, next step is the detection of the crosstree and to estimate the position of the screws and nuts. To achieve this firstly the holes in the crosstree are detected, and from these, the 3D position of the nuts is calculated. The steps for this are listed below:

- a. Detection of horizontal lines.
- b. Region of interest (ROI) creation including the cross-arm.
- c. Thresholding of the ROI.
- d. Holes detection by circularity.
- e. Holes matching based on the fundamental matrix.
- f. Triangulation of the position of each hole is based on two images.
- g. Line fitting of the triangulated holes.
- h. Estimation of the nuts position.

To perform the holes matching the following property of the stereo systems is used:

$$p'Fp = 0, \quad (14)$$

where F is the Fundamental Matrix of the stereo system and p is a 3D point in the scene captured by the cameras.

3 Experiments

The experiment was developed in the Tatsuno laboratory under controlled lighting, i.e., lighting with conventional lamps. The cameras used are Sony EVI-D70 with resolution 768 x 494 pixels. The system was tested with Japanese harnesses and utility-pole. After acquiring both stereoscopic pictures of system, we proceed to conversion from RGB to HSV and the result is shown in Fig. 2, to filter by color. In this case, using the ranges defined in Table 1.

Table 1. Color range for the interest objects (Utility-poles).

	Min	Max
Left Image Hue	54	198
Left Image Saturation	7	86
Left Image Value	20	64
Right Image Hue	23	213
Right Image Saturation	9	75
Right Image Value	20	66

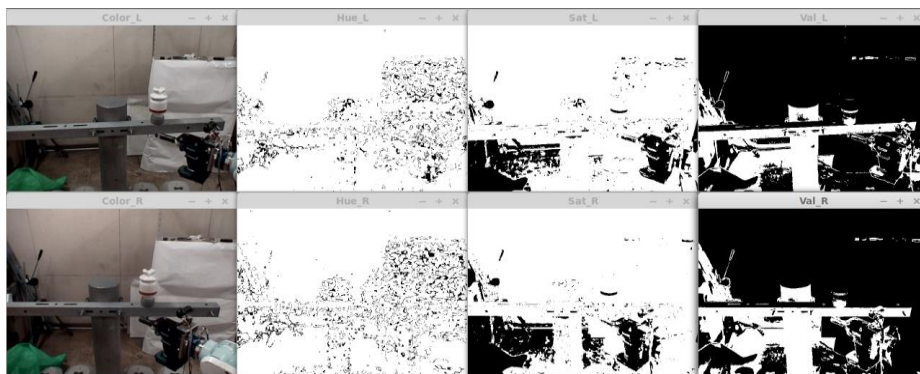


Fig. 2. Stereo image acquisition and RGB to HSV conversion.

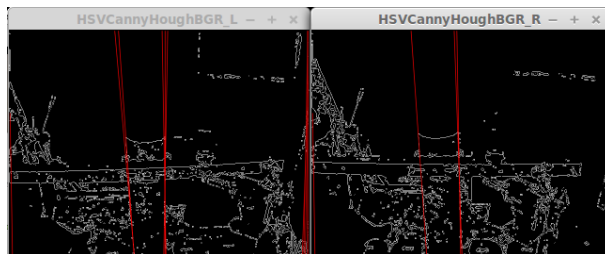


Fig. 3. Lines detected in a range of 80 to 100 degrees.

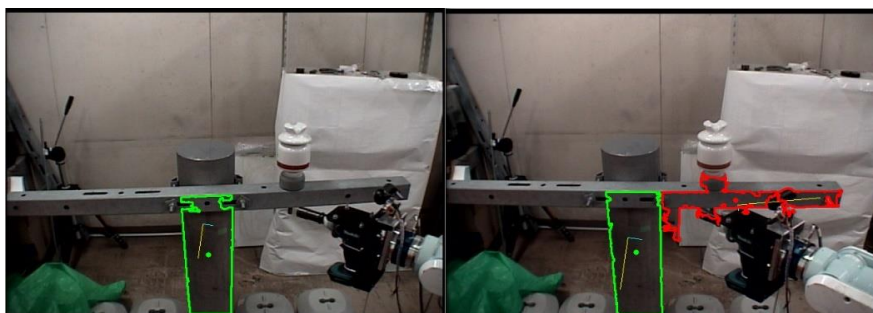


Fig. 4. Utility pole segment detected by using color, shape and dimensions.

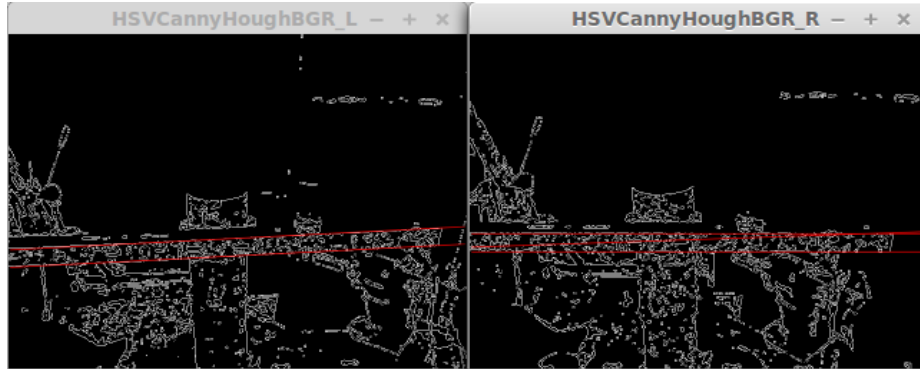


Fig. 5. Lines detected in a range of -10 to 10 degrees.



Fig. 6. Cross-arm detected and segmented.



Fig. 7. Cross-arm ROI, binarized and complemented.

The next stage (IV) is to find contours in the thresholded (BW) images, using the Canny method for it. Then, in the resulting images, approximately vertical lines are detected (V stage), between 80 and 100 degrees. This aims to separate the pole and background objects that have approximately the same color, which would alter the shape of the blob detected, affecting the invariants calculation.

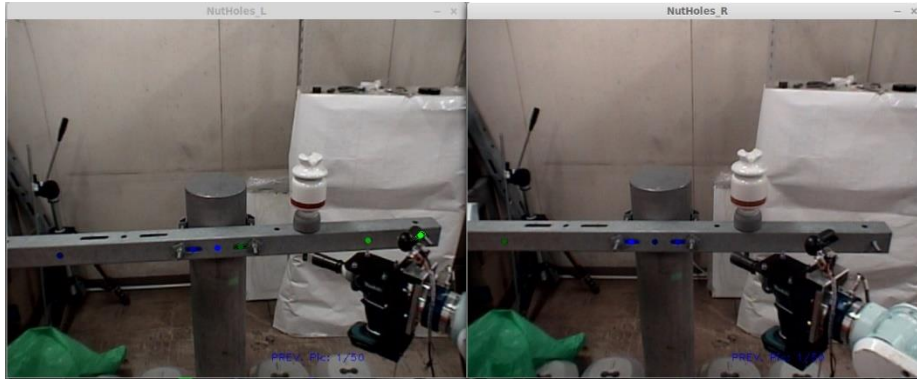


Fig. 8. Detected holes.

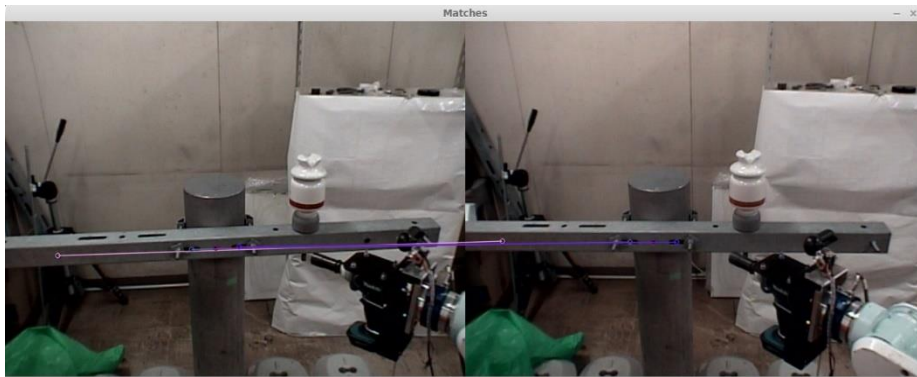


Fig. 9. Matched Holes.

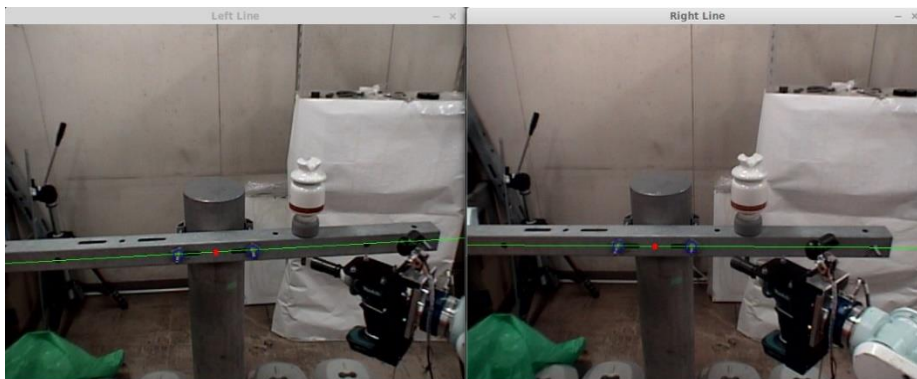


Fig. 10. Located nuts.

The selection of the threshold for the lines detection by Hough method is an important issue because if a very high threshold is selected only one side of the pole could be detected, whereas with a very low threshold, many lines are detected and if

some of these lines cross the pole also affects the detection process. In this case, 72 and 71 values have been selected for the left and right cameras, respectively. In Fig. 3, it can be seen the lines detected with these thresholds.

Once the vertical lines are found, there are negated and AND operated with the color thresholded images in order to separate the pole from the background (VI stage). At this point we have the blobs in the image which correspond to the interest objects; for every blob, Hu invariants, slope and area are calculated in order to classify the blob (VIII stage).

Note: A prior training run is required for the Neural Network (NN) to be able to properly classify. Objects classified as utility-poles are highlighted in green, objects classified as not pole are highlighted in red, as shown in Fig. 4. If the NN is unable to accurately classify the object, it is highlighted in blue.

After this stage the pole has been found in both images (in case it appears at the cameras view) and a triangulation is performed using the pole centroid in both images to spatially locate it, relative to the left camera (stage IX).

Once the pole has been detected, next step is the detection of the crosstree and to estimate the position of the screws and nuts. To achieve this firstly the holes in the crosstree are detected, and from these, the 3D position of the nuts is calculated. The Fig. 5 shows the (a) step.

After the lines are detected a ROI is created with them, as shown in Fig. 6. The Fig. 7 and 8 show the result of the (c) and (d) steps.

To perform the holes matching the stereo epipolar property equation (14) is used. Instead of 0 a sufficient small ϵ value is adopted.

This value change with the stereo calibration. So, values between 0.05 and 0.15 are recommended to start.

After this step the holes have been detected and matched in both images, whereby the position of each of them can be triangulated and proceed to step g. It makes a 3D line fitting with all hole points; finding this line and knowing the physical distance from the center hole of the cross-arm to the nuts, the position of these can be calculated, as shown in Fig. 9.

Finally, the 3D position of the last point is found, in this case, the upper right corner of the front face of the crossarm, and thus the plane thereof is calculated, with this, the normal vector to that plane is obtained, i.e., the direction of the screws.

Therefore, although the system supports changes in perspective, must be ensured that both cameras can watch the right corner of the front face of the cross-arm to calculate the orientation of the screws as illustrates Fig. 10.

4 Discussion

This work is an extension of that developed in reference (Barranco-Gutierrez, Martínez Díaz, & Gomes Torres, An Approach for Utility Pole Recognition in Real Conditions, 2014). Table 2 shows the main differences, which include crosstree and nuts detection and localization. The last stage has not been tested outdoor yet; however, pole detection was successfully tested before, in extreme illumination conditions.

The neural network was trained with 50 images from each camera. In each training image one segmented object from the true class (UP) and one segmented object from

Table 2. Methods comparison.

Feature	Barranco et al. 2014	Actual proposal
Utility-pole detection	Yes	Yes
Under Solar illumination	Yes	Not
Cross-arm detection and location	Not	Yes
Nuts detection and location	Not	Yes
Japanese UP detection and location	Not	Yes
Mexican UP detection and location	Yes	Yes

Table 3. Confusion matrix.

Output/target	UP	Non-UP
UP	40%	0%
Non-UP	10%	50%

Table 4. Proposals comparison.

	(Zhang, et al., 2018)	(Watanabe, 2018)	Actual proposal
Average precision	79%	72%	90%

the false class (non-UP) were selected. The training error reached a value below 1%. The whole system was tested in real time with 25 non-training images from each camera. The system tried to select the UPs among all the segmented objects into each processed image.

Table 3 shows the confusion matrix resulting from the test where the percentage of true positives is 40%, true negatives 0%, false positives 10% and false negatives 50%. Note that, in some cases, the system was unable to detect the UP due to a bad segmentation of the image.

However, once detected the UP, the system detected the cross arm and the nuts 100% of the time. In Table 4 is shown the comparison between some actual proposals on this topic.

5 Conclusions

The proposed computer vision system designed to detect Japanese utility-poles and its nuts and crosstree, identifies correctly the aims in ninety percent. Which means it has a good performance under the conditions that have it now. Although neural networks have shown great ability to classify nonlinear problems, the HSV color model

description and the shape characterization of Hu invariants, greatly helped to obtain the results shown in the work. Regarding the measurement of the dimensions of objects in the 3D environment, stereoscopic configuration helps to facilitate the calculations of depth and longitudinal dimensionality of objects. So, it has a classification by color, shape and dimensions of interest objects. Cables and ceramic insulator type bell detection is the next stage to accomplish for the robotic maintenance task. To work again under solar lighting conditions and test the entire system in real conditions.

Acknowledgments. The authors greatly appreciate the support of PROMEP and CONACyT with the project 215435. Also, we want to thanks to Cesar Cerezo by his valuable support to construct this work.

References

1. Barranco-Gutierrez, A.I., Martínez-Díaz, S., Gomes-Torres, J.L.: An approach for utility pole recognition in real conditions. *Lecture Notes in Computer Science*, pp. 113–121, Springer (2014)
2. Barranco-Gutierrez, A.I., Martínez-Díaz, S., Gomez-Torres, J.L.: *Visión estereoscópica con Matlab y OpenCV*. Pearson (2018)
3. Berwick, D., Lee, S.: A chromaticity space for specularly, illumination color and illumination pose-invariant 3-D object recognition. *International Conference on Computer Vision*, pp. 165–170 (1998)
4. Cetin, B.: Automated electric utility pole detection from aerial images. *SOUTHEASTCON'09*, pp. 44–49 (2009)
5. Cianchetti, C.L.: Soft robotics: new perspectives for robot bodyware and control. *Frontiers in Bioengineering and Biotechnology*, 1 (2014)
6. Halawani, S.M., Sunar, M.S.: Interaction between sunlight and the sky colour with 3D objects in the outdoor virtual environment. *Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, pp. 470–475, IEEE Computer society (2010)
7. Igor, K., Amit, A., Ehud, R.: Color invariants for person reidentification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1622–1634 (2013)
8. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer vision for autonomous vehicles: problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, pp. 1–308 (2020)
9. Nakajima, C.: Automatic recognition of facility drawings and street maps utilizing the facility management database. *Third International Conference on Document Analysis and Recognition*, pp. 516–519 (1995)
10. Watanabe, J.I.: Electric pole detection using deep network based object detector. *SPIE Remote Sensing*, SPIE (2018)
11. He, K.T.: An example of open robot controller architecture for power distribution line maintenance robot system. *World Academy of Science, Engineering and Technology*, pp. 266–271 (2008)
12. Zhang, W., Witharana, C., Li, W., Zhang, C., Li, X., Parent, J.: Using deep learning to identify utility poles with crossarms and estimate their locations from google street view images. *Sensors*, pp. 1–21 (2018)

José Luis Gomez Torres, Saúl Martínez Díaz, Iris Iddaly Méndez Gurrola, et al.

13. Zhang, Z.: Flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1330–1334 (2000)
14. Zisserman, A., Hartley, R.: *Multiple view geometry in computer vision*. Cambridge University Press (2003)

Propuesta de un sistema de control, monitoreo y asistencia para optimización de recursos energéticos en el hogar

Leandro Espino Espino¹, Yotziri Paloma Pérez Ríos¹, Edgar Gonzalo Cossio Franco²

¹Espinoleandroo,
México

²Universidad del Valle de Atemajac,
México

{espinoleandroo, ypelekai}@gmail.com,
edgar.cossio@univa.mx

Resumen. Este artículo presenta el proceso llevado a cabo para generar la propuesta de un sistema el cual contará con una aplicación móvil multiplataforma para el control y monitoreo de recursos del hogar, como lo son el agua potable, energía eléctrica, gas LP y la temperatura con humedad del aire; en conjunto con una red neuronal para la generación de tips para la optimización del consumo de estos recursos energéticos del hogar; dicho algoritmo se encontrará en una plataforma online acompañado de la base de datos, la cual contendrá todos los registros de los sensores del hogar, y de una API REST mediante la cual se podrá tener acceso al sistema desde cualquier lugar, siempre y cuando se cuente con una conexión a internet.

Palabras clave: IoT, Esp 32, Redes Neuronales, API REST, Aplicación Móvil.

Proposal for a Control, Monitoring and Assistance System for Optimization of Energy Resources at Home

Abstract. This article presents the process carried out to generate the proposal of a system which will have a multiplatform mobile application for the control and monitoring of household resources, such as drinking water, electricity, LP gas and temperature with humidity from air in conjunction with a neural network for the generation of tips for optimizing the consumption of these energy resources in the home. This algorithm is present on an online platform accompanied by the database, which contains all the records of the home sensors, and a REST API

through which the system can be accessed from anywhere, as long as it is counted with an internet connection.

Keywords: IoT, Esp 32, Neural Networks, REST API, Mobile Application.

1. Introducción

Los sistemas de domótica con inteligencia artificial aplicada es una tecnología de tendencias actual y futura, en el mercado que hace que la vida sea más simple y fácil de controlar. La automatización del hogar se puede definir como la eliminación de la interacción humana [2].

El Internet es una de las creaciones más importantes de la humanidad para la comunicación y ahora se debe tomar en cuenta el crecimiento que ha tenido el Internet de las cosas (IoT) con una gran cantidad de proyectos con un sin fin de funcionalidad, aunado a este crecimiento también se tiene el exponencial aumento del uso de smartphones y tablets el cual se elevó a 12.5 mil millones en 2010 el número de dispositivos conectados a internet, esto significa que desde el 2010 existen más dispositivos conectados a internet que personas, ya que en ese año la población mundial era de 6.8 mil millones [11]. Y a medida que Internet de las cosas crezca en los próximos años, más dispositivos se unirán a la lista de dispositivos conectados [2].

Con base en ello, los teléfonos móviles son la tecnología más popular utilizada principalmente hoy en día como herramienta de comunicación; puede desempeñar un papel importante en las casas inteligentes [3, 6]. Actualmente se presta bastante atención al teléfono móvil invirtiendo una gran cantidad de tiempo en el uso de aplicaciones. Las personas en promedio invierten 3 horas con 40 minutos diarios por usuario [15], en resumen, al final del día las Personas y Objetos interactúan entre sí mediante Procesos, generando Datos, cubriendo los pilares de IoT.

En la actualidad, cada organización está dedicando su I+D (Unidad de Investigación y Desarrollo) hacia IoT [8] y al empleo del cómputo en la nube, lo que se define como un modelo que permite, el acceso bajo demanda a redes ubicuas para compartir un conjunto configurable de recursos de computación, dichos servicios virtuales son actualmente escalados masivamente y manejados como recursos agrupados que son configurados continuamente. [12]

Considerando además que la seguridad y la comodidad son factores que las personas buscan en un sistema de IoT [17], es necesario hacer necesario la implementación de al menos una computadora para realizar un control y monitoreo de eventos de manera inteligente [4]. Una alternativa bastante efectiva es el empleo de IA (Inteligencia Artificial) debido a que una colección de programación potente y rigurosa [1], ideal en la toma de decisiones de manera autónoma en el sistema de IoT [9, 10, 13]. Específicamente la utilización de redes neuronales coadyuva al desempeño del sistema debido a que son modelos computacionales, paralelos, compuestos de unidades procesadoras y adaptativas con una alta interconexión entre ellas si se emplea más de una [19].

1.1. Antecedentes

En el proyecto HAUAR: Home Automation Using Action Recognition [7] los autores desarrollan sistemas de automatización y monitoreo del hogar en el día a día.

En el proyecto se logra la automatización implementando módulos de sensores y cámaras de video; integrándolos a una red local con una raspberry pi donde el algoritmo HAUAR con gran eficiencia opera electrodomésticos sin intervención de humanos, solamente con el puro reconocimiento de imagen para conocer la acción que realiza la persona.

En otra perspectiva de la automatización del hogar en el trabajo de investigación Development of IoT Monitoring Device and Prediction of Daily Life Behavior [13] los autores proponen un sistema de monitoreo de una persona dentro de una habitación, mediante sensores generan datos para posteriormente analizarlos y predecir el comportamiento de la persona. Utilizando dos algoritmos de aprendizaje automático SVM (Vector de soporte Machine) y LSTM (memoria a largo plazo) para predecir el comportamiento de la persona. En suma, a su trabajo realizado, los autores resaltan lo costoso que es implementar un hogar inteligente, por ello su sistema es económico.

2. Problemática

Con base en el estudio realizado en el presente trabajo, mediante el análisis de resultados del instrumento mostrado en la sección 5.1 se obtiene que la mayoría de las personas han escuchado en algún momento el término de automatización del hogar, y tal vez sin tener un amplio conocimiento en lo que con lleva la automatización, les interesa implementarla, con un sin fin de funcionalidades, puesto que en cada hogar existen distintas necesidades por cubrir. Un sector de la población está interesado en tener un control de seguridad de su casa, otro sector del público se preocupa más por el ahorro de recursos energéticos y un tercer sector lo implementaría simplemente por comodidad.

Por tanto, se deduce que una tercera parte de las personas están interesadas en la automatización de sus hogares con la finalidad de reducir el consumo de recursos energéticos según el estudio realizado [17]. En suma, la problemática radica, en que hoy en día una tercera parte de la contaminación mundial proviene de procesos para la generación de energía eléctrica [18] lo cual ha generado la búsqueda de nuevas fuentes de energías renovables, así mismo la generación de conciencia en las personas, con respecto a la contaminación, orienta a la población a buscar la manera de reducir el consumo de energía.

Agregado a ello, el hecho de que en los últimos años se han desarrollado una enorme cantidad de proyectos en el área de IoT, sin embargo, partiendo de esos proyectos, los que están más centrados en la automatización, muchos de ellos no cubren un factor importante, tener acceso a los datos desde cualquier lugar, en su mayoría requieren que la persona se encuentre en mismo sitio de la automatización para contar con acceso a la red en la cual se ejecuta las acciones. Esto último, repercute en el tercer sector del público en el estudio realizado, público el cual busca la automatización de su hogar por comodidad.

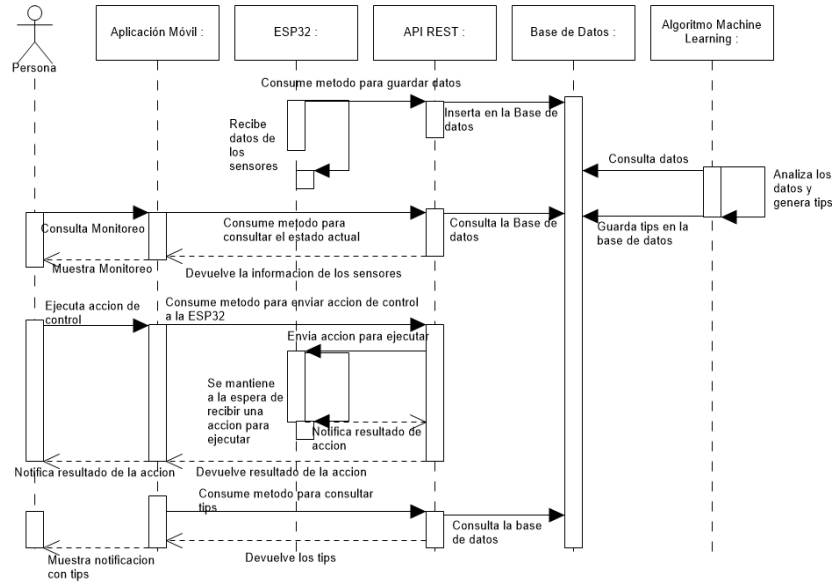


Fig. 1. Diagrama de secuencia del proceso de propuesta o del sistema.

3. Propuesta

La presente investigación propone una plataforma mediante la cual ciertos controles de la casa se comunican. El usuario asiste a todas las funciones de IoT mediante el control de una aplicación móvil multiplataforma, es decir, la aplicación se podrá ejecutar en los sistemas operativos Android y IOS a partir de sus versiones 4.1 y 8 respectivamente [16] y toda la información relacionada se envía a la nube.

Es de vital importancia mencionar que el diseño del sistema no solo utiliza tecnología IoT sino también la función de inteligencia artificial para predicciones implementando la técnica de aprendizaje profundo gated recurrent unit (GRU) y la nube.

Las predicciones se pueden hacer a través de un motor predictivo que a su vez se puede utilizar en un futuro próximo. Para su realización se propone emplear GRU que es un algoritmo de IA supervisada. Este proyecto se puede utilizar como un sistema de recomendación para las personas mayores y las personas con discapacidad física que no pudieran hacer su trabajo de manera independiente y fácil.

La intención es que mediante la recolección de datos recabados a través de la interacción con la aplicación se realicen un módulo de tips o recomendaciones en el móvil que ofrezca información a futuro para la mejora en la toma de decisiones del usuario sobre el consumo de recursos que tenga agregados en el sistema central.

Con esta ventaja, las personas obtienen un asistente para administrar el hogar y sus necesidades. Además de que emplea tecnología de comunicación inalámbrica que proporciona acceso remoto desde tableta o teléfono inteligente.

Requerimientos	Análisis del Sistema	Aprendizaje y Capacitación	Desarrollo del Diseño
<p>Educción de requerimientos</p> <p>Especificación de requerimientos</p>	<p>Desarrollo de alternativas</p> <p>Selección de componentes del sistema</p>	<p>Indagación de tecnologías existentes</p> <p>Conocimiento de la tecnología seleccionada</p>	<p>Creación del paquete de datos técnicos</p> <p>Diseño y modelado de base de datos</p> <p>Diseño de interfaz del sistema</p> <p>Diseño del plan de pruebas</p>
Fase 1	Fase 2	Fase 3	Fase 4

Fig. 2. Fases del proceso.

4. Metodología

“Una actividad busca lograr un objetivo amplio y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software.” [5]. Para este proceso se realizarán distintas actividades por fases con la finalidad de tener un control de proceso, a continuación, se presenta una descripción general de dichas actividades en la Fig. 2.

2.1. Requerimientos

Educción de requerimientos: El verbo educir se define como sacar una cosa de otra y se ha adoptado por la dificultad que supone identificar los requisitos de un sistema de información. En este proyecto la educación de requisitos se realizó mediante una técnica simple, encuestas al público en general.

La encuesta se realizó mediante la plataforma de Google Forms y se compartió a los conocidos de los autores, la población total a la que se aplicó la encuesta fue de 120 personas; las preguntas realizadas fueron 10 preguntas de opción múltiple y una pregunta abierta, las cuales se enlistan en la sección de resultados con sus respectivos resultados.

Especificación de requerimientos: Esta etapa es en la cual se aterriza toda la información obtenida de las encuestas en requerimientos técnicos, para realizar la especificación se implementará un Listado de Requerimientos el cual se presenta en la sesión de resultados utilizando los Tags: Sistema, Función, Definición, Interfaz, Software, Hardware y Comunicación.

Al realizar un análisis de las respuestas de la encuesta, se deduce que las personas buscan el control de la automatización de su hogar en su Smartphone, tal como lo indica la gráfica de la pregunta #4, es por ello que se diseñará el sistema enfocado a una aplicación móvil para el uso de las personas. Por su parte el resultado de la pregunta #7 indica que las personas buscan tener el control y monitoreo de su hogar desde cualquier sitio, sin encontrarse dentro de la casa, por consiguiente, se plantea que la base de datos se encuentre en un servidor en internet para acceder a los datos a través de un API REST desde cualquier lugar donde se cuente con conexión a internet.

Debido a la alta aceptación de la inteligencia artificial por parte de las personas encuestadas, se propone el uso de un sistema de redes neuronales para la generación de tips de ahorro de recursos ya que los encuestados muestran interés por la obtención de tips personalizados acorde al consumo de recursos en el hogar.

3.1. Análisis de sistema

Desarrollo de alternativas: En esta etapa, se analizará el Listado de Requerimientos, y visualizando cada requerimiento como una meta a cumplir se generarán propuestas de solución para cada uno.

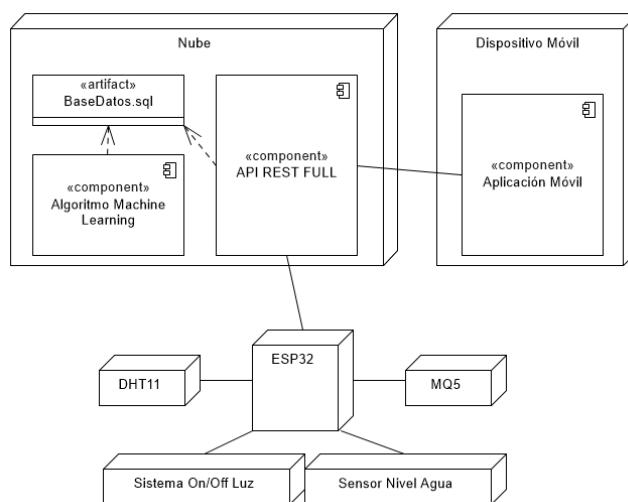


Fig. 3. Arquitectura del sistema.

Tecnología o componente	Descripción
ESP32	El dispositivo cuenta con conectividad de wifi y bluetooth, con un micro-procesador integrado de 32 bits.
DHT11	El sensor permite medir la temperatura y humedad en el aire.
MQ5	Este sensor tiene la finalidad de detectar la presencia de gas LP.
Sensor nivel de agua	Este sensor proporciona el nivel de agua en el contenedor principal de la casa, con este dato se tendrá un historial del nivel de agua.
Sistema ON/OFF de luz	Este sistema nos permitirá poder apagar y encender los focos de la casa desde la aplicación móvil en conjunto con los apagadores convencionales.
GRU	Son un tipo de red neuronal simple y con un gran desempeño trabajando con datos pequeños.
API REST	El API REST se estructura mediante la arquitectura modelo vista controlador introduciendo una capa de servicios. Todo esto con el framework Codeigniter en su versión 3.

Selección de componentes del sistema: El abanico de propuestas generadas será evaluado con los criterios de selección para obtener los componentes que integrarán al sistema.

4.1. Aprendizaje y Capacitación

Indagación de las tecnologías existentes: Se indagará sobre las tendencias tecnológicas que sean de utilidad para el desarrollo del sistema. En esta etapa se consultaron distintas fuentes.

Conocimiento de tecnología seleccionada: Se estudian a fondo las tecnologías seleccionadas en la etapa de Selección de componentes del sistema.

4.2. Desarrollo de diseño

Creación del paquete de datos técnicos: El paquete de datos técnicos contiene, diagramas UML, diccionarios de datos, estándares de codificación y mapas mentales necesarios para la guiar el proceso de desarrollo del sistema.

Diseño y Modelado de la Base de Datos: Mediante técnicas de normalización diseñar la base de datos que satisfaga las necesidades del sistema.

Diseño de interfaces del sistema: Mediante Mockups se diseñarán las interfaces gráficas del sistema en el apartado 5.3 se muestran el prototipo de diseño en la Fig. 4.

Diseño del plan de pruebas: Se diseñó un plan de pruebas con dos objetivos, orientar las pruebas realizadas durante el desarrollo y depuración del sistema, pero principalmente para ser ejecutado en la Fase de Verificación y Validación.

5. Resultados

En la sección de resultados se presentan tablas, gráficas, ilustraciones y diagramas con los distintos productos generados en cada una de las fases de la investigación.

5.1. Gráficas de las respuestas de la encuesta

Pregunta	Resultados
1 ¿Has oído hablar sobre la automatización de hogares (domótica)?	○ Si: 80% ○ No: 20%
2 ¿Te gustaría implementar un sistema de automatización en tu hogar?	○ Si: 94.2% ○ No: 5.8%
3 ¿Con qué finalidad implementarías un sistema de automatización en tu casa?	○ Ahorro energético: 31.1% ○ Comodidad: 31.9% ○ Seguridad: 32%
4 ¿Desde dónde te gustaría tener el control de tu hogar?	○ Celular: 31.7% ○ Computadora: 0% ○ Ambos: 68.3%
5 ¿Qué importancia tiene para ti, monitorear tu hogar cuando sales de ahí?	○ De vital importancia: 40% ○ Es importante para mí: 59.2% ○ No me importa: 0.8%

6 ¿En qué situación implementarías un control y monitoreo de hogar?	<ul style="list-style-type: none"> ○ Para estar al tanto del hogar de personas adultas que viven solas: 15% ○ Para estar al tanto de mis hijos cuando salgo de casa: 13.3% ○ Para estar al tanto de mi hogar cuando no estoy: 71.2%
7 ¿Cuál es tu opinión respecto a recibir tips personalizados para ti sobre el ahorro de energía en tu hogar?	<ul style="list-style-type: none"> ○ Me serían muy útiles y los implementaría 61.7% ○ Los considera en cuenta 37.5% ○ Me resultaría indiferente esta información: 0.8%
8 ¿Cuál de los siguientes rubros genera mayor gasto económico en tu hogar?	<ul style="list-style-type: none"> ○ Energía Eléctrica: 61.7% ○ Gas LP: 32.5% ○ Agua potable: 5.8%
9 ¿Consideras útil el monitoreo de consumo de los recursos tales como energía eléctrica, gas LP y agua potable?	<ul style="list-style-type: none"> ○ Si: 94.2% ○ No: 5.8%
10 ¿Cuál es tu opinión respecto a los temas de Inteligencia Artificial y aprendizaje de máquinas?	<ul style="list-style-type: none"> ○ Considero que mejoran nuestras vidas y salvaran miles de ellas: 97.5% ○ Considero que es una amenaza para la humanidad 2.5%

5.2. Listado de requerimientos

Descripción	TAG
El sistema contará con una aplicación móvil, multiplataforma, la cual permitirá al usuario el control y monitoreo de recursos energéticos del hogar.	Sistema
El usuario tendrá acceso al control y monitoreo de su hogar desde cualquier lugar, siempre y cuando cuente con acceso a internet.	Definición
El monitoreo del hogar consistirá en: <ul style="list-style-type: none"> – Temperatura y humedad en el aire. – Presencia de gas LP – Nivel de agua en el depósito principal – Estado (encendido/apagado) de los focos 	Definición
El control del hogar consistirá en el encendido y/o apagado de los focos del hogar.	Definición
La temperatura y humedad en el aire se medirán usando el sensor DHT11.	Hardware
La presencia de gas LP se medirá mediante el sensor MQ5.	Hardware

El nivel de agua dentro del contenedor principal de la casa se medirá usando un sensor de nivel de líquidos.	Hardware
Los sensores se comunicarán a un dispositivo ESP32 para la transmisión de datos.	Hardware
El dispositivo ESP32 deberá transmitir los datos recolectados por los sensores a la base de datos a través del API REST.	Comunicación
La aplicación móvil consultará a la base de datos los datos para el monitoreo a través del API REST	Comunicación
La aplicación móvil publicará en la base de datos las acciones ejecutadas por el control a través del API REST	Comunicación
En el mismo servidor donde se ubique la base de datos, se ubicará el API REST para el acceso a ella, y el sistema de red neuronal	Definición
El sistema de red neuronal se ejecutará como una tarea programada del servidor para realizar un análisis de los datos y generar tips de optimización.	Sistema
Los tips que se generarán serán sobre: <ul style="list-style-type: none"> - consumo de agua potable - consumo de energía eléctrica - prevención de fugas de gas LP 	Definición / Función
El dispositivo ESP32 consultará constantemente la publicación de acciones en la base de datos a través del API REST	Función
El API REST contará con autenticación para las peticiones http	Función
El usuario recibirá una respuesta sobre la ejecución de su acción, para ser notificado si la acción se ejecutó correctamente o no.	Función
El usuario recibirá los tips mediante notificaciones.	Función
La aplicación móvil consultará cada determinado lapso de tiempo la existencia de nuevos tips en la base de datos a través del API REST.	Función
La información mostrada en la sección de monitoreo se actualizará en iteraciones dentro del rango de 20 segundos a 30 segundos.	Función
El dispositivo ESP32 publicará en la base de datos los valores de los sensores en iteraciones dentro del rango de 5 segundos a 10 segundos	Función

5.3. Mockups



Fig. 4. Diseño de la aplicación móvil.

5.4. Plan de pruebas

CP#	Datos de entrada o Acción	Resultado Esperado
CP1	El usuario accede a la sección de control en la aplicación móvil y ejecuta la acción de apagar el foco seleccionado.	Recibe una notificación informando que el foco ha sido apagado. De lo contrario se mostrará una notificación informando que no fue posible apagar el foco.

CP2	El usuario ingresa a la sección de monitoreo en la aplicación móvil.	Visualiza gráficamente en la pantalla: <ul style="list-style-type: none">● Temperatura y humedad en el aire.● Presencia de gas lp● Nivel de agua en el depósito principal● Estado (encendido/apagado) de los focos
CP3	El usuario accede a la sección de control en la aplicación móvil y ejecuta la acción de encender el foco seleccionado.	Recibe una notificación informando que el foco ha sido encendido. De lo contrario se mostrará una notificación informando que no fue posible encender el foco.
CP4	El usuario ingresa a la sección de monitoreo, y manualmente enciende un foco que en el monitoreo se muestra apagado	En un lapso no mayor a 35 segundos el cambio se deberá ver reflejado en la sección de monitoreo de la aplicación móvil

6. Conclusiones

Se considera que la implementación de sistemas complejos que se involucren en la Industria 4.0 son un factor muy importante para impulsar el crecimiento del ecosistema de tecnología en México.

Con base en los resultados obtenidos en la aplicación de la encuesta, se concluye que la sociedad se encuentra abierta totalmente a emplear tecnologías como IA. Por lo tanto, es de vital importancia continuar con el desarrollo del producto con la intención de introducirlo al mercado.

7. Trabajo futuro

A futuro se pretende realizar el prototipo del sistema físico aunado al desarrollo de la aplicación móvil multiplataforma que funcionará como el modelo de recomendaciones proporcionadas por el modelo de predicciones para consulta del usuario.

Además, se contempla la opción de realizar un panel interactivo que podrá ser visualizado en un touchscreen, el cual se encontrará en comunicación con el prototipo del sistema central de monitoreo para mostrar los datos recolectados por los sensores instalados en el hogar. En la Fig. 5 se muestra el proceso a seguir en las siguientes fases del proyecto para darle continuidad a la investigación y de esta manera implementar el diseño de la propuesta.

Implementación del Diseño	Verificación y Validación	Despliegue
Codificación	Ejecución del plan de pruebas	Implementación del sistema
Integración con la base de datos	Análisis de resultados de pruebas	Retroalimentación
Documentación	Seguimiento a errores	Soporte y Mantenimiento
Fase 4	Fase 5	Fase 6

Fig. 5. Fases a concluir en el trabajo futuro.

Referencias

1. Speel, P.H., Schreiber, A.Th., van Joolingen, W., van Heijst, G., Beijer, G.J.: Conceptual modelling for knowledge-based systems. Encyclopedia of Computer Science and Technology, Marce Dekker Inc., New York (2001)
2. Shopan-Dey, A.R., Sandip, D.: Home automation using IOT. IRJET, 2(3), pp. 1965–1970 (2016)
3. Wong, M.C.: A phone-based remote controller for home and office automation. IEEE Transactions on Consumer Electronics, 40(1), pp. 28–34 (1994)
4. Nieto, J.P.: Detección y diagnóstico de fallas en sistemas eléctricos de potencia combinando una red neuronal autoasociativa y una red neuronal probabilística. Research in Computing Science, 148, pp. 99–112 (2019)
5. Pressman, R.S.: Ingeniería del Software, Un enfoque práctico. McGraw-Hill (2010)
6. Sikandar, M., Khiyal, H., Khan, A.: SMS based wireless home appliance control system (HACS) for automating appliances and security. Issues in Informing Science and Information Technology, 6 (2009)
7. Shashank Kotyan, Nishant Kumar, Pankaj Kumar Sahu, Venkanna Udutalapally: HAUAR: Home Automation Using Action Recognition (2019)
8. Bilgeri, D., Fleisch, E., Wortmann, F.: How to IoT affects multibusiness industrial Companies: IoT organizational archetypes. International Conference on Information Systems (ICIS) (2018)
9. O'Leary, D.E.: Artificial intelligence and big data. IEEE Intell. Syst., 28, pp. 96–99 (2013)
10. Ponsizewska-Maranda, A., Kaczmarek, D.: Selected methods of artificial intelligence for internet of things conception. In: Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1343–1348 (2015)
11. Evans, D.: Internet de las cosas como la próxima evolución de Internet lo cambia todo. Cisco Internet Business Solutions Group (IBSG) (2011)

12. Avila-Mejía, O.: Computación en la nube. Depto. de Ingeniería Eléctrica, UAM-I. (2011)
13. Wilks, Y.: An artificial intelligence approach to machine translation. Stanford University, Department of Computer Science (1972)
14. Maharjan, R., Shiraishi, K., Yamamoto, T., Yamamoto, Y., Ohshima, H.: Development of IoT monitoring device and prediction of daily life behavior. In: Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, pp. 584–588. (2019)
15. Grance, M.T.: The NIST definition of cloud computing. National Institute of Standards and Technology, NIST Special Publication, pp. 800–145 (2011)
16. Yi Min Shum: Visible body: Yi Min Shum Xie, situación global mobile 2020 (2020).
17. Leandro et al.: Estudio de opinión sobre uso de IoT en hogares (2020)
18. Pinzón, S.E., Hernández, J.A., Arroyo-Figueroa, G.: Aplicación de modelos auto regresivos para la predicción de generación de energía eléctrica a partir de datos eólicos. *Research in Computing Science*, 139, pp. 47–57 (2017)
19. Serrano, A.J., Soria, E., Martin, J.D.: Redes neuronales artificiales (2019)

Modelo inteligente para determinar el esfuerzo de desarrollo de software mediante machine learning

Edgar Gonzalo Cossio Franco¹, Mariela Chávez Marcial²,
Emma Yesenia Rivera Ruiz², Yotziri Paloma Pérez Rios²,
Marco Julio Franco Mora²

¹ Instituto de Información Estadística y Geográfica de Jalisco,
México

² Tecnológico Nacional de México / ITS de Ciudad Hidalgo,
México

edgar.cossio@iieg.gob.mx, mchavez@cdhidalgo.tecnm.mx,
{yess.rivera.1311, ypelekai, darkmarksdoe}@gmail.com

Resumen. La estimación de costos, esfuerzo, personal y tiempo siempre ha sido un tema fundamental para cualquier desarrollo de software, de cualquier tamaño, dado a que desarrollar un producto consta de múltiples variables y tomar la decisión correcta a la hora de realizar y cotizar un proyecto conlleva dinero, personal y tiempo, para esto existen múltiples herramientas. COCOMO junto a Puntos de Función son las herramientas que contemplan variables importantes respecto a un desarrollo y que se utilizarían para la fase de entrenamiento. Este trabajo abarca la creación de una herramienta que mediante la implementación de Machine Learning determina el esfuerzo de desarrollo.

Palabras clave: COCOMO, Machine Learning, Predicción, Puntos de función.

Intelligent Model for Determining Software Development Effort using Machine Learning

Abstract. Estimating costs, effort, personnel and time has always been a fundamental issue for any software development, of any size, since developing a product consists of multiple variables and making the right decision when making and pricing a project It takes money, staff and time, for this there are multiple tools. COCOMO together with Function Points are the tools that contemplate important variables regarding a development and that would be used for the training phase. This work covers the creation of a tool that through the implementation of Machine Learning determines the development effort.

Keywords: COCOMO, Machine Learning, Prediction, Functions Points.

1. Introducción

Entregas a tiempo, control de presupuesto y alta calidad de los productos son objetivos críticos para la administración de proyectos de software. El costo, la calidad y la entrega del software se ven afectados por la precisión de la estimación del esfuerzo del software [13]. Las prácticas de ingeniería de software tienen características específicas que diferencian este campo de la ingeniería tradicional.

En particular, varios factores afectan la estimación del esfuerzo de software en organizaciones y proyectos, incluidos procesos de software inconsistentes y definiciones de medición en proyectos, diversidad sustancial entre proyectos, y diferencias extremas en los tamaños de los productos.

En consecuencia, estas situaciones crean desafíos en la práctica de la estimación del esfuerzo del software, lo que dificulta el grado de rendimiento en la precisión de la estimación. Muchos estudios se han centrado en el desarrollo de modelos y técnicas de estimación de costos de software [9].

Estos incluyen modelos algorítmicos, como COCOMO [3], SLIM [8], SEER-SEM [4] y técnicas de machine learning [27]. Estos modelos y las técnicas se han introducido y utilizado en la industria del software.

En todos los modelos, las entradas y las relaciones son específicas del dominio y dependen totalmente de la opinión de los expertos. Por esta razón, estos modelos tienden a funcionar mal o incluso fallan cuando se intenta cambiar los límites de su aplicación. En tal escenario, existe la necesidad de una técnica que pueda sustituir el juicio experto. Debido a que la mayoría de los estudios en los que diferentes profesionales de software estiman la misma tarea de desarrollo de software informan una gran variación de las estimaciones de esfuerzo. [19].

Entre otras técnicas, Machine Learning, representa una interesante perspectiva que, mediante, la recopilación de datos, la adquisición de conocimientos, la clasificación, el reconocimiento de patrones conforma un proceso que permite predecir escenarios.

Cuando se estima el tamaño del software, se pueden obtener varias métricas, como el esfuerzo, el tiempo y el costo necesario para desarrollar un proyecto, costos de mantenimiento, métricas de calidad y productividad del equipo [5]. La pregunta es qué tan bien funcionan los métodos propuestos por estos modelos en lo que respecta al error de medición de los datasets y las estadísticas comparativas de rendimiento con otras técnicas.

El objetivo general de la presente propuesta es desarrollar un modelo inteligente para determinar el esfuerzo de software basado en un algoritmo supervisado por machine learning haciendo uso de un dataset público. La propuesta es utilizar COCOMO I junto a Puntos de Función para la fase de entrenamiento. Luego, se determina la efectividad de la técnica de aprendizaje automático para el campo de estimación del esfuerzo y se extrae la conclusión posterior al análisis.

Con este trabajo de investigación, se intenta no solo desarrollar una mejor oportunidad para la estimación de costos, sino que también se pretende aplicar machine learning a la ingeniería de software. Dicha fusión de áreas de conocimiento entre Inteligencia Artificial (IA) e Ingeniería de Software (IS) se conoce como SBSE (ingeniería de software basada en búsqueda).

1.1. Estimación

Ha habido una serie de esfuerzos entre desarrolladores de software, ingenieros e investigadores para definir modelos de estimación del esfuerzo del proyecto, destacándose los modelos de análisis de puntos de función y COCOMO [6, 30]. Estos modelos calculan la distancia entre el proyecto de software que se estima y el histórico de proyectos de software (entradas como el tamaño, el número de funciones de atributos y otros generadores de costos), recuperando los más similares para definir una estimación de esfuerzo [31].

La Tabla 1 muestra los trabajos publicados de Análisis de puntos de función de Keremer [32], Chandrasekaran and Kumar [33] y Arnuphaptrairong [34]. Los trabajos de COCOMO de MayaZaki and Mori [35], Keremer [32], Chandrasekaran and Kumar [33]. En los trabajos se incluye la media del error relativo utilizada para evaluar la precisión de los modelos de estimación de costos [30], observándose que solo el estudio de Chandrasekaran and Kumar [33] presenta un porcentaje bajo de error en la estimación.

Tabla 1. Estudios de los modelos de análisis de puntos de función y COCOMO [30].

No.	Modelo de Estimación	Autor	Año de publicación	Media del error relativo (%)
1	Análisis de puntos de función	Kemer	1987	102.74
2	Análisis de puntos de función	Chandrasekaran and Kumar	2012	13.8
3	Análisis de puntos de función	Arnuphaptrairong	2013	82
4	COCOMO	MayaZaki and Mori	1985	165.6
5	COCOMO	Kemerer	1987	583.82
6	COCOMO	Chandrasekaran and Kumar	2012	8.4

Múltiples técnicas y paradigmas están disponibles en la industria y el machine learning es uno de los enfoques crecientes en esta área [15], la introducción de algoritmos de aprendizaje automático combinados con el modelo COCOMO han mejorado el rendimiento de la predicción de los modelos en comparación con enfoques estadísticos, ya que permite el ajuste de los parámetros y el aprendizaje de los datos de la experiencia [36]. La tabla 2 muestra el resumen de las técnicas de aprendizaje automático aplicadas del año 2007 al 2018 [37].

La Tabla 3 muestra la precisión de predicción entre las técnicas Linear Regression (LR), Artificial neural network (ANN), Support Vector Regression (SVR) y K-Nearest

Tabla 2. Técnicas de aprendizaje automático aplicadas del año 2007 al 2018 [37].

Year	Techniques	Dataset
2007	NeuroFuzzy	NASA&COCOMO
2015		
2008	PSO	NASA
2008	Regression Techniques	COCOMO
2010	Genetic Algorithm	-Turkish % Industry data set -COCOMO & NASA
2013,2015		
2011	ANN	COCOMO & NASA, PROMISE repository
2014		
2011	Fuzzy Logic	NASA
2018		
2013	PSO-ANN-COCOMO II	COCOMO I &NASA93
2014	Radial Basis function NN	COCOMO & NASA
2014	Bayesian with PSO	NASA 93
2014	PCA with ANN	COCOMO 81
2014	Simulated Anncaling	NASA
2014	Hybrid COCOMO and FP	NASA
2015	Bee Colony Optimization	Interactive Voice Response Softwareproject dataset
2016		
2017	Cuckoo, Hybrid Cuckoo Optimization	NASA60, NASA63, NASA93, MAXWELL, KEMERER and MIYAZAKI
2018		
2016	COCOMO TLBO	NASA
2016	Differential Evolution Method	NASA
2016	Hybrid PSO-ANFIS	NASA
2016	Bat Algorithm	NASA 93
2016	COCOMO-GA	NASA
2017	ANN-Dragonfly Algorithm	NASA

Neighbors (KNN) combinados con COCOMO, estableciendo el porcentaje relativo de error [28]. Se observa el algoritmo ANN con mejor rendimiento.

El resultado del trabajo actual radica en la ecuación empírica de COCOMO I y el algoritmo de aprendizaje supervisado de regresión lineal.

COCOMO I

El modelo de esfuerzo se puede escribir en la siguiente forma general:

$$PM = A * Size^B * \prod_{l=1}^P EM_l, \quad (1)$$

Ec. 1 Fórmula general COCOMO [24], donde:

PM = Esfuerzo estimado calculado en meses,

Tabla 3. Comparación de rendimiento entre LR, ANN, SVR y KNN [28].

Performance Comparison	LR	ANN	SVR	KNN
Correct Prediction on Training Data	%74	%87	%95	%68
Correct Prediction on Testing Data	%60	%95	%80	%60

- A = Constante multiplicativa,
- Size = Tamaño estimado del software, medido en KSLOC,
- B = Factores de escala,
- EM = Multiplicadores de esfuerzo.

En COCOMO 81, el término B es una constante exponencial que generalmente es mayor que 1.0, lo que indica deseconomías de escala.

Regresión lineal

El modelo para la regresión lineal múltiple se expresa como:

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \epsilon_i. \quad (2)$$

Ec. 2 Fórmula general de Regresión Lineal Múltiple [24], donde $\beta_0, \beta_1, \dots, \beta_p$ son los coeficientes de regresión, y ϵ_i es el término de error para la i -ésima observación.

1.2. COCOMO

La estimación del esfuerzo es un desafío en cada proyecto de software [7]. Las estimaciones afectarán los costos y las expectativas según el cronograma, la funcionalidad y la calidad. Si bien las estimaciones de expertos se usan ampliamente, son difíciles de analizar y la calidad de la estimación depende de la experiencia de expertos de proyectos similares.

Alternativamente, se pueden utilizar modelos de estimación más formales. Tradicionalmente, el tamaño del software estimado en la cantidad de líneas de código fuente (SLOC), puntos de función (FP) y puntos de objeto (OP) se utilizan como entrada a estos modelos, p. COCOMO y COCOMO II [2].

COCOMO desarrollado por Barry Boehm [3] permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto.

El modelo COCOMO 81 identifica 15 multiplicadores de esfuerzo (ver Tabla 4), mientras que COCOMO II usa 17 en su modelo Post-Arquitectura. Además, podemos transformar los modelos COCOMO en una forma lineal aplicando la transformación logarítmica de ambos lados con la siguiente ecuación (COCOMO 81):

$$\log(PM) = \beta_0 + \beta_1 \log(\text{Size}) + \beta_2 \log(EM_1) + \dots + \beta_{16} \log(EM_{15}). \quad (3)$$

Ec. 3 es transformación algorítmica de COCOMO I [24].

Tabla 4. Valores de costo COCOMO 81 [24].

Cost Driver	Very Low	Low	Nominal	High	Very High	Extra High
ACAP	1.46	1.19	1.00	0.86	0.71	
PCAP	1.42	1.17	1.00	0.86	0.70	
AEXP	1.29	1.13	1.00	0.91	0.82	
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
TURN		0.87	1.00	1.07	1.15	
VIRT		0.87	1.00	1.15	1.30	
STOR			1.00	1.06	1.21	1.56
TIME			1.00	1.11	1.30	1.66
RELY	0.75	0.88	1.00	1.15	1.40	
SCED	1.23	1.08	1.00	1.04	1.10	

1.3. Puntos de función

El tamaño es uno de los atributos del software y se define como un conjunto completo de funcionalidades comerciales que el software proporciona cuando se implementa el método de análisis de puntos de función el cual fue introducido por primera vez por Albrecht en [10, 11]. En este método, el Software se divide en Funciones de datos y Transacciones de datos. Las funciones de datos se dividen en archivos lógicos internos, archivos de interfaz externa.

Cada uno está hecho de tipos de datos y elementos de registro. La función de transacciones se realiza con entradas externas, salidas externas y consultas externas (ver Figura 2).

El tamaño del software se calcula después de identificar la complejidad de los puntos de función y encuentra su suma global que se conoce como puntos de función no ajustados. Los puntos de función luego se ajustan considerando los aspectos técnicos y operativos del proyecto (ver Figura 1).

1.4. Aprendizaje máquina en estimación de software

Para estimar el esfuerzo, la duración y el costo del software, se implementan enfoques algorítmicos y no algorítmicos, son conocidos como modelos de aprendizaje automático para predecir el software EDC (Estimation of effort, Cost and Duration) [1].

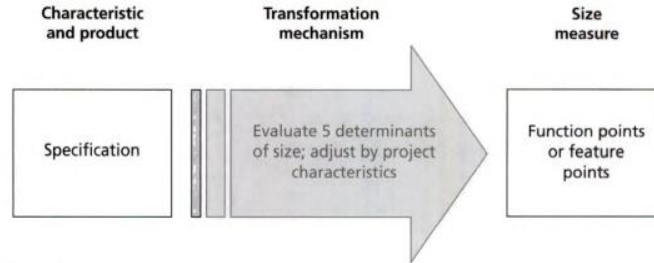


Fig. 1. Transformación de requisitos en puntos de función o características [22].

Type of Input	Count		
	Simple	Average	Complex
Number of external inputs	3	4	6
Number of external outputs	4	5	7
Number of external queries	3	4	6
Number of internal logical files	7	10	15
Number of external interface files	5	7	10

Fig. 2. Recuento inicial de puntos de función [22].

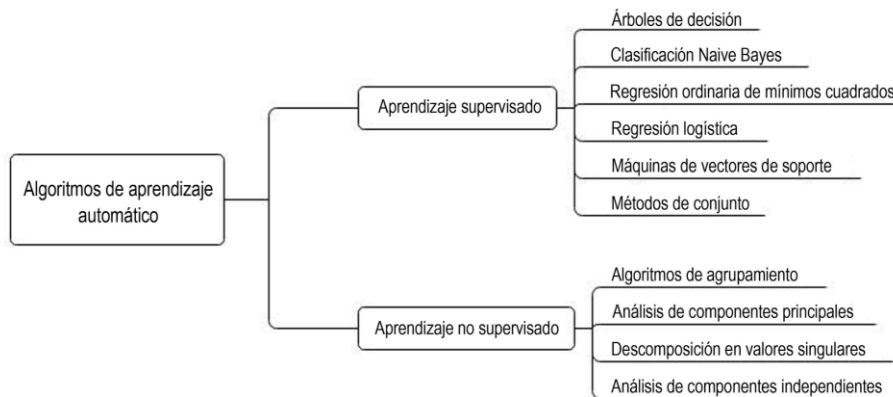


Fig. 3. Algoritmos esenciales en Machine Learning. Creación propia a partir de Raona [23].

El proceso de aprendizaje en ML (Machine Learning) puede partir de una tarea descriptiva o predictiva elegida de una tarea confiable, no supervisada o supervisada. El primero se basa en patrones que no están vinculados a ninguna variable en el conjunto de datos de entrenamiento [17] para fines tales como agrupamiento, reglas de asociación o detección de anomalías.

El aprendizaje supervisado debe tener entradas definidas explícitamente correspondientes a salidas conocidas por una variable objetivo determinada [18] que se utiliza para la clasificación o predicción.

Para ambos hay numerosos algoritmos disponibles que pueden aplicarse según el resultado deseado.

2. Problemática

Cumplir en tiempo y forma el desarrollo de un proyecto de software depende de la cantidad de esfuerzo destinada al mismo, lo cual influye directamente en el tiempo que deba invertirse para finalizarlo con éxito.

Actualmente las empresas que se dedican al desarrollo de software emplean técnicas para estimaciones con la finalidad de predecir variables que permitan proporcionar datos certeros sobre los recursos que serán empleados durante el ciclo de vida de un proyecto. La estimación en proyectos de software es una tarea extremadamente compleja, que requiere, entre otras cosas, disponer de información detallada del proyecto o de los proyectos a estimar, realizar una primera planificación del proyecto y conocer los recursos disponibles [21].

Dependiendo de las técnicas empleadas pueden encontrarse variaciones en los resultados que se arrojan a raíz de su aplicación, provocando la alteración de los datos de vital importancia que pueden llegar a afectar de manera considerable.

El surgimiento de nuevos sistemas integrados a tecnologías emergentes que emplean IA y ML, están tomando la delantera ofreciendo a las empresas la oportunidad de realizar estimaciones de software de manera sencilla, rápida y eficiente, ya que sus motores son entrenados con datos basados en resultados de proyectos de éxito ya finalizados; lo cual provee a dichos sistemas de predicciones certeras para calcular con base en datos del nuevo proyecto un conjunto de salidas más aproximadas a la inversión requerida en cuestión de tiempo, esfuerzo y recursos.

3. Propuesta

El sistema propone mejorar las estimaciones de las variables de duración y consumo de recursos al momento de desear realizar un desarrollo de software; esto basándose en datos obtenidos de proyectos de éxito ya concluidos, para lo cual se pretende emplear algoritmos predictivos de ML como lo es Regresión Lineal.

La intención es ofrecer una alternativa a los problemas de estimación ya mencionados a lo largo del documento, la solución propuesta es ofrecer variables de estimación con un porcentaje alto de confiabilidad que a la vez coadyuve a la agilización en el proceso de generación de datos a partir de la utilización de técnicas de inteligencia artificial para sustituir la tarea de cálculos manuales.

4. Metodología

Como parte de la metodología se utilizó regresión lineal dado que la estimación del esfuerzo del software se reconoce como un problema de dicha naturaleza [12]. A continuación, se muestra la secuencia de pasos en la construcción del modelo. (Ver Figura 4).

4.1. Dataset

El término dataset se refiere a un archivo que contiene uno o más registros [20]. El dataset en esta investigación fue recopilado por Jairus Hihn y donado a The PROMISE

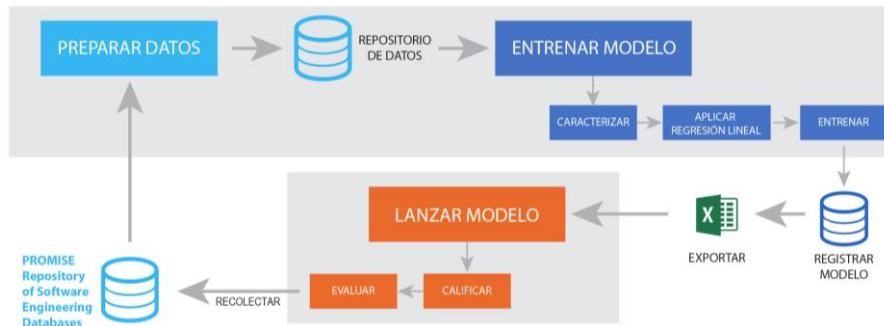


Fig. 4. Metodología para la construcción del Modelo inteligente. [26].

Repository of Software Engineering Databases [16]. Cubre información sobre 93 proyectos de desarrollo de software documentados por la NASA, entre los años 1971 y 1987, los cuales incluyen 24 atributos que cubren la información de los proyectos, 15 atributos discretos estándar de COCOMO I en el rango Very_Low hasta Extra_High, 7 que describen el proyecto; una línea de medida de código y un campo objetivo del esfuerzo real en persona meses.

Los proyectos de la NASA se utilizan para entrenar y probar el uso de técnicas inteligentes de estimación de esfuerzo en la producción de software [28, 29]. Se puede argumentar el overfitting en este caso con el hecho de que se utilizó ese dataset solamente para entrenamiento y en la actualidad los datos que alimentan es decir el 20% es para el testing.

4.2. Técnicas aplicadas

La construcción del modelo y el entrenamiento se realizó por medio de Microsoft Azure Machine Learning Studio (Classic). Azure ML [25] es una herramienta colaborativa que se basa en el manejo interactivo de componentes para crear, probar e implementar soluciones de análisis predictivo en datos; está diseñado para trabajar con datos tabulares, como datos de texto delimitados o datos estructurados de una base de datos.

Azure Machine Learning Studio publica modelos como web services que pueden generarse desde la misma herramienta. Para el proyecto se realizó la descarga del archivo excel que al ser ejecutado en local despliega un apartado del lado derecho que indica que el archivo se encuentra listo para la conexión con el web service de Azure ML que se creó, los nuevos datos de entrada son agregados en la hoja de cálculo de acuerdo al orden de las categorías que incluidas en el dataset usado, desde donde serán tomados para entregar el resultado de la predicción.

En la figura 5 se muestra el proceso de construcción del modelo para el cual fue necesario:

- Depurar el conjunto de datos, seleccionando las columnas numéricas,
- Fraccionar el conjunto de datos en un 0.8 para el entrenamiento,

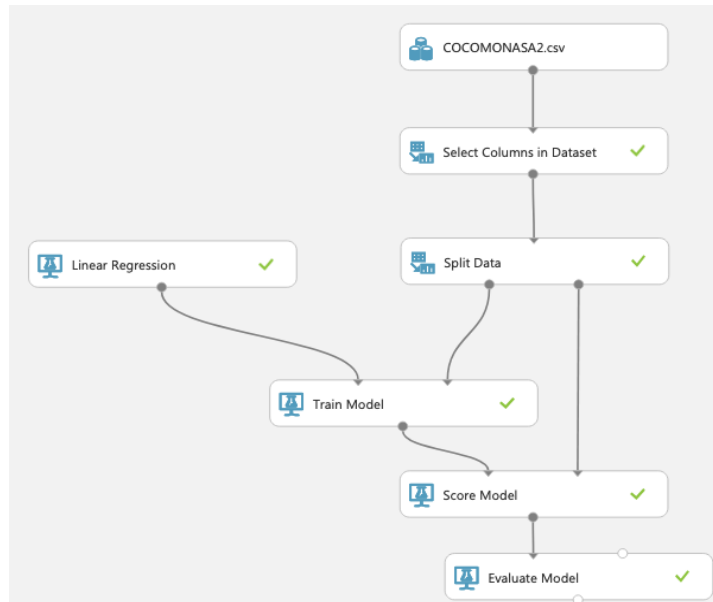


Fig.5. Modelo inteligente. Creación propia.

Tabla 5. Headers y contenido del dataset.

Headers		Contenido (Valores)	
Id	projectname	1-100	Nombre del Proyecto
System type	NASA center	Ground System(g), Flight System(f)	1 to 5
year	mode	Año del proyecto	Embedded, Organic, Semidetached
	Category	Avionics, application_ground, avionicsmonitoring, batchdataprocessing, communications, datacapture, launchprocessing, missionplanning, monitor_control, operatingsystem, realdataprocessing, science, simulation, utility.	

- Aplicar regresión lineal,
- Entrenar el modelo mediante la variable KSLOC,
- Calificar el modelo,
- Evaluar el modelo.

4.3. Prueba y entrenamiento del modelo propuesto

Como parte del proceso de entrenamiento y pruebas, el dataset fue sometido a una serie de pasos, los cuales se indican en la siguiente lista, con los cuales fue posible obtener resultados.

Tabla 6. Atributos generales y su descripción.

Siglas	Nombre	Descripción
RELY	Required Software ReliabiLiTY	Prueba de Confianza del Software en un periodo de tiempo y ambiente específico.
DATA	DATAbase size	Tamaño de la base de datos.
CPLX	Process COMPLExity	Evalúa las tareas o eventos por los que fluye un producto o actividad antes de pasar al estado de salida.
TIME	TIME Constraint	Restricción de Tiempo para CPU, acción dinámica que solicita un thread y un scheduler por un intervalo de tiempo.
STOR	Main MemORy ConSTraint	Limita el tamaño de la memoria utilizada buscando mejorar el rendimiento.
VIRT	VIRTual Machine Volatility	Refiere al complejo de Hardware y Software.
TURN	Computer TURNaround time	Tiempo de respuesta de la computadora.

Tabla 7. Atributos del personal y su descripción.

Siglas	Nombre	Descripción
ACAP	Analyst CAPability	Prueba de Confianza del Software en un periodo de tiempo y ambiente específico.
AEXP	Application EXPerience	Toma experiencia de los campos UI, UX Networking, Middleware, así como lenguajes y herramientas.
PCAP	Programming CAPability	Toma la productividad de una persona y su capacidad para construir software.
VEXP	Virtual Machine EXPerience	Mide la experiencia del personal en el uso, configuración y familiaridad con máquinas virtuales.
LEXP	Programming Language EXPerience	Refiere a la experiencia (Intern, Jr., SSr., Sr.) respecto a los lenguajes de programación del proyecto.
MODP	MODern Programming Practices	Prácticas de programación modernas se puede entender mejor como “Practicas maduras de Ingeniería de Software”.
TOOL	Use Software TOOLS	Herramientas que saben y pueden usar tanto en testing, desarrollo y maquetación.
SCED	Required Development SChEDule	Esta variable es única e importante, ya que un proyecto con un tiempo acelerado requerirá más esfuerzo que un proyecto en su tiempo óptimo.

Tabla 8. Atributos de desarrollo y su descripción.

Siglas	Nombre	Descripción
KSLOC	Thousan (K) Lines of Code	Esta variable es importante, ya que un proyecto con un tiempo acelerado requerirá más esfuerzos qwue un proyecto en su tiempo óptimo.
Effort	Effort per Month	Este último valor, mostrara el esfuerzo requerido en meses. Cada mes es equivalente a 152, que incluye horas de desarrollo y gestión (Development & Management).

Etapas:

- 1.- Pre procesar datos (Datos necesarios),
- 2.- Garantizar valores numéricos (int, float),
- 3.- Normalizar etiquetas,
- 4.- Evitar datos vacíos.

El conjunto de datos sometido a entrenamiento y prueba se describe a continuación: En la tabla 5, podemos apreciar los primeros headers, los cuales hablan respecto a información de los proyectos.

A continuación, se muestran 15 atributos de COCOMO I los cuales tendrán un rango desde Very Low hasta Extra High, los rangos posibles son:

vl	- Very Low
l	- Low
n	- Nominal
h	- High
vh	- Very High
xh	- eXtra High

En la tabla 6 se puede ver los atributos generales del proyecto. En la tabla 7 se muestran los atributos del personal. En la tabla 8 podemos ver los atributos de desarrollo del proyecto.

5. Resultados

Este trabajo muestra como resultado la implementación de un modelo inteligente que permite determinar el esfuerzo de desarrollo de software mediante machine learning. Los resultados del experimento al cual fue sometido el conjunto de datos bajo el enfoque de regresión lineal fue prometedor dado que en el mejor resultado mostró una precisión del .40 en el coeficiente de determinación. Por supuesto es importante hacer hincapié en la búsqueda e implementación de otras técnicas y la modificación en cuestión de configuración dentro del modelo propuesto, para esperar como resultado el incremento de la exactitud que se busca.

6. Conclusiones

Se resalta que la implementación de Inteligencia Artificial y Machine Learning comenzará a convertirse en un soporte primordial en el desarrollo de herramientas que coadyuven en la ejecución de procesos dentro del área de la ingeniería de software. Mediante el desarrollo de sistemas inteligentes como el propuesto en el presente trabajo se busca impulsar la idea de desarrollos que además de ser innovadores se centren en

la necesidad de garantizar resultados certeros, mejorando de esta forma la estimación de recursos durante el proceso de desarrollo de software.

7. Trabajo futuro

Se contempla comunicar el modelo mediante un web service customizado que desde el servidor se comunice con Python mientras que el cliente permitirá la entrada de datos a través de un formulario construido a partir de HTML, CSS y JS.

La finalidad del desarrollo web es realizar el intercambio de datos de entrada en la UI y la salida de los resultados generados por el modelo, mejorando la UX.

Además de que considera ideal que sea de uso libre lo cual implica el pago de un servidor público para albergar el sistema y ofrecerlo a cualquier persona que desee realizar estimaciones con base en los datos de su proyecto.

Referencias

1. Prasad, B.M.G., Sreenivas, P.V.S.: An implementation of software effort duration and cost estimation with statistical and machine learning approaches. *International Journal of Computer Engineering and Technology*, 10(1), pp. 81–93 (2019)
2. Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R.: *Cost models for future software life cycle processes: COCOMO 2.0*. USC Center for Software Engineering (1995)
3. Boehm, B., et al.: *Software Cost Estimation in COCOMO II*. Prentice Hall (2000)
4. Galorath, D., Evans, M.W.: *Software sizing, estimation, and risk management*. Auerbach Publications (2006)
5. Parthasarathy, M.A.: *Practical software estimation*. Addison-Wesley, pp. 47–72 (2007)
6. Nasir, M.: A survey of software estimation techniques and project planning practices. *Proc. 7th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD)*, pp. 305–10, IEEE Computer Society (2006)
7. Parastroo Mohagheghi, Bente Anda, Reidar Conradi: Effort estimation of use cases for incremental large-scale software development. *27th International Conference on Software Engineering* (2005)
8. Putnam, L.H.: A general empirical solution to the macro software sizing and estimating problem. *IEEE Transactions on Software Engineering*, 4, pp. 345–361 (1978)
9. Du, W.L., Capretz, L.F., Nassif, A.B., Ho, D.: A hybrid intelligent model for software estimation. *Journal of Computer Science*, 9 (2013)
10. Albrecht, A.: Measuring application development productivity. In *IBM Application Development Symp.*, 1079, pp. 83–92 (1979)
11. Albrecht, A.J., Gaffney, J.E.: Software function, source lines of code, and development effort prediction: a software science validation. *IEEE transactions on software engineering*, 6, pp. 639–648 (1983)
12. Adriano, L.O., Petrónio, L.B., Ricardo, M.F., Márcio, C.: GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Information & Software Technology*, 52(11), pp. 1155–1166 (2010)
13. Nassif, A.B., Capretz, L.F., Ho, D.: Software estimation in the early stages of the software life cycle. *Nanded, Maharashtra, India*, pp. 5–13 (2010)
14. Peraphon Sophatsathit: An exploratory survey of phase-wise project cost estimation techniques. *Advanced Virtual and Intelligent Computing (AVIC) Center* (2014)

15. Somya Goyal, Anubha Parashar: Machine learning application to improve COCOMO model using neural networks. *Information Technology and Computer Science*, 3, pp. 35–51 (2018)
16. Hihn, J.: COCOMO NASA 2 / Software cost estimation. *The Promise Repository of Software Engineering Databases* (2006)
17. Linoff, G.S., Berry, M.J.A.: *Data mining techniques: for marketing sales, and customer relationship management*. John Wiley & Sons (2011)
18. Cios, K., Pedrycz, W., Swiniarski, R., Kurgan, L.: *Data mining a knowledge discovery approach*. Springer (2007)
19. Grimstad, S., Jorgensen, M.: Inconsistency of expert judgment-based estimates of software development effort. *J Syst Software*, 80(11), pp. 1770–1777 (2007)
20. IBM Knowledge Center: What is a dataset?. https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zconcepts/zconc_datasetintro.htm (2016)
21. Ruíz, Y.C., Cordero, M.D.: Software projects estimation integrating Bohem and Humphrey method's. *Revista Cubana de Ciencias Informáticas*, 7(3) (2013)
22. Lawrence, P.S., Wu, F., Lewis, R.: *Software cost estimation and sizing methods: issues and guidelines*. Santa Monica (2005)
23. Raona: Visible body: Los 10 Algoritmos Esenciales en Machine Learning. <https://www.raona.com/los-10-algoritmos-esenciales-machine-learning/> (2017)
24. Nguyen, V., Steece, B., Boehm, B. A constrained regression technique for cocomo calibration. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'08)*. Association for Computing Machinery, pp. 213–222 (2008)
25. Microsoft: Visible body: What is Machine Learning Studio (classic)?. <https://docs.microsoft.com/es-es/azure/machine-learning/studio/what-is-ml-studio> (2020)
26. MCKnight, W.: Visible body: GIGAOM: Delivering on the Vision of MLOps v1.0 A maturity-based approach. <https://gigaom.com/report/delivering-on-the-vision-of-mlops/> (2020)
27. Scikit-Learn: <https://scikit-learn.org/stable/> (2020)
28. Zeynab Abbasi Khalifelua, F.S.: Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation. *ScienceDirect*, pp. 65–71 (2012)
29. Sheta, A.: Estimation of the COCOMO model parameters using genetic algorithms for nasa software projects. *Journal of Computer Science*, pp. 118–123 (2006)
30. Arnuphaptrairong, T.: A literature survey on the accuracy of software effort estimation models. *International MultiConference of Engineers and Computer Scientists*, (2016)
31. Sweta-Kumari, S.P.: Comparison and analysis of different software cost estimation methods. *International Journal of Advanced Computer Science and Applications*, pp. 153–157 (2013)
32. Kemerer, C.F.: An empirical validation of software cost estimation models. *Communication of the ACM*, 30(5), pp. 416–429 (1987)
33. Chandrasekaran, R., Kumar, R.V.: On the estimation of the software effort and schedule using constructive cost Model-II and function point analysis. *International Journal of Computer Applications*, 44(9), pp. 3844 (2012)
34. Arnuphaptrairong, T.: Early stage software effort estimation using function point analysis: an empirical validation. *International Journal of Design, Analysis and Tools for Integrated Circuits and Systems*, 4(1), pp. 15–21 (2013)
35. MayaZaki, Y., Mori, K.: COCOMO Evaluation and tailoring. In *Proceeding of the 8th International Conference on Software Engineering of the IEEE*, pp. 292–299 (1985)
36. Satapathy, S.M.: *Effort estimation methods in software development using machine learning algorithms*. Department of Computer Science and Engineering, National Institute of Technology Rourkela (2016)

Modelo inteligente para determinar el esfuerzo de desarrollo de software mediante machine learning

37. Siti Hajar Arbain, N.A.: Adoption of machine learning techniques in software effort estimation: an overview. IOP Conference Series: Materials Science and Engineering (2019)

Selección de acciones para la navegación de un robot móvil basada en fuzzy Q-learning

Elizabeth López Lozada, Elsa Rubio Espino, Juan Humberto Sossa Azuela,
Víctor Hugo Ponce Ponce

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

elizabeth.l.lozada@gmail.com,
{erubio, hsossa}@cic.ipn.mx

Resumen. Los robots móviles deben contar con la capacidad de moverse dentro de un entorno de trabajo bajo la menor intervención de un supervisor humano, para ello se les provee de algoritmos y herramientas que les permiten tomar decisiones de acuerdo a su estado actual y a las variables de su entorno. En este artículo, se describe una arquitectura para la selección de acciones con el método de Fuzzy Q-Learning (FQL) con el fin de que un robot móvil pueda tomar decisiones con base en el nivel de carga de batería y pueda escoger entre desplazarse hacia su objetivo principal, ir a una estación de carga de baterías o detenerse. Adicionalmente, como entrada se toman en consideración las distancias que existen entre el robot y los posibles destinos. Para lograrlo, en la arquitectura propuesta se integró un módulo de planificación de ruta que utiliza el método de campos potenciales artificiales, para que un robot pueda navegar de forma reactiva sobre su entorno de trabajo.

Palabras clave: Robots Móviles Navegación Planificación de Ruta Fuzzy Q-Learning Campos Potenciales Artificiales.

Selection of Actions for the Navigation of a Mobile Robot Based on Fuzzy Q-Learning

Abstract. Mobile robots must have the ability to move within a work environment under the least intervention of a human supervisor, for this they are provided with algorithms and tools that allow them to make decisions according to their current state and the variables of its surroundings. In this article, an architecture for the selection of actions with the Fuzzy Q-Learning (FQL) method is described so that a mobile robot can make decisions based on the battery charge level and can choose between moving towards your main objective, going to a battery charging station or stopping. Additionally, the distances between the robot and the possible destinations are taken into consideration as input. To achieve this, a route planning module was integrated into the proposed architecture that uses the artificial

potential fields method, so that a robot can reactively navigate its working environment.

Keywords: Mobile robots, Navigation, Route planning, Fuzzy Q-learning, Artificial potential fields.

1. Introducción

Los robots móviles se usan en una gran cantidad de aplicaciones, por ejemplo, en vigilancia, exploración, operaciones de búsqueda y rescate, limpieza, automatización industrial, construcción o en museos. Para cumplir con sus tareas, deben contar con algoritmos robustos que les permitan navegar a través de su entorno y evadir los obstáculos que se encuentren en el camino. La navegación para los robots móviles incluye todas las acciones que llevan a que un robot se desplace desde su posición actual hasta el destino y se puede dividir en dos tipos: global y local [1,3].

Una de las tareas esenciales de la navegación en robots móviles es la planificación de rutas, cuyo objetivo es generar las entradas de referencia que determinen una ruta libre de colisiones, para que un robot ejecute la trayectoria planeada [3]. Para determinar el desplazamiento del robot y que pueda evadir obstáculos se han usado sistemas de inferencia difusos [12,13], aprendizaje reforzado [4, 6, 16], campos potenciales artificiales [7, 8, 9, 10, 14] y redes neuronales [15]. Cada método tiene sus ventajas y desventajas, por ejemplo, el método de campos potenciales artificiales se usa para la planificación de rutas, ya que es simple y tiene un costo computacional bajo [10].

En todos estos casos, los autores sólo consideran la información proveniente de los sensores de distancia, sin embargo, no consideran la descarga eléctrica de la batería que ocurre durante la operación del robot. En [11], determinan que el robot hace una planificación de la trayectoria donde hay un punto de partida, cinco nodos que el robot tiene que visitar, y el último nodo corresponde a una estación de servicio donde el robot puede reponer su batería. Estos robots deben contar con dos características importantes: autonomía y autosuficiencia. Con esto, surge la pregunta ¿puede un robot aprender a seleccionar acciones de forma autónoma? La respuesta es sí, de manera que este trabajo busca mostrar una forma en que un robot podría aprender a tomar decisiones en base a prueba y error. Es por ello, que en este documento se presenta una arquitectura basada en el método de Fuzzy Q-Learning (FQL) junto con un módulo para la planificación de rutas, a fin de que un robot tenga la capacidad para tomar decisiones de forma independiente en base al nivel de batería y seleccione entre ir a la estación de carga de baterías, ir a un destino predeterminado o esperar.

Este artículo está organizado como sigue: en la sección 2, se presentan los conceptos y las ecuaciones básicas del método de Fuzzy Q-Learning; en la sección 3, se presenta el método de campos potenciales que se utiliza para la planificación de la trayectoria del robot hacia el destino seleccionado; en la sección 4, se describe la arquitectura propuesta en este trabajo; en la sección 5 se presentan las simulaciones y los resultados obtenidos; y finalmente en la sección 6 se dan las conclusiones obtenidas durante el desarrollo de este trabajo.

2. Fuzzy Q-Learning

El método de Fuzzy Q-Learning (FQL) [2] puede verse como una extensión de los sistemas de inferencia difusos (FIS), en donde las reglas difusas definen el estado del agente de aprendizaje. Cuando las entradas de las reglas son convertidas en valores difusos, caen en una regla difusa que corresponde a un estado; la fuerza de la regla difusa α_i , ayuda a definir el grado de que el agente se encuentre en un estado. Con esto el sistema de aprendizaje escoge una acción a_i del conjunto de acciones A en cada regla, se llama $a[i,j]$ a la j -ésima acción posible en la regla i y $q[i,j]$ a su correspondiente valor q . Con esto, el FIS se construye de la siguiente forma:

Si x es S_i entonces $a[i,1]$ con $q[i,1]$ o ... o $a[i,j]$ con $q[i,j]$.

Al final, el agente de aprendizaje debe encontrar la mejor solución para cada regla, es decir, la acción con el mejor valor q .

El valor q se obtiene de una tabla que contiene $i \times j$ valores q , las dimensiones de la tabla corresponden al número de reglas por cantidad de acciones. Las acciones se seleccionan con una política de exploración-explotación, que está basada en la calidad del par estado-acción. Se propone que la probabilidad de exploración-explotación esté dada por $\varepsilon = 10/(10 + T)$, en donde T corresponde al número del paso. Esta probabilidad se usa para compensar entre la exploración y el control del algoritmo, y eliminar la exploración de forma gradual. Para calcular la acción inferida y el valor q , se ocupan las ecuaciones en (1):

$$a(x) = \frac{\sum_{i=1}^N \alpha_i \times a(i, i^o)}{\sum_{i=1}^N \alpha_i(x)}, \quad Q(x, a) = \frac{\sum_{i=1}^N \alpha_i \times q(i, i^o)}{\sum_{i=1}^N \alpha_i(x)}, \quad (1)$$

en donde $Q(x, a)$ es una función evaluada en x y a ; x es el estado o regla y a es la acción inferida; i es el estado o regla en el que se encuentra el agente; i^o es el índice de la acción seleccionada; α_i es la fuerza de la regla; y N es un número positivo, $N \in \mathbb{N}^+$ y corresponde al número total de reglas. La función de valor-estado se calcula con la ecuación (2):

$$V(x, a) = \frac{\sum_{i=1}^N \alpha_i(x) \times q(i, i^*)}{\sum_{i=1}^N \alpha_i(x)}, \quad (2)$$

en donde i^* corresponde al índice de la acción óptima, es decir, la acción que tiene el valor q más alto y x es un estado. Y la derivada de la función Q se define como en la ecuación (3), en donde r corresponde a la recompensa y γ es el factor de descuento:

$$\Delta Q = r + \gamma V(x, a) - Q(x, a). \quad (3)$$

Con la ecuación (4) se obtiene un valor de elegibilidad $e[i, j]$, que se usa durante la actualización del valor q en donde γ es un parámetro, $0 \leq \gamma \leq 1$, llamado factor de descuento, λ es el parámetro de decaimiento, en donde $\lambda \in [0, 1]$, i es el número de la regla y j es la acción seleccionada:

$$e[i, j] = \begin{cases} \lambda \gamma e[i, j] + \frac{\alpha_i(x)}{\sum_{i=1}^N \alpha_i(x)}, & \text{si } j = i^o, \\ \lambda \gamma e[i, j], & \text{en otro caso.} \end{cases} \quad (4)$$

Finalmente, para actualizar el valor q se utiliza el valor de elegibilidad $e[i, j]$, la derivada de la función Q y una pequeña fracción positiva $\epsilon \in (0, 1]$, la cual influye en la tasa de aprendizaje. En la ecuación (5) se muestra la función de actualización del valor q :

$$\Delta q[i, j] = \epsilon \times \Delta Q \times e[i, j]. \quad (5)$$

3. Campos potenciales artificiales

El uso de los campos potenciales en la navegación de robots fue propuesto por Khatib [5] en 1986, en su trabajo presentó el concepto de campos potenciales artificiales para que robots móviles evadan obstáculos en tiempo real. Su teoría se fundamenta en el uso de fuerzas de atracción para alcanzar el objetivo y en fuerzas de repulsión para evitar colisionar con los obstáculos, y que pueden ser calculadas a través de un conjunto de ecuaciones sencillas. La fuerza de atracción es definida en la ecuación (6) como una función relativa de distancia entre el robot y el objetivo, en donde básicamente el robot es atraído hacia el objetivo por el campo de atracción que este tiene [7]. La fuerza de repulsión, en la ecuación (7), es la que indica al robot la presencia de obstáculos en un umbral definido y lo empuja lejos de los obstáculos. Finalmente, se calcula una fuerza resultante que equivale a la suma de las fuerzas de atracción y repulsión:

$$F_{attr}(q, g) = -\xi p(q, g), \quad (6)$$

en donde ξ es un factor de escala positivo; $p(q, g)$ es la distancia euclidiana entre el robot y el objetivo; q es la posición del robot; y g es la posición del objetivo:

$$F_{rep}(q) = \begin{cases} \eta \left(\frac{1}{p(q_i, p_{o_j})} - \frac{1}{p_{o_j}} \right) \frac{p^2(q_i, p_{o_j})}{p(q_i, p_{o_j})}, & \text{si } d < p_{o_j}, \\ -\frac{\eta}{p_{o_j}}, & \text{si } d = p_{o_j}, \\ 0, & \text{si otro caso,} \end{cases} \quad (7)$$

en donde η es el factor de repulsión; $p(q_i, p_{o_j})$ es la distancia entre el robot y el obstáculo; p_{o_j} es el umbral entre el robot y el radio del obstáculo; y d es la distancia medida.

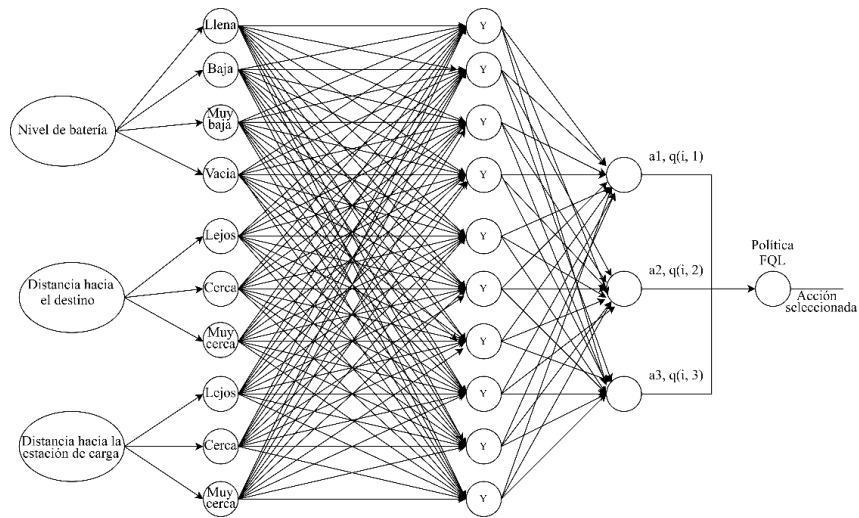


Fig. 1. Arquitectura del sistema de Fuzzy Q-Learning.

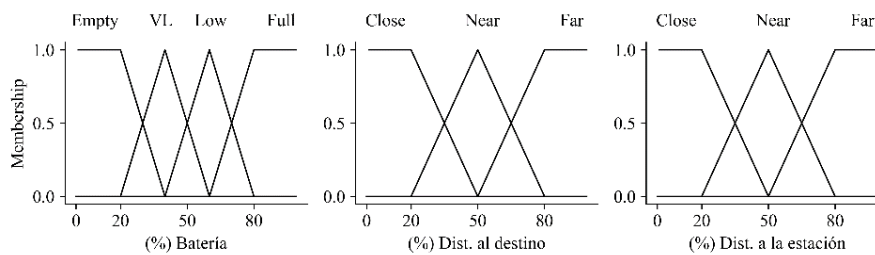


Fig. 2. Funciones de membresía.

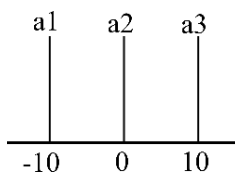


Fig. 3. Funciones singleton de las salidas.

4. Arquitectura propuesta

La arquitectura propuesta se representa en el diagrama de la Figura 1. Las entradas están definidas como el nivel de batería, la distancia hacia el objetivo y la distancia hacia la estación de carga de la batería, las cuales están conectadas a las funciones de membresía correspondientes.

Para la primera entrada, están definidos cuatro conjuntos difusos que corresponden al nivel de la batería llena, baja, muy baja y vacía; la segunda y tercera entrada se conectan a las funciones de membresía lejos, cerca y muy cerca. En total se tienen 36 reglas difusas y hay tres posibles acciones que están asociadas a un valor numérico q , que pueden ser seleccionadas de acuerdo con la política de exploración-explotación descrita en la sección 2.

Algoritmo 1. Navegación en base al nivel de batería.

Entrada:

robot \leftarrow Posición del robot
objetivo \leftarrow Posición del objetivo
estación \leftarrow Posición de la estación de carga
obstáculos \leftarrow Posición de los obstáculos
 $\gamma \leftarrow$ Factor de descuento
 $\alpha \leftarrow$ Tasa de aprendizaje
Inicialización del sistema.
destino \leftarrow objetivo.
mientras robot \neq destino **hacer**
 estado \leftarrow Obtener el estado actual.
 acción \leftarrow Seleccionar una acción
 destino \leftarrow Actualizar el destino
 salida \leftarrow Calcular la salida global usando la ecuación (1)
 $q \leftarrow$ Calcular el valor Q con la ecuación (2)
 Ejecutar la acción
 nuevo estado \leftarrow Obtener el nuevo estado.
 recompensa \leftarrow Obtener la recompensa con la función (9)
 valor estados \leftarrow Calcular con la ecuación (3)
 $\Delta Q \leftarrow$ Calcular delta Q con la ecuación (6)
 elegibilidad \leftarrow obtener con la ecuación (5)
 $\Delta q[i, j] \leftarrow \epsilon \times \Delta Q \times$ elegibilidad
 Actualizar el valor Q

En la Figura 2 se muestran los conjuntos que forman las funciones de membresía para cada entrada. La figura izquierda corresponde a la función de membresía para el nivel de carga de batería, la figura central es para la entrada de la distancia hacia el destino y la figura derecha es para la distancia hacia la estación de carga.

Las posibles salidas están definidas por las funciones de tipo singleton, que permiten asociar un valor numérico a una salida puntual, en este caso a cada una de las acciones que puede ejecutar el sistema, en la Figura 3 se muestra el conjunto de funciones que se utilizan.

Mientras que la función de recompensas está dada por la expresión (8), con $\gamma = 0.5$ y $\alpha = 0.01$.

$$r(t) = \begin{cases} +10, & \text{si el robot esta cerca del objetivo,} \\ +5, & \text{si el robot esta cerca de la estación,} \\ -20, & \text{si el robot esta alejado y el nivel de batería es muy bajo,} \\ 1, & \text{en otro caso.} \end{cases} \quad (8)$$

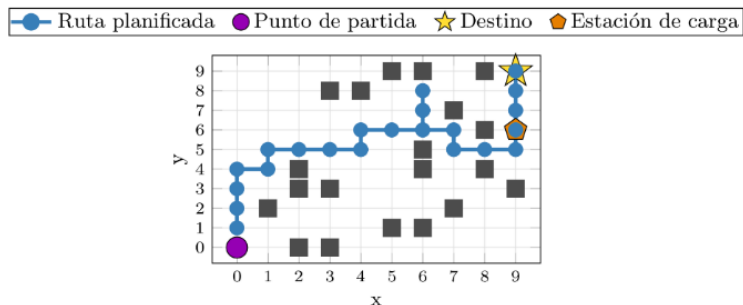


Fig. 7. Ruta que toma el agente para llegar al destino.

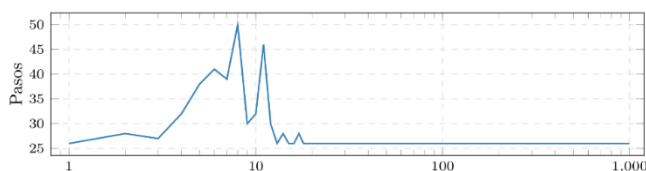
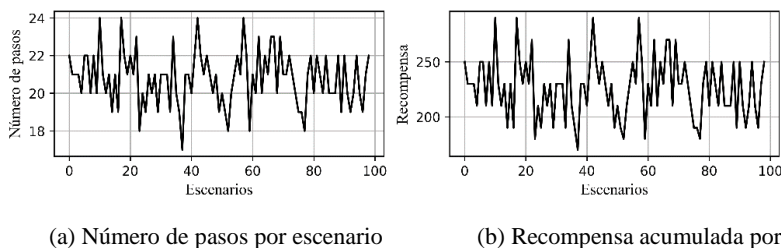


Fig. 8. Pasos para que el robot complete la tarea durante cada época.



(a) Número de pasos por escenario

(b) Recompensa acumulada por escenario

Fig. 9. Comportamiento del robot en diferentes escenarios.

Finalmente, el Algoritmo 1 muestra el conjunto de pasos que se siguen para llegar al destino usando FQL y el módulo de campos potenciales.

5. Simulaciones y resultados

A continuación, se colocan los resultados obtenidos de las simulaciones en un escenario que cuenta con 20 obstáculos, un objetivo y una estación de carga, los cuales están representados por cuadrados, una estrella y un pentágono respectivamente.

Para fines ilustrativos, las figuras que se muestran en esta sección corresponden a uno de los escenarios en donde el sistema fue simulado.

La generación de ruta es uno de los primeros pasos que se realiza, por ello en la Figura 4a se muestra la ruta generada para llegar a la posición objetivo y en la Figura 4b la ruta generada para llegar a la estación de carga.

En la Figura 5 se muestran las acciones seleccionadas y la recompensa acumulada durante el desplazamiento, en donde se observa una curva ascendente hasta que llega al

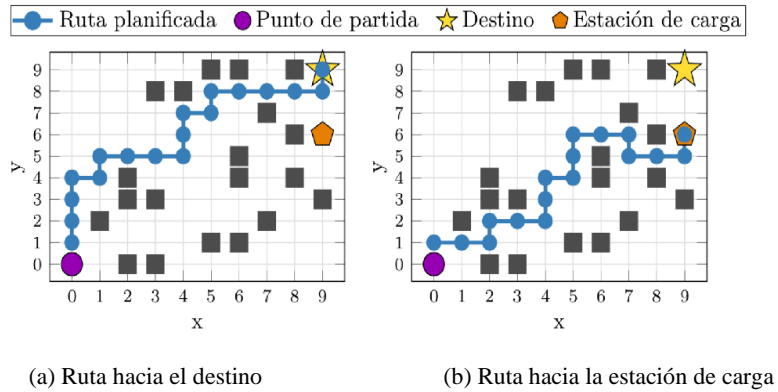


Fig. 4. Rutas generadas.

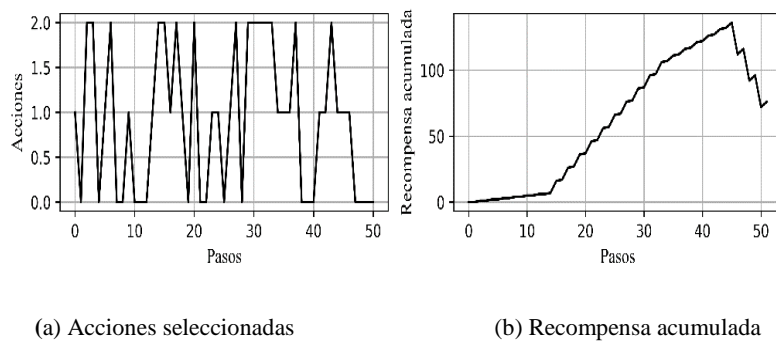


Fig. 5. Comportamiento del robot durante la simulación.

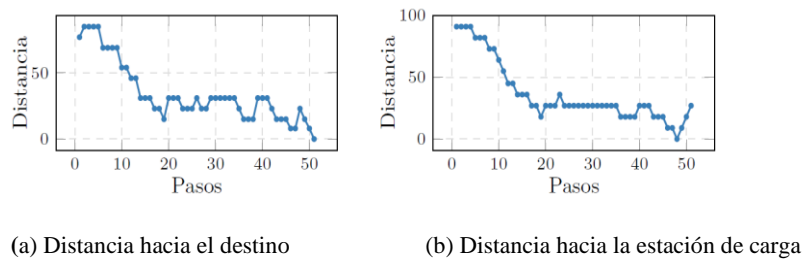


Fig. 6. Comportamiento de las distancias durante la simulación.

paso 45. A partir de este paso la recompensada asignada es negativa, lo cual se debe a que el nivel de batería es bajo porque el agente no ha logrado llegar a su destino.

El comportamiento de las distancias durante la simulación se muestra en la Figura 6 y se observa como disminuye la distancia entre el destino y el robot durante cada paso de la simulación.

A partir del paso 45, se observa que la distancia entre el robot y la estación de carga se acorta, mientras que la distancia entre el robot y el objetivo se incrementa, sin embargo, después de que el robot llega la estación de carga continua su camino hasta llegar al objetivo.

En la Figura 7 se muestra la ruta que tomó el robot hasta llegar al objetivo, la cual es diferente a la mostrada en la Figura 4a causa de las acciones seleccionadas por el módulo de FQL. Se aprecia que el agente llegó a la estación de carga de batería y después siguió su camino hacia el objetivo.

Y en la Figura 8 se muestran los resultados de la simulación después de que el destino fue alcanzado durante 1000 épocas. Mientras el robot aprende la cantidad de pasos es variante y a partir de la época número 18, los pasos que le toma al robot completar su tarea se estabilizan en 26 pasos.

Para finalizar, en la Figura 9 se muestra el comportamiento del sistema bajo 100 escenarios distintos con un total de 20 obstáculos en cada escenario. En la Figura 9a, se visualiza el número de pasos que le tomó al sistema llegar al destino después de un entrenamiento de 1000 épocas, en cada uno de los 100 escenarios.

Mientras que, en la Figura 9b se muestra la recompensa acumulada durante la última época de entrenamiento de cada escenario.

6. Discusiones

Este trabajo presentó un sistema de navegación para robots móviles basado en una arquitectura FQL, que permite a un robot tomar decisiones de forma autónoma en base a su nivel de carga de batería mientras se mueve de un punto de partida a un punto de destino. Utilizando FQL, el sistema aprende a través del ensayo-error con un paradigma de aprendizaje de refuerzo, en el que la definición de la función de recompensa tuvo un papel importante en el proceso de aprendizaje del sistema.

Se probaron diferentes funciones hasta que se eligió la función (8). En los escenarios usados los obstáculos y los posibles destinos se colocaron en diferentes posiciones. Las variaciones que se presentan de la ruta que se genera con método de planificación de ruta. En algunos casos cuando el robot se encuentra muy cerca de la estación de carga y el valor difuso del nivel de batería cae en un conjunto diferente a lleno, el robot sigue la ruta hacia la estación de carga.

Entre todas las ventajas que se encontraron al seleccionar esta arquitectura, se distingue la posibilidad de que el experto pueda definir el número de estados en los que puede caer el sistema, así como el número de reglas por las que se compone el FIS. Si se utiliza un Q-learning clásico, el número de estados podría ampliarse al intervalo de mediciones del voltaje de la batería, es decir, 100 estados si se usa una escala de nivel porcentual entera de 0 a 100, o incluso más estados, si el dispositivo de medición de la carga dispone de una escala de milivoltios y se ocupan los milivoltios como estados.

Entre las desventajas, el sistema podría no elegir necesariamente el camino más corto en todo momento, ya que en algunos pasos el agente de aprendizaje puede seleccionar permanecer en modo de espera mientras se determinaba el destino a seguir. Sin embargo, con el aprendizaje adquirido, el sistema logra seleccionar las acciones que le permiten completar su tarea.

7. Conclusiones

La arquitectura propuesta para la toma de decisiones ayuda a que un robot móvil sea capaz de seleccionar entre un conjunto de acciones, las cuales le permiten cumplir su objetivo, en este caso el de llegar a un destino predeterminado. Al usar el método de fuzzy Q-Learning la complejidad del sistema es asignada por el número de reglas que se definieron.

A diferencia del método de Q-Learning clásico, en donde los estados corresponderían al total de posibles mediciones del nivel de batería, con el método propuesto se limita la cantidad de estados a los rangos asignados con las funciones de membresía. Se muestra que el método propuesto ayuda a que un robot aprenda a seleccionar tareas de forma autónoma y con esto complete su tarea. En trabajos futuros el número de acciones que tiene el sistema pueden ser aumentadas en caso de que el robot tenga que cumplir con otras tareas.

Agradecimientos. Se agradece el apoyo prestado para desarrollar este proyecto al Instituto Politécnico Nacional (IPN), al Consejo Nacional de Ciencia y Tecnología (CONACYT), y a la Secretaría de Investigación y Posgrado (SIP) a través de los proyectos 20180943, 20190007, 20195835, 20200630 y 20201397

Referencias

1. Agarwal, D., Bharti, P.S.: Nature inspired evolutionary approaches for robot navigation: survey. *Journal of Information and Optimization Sciences*, 2(41), pp. 421–436 (2020)
2. Glorennec, P.Y., Jouffe, L.: Fuzzy q-learning. In: *Proceedings of 6th International Fuzzy Systems Conference, IEEE Barcelona*, 2(4864), pp. 659–662 (1997)
3. Gul, F., Rahiman, W., Nazli-Alhady, S.S.: A comprehensive study for robot navigation techniques. *Cogent Engineering*, 6, pp. 1–25 (2019)
4. Jiang, L., Huang, H., Ding, Z.: Path planning for intelligent robots based on deep q-learning with experience replay and heuristic knowledge. *Journal of Automatica Sinica*, pp. 1–11 (2019)
5. Kathib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1), pp. 90–98 (1986)
6. Liu, J., Qi, W., Lu, X.: Multi-step reinforcement learning algorithm of mobile robot path planning based on virtual potential field. In: *Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators ICPCSEE'17: Data Science* (2017)
7. Lyu, H., Yin, Y.: COLREGS-Constrained real-time path planning for autonomous ships using modified artificial potential fields. *The Journal of Navigation*, 72(3), pp. 588–608 (2019)
8. Matoui, F., Boussaid, B., Metoui, B., Frej, G.B., Abdelkrim, M.N.: Path planning of a group of robots with potential field approach: decentralized architecture. *IFAC-PapersOnLine*, 50(1), pp. 11473–11478 (2017)
9. Park, J.W., Kwak, H.J., Kang, Y.C., Kim, D.W.: Advanced fuzzy potential field method for mobile robot obstacle avoidance. *Computational Intelligence and Neuroscience*, pp. 1–13 (2016)
10. Rostami, S.M.H., Sangiaiah, A.K., Wang, J., Liu, X.: Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *EURASIP Journal on Wireless Communications and Networking*, 70(1), pp. 1–19 (2019)

11. Shidujaman, M., Samani, H., Raayatpanah, M.A., Mi, H., Premachandra, C.: Towards deploying the wireless charging robots in smart environments. In: International Conference on System Science and Engineering (ICSSE'18), pp. 1–6 (2018)
12. Subbash, P., Chong, K.T.: Adaptive network fuzzy inference system based navigation controller for mobile robot. *Front Inform Technol Electron Eng.*, 20(2), pp. 141–151 (2019)
13. Tiwari, A.K., Guha, A., Pandey, A.: Dynamic motion planning for autonomous wheeled robot using minimum fuzzy rule based controller with avoidance of moving obstacles. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), pp. 4192–4198 (2019)
14. Tuazon, J.P.C., Prado, K.G.V., Cabial, N.J.A., Enriquez, R.L., Rivera, F.L.C., Serrano, K.K.D.: An improved collision avoidance scheme using artificial potential field with fuzzy logic. In: *Tencon'16, IEEE Region 10 Conference*, Singapore (2016)
15. Wei, P.A., Tsai, C.C., Tai, F.C.: Autonomous navigation of an indoor mecanum wheeled omnidirectional robot using segnet. In: *Proceedings of National Symposium on System Science and Engineering*, Taiwan (2019)
16. Zhang, J.: A hybrid reactive navigation strategy for a non-holonomic mobile robot in cluttered environments. In: *Proceedings of the 38th Chinese Control Conference (CCC)* (2019)

Atribución de autoría de mensajes de Twitter a través del análisis sintáctico automático

Francisco Antonio Castillo Velásquez, José Luis Martínez Godoy,
María del Consuelo Patricia Torres Falcón, Jonny Paul Zavala De Paz,
Adela Becerra Chávez, José Amilcar Rizzo Sierra

Universidad Politécnica de Querétaro,
México

```
{francisco.castillo, jose.martinez, jonny.zavala,  
consuelo.torres, adela.becerra}@upq.mx,  
jose.rizzo@upq.edu.mx
```

Resumen. Dentro del Procesamiento del Lenguaje Natural (PLN) se ha considerado que la tarea de análisis sintáctico representa un alto costo computacional, por lo que su aplicación a varias tareas se ha visto limitada. Una de estas es la atribución de autoría de textos (AAT), que se encarga de responder a la cuestión de quién es el autor de un texto, dando algunos ejemplos previos de ese autor (conjunto de entrenamiento). A pesar del costo computacional, los trabajos de clasificación en este campo han dado buenos resultados para textos largos (por ejemplo, libros), pero el estudio de textos cortos ha quedado rezagado. En este trabajo de investigación se propone un modelo computacional basado en n-gramas sintácticos de dependencias para la AAT en Twitter. Como estos mensajes tienen una longitud corta (máximo 280 caracteres) el desempeño del parser sintáctico no es materia de preocupación. La metodología utilizada consistió en la compilación de tweets en español (corpus), su procesamiento en tareas de etiquetación de partes de oración (PosTags) para formar un baseline, la aplicación del análisis sintáctico de dependencias al corpus, la generación de gramas de PosTags y de dependencias sintácticas, la generación de archivos con datos de entrenamiento y clasificación y la aplicación de métodos supervisados de aprendizaje automático a estos archivos. Aunque la mayoría de los resultados no son alentadores, hay un conjunto que permiten ver la factibilidad de la AAT en mensajes de textos cortos.

Palabras clave: Atribución, Autoría, N-Gramas, Dependencias Sintácticas.

Attribution of Authorship of Twitter Messages through Automatic Syntactic Analysis

Abstract. Within the Natural Language Processing (NLP), it has been considered that the syntactic analysis task represents a high computational cost, so its application to various tasks has been limited. One of these is the attribution of authorship of texts (AAT), which is responsible for answering the question of who is the author of a text, giving some previous examples of that

author (training set). Despite the computational cost, classification work in this field has given good results for long texts (e.g., books), but the study of short texts has lagged. In this research work, a computational model based on syntactic n-grams of dependencies for the AAT in Twitter is proposed. As these messages are short in length (maximum 280 characters) the performance of the syntactic parser is not a matter of concern. The methodology used consisted of the compilation of tweets in Spanish corpus, their processing in tasks of tagging parts of speech (PosTags) to form a baseline, the application of the syntactic analysis of dependencies to the corpus, the generation of n-grams of PosTags and of syntactic dependencies, the generation of files with training and classification data, and the application of supervised methods of machine learning to these files. Although most of the results are not encouraging, there is a set that allows seeing the feasibility of AAT in short text messages.

Keywords: Attribution, Authorship, N-grams, Syntactic Dependencies.

1. Introducción

La atribución de autoría hace frente a una pregunta antigua y difícil: cómo asignar un texto de una autoría desconocida o disputada a un miembro de un conjunto de autores candidatos de quienes se tienen ejemplos de textos sin disputa [9]. A pesar de su aplicación a trabajos literarios, la rápida expansión de texto en línea en Internet (blogs, mensajes de correo electrónico, posts en redes sociales, etc.), revelan que las aplicaciones prácticas de la atribución de autoría son asociadas, por lo general, con trabajos forenses [1].

Los enfoques automatizados a este problema involucran el uso de métodos de aprendizaje o estadísticos [13]. Desde el punto de vista del aprendizaje automático, la atribución de autoría puede verse como una tarea de clasificación multi-clase y de etiquetación [11]. Existen dos etapas básicas: primero, los textos pueden representarse apropiadamente como vectores de valores numéricos y, luego, un algoritmo de clasificación puede usar estos vectores para estimar la probabilidad de asociación de un texto a una clase.

Desde entonces se han propuesto cientos de características estilométricas. Estas pueden distinguirse en las siguientes cinco categorías de acuerdo al análisis textual que requieren [13]: características léxicas (frecuencia de palabras funcionales, frecuencia de n-gramas de palabras, medidas de la riqueza del vocabulario, etc.), características de carácter (frecuencia de letras y n-gramas de carácter), características sintácticas (frecuencia de etiquetas POS, mediciones en la estructura de oraciones y frases, frecuencia de reglas de reescritura, etc.), características semánticas (mediciones de sinónimos, mediciones de dependencias semánticas, etc.) y características específicas de la aplicación (tamaño de fuente, color de fuente, frecuencias de palabras específicas, etc.). Hasta ahora, algunos estudios han demostrado que las mediciones más efectivas son las de características léxicas y de carácter.

El presente trabajo de investigación contribuye a la verificación de la factibilidad de aplicación del análisis sintáctico automático en el proceso de AAT para textos cortos, para nuestra ejemplificación, con mensajes de la red social Twitter.

2. Trabajo relacionado

Existen pocas referencias para trabajos sobre AAT en Twitter. En [19], los autores trabajaron con Ruby para obtener 393 características, como longitud de palabras, frecuencia de caracteres, palabras funcionales y longitud del texto, para procesarlas en clasificadores implementados en Matlab, con resultados de clasificación correcta no más allá del 40%.

Boutwell [22] realizó pruebas con un clasificador Naïve Bayes para n-gramas de caracter. La autora experimentó con 50 autores y dos conjuntos de entrenamiento (120 y 230). Complementó su trabajo con una serie de pruebas para estudiar el efecto de unir varios tweets en un solo documento.

Otro trabajo relacionado se encuentra en [21], donde los autores trabajaron con un conjunto de características que incluyen n-gramas de caracteres y de palabras, aunque en los experimentos solo se usó un tweet como documento para las pruebas (test). También trabajaron con los conceptos de *k-signature* y patrones flexibles que, al integrarlo con los n-gramas, lograron una mejora en sus resultados (hasta un 70% de clasificación correcta).

En [23] los autores propusieron un conjunto de marcadores estilísticos para la AAT para mensajes de Twitter.

Estos marcadores incluyeron emoticones, interjecciones, caracteres de puntuación, abreviaciones y otras características de bajo nivel. Solo trabajaron para tres autores con SVM, alcanzando una clasificación correcta del 63%.

3. Metodología usada

En este apartado se explica el modelo propuesto de trabajo, iniciando con la compilación del corpus, el proceso de obtención de los n-gramas y finalizando con la tarea de clasificación.

Los experimentos consistieron en la generación de n-gramas (mediante una herramienta libre de extracción); la compilación de los n-gramas hace referencia al almacenamiento dinámico de los n-gramas únicos (haciendo uso de estructuras matriciales y la implementación de un algoritmo para su manipulación); la refinación, que es la puesta a punto de los n-gramas únicos para que los caracteres no reconocidos por Weka sean detectados y cambiados; la generación de archivos Weka; la aplicación de estadísticas de frecuencias y la aplicación de procesos de clasificación con los modelos de SVM (SMO), Naïve Bayes y J48 para distintos tamaños de perfiles (perfiles). Usamos un baseline tradicional con clasificación *NaïveBayes Multinomial Text* para el texto original como para etiquetas de *PoS*.

3.1. Compilación del corpus

La parte inicial del trabajo fue la compilación de un corpus de 600 tweets en español, intentando verificar la autenticidad de los mismos y seleccionando aquellos que no contuvieran caracteres de tipo emoticón. Este corpus está disponible para la comunidad investigadora.

Tabla 1. Rutas de dependencias para el grafo de la figura 2.

ao mod	cd v
ao cc conj	cd s
ao cc pass	cd cd spec
ao f	cd cd sp sn
cd conj	f
cd ci	

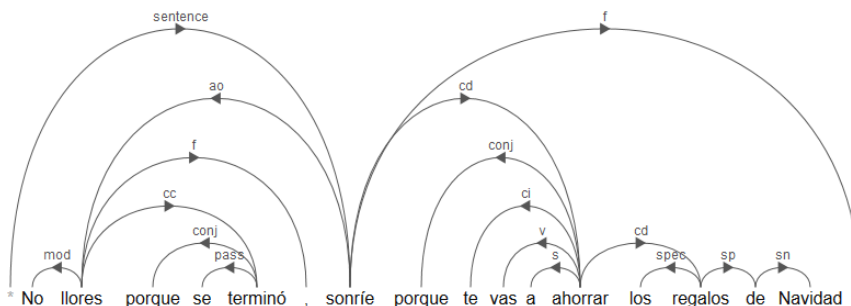


Fig. 1. Árbol de dependencias para una oración en español, generado por FreeLing.

1	No	no	RN	RN	pos=adverb type=negative	-	-	-	2	mod	-
2	llores	llorar	VMSP2S0	VMS	pos=verb type=main mood=subjunctive tense=present person=2 num=singular	-	-	-	7	ao	- llorar.
3	porque	porque	CS	CS	pos=conjunction type=subordinating	-	-	-	5	conj	-
4	se	se	P00CN00	PO	pos=pronoun gen=common num=invariable	-	-	-	5	pass	-
5	terminó	terminar	VMI3S00	VMI	pos=verb type=main mood=indicative tense=past person=3 num=singular	-	-	-	2	cc	- termina
6	,	,	Fc	Fc	pos=punctuation type=comma	-	-	-	2	f	-
7	sonríe	sonreír	VMIP3S0	VMI	pos=verb type=main mood=indicative tense=present person=3 num=singular	-	-	-	0	sentence	- sonreír
8	porque	porque	CS	CS	pos=conjunction type=subordinating	-	-	-	12	conj	-
9	te	te	FP2C300	FP	pos=pronoun type=personal person=2 gen=common num=singular	-	-	-	12	ci	-
10	vas	ir	VMIF2S0	VMI	pos=verb type=main mood=indicative tense=present person=2 num=singular	-	-	-	12	v	- ir.00
11	a	a	SP	SP	pos=adposition type=preposition	-	-	-	12	s	-
12	ahorrar	ahorrar	VMB0000	VM	pos=verb type=main mood=infinitive	-	-	-	7	cd	- ahorrar
13	los	el	DA0MFO	DA	pos=determiner type=article gen=masculine num=plural	-	-	-	14	spec	-
14	regalos	regalo	NCMF000	NC	pos=noun type=common gen=masculine num=plural	-	-	-	12	cd	-
15	de	de	SP	SP	pos=adposition type=preposition	-	-	-	14	sp	-
16	Navidad	navidad	NF00000	NF	pos=noun type=proper	-	-	-	15	sn	-
17	.	.	Fp	Fp	pos=punctuation type=period	-	-	-	7	f	-

Fig. 2. Análisis CONLL para la misma oración de la figura 2, generado por FreeLing.

A continuación, se muestran algunos ejemplos de tweets que forman parte del corpus. Se ha respetado la redacción original.

- Algunas mentiras son de carne y hueso.*
- un clásico cuando haces una pregunta y como no saben que contestar te empiezan a insultar.*
- No llores porque se terminó, sonríe porque te vas a ahorrar los regalos de Navidad.*
- Cómo cambian las cosas!!*
- Cuando algo se me mete en la cabeza...imposible parar hasta que lo consigo*
- El dinero no puede comprar la felicidad, pero la pobreza no puede comprar nada.*

3.2. Generación de n-gramas

Tomando como punto de referencia la segunda oración del ejemplo 1 (“Cómo cambian las cosas!!”) podemos generar los bigramas de caracteres *Có*, *óm*, *mo*, *o*, *_*, *c*, *ca*, *am*, *mb*, *bi*, *ia*, *an*, *n*, *_*, *l*, *la*, *as*, *s*, *_*, *c*, *co*, *os*, *sa*, *as*, *s*! y *!!*. También podemos generar los trigramas *Cóm*, *ómo*, *mo*, *_*, *o*, *c*, *_*, *ca*, *cam*, *amb*, *mbi*, *bia*, *ian*, *an*, *_*, *n*, *l*, *_*, *la*,

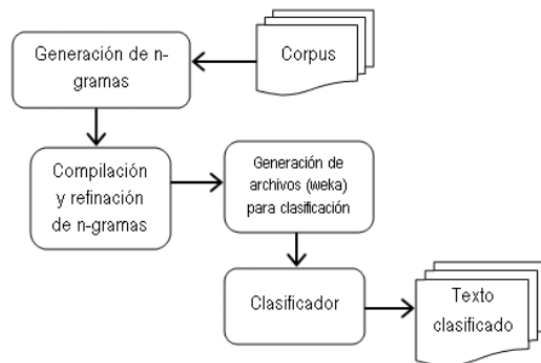


Fig. 3 Modelo de trabajo propuesto.

Tabla 2. Bigramas posibles para las rutas de dependencias de la tabla 1.

ao mod	ao f	cd cd
ao cc	cd conj	cd spec
cc conj	cd ci	cd cd
ao cc	cd v	cd sp
cc pass	cd s	sp sn

Tabla 3. Trigramas posibles para las rutas de dependencias de la tabla 1.

ao cc conj	cd cd sp
ao cc pass	cd sp sn
cd cd spec	

etc. La idea de trabajar con gramas es muy simple y tiene la ventaja adicional que puede aplicarse prácticamente para cualquier lenguaje.

Nuestro modelo hace un análisis estadístico de las apariciones de los gramas en cada una de las oraciones. Se pretende obtener un conjunto de características definitorias del estilo de un autor basado en este fundamento léxico de caracteres. Un análisis de este nivel (superficial) no necesita de un procesamiento profundo de las oraciones, como lo hace un análisis sintáctico (tanto de dependencias como de constituyentes).

El modelo del trabajo se resume en la figura 4, donde el paso inicial (generación del corpus) se dividió en dos conjuntos: los tweets originales y las dependencias generadas por el parser sintáctico FreeLing.

Para la misma frase de ejemplo, el parser de dependencias del FreeLing regresa el árbol de dependencias de la Figura 2 y el análisis en formato CONLL (Figura 3):

La información CONLL es transferida a un módulo programado en Java para obtener todas las rutas de dependencias posibles. Para el ejemplo se obtendrían las rutas mostradas en la Tabla 1.

Estas rutas son procesadas con text2ngram [17] para obtener los gramas de dependencias. Por ejemplo, los bigramas posibles para el conjunto de la Tabla 1 se muestran en la Tabla 2 y para los trigramas en la Tabla 3 el único cuatrigrama posible es cd cd sp sn.

```

@relation autores
@attribute sn_sp_sn numeric
@attribute sp_sn_spec numeric
@attribute cc_sn_spec numeric
@attribute cc_sn_sp numeric
@attribute cc_sn_S numeric
... ..
@attribute cd_S_S numeric
@attribute cd_S_creg numeric
@attribute autores {marioruiz,mzavalag,rauljimenez,coffeejaay,egolepsia,lusagania}
@data
0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
... ..
1,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
5,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

```

Fig. 4. Ejemplo de archivo arff generado para su posterior clasificación.

En la generación de estos gramas, text2ngram realiza un análisis de frecuencias, lo que agrega un valor numérico al final de cada grama. Esta información (gramas y frecuencias) se almacenó en un archivo de texto para después cargarse y procesarse en una hoja de cálculo, obteniendo una ordenación de mayor a menor frecuencia. Una vez ordenados cada conjunto de gramas por autor, se realiza una recopilación de todos los gramas de todos los autores y se aplica de nueva cuenta una ordenación por frecuencia. De este conjunto final ordenado, obtenemos los gramas que pasarán a ser atributos en los archivos .arff, que en nuestro caso seleccionamos los primeros, 15, 30, 60 y 90 más frecuentes.

Estas primeras tareas las podemos fácilmente identificar en el modelo propuesto de trabajo de la Figura 4.

La implementación de la generación de archivos Weka es un proceso semi-automático a través de un programa en Java, el cual está disponible para la comunidad académica e investigadora [18], junto con el corpus compilado y los archivos Weka generados.

3.3. Proceso de clasificación

Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Proporciona herramientas para el pre-procesamiento, clasificación, regresión, clustering y visualización de datos. Es un software de código abierto basado en los términos de GNU-GPL. En la Figura 5 se muestra un bosquejo de un archivo arff resultante, en el cual se identifican las tres secciones características de este tipo de archivos: nombre de la relación, atributos y datos.

Cada registro de la sección de datos (@data) representa un tweet y cada valor numérico representa las ocurrencias de un grama en particular en ese tweet. Por ejemplo, la primera terna de valores del último registro (5,1,0) significa que los gramas

Tabla 4. Baseline para 100 instancias por autor en texto original y partes de oración (PoSTags), con clasificación por Naïve Bayes Multinomial Text (clasificación con texto completo original).

# Aut	Autores	Texto original	PoSTags
3	CoffeeJaay egolepsia LuSagania	0.53	0.52
3	Soymarioruiz Mzavalagc Raul_Jimenez9	0.67	0.63
6	CoffeeJaay egolepsia LuSagania Soymarioruiz Mzavalagc Raul_Jimenez9	0.47	0.44

sn_sp_sn, *sp_sn_spec* y *cc_sn_spec* aparecen cinco, una y cero veces en el tweet, respectivamente. El último valor, que en nuestro ejemplo no es numérico, representa al autor del tweet.

El proceso de entrenamiento y clasificación también se llevó a cabo con el software Weka, que proporciona una diversidad de métodos. En particular, fueron usados los clasificadores NaiveBayes, Optimización Mínima Secuencial (SMO - Support Vector Machines) y árboles de decisión (J48), ya que estos han mostrado buenos resultados en otros trabajos de investigación, como en [12]. La configuración de la clasificación fue con una validación cruzada de 10 iteraciones y un porcentaje de división de 2/3.

4. Resultados experimentales

Los experimentos fueron desarrollados sobre los datos de un corpus para el problema de atribución de autoría. Con el corpus de 600 tweets se hicieron dos pruebas: una con 300 y otra con 600, o su equivalente, para 3 y 6 autores, ya que se recopilaban 100 tweets para cada autor. En la tabla 4 se muestra la información del baseline, con clasificación (basada en la medida de Cross Validation) del algoritmo Naïve Bayes Multinomial Text para el texto original de los tweets como para las partes de oración.

En los resultados que se mostrarán a continuación, usamos el término "*profile size*" para representar los primeros n-gramas/sn-gramas más frecuentes; por ejemplo, un tamaño del profile de 30 significa que se usaron solo los primeros 30 n-gramas más frecuentes. Probamos varios umbrales para el profile y seleccionamos 4 de ellos, como se muestra en todas las tablas de resultados.

Cuando alguna celda de la tabla contiene NA (*not available, no aplica*) significa que nuestros datos fueron insuficientes para obtener el número correspondiente de n-gramas. Sucede solo con los bigramas, ya que en general hay menos bigramas que trigramas, etc. En estos casos el número total de todos los bigramas es menor que el tamaño del profile.

Tabla 5. Resultados de la clasificación para el corpus de 3 autores, 60 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	3	4
15	NB	0.48	0.38	0.30
	SVM	0.53	0.52	0.35
	J48	0.40	0.45	0.28
30	NB	0.64	0.47	0.42
	SVM	0.70	0.50	0.40
	J48	0.53	0.55	0.30
60	NB	0.69	0.48	0.47
	SVM	0.64	0.38	0.42
	J48	0.50	0.38	0.32
90	NB	0.62	0.48	0.42
	SVM	0.60	0.38	0.40
	J48	0.52	0.38	0.33

Tabla 6. Resultados de la clasificación para el corpus de 3 autores, 150 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	3	4
15	NB	0.50	0.45	0.45
	SVM	0.54	0.50	0.41
	J48	0.43	0.53	0.43
30	NB	0.51	0.47	0.45
	SVM	0.60	0.48	0.42
	J48	0.44	0.43	0.45
60	NB	0.53	0.51	0.45
	SVM	0.59	0.47	0.48
	J48	0.45	0.45	0.45
90	NB	0.51	0.49	0.49
	SVM	0.55	0.47	0.47
	J48	0.46	0.49	0.45

La tarea de clasificación consiste en seleccionar características para construir el modelo de espacio de vectores, algoritmos supervisados de entrenamiento y clasificación – decidir a qué clase pertenece el texto –en nuestro modelo de espacio de vectores. En este trabajo presentamos resultados para tres clasificadores: SVM (SMO), Naïve Bayes y J48.

En los resultados mostrados en la tabla 5, es interesante notar que los que representan una mayor exactitud son dados por los clasificadores SVM y Naïve Bayes para bigramas, alcanzando un 70% el primero de ellos; de hecho, los mejores resultados

Tabla 7. Resultados de la clasificación para el corpus de 3 autores, 300 instancias.

Tamaño del profile	Clasificador	Tamaño del n-grama		
		2	3	4
15	NB	0.49	0.45	0.43
	SVM	0.52	0.43	0.38
	J48	0.49	0.40	0.39
30	NB	0.50	0.46	0.42
	SVM	0.57	0.45	0.41
	J48	0.54	0.43	0.38
60	NB	0.54	0.47	0.40
	SVM	0.57	0.45	0.43
	J48	0.51	0.40	0.38
90	NB	0.56	0.49	0.40
	SVM	0.59	0.45	0.44
	J48	0.51	0.41	0.38

Tabla 8. Resultados de la clasificación para el corpus de 6 autores, 600 instancias.

Tamaño del profile	Clasificador	Tamaño del n-grama		
		2	3	4
15	NB	0.32	0.28	0.25
	SVM	0.34	0.29	0.23
	J48	0.31	0.29	0.24
30	NB	0.36	0.31	0.27
	SVM	0.36	0.30	0.27
	J48	0.30	0.26	0.24
60	NB	0.36	0.31	0.26
	SVM	0.37	0.31	0.26
	J48	0.31	0.28	0.24
90	NB	0.37	0.31	0.26
	SVM	0.37	0.32	0.28
	J48	0.36	0.28	0.22

están para estos dos clasificadores en bigramas, salvo el caso de NB con un profile de 15.

En la tabla 6 se muestran los resultados de clasificación con 150 instancias, es decir, 50 tweets para cada uno de los 3 autores. Los mejores resultados los arroja el clasificador SVM para un tamaño de profile de 30 y 60 bigramas.

En la tabla 7 se muestran los resultados de clasificación para 3 autores, 100 tweets de cada uno de ellos, dando un total de 600 instancias. De nueva cuenta el clasificador

SVM obtiene los mejores resultados para cualquier tamaño de profile. Con las tablas 5-7, que representan los corpus de 3 autores, en general hay un mejor comportamiento con SVM, aunque las cifras no son tan alentadoras como se esperaban. La clasificación más alta está entre 57%-60%, para bigramas.

En la tabla 8 se muestran los resultados de clasificación para el corpus más grande compilado: 6 autores, 100 tweets para cada uno de ellos, dando un total de 600 instancias. Los resultados bajan drásticamente, no dando posibilidad de augurar mejores resultados con más autores y/o registros.

En este caso, las cifras del baseline (Tabla 4) resultan mejores, ya que alcanzaron un 47%, contra un 37%, que fue el mejor resultado para los gramas de dependencias sintácticas.

Se desarrolló una interfaz para la demostración del método, donde es posible configurar la elección de archivos a compilar, la frecuencia y tamaño de los n-gramas, entre otros parámetros.

Aunadas a estas aportaciones científicas, las de carácter técnico son la creación de una herramienta estadística para visualizar diversos resultados de los procesos involucrados y el corpus procesado para los experimentos (disponible para la comunidad científica [18]).

5 Conclusiones y trabajo futuro

Los resultados de los experimentos demuestran la factibilidad de usar modelos computacionales simples para la tarea de atribución de autoría de textos cortos, como correos electrónicos, tweets, mensajes de chats, etc., aunque solo para un conjunto reducido del corpus y n-gramas pequeños (la clasificación correcta más alta fue de 70% para solo 3 autores). La contribución más importante es la simplicidad y la generalidad del modelo (puede ser aplicados a cualquier lenguaje). Como trabajo futuro se pretende trabajar con la definición de nuevas características que, aunadas a los gramas de rutas de dependencias, mejoren substancialmente los resultados de clasificación; usar datasets estándares, como los que proporciona PAN-CLEF, para poder comparar nuestro método con otros que ya los han usado; y por último, probar otros métodos de clasificación, como Random Forest.

Referencias

1. Abbasi, A., Chen, H.: Sentiment analysis in multiple languages: feature selection for opinion classification in web forums. *ACM Trans. Information Systems*, 26(3), pp. 9–12 (2008)
2. Argamon, S., Levitan, S.: Measuring the usefulness of function words for authorship attribution. In: *Proc. of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing* (2005)
3. Agarwal, A.: Contextual phrase-level polarity analysis using lexical affect scoring and syntactic N-Gram. In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL)*, pp. 24–32 (2009)
4. Baayen, H.: Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, pp. 121–131 (1996)
5. Chaski, C.: Who wrote it? Steps towards a science of authorship identification. *National Institute of Justice Journal*, pp. 15–21 (1997)

6. De Marneffe, M.: Generating typed dependency parses from phrase structure parses. In: Proc. of LREC (2006)
7. Gamon, M.: Linguistic correlates of style: Authorship classification with deep linguistic analysis features. Proceedings of COLING 2004, pp. 611–617 (2004)
8. Hollingsworth, C.: Syntactic Stylometric: Using Sentence Structure for Authorship Attribution. Ms. Thesis, University of Georgia (2012)
9. Juola, P.: Questioned electronic documents: empirical studies in authorship attribution. In: Research Advances in Digital Forensics II Heidelberg: Springer. pp. 5–8 (2006)
10. López, A.: Atribución de autoría utilizando distintos tipos de características a través de una nueva representación. Tesis de maestría. INAOE, pp. 11 (2012)
11. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), pp. 1–47 (2002)
12. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic dependency-based N-grams as classification features. LNAI 7630, pp. 1–11 (2012)
13. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, pp. 538–566 (2009)
14. Stefanova, M.: El potencial discriminatorio de las secuencias de categorías gramaticales en la atribución forense de autoría de textos en español. Tesis doctoral, pp. 41–63 (2009)
15. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic N-grams as machine learning features for natural language processing. Expert Systems with Applications, 41(3), pp. 853–860 (2013)
16. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional: n-gramas sintácticos, filtrados y generalizados. 166 p. (2013)
17. N-Gram Extraction Tool: <https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/homepages.inf.ed.ac.uk/s0450736/ngram.html> (2020)
18. Carpeta compartida de recursos del presente artículo: https://drive.google.com/open?id=0BwJ_YuKc8LgKWFgzc1JHYlhVGs (2020)
19. Castro, A.: Author identification on twitter. In: Hardesty, Third IEEE International Conference on Data Mining, pp. 705–708 (2003)
20. Green, R.M., Sheppard, J.W.: Comparing frequency- and style-based features for twitter author identification. In: C. Boonthum-Denecke & G. M. Youngblood (eds.), Flairs Conference AAAI Press (2013)
21. Schwartz, R., Tsur, O., Rappoport, A., Koppel, M.: Authorship attribution of micro-messages. In: EMNLP Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 1880–1891 (2013)
22. Boutwell, S.: Authorship attribution of short messages using multimodal features. Master's thesis, Naval Postgraduate School (2011)
23. Silva, R., Laboreiro, G., TimGrant, L., Oliveira, E., Maia, B.: Twazn me!!! Automatic authorship analysis of micro-blogging messages. In: Proc. of the 16th International Conference on Natural Language Processing and Information Systems, NLDB'11, pp. 161–168, Berlin, Heidelberg (2011)

A Data Fusion Approach of Physical Variables Measured through a Wireless Sensor Network

Thomas Alejandro Arias Pelayo, Guillermo Molero Castillo,
Everardo Bárcenas, Rocío Aldeco Pérez

Universidad Nacional Autónoma de México,
Facultad de Ingeniería,
Mexico

thomas.arias95@gmail.com,
{gmolero, raldeco}@fib.unam.mx,
ebarcenas@unam.mx

Abstract. At present, current data fusion methods are a useful tool for integrating data sources, prior to data analytics, and provide a unified view of an observed phenomenon or event. This paper presents the development of the Monte Carlo method, as a data fusion mechanism, obtained from a wireless sensor network. This network of sensors was designed and installed in a closed environment of human occupation. The data collected was of physical variables, such as temperature, humidity, and dust density, which were stored in the cloud through ThingSpeak, which is an open-source platform for the Internet of Things. As a result, it was succeeded in data fused properly and the method was evaluated through the root-mean-square error. Undoubtedly, fused values can be useful, for example, for the analysis of the thermal comfort of users in closed environments, where there are minimal ventilation rates and adequate indoor air quality is needed.

Keywords: Data Fusion, Monte Carlo Method, Wireless Sensor Network, Physical Variables.

1 Introduction

Today, the automation of industrial processes, the improvement of workforce capabilities and the development of new products through Artificial Intelligence benefits society as part of the digital transformation process in this era of the fourth industrial Revolution. However, of all the technologies to consider for their potential, without a doubt, wireless sensor networks (WSN) are proving their usefulness for measuring and storing data on a specific environment, as a key technology to implement the Internet of Things (IoT), whose main characteristic is its low energy consumption and its deployment in inaccessible locations or even integrated within structures.

It is important to note that currently, wireless sensor networks are autonomous devices that work collaboratively, distributed throughout an area of interest and

whose objective is to monitor physical or environmental parameters, such as temperature, sound, vibrations, pressure, movement, or pollutants. Each element of the sensor network communicates wirelessly, offering a flexible system, easy to install, and scalable in large quantities. However, as the volume of data grows, so does the need to combine data from sensors to extract useful information, allowing an understanding of the environment and making timely decisions.

Faced with this situation, data fusion systems are an efficient solution to this problem. Data fusion is the process of associating and combining data at various levels, which come from different sources, such [1]: sensors, databases, signals, decisions, and others. This field of knowledge is used in different areas, such as [2]: signal processing, statistics, inference, and artificial intelligence. It has greater advances in applications of automatic target recognition, autonomous vehicle navigation, remote sensing, virtual environments, smart homes, threat identification, among others.

The idea of combining data from various sources, such as a network of sensors, is to obtain a global and unified vision of the observed phenomenon. Therefore, data fusion provides a formalization to data integration, in such a way that the results help in the decision-making process about a certain event, object, or action [3].

This paper aims to provide a vision of the progress of the development of a method for data fusion of physical variables, obtained in a closed space of human occupation, such as a room of a house, office and other places of rest or work, through a wireless sensor network. In particular, the analysis of the three variables was carried out, such as temperature, humidity, and dust density.

The document is organized in the following way, Section 2 presents the background of data fusion, some of the most significant contributions of data fusion are mentioned, its applications are discussed, the use of wireless sensor networks and the related jobs. Section 3 describes the method established as a proposed solution. Section 4 presents the results obtained, based on an application example, and Section 5 summarizes some conclusions and future work.

2 Background

As mentioned, data fusion is about the integration of data that comes from different sources. The purpose is to have a unified and refined view of them, in order to analyze them and give a conclusion. The data fusion as a field of knowledge emerged with military and robotics applications [10].

At present, one of the areas that have given the greatest impetus to data fusion is data science, where methods are used to extract, transform and load data sources as part of the data engineering process, prior to data analytics [11, 12].

Another area, which has also benefited from data fusion, is IoT since this is a technology that depends on interconnected objects, capable of communicating for the collection of data about a certain event [13].

2.1 Wireless Sensor Network

A wireless sensor network is made up of a set of interconnected sensor nodes that cooperate to measure and control a certain physical context that surrounds them, thus allowing interaction between people and devices or machines. These sensors work within a field of action to capture and transform signals into data, which are sent through other nodes to a common output, such as the Internet or databases [14].

A sensor network uses wireless or wired communication, and can be physically organized within an area where events occur in different ways, such as [7]: a) distributed, where there is no central node, but if a node stops function, it does not affect any other; b) centralized, where all nodes are peripheral except the central node, which is the only one that emits information; c) decentralized, where there is a collective hub of nodes, if one node goes down, several nodes may stop working, but the entire network does not go down.

In recent years, along with the growth of the Internet of Things, wireless sensor networks are gaining relevance in the industry, academia, and society in general, as a global infrastructure that allows offering advanced services through the interconnection of physical and virtual objects, thanks to the interoperability of information technologies [15], which, in turn, guarantee security and privacy requirements [17]. Thus, the applications are varied, among which agriculture, smart cities, smart homes, smart environments, industry, logistics, environment, military solutions, security, health, among others, stand out.

2.2 Data Fusion

Data fusion is a field of knowledge used to combine data produced by one or more sensors or data sources so that you can have a better estimate of the amount being measured [16]. Current ideas of data fusion are an efficient solution alternative to the increasing availability of data volumes. This is to combine data from various sources to obtain a complete description of an observed phenomenon [7].

On the other side, various definitions have emerged in data fusion. One of these points out that it is a multi-level, multi-stage process for the detection, association, correlation, estimation, and combination of data from one or more sources to achieve complete estimates and evaluations [18]. Recent examples of data fusion cover a wide range of areas, such as engineering, medicine, traffic control, environment, artificial intelligence, robotics, among others.

There are, therefore, application areas that until recently have been little explored by the data fusion community, some of these are [20]: the development of systems capable of incorporating hard and soft data, systems of data fusion in dynamic environments and real-time processing, as well as the incorporation of data fusion in smart sensors.

2.3 Monte Carlo Method

The Monte Carlo method is a non-deterministic technique, used to approximate numerical estimates to a great variety of mathematical problems. This method is

named in reference to the Monte Carlo Casino (Monaco), for being the capital of the game of chance, as roulette is a simple generator of random numbers [19]. This method bases its operation on simulation, which consists of repeating the characteristics and behaviors of a real system. The objective is to imitate the behavior of real variables to, as far as possible, analyze or predict simple or complex future values.

Currently, this method is a fundamental part of the Ray Tracing algorithms for the generation of 3D images [21]. The method may vary depending on the application, but in general, it follows the following steps: i) it defines the domain of the inputs; ii) it entries with random values are generated, following a probability distribution that is within the domain; iii) the inputs in the function are evaluated, and iv) the calculated values are gathered.

Being a probabilistic analysis algorithm, the possibility of error is given by the number of errors divided by the number of times the test is performed. Following the central limit theorem, Monte Carlo has an absolute error given by:

$$\frac{1}{\sqrt{N}}.$$

This indicates that if the number of trials is increased, the probability of the error tends to zero [19]. An advantage of the Monte Carlo simulation is that it can give a good approximation of the exact value. In this way, this method can be used to measure reliability in a sensor network [8]. The equation used is the following:

$$\langle b - a \rangle \frac{1}{N} \sum_{n=1}^{\infty} f(x_i),$$

where:

- $\langle b - a \rangle$ are the intervals of the function,
- N is the number of tests,
- x_i is the random value that is evaluated in the function.

2.4 Related Works

A growing problem in data fusion is dealing with data inconsistency. For this, there are various methods that can be used, such as sending the same data packet a second time. Once sent, the similarities of the two packets are compared and outliers are removed. Another option is to compare previous data measurements and then give a certain level of confidence to the sensors that were used [4]. Another method is to assign a series of weights hierarchically to the sensors [5], but none of these methods deal with the conflict, from the point of view of inconsistency, generated by using different sensors. Table 1 summarizes some identified works, as part of the literature review.

Therefore, due to the promising future of wireless sensor networks, there is a natural interest to embrace data fusion applications with this type of technology to monitor different phenomena and events efficiently.

Table 1. Related Works.

Author	Description	Method used	Limitations
Motro <i>et al.</i> (2006) [6]	It is a data source integration system, called Fusionplex.	The process consists of finding inconsistencies in data tuples, that is, inconsistencies are detected in the instances and eliminated according to the criteria.	High computational cost to resolve conflicts in commercial applications.
Kumar <i>et al.</i> (2009) [4]	Detects data inconsistency in sensors before it is sent.	Uses a neural network to detect the inconsistency of data through its variance.	It does not analyze the inconsistency of the data.
Frikha <i>et al.</i> (2015) [5]	Assigns a hierarchy to sensors to eliminate subjectivity and reduce inconsistency. The higher the hierarchy, the greater the weight.	The sensors were compared in pairs using matrices. The weights of the sensors are added.	It is required to define the criteria to assign the initial hierarchies.
Bakr <i>et al.</i> (2017) [7]	Three data fusion approaches are presented to deal with inconsistency: a) model-based, b) redundant, and c) fusion-based.	The treatment of inconsistency of data from various sources is described.	Lacks a process of fusing data from multiple sources.

Table 2. Characteristics of the NodeMcu board.

	Characteristic	Value
1	Power supply	5 V
2	Output voltage on pins	3.3 V
3	Processing speed	80 - 160 MHz
4	Size	29.1 x 63.3 mm

3 Method

The working method defined for data fusion of physical variables, measured in a closed environment of human occupation, was divided into four stages: a) installation of the sensor network, b) data collection, c) data fusion, and d) evaluation of the fusion method.

a) Installation of the sensor network

In this stage, two sensor nodes were installed at the ends of a room. The NodeMcu board (ESP8266) was programmed to send data to the ThingSpeak platform, which allows storing, visualizing, and analyzing data in real-time, using cloud services for IoT applications. Using the cloud has advantages, since it allows access to the data collected from anywhere in the world, using any device with an Internet connection. In addition, it provides useful tools for users. Table 2 summarizes the main characteristics of the NodeMcu board, the required values of voltage, speed, and size. This board is used for the development of mobile IoT applications.

Table 3. Characteristics of the DHT11 sensor.

	Characteristic	Value
1	Power supply	3.3 – 5 V
2	Output voltage on pins	3.3 V
3	Humidity range	20 – 90% with 5% precision
4	Temperature range	0 – 50°C with ± 2°C of precision
5	Size	14 x 18 x 5.5 mm

Table 4. Characteristics of the dust sensor.

	Characteristic	Value
1	Power supply	5 V
2	Current consumption	Icc: MAX 20 mA
2	Pulse cycle	10 ± 1ms
3	Size	46.0 x 30.0 x 17.6 mm

Table 5. Extract of data collected through the implemented sensor network.

Date	Temperature (°C)	Humidity (%)	Dust Density (mg/m ³)
2020-05-30 19:54:38	28.9	67	0.02
2020-05-30 19:59:38	29.1	69	0.01
2020-05-30 20:04:39	29.0	71	0.02
...
2020-06-23 09:37:12	32.1	64	0.01
2020-06-23 09:42:13	32.2	65	0.01
2020-06-23 00:47:13	32.2	66	0.02

The DHT11 sensor includes a humidity and temperature measurement component that connects to an 8-bit microcontroller to ensure quality and fast response using low voltage. Table 3 summarizes the main characteristics of the DHT11 sensor, it indicates the power values required to make it work, its size, and the range of humidity and temperature values that it can measure.

On the other side, the GP2Y1010AU0F sensor was used to detect reflected light from dust and smoke particles. This type of sensor can distinguish between dust and smoke particles by means of a pulse pattern in the output voltage. Table 4 presents some of its most important characteristics, such as the pulse cycle it uses to detect dust particles, size and feed.

b) Data collection

To send data to the ThingSpeak channels and maintain the integrity of the data, the platform has a function called “thingSpeakRead” that imports the non-null values. The data that comes from the two nodes were grouped. To carry out this task, two channels were created, where the data was stored and public access was granted for it to be read by any user [22, 23]. The data was collected in a period of five weeks, from

May to June 2020. Table 5 shows, as an example, an extract of the captured data.

c) Data Fusion

As part of the data fusion process, the Monte Carlo method was used to verify the variation in the data that was obtained through the sensor nodes. This method was chosen for its ability to numerically operate complex systems, maintaining the input and output relationship, with a certain degree of uncertainty. Subsequently, the data were fused for each of the collected variables, that is, temperature, humidity, and dust density.

The Monte Carlo method was chosen since it allows us to identify what is going to happen and its probability of occurrence. In addition, it is important to point out that through this method it was possible to identify possible conflicts generated in the data, since if this situation is ignored, the resulting data could represent, in an erroneous way, the real data obtained in the experimentation.

d) Evaluation of the fusion method

The Root Mean Square Error (RMSE) was used to evaluate the fusion method. Through this measurement, the difference between the estimated (predicted) values by the Monte Carlo method and the values captured by the sensors is calculated. The formula used was the following:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^n \left(\frac{d_i - f_i}{\sigma_i} \right)^2},$$

4 Results

The results of the data fusion for each of the variables analyzed are shown below.

a) Temperature

The probability distributions of the values obtained for temperature were plotted, as well as the data fusion. Fig. 1 shows the measurements of sensors 1 (blue) and 2 (green), which are close to the 50th percentile of each. While the values of the data fusion (red) are between the measurements of the two sensors, with a slight similarity to the values obtained from sensor 2.

b) Humidity

The probability distributions of the humidity percentages were plotted, as well as the data fusion (Fig. 2). The most probable values for sensor 1 (blue) and sensor 2 (green) were found to be within the values of the data fusion (red). Sensor 1 (blue) measurements have average behavior relative to sensor 2 and data fusion.

c) Dust Density

Fig. 3 shows the probability distribution of the dust density percentages measured by the two sensors, as well as the values of the data fusion. Sensor 2 (green) values were found to be unusual, as they are skewed to the right.

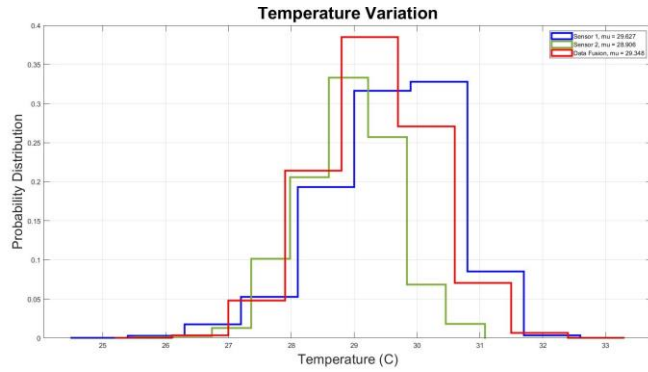


Fig. 1. Probability distribution of temperature data fusion.

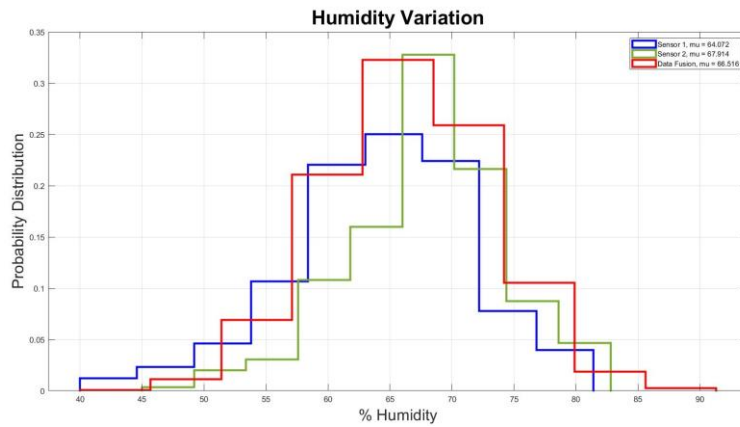


Fig. 2. Probability distribution of humidity data fusion.

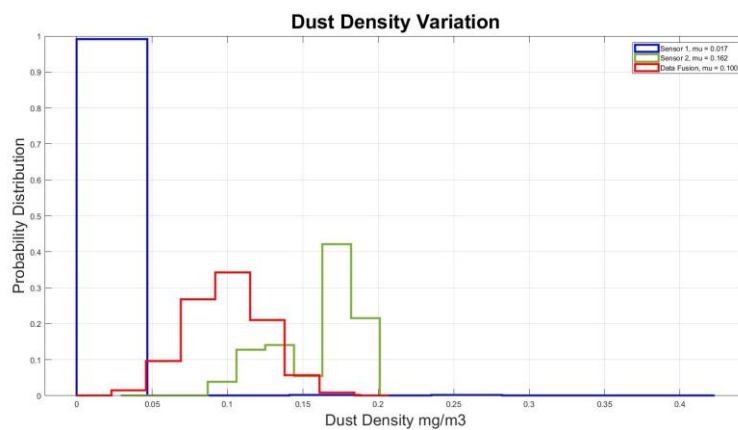


Fig. 3. Probability distribution of dust density data fusion.

Table 6. Mean square error estimation.

Variable	Sensor	RMSE
Temperature	Sensor 1	1.3892
	Sensor 2	1.2586
Humidity	Sensor 1	10.1487
	Sensor 2	9.2745
Dust density	Sensor 1	0.0895
	Sensor 2	0.0735

This may be due to sensor 2 being located above a bookcase and that the amount of dust that was raised was greater due to the cleaning carried out in this part of the room. In addition, it was observed that sensor 1 (blue) presented less variation, with a probability of density less than 0.05, that is, almost 99%. Although most of the data obtained in sensor 1 varied from 0.00 to 0.05, there were also some cases of variability between 0.07 to 0.4. This difference, with respect to the other sensor, is due to the fact that in that area of the room there is almost no air current and no cleaning was done.

On the other side, Table 6 shows the results obtained from the mean square error measurements with respect to the data fusion. Low levels of error were reached in temperature (1.38 and 1.25%) and dust density (0.08 and 0.07%); while, for humidity higher percentages of error were observed, that is, 9.27 and 10.14%. This may be because some days it rains and others don't. When it rains, the humidity increases, that is, there is high atmospheric saturation, which can give unusual values. Consequently, the precision level for temperature was 98.7%, humidity 90.3%, and dust density 99%. Given that 99% of the density levels of the powder were between 0.00 and 0.05, the RMSE value for the two sensors that measured this variable was low, that is, almost 1% error.

It is important to mention that the accuracy of the Monte Carlo method increases as more simulations are performed. Therefore, the computational cost also increases.

5 Conclusion

A network of sensors was implemented in a rest and work area, given the current context of the COVID-19 disease, such as a room in a home, in order to obtain data on temperature, humidity and dust density. For this, a NodeMcu development card (ESP 8266) with Wi-Fi connection capabilities was used in order to connect to the ThingSpeak platform to collect, visualize and analyze data from the sensors.

DHT11 sensors were used to measure temperature and humidity, and GP2Y1010AUOF sensors were used to measure dust density. The data obtained by the sensors can be consulted in real time through the two channels created in ThingSpeak. Sensor data can also be downloaded through the "Extract Data" function on the same channel

The values obtained for temperature, humidity, and density of the powder were correctly fused using the Monte Carlo method. The values of the sensors and the

values obtained through the method used were compared. In addition, low error levels were reached, which were evaluated by means of the root-mean-square error, which allowed to verify the accuracy of the obtained values. As future work, it will be sought to analyze based on the fused values of temperature, humidity, and dust density, if the observation environment, in this case, a room, is within the thermal comfort zone defined according to the ANSI/ASHRAE 55 standard. This type of analysis is useful to identify minimum ventilation rates and other measures to provide adequate indoor air quality in human-occupied spaces.

It will also consider expanding the implementation with different techniques, such as Artificial Neural Networks, Kalman Filter, among others, with the purpose of comparing the results and obtaining better and better-fused data.

Acknowledgment. This work was supported by UNAM-PAPIIT IA105320.

References

1. Kale, D., Aparadh, S.: a study of a detection and elimination of data inconsistency in data integration. *International Journal of Scientific Research in Science, Engineering and Technology*, 2(1), pp. 532–535 (2016)
2. Shahbazian, E., Bergeron, P., Duquet, J., Jouan, A., Valin, P.: Data fusion applications for military and civilian purposes developed DND/LM Canada decision support testbed. In: *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, 1, pp. 420–424 (1999)
3. Durrant-Whyte, H., Henderson, T.: Multisensor data fusion. In: Siciliano B., Khatib O. (eds) *Springer Handbook of Robotics*. Springer, Berlin, Heidelberg (2008)
4. Kumar, M., Garg, D.: Multi-sensor data fusion in presence of uncertainty and inconsistency in data. In: Milisavljevic (Ed.), *Sensor and data fusion*, pp. 225–244, Intech (2009)
5. Frikha, A., Moalla, H.: Analytic hierarchy process for multi-sensor data fusion based on belief function theory. *European Journal of Operational Research*, 241(1), pp. 133–147 (2015)
6. Motro, A., Anokhin, P.: Fusionplex: resolution of data inconsistencies in the integration of heterogeneous information sources. *Information Fusion*, 7(2), pp. 176–196 (2006)
7. Bakr, M., Lee, S.: Distributed multisensor data fusion under unknown correlation and data inconsistency. *Sensors*, 17(11), pp. 1–31 (2017)
8. Yang, Q., Chen, Y.: Monte carlo methods for reliability evaluation of linear sensor systems. *IEEE Transactions on Reliability*, 60(1), pp. 305–314 (2011)
9. Schiavon, S., Hoyt, T., Piccioli, A.: Web application for thermal comfort visualization and calculating according to ASHRAE Standard 55. *Building Simulation*, 7(4), pp. 321–334 (2014)
10. Muñoz, J.: Estimación Bayesiana para el tratamiento de la inconsistencia en la fusión de flujos de datos de fuentes físicas. *Doctoral Thesis, Universidad Veracruzana* (2020)
11. Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J., Ramakrishnan, R., Shahabi, C.: Big data and its technical challenges. *Communications of the ACM*, 57(7), pp. 86–94 (2014)
12. Zhu, X., Wu, G., Ding, W.: Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 26(1), pp. 97–107 (2014)
13. Cecchinell, C., Jimenez, M., Mosser, S., Riveill, M.: An architecture to support the collection of big data in the internet of things. In: *IEEE World Congress on Services*, pp. 442–449 (2014)

14. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. *Computer Networks*, 38(4), 393–422 (2002)
15. Ramos, A., Varum, T., Matos, J.: Compact Multilayer Yagi-Uda Based Antenna for IoT/5G Sensors. *Sensors*, 18(9), 2914 (2018)
16. Wah-Ng, G., Tan, C., Poh-Ng, T., Siow, S.: Assessment of Data Fusion Systems. In: 9th International Conference on Information Fusion, 1–8 (2006)
17. Contreras-Masse, R., Ochoa-Zezzatti, A., García, V., Elizondo, M.: Selection of IoT Platform with Multi-Criteria Analysis: Defining Criteria and Experts to Interview. *Research in Computer Science*, 148(11), 9–19 (2019)
18. Blasch, E., Steinberg, A., Das, S., Llinas, J., Chong, C., Kessler, O., Waltz, E., White, F.: Revisiting the JDL model for information exploitation. In: Proceedings of the 16th International Conference on Information Fusion, IEEE, 129–136 (2013)
19. Illana, J.: Metodos Monte Carlo. Departamento de Física Teórica y del Cosmos, Universidad de Granada, Spain, 1–47 (2013)
20. Barnaghi, P., Tonjes, R., Holler, J., Hauswirth, M., Sheth, A., Anantharam, P.: Real Time IoT Stream Processing and Large-scale Data Analytics for Smart City Applications. In: European Conference on Networks and Communications, 1–36 (2014).
21. Kajiya, J.: The Rendering Equation. *ACM SIGGRAPH Computer Graphics*, 20(4), 143–150 (1986)
22. Channel for sensor 1, https://thingspeak.com/channels/1010164/private_show. Last accessed in August 10, 2020
23. Channel for sensor 2, https://thingspeak.com/channels/1010156/private_show. Last accessed in August 10, 2020

A Physical Variable Data Fusion Approach as Basis for the Reasoning Process in Ambient Intelligence

Julio Muñoz Benítez, Guillermo Molero Castillo, Everardo Bárcenas,
Rocío Aldeco Pérez, Alejandro Velázquez Mena

Universidad Nacional Autónoma de México,
Facultad de Ingeniería,
Mexico

juliomunoz@uv.mx, gmoleroca@fib.unam.mx,
ebarcenas@unam.mx, raldeco@fib.unam.mx,
mena@fib.unam.mx

Abstract. Recently, interest has increased in evaluating environmental quality, for extended periods, in closed spaces of human occupation. This is due to the search for well-being in the health of the users. Therefore, it is important to analyze the behavior of environmental conditions, acceptable to people, in certain periods and workspaces, in accordance with the standards established by the Health sector. This work describes the implementation of a network of sensors to acquire data on certain variables, such as temperature, humidity, and air pollution. In this sense, data fusion provides a way to unify data as support for the analysis of various sources. In addition, Bayesian estimation is presented as a data fusion method of the captured values. The results showed that the data was fused with high precision, which can be used to analyze healthy atmospheric levels for users in closed environments, as a basis for the reasoning process in Ambient Intelligence.

Keywords: Ambient Intelligence, Bayesian Estimation, Data Fusion, Physical Variables, Sensor Network.

1 Introduction

Ambient Intelligence (AmI) is an emerging area in computer science that provides intelligence to real-life environments, where daily activities are performed, making them sensitive to the user and their actions by the use of electronic devices, sensor networks, context awareness, and artificial intelligence [4, 7] At present, the interest in this area has increased due to advances in sensor networks and their integration into daily activities, as well as the growing need to perceive and analyze the environment in order to ensure healthy levels in areas of human occupation environments [11, 10].

Because AmI is designed for real-world environments, the effective use of wireless sensor networks has been important in a variety of applications, such as environmental monitoring. The objective is to obtain useful information for making decisions about a certain event, object, or action [3], where the information is used to identify patterns of

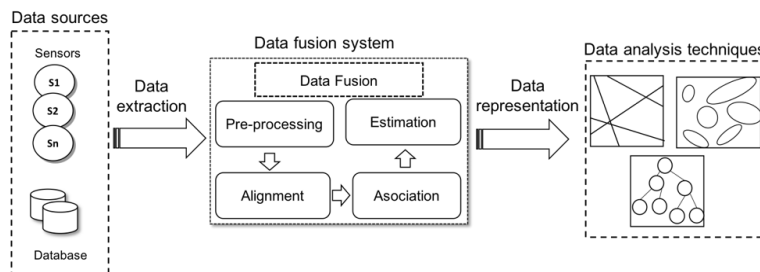


Fig. 1. Conceptual representation of a fusion system.

behavior over time, for example, to analyze concentrations of air pollution, temperature or humidity [14]. For this, it is necessary to analyze different data sources [19], such as those obtained through sensors.

Nevertheless, analyzing data collected through sensor networks is a complex challenge [4, 14, 19], as large volumes of data can be generated, making it difficult to analyze the captured information. Also, if the sensors are faulty or inaccurate, caused by ambient noise, missing or incorrect values can occur, leading to erroneous analysis or incorrect predictions [13]. Therefore, when using data from multiple sources, there must be a way to combine them, prior to analysis, since integrated data provide a comprehensive, unified, consistent, and accurate picture of a situation of interest.

In this sense, data fusion facilitates the detection, association, and integration of data [26], which come from different sources, such as signals from a sensor network [9]. This field of knowledge has been used in various areas, such as [9, 26]: signal processing, estimation, statistics, inference, and artificial intelligence. Today, it has significant advancements in applications for automatic target recognition, autonomous vehicle navigation, remote sensing, and threat identification.

As an example, Fig. 1 shows the diversity of sources, in this case, sensors, and the fusion process, where the data is integrated with the purpose of carrying out an analysis on a certain event.

One of the methods used in data fusion is Bayesian estimation [8, 15, 16, 17]. This method performs the fusion of the probability distribution of subsequent measurement, based on a set of probability distributions from previous measurements [15, 24]. That is, this is a way to represent the dynamism of the environment and the behavior of data sources [13, 15]. Thus, after data fusion, as part of AmI, the analysis of these fused data sources can be useful, for example for the analysis of environmental variables, according to established Health standards, such as the ANSI/ASHRAE 55-2017 (ASHRAE, American Society of Heating, Refrigeration and Air-conditioning Engineers), which defines the ambient thermal conditions for human occupation in a given workspace [1].

This paper presents the results of Bayesian estimation for the fusion of data from physical variables measured in a work area of a university environment. For the collection of data, humidity, temperature, and air pollution sensors were installed. This data source represents an important field of action as a basis for the generation of knowledge in AmI, specifically for the analysis of the thermal comfort of users in closed environments.

2 Background

At present, the areas that have increased the use of data fusion are data science, pattern recognition, and artificial and environmental intelligence [2, 5, 7], where fusion methods are used in the data extraction, integration, and loading in order to serve as a basis for their analytics.

Precisely, in AmI there is a growing need to perceive and analyze environmental conditions in order to improve the environment of users in the performance of their activities. One of the main characteristics of AmI is the use of electronic devices and communication techniques to perceive the environment, that is, to use sensor networks to monitor environmental conditions. One of these conditions of interest is the monitoring of humidity, temperature, and air pollution [20], which are atmospheric conditions important for the health and well-being of users.

For example, in the case of air pollution, it increases people's risk of contracting a respiratory disease [23, 27], given the presence of harmful particles, which not only depend on the gases generated by vehicles or gases emitted by industrial systems, also depends on environmental conditions such as humidity, temperature, rain, wind, to name a few. Therefore, these environmental conditions must be taken into account for their analysis and subsequent execution of an action plan for the benefit of people and the environment.

In this sense, data fusion is a key and critical aspect of systems with diverse data sources, such as sensors. The goal is to integrate data efficiently to overcome the limitations encountered when using a single source [10]. The fusion of data from various sources represents a significant advantage in the unification of data, to obtain a global, robust, and complete vision of the event or situation that is being observed and analyzed [25].

Data fusion has been used in various disciplines, such as pattern recognition, artificial neural networks, the decision-making process, to name a few. Currently, one of the areas that have driven the use of data fusion is AmI, where fusion methods are used in the extraction, integration, and loading of data to serve as support in the reasoning process from data collected from different sources [19].

On the other side, the analysis of thermal conditions is of importance since the World Health Organization (WHO) defines a set of diseases caused by environmental factors in closed spaces, called sick building syndrome [18, 21]. This syndrome is a set of annoyances caused by poor ventilation that prevents air renewal, temperatures not suitable for user activities, humidity levels that benefit the reproduction of micro-organisms, and the concentration of particles in the air. The symptoms of sick building syndrome affect users, generating discomforts such as [18]: headaches, nausea, dizziness, colds, irritation of the respiratory tract, eyes, skin, as well as persistent allergies.

Thus, thermal analysis is one of the fundamental aspects that determine the indoor environmental quality and is directly related to user satisfaction and energy use in buildings [22]. The purpose is to maintain healthy and adequate environmental levels that guarantee the comfort, health, and productivity of the users in the performance of their activities.

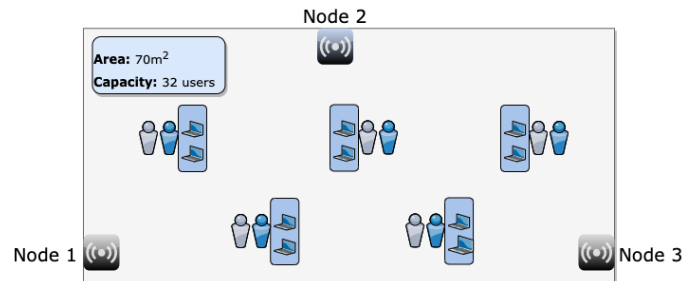


Fig. 2. Sensor network deployment in the workspace.

3 Method

As method, three work stages were defined, the first was the design and implementation of the sensor network for the measurement of environmental variables. In a second stage, the environmental variables were measured with a frequency of one minute, during a period of four weeks, that is, during the last weeks of spring. This period was defined as a period prior to summer, with the objective of obtaining data during the transition days of both seasons (spring and summer), that could be used to compare with a future analysis during winter. The third stage consisted of the use of Bayesian estimation as a data fusion method. This method was based on obtaining estimates based on previously collected data for temperature, humidity, and air pollution.

3.1 Sensor Network

The sensor network was implemented, providing it with sensing, processing, storage, and data transmission capabilities, in order to generate the data stream. Figure 2 shows the deployment of the sensor network in the physical space, where they were installed. Five nodes were installed, three to measure temperature, humidity, and air pollution (nodes 1, 2, and 3).

Each of these nodes was integrated into a NodeMCU board, which consists of an ESP8266 processor with integrated WiFi technology b/g/n [6]. Sensors were connected to the general-purpose inputs and the digital-analog converter, so data was processed and stored in the memory of the NodeMCU board. The sensors used were a) DHT11 for the measurement of relative humidity and temperature, and b) MQ-135 to determine air pollution and concentration of particles in the environment.

3.2 Data Capture

As mentioned, the nodes of the sensor network were configured to measure environmental variables with a frequency of one minute for four weeks. This period was defined according to the ASHRAE 55-2017 standard, which establishes that the acquisition intervals must be five minutes or less and the sampling period greater than two hours or be directly determined based on the occupation schedules of the area of interest [1].

Table 1. Extract of data collected through the implemented sensor network.

Date	Temperature (°C)			Humidity (%)			Air Pollution (PPM)		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
2020-05-25 07:01:00	24.0	24.0	24.0	45.0	48.0	47.0	83.9	83.3	81.6
2020-05-30 07:02:00	24.0	23.0	23.0	47.0	48.0	47.0	84.3	83.8	81.4
2020-06-21 18:59:00	24.0	24.0	23.0	45.0	47.0	47.0	77.6	76.3	67.0
2020-06-21 19:00:00	23.0	24.0	23.0	46.0	45.0	47.0	77.7	76.5	67.4

Regarding the data collected, these measurements were made in a university work-space, where according to the ASHRAE 55-2017 standard, the activities carried out are classified as typical office tasks, where there is a low metabolic rate, that is, a low level of transformation of chemical energy into heat [1].

In this sense, for data collection, each of the sensors handled a local log in comma-separated values (CSV) format. Sensor data is generated as raw data including date and time data, physical variables, and sensor identifier. This data is extracted and groped into a single table that serves as a source for the fusion method. Furthermore, with the use of WiFi technology, the data were grouped in a single table. As an example, Table 1 shows an extract of the data collected through the sensors on temperature, humidity, and air pollution.

Based on the measurements obtained, the temperature recorded values in degrees Celsius, the humidity as a percentage of water present in the environment, and the air pollution in parts per million.

3.3 Bayesian Estimation

After obtaining the data, Bayesian estimation was used as a data fusion algorithm, based on the Bayes theorem, which uses the conditional probability to estimate an event X , after a previous measurement Y [13, 15]. The probabilistic information of the measurement Y about the event X is also described by a probability density function $P(x,y)$, known as conditional probability function:

$$P(x|y) = \frac{P(x) \cdot P(y|x)}{P(y)}.$$

In this sense, Bayesian estimation reflects the probability of the occurrence of measurement based on previous data. This probability can be updated as recent data measurements are made. In this way, when updating the probabilities of the measurements, Bayesian estimation can model the behavior of the environmental variables that generate data streams [12, 15, 19].

Thus, this paper presents the use of the conditional probability function to represent the estimated result from the fusion of sensor measurements, represented as $S_{sensor\ number}^{time}$. When replacing the probabilities of the measurements made by the sensors for Temperature (T), Humidity (H), and Air Pollution (C), we obtain:

$$P(T_i^1|T_i^0) = \frac{P(T_1^1) \cdot P(T_2^1) \cdot P(T_3^1) \cdot P(T_1^1|T_1^0) \cdot P(T_2^1|T_2^0) \cdot P(T_3^1|T_3^0)}{P(T_1^0) \cdot P(T_2^0) \cdot P(T_3^0)}, \quad (1)$$

$$P(H_i^1|H_i^0) = \frac{P(H_1^1) \cdot P(H_2^1) \cdot P(H_3^1) \cdot P(H_1^1|H_1^0) \cdot P(H_2^1|H_2^0) \cdot P(H_3^1|T_3^0)}{P(H_1^0) \cdot P(H_2^0) \cdot P(H_3^0)}, \quad (2)$$

$$(C_i^1|C_i^0) = \frac{P(C_1^1) \cdot P(C_2^1) \cdot P(C_3^1) \cdot P(C_1^1|C_1^0) \cdot P(C_2^1|C_2^0) \cdot P(C_3^1|C_3^0)}{P(C_1^0) \cdot P(C_2^0) \cdot P(C_3^0)}. \quad (3)$$

In this way, data fused were obtained based on the previous measurements on each of the environmental variables, that is, integrated data were obtained for temperature, humidity, air pollution, respectively. Subsequently, these integrated values will be used for the thermal comfort analysis, under the conditions of the ASHRAE 55-2017 standard.

4 Results

As a result of the fusion process, based on the Bayesian estimate, integrated values were obtained for each of the environmental variables. These integrated values were projected in graphs that represent the conditional probability distribution of previous and future measurements, which were obtained by the sensor network.

4.1 Bayesian Estimation

Figure 3, part A, shows the temperature values measured in nodes 1, 2, and 3, through DHT11 sensors. The measurements of sensors 1 (blue) and 2 (orange) have similar behavior, with slight variations (values of 24.5 and 24.9 °C, respectively), while sensor 3 (green) presented a difference with respect to the other sensors (23.2 °C). This could be due to the characteristics of the workspace and the activities carried out, that is, a closed work environment with ventilation through a window and the main access door. Thus, based on the measurements, integrated values (red) were obtained. These fused values represent a general behavior of the sensation of heat in the environment. The mean value of this fusion was 23.8 °C, with a minimum and maximum of 19.6 and 28.7 °C respectively. It is important to note, that the fusion takes into account the probability of the measured values, in this way, the fusion is not based on the average of the

Table 2. Estimated root-mean-square error for the fused data.

Variable	Sensor	RMSE
Temperature	1	1.429
	2	1.424
	3	1.677
Humidity	1	1.577
	2	3.113
	3	1.760
Air pollution	1	1.447
	2	1.459
	3	2.040

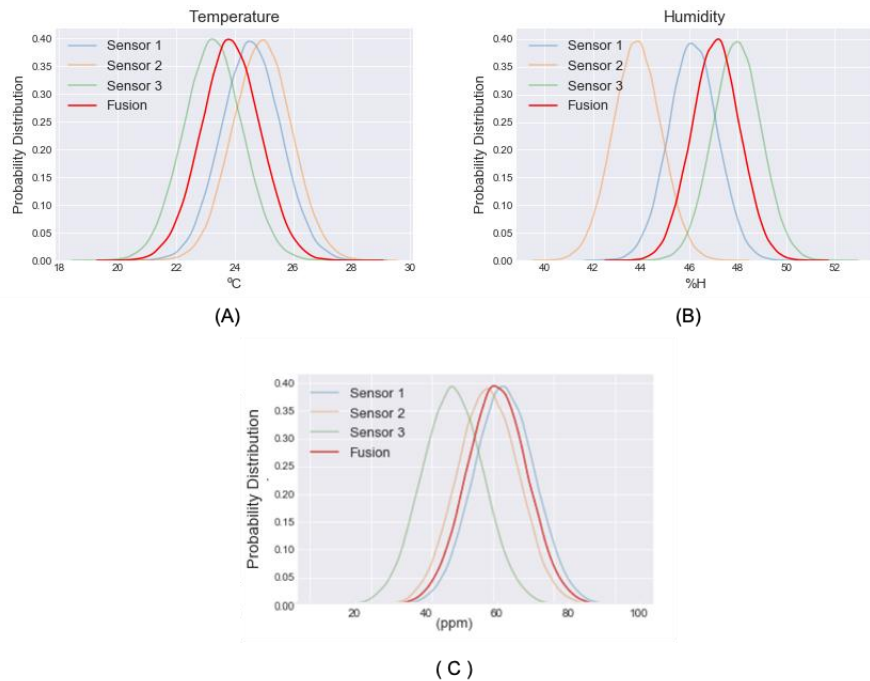


Fig. 3. Data fusion of variables temperature, humidity, and air pollution.

measurements, it represents the estimated values based on the probabilities of previous values.

Similarly, Figure 3 –part B– shows the values of the humidity percentage in the environment, acquired by the DHT11 sensors (nodes 1, 2, and 3). The measurements of sensor 1 (blue) and 3 (green) had high measurements, with the highest values recorded by sensor 3, presenting an average humidity sensation of 46 and 47.9%, respectively; while sensor 2 (orange) showed an average of 43.8%, recording lower values with respect to the measurements of sensors 1 and 3, the latter with the highest values.

This may be due to the location of the sensor, which is close to the window, where there is a greater amount of heat due to solar intensity and, at the same time, in contact with the exterior, concentrates a greater amount of water present in the environment.

Thus, based on humidity measurements, fused values (red) were obtained. These values were 47%, with a minimum of 42.8% and a maximum of 51.4%.

Air pollution, determined by levels of concentration of particles in the environment, is another of the physical variables that were analyzed due to its importance in determining healthy levels for closed environments. In Figure 3 –part C– it was observed that the values of sensors 1 (blue) and 2 (orange) present a similar probability distribution, with averages of particles per million (PPM) of 62 and 58, respectively. While sensor 3 (green) showed a behavior different from the other two sensors, whose PPM average was 47. The variation of this sensor is attributed to its location, which is close to a window that encourages airflow and ventilation. On the other hand, the result

of the fused values (red) was influenced by measurements from sensors 1 and 2, with an average of 60 PPM.

4.2 Data Fusion Validation

As part of the data fusion validation process, the root-mean-square error (RMSE) was used. In this sense, the estimated data were compared with the measurements acquired by the sensor network. The summary of the results obtained for each variable is shown in Table 2.

Based on the results obtained, it was observed that the RMSE of data fusion reached low error levels. For example, for the case of measured temperatures, there are error percentages that range between 1.4 and 1.7%, which represents that the data fusion reached a high level of accuracy, that is, above 98.4%. These ensure that the precision of the data fusion method is trustworthy. A similar case occurs with the other measurements, whose levels of accuracy exceed 97.8% for humidity, 98.4% for air pollution. Consequently, the error results obtained allow us to validate, with a high confidence level, the Bayesian estimation method used for data fusion.

5 Conclusions

Data fusion is an area of research that has received great interest, due to the incorporation of sensors in computer systems and the need for these systems to interact with the environment that surrounds them, adapting to situations to provide appropriate services whenever necessary.

A sensor network was designed and implemented to collect data on physical variables in a certain university workspace. For this, DHT11 sensors were used to measure the percentage of water in the environment and the thermal sensation, and MQ-135 to measure the concentration of particles in the environment. These sensors were connected to a NodeMCU-type controller board, with storage and connection capabilities through WiFi technology, with the aim of storing information about the environment, exploiting the versatility that this board offers when programmed in the Python language.

Bayesian estimation was used as a fusion method, in order to obtain integrated data from previous measurements (real values). The purpose of obtaining fused values is to make subsequent analyzes about the thermal comfort of users in closed workspaces.

To validate the fusion method, the RMSE was used, whose quantification of the difference to the table of the fused values for each physical variable, with respect to the values acquired by the sensors, allowed to determine the level of accuracy of the data fusion. In this way, data fusion results were obtained, with a confidence level above 98% on average in each of the variables analyzed.

Environmental analysis is of importance to avoid unhealthy levels that propitiate the symptoms caused by the syndrome of the sick building, having as priority users' health in closed environments.

As future work, we contemplate expanding the extension of the sensor network to monitor other areas of interest, in order to analyze the thermal comfort of users in

different work areas. Likewise, it is contemplated to extend the monitoring period to define work cycles or seasons in order to adapt parameters associated with each climatic season of the year.

Acknowledgment. This work was supported by UNAM-PAPIIT IA105320.

References

1. American Society of Heating, Refrigeration and Airconditioning Engineers (ASHRAE): Thermal Environmental Conditions for Human Occupancy Standard (2017)
2. Bernardos, A., Tarrío, P., Casar, J.: A data fusion framework for context-aware mobile services. In *Multisensor Fusion and Integration for Intelligent Systems*, pp. 606–613 (2008)
3. Bogaert, P., Gengler, S.: Bayesian maximum entropy and data fusion for processing qualitative data: theory and application for crowdsourced cropland occurrences in Ethiopia. *Stochastic Environmental Research and Risk Assessment*, 31(1) (2017)
4. Cook, D., Augusto, J., Jakkula, V.: Ambient Intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4), pp. 277–298 (2009)
5. De Paola, A., Ferraro, P., Gaglio, S., Re, G., Das, S.: An adaptive bayesian system for context-aware data fusion in smart environments. *IEEE Transactions on Mobile Computing*, 16(6), pp. 1502–1515 (2017)
6. Espressif Systems IOT Team: ESP8266EX Datasheet (2015)
7. Gellersen, H., Schmidt, A., Beigl, M.: Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications*, 7(5), pp. 341–351 (2002)
8. Guerriero, M., Wheeler, F., Koste, G., Dekate, S., Choudhury, N.: Bayesian data fusion for pipeline leak detection. In: *19th International Conference on Information Fusion*, 1(1), pp. 278–285 (2016)
9. Hall, D., Llinas, J.: An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1), pp. 6–23 (1997)
10. Jenkins, M., Gross, G., Bisantz, A., Nagi, R.: Towards context aware data fusion: modeling and integration of situationally qualified human observations to manage uncertainty in a hard+soft fusion process. *Information Fusion*, 21, pp. 130–144 (2015)
11. Jin, L., Zhang, Y., Zhang, Z.: Human responses to high humidity in elevated temperatures for people in hot-humid climates. *Building and Environment*, 114, pp. 257–266 (2017)
12. Kabir, G., Demissie, G., Sadiq, R., Tesfamariam, S.: Integrating failure prediction models for water mains: bayesian belief network-based data fusion. *Knowledge-Based Systems*, 85, pp. 159–169 (2015)
13. Khaleghi, B., Khamis, A., Karray, F., Razavi, S.: Multisensor data fusion: a review of the state-of-the-art. *Information Fusion*, 14(1), pp. 28–44 (2013)
14. Khattak, A., Akbar, N., Aazam, M., Ali, T., Khan, A., Jeon, S., Hwang, M., Lee, S.: Context representation and fusion: advancements and opportunities. *Sensors*, 14(6), pp. 9628–9668 (2014)
15. Kumar, M., Garg, D.: Multi-sensor data fusion in presence of uncertainty and inconsistency in data. In: *Milisavljevic, N. (ed.) Sensor and Data Fusion*, pp. 225–244 (2009)
16. Lahat, D., Adaly, T., Jutten, C.: Challenges in multimodal data fusion. In: *Signal Processing*, pp. 101–105 (2014)
17. Lee, P., You, P., Huang, Y., Hsieh, Y.: Inconsistency detection and data fusion in USAR task. *Engineering Computations*, 34(1), pp. 18–32 (2017)
18. Maddalena, R., Mendell, M., Eliseeva, K., Chan, W., Sullivan, D., Russell, M., Satish, U., Fisk, W.: Effects of ventilation rate per person and per floor area on perceived air quality,

- sick building syndrome symptoms, and decision-making. *Indoor air*, 25(4), pp. 362–370 (2015)
19. Muñoz-Benítez, J., Molero-Castillo, G., Benítez-Guerrero, E.: Data fusion architecture of heterogeneous sources obtained from a smart desk. In: *Intelligent Data Sensing and Processing for Health and Well-Being Applications*, pp. 23–40, Elsevier (2018)
 20. Ohtsuki, T.: A smart city based on ambient intelligence. *IEICE Transactions on Communications*, 100(9), pp. 1547–1553 (2017)
 21. Redlich, C., Sparer, J., Cullen, M.: Sick-building syndrome. *The Lancet*, 349(9057), pp. 1013–1016 (1997)
 22. Schiavon, S., Hoyt, T., Piccioli, A.: Web application for thermal comfort visualization and calculation according to ASHRAE Standard 55. In: *Building Simulation*, 7, pp. 321–334, Springer (2014)
 23. SEMARNAT: Programa de gestión para mejorar la calidad del aire en el estado de Veracruz de Ignacio de la llave 2016-2025 (2017)
 24. Siciliano, B., Khatib, O.: *Handbook of Robotics*. Springer (2016)
 25. Voids, S., Patterson, D., Patel, S.: Sensor Data Streams. In: Olson, J., Kellogg, W. (eds.) *Ways of Knowing*, pp. 291–322, Springer Science & Business (2014)
 26. White, F.: *Data Fusion Lexicon*. Tech. Rep., Joint Directors of Laboratories (1991)
 27. World Health Organization: *Who guidelines for indoor air quality: selected pollutants* (2010)

Cómputo bioinspirado en la prueba de software: una revisión sistemática de la literatura

Jethran E. Gómez San Gabriel, Ángel J. Sánchez García,
Karen Cortés Verdín

Universidad Veracruzana,
Facultad de Estadística e Informática,

jethran18@outlook.com,
{angesanchez, kcortes}@uv.mx

Resumen. Uno de los pilares para que la Industria 4.0 pueda ser una realidad, es el desarrollo de Software de calidad que permita la interconectividad y manejo de dispositivos que utilizamos. Para que un producto de software sea de calidad es necesario que sus requisitos sean validados y verificados, y es en la fase de prueba donde se realizan estos procedimientos. Para esto, es necesario que se haga una completa y correcta ejecución de un Plan de pruebas. Sin embargo, por el tiempo que implica desde su planificación hasta su ejecución la fase de prueba, este no puede ser terminado y en el peor de los casos iniciada. Esta Revisión Sistemática de la Literatura identifica las aportaciones que se han hecho con el Cómputo Bioinspirado y otros tipos de algoritmos de optimización en el proceso de la prueba de Software. Es así como, se identificaron 25 artículos, de cuales se analizaron los resultados de los algoritmos utilizados en un proceso de Prueba de Software. A partir de los resultados obtenidos se identificaron los algoritmos que se han utilizado tanto Evolutivos como de Inteligencia Colectiva, en qué problemas de la Prueba de Software han sido aplicados, así como los beneficios que se han obtenido al utilizarlos.

Palabras clave: Inteligencia Artificial, Prueba de Software, Cómputo Bioinspirado, Revisión Sistemática de la Literatura, Calidad de Software.

Bio-Inspired Computing in Software Testing: A Systematic Review of the Literature

Abstract. One of the pillars for Industry 4.0 to become a reality is the development of quality Software that allows interconnectivity and management of the devices that we use. For a software product to be of quality it is necessary that its requirements be validated and verified, and it is in the testing phase that these procedures are carried out. For this, it is necessary to carry out a complete and correct execution of a Test Plan. However, due to the time that the test phase implies from its planning to its execution, it cannot be finished and in the worst case started. This Systematic Literature Review identifies the contributions that

have been made with Bio-inspired Computation and other types of optimization algorithms in the Software testing process. Thus, 25 articles were identified, of which the results of the algorithms used in a Software Testing process were analyzed. From the results obtained, the algorithms that have been used both Evolutionary and Collective Intelligence were identified, in which Software Testing problems they have been applied, as well as the benefits that have been obtained when using them.

Keywords: Artificial Intelligence, Software Testing, Bio-inspired Computing, Systematic Literature Review, Software Quality.

1. Introducción

El desarrollo de software hoy en día está presente en todas las aplicaciones y la infraestructura de la Industria 4.0. Es por ello que la Ingeniería de Software tiene como una de sus metas ofrecer calidad a sus productos. Un software de calidad es reflejado por el uso de metodologías, estándares y una buena ejecución de pruebas. Para el desarrollo de software existen modelos de desarrollo de software que ayudan a tener una buena ejecución de las fases de desarrollo que son: Análisis, Diseño, Construcción, Prueba y Mantenimiento. En cada una de las fases se espera que haya revisiones para el cumplimiento de los requisitos funcionales y no funcionales.

En la fase de pruebas se espera que se verifiquen y validen los requisitos funcionales. En esta fase se busca establecer un mínimo de casos de prueba que abarquen todos los requisitos. Las pruebas de software son realizadas para que el producto salga con el mínimo de defectos posibles. Los casos de prueba (CP) son usados para verificar el cumplimiento de los requisitos del software [1].

Elaborar un caso de prueba puede ser un trabajo fácil pero tedioso ya que se deben generar múltiples entradas y posibles salidas para una sola funcionalidad. No obstante, al realizar una planeación de las pruebas, se suele no dar el tiempo necesario para que puedan ser aplicadas. Por lo tanto, es posible que al poco tiempo dado a la fase de pruebas no se pueda ejecutar una planeación que abarque cada uno de los requisitos funcionales. Debido a lo cual, es necesario optimizar la generación y ejecución de pruebas.

Por otro lado, uno de los elementos que se maneja en la Industria 4.0 para que podamos tener procesos automatizados y toma de decisiones objetivas, es el uso de Inteligencia Artificial, específicamente los Algoritmos Bioinspirados como una opción para procesos de optimización (numérica o combinatoria). Es por esto que en la presente Revisión Sistemática de Literatura (RSL) se plantean algunas preguntas de investigación que permitan dar un conocimiento sobre el uso de Algoritmos Bioinspirados en las pruebas de software.

Este artículo está organizado como sigue: en la sección 2 se muestra el método de investigación. En la sección 3 se presentan y discuten los principales resultados obtenidos. En la sección 4 y última, se comparten las conclusiones y el trabajo futuro.

Tabla 1. Preguntas de investigación.

Pregunta	Motivación
P1.- ¿Qué estrategias de Cómputo Bioinspirado se han usado en la prueba de software?	Conocer las estrategias de Cómputo Bioinspirado que se han aplicado a la fase de Prueba de Software.
P2.- ¿Qué tipos de problemas en la Prueba de Software se han abordado con el Cómputo Bioinspirado?	Conocer el tipo de problemas de Prueba que se han sido abordados con este enfoque.
P3.- ¿Qué beneficios se han obtenido al utilizar Cómputo Bioinspirado en la prueba de Software?	Conocer los beneficios de la aplicación de estas técnicas de optimización en la fase de Prueba.

Tabla 2. Bibliotecas digitales seleccionadas.

Biblioteca	Descripción	Enlace
ScienceDirect	Colección de investigación de Ciencias Físicas e Ingeniería.	https://www.sciencedirect.com/
SpringerLink	Colección de revistas, libros y obras científicas, tecnológicas y médicas.	https://link.springer.com/
IEEE Xplore	Acceso a contenido científico y técnico de libros, revistas, conferencias y cursos.	https://ieeexplore.ieee.org/
ACM Digital Library	Biblioteca digital dirigida principalmente al campo informático.	https://dl.acm.org/

Tabla 3. Criterios de selección.

Criterios de inclusión	Criterios de exclusión
C11.- Artículos escritos en inglés.	CE1.- Artículos que el año de publicación sea menor al año 2000
C12. Artículos que en el resumen o introducción toque al menos dos términos de búsqueda empleados en la cadena.	CE2.- Trabajos en presentaciones o reportes técnicos.
C13.- Artículos que traten sobre el Cómputo Bioinspirado en la prueba de Software.	CE3.- Artículos repetidos

2. Método de investigación

La Revisión Sistemática de la Literatura (RSL) fue basada en la guía provista por Kitchenham y Charters [3]. Esta investigación fue conducida en tres fases: (1) planeación, (2) ejecución de la búsqueda y (3) el reporte.

Tabla 4. Fases de búsqueda para la exclusión.

Fases	Criterios
Fase 1	Artículos que sean menores al año 2000
Fase 2	Artículos que no sean académicos o de investigación
Fase 3	Artículos que no se encuentren escritos en inglés
Fase 4	Artículos que en el resumen o introducción no toquen al menos dos términos de búsqueda
Fase 5	Artículos que no traten sobre el Cómputo Bioinspirado en la prueba de software
Fase 6	Revisión de artículos similares (continuación de trabajos).

Tabla 5. Número de estudios por fase.

Biblioteca digital	Fase 0	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6
SpringerLink	30,095	20,717	8,960	8,947	9	6	1
ScienceDirect	25,295	21,615	16,792	16,792	15	2	0
ACM Digital Library	75,400	43,541	35,076	35,076	5,591	22	12
IEEE Xplore	513	349	74	74	12	12	12

2.1. Planeación

Preguntas de investigación. En esta sección se establecieron las preguntas de investigación (PI) que se desean contestar con esta RSL. A través de ellas se pretende exponer trabajo previo sobre Cómputo Bioinspirado en la Prueba de Software, así como los resultados que se han obtenido al usarlas y encontrar nichos de oportunidad para futuras investigaciones en este campo. Las PI seleccionadas se muestran en la Tabla 1.

Selección de fuentes y estrategia de búsqueda. Las fuentes que se utilizaron para realizar esta RSL fueron las cuatro bibliotecas digitales mostradas en la Tabla 2, debido a que los artículos, al ser publicados en estas bases de datos, pasan por un proceso de arbitraje por expertos en el área. Además, se optaron por estas cuatro fuentes de información ya que contienen artículos del área de la Informática, Ingeniería y disciplinas afines.

Después de la identificación de las palabras clave descritas: *evolutionary*, *Optimization*, *software* y *test*, se procedió a la elaboración de la cadena de búsqueda mostrada a continuación:

((*evolutionary*) AND (*optimization*) AND (*software test*)).

Criterios de selección. Para la selección de los estudios a revisar, se definieron los criterios de inclusión y exclusión que se muestran en la Tabla 3.

Dados los siguientes criterios definidos, se procedió a realizar la selección de los estudios primarios. Para la identificación de los estudios se realizó un proceso por fases de búsqueda como se muestra en la Tabla 4.

Tabla 6. Puntos clave de la revisión de los estudios.

Problemas en la prueba de software	Estrategia	Beneficio en la prueba de software	Estudios primarios
Búsqueda de ruta más eficiente	Algoritmo Genético/Genetic algorithm (GA)	Mayor cobertura	[5], [25], [26]
Generación de datos de prueba	Optimización de Enjambre de Partículas/ Particle swarm optimization (PSO)	Mayor cobertura	[5], [8], [17]
Generación de datos de prueba	Algoritmo Genético/Genetic algorithm (GA)	Mayor cobertura	[14], [15], [16]
Búsqueda de ruta más eficiente	Colonia de Hormigas/ Ant Colony (AC)	Priorización de rutas prueba	[9], [22]
Búsqueda de ruta más eficiente	Prueba Estadística	Mayor cobertura	[18], [17]
Generación de datos de prueba	Algoritmo Genético/ Genetic algorithm (GA)	Priorización de rutas prueba	[15], [16]
Generación de datos de prueba	Optimización Combinatoria	Sin especificar	[28]
Búsqueda de ruta más eficiente	Recocido Simulado/ Simulated Annealing (SA)	Mayor cobertura	[26]
Generación de datos de prueba	Evolución diferencial/ Differential Evolution (ED)	Sin especificar	[10]
Generación de datos de prueba	Algoritmo Genético Inmune/ Immune Genetic Algorithm (IGA)	Mayor cobertura	[12]
Generación de datos de prueba	Colonia Artificial de Abejas / Artificial Bee Colony (ABC)	Mayor cobertura	[16]
Generación de datos de prueba	Colonia Artificial de Abejas / Artificial Bee Colony (ABC)	Priorización de rutas prueba	[16]
Generación de datos de prueba	Algoritmo Genético/ Genetic algorithm (GA)	Sin especificar	[11]
Generación de datos de prueba	Optimización de Enjambre de Partículas/ Particle swarm optimization (PSO)	Sin especificar	[11]
Generación de datos de prueba.	Varios	Sin especificar	[13]
Mejorar la cobertura de las pruebas	Algoritmos Evolutivos Multiobjetivo / Multi-Objective Evolutionary Algorithms (MOEA)	Mayor cobertura	[6]
Generación de datos de prueba.	Algoritmos Genéticos y Bosque Aleatorio	Mayor cobertura	[24]
Mejorar la cobertura de las pruebas	Optimización de Enjambre de Partículas/ Particle swarm optimization (PSO)	Priorización de casos de prueba	[7]
Mejorar la cobertura de las pruebas	Colonia de Hormigas/ Ant Colony (AC)	Mayor cobertura	[19]
Mejorar la cobertura de las pruebas	Algoritmo Genético/Genetic algorithm (GA)	Mayor cobertura	[19]
Mejorar la cobertura de las pruebas	Colonia de Hormigas/ Ant Colony (AC)	Priorización de ruta prueba	[20]
Mejorar la cobertura de las pruebas	Algoritmo Genético/ Genetic algorithm (GA)	Priorización de ruta de prueba	[21]
Búsqueda de ruta más eficiente	Varios	Mayor cobertura	[27]
Sin especificar	Varios	Sin especificar	[23]

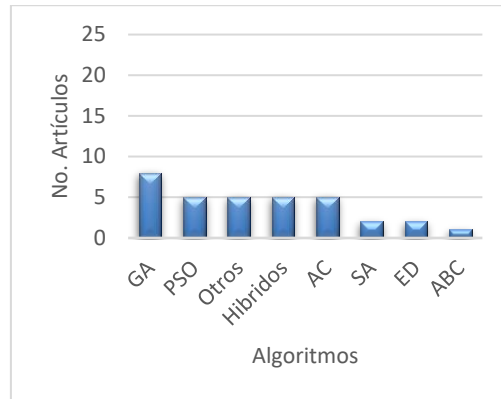


Fig. 1 Algoritmos identificados que se han usado en la Prueba de Software.

Método de síntesis de datos. Para las respuestas las preguntas de investigación, se utilizó el método de síntesis narrativa explicada en la guía de Kitcheman y Charters [3]. Fue posible hacer una extracción de datos mediante una lectura detallada de cada uno de los estudios.

2.2. Ejecución de la búsqueda

Una vez realizada la búsqueda con los criterios definidos en las fuentes seleccionadas, se obtuvieron 53 artículos. Sobre estos artículos obtenidos, se realizó una revisión sobre los títulos, resúmenes y conclusiones, que dieran indicios de las respuestas de alguna pregunta de investigación, obteniendo 35 artículos. Después, se realizó una revisión a profundidad de los artículos que dio como resultado una cantidad de 25 cabe aclarar que se dejaron de lado 7 artículos ya que no se encontraban disponibles. La cantidad de estudios seleccionados por fase de cada una de las bibliotecas se muestra en la Tabla 5.

3. Discusión de resultados

En esta sección se ofrece un resumen de los resultados de la Revisión Sistemática de la Literatura, los resultados de la extracción de datos obtenidas y las respuestas correspondientes de las preguntas de investigación. La Tabla 6 describe estos puntos clave de la extracción de la información.

Para responder a las preguntas, se utilizaron los diferentes hallazgos en los estudios vistos en la Tabla 6.

1. ¿Qué estrategias de Cómputo Bioinspirado se han usado en la prueba de software?

En la Fig. 1 se pueden observar diferentes tipos de Algoritmos Bioinspirados, de los cuales se mencionan (por sus nombres en inglés) *Genetic Algorithm (GA)*, *Particle Swarm Optimization (PSO)*, *Ant Colony (AC)*, *Simulated Annealing (SA)*, *Differential*

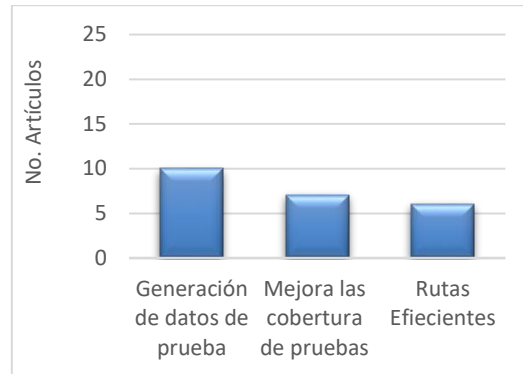


Fig. 2 Problemas de la Prueba de Software abordados con Cómputo Evolutivo.

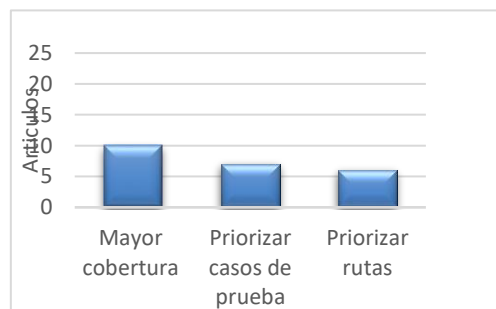


Fig. 3 Beneficios encontrados en los estudios en la fase de prueba.

Evolution (DE) y *Artificial Bee Colony (ABC)*. Además, en la categoría de otros se englobaron otros enfoques de algoritmos como optimización combinatoria y otros algoritmos de optimización [4, 17, 18, 27]. Mientras que los híbridos son una combinación de dos o más algoritmos que su principal función es mejorar las deficiencias de otro algoritmo.

2. ¿Qué tipos de problemas en la prueba de software se han abordado con el Cómputo Bioinspirado?

En la Fig. 2 se puede observar que existe un mayor número de estudios que hablan sobre la generación de casos de prueba (GCP). Se busca una solución a este problema, ya que cuando se elaboran un mínimo de Casos de Prueba (CP) ocurre que los datos de entrada y salida no abarcan los posibles defectos con los que el sistema cuenta. Algoritmos como AG intentan encontrar las variables de entrada de una manera eficiente y rápida para uno o varios CP.

Esto nos lleva al segundo problema que es la mejora de cobertura de prueba. Cuando se hace un conjunto de CP a un sistema, se espera entrar al mayor número de condiciones posibles [2]. Es por eso por lo que es importante tener un buen conjunto de datos entradas que abarquen el mayor número de rutas.

Por último, se encuentra el problema de *rutas eficientes o priorización de rutas*. En los estudios consultados se encontró una relación con las pruebas de regresión, las cuales están basadas en la idea de probar un sistema que ya fue aplicado en un conjunto de CP, pero al que se le han introducido nuevas funcionalidades o cambios [20]. La pregunta a este problema es ¿Es necesario ejecutar todas las pruebas?, con lo que la optimización busca escoger una ruta específica que tenga una mejor cobertura.

Es importante aclarar que cualquiera de los algoritmos mencionados en la sección anterior puede ofrecer una solución a cualquiera de estos tres problemas, pero hay unos que ofrecen mejor solución a un problema dado que en los estudios se hicieron evaluaciones a diferentes programas con diferentes AE [9, 14].

3. ¿Qué beneficios se han obtenido al utilizar *Cómputo Bioinspirado* en la prueba de software?

En la Fig. 3 se puede observar los beneficios encontrados en esta investigación. Se pudo capturar que los beneficios principales son tener una mejor cobertura de los CP ejecutados respecto a otros AE. En los estudios se observó que la cobertura es dada por la función de aptitud aplicado a un producto de software. El individuo óptimo sería el que mayor cobertura encuentre en Grafo de Flujo de Control (GFC) por cada una de sus ramas [14].

En el segundo caso está el priorizar los casos de prueba que garantizan el ejecutar un número mínimo de casos de prueba que abarquen la mayor cobertura, es decir, un CP garantice un criterio dado [15, 21].

En el tercer caso se busca priorizar las rutas con las más se cuenta por diferentes criterios, dos de estos pueden ser basados por costo y tiempo. Dependiendo de cómo sean seleccionados los operadores el algoritmo encontrará la ruta óptima.

4. Conclusiones y trabajo futuro

Este trabajo presenta una Revisión Sistemática de la Literatura enfocada a las búsquedas de aplicaciones de *Cómputo inspirado* en la naturaleza en la Prueba de Software. La motivación principal es la búsqueda de nichos de oportunidad en la Prueba de Software, que ayuden al aseguramiento de la calidad en el proceso y el producto de Software, y por ende, impactar en la calidad de las aplicaciones desarrolladas en la Industria 4.0.

El *Cómputo Bioinspirado* en el área de la Prueba de Software ha tenido un gran interés en los últimos 15-20 años. Se encontraron trabajos recientes que buscan optimizar en 3 diferentes problemas. También, se encontraron trabajos donde se comparaban resultados de algunos Algoritmos Evolutivos e Inteligencia Colectiva para dependiendo del problema en cuestión.

Con base en los artículos estudiados se observa una gran oportunidad en los algoritmos *Bioinspirados Híbridos*. Estos presentan una gran ventaja sobre los algoritmos *bioinspirados* per sé que cuentan con algunas deficiencias. En el uso de otros algoritmos que no necesariamente fueran *bioinspirados*, como pueden ser los

algoritmos estadísticos o procesos gaussianos, se encuentra una oportunidad de mejorar para el cambio de los operadores que puedan dar mejores resultados.

Se puede observar que los Algoritmos bioinspirados han tenido una gran aportación en la fase de Prueba de Software, resolviendo tres problemas principales que pueden hacer que los costos y tiempos se reduzcan drásticamente, especialmente en la búsqueda de mayor cobertura de pruebas. Esto debido a que se tiene una buena exploración de la mayoría del espacio de búsqueda, siendo el Algoritmo Genético el más utilizado.

También se pudo observar que varios de los experimentos realizados eran hacia sistemas pequeños, dejando para trabajo futuro el uso de algoritmos bioinspirados para sistemas grandes. Además, con el uso de algoritmos híbridos, se ofrece una nueva mejora en los resultados. Por lo tanto, se encuentra un nicho de oportunidad en el uso del Cómputo Bioinspirado en el área de fase de Pruebas combinado con otros enfoques de optimización e incluso de Aprendizaje Máquina.

Referencias

1. Sánchez, J.: Pruebas de Software. Fundamentos y Técnicas, 1, http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf (2015)
2. Nidhra, S.: Black box and white box testing techniques - a literature review. *International Journal of Embedded Systems and Applications*, 2(2), pp. 29–50 (2012)
3. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Department of Computer Science University of Durham, Durham, UK, Tech. Rep. (2007)
4. Sagarna, R., Lozano, J.: Variable search space for software testing. In: *International Conference on Neural Networks and Signal Processing*, Nanjing, China: IEEE (2003)
5. Koleejan, C., Xue, B., Zhang, M.: Code coverage optimisation in genetic algorithms and particle swarm optimisation for automatic software test data generation. In: *IEEE Congress on Evolutionary Computation (CEC)*. Wellington, New Zelanda: IEEE (2015)
6. do Nascimento Ferreira, T., Kuk, J., Pozo, A., Vergilio, S.: Product selection based on upper confidence bound MOEA/DDRA for testing software product lines. In: *IEEE Congress on Evolutionary Computation (CEC)*, Curitiba, Brazil: IEEE (2016)
7. Hla, K., Choi, Y., Sou-Park, J.: Applying particle swarm optimization to prioritizing test cases for embedded real time software retesting. In: *IEEE 8th International Conference on Computer and Information Technology Workshops*, Korea: IEEE (2008)
8. Abdul, R., Ejaz, N., Abbas, Q., Rehman, S.U., Shahid, A.A.: PSO based test coverage analysis for event driven software. In: *2nd International Conference on Software Engineering and Data Mining, SEDM'10*, pp. 219–224 (2010)
9. Sayyari, F., Emadi, S.: Automated generation of software testing path based on ant colony. In: *2nd International Congress on Technology, Communication and Knowledge, ICTCK'15*, pp. 435–440, Institute of Electrical and Electronics Engineers Inc. (2016)
10. Becerra, R.L., Sagarna, R., Yao, X.: An evaluation of differential evolution in software test data generation. In: *IEEE Congress on Evolutionary Computation, CEC*, pp. 2850–2857 (2009)
11. Lațiu, G.I., Creț, O.A., Văcariu, L.: Automatic test data generation for software path testing using evolutionary algorithms. In: *Proceedings 3rd International Conference on Emerging Intelligent Data and Web Technologies, EIDWT*, pp. 1–8 (2012)

12. Bouchachia, A.: An immune genetic algorithm for software test data generation. pp. 84–89, Institute of Electrical and Electronics Engineers (IEEE) (2008)
13. Feldt, R., Poulding, S.: Broadening the search in search-based software testing: it need not be evolutionary. In: Proceedings 8th International Workshop on Search-Based Software Testing, SBST'15, pp. 1–7, Institute of Electrical and Electronics Engineers Inc. (2015)
14. Zhu, Z., Xu, X., Jiao, L.: Improved evolutionary generation of test data for multiple paths in search-based software testing. In: IEEE Congress on Evolutionary Computation, CEC'17, pp. 612–620, Institute of Electrical and Electronics Engineers Inc. (2017).
15. Xu, X., Jiao, L., Zhu, Z.: Boosting search based software testing by using ensemble methods. In: IEEE Congress on Evolutionary Computation, CEC'18, Proceedings, Institute of Electrical and Electronics Engineers Inc. (2018)
16. Mann, M., Tomar, P., Sangwan, O.P.: Bio-inspired metaheuristics: evolving and prioritizing software test data. *Applied Intelligence*, 48(3), pp. 687–702 (2018)
17. Poulding, S., Alexander, R., Clark, J. A., Hadley, M. J.: The optimisation of stochastic grammars to enable cost-effective probabilistic structural testing. In: GECCO 2013, Proceedings of the 2013 Genetic and Evolutionary Computation Conference, pp. 1477–1484. <https://doi.org/10.1145/2463372.2463550> (2013)
18. Deshmukh, J., Horvat, M., Jin, X., Majumdar, R., Prabhu, V. S.: Testing cyber-physical systems through Bayesian optimization. *ACM Transactions on Embedded Computing Systems*, Vol. 16, Association for Computing Machinery (2017)
19. Chivilikhin, D., Ulyantsev, V., Tsarev, F.: Test-based extended finite-state machines induction with evolutionary algorithms and Ant Colony Optimization. In: GECCO'12 Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Companion, pp. 603–606 (2012)
20. Singh, Y., Kaur, A., Suri, B.: Test case prioritization using ant colony optimization. *ACM SIGSOFT Software Engineering Notes*, 35(4), 1–7 (2010)
21. Mayan, J. A., Ravi, T.: Test case optimization using hybrid search technique. In: ACM International Conference Proceeding Series, Association for Computing Machinery (2014)
22. Srivastava, P. R.: Structured testing using ant colony optimization. In: ACM International Conference Proceeding Series, pp. 203–207 (2010)
23. Ali, S.: Search-based test optimization for software system. In: GECCO 2018 Companion, Proceedings of the 2018 Genetic and Evolutionary Computation Conference Companion, pp. 328–348, Association for Computing Machinery, Inc. (2018)
24. Satya, S. P. K., Prasad, S. U. M. D., Gopi, K. D.: Recommendation and regression test suite optimization using heuristic algorithms. In: ACM International Conference Proceeding Series, pp. 202–203, Association for Computing Machinery (2015)
25. Rao, K. K., Raju, G., Nagaraj, S.: Optimizing the software testing efficiency by using a genetic algorithm. *ACM SIGSOFT Software Engineering Notes*, 38(3), 1–5 (2013)
26. Yan, L., Hu, W., Han, L.: Optimize SPL test cases with adaptive simulated annealing genetic algorithm. In: ACM International Conference Proceeding Series. Association for Computing Machinery (2019)
27. Din, F., Zamli, K. Z.: Fuzzy adaptive teaching learning-based optimization strategy for GUI functional test cases generation. In: ACM International Conference Proceeding Series, pp. 92–96, Association for Computing Machinery (2018)
28. Shir, O. M., Doerr, C., Bäck, T.: Compiling a benchmarking test-suite for combinatorial black-box optimization: A position paper. In: GECCO 2018 Companion, Proceedings of the 2018 Genetic and Evolutionary Computation Conference Companion, pp. 1753–1760, Association for Computing Machinery, Inc (2018)

Diseño y construcción de sensor de humedad IoT hecho de PVDF para aplicaciones en la industria del cultivo

Lizabeth Berenice Contreras Cariño, Carlos Omar González Morán

Universidad Autónoma del Estado de México,
Laboratorio de Investigación de Materiales y Procesos Inteligentes,
México

lbcc050495@gmail.com

Resumen. Este proyecto se realizó con el fin de desarrollar el diseño y construcción de un sistema de adquisición de datos que se ven reflejados en una aplicación móvil con sistema Android, en el cual se observan los niveles de humedad del sustrato de una planta. En este punto el sensor que fue hecho de fluoruro de un material ferroeléctrico PVDF (Polyvinil Difloruro) utilizando la técnica de Electrospinning. Este sensor desarrollado se encuentra en la tierra de la planta y este envía los datos de humedad a la aplicación móvil. El prototipo se implementó sobre una placa de circuito impreso PCB propio para el microcontrolador ATtiny85, que funciona como el ordenador del sistema de adquisición y este a través de un módulo Bluetooth (HC-05) manda del sensor de PVDF los datos de la humedad a la aplicación móvil.

Palabras clave: Industria 4.0, Sensor, Humedad, IoT, Microcontrolador, PVDF, ATtiny85, 3D.

Design and construction of an IoT Humidity Sensor Made of PVDF for Applications in the Crop Industry

Abstract. This project was carried out in order to develop the design and construction of a data acquisition system that is reflected in a mobile application with an Android system in which the soil moisture levels of a plant are observed. At this point the sensor that was made of polyvinylidene fluoride (PVDF) using the Electrospinning technique. This sensor in the soil of the plant, this sends the soil moisture data to the mobile application. The prototype was implemented on its own PCB printed circuit board for the ATtiny85 microcontroller, which works as the acquisition system computer and this, through a Bluetooth module (HC-05), sends the soil moisture data to the PVDF sensor to the mobile app.

Keywords: Industry 4.0, Sensor, Humidity, IoT, Microcontroller, PVDF, ATtiny85, 3D.

1. Introducción

La tecnología de la Industria 4.0 esta transformando la agricultura ya que las necesidades concretas de nuestro tiempo van cambiando, la agricultura se enfrenta a un gran reto, puesto que aumentará la demanda de producción y también se deberá limitar el impacto al medio ambiente. Hoy en día algunas de las tecnologías más avanzadas están relacionadas con los cultivos para cambiar el modelo productivo. Los sensores actualmente los encontramos en todos lados y están revolucionando las empresas que se dedican a la automatización de invernaderos y/o cultivos, los sensores de humedad complementados con las IoT (Internet de las cosas) buscan el cuidado más eficiente y preciso de los cultivos.

Una de las ventajas de las IoT es que son de hardware libre y están basadas en una placa que contiene un microcontrolador que se puede programar mediante el Software de Arduino IDE, este nos ayuda a hacer una extensión de la electrónica para cualquier proyecto y funciona como un sistema embebido.

Las IoT (Internet de las cosas) están impactando la ciencia, tecnología, medio ambiente, entre otros. Una de sus aplicaciones es la adquisición de datos y la capacidad de guardarlos por medio de sensores para a su vez reflejar resultados en dispositivos que interactúan con el usuario final. Así se desarrollan dispositivos más económicos y compactos en tamaño, gracias a las IoT se puede monitorear o controlar con teléfonos inteligentes. También tenemos como parte de la industria 4.0 los diseños 3D, es una manera de crear formas o imágenes muy parecidas a la realidad con las tres dimensiones (X, Y, Z) del objeto a crear, hacemos que se conjugue el espacio real con los objetos y volúmenes. Hoy en día es posible la simulación 3D mediante cálculos basados en la proyección de entornos tridimensionales sobre pantallas bidimensionales como un simple monitor de una PC, entre otros. Por otro lado, las aplicaciones móviles efectúan cualquier tarea que le sea asignada y generalmente se utilizan para enviar o recibir datos de algún destino, lo cual actualmente ya se considera como una herramienta que fomenta la creatividad, innovación, comunicación que mantienen los dispositivos en vanguardia.

Nuestro prototipo permite al usuario que a través de una interfaz gráfica App, pueda ver el porcentaje de humedad que tiene la tierra que para nuestra prueba usamos como muestra la planta Bambú Palm (*Chamaedorea seifrizii*), tomando en cuenta que este prototipo puede ser empleado invernaderos o espacios de cultivo, una vez que ya se monitorizó la humedad por medio del sensor PVDF se podrá implementar un sistema de riego automático para la planta a través de comandos desde la aplicación móvil de nuestro teléfono, además de que la aplicación contiene una serie de tutorías con respecto a la planta en uso, esto con el objetivo de enseñar a las personas que no tienen conocimientos en el cuidado de las plantas y guiarse de la aplicación móvil para hidratarlas automáticamente con el sistema de riego a través de la comunicación de Bluetooth. Para la elaboración de sensor de humedad recurrimos a la técnica de electrospinning o electro hilado que ha sido ampliamente estudiada durante los últimos años gracias a la posibilidad de crear fibras en escala de micrómetros y nanómetros para una gran variedad de aplicaciones industriales, esta técnica aporta a los elementos desarrollados diversas características como: amplia superficie por unidad de área, porosidad y una serie de propiedades mecánicas, siendo atractiva a nivel

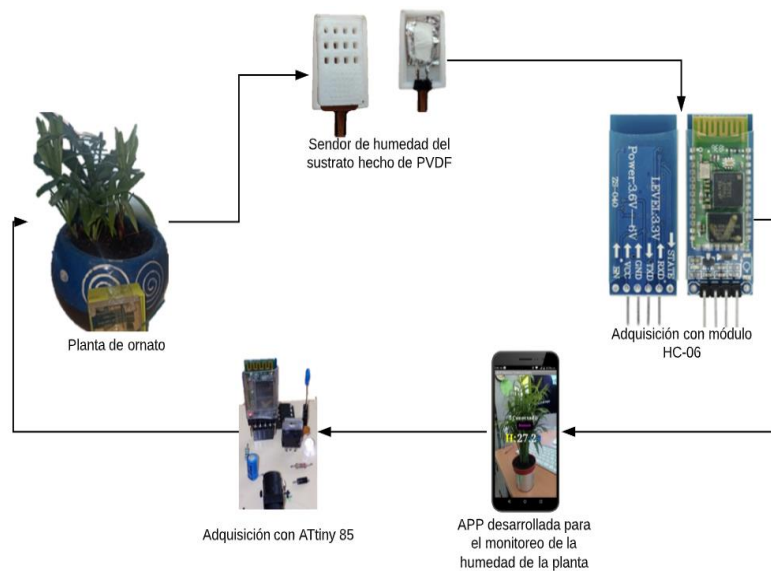


Fig. 1. Desarrollo del sistema de adquisición de datos utilizando tecnologías móviles a través de un sensor de humedad hecho de polivinil difluoruro (PVDF).

biotecnológico, la técnica es versátil ensamble lo que ha permitido procesar una gran variedad de polímeros.

La manera en la que se determinará la humedad de la planta con el sensor PVDF es saber el contenido de agua de la tierra de la planta y utilizar los cambios en la constante dieléctrica de la capacitancia que se puede calcular si se conocen la geometría de los conductores y las propiedades dieléctricas del aislante entre los conductores, que para nuestro caso son las paredes de nuestro sensor PVDF.

Por ejemplo, la capacitancia de un capacitor construido con dos placas paralelas ambas de área A separados por una distancia d es aproximadamente igual a la siguiente expresión matemática:

$$C = \frac{\epsilon A}{d},$$

dónde: C = Capacitancia (Sensor PVDF), ϵ = Constante dieléctrica (Tierra de la planta), d (separación de las placas) [9], A = Área superficial (polímero PVDF).

2. Metodología

En este proyecto se diseñó y desarrolló el prototipo de un sistema de adquisición de datos que obtiene el porcentaje de humedad [10] a nivel de la tierra de la planta de ornato a través de una aplicación móvil.



Fig. 2. Máquina de Electrospinning durante el proceso de inyección de la capa de material PVDF.

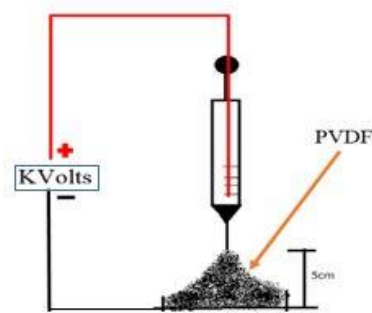


Fig. 3. Chorro eyectado de la concentración de PVDF.

2.1. Diseño del prototipo

El sistema de adquisición de datos hace la función de obtener el porcentaje de humedad de la planta de ornato a través del sensor PVDF que manda los datos al microcontrolador ATtiny85 y mediante módulo Bluetooth (HC-06) la información se despliega a una aplicación móvil, como se ve en la Figura 1.

2.2. Diseño del sensor PVDF

Se realizó una solución precursora en donde se utilizaron 9 pellets de PVDF para obtener 0.6321(gr) que se disolvieron en 2.4 (gr) de solvente (*N, N* dimetilformamida-DMF) durante 20 minutos aproximadamente para obtener una concentración del 18%, posteriormente este se colocó en un contenedor de jeringa común en la máquina de Electrospinning trabajando a 9.31 kV durante 16 minutos como se muestra en la Figura 2 [9].



Fig. 4. Forma sólida de la concentración del PVDF.



Fig. 5. Impresora 3D.



Fig. 6. Modelo de impresión 3D para sensor PVDF.

El chorro de polímero se hace desde una altura de 5 centímetros de alto y 2.5 centímetros de ancho como se ve en la Figura 3. Creando así una membrana de PVDF de 100 micrometros como se ve en la Figura 4.

Una vez que se obtiene la membrana de PVDF se hace el diseño 3D para recubrir el sensor. El diseño se hace desde la página Tinkercad.com y se obtiene el archivo .stl para así imprimirse en la impresora 3D CREATOR PRO como aparece en la Figura 5 y 6. El sensor PVDF se ensambla con salidas (pines) para su monitoreo como se ve en la Figura 7.

El sensor PVDF de humedad se encuentra emparedado entre dos placas de aluminio, estas placas se conectan a dos pines del microcontrolador como se muestra en la Figura 8.

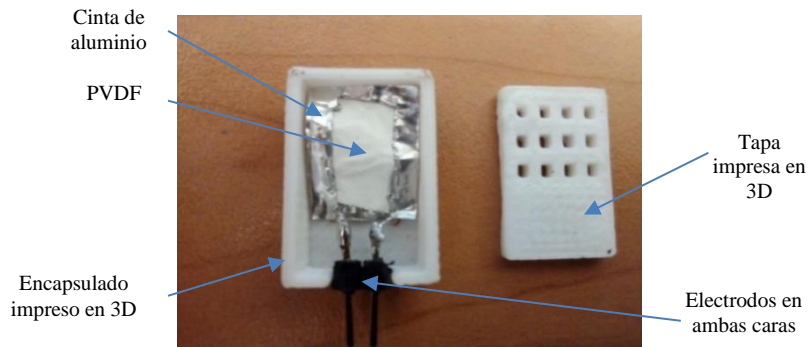


Fig. 7. Sensor de humedad puesto en su carcasa impresa en 3D.

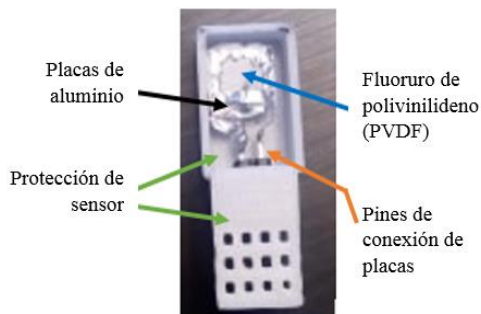


Fig. 8. Estructura del sensor de humedad PVDF con su encapsulado impreso en 3D.

El prototipo de sensor funciona como sensor de humedad de la tierra, con una película de PVDF y el circuito electrónico para controlar la cantidad de agua en la tierra.

2.3. Diseño y creación del prototipo

Para este prototipo, el diagrama eléctrico se desarrolló en el software Circuit Wizard así también para el diseño del circuito impreso para el microcontrolador ATtiny85, a este se le conecta el dispositivo PVDF previamente mostrado quedando el diagrama eléctrico completo como se muestra en la Figura 9.

Es importante mencionar que el diseño de la placa fue para sustituir la tecnología de las tarjetas comerciales de Arduino, haciendo un prototipo propio, más compacto, de menor costo y adaptado para el monitoreo de la humedad del sustrato en plantas de ornato o cultivo. Se utilizó un conector JACK de alimentación hembra para una alimentación de 12 volts, un módulo Bluetooth (HC-06) para él envió de datos a la aplicación móvil, un microcontrolador ATtiny85, también se incluyó un regulador de voltaje L7805 para que no existan variaciones de alimentación al circuito, de esta manera se diseñó el circuito que se muestra en la Figura 10 y la Figura 11.

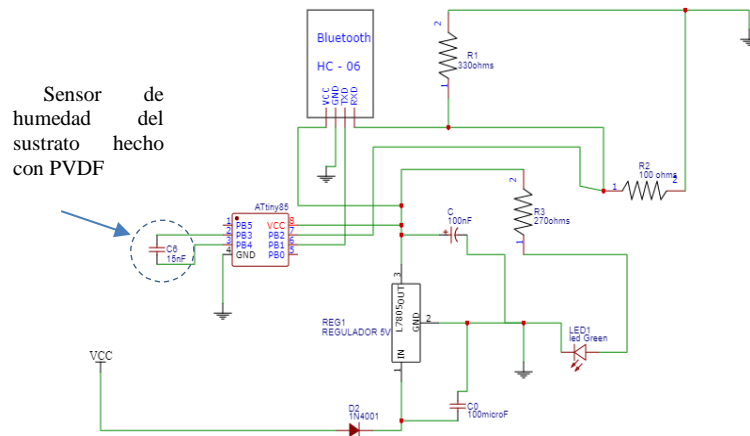


Fig. 9. Diagrama eléctrico del microcontrolador ATTiny85 con Bluetooth.

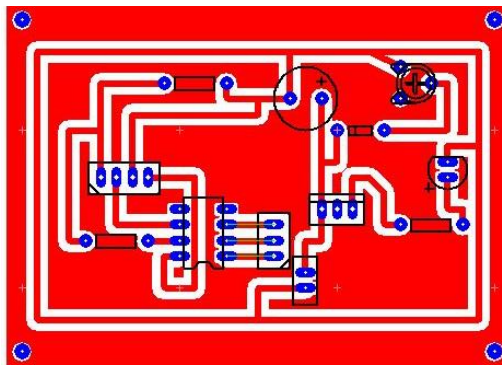


Fig. 10. Diagrama eléctrico del circuito del sistema de adquisición de datos.



Fig. 11. Prototipo físico terminado para la adquisición de la humedad.

Tabla 1. Comparativa de los valores de capacitores comerciales medidas entre el microcontrolador Atiny85 y la placa Arduino Uno.

Valor comercial	Arduino UNO(pF)	ATTiny85 (pF)
Capacitancia=60	62	42
	61	43
	61.7	42.5
Capacitancia=2	2	3.3
	2.3	3
	2.8	2.9
Capacitancia=10	10	8.4
	11	8.2
	10.5	8.3
Capacitancia=100	97	67.9
	96	67
	96.5	67.3
Capacitancia=19	19	14.2
	19	14
	18.7	14.4
Capacitancia=7	7	6.5
	6	6.3
	6.5	6.6

3. Pruebas

Se hicieron las pruebas al prototipo para saber qué valores se obtenían al medir la capacitancia de capacitores comerciales, al hacer las pertinentes comparaciones logramos obtener los siguientes datos como se muestran en la Tabla 1.

4. Resultados

Los datos de humedad se obtuvieron al conectar el sensor que se encuentra en la planta Bambú Palm y se monitoreo a través de la aplicación móvil durante 4 días como se muestra en la Figura 12. Inciso B). De los datos de la Tabla 3 datos se obtuvo la siguiente gráfica de la Fig.13. A su vez se conectó la aplicación móvil al módulo Bluetooth (HC-06) en donde se obtuvieron los datos de inicio y comienzo de la prueba en donde la aplicación a través de la interfaz gráfica nos muestra algunos de los datos obtenidos como se muestra en la Fig.12. Inciso A). Estos datos se monitorearon durante 4 días a la planta se le agregó una cantidad de 50 ml de agua con una temperatura de 26°C. La distancia que hay entre el sensor y la placa de monitoreo es de 10 cm.

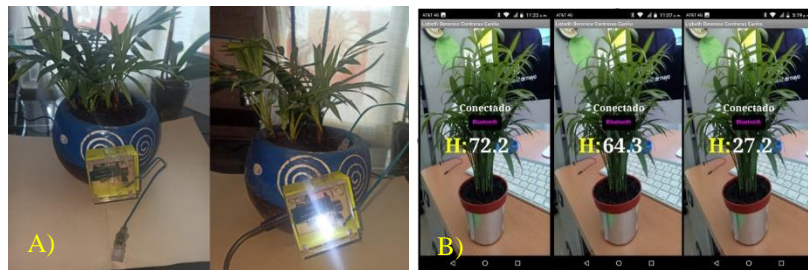


Fig. 12. A) Prototipo con ATtiny85 y módulo Bluetooth (HC-06) a una planta de ornato. B) Pantalla de una App de un dispositivo móvil (inicio y término de la prueba).

Tabla 2. Los datos obtenidos fueron los siguientes durante los días de monitorización.

ANÁLISIS DE VARIANZA						
Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Muestra	30403.5956	5	6080.719111	40916.98841	2.46E-46	2.620654148
Columnas	739.84	1	739.84	4978.362617	2.40275E-29	4.259677273
Interacción	1116.81333	5	223.3626667	1503.001121	3.9115E-29	2.620654148
Dentro del grupo	3.56666667	24	0.148611111			
Total	32263.8156	35				

Se hizo el análisis factorial “ANOVA”, y se obtuvo que el cálculo que hay entre la varianza F (capacitancia) y el valor crítico para F (Arduino Uno y ATTiny85) son la varianza que existe entre ambas placas. En la Tabla 2 nos muestra que no hay una diferencia por cada medición de capacitancia entre los diferentes valores críticos que son las placas de Arduino Uno y ATTiny85. El resultado de la interacción que se tiene esta dentro de nuestra Distribución F, el cual indica que no tenemos diferencia significativa entre los datos de las mediciones de las capacitancias entre los valores críticos.

Con respecto a los resultados obtenidos en este experimento y en experimentos anteriores [9], se comprueba el comportamiento lineal del sensor PVDF aplicado como sensor de humedad.

5. Conclusiones

En general este trabajo representa una forma de medir la humedad de manera inalámbrica utilizando la tecnología Bluetooth y guardando la información en un dispositivo móvil para generar un archivo .csv, el cual nos permite enviar esta información vía internet a un correo electrónico para tener el monitoreo de la humedad de manera remota. Es importante mencionar que como trabajo a futuro se pretende

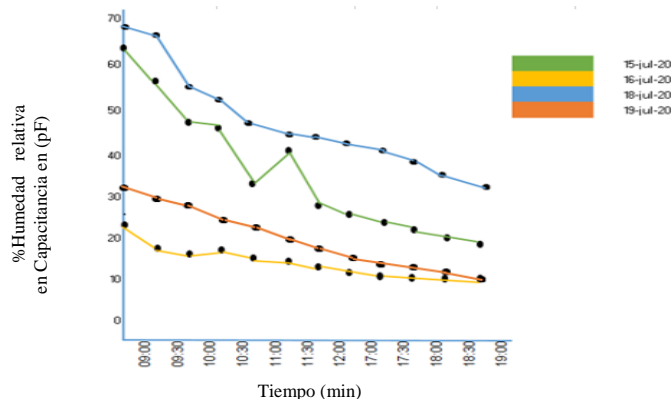


Fig. 13. Grafica de datos del nivel de humedad en base a la capacitancia del sensor PVDF.

Tabla 3. Los datos obtenidos fueron los siguientes durante los días de monitorización.

Riego 1				Riego 2			
Hora	Nivel de humedad	Hora	Nivel de humedad	Hora	Nivel de humedad	Hora	Nivel de humedad
15 de julio del 2020		16 de julio del 2020		18 de julio del 2020		19 de julio del 2020	
9:00	63.5	9:00	25.18	9:00	69.1	9:00	32.89
9:30	57.2	9:30	19.7	9:30	68.3	9:30	29.7
10:00	47.8	10:00	18.6	10:00	56.9	10:00	27.453
10:30	46.5	10:30	19.598	10:30	51.3	10:30	23.14
11:00	31	11:00	18.1	11:00	49.7	11:00	22.7
11:30	44.93	11:30	17.3	11:30	46.9	11:30	20
12:00	29.1	12:00	15	12:00	44.589	12:00	19.36
17:00	26.6	17:00	14.387	17:00	43.6	17:00	17.98
17:30	25	17:30	14	17:30	42.9	17:30	17.1
18:00	24.923	18:00	13.6	18:00	38.753	18:00	16.49
18:30	22.7	18:30	12.99	18:30	37.89	18:30	15.3
19:00	21.9	19:00	12	19:00	35.7	19:00	14.987

hacer un análisis de datos utilizando inteligencia artificial para poder determinar patrones de comportamiento en la tierra para el desarrollo de sistemas de riego agrícola inteligentes, para así poder optimizar el consumo de agua.

Referencias

1. UNAM: Depafquimunam. http://depa.fquim.unam.mx/amyd/archivero/Capacitores_21660.pdf (2016).
2. El jardín: Sistema de riego. <http://www.eljardin.ws/riego/historia-del-riego.html> (2015)

3. Core: Piisa. <https://core.ugr.es/sc-piisa/material/micro/sesion1/intro.pdf> (2014)
4. Duque-Sánchez, L.M., Rodríguez, L., López, M.: Electrospinning: la era de las nanofibras. *Rev. Iber. Polímeros*, 14(1), pp. 10–27 (2014)
5. Jera, J.S.: Historia del riego. www.doh.gov.cl/publicacionesyestudios/Documents/Historia%20del%20Riego.pdf (2015)
6. Kouro, S.: Sensores de Humedad. Universidad Técnica Federico Santa María, 25 (2001)
7. Dave, E.: Internet de las cosas cómo la próxima evolución de internet lo cambia todo. Cisco Internet Business Solutions Group (IBSG), pp. 2 (2012)
8. Barrio, M.: Internet de las cosas. Ed. Reus S.A., Madrid España, pp. 17–19 (2018)
9. González, O.: System for controlling the moisture of the soil using humidity sensors from a polyvinylidene fluoride fiber mats. EU: American Scientific Publishers (2013)
10. Torres, P.: Identificación de temperatura y humedad en un prototipo deshidratador de pulpa de *Annona Muricata*. Colombia: Ciencia y Poder Aéreo (2015)
11. Seminis: Los principales sistemas de riego. Seminis, Recuperado de: www.seminis.mx/blog-los-principales-sistemas-de-riego/ (2016)
12. Zarazaga, F.J.: Agricultura 4.0: las tecnologías de la industria 4.0 aplicadas al campo. Grupo de Sistemas de Información Avanzados (IAAA) (2017)
13. Mateos Maza, M.: El diseño 3D. <https://dibujourjc.files.wordpress.com/2015/02/marta-mateos-maza.pdf> (2014)

Algoritmo para el conteo de modelos en FNC

Pedro Bello López, Guillermo De Ita Luna, Meliza Contreras González,
Miguel Rodríguez Hernández

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

{pb5pbello, vikax68, mikelrodriguezr}@gmail.com,
comdeita@inaoep.mx,

Resumen. En este documento se presenta un algoritmo para el conteo de modelos de una fórmula especificada en forma normal conjuntiva (FNC), utilizando patrones representados como cadenas de ceros, unos y asteriscos lo que permite un mejor tratamiento computacional. Además, con el algoritmo desarrollado se generó una aplicación web para realizar pruebas con fórmulas en FNC. La estrategia utilizada en la propuesta algorítmica se basa en considerar el mismo conjunto de variables para todas las cláusulas, de esta forma podemos saber el total de modelos, donde el máximo es 2^n . El proceso consiste en aplicar el operador de revisión mediante los patrones mencionados para reducir los modelos, por lo que al final del proceso se obtiene el total de modelos de F.

Palabras clave: Algoritmo, Satisfactibilidad, #SAT, NP-Completo, Sistema.

Algorithm for Counting Models in FNC

Abstract. This document presents an algorithm for the counting of models of a formula specified in conjunctive normal form (FNC), using patterns represented as strings of zeros, ones and asterisks, which allows a better computational treatment. In addition, with the developed algorithm, a web application was generated to perform tests with formulas in FNC. The strategy used in the algorithmic proposal is based on considering the same set of variables for all the clauses, in this way we can know the total number of models, where the maximum is 2^n . The process consists of applying the revision operator through the aforementioned patterns to reduce the models, so at the end of the process the total number of F models is obtained.

Keywords: Algorithm, Satisfiability, #SAT, NP-Complete, System.

1. Introducción

El problema de satisfacibilidad (SAT) es considerado el primer problema NP-completo demostrado por Stephen Cook en el año de 1971. El problema SAT es central en la teoría de la computación. En la práctica, SAT es fundamental en la resolución de problemas de razonamiento automático, en el diseño y fabricación asistido por computadora, en el desarrollo de base de datos, en el diseño de circuitos integrados, en la integración de módulos para lenguajes de alto nivel, etc. El problema SAT en general es un problema de decisión, lo definimos como: dada una fórmula booleana F en Forma Normal Conjuntiva (FNC), formada de un conjunto de cláusulas con n variables, se debe encontrar una asignación a dichas variables tal que la F sea satisfactible, es decir que exista al menos una asignación de valores que haga verdadera la fórmula F .

Los problemas NP-completo son los problemas que caracterizan la clase de tal forma que al encontrar una solución del problema SAT se pueden resolver los demás problemas de la clase ya que se puede hacer una transformación en tiempo polinomial al problema correspondiente.

De esta forma el problema SAT consiste en decidir si F es satisfactible. Mientras que el problema de optimización MaxSAT consiste en determinar el número máximo de cláusulas que se pueden satisfacer simultáneamente [1]. Y el problema de conteo #SAT consiste en contar el número de asignaciones que satisfacen a F . Nuestra propuesta algorítmica propone un método para #SAT, es decir presentamos un algoritmo y su implementación computacional para contar el número de asignaciones (modelos) de una fórmula F en FNC.

Consideramos para cada cláusula el mismo conjunto de n variables y formamos cadenas de ceros, unos y asteriscos (*), el * se usa cuando una variable no aparece en la cláusula, posteriormente aplicamos un proceso de revisión para generar cláusulas independientes entre sí, es decir cláusulas que tengan una literal contradictoria. El método es iterativo tomando una a una las cláusulas de la fórmula F iniciando con el valor de 2^n modelos y se le van restando modelos de acuerdo al número de asteriscos ($2^{\#ast}$) de las cláusulas resultantes al aplicar el proceso de revisión. En la Fig.1 se muestra el proceso completo para la propuesta de conteo de modelos, la transformación de la fórmula en FNC a su representación de ceros, unos y * se realiza de forma manual por el usuario del sistema.

Dentro de las múltiples aplicaciones de #SAT, se encuentran las aplicaciones sobre el razonamiento proposicional, el cual constituye una pieza fundamental de la Inteligencia Artificial. Los problemas de satisfacibilidad SAT, MaxSAT y #SAT son problemas completos en las clases de complejidad NP, MAX NP y #P [2]. Un punto importante al trabajar con SAT, MaxSAT y #SAT en nuestro caso, es que, como representantes de sus respectivas clases de complejidad, hallar nuevas estrategias algorítmicas que los resuelven, implica que tales estrategias se pueden extender a los demás problemas de la clase correspondiente.

En este trabajo se presenta en la sección 2 el trabajo relacionado con el tópico del problema SAT sus principales aplicaciones, sus alcances y limitaciones. Posteriormente en la sección 3 se presenta la propuesta algorítmica para el conteo de modelos de una fórmula en FNC, se incluyen algunas consideraciones iniciales para el desarrollo del

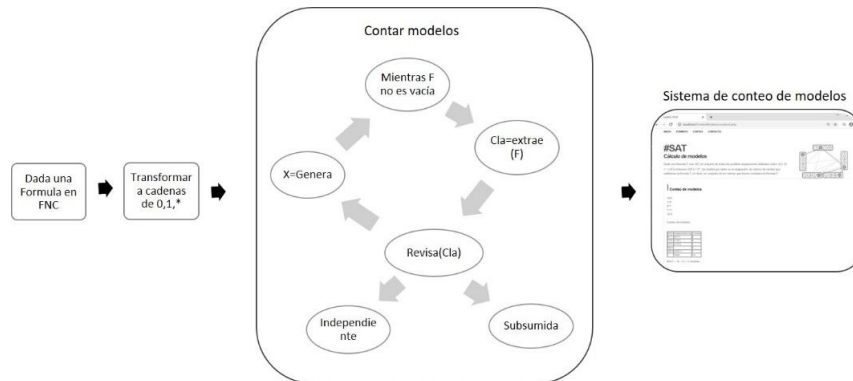


Fig. 1. Proceso general para el conteo de modelos.

algoritmo y a través de ejemplos se describe el algoritmo desarrollado. En la sección 4 se presentan los resultados del algoritmo implementado en una aplicación web y finalmente en las conclusiones se describe la idea general de la complejidad en tiempo de la propuesta del algoritmo para el conteo de modelos en FNC.

2. Trabajo relacionado

Los problemas de satisfactibilidad son estudiados intensivamente en las ciencias computacionales, debido a que son sin duda, los problemas más importantes en el campo de la complejidad computacional [3-4m], y una de las aplicaciones interesantes es cuando contribuyen significativamente a simplificar un problema que por naturaleza incluye procesos exponenciales. Existen diversas líneas de investigación en torno a estos problemas y las aplicaciones son muy variadas.

Dentro de los problemas de satisfactibilidad de restricciones, se encuentra el problema de decisión, que consiste en determinar si una fórmula es satisfactible o no. El problema de conteo asociado, es saber el número de asignaciones que satisfacen a una formula Booleana, conocido como el problema #SAT. Existen muchos problemas no resueltos en su totalidad que se derivan del problema #SAT, por lo que se tiene un constante interés por parte de los investigadores de diversas áreas, por encontrar diferentes formas de resolver el problema #SAT, en parte, debido al impacto que tendría una buena solución de este problema en muchas otras disciplinas y aplicaciones. Sabemos que en su completa generalidad, #SAT es un problema de la clase #P-completo. No se tiene aún demasiado conocimiento sobre las restricciones a #SAT donde su solución sea eficiente, esta es una de las razones por las que muchos investigadores están investigando en esta are de la ciencia.

Dentro de las múltiples aplicaciones de #SAT, se encuentran las aplicaciones sobre el razonamiento proposicional, el cual constituye una pieza fundamental de la Inteligencia Artificial y en los sistemas de agentes y agentes inteligentes que operan de forma robusta ante cambio repentinos impredecibles [4-1m].

En [5-2m] prueban que contar asignaciones falsificantes de lenguajes proposicionales es intratable aún para Horn y fórmulas monótonas, y aun cuando el tamaño de las cláusulas y el número de ocurrencias de las variables son extremadamente limitadas. Esto se debería contrastar con el caso de razonamiento deductivo, donde las teorías de Horn y las teorías con cláusulas binarias son reconocidas por la existencia de algoritmos de satisfactibilidad en tiempo lineal. Lo cual sorprende que aun aproximando el número de asignaciones que satisfacen es intratable para la mayoría de estas teorías restringidas, es decir, “aproximar” al razonamiento aproximado. También identifica algunas clases limitadas de fórmulas proposicionales para las cuales se pueden dar algoritmos eficientes para contar asignaciones satisfactorias.

En [6-3m] muestra cómo se extiende el problema de intratabilidad con el método de Monte Carlo de cadenas de Markov, así también resaltar los límites naturales del método. Utiliza técnicas de acoplamiento antes de introducir “acoplamiento del camino” una nueva técnica, la cual, simplifica radicalmente y mejora sobre métodos previos en el área.

En [5-2m] y [6-3m] se tiene el inconveniente de que no garantizan la complejidad en tiempo lineal, además de que al contar los modelos de Σ , no distinguen entre los modelos en los que una variable x toma el valor de 1 de aquellos en el que la misma variable x toma el valor de 0.

Se presentan algoritmos para contar modelos proposicionales del problema #SAT [7-5m]. Los algoritmos utilizan descomposiciones arbóreas de ciertos grafos asociados con la fórmula FNC dada; en particular consideran grafos de incidencia, dual y primordial. Describe el algoritmo coherentemente para una comparación directa con suficiente detalle para realizar una implementación razonablemente fácil. También discute varios aspectos de los algoritmos incluyendo tiempo del caso peor y requerimientos de espacio de almacenamiento. Podemos mencionar que existen algoritmos o técnicas de conteo de modelos. En [8-6m] se dividen las técnicas de conteo práctico de modelos en dos categorías: conteo aproximado y conteo exacto.

En [9-7m] se introduce el problema Max#SAT, una extensión del conteo de modelos (#SAT). Dada una fórmula sobre el conjunto de variables X , Y y Z , el problema Max#SAT es maximizar sobre las variables X el número de asignaciones a Y que se puede extender a una solución con algunas asignaciones a Z . Muestra también que Max#SAT tiene aplicaciones en muchas áreas mostrando como se puede utilizar para resolver problemas en inferencia probabilística, planeación, síntesis de programas y análisis de flujo de información cuantitativa. También proporciona un algoritmo el cual, haciendo únicamente muchas llamadas polinomialmente a una máquina oráculo NP puede aproximar el conteo máximo dentro de un error multiplicativo válido.

Se desarrolla GANAK, un contador de modelos exacto probabilístico escalable que supera los contadores de modelos aproximados y exactos, ApproxMC3 y #SAT respectivamente. En sus experimentos, el conteo de modelos regresado por GANAK fue igual al conteo exacto de modelos para todos los puntos de referencia [10-8m].

Existen algoritmos exhaustivos de última generación para contar modelos, por ejemplo, un algoritmo DPLL con semántica formal [11], una prueba de la corrección y diseño modular. El diseño modular se base en la separación del núcleo principal del

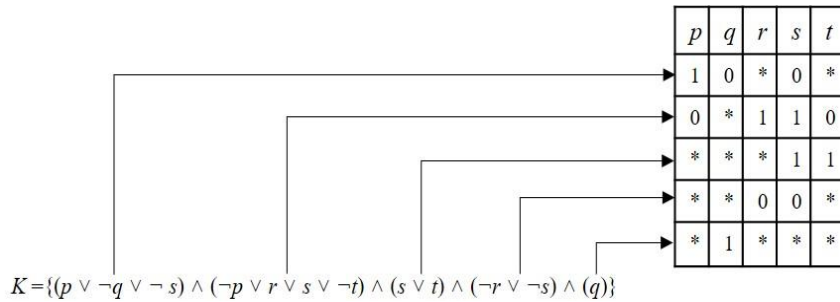


Fig. 2. Transformación de una fórmula en FNC a cadenas de 0,1 y *.

algoritmo de conteo de modelo de las técnicas de solución de SAT, como se aborda en [12].

Recientemente los modelos probabilísticos para contar modelos han tenido mucha aceptación e importancia como el algoritmo que es inspirado en una estimación estadística para refinar continuamente una estimación probabilística del conteo de modelos para una fórmula, así cada consulta XOR-racionalizada dé como resultado tanta información como sea posible [13].

Los solucionadores modernos SAT también están causando un impacto significativo en los ámbitos de la verificación de software, la resolución de las limitaciones en la inteligencia artificial, la investigación operativa, así como el diagnóstico de errores en circuitos digitales [14] y la variabilidad de los sistemas configurables [15]. Muchos otros problemas de la decisión, como los problemas de coloración de grafos, problemas de planificación y programación de problemas, puede ser codificado en SAT.

3. Algoritmo para el conteo de modelos

Algunas consideraciones básicas para el algoritmo de conteo de modelos son:

- Una fórmula F en FNC, es una conjunción de cláusulas.
- Una cláusula está formada por un conjunto de n variables.
- Una literal es una variable con valor x o su negación $\neg x$.
- Cada cláusula es mapeada a un conjunto de 0,1 y * como se muestra en la Fig. 2. Cuando una literal no aparece se coloca un asterisco (*) lo que significa que puede cualquiera de dos valores $\{0,1\}$.
- Un modelo es una asignación de valores para las literal que al realizar la conjunción de las cláusulas obtenemos un valor de verdadero (satisfactible) para la fórmula F .
- El problema $\#SAT(F)$ entonces es el total de modelos que hacen satisfactible la fórmula F .

Cuando transformamos una cláusula C en base al mismo conjunto de n variables, tenemos que el número de modelos (Mod) satisfactibles está dado por $Mod(C) = 2^n - 2^{Ast(C)}$, donde $Ast(C)$ es el número de asteriscos de la cláusula C . Por ejemplo sea $C=(1***0)$, donde $n=5$ variables, entonces $Mod(C)=2^5-2^3=32-8=24$.

Independientes		Subsumidas		Genera nuevas cláusulas	
K	0 1 * 1 *	K	* 1 * 1 *	K	0 1 * 1 *
C	0 1 * 0 1	C	0 1 * 1 1	C	0 * * * 0

Fig. 3. Operaciones básicas para el algoritmo de conteo de modelos.

Dentro del proceso general del conteo de modelos utilizamos tres operaciones básicas:

- **Clausulas independientes *Ind()***: dos cláusulas (K , C) son independientes si tienen variables contrarias para una misma posición en dichas cláusulas.
- **Clausula subsumida *Sub()***: una cláusula C es subsumida en K es subsumida si los modelos de C están contenidos en K .
- **Generación de nuevas cláusulas *Gen()***: dadas K y C dos cláusulas se generan NC cláusulas si $C_i = *$ y K_i es una literal con un valor diferente de $*$.

La Fig. 2 muestra las tres operaciones básicas para el conteo de modelos, la operación de generación de nuevas cláusulas puede generar de 1 a máximo $m-1$ cláusulas nuevas. La propuesta de algoritmo de conteo de modelos es:

Entrada:

Sea $F = \{f_1, f_2, \dots, f_m\}$ un conjunto de m cláusulas en FNC, con n =número de variables de cada cláusula

Salida:

S , conjunto de cláusulas resultantes del conteo
TotalMod, total de modelos

- 1: **Inicio**
 - 2: $TotalMod = 2^n - 2^{Ast(f_1)}$
 - 3: $S = \text{extraer}(F)$;
 - 4: **Para** cada Cláusula f_i en F , $i=2$, hasta m **Hacer**
 - 5: $C = f_i$
 - 6: **Mientras** S no este vacío **hacer**
 - 7: $in = Ind(S, C)$; //verifica independencia
 - 8: $nc = Gen(S, C)$; //obtienen # de nuevas cláusulas
 - 9: **Si** ($in == 1$ y $vacia(F)$) $S = \text{añadir}(C)$;
 - 10: **Sino Si** ($nc > 0$)
 - 11: **Para** $i \leftarrow 0$ hasta nc
 - 12: $F = \text{añadir}(nueva_clausula_i)$;
 - 13: **FinPara**
 - 14: **Sino** $S = \text{extraer}(F)$; // cláusula subsumida
 - 15: **FinSi**
 - 16: **FinSi**
 - 17: $x = \#Ast(F_i)$; // $2^{\#}$ de asteriscos de una clausula
 - 18: $TotalMod = TotalMod - x$;
 - 19: **FinMientras**
 - 20: **FinPara**
 - 21: **Fin**
-

Las operaciones de *Ind()* y *Gen()* corresponde a ciclos simples de 1 a *n* variables que permite determinar la operación que se debe aplicar, las operaciones son excluyentes para cada cláusula. Cuando se llega al caso de cláusula subsumida se detiene el ciclo para continuar con otra cláusula de la fórmula *F*.

La generación de nuevas cláusulas aumenta la revisión con las nuevas cláusulas generadas y para el caso de independencia al terminar de revisar toda la fórmula *F* se agrega a la salida *S* que representa la fórmula reducida para el conteo de modelos.

3.1 Aplicación del algoritmo

Para ejemplificar la ejecución del algoritmo consideremos el siguiente ejemplo. Sea $F = \{(p \vee \neg r) \wedge (\neg q \vee \neg r \vee s) \wedge (\neg p) \wedge (q \vee s) \wedge (p \vee \neg q \vee r \vee s)\}$ una fórmula en FNC con *m*=5 cláusulas y *n*=4 variables. Transformamos cada cláusula con el patrón de ceros, unos y asteriscos y obtenemos $F = \{(1*0*) (*001) (0***) (*1*1) (1011)\}$.

Aplicando el algoritmo inicia con:

$S = \{(1*0*)\}$; *S* representa el resultado del proceso de revisión de la fórmula almacenada en *F*.

$TotalMod = 16 - 2^2 = 16 - 4 = 12$; *TotalMod* inicia con el máximo de modelos posibles de *F* (16 modelos) menos los modelos de la primera cláusula (4).

En la Tabla 1 se muestra la ejecución del algoritmo de conteo de modelos, se observa que en el paso *i*=3 se generaron 3 nuevas cláusulas debido a que la cláusula *F*₃ tiene más asteriscos.

Podemos reducir el número de comparaciones si aplicamos un ordenamiento en base al número de asteriscos de cada cláusula. Consideremos el mismo ejemplo de la Tabla 1. $F = \{(1*0*) (*001) (0***) (*1*1) (1011)\}$, aplicando ordenamiento, obtenemos $F = \{(0***) (1*0*) (*1*1) (*001) (1011)\}$. Aplicando el algoritmo de conteo obtenemos:

$S = \{(0***)\}$; *S* almacena la primera cláusula de la fórmula *F*.

$TotalMod = 16 - 2^3 = 16 - 8 = 8$; representa el inicio del conteo.

Tabla 1. Conteo de modelos.

<i>i</i>	<i>F_i</i>	<i>S</i>						<i>x</i>	<i>TotalMod</i>	
2	*001	1*0*						0001	1	12-1=11
		0001								
3	0***	1*0*	0001					01**	4	11-7=4
		<i>Ind()</i>	01**					001*	2	
			001*					0000	1	
			0000							
4	*1*1	1*0*	0001	01**	001*	0000		0	4-0=4	
		01*1	<i>Ind()</i>	<i>Sub()</i>						
5	1011	1111	<i>Ind()</i>	<i>Ind()</i>	<i>Ind()</i>	<i>Ind()</i>	1111	1	4-1=3	
		1*0*	0001	01**	001*	0000	1111			
		<i>Ind()</i>	<i>Ind()</i>	<i>Ind()</i>	<i>Ind()</i>	<i>Ind()</i>	<i>Ind()</i>	1011	1	3-1=2

$S = \{(1*0*) (0001) (01**) (001*) (0000) (1111) (1011)\}$

El total de modelos de la fórmula $F = Mod(F) = 2$.

Table 2. Conteo de modelos aplicando ordenamiento.

i	F_i	S			x	$TotalMod$
2	$1*0*$	$0***$ $Ind()$			$1*0*$ 4	$8-4=4$
3	$*1*1$	$0***$ $11*1$	$1*0*$ 1111		1111 1	$4-1=3$
4	$*001$	$0***$ 1001	$1*0*$ $Sub()$	1111		$3-0=3$
5	1011	$0***$ $Ind()$	$1*0*$ $Ind()$	1111 $Ind()$	1011 1	$3-1=2$

$S = \{(0***) (1*0*) (1111) (1011)\}$
 El total de modelos de la fórmula $F = Mod(F) = 2$.

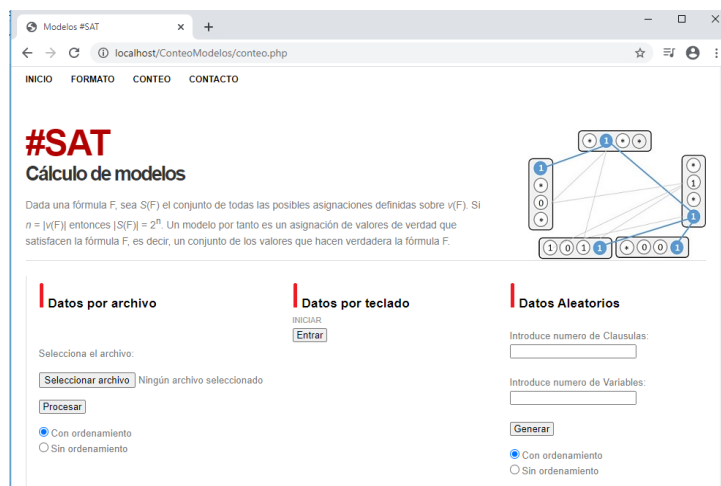


Fig. 4. Pantalla para el conteo de modelos.

La Tabla 2 muestra la ejecución del algoritmo de conteo de modelos aplicando un ordenamiento descendente de acuerdo al número de asteriscos de cada cláusula. Se reduce el número de comparaciones debido a que tomamos como cláusula inicial ($0***$) que elimina más modelos al iniciar el proceso de revisión.

4. Resultados

Como resultado del diseño del algoritmo de conteo de modelos se cuenta con un sistema en web implementado en el lenguaje de programación PHP. El sistema permite realizar pruebas del conteo de modelos a través de archivos de texto, de forma manual o generando entradas aleatorias ver Fig. 4. El sistema se ejecuta en un servidor local.

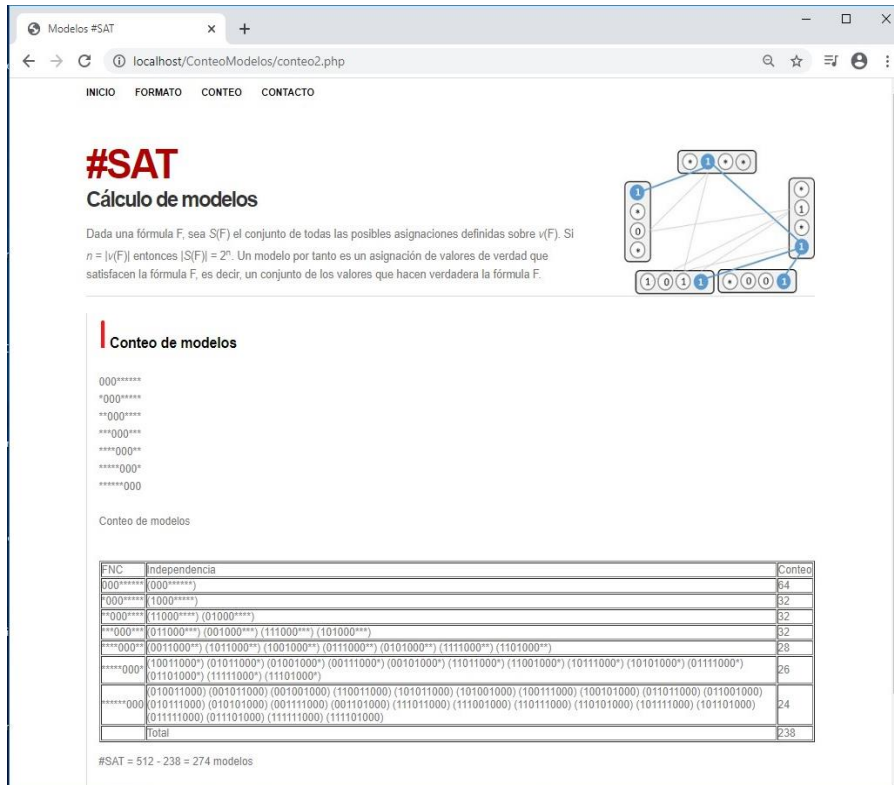


Fig. 5. Ejemplo de la salida del conteo de modelos.

Para realizar el cálculo utilizando la entrada de archivo, se debe tener un archivo de texto, donde en la primera línea debe haber el número de cláusulas y el número de variables por cláusula, posteriormente se coloca una cláusula por línea usando la representación de 0,1 y asteriscos. Por ejemplo, considere el archivo entrada9.txt con 7 cláusulas y 9 variables:

```

7 9
000*****
*000*****
**000****
***000***
****000**
*****000*
*****000
    
```

La Fig. 5 muestra el cálculo del conteo de modelos para el archivo entrada9.txt, se muestra la entrada original y en la tabla el conjunto de cláusulas que se van generando, se va reduciendo paso a paso el número de modelos. El sistema web de forma automática crea un archivo de texto (salida.txt) para guardar los resultados obtenidos de cada ejecución.



Fig. 6. Conteo de modelos del ejemplo del asistente virtual A.

Tabla 3. Ejemplos de ejecución de algoritmos para entradas aleatorias.

Cláusulas	Variables	Modelos
10	10	1024 - 128 = 896
20	10	1024 - 231 = 793
50	20	1048576 - 11741 = 1036835
100	20	1048576 - 20243 = 1028333
200	30	1073741824 - 1065208 = 1072676616
300	40	1099511627776 - 41099096 = 1099470528680

En la Tabla 3 se describe varios ejemplos de ejecución del algoritmo de conteo de modelos en forma aleatoria. Para estos ejemplos el tiempo de ejecución no es significativo ya que se utilizó en promedio 1.5 segundos.

En el área de razonamiento automático se modelan problemas con lógica proposicional, por ejemplo, dado el siguiente conjunto de proposiciones de un asistente virtual A:

- p : comprar pañales
- q : pañales de la marca
- r : marca blanca
- s : cantidad requerida
- t : paquete individual

Formamos el conjunto de instrucciones del asistente virtual, $A = \{\text{comprar pañales, no hay pañales o no hay pañales de la marca o hay marca blanca, no hay la cantidad requerida o hay pañales de la marca, no hay paquete individual o hay pañales de marca, hay paquete individual}\}$, transformando A en su FNC, tenemos que $A = \{(p) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg s \vee q) \wedge (\neg t \vee q) (t)\}$, convertimos A en su patrón correspondiente de 0,1 y *, obteniendo $A = \{(1****) (001**) (*1*0*) (*1**0) (****1)\}$. La Fig. 6 muestra el conteo de modelos de la formula A utilizando la entrada por teclado.

5. Conclusiones

Se presenta una propuesta algorítmica para el conteo de modelos de una fórmula en FNC utilizando una representación de cadenas de 0,1 y *'s. Esta representación nos permite mejorar el proceso de conteo de modelos de manera algorítmica.

Con el objetivo de proveer un mecanismo de pruebas, se generó una versión en web del algoritmo que permite hacer cálculos con datos de teclado, de archivo o de forma aleatoria.

La complejidad en tiempo del algoritmo viene dada por el número de m cláusulas y el número de n variables de cada cláusula, se procesa cada cláusula una a una y se generan a lo más $n-1$ nuevas cláusulas que también son revisadas para generar independencia, en el peor caso se procesan $m*n*(n-1, n-2, n-3, \dots, 1)$ operaciones, debido a que una cláusula puede tener $n-1$ asteriscos (para cláusulas unitarias), y cada una de estas nuevas cláusulas pueden generar $n-2$ nuevas cláusulas a procesar. Al implementar el sistema utilizamos 3 ciclos anidados de orden $m*n$. La complejidad algorítmica en general del problema de satisfactibilidad para fórmulas a partir de 3 variables es exponencial, sin embargo, nuestro algoritmo, aunque sigue siendo exponencial puede ser acotado por un parámetro en términos del número de cláusulas y el número de variables de cada cláusulas.

Este tipo de algoritmos de conteo de modelos puede ser utilizado en el área de razonamiento automático, en bases de conocimiento en la revisión de creencias y otras aplicaciones de la inteligencia artificial.

Referencias

1. De-Ita, G., Gómez, H., Merino, B.: Algorithm to count the number of signed paths in an electrical network via boolean formulas. *Acta Universitaria*, 22, pp. 69–74, Universidad de Guanajuato (2012)
2. Garey, M.R., Johnson, D.S.: *Computers and intractability: A Guide to the theory of NP-Completeness*, W.H. Freeman and Co., New York (1979)
3. Papadimitriou, C.H.: *Computational complexity*. Addison Wesley (1994)
4. Guillen, C., López, A., De-Ita G.: Diseño de algoritmos combinatorios para #SAT y su aplicación al razonamiento proposicional, Reporte Técnico No. CCC-05-005 (2005)
5. Roth, D.: On the hardness of approximate reasoning, *Artificial Intelligence*, pp.273–302 (1996)
6. Russ, B.: *Randomized algorithms: Approximation, generation, and counting*. Springer-Verlag London Berlin Heidelberg (2001)
7. Samer, M., Szeider, S.: *Algorithms for propositional model counting*. Elsevier (2009)
8. Gomes, C., Sabharwal, A., Selman, B.: *Model counting, handbook of satisfiability*. 20, IOS Press (2008)
9. Fremont, D., Rabe, M., Seshia, S.: Maximum model counting. In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3885–3892 (2017)
10. Sharma, S., Roy, S., Soos, M., Meel, K.S.: GANAK: A scalable probabilistic exact model counter. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, pp. 1169–1179 (2019)

11. Silva, F., Neves, J., Dias, S., Zárate, L., Song, M.: A hybrid approach to solve SAT and UNSAT problems. *IEEE Latin America Transactions*, 18(4) (2020)
12. Oztok, U., Darwiche, A.: An exhaustive DPLL algorithm for model counting. *Journal of Artificial Intelligence Research*, 62, pp 1–32 (2018)
13. Kim, S., McCamant, S.: Bit-vector model counting using statistical estimation, tools and algorithms for the construction and analysis of systems. 10805, pp. 133–151 (2018)
14. Gaber, L., Hussein, A. Mahmoud, H. Mourad. M.: Mohammed moness computation of minimal unsatisfiable subformulas for SAT-based digital circuit error diagnosis. *Journal of Ambient Intelligence and Humanized Computing Springer-Verlag* (2020)
15. Sundermann, C., Thüm, T., Schaefer, I.: Evaluating #SAT solvers on industrial feature models. In: *ACM International Conference Proceeding Series*, a3 (2020)

Evaluación del desempeño de sensores infrarrojo, ultrasónico y visión para el conteo de vehículos

Juan José Lozano Barrón, Osslán Osiris Vergara Villegas,
Salvador Noriega Morales, Vianey Guadalupe Cruz Sánchez

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

al194485@alumnos.uacj.mx,
{overgara, snoriega, vianey.cruz}@uacj.mx

Resumen. El tráfico vehicular es un problema social típico de las grandes urbes y su solución no es fácil, aunque si es necesaria. Una forma de comenzar a plantear soluciones consiste en realizar un análisis de tráfico en las zonas donde se presentan las congestiones. Una tarea importante de dicho análisis consiste en el conteo de los vehículos que circulan en determinados horarios. En el presente artículo se aborda una propuesta para desarrollar un prototipo para simular diferentes condiciones de tráfico vehicular en una avenida con tres carriles. Primero, se explica la conceptualización del prototipo y se ofrecen detalles sobre su construcción. Después, se presentan los algoritmos para el conteo de vehículos por medio de un sensor infrarrojo, un ultrasónico, visión y sus combinaciones. Posteriormente, se ofrecen explicaciones de las tres pruebas realizadas para evaluar el desempeño de cada uno de los sensores en la tarea de conteo de tres, cinco y nueve vehículos, para finalmente realizar una recomendación de uso. Después de las pruebas con tres y cinco vehículos el mejor desempeño se obtuvo con visión con una precisión de 100% y 96.66%, respectivamente. Para la prueba con nueve vehículos la mejor precisión de 80% se obtuvo con la combinación de sensores infrarrojo y visión.

Palabras clave: Conteo de Vehículos, Sensor Infrarrojo, Sensor Ultrasónico, Sensor de Visión, Simulación de Tráfico.

Performance Evaluation of Infrared, Ultrasonic, and Vision Sensors for Vehicle Counting

Abstract. Vehicle traffic is a typical social problem in large cities, and its solution is not easy, although it is necessary. One way to start proposing solutions is to carry out a traffic analysis in the areas where congestions occur. An essential task of this analysis consists of counting the vehicles that circulate at certain times. In this paper, we present a proposal to develop a prototype to simulate different vehicular traffic conditions on a three-lane avenue. First, we offer the conceptualization of the prototype and the details on its construction. Next, we present the algorithms for vehicle counting using sensors such as

infrared, ultrasonic, vision, and their combinations. Then, explanations of the three tests conducted to evaluate the performance of each sensor in the task of counting three, five, and nine vehicles are offered, to make finally a recommendation for use. After testing with three and five vehicles, the best performance was obtained with a vision sensor with an accuracy of 100% and 96.66%, respectively. For the test with nine vehicles, the accuracy of 80% was obtained with the combination of infrared and vision.

Keywords: Vehicle Counting, Infrared Sensor, Ultrasonic Sensor, Vision Sensor, Traffic Simulation.

1. Introducción

Debido a la reducción de precios y a la alta demanda, la industria de producción automotriz proyecta que la congestión vehicular aumentará año con año, lo que constituye un peligro para la calidad de vida urbana [1]. Efectivamente, en casi todo el mundo la congestión vehicular ha ido en aumento debido al crecimiento de la población y su necesidad de desplazarse a diferentes lugares. Debido al tráfico, a una persona puede tomarle un tiempo considerable el trasladarse con su vehículo de un punto a otro, aun cuando la distancia sea relativamente corta [2].

El tráfico vehicular conlleva a diversas implicaciones físicas y sociales, por ejemplo, aumenta la probabilidad de accidentes, genera problemas económicos, de contaminación, de salud, entre otros, por lo que vale la pena realizar esfuerzos para reducirlo [3]. La construcción de caminos y autopistas de alta capacidad puede ser una solución para la reducción de tráfico, sin embargo, es muy costosa y no es viable en ciudades que cuentan con poco espacio [4]. Otra alternativa consiste en utilizar tecnologías como satélites, cámaras, y redes de sensores, para obtener datos y analizarlos con el objetivo de informar a los conductores sobre las condiciones de un camino y desviarlos o redirigirlos en tiempo real hacia vías menos congestionadas [5].

Una de las tareas esenciales para proponer estrategias de reducción de tráfico consiste en la vigilancia y análisis de la cantidad de vehículos que circulan por un camino en un horario en particular [6]. Para realizar la vigilancia se deben detectar los vehículos y además contarlos para posteriormente analizar los datos obtenidos y proponer estrategias de mitigación [7].

El conteo de vehículos se puede realizar de manera manual o automática. El conteo manual es realizado de forma visual por una o varias personas especializadas. Desafortunadamente, suele ser costoso e impreciso debido a condiciones como el cansancio, distracciones y errores en la captura de información. Por otro lado, el conteo automático consiste en recopilar la información durante largas jornadas por medio de un equipo especializado. Los sensores son los equipos que se utilizan para los conteos, y la información adquirida es transferida a un sistema para su almacenamiento y procesamiento [8]. Los sensores que frecuentemente se utilizan para resolver la tarea de conteo de vehículos son: lazo inductivo, flujo magnético, radar de microondas, ultrasónico, radar láser, lidar, infrarrojo, mangueras neumáticas, cámaras digitales, y combinaciones de diferentes tecnologías [9]. De acuerdo con la literatura, los sensores magnéticos, ultrasónicos, infrarrojos y de microondas son los más utilizados para el conteo de vehículos, sin embargo, tienen limitaciones cuando

se quiere obtener información detallada [10]. En los últimos años, las cámaras digitales se han utilizado para la detección y conteo de vehículos, sin embargo, presentan la desventaja de necesitar un gran poder de cómputo para extraer la información necesaria [11].

El procedimiento general que se sigue para el conteo es el de instalar los instrumentos de captura en un lugar en particular y comenzar la adquisición de los datos. Sin embargo, se deben realizar varias pruebas antes de definir el lugar correcto para colocar los sensores, por lo que se puede recurrir a simulaciones para comenzar a detectar cómo sobrellevar las condiciones del entorno [12].

En el presente artículo se describen los detalles para la construcción de una plataforma para simular las condiciones de tráfico vehicular en una avenida con tres carriles. Posteriormente, se plantea un algoritmo para el conteo de vehículos por medio de tres sensores: visión, ultrasónico e infrarrojo y sus respectivas fusiones.

La gran ventaja de contar con un prototipo, a diferencia de lo que sucede con las pruebas en un ambiente real, es que se reduce la incertidumbre sobre la implementación de software, se obtiene un mayor conocimiento del sistema con una curva de aprendizaje menor, y se puede entender mejor el problema antes de implementarlo a la realidad. En la plataforma propuesta se pueden controlar la ubicación y el número de vehículos para observar cuál sensor tiene el mejor desempeño en la tarea de conteo. Además, se puede controlar la iluminación, se pueden agregar otros sensores y probar otros algoritmos de conteo. Las principales contribuciones del presente artículo son:

1. Se describen los detalles para la construcción de un prototipo para simular las condiciones de un camino.
2. Se desarrolla un algoritmo para el conteo de vehículos con tres tipos de sensores y sus respectivas combinaciones.
3. Se presenta una evaluación del desempeño de tres tipos de sensores para el conteo de vehículos.
4. Se ofrece una recomendación del tipo de sensor a utilizar correspondiente al tamaño del flujo vehicular.

El resto del artículo se encuentra organizado de la siguiente manera. En la sección 2 se presenta la explicación de los materiales utilizados para la construcción del prototipo y las estrategias para el conteo de vehículos. Los experimentos realizados y los resultados obtenidos se muestran en la sección 3. Finalmente, en la sección 4 se muestran las conclusiones obtenidas del proceso de investigación.

2. Materiales y métodos

La metodología del presente artículo se compone principalmente de dos etapas, en la primera, se explican los detalles para la construcción del prototipo y en la segunda se implementan los algoritmos para el conteo de vehículos.

2.1. Desarrollo del prototipo

Para diseñar la arquitectura del prototipo se utilizó el software SolidWorks y la base de su funcionamiento es el de una banda transportadora. El objetivo es, que los autos se puedan colocar en tres diferentes carriles con diferentes configuraciones. Después, se definieron los componentes para la construcción, divididos en la parte mecánica y la parte electrónica.

Para construir la parte mecánica se utilizó aluminio extruido de 40mm x 40mm. Para la fabricación de la base del motor que moverá las bandas y el semáforo se empleó delrin color negro. Además, se utilizó una placa de ABS negro, seis baleros y tres bandas para simular los carriles, mientras que, para los ejes del mecanismo se utilizó acero A2. Las dimensiones del prototipo son 35.56 cm de largo, 11.05 cm de ancho y 11.05cm de alto sin tomar en cuenta el semáforo y soporte de los sensores. El semáforo tiene una altura de 10.79 cm con respecto a la base del prototipo. En la Fig. 1a, se muestra la versión final del prototipo en 3D.

Para la colocación de los autos se utilizaron nueve imanes de neodimio de 8mm x 8mm x 12mm, que se pueden colocar en diferentes posiciones. También, se colocó una estructura de aluminio tipo soporte universal para posicionar la cámara, que consta de dos varillas de aluminio, una de 1.27 cm de diámetro y 26 cm de largo, la segunda tiene el mismo diámetro y su longitud es de 20 cm, y dos adaptadores para el ajuste de la posición y ángulo de la cámara. La cámara utilizada es una logitech c270.

El control electrónico del mecanismo se realizó con un microcontrolador Arduino uno, junto con un relevador para activar/desactivar la banda transportadora. Además, se utilizó un motor tipo “D” que trabaja a 12 VCD dando 62 RPM, con un torque de 4.9 kgf/cm y un regulador de velocidad. En el semáforo se colocó un sensor ultrasónico HC-SR04, y tres sensores infrarrojos E18-D80NK. En la Fig. 1b, se muestra una imagen real del prototipo final.

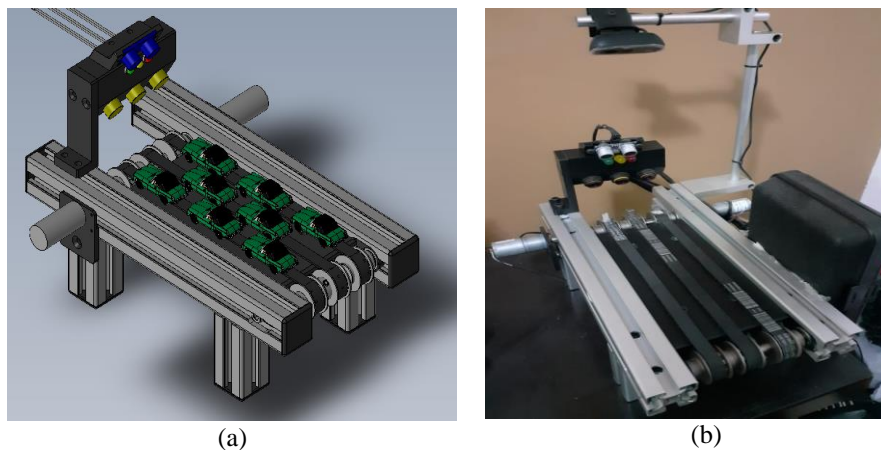


Fig. 1. Prototipo para simulación de tráfico. (a) 3D, y (b) real.

2.2. Algoritmos para el conteo de vehículos

Después de la construcción del prototipo se realizó la implementación de los algoritmos de conteo con los diferentes sensores mencionados en la subsección 2.1. Además, se desarrolló una interfaz gráfica de usuario (GUI, por sus siglas en inglés) para seleccionar los algoritmos que el usuario quiera probar como se muestra en la Fig. 2. Las combinaciones que se pueden seleccionar son: sensor ultrasónico, sensor infrarrojo, visión, infrarrojo + visión, ultrasónico + visión e infrarrojo + ultrasónico visión.

Los sensores infrarrojos se colocaron a una distancia de 6.7 cm a un ángulo de 60° con respecto a la base del prototipo. Los sensores infrarrojos son dispositivos electrónicos que miden y detectan la radiación de los objetos dentro de su campo de visión, por lo que al momento en que se interrumpe dicha radiación se logra la detección del objeto, en este caso un vehículo [13]. Para la implementación, se agregó en el algoritmo una secuencia para contar una sola vez cuando se detecta el objeto, ya que si no se podría estar contando en repetidas ocasiones el mismo objeto.

Un sensor ultrasónico es una tecnología relativamente barata que se utiliza para medir la distancia hacia un objeto mediante el uso de ondas [14]. Los sensores ultrasónicos transmiten ondas de energía en una frecuencia entre 25 y 50 Khz. Para su funcionamiento el cabezal del sensor emite una onda ultrasónica y recibe la onda reflejada por el objeto. La distancia se calcula al medir el tiempo que pasa entre la emisión y la recepción de la onda ultrasónica. El sensor fue instalado a 11.04 cm con respecto a la base del prototipo y con un ángulo de 30°. Para realizar el conteo en tiempo real se obtiene un valor análogo del sensor y se realiza su respectiva interpretación. Además, se estableció una condición para discernir cuando se sensa un objeto de cuando se sensa el carril.

Por otra parte, los sensores de visión capturan imágenes para analizarlas mediante un software. Para detectar un vehículo dentro de una escena primero se debe segmentar, es decir, saber dónde está localizado y el área que ocupa, y posteriormente se puede realizar el conteo [15]. El sensor de visión se colocó a 27 cm de distancia con respecto a la base del prototipo. Debido a que el sensor de visión se encuentra colocado en una posición fija conocida, y dado que las condiciones de iluminación no fueron controladas, el algoritmo de conteo representó un reto significativo.

Primero, se toma una fotografía de la escena que contiene los vehículos. Después la imagen obtenida en RGB (Red, Green, Blue por sus siglas en inglés) es transformada a su representación en escala de gris. La conversión es realizada debido a que el color de un vehículo no fue tomado en cuenta para el conteo. Para realizar la conversión se separó cada uno de los planos, por lo que se obtienen tres imágenes en escala de gris.

Luego se procede a calcular el valor del umbral global por medio del método de Otsu. El algoritmo de Otsu permite obtener de manera automática un umbral que produzca la mejor separación entre clases mediante la maximización de la varianza. El umbral óptimo k^* que maximiza la varianza entre clases σ_B^2 se calcula mediante la ecuación 1:

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]}, \quad (1)$$

donde μ_T es la media total de la imagen original, $\omega(k)$ es la probabilidad de ocurrencia de una clase, y $\mu(k)$ es la media de la clase. Los resultados obtenidos fueron 0.1247 para la matriz roja, 0.0706 para la matriz verde y 0.0353 para la matriz azul. Después, cada imagen es binarizada con respecto al umbral calculado, y finalmente las tres imágenes resultantes son sumadas para obtener la imagen binarizada total.

Como resultado de la binarización se logra desaparecer el fondo de las imágenes. Posteriormente, se realiza un proceso de definición de regiones de interés (ROI). La ROI es definida de manera automática, debido a que se conoce dónde se encuentra colocado el sensor de visión, por lo que el recorte de la imagen que define la ROI, y que siempre permite conservar la zona donde los vehículos están colocados, se realiza siempre en el mismo lugar. Posteriormente, se ensacharon las líneas en la imagen para poder tener una mejor visualización de los vehículos. Lo anterior, se realizó por medio de una operación de dilatación, que empleó una línea vertical como elemento estructural, como se muestra en la ecuación 2:

$$A \oplus B = \left[(\hat{B})_z \cap A \right] \subseteq A, \quad (2)$$

donde A y B son conjuntos en Z^2 . Se comienza con la reflexión de B desde su origen desplazado por z , entonces, la dilatación se obtiene del conjunto de todos los desplazamientos z tal que \hat{B} y A se superponen en al menos un elemento.

Después, se utilizó un algoritmo de etiquetado que utiliza las componentes conectadas de un objeto (8 vecino). Se comienza con la definición de un píxel de inicio y se establece la primera etiqueta en 1. Si el píxel es parte del objeto y no ha sido etiquetado, entonces se le asigna la etiqueta actual y se agrega como primer elemento de la cola. Si el píxel es parte del fondo o ya fue etiquetado se revisa el siguiente píxel. Se revisa el píxel de la cola y se revisan sus vecinos, si el vecino no es parte del fondo, y no se ha etiquetado, se le asigna la etiqueta actual y se añade a la cola. El proceso se repite hasta que no hay elementos en la cola. Finalmente, se revisa el siguiente píxel y se incrementa la etiqueta en 1. Como resultado se pudo determinar si un píxel pertenece al mismo objeto o no, y así poder discernir entre el espacio ocupado por un vehículo y por otro, con lo que concluye la etapa de localización del vehículo. Después, se calculan los centroides de cada objeto detectado y a partir de allí se dibuja una caja (bounding box) para encerrar el objeto. Finalmente, se calcula el área ocupada por el objeto dentro de la caja para determinar si es un vehículo, y a partir de esa región se realiza el conteo final.

3. Experimentación y resultados

Para comenzar la experimentación, se realizaron pruebas preliminares con uno y dos vehículos, para observar si efectivamente se realizaba el conteo. Para el caso del sensor de visión, se introdujeron objetos más pequeños y más grandes que los vehículos para asegurar que no eran contados. Posteriormente, se realizaron las pruebas finales.

En total se realizaron tres pruebas diferentes para observar la robustez de los algoritmos implementados.

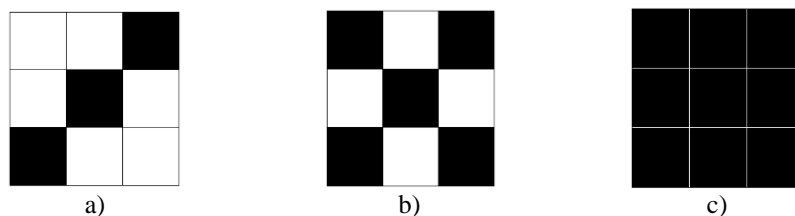


Fig. 3. Configuración de vehículos para las pruebas. (a) tres vehículos, (b) cinco vehículos y (c) nueve vehículos.

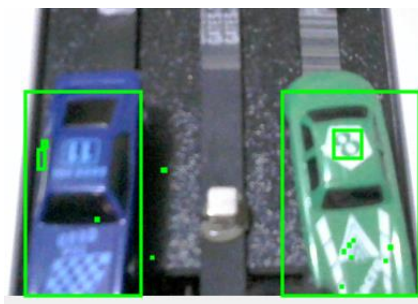


Fig. 4. Ejemplo del conteo de vehículos con el sensor de visión.

Tabla 1. Resultados obtenidos para las pruebas de conteo de vehículos.

No. de vehículos	IR	US	VIS	IR+VIS	US+VIS	IR+US+VIS
3	93.33%	0%	100%	93.3%	13.33%	20%
5	90%	10%	96.66%	93.33	3.33%	6.66%
9	0%	0%	60%	80%	23.3%	66.6%

Para la experimentación se definió una velocidad constante del mecanismo de 0.234 km/h. Además, para cada prueba se utilizaron tres (uno por carril), cinco (dos en los laterales y uno al centro), y nueve (tres en cada carril) autos con diferentes configuraciones como se puede observar en la Fig. 3.

Cada una de las pruebas consiste en tres vueltas de los vehículos pasando por enfrente de los sensores, además, cada prueba se realizó 30 veces y cada vez se contabilizaron los aciertos y errores. Debido a motivos de espacio, en el presente artículo no se muestran todos los datos obtenidos, sin embargo, se encuentran disponibles si se solicitan a los autores. En la Fig. 4 se muestra un ejemplo del conteo de vehículos con visión. Se debe observar que, aunque se detectan diferentes objetos, solamente se cuentan los que corresponden al área de un vehículo.

Como se puede observar en la figura 4 se detectan algunas pequeñas áreas que deben ser consideradas como errores debido a que no representan un vehículo. Dichas áreas son detectadas como parte de la región de interés en la búsqueda de píxeles contiguos en la vecindad 8. Sin embargo, fueron descartadas debido a sus tamaños ya que en el algoritmo cualquier área que es igual a uno es descartada y solamente se conservan las áreas correspondientes a los vehículos.

En la Tabla 1 se muestra un resumen de los resultados obtenidos. Es importante mencionar que se utilizaron las abreviaciones IR para infrarrojo, US para ultrasónico y VIS para visión. Los resultados de la Tabla 1 fueron obtenidos después de varias pruebas exhaustivas, aunque por motivos de espacio no se muestran los datos completos (resultados de cada corrida), los presentados explican el comportamiento del sistema ante situaciones particulares. Como se puede observar, para el caso con tres vehículos, cuando interviene el sensor US los resultados no son favorables, debido a que no se logró contar ningún vehículo. Sin embargo, se puede observar que usando sólo visión se obtiene el mejor conteo, seguido por el sensor IR y la integración IR+VIS para los cuales sólo se produjeron dos fallas. Vale la pena destacar que como resultado del conteo se pueden obtener tres opciones, una que se cuenten todos los vehículos, otras en el que se cuenten más, y otra en la que se cuenten menos.

Para el caso del conteo de cinco vehículos se puede observar que nuevamente es con el sensor US con el que se obtienen los resultados más desfavorables. Además, con el sensor de visión se obtienen los mejores resultados, solamente se comete una falla, seguido por el sensor IR.

La prueba de conteo con nueve vehículos es la más difícil de las tres debido a que los vehículos están muy juntos. Se debe tomar en cuenta que el algoritmo debe realizar la localización y determinar correctamente la separación de los vehículos, de lo contrario se podría confundir y determinar que solamente hay un objeto. En la Tabla 1, se puede observar que los sensores IR y US no pudieron contar ningún vehículo. Mientras que, el mejor resultado se obtiene con la combinación de IR + VIS seguido de IR + US + VIS.

3.1. Discusión

Con las pruebas presentadas se determinó que el prototipo puede ser utilizado efectivamente para simular tráfico en un ambiente controlado. Gracias al uso del *Arduino uno*, es posible agregar más sensores a la plataforma. Otra de las ventajas es que la plataforma se puede programar en distintos lenguajes o intérpretes de código, con lo cual se vuelve un prototipo flexible en cuestiones de programación.

En lo que se refiere a la tarea de conteo de vehículos se observó que diversos factores pueden afectarlo o beneficiarlo. En general, el sensor de visión es el que ofrece mejores resultados para la prueba de conteo con tres y cinco vehículos. Debe destacarse, que el sensor ultrasónico es el que presenta el peor desempeño para las tres pruebas. Además, se observa que el uso del sensor infrarrojo combinado con el sensor de visión ofrece el mejor resultado en la prueba con nueve vehículos, en donde se observó que el resto de los métodos no fueron tan efectivos.

La prueba con nueve vehículos es importante porque permite simular un escenario con congestión vehicular en la que, por la cercanía, incluso, se podría tener una situación de oclusión de algunos de los vehículos.

Una de las ventajas de crear un prototipo además de la simulación es la versatilidad que ofrece para poder agregar sensores y para futuros trabajos y pruebas se sugiere agregar sensores como lidar, radares y tal vez neumáticos.

El prototipo se puede utilizar para realizar el conteo de vehículos en un entorno controlado, además, se puede explotar de manera didáctica para que los estudiantes

puedan experimentar con más sensores o métodos de procesamiento digital de imágenes. El prototipo es funcional y gracias a la plataforma usada para el microcontrolador se puede programar en distintas interfaces de desarrollo.

Como resultado de las pruebas se recomienda que para un conteo de tráfico en un escenario real se utilice una combinación de un sensor infrarrojo y un sensor de visión. La fusión de la información de los dos sensores permite compensar los errores cometidos y generar un conteo robusto.

Es importante resaltar que, los resultados son específicos para cada sensor con el fin de aumentar las probabilidades de que las pruebas en campo den mejores resultados y no tener que realizar todas las pruebas directamente en un ambiente real de tráfico vehicular.

4. Conclusiones

En el presente artículo se mostraron los detalles para construir un prototipo para simular diferentes condiciones de tráfico vehicular en una avenida con tres carriles. El prototipo fue utilizado para observar y comparar el desempeño de los sensores infrarrojo, ultrasónico y visión en la tarea de conteo de vehículos.

Una de las grandes ventajas del prototipo es que se tiene la capacidad de correr pruebas en cualquier momento, y de simular distintos escenarios de congestión vehicular. Como resultado del experimento, se observó que en la prueba con nueve vehículos que simula congestión vehicular, lo mejor es utilizar una fusión de los sensores infrarrojo y visión.

Para trabajos futuros se recomienda evaluar el prototipo a diferentes velocidades y además trabajar con visión nocturna, ya que el prototipo fue evaluado siempre con luz natural. Además, se realizarán pruebas de los algoritmos propuestos en escenarios de tráfico reales. Finalmente, sería recomendable adaptar la plataforma para incluir más sensores como los lidar y radar, y después desarrollar sus respectivos algoritmos para conteo.

Referencias

1. Zheng, H., Chang, W., Wu, J.: Traffic flow monitoring systems in smart cities: Coverage and distinguishability among vehicles. *Journal of Parallel and Distributed Computing*, 127, pp. 224–237 (2019)
2. Li, Z., Huang, J.: How to mitigate traffic congestion based on improved ant colony algorithm: A case study of a congested old area of a metropolis. *Sustainability*, 11(1140), pp. 1–15 (2019)
3. Wen, L., Kenworthy, J., Guo, X., Marinova, D.: Solving traffic congestion through street renaissance: A perspective from dense asian cities. *Urban Science*, 3(18), pp. 1–21 (2019)
4. Reza, M., Zarrabi, H., Hafeez, R., Shamshirband, S., Raymond, K.: Applications of computational intelligence in vehicle traffic congestion problem: A Survey *Soft Computing*, 22, pp. 2299–2320 (2018)
5. Meneguette, R., Filho, G., Guidoni, D., Pessin, G., Villas, L., Ueyama, J.: Increasing intelligence in inter-vehicle communications to reduce traffic congestions: Experiments in urban and highway environments, *Plos One*, 11(8), pp. 1–25 (2016)

6. Shen, Z., Wang, W., Shen, Q., Zhu, S., Fardoun, H., Lou, J.: A novel learning method for multi-intersections aware traffic flow forecasting. *Neurocomputing*, 398, pp. 477–484 (2020)
7. Walid, B., Refai, H.: Intelligent vehicle counting and classification sensor for real-time traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 19(6), pp. 1784–1794 (2018)
8. Rosas-Arias, L., Portillo-Portillo, J., Hernandez-Suarez, A., Olivares-Mercado, J., Sanchez-Perez, G., Toscano-Medina, K., Perez-Meana, H., Sandoval, A., García, L.: Vehicle counting in video sequences: An incremental subspace learning approach. *Sensors*, 19, pp. 1–16 (2019)
9. Taghvaeeyan, S., Rajamani, R.: Portable roadside sensors for vehicle counting, classification and speed measurement. *IEEE Transactions on Intelligent Transportation Systems*, 15(1), pp. 73–83 (2014)
10. Dai, Z., Song, H., Wang, X., Fang, Y., Yun, X., Zhang, Z., Li, H.: Video-based vehicle counting framework. *IEEE Access*, 7, pp. 64460–64470 (2019)
11. Yang, H., Qu, S.: Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition. *IET Intelligent Transport Systems*, 12, pp. 75–85 (2017)
12. Xia, Y., Shi, X., Song, G., Geng, Q., Liu, Y.: Towards improving quality of video-based vehicle counting method for traffic flow estimation. *Signal Processing*, 120, pp. 672–681 (2016)
13. Ai, H., Zheng, Y.: Characterization of a traffic management system using pyroelectric infrared sensors. *Instrumentation Science & Technology*, 43, pp. 319–333 (2015)
14. Liu, J., Han, J., Lv, H., Li, B.: An ultrasonic sensor system based on a two-dimensional state method for highway vehicle violation detection applications. *Sensors*, 15, pp. 9000–9021 (2015)
15. Rabbouch, H., Saâdaoui, F., Mraïhi, R.: Unsupervised video summarization using cluster analysis for automatic vehicles counting and recognizing. *Neurocomputing*, 260, pp. 157–173 (2017)

Sobre la algebrización de la lógica paraconsistente CG^3

Miguel Pérez Gaspar, Everardo Bárcenas

Universidad Nacional Autónoma de México,
Facultad de Ingeniería,
México

miguel.perez@fib.unam.mx, ebarcenas@unam.mx

Resumen. Las lógicas paraconsistentes son una familia de lenguajes formales con variadas aplicaciones en las ciencias computacionales, particularmente, en el campo de la inteligencia artificial, las lógicas paraconsistentes han sido exitosamente aplicadas en programación lógica, razonamiento difuso, y hasta en la construcción de redes neuronales paraconsistentes. G^3 es una lógica 3-valuada con un único valor de verdad representado por 1. CG^3 es una lógica paraconsistente, 3-valuada que extiende a G^3 con dos valores de verdad representados por 1 y 2. El estado del arte de CG^3 comprende una semántica de tipo Kripke y una axiomatización de tipo Hilbert inspirada en la técnica Lindenbaum-Los. En este trabajo, se demuestra que existe una algebrización de la lógica CG^3 en el sentido de Blok y Pigozzi. Este resultado fundamenta el desarrollo de sistemas de razonamiento paraconsistentes.

Keywords: Lógicas Paraconsistentes, Algebrización Blok-Pigozzi, Lógica CG^3 .

On the algebrization of paraconscious logic CG^3

Abstract. Paraconsistent logics are a family of formal languages with varied applications in computer science, particularly in the field of artificial intelligence, paraconsistent logics have been successfully applied in programming, fuzzy reasoning, and even in the construction of paraconsistent neural networks. G^3 is a 3-valued logic with a single truth value represented by 1. CG^3 is a paraconsistent, 3-valued logic that extends G^3 with two truth values represented by 1 and 2. The state of the CG^3 art comprises a Kripke-type semantics and a Hilbert-type axiomatization inspired by the Lindenbaum-Los technique. In this work, it is shown that there is an algebrization of the CG^3 logic in the sense of Blok and Pigozzi. This result supports the development of paraconsistent reasoning systems.

Keywords: Paraconsistent Logics, Blok-Pigozzi Algebrization, CG^3 Logic.

1. Introducción

Es bien conocido el origen filosófico, y su aplicación en los fundamentos de las matemáticas, de los lenguajes lógicos, o simplemente lógicas. En el campo de las ciencias computacionales, diversas lógicas son utilizadas en la especificación de lenguajes de programación, esto es, los programas pueden ser caracterizados como demostraciones en sistemas de inferencia lógicos (isomorfismo Curry-Howard) [10]. Esto hace posible la participación de variadas lógicas en la Inteligencia Artificial (IA): desde usos en el proceso de implementación con información analítica, hasta en la demostración de correctitud de algoritmos. Existen incluso casos en los que ciertas teorías lógicas han servido para el desarrollo de paradigmas de programación, tales como la programación lógica [23].

Las lógicas paraconsistentes forman una familia de lenguajes diseñados para el análisis y razonamiento a partir de inconsistencias desde el punto de vista de la lógica clásica), como a menudo es requerido en diversas problemáticas en la IA, tales como el procesamiento de señales e imágenes y sistemas expertos [25]. Dentro de las familias de lógicas paraconsistentes, las lógicas anotadas, que abarcan la teoría de conjuntos difusos, son las que más ampliamente han sido aplicadas en la IA [2]. Otro campo de aplicación de las lógicas paraconsistentes es el de razonamiento no monótono, concepto fundamental en el desarrollo de sistemas inteligentes. En [3,4] se puede encontrar una semántica estándar para el razonamiento no monótono para lógicas anotadas y programas lógicos anotados.

Las lógicas multivaluadas son lógicas no clásicas [5]. Al igual que en la lógica clásica, las lógicas multivaluadas también disfrutan del principio de funcionalidad de verdad, a saber, el valor de verdad de una oración compuesta es determinada a través de los valores de verdad de sus oraciones componentes, y permanece igual cuando una de sus oraciones componentes se reemplaza por otra oración con el mismo valor de verdad. Sin embargo, en contraste con el caso clásico, las lógicas multivaluadas no restringen el número de valores de verdad a solo dos, un conjunto más grande de grados de verdad es la característica distintiva en el contexto de muchos valores.

En [14], podemos encontrar un resumen detallado de lógicas multivaluadas. Algunos sistemas de lógicas multivaluados se presentan como familias de sistemas de valores finitos e infinitos uniformemente definidos, por ejemplo, la lógica de Lukasiewicz, la lógica de Gödel, los sistemas basados en la norma t , los sistemas 3-valuados, el sistema de 4 valores de Dunn-Belnap, Sistemas de productos. Los principales tipos de cálculo lógico para sistemas de lógicas multivaluadas son el cálculo de tipo Hilbert, el cálculo de secuentes de tipo Gentzen o Tableaux [14]. Una amplia clase de lógicas infinitamente valoradas presentadas por [24].

La lógica clásica, así como la lógica intuicionista sufren un inconveniente a la hora de razonar con información inconsistente. Según el principio de explosión, también conocido como “*ex contradictione sequitur quodlibet*”, toda teoría o base de conocimiento inconsistente es totalmente trivial, lo que hace que estas lógicas sean inútiles para razonar con inconsistencias. Como resultado, las alternativas a la lógica clásica que no tienen este inconveniente tienen evolucionado, llamados enfoques “paraconsistentes”. En 1954 F. Asenjo, en su disertación doctoral, propone por primera vez utilizar lógicas multivaluadas para generar lógicas paraconsistentes (lógicas cuya relación de consecuencia lógica semántica o teoría de prueba no es explosiva [21]).

El enfoque de muchos valores es abandonar esta suposición clásica y permitir más de dos valores de verdad. La estrategia más común es usar tres valores de verdad: verdadero, falso y ambos (verdadero y falso) para las evaluaciones de fórmulas.

George Boole introdujo el álgebra de la lógica o lógica algebraica en [7], como un sistema algebraico explícito que muestra la estructura matemática subyacente de la lógica. La metodología iniciada por Boole fue continuada en el siglo XIX por el trabajo de A. De Morgan, W. S. Jevons, C. S. Peirce y E. Schröder. Un resumen de estos trabajos se puede encontrar en [8]. La relación entre lógica y álgebra desde la perspectiva contemporánea se remonta a las ideas de Lindenbaum y Tarski, en las cuales las fórmulas de una lógica dada son interpretadas por álgebras con operaciones asociadas con los conectivos lógicos. En [6], Blok y Pigozzi propusieron una generalización de las técnicas de algebrización originales para abarcar un rango más amplio de lógica. Después de esto, se sugirieron varias generalizaciones del método de Blok y Pigozzi en la literatura [11, 12, 13].

La lógica C1 de da Costa fue introducida con la intención recuperar el razonamiento clásico. Mortensen en su trabajo *Every quotient algebra for c1 is trivial* (1980), probó que no se puede algebrizar con el método de Lindenbaum-Tarski. Lewin et al., en su trabajo *C1 is not algebraizable* (1991), probaron que no es algebrizable Blok-Pigozzi. Es sabido que la lógica CG'3 es una extensión de C1. En este trabajo mostramos que CG'3 es algebrizable en el sentido Blok-Pigozzi, lo que permitiría decir que CG'3 es una lógica con los requerimientos que da Costa pedía para C1 pero que es algebrizable.

Este artículo está organizado de la siguiente manera: En la sección 2, presentamos algunas definiciones conocidas y resultados acorde a la ambientación del presente manuscrito con el fin de facilitar la lectura del texto. Estudiamos la lógica CG'3 que se define en términos de cuatro conectivos \wedge , \vee , \rightarrow y \neg donde la implicación es una implicación deductiva. En la sección 3, establecemos que CG'3 es una lógica algebrizable con el método de Blok-Pigozzi. Finalmente, en la última sección, presentamos una lista de problemas abiertos para ser estudiados en el futuro.

2. Antecedentes

Primero presentamos la sintaxis de las fórmulas lógicas consideradas en este artículo. Seguimos la notación estándar y las definiciones básicas como W. Carnielli y M. Coniglio en [9].

Definición 1 (Conjunto proposicional). Un conjunto proposicional es un conjunto Θ de símbolos llamados conectivos, junto con la información relativa a la aridad de cada conectivo.

Los siguientes símbolos se usarán como conectivos lógicos: \wedge (conjunción, binario); \vee (disyunción, binario); \rightarrow (implicación, binario); \neg (negación débil, unario); (operador de inconsistencia, unario); \sim (negación fuerte, unario); \perp (fórmula bottom, 0-ary).

Definición 2 (Lenguaje proposicional). Sea $\text{Var} = \{p_1, p_2, \dots\}$ un conjunto numerable de variables proposicionales, y sea Θ cualquier conjunto proposicional. El lenguaje proposicional generado por Θ de Var será denotado por $L\Theta$.

Definición 3 (Lógica estándar). Una lógica L definida sobre un lenguaje L que tiene una relación de consecuencia \vdash , es Tarskian si cumple las siguientes propiedades, para cada $\Gamma \cup \Delta \cup \{\alpha\} \subseteq L$:

- i. si $\alpha \in \Gamma$ entonces $\Gamma \vdash \alpha$;
- ii. si $\Gamma \vdash \alpha$ y $\Gamma \subseteq \Delta$ entonces $\Delta \vdash \alpha$;
- iii. si $\Delta \vdash \alpha$ y $\Gamma \vdash \beta$ para cada $\beta \in \Delta$ entonces $\Gamma \vdash \alpha$.

Una lógica que satisface el inciso (ii) es llamada monótona. Una lógica L se dice que es finitaria si cumple lo siguiente:

- iv. si $\Gamma \vdash \alpha$ entonces existe un subconjunto finito Γ_0 de Γ tal que $\Gamma_0 \vdash \alpha$. Una lógica L definida sobre un lenguaje proposicional L generado por un conjunto de variables proposicionales se llama estructural, si cumple con la siguiente propiedad:
- v. si $\Gamma \vdash \alpha$ entonces $\sigma[\Gamma] \vdash \sigma[\alpha]$, para cada sustitución σ de fórmulas para variables.

Una lógica proposicional es estándar, si es tarskiana, finitaria y estructural. De ahora en adelante, una lógica L será representada por un par $L = \langle L, \vdash \rangle$, donde L y \vdash denota el lenguaje y la relación de consecuencia de L , respectivamente. L es generada por un lenguaje proposicional Θ de Var , esto es, $L = L_\Theta$ entonces escribiremos $L = \langle \Theta, \vdash \rangle$.

Sea $L = \langle L, \vdash \rangle$ una lógica. Sea α una fórmula en L y sean $X_1 \dots X_n$ una secuencia finita (para $n \geq 1$) tal que para cada X_i es o bien un conjunto de fórmulas en L o una fórmula en L . Entonces, como es usual, $X_1, \dots, X_n \vdash \alpha$ representará $X_1 \cup \dots \cup X_n \vdash \alpha$, donde, para cada i , X_i es X_i , si X_i es un conjunto de fórmulas, o X_i es $\{X_i\}$, si X_i es una fórmula.

Definición 4 (Lógica paraconsistente). Una lógica Tarskiana L es paraconsistente si tiene una negación (primitiva o definida) \neg tal que $\alpha, \neg\alpha \not\vdash \beta$ para algunas fórmulas α y β en el lenguaje de L .

Observación 1. Si L tiene una implicación deductiva \rightarrow , en el sentido que se satisface el meta-teorema de la Deducción MTD, entonces L es paraconsistente si y sólo si la fórmula $\varphi \rightarrow (\neg\varphi \rightarrow \psi)$ no es válida, es decir, la ley de explosión no es válida L con respecto a la negación \neg . Esto es, la negación \neg no es explosiva.

Ahora, presentamos la noción de lógica de formal inconsistencia.

Definición 5 (Lógica de Formal Inconsistencia). Sea $L = \langle \Theta, \vdash \rangle$ una lógica estándar. Supongamos que Θ de L contiene una negación \neg , y sea $\circ(p)$ un conjunto de fórmulas no vacío que depende exactamente de la variable proposicional p . En consecuencia, L es una lógica de Formal Inconsistencia, (LFI), con respecto a \neg y $\circ(p)$ si lo siguiente se tiene:

- i. $\varphi, \neg\varphi \not\vdash \psi$ para algunas φ y ψ ;
- ii. hay dos fórmulas α y β tal que;
 - a. $\circ(\alpha), \alpha \not\vdash \beta$;
 - b. $\circ(\alpha), \neg\alpha \not\vdash \beta$;
- iii. $\circ(\varphi), \varphi, \neg\varphi \vdash \psi$ para cada φ y ψ . Observación 2.

- Cuando \circ es un singular, sus elementos se denotan por $\circ p$, donde \circ es el operador de consistencia.
- Una lógica que satisface la propiedad (iii) se llama suavemente explosiva.

Finalmente, definimos una noción más fuerte de LFI para más referencia, vea [9].

Definición 6 (Lógica de Formal Inconsistencia Fuerte). Sea $L = \langle \Theta, \vdash \rangle$ una lógica estándar. Supongamos que Θ de L contiene una negación \neg , y sea

$\circ(p)$ un conjunto de fórmulas no vacío que depende exactamente de la variable proposicional p . Entonces L una LFI fuerte con respecto a \neg y $\circ(p)$ si lo siguiente se tiene:

- i. hay dos fórmulas α y β tal que:
 - a. $\alpha, \neg\alpha \not\vdash \beta$;
 - b. $\circ(\alpha), \alpha \not\vdash \beta$;
 - c. $\circ(\alpha), \neg\alpha \not\vdash \beta$; y
 - ii. $\circ(\varphi), \varphi, \neg\varphi \vdash \psi$ para cada φ y ψ . Observación 3.
- Cualquier LFI fuerte es una LFI.
- Si L es una lógica proposicional entonces L es una LFI fuerte siempre que se cumpla lo siguiente:
- i. hay dos variables proposicionales p y q tal que:
 - a. $p, \neg p \not\vdash q$;
 - b. $\circ(p), p \not\vdash q$;
 - c. $\circ(p), \neg p \not\vdash q$; y
 - ii. $\circ(\varphi), \varphi, \neg\varphi \vdash \psi$ para cada φ y ψ .

Definición 7 (Algebrización Blok-Pigozzi). Sea Θ un conjunto proposicional, y sea L una lógica proposicional estándar definida sobre el lenguaje $L\Theta$, con una relación de consecuencia $\vdash L$. Entonces L es algebrizable en el sentido de Blok y Pigozzi si existe un conjunto no vacío $\Delta(p1, p2) \subseteq L\Theta$ de fórmulas dependiendo de las variables $p1$ y $p2$, y un conjunto no vacío $E(p1) \subseteq L\Theta \times L\Theta$ de pares de fórmulas dependiendo de la variable $p1$ satisfaciendo las siguientes propiedades:

- i. $\vdash L \delta(p1, p1)$, para cada $\delta(p1, p2) \in \Delta(p1, p2)$;
- ii. $\Delta(p1, p2) \vdash L \delta(p2, p1)$, para cada $\delta(p1, p2) \in \Delta(p1, p2)$;
- iii. $\Delta(p1, p2), \Delta(p2, p3) \vdash L \delta(p1, p3)$, para cada $\delta(p1, p2) \in \Delta(p1, p2)$;
- iv. $\Delta(p1, pn+1), \dots, \Delta(pn, p2n) \vdash L \delta(\#(p1, \dots, pn), \#(pn+1, \dots, p2n))$, para cada $\delta(p1, p2) \in \Delta(p1, p2)$, cada conectivo n -ario, $\#$ de Θ y cada $n \geq 1$;
- a. $(\forall) p1 \vdash L \delta(\gamma(p1), s(p1))$, para cada $\delta(p1, p2) \in \Delta(p1, p2)$ y cada $\langle \gamma(p1), \epsilon(p1) \rangle \in E(p1)$;
- v. $\{\delta(\gamma(p1), \epsilon(p1)) : \delta(p1, p2) \in \Delta(p1, p2), \langle \gamma(p1), \epsilon(p1) \rangle \in E(p1)\} \vdash L p1$.

Los conjuntos $\Delta(p1, p2)$ y $E(p1)$ denominan formulas de sistema de equivalencia de fórmulas y sistema de ecuaciones definidas, respectivamente.

Definición 8 (Relación). Sea Θ un conjunto proposicional, y sea $\Theta \subseteq L\Theta \times L\Theta$ una relación definida sobre el álgebra de fórmulas $L\Theta$ si cumple las siguientes propiedades:

- i. $\alpha\theta\alpha$ para cada $\alpha \in L\Theta$ (reflexividad).
- ii. $\alpha\theta\beta$ implica que $\beta\theta\alpha$ para cada $\alpha, \beta \in L\Theta$ (simetría).
- iii. $\alpha\theta\beta$ y $\beta\theta\gamma$ implica que $\alpha\theta\gamma$ para cada $\alpha, \beta, \gamma \in L\Theta$ (transitividad).
- iv. Dados α_i y β_i en $L\Theta$ (para $1 \leq i \leq n$) tal que $\alpha_1\theta\beta_1, \dots, \alpha_n\theta\beta_n$, entonces $\#(\alpha_1, \dots, \alpha_n)\theta\#(\beta_1, \dots, \beta_n)$ para cada conectivo n -ario $\#$ de Θ y cada $n \geq 1$.
- v. Una congruencia θ en $L\Theta$ es trivial si ya sea $\theta = \{(\alpha, \alpha) : \alpha \in L\Theta\}$ o $\theta = L\Theta \times L\Theta$.

Definición 9 (Congruencia lógica). Sea L una lógica estándar definida sobre el lenguaje $L\Theta$.

- i. Una congruencia θ en $L\Theta$ es compatible con una teoría $\Gamma \subseteq L\Theta$ si se satisface lo siguiente:
 - ii. $\alpha\theta\beta$ y $\Gamma \vdash L \alpha$ implica que $\Gamma \vdash L \beta$.
- i. Una congruencia θ en $L\Theta$ es una congruencia lógica en L si θ es compatible con cada teoría Γ . Equivalentemente, θ es una congruencia lógica en L si, para cada α y β :
 - iii. $\alpha\theta\beta$ implica que $\alpha \vdash L \beta$ y $\beta \vdash L \alpha$.

La manera usual de definir la semántica multivaluada de una lógica es a través de una matriz. Introducimos la definición de la matriz determinista, también conocida como matriz lógica o simplemente como matriz. En [16], podemos encontrar una discusión exhaustiva sobre lógica multivaluada y algunos ejemplos.

Definición 10 (Matriz). Dada una lógica L en el lenguaje L , la matriz de L es una estructura $M = (D, D^*, F)$, donde:

- i. D es un conjunto no vacío de valores de verdad (dominio).
- ii. D^* es un subconjunto de D (conjunto de valores designados).
- iii. $F = \{f_c \mid c \in C\}$ es un conjunto de funciones de verdad, con una función por cada conectivo lógico c de L .

Definición 11 (Interpretación). Dada una lógica L en el lenguaje L , una interpretación t , es una función $t: \text{Var} \rightarrow D$ que asigna variables proposicionales a elementos en el dominio.

Cualquier interpretación t se puede extender a una función a todas las fórmulas en $L\Sigma$ como de costumbre, es decir, aplicando recursivamente las funciones de verdad de los conectivos lógicos en F . Si t es una interpretación en la lógica L , diremos que t es una interpretación L . Las interpretaciones nos permiten definir la noción de validez en este tipo de semántica de la siguiente manera:

Definición 12 (Fórmula válida). Dada una fórmula φ y una interpretación t en una lógica L , decimos que la fórmula φ es válida bajo t en L , si $t(\varphi) \in D^*$, y lo denotamos como $t \models L \varphi$.

Tengamos en cuenta que la validez depende de la interpretación, pero si queremos hablar de “verdades lógicas” en el sistema, entonces la validez debe ser absoluta, como se indica en la siguiente definición:

Tabla 1. Funciones de verdad para los conectivos $\vee, \wedge, \rightarrow, \neg$ en CG'3.

$f\vee$	0	1	2	$f\wedge$	0	1	2	$f\rightarrow$	0	1	2	$f\neg$
0	0	1	2	0	0	0	0	0	2	2	2	0
1	1	1	2	1	0	1	1	1	0	2	2	1
2	2	2	2	2	0	1	2	2	0	1	2	2

Definición 13 (Tautología). Dada una fórmula φ en el lenguaje de una lógica L , decimos que φ es una tautología en L , si para cada posible interpretación, la fórmula φ es válida, y la denotamos como $\models_L \varphi$.

Si φ es una tautología en la lógica L , decimos que φ es una L -tautología.

2.1. La lógica CG'3

La lógica CG'3 fue introducida en [17]. Los autores, la definieron como una lógica 3-valuada en donde la matriz esta dada por la estructura $M = \langle D, D^*, F \rangle$, donde $D = \{ 0, 1, 2 \}$, el conjunto D^* de valores designados $\{ 1, 2 \}$, y F es el conjunto de funciones de verdad definidas en la Tabla 1.

Observación 4.

1 Note que \rightarrow es una implicación deductiva: $\Gamma, \alpha \models_{CG'3} \beta$ si y sólo si $\Gamma \models_{CG'3} \alpha \rightarrow \beta$.

$$2 \quad \alpha \rightarrow \beta.$$

1 Considerando el orden natural $0 \leq 1 \leq 2$ en D , \vee corresponde al supremo:

\wedge corresponde al ínfimo y \rightarrow es el residuo de \wedge :

$$z \wedge x \leq y \text{ si y sólo si } z \leq x \rightarrow y, \text{ para cada } x, y, z \in D.$$

2 La lógica CG'3 fue axiomatizada en [19], aplicando el método de Lindenbaum-Los. Además los autores definen tres conectivos: $\sim\varphi := \varphi \rightarrow \perp$ (Negación fuerte), $\bullet\varphi := \sim\sim\varphi \wedge \neg\varphi$ (Operador de insoncistencia) $\circ\varphi := \neg\bullet\varphi$ (Operador de consistencia).

3. La algebrización de CG'3

En [6], Blok y Pigozzi dieron un concepto matemático de lógica algebrizable. La idea principal de esta definición es la siguiente: Una lógica es algebrizable si existe una clase de álgebras asociadas a la lógica de la misma manera que la clase de álgebras booleanas está relacionada con la lógica proposicional clásica.

Proposición 14 La lógica CG'3 es una LFI fuerte con un operador de consistencia \circ como se definió anteriormente.

Demostración. Suponga que p y q son dos variables proposicionales diferentes. Al considerar la interpretación v_1 tal que $v_1(p) = 1$, $v_1(\neg p) = 2$, y $v_1(q) = 0$, resulta que $p, \neg p \models_{CG'3} q$ y la cláusula (i.a) de la Observación 3 se verifica. Considere la

interpretación v_2 tal que $v_2(p) = 2$, $v_2(\circ p) = 2$, y $v_2(q) = 0$, se tiene que $p, \circ p \models_{CG'3} q$ y la cláusula (i.b) de la Observación 3 se cumple.

Ahora, considerando la interpretación v_3 tal que $v_3(\neg p) = 1$, $v_3(\circ p) = 2$, y $v_3(q) = 0$, resulta que $p, p \models_{CG'3} q$ y la cláusula (i.c) de la Observación 3 se verifica. Finalmente, no hay una interpretación que haga a φ , $\neg\varphi$, y $\circ\varphi$ simultáneamente verdaderas. Por lo tanto, el inciso (ii) de la Observación 3 se satisface. Por lo tanto, $CG'3$ es una LFI fuerte con respecto a \neg y \circ .

Proposición 15 Sea h una interpretación para $CG'3$. Entonces:

- i. $h(p_1 \rightarrow p_2) \in D^*$ si y sólo si $h(p_1) = 0$ o $h(p_2) \in D^*$;
- ii. $h(p_1 \wedge p_2) \in D^*$ si y sólo si $h(p_1) \in D^*$ y $h(p_2) \in D^*$;
- iii. $h(p_1 \vee p_2) \in D^*$ si y sólo si $h(p_1) \in D^*$ o $h(p_2) \in D^*$;
- iv. $h(p_1 \leftrightarrow p_2) \in D^*$ si y sólo si o bien $h(p_1) \in D^*$ y $h(p_2) \in D^*$, o $h(p_1) = h(p_2) = 0$.

Demostración. Inmediato de las tablas de verdad.

Definición 16 Sea $\delta(p_1, p_2)$ la siguiente fórmula de $L\Sigma$:
 $\delta(p_1, p_2) = (p_1 \leftrightarrow p_2) \wedge (\circ p_1 \leftrightarrow \circ p_2)$

Proposición 17 Sean α y β fórmulas en $L\Sigma$. Entonces para cada interpretación h para $CG'3$, se tiene que:

1. $h(\delta(\alpha, \beta)) \in D^*$ si y sólo si tenemos:
 - a. o bien $h(\alpha) \in D^*$ y $h(\beta) \in D^*$, o $h(\alpha) = h(\beta) = 0$; y
 - b. $h(\circ\alpha) = h(\circ\beta)$
2. Así, $h(\delta(\alpha, \beta)) \in D^*$
3. $H(\bullet\alpha \rightarrow \alpha) \in D^*$.
4. si y sólo si $h(\alpha) = h(\beta)$.
5. $h(\circ\alpha) = h((\bullet\alpha \rightarrow \alpha))$.
6. $h(\delta(\alpha, \bullet\alpha \rightarrow \alpha)) \in D^*$ si y sólo si $h(\alpha) \in D^*$.

Demostración

1. Parte “Sólo si”. $H(\delta(\alpha, \beta)) \in D^*$ si y sólo si de acuerdo con la Proposición 15(ii), $h(\alpha \leftrightarrow \beta) \in D^*$ y $h(\alpha \leftrightarrow \beta) \in D^*$. Por la Proposición 15(iv), $h(\alpha \leftrightarrow \beta) \in D^*$ es equivalente a o bien $h(\alpha) \in D^*$ y $h(\beta) \in D^*$, o $h(\alpha) = h(\beta) = 0$, mientras $h(\circ\alpha \leftrightarrow \circ\beta) \in D^*$ si y sólo si o bien $h(\circ\alpha) \in D^*$ y $h(\circ\beta) \in D^*$, o $h(\circ\alpha) = h(\circ\beta) = 0$. De la definición de \circ , $h(\circ\alpha \leftrightarrow \circ\beta) \in D^*$ es equivalente a $h(\circ\alpha) = h(\circ\beta)$. Ahora, supongamos que $h(\delta(\alpha, \beta)) \in D^*$. Si $h(\alpha) = 2$ y $h(\beta) = 1$, entonces $h(\circ\alpha) = 2$ y $h(\circ\beta) = 0$, violando que $h(\circ\alpha) = h(\circ\beta)$.
2. Análogamente, es imposible tener que $h(\alpha) = 1$ y $h(\beta) = 2$. Esto muestra que $h(\alpha) = h(\beta)$. Parte “Si” es obvia, a la luz de las cláusulas.
3. La prueba es directa.

4. Si $h(\circ\alpha) = 2$ entonces $h(\alpha) \in \{0, 2\}$ y $h(\bullet\alpha) = 0$ y así $h(\bullet\alpha \rightarrow \alpha) = 2$; por lo tanto, $h(\circ(\bullet\alpha \rightarrow \alpha)) = 2$. Si $h(\circ\alpha) = 0$ entonces $h(\alpha) = 1$ y así $h(\bullet\alpha) = 2$; por lo tanto, $h(\bullet\alpha \rightarrow \alpha) = 1$; así, $h(\circ(\bullet\alpha \rightarrow \alpha)) = 0$.
5. Parte "Sólo si". Por el inciso 1, $h(\delta(\alpha, \bullet\alpha \rightarrow \alpha)) \in D^*$ implica que $h(\alpha) = h(\bullet\alpha \rightarrow \alpha)$. Entonces, por el inciso 2, $h(\alpha) \in D^*$. Parte "Si". Supongamos que $h(\alpha) \in D$. Por el inciso 2, $h(\bullet\alpha \rightarrow \alpha) \in D^*$ y, por el inciso 3, $h(\circ\alpha) = h(\circ(\bullet\alpha \rightarrow \alpha))$. Finalmente, aplicando inciso 1, $h(\delta(\alpha, \bullet\alpha \rightarrow \alpha)) \in D^*$.

Teorema 18 La lógica CG'3 es algebrizable en el sentido de Blok y Pigozzi con un sistema de fórmulas equivalentes dado por $\Delta(p1, p2) = \{\delta(p1, p2)\}$ y un sistema de ecuaciones definidas dado por $E(p1) = \{p1, \bullet p1 \rightarrow p1\}$.

Demostración. Es fácil demostrar que el sistema $\Delta(p1, p2)$ cumple las condiciones (i)-(iv) según el inciso 1 de la Proposición 17. Por el inciso 4 de la misma proposición, las condiciones (v)-(vi) se siguen fácilmente.

4. Conclusiones y trabajo futuro

CG'3 se define por su semántica multivaluada, la matriz de la lógica CG'3 está dada por $M = (D, D^*, F)$; donde el dominio es $D = \{0, 1, 2\}$ y el conjunto de valores designados es $D^* = \{1, 2\}$. Esta lógica es paraconsistente y puede verse como una extensión de la lógica G'3 también introducida por Osorio en 2008, [18]. En este artículo, ampliamos los estudios sobre esta lógica al presentar algunos resultados relacionados con lógica algebraica. El resultado principal del trabajo es la algebrización de CG'3 usando la técnica de Blok y Pigozzi. Este resultado nos abre las puertas para el desarrollo, implementación y aplicación de sistemas de inferencia paraconsistente robustos [20]. Entre las aplicaciones, es de nuestra particular atención la verificación de sistemas [15].

Otra pregunta de investigación de nuestro interés es con respecto a la relación de CG'3 y la familia de lógicas paraconsistentes anotadas, definidas por Subrahmanian en [22]. Hoy en día se conocen muchas aplicaciones de lógicas paraconsistentes en diversos campos de las ciencias computacionales, como: circuitos eléctricos, razonamiento no monótono, sistemas de control, automatización y robótica, por mencionar algunos [1].

References

1. Abe, J.M., Nakamatsu, K., da Silva Filho, J.I.: Three decades of paraconsistent annotated logics: a review paper on some applications. *Procedia Computer Science*, 159, pp. 1175–1181 (2019)
2. Abe, J.M.: Fundamentos da lógica anotada. *Foundations of Annotated Logics*. Ph. D. Thesis, University of São Paulo (1992)
3. Abe, J.M.: Paraconsistent intelligent-based systems: New trends in the applications of paraconsistency. 94, Springer (2015)
4. Abe, J.M., Akama, S., Nakamatsu, K.: Introduction to annotated logics: foundations for paracomplete and paraconsistent reasoning. 88, Springer (2015)

5. Blass, A.: 2003 annual meeting of the association for symbolic logic. *Bulletin of Symbolic Logic* 10(1), 120–145 (2004). <https://doi.org/10.1017/S1079898600004248>
6. Blok, W.J., Pigozzi, D.: *Algebraizable logics*, vol. 77. American Mathematical Soc. (1989)
7. Boole, R.: *The mathematical analysis of logic*, 1847 (1847)
8. Burris, S., Legris, J.: The algebra of logic tradition. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2018 edn. (2018)
9. Carnielli, W.A., Coniglio, M.E.: *Paraconsistent logic: Consistency, contradiction and negation*. Springer (2016)
10. Curry, H.B.: Functionality in combinatory logic. In: *Proceedings of the National Academy of Sciences* 20(11), 584–590 (1934)
11. Czelakowski, J.: Protoalgebraic logics. In: *Protoalgebraic Logics*, pp. 69–122. Springer (2001)
12. Font, J.M., Jansana, R.: *A general algebraic semantics for sentential logics*. Vol. 7. Cambridge University Press (2017)
13. Font, J.M., Jansana, R., Pigozzi, D.: A survey of abstract algebraic logic. *Studia Logica* 74(1-2), 13–97 (2003)
14. Gottwald, S.: Many-valued logic. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edn. (2017)
15. Limón, Y., Bárcenas, E., Benítez-Guerrero, E., Molero, G.: On the consistency of context-aware systems. *J. Intell. Fuzzy Syst.* 34(5), 3373–3383 (2018)
16. Malinowski, G.: *Many-Valued Logics*. Oxford University Press (1993)
17. Osorio, M., Carballido, J.L., Zepeda, C., et al.: Revisiting Z. *Notre Dame Journal of 4 Formal Logic* 55(1), 129–155 (2014)
18. Osorio Galindo, M., Carballido Carranza, J.L.: Brief study of g^3 logic. *Journal of Applied Non-Classical Logics* 18(4), 475–499 (2008)
19. Pérez-Gaspar, M., Barcenas, E.: Completeness for the paraconsistent logic CG (2019)
20. Pérez-Gaspar, M., Hernandez-Tello, A., Arrazola Ramírez, J., Osorio Galindo, M.: An axiomatic approach to CG^3 logic. *Logic Journal of the IGPL* (2020)
21. Priest, G., Tanaka, K., Weber, Z.: Paraconsistent logic. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edn. (2018)
22. Subrahmanian, V.: On the semantics of quantitative logic programs. In: *SLP*, pp. 173–182 (1987)
23. Thomason, R.: Logic and artificial intelligence. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2020 edn. (2020)
24. Wieckowski, B., Metcalfe, G., Olivetti, N., Gabbay, D.: Proof theory for fuzzy logics. *Applied logic series*, vol. 36, *Bulletin of Symbolic Logic* 16(3), 415–419 (2010)
25. Zamansky, A.: On recent applications of paraconsistent logic: an exploratory literature review. *Journal of Applied Non-Classical Logics* 29(4), 382–391 (2019)

A Systematic Mapping Study of Artificial Intelligence in Software Requirements

Alfonso Robles Aguilar, Jorge Octavio Ocharán Hernández,
Ángel Juan Sánchez García

Universidad Veracruzana,
Facultad de Estadística e Informática,
Mexico.

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

Abstract. Artificial Intelligence techniques are used in a wide variety of domains. The application of Artificial Intelligence (AI) in Software Engineering has assisted activities that are time-consuming and resource-intensive such as those related to software requirements. This paper focuses on the use of AI techniques in Requirements Engineering (RE). The RE activities are critical in the life cycle of the software since they are the foundation of further phases. This paper presents the results of a Systematic Mapping Study that reports the extensive use of AI techniques in different RE activities, including recent trends, results, and future work.

Keywords: Requirements Engineering, Artificial Intelligence, Systematic Mapping Study, Software Development.

1 Introduction

In recent years, the interest in the application of Artificial Intelligence (AI) techniques, algorithms, and methods in Software Engineering (SE) has increased, seeking to address the problems and challenges that arise in the creation of software systems [1]. AI represents an opportunity to automate processes in the software industry. As a result, the successful use of AI techniques in almost all areas of SE has been reported, highlighting search, optimization, classification, learning, and prediction algorithms [1]. Within the SE, the closest phase to the user is the Requirements Analysis [2], that is part of a whole discipline called Requirements Engineering (RE), which in addition to the analysis, covers the elicitation, specification, and verification of the Software Requirements (SR) [3]. A SR is a capability that must be met or possessed by a software system to satisfy a contract, standard, specification, or other documents [4]. This definition allows us to glimpse the importance that RE has in SE since they reflect the user's needs for the system.

The aim of this paper is twofold. Firstly, to make a succinct presentation of the state the art in the use of AI techniques in RE activities in such a way Requirements Engineers could consult a compilation of the existing techniques to find a solution to their problems or improve their processes. Secondly, to identify research gaps in the use and optimization of AI techniques.

This paper is organized as follows. Section II presents previous works on the use of AI in RE activities. Section III presents the systematic review method used in this research. Section IV discusses the results of this study, and Section V presents the contribution of this paper and our conclusions.

2 Related Work

The interest to know methodologies to support RE has increased, such as the review conducted by Dermeval *et al.*, where they report the application of ontologies in RE as well as how ontologies support RE [5].

Due to Harman's report of increased interest in using AI in SE [1], we can assume that there is also research in this area, and AI techniques have been applied in RE seeking to improve the processes and results of their activities. As evidence of the interest in this area, five reviews related to this Systematic Mapping Study (SMS) were found in preliminary research.

Sorte *et al.* [6] reviewed the state of the art of AI use throughout the software development life cycle. This implies that no area is delved into, such as RE, with only four papers are reported, which gives us the opportunity to explore deeper this topic.

Águila and del Sagrado [2] focus on finding how and where Bayesian Networks (BN) are applied, an area of AI [7], in the RE. This paper suggests that BN has not been applied in all areas of the RE, but that there is synergistic cooperation between these two, as they report cases in which some RE activities have been improved using BN. Since this paper is from 2016, we can assume that, due to the reported synergy, the investigation has been continued in the use of BN in RE, and there are works from more recent years that were not reported. This also justifies the creation of our paper, in addition to the fact that we wanted to cover more areas of AI and not just BN.

Gramajo *et al.* [8] and Iqbal *et al.* [9] expose papers where Machine Learning (ML) techniques are used, another area of AI [7], in the RE. Both reviews present the use of ML algorithms exclusive int RE, such as Support Vector Machine (SVM), Naïve Bayes, and Neural Networks. The purpose of our mapping, unlike these two, is to report more than one area.

Finally, Haq *et al.* [10] have a review similar to this one, in which they explore the uses of AI in RE, with the difference that they classified RE in phases, this study that focuses on RE activities. Furthermore, except for the first and second International Workshop on Artificial Intelligence for Requirements Engineering, the sources from which the studies were obtained are not explicitly specified nor the search string used in the electronic databases.

As can be seen from the number of reviews, there is considerable interest in the use of AI in RE, so a more updated version with scope for more AI areas may be more useful for researchers in the area.

Table 1. Keywords.

Concept	Keywords
Requirements Engineering	software requirements engineering, requirements engineering, software requirements analysis, software requirements specification, software requirements verification
Artificial Intelligence	Artificial Intelligence

3 Research Method

This SMS was conducted according to the guidelines proposed by Kitchenham and Charters [11]. They include the specification of the research questions, primary study search and selection, and data extraction and data synthesis processes.

3.1 Planning

Research Questions. The research questions (RQs) relating to this SMS are:

- RQ1: What are the AI techniques used in RE?
- RQ2: What RE activities have been addressed with these techniques?

The motivation of the RQ1 is to identify the use of AI techniques that have been applied in RE. Regarding the reason for RQ2 is to discover in which specific RE activities, AI techniques have been applied.

Search strategy. The search strategy was an automated search, looking for the primary studies among the results that electronic databases show from a search string.

Keywords. These words (Table 1) were selected based on the Research Questions. Regarding RQ1, the most important word that was found was “Artificial Intelligence,” which resulted in a Keyword for the search. Regarding RQ2, “Requirements Engineering” was identified, and as Keywords, all phases of the RE were selected to cover every aspect of the process in the search.

Search string. After trying different strings using the keywords and analyzing the results they showed, the following search string was formulated, in order to cover studies that apply RE or at least some of its phases:

(“software requirements engineering” OR “requirements engineering” OR “software requirements analysis” OR “software requirements specification” OR “software requirements verification”) AND (“artificial intelligence”)

It should be clarified that one of the limitations of our study is that by wanting to cover all AI, it is likely that the search chain omits papers that are valuable for our mapping but if the paper didn’t use the concept AI, the chain could not show them.

Table 2. Inclusion criteria.

ID	Description
IC1	They were published between 2015 and 2019.
IC2	Written in English.
IC3	The title, abstract, or keywords suggests that the paper answers at least one research question.
IC4	Answer at least one research question.

Table 3. Exclusion criteria.

ID	Description
EC1	Reports the requirements of some products other than software.
EC2	It is not a journal or proceedings paper.
EC3	Reports requirements for an AI system but does not use AI in the requirements process.
EC4	Duplicate papers (same research in different databases).

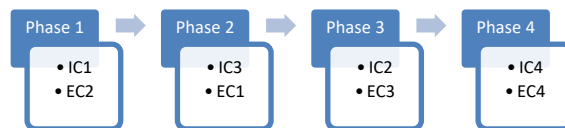


Fig. 1. Primary studies selection process.

Electronic database resources. The sources used for the search and selection of primary studies are four electronic databases: ScienceDirect, SpringerLink, IEEE Xplore, and ACM Digital Library. These four sources of information were chosen because they contain papers from the area of computer science, SE, and related disciplines, following the path marked by the Keywords to obtain the necessary papers in this area. Furthermore, when published in these electronic databases, the papers went through an arbitration process by experts in the field.

Primary study selection criteria. Table 2 and Table 3 show the used criteria to determine if a study was considered a Primary Study (PS) or the paper did not meet what was necessary to be included.

Primary study selection procedure. The phases in which these selection criteria were applied to choose the PS are the ones shown in Fig. 1. When doubts were raised regarding the inclusion of the papers, the second author (AI expert) and the third author (SR expert) decided whether the paper should be a primary study or not.

Data extraction. Data were extracted using the extraction template shown in Table 4. We extracted two kinds of data from each one of the primary studies selected, that were considered necessary in order to answer the research questions. As recommended by the method, the first five data extracted correspond to publication details. The sixth and seventh data are directly related to RQ1 and RQ2 and correspond to context information; these data can answer both questions.

Table 4. Data extraction template.

Kind of data	Data extracted
Publication details	Title
	Authors
	Year
	Source
	Author keywords
Context	AI technique
	RE activity

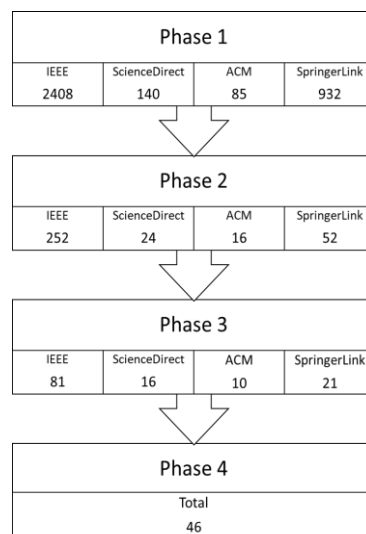


Fig. 2. Primary studies selection phases.

Data synthesis. For data synthesis, the Thematic Synthesis strategy proposed by Cruzes and Dyba [12] was followed for the analysis of qualitative data. This is because this is a method that has been found useful in qualitative data analysis in the SE area reviews.

3.2 Conducting

As shown in Fig. 2, the string applied in digital libraries yielded a total of 3,565 documents, applying the publication date, and study type filters (Phase 1). After Phase 2, a total of 344 papers made it to Phase 3, and in this phase, 128 papers met the selection criteria (IC2, EC3). Finally, 46 papers fulfilled all the selection criteria, and these were selected as PS.

The process for carrying out this review is like many other reviews in the area of Software Engineering since it used methodologies already used by experts in the field. In order to achieve this, the previously mentioned guidelines were followed.

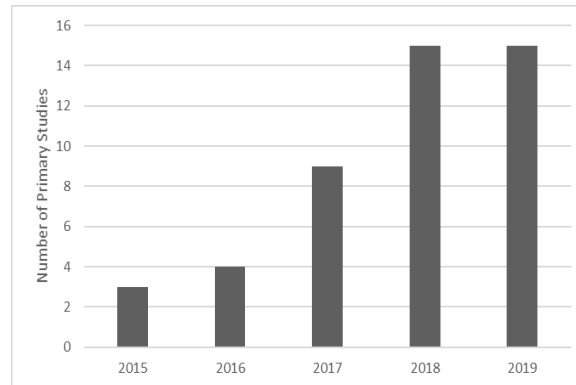


Fig. 3. Number of papers per year publication.

4 Results

Forty-six studies were selected, where 102 cases of the use of some AI technique in some RE activity were reported since some papers reported using two or more techniques in one or more activities. Regarding the authors, 141 different names were identified, of which only ten were present in more than one paper (precisely two papers each), these are Davide Dell'Anna, Fabiano Dalpiaz, Zahra Shakeri Hossein Abad, Didar Zowghi, Martin Glinz, Guenther Ruhe, Ralph Samer, Muesluem Atas, Alexander Felfernig, and Atushi Ohnishi.

It can be seen in Fig. 4 a definite growing interest in the use of AI in RE in recent years. Having three papers published in 2015 to 15 in 2018. We can observe a clear trend since publications have quintupled in just two years and maintaining interest in 2019 with 15 publications equally to the previous year.

In order to answer the research questions, the thematic synthesis process shown in Fig. 5 was followed, and 37 themes were obtained from the data that answer the research questions. The relationship among these themes is shown in Fig. 6. In respect of techniques, these were organized into groups of algorithms. Of the 46 papers, 102 reports were obtained, which were grouped into 20 higher-order themes. To achieve this, techniques that belong to the same family, such as Naïve Bayes Multinomial and Naïve Bayes, were put in the same theme.

To cite another example, neural networks such as the Convolutional Neural Network and the Recurrent Neural Network, as well as the rest of the Neural Networks, were placed in the same theme. Regarding RE activities, were grouped in 17 higher-order themes of what best described the work performed by the technique in the RE process.

For example, the cases in which requirements were classified were grouped into one (Requirements Classification), whether they were classification of non-functional, functional, security requirements, or any other way of classifying them. Another example was the theme classified as "Requirements Extraction" Activities that extracted requirements from a document such as AppStore and Twitter reviews were located here.

A Systematic Mapping Study of Artificial Intelligence in Software Requirements

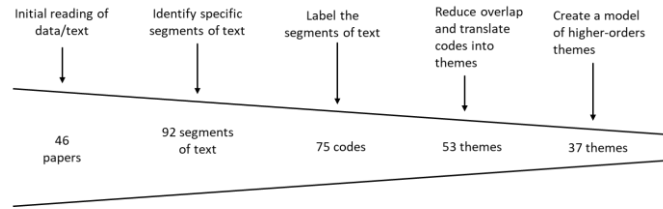


Fig. 4. Thematic synthesis process.

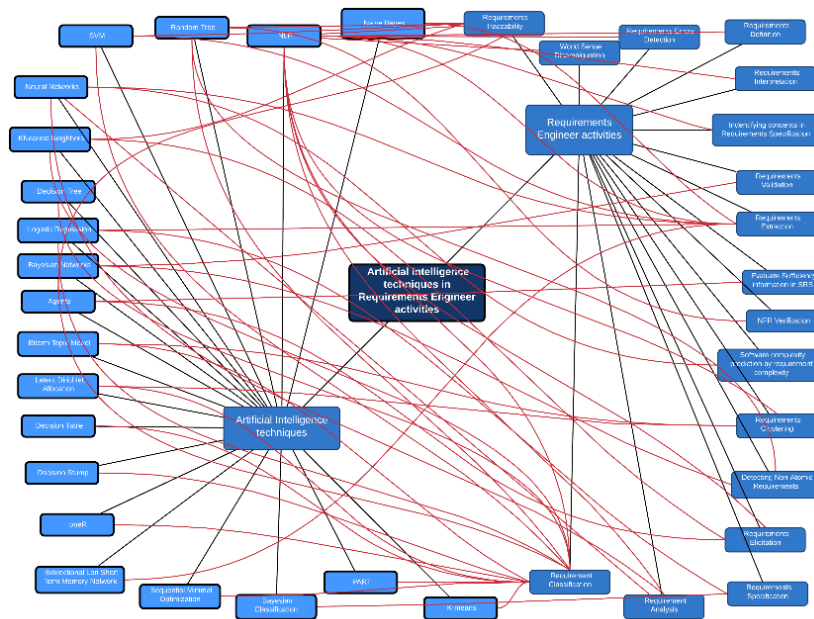


Fig. 6. Thematic map result.

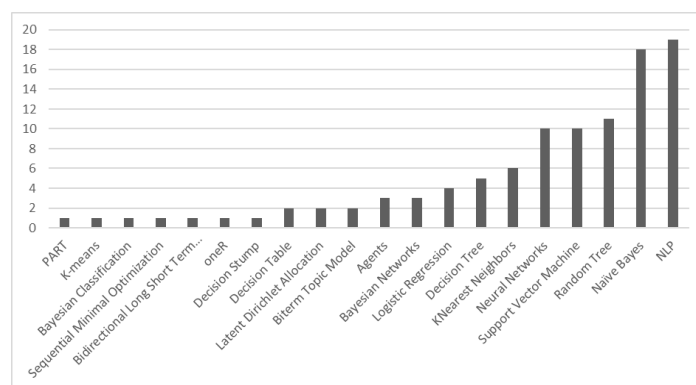


Fig. 7. Number of reports per AI technique.

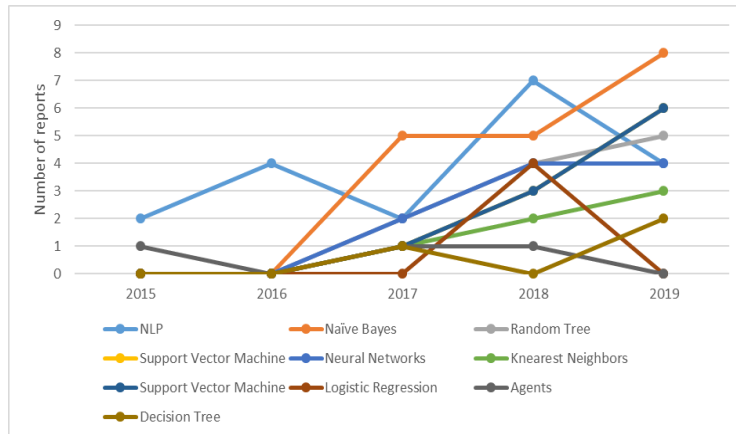


Fig. 8. Number of main AI techniques reports per year.

In response to **RQ1**, we found that the use of 20 different AI techniques in RE activities has been reported. We can see the extensive use of Natural Language Processing (NLP) since it accounted for 19 reports, representing 18.62% of the total. Also, Naïve Bayes reports the use of 18 cases in RE activities, representing 17.64% of cases. Random Tree reported 11 cases and tied with ten reports each, we found that SVM and Neural Networks also have records of being used in RE.

The last five techniques mentioned sum up 68 reports, which represent 66.68% of the total, suggesting that there is a remarkable concentration in the use of these five techniques since the remaining 15 techniques only account for a third of the total. The remaining can be found in Fig. 7.

As shown in Fig. 8, in the last three years there has been an increasing trend in the use of NLP, K-Nearest Neighbors, Naïve Bayes, SVM, Neural Networks, and Random Tree. Logistic Regression had a remarkable use in 2018 but had no record in any other year. It can be seen a clear trend in the use of Naïve Bayes.

In order to answer **RQ2**, a total of 17 different activities (Fig. 9) were found that are reported to have been carried out by IA. As we can notice, there is an extensive study in the Requirements Classification. Of the 102 cases reported, 59 sought ways to classify requirements, representing 57.84% of the total, a significant amount considering that it is only one of 18 activities. Given the large amount of work on requirements classification, a requirement engineer will be able to find many suggestions to use. Regarding Requirements Analysis and Requirements Extraction, nine and eight reports were found, respectively, in which AI techniques were used.

In the table in Appendix A¹, the specific use that is given to AI techniques in RE activities can be seen. With it, we can identify the relations that exist between AI techniques and RE activities, as well as the references to all 46 papers selected as primary studies (Appendix B²). Appendix A and B are available on the Web.

In addition, we can see that Naïve Bayes has been used 16 times in Requirements Classification throughout 11 papers (this happened because some papers use more

¹ <https://bit.ly/3kVoGHV>

² <https://bit.ly/2S93PEq>

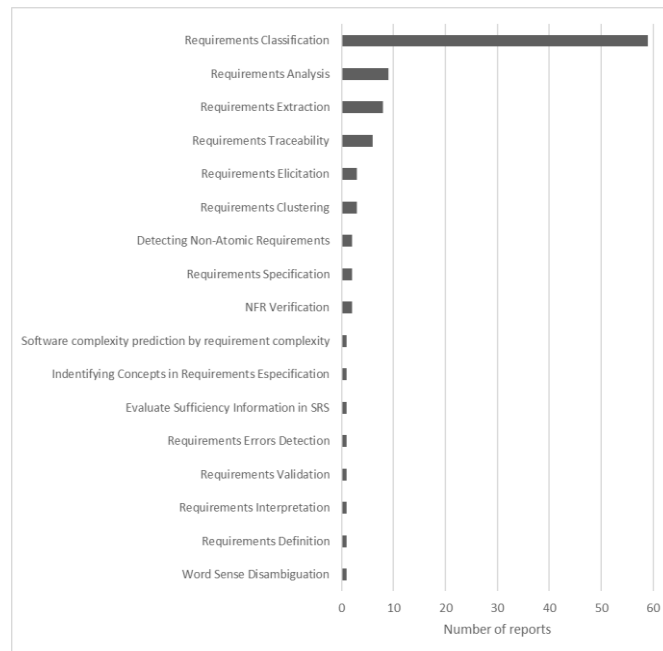


Fig. 9. Number of reports per RE activity.

than one type of Naïve Bayes). The following two areas with most reports (tied with eight reports each in seven papers) are the use of SVM and Random Trees, both in Requirements Classification. In this table, it can be seen the vast catalog of AI techniques that exist to classify requirements: 17 ways to classify AI requirements were found in this SMS. Only 3 of all the AI techniques in this work have not been applied for this activity.

Another point to note is that the Requirements Analysis, even being the second most reported activity, has only been addressed by two AI techniques: NLP and Neural Networks. Another noteworthy point is that the Requirements Extraction, although it only has eight reports, has tried to tackle with six different types of AI techniques. The table in Appendix A can be an excellent opportunity for researchers to get to know the least studied areas and have quick access to existing records.

5 Conclusions and Future Work

An SMS was conducted to answer the two research questions, applying a search string in four electronic databases where 46 studies were found that report of the use of AI techniques in RE. After analyzing those studies, a detailed summary of the AI techniques that have been used in the RE was drawn, where the use of 20 techniques in 17 different RE activities is reported. The most used techniques are NLP and Naïve Bayes. The most addressed activity is Requirements Classification with 59 reports. In them, it is noted that this activity has been carried out with 19 different AI techniques, being Naïve Bayes the most used with 16 reports. As future work, as this paper is part

of more extensive research work, we will seek to know the use of AI in other areas of Software Engineering, as well as obtain in more detail the data from this paper.

References

1. Harman, M.: The role of artificial intelligence in software engineering. In: Proceedings of the First International Workshop on Realizing AI Synergies in Software Engineering, pp. 1–6, IEEE Press (2012)
2. Del Águila, I.M., Del Sagrado, J.: Bayesian networks for enhancement of requirements engineering: a literature review. *Requir. Eng.*, 21, pp. 461–480 (2016)
3. Bourque, P., Fairley, R.E.: SWEBOK: guide to the software engineering body of knowledge. IEEE Computer Society (2014)
4. ISO/IEC/IEEE: ISO/IEC/IEEE International Standard - Systems and software engineering-Vocabulary. ISO/IEC/IEEE 24765:2010(E), pp. 1–418 (2010)
5. Dermeval, D., Vilela, J., Bittencourt, I.I., Castro, J., Isotani, S., Brito, P., Silva, A.: Applications of ontologies in requirements engineering: a systematic review of the literature. *Requir. Eng.*, 21, pp. 405–437 (2016)
6. Sorte, B., Joshi, P., Jagtap, V.: Use of artificial intelligence in software development life cycle: A state of the art review. *Int. J. Technol. Manag.*, 03, pp. 2309–4893 (2015)
7. Russell, S., Norvig, P.: Artificial intelligence: A modern approach. Pearson (2016)
8. Gramajo, M.G., Ballejos, L., Ale, M.: Software requirements engineering through machine learning techniques: A literature review. In: IEEE Biennial Congress of Argentina (ARGENCON), pp. 1–7 (2018)
9. Iqbal, T., Elahidoost, P., Lúcio, L.: A bird's eye view on requirements engineering and machine learning. In: 25th Asia-Pacific Software Engineering Conference (APSEC), pp. 11–20 (2018)
10. Haq, B., Nadeem, M., Ali, I., Ali, K., Raza, M., Rehman, M.U.: Use of expert system in requirements engineering process a systematic literature review. In: 2019 UK/ China Emerging Technologies (UCET), pp. 1–5 (2019)
11. B. Kitchenham, S.C.: Guidelines for performing systematic literature reviews in software engineering (2007)
12. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: International Symposium on Empirical Software Engineering and Measurement, pp. 275–284 (2011)

Detector de precios de gasolina basado en visión artificial y computación en la nube

Mircel K. García Rodríguez¹, Osslán O. Vergara Villegas¹,
Ivón O. Benítez González²

¹ Universidad Autónoma de Ciudad Juárez,
México

² Universidad Tecnológica de La Habana José Antonio Echeverría,
Cuba

all83283@alumnos.uacj.mx, overgara@uacj.mx,
ivonoristelabg@gmail.com

Resumen. El reconocimiento óptico de caracteres dentro de los sistemas de visión permite automatizar información y procesos en los que con más frecuencia la intervención del ser humano es cuestionada. Tal es el caso de la actualización manual de los precios de la gasolina en las estaciones de servicio de combustibles que ha provocado incongruencias entre el precio real y el publicado en las plataformas de la Comisión Reguladora de Energía (CRE) en México. El objetivo del artículo es presentar un detector automático de precios para las gasolinas que se ofertan en una estación de servicio usando una cámara digital, así como un procedimiento para enviar la información a la nube de Microsoft Azure. Así, será posible mostrar a los usuarios de forma automática el precio exacto de la gasolina a través de una aplicación móvil o una página web. La metodología se divide en 2 etapas: i) implementar el detector de precios de la gasolina, y ii) desplegar una función en la nube de Azure y almacenar el precio de la gasolina en una base de datos. De acuerdo con los resultados obtenidos se concluye que el detector reconoce los precios de la gasolina con una exactitud del 90%, lo cual es competitivo con los resultados presentados en la literatura. Además, las transferencias a la nube Azure de los precios detectados fueron 100% exitosas si la conexión a Internet está activa.

Palabras clave: Computación en la Nube, Reconocimiento Óptico de Caracteres, Flexibilidad, Escalabilidad.

Gasoline Price Detector Based on Machine Vision and Cloud Computing

Abstract. Optical character recognition within vision systems allows to automate information and processes in which the intervention of the human being is frequently questioned. The case of the manual update of gasoline prices at fuel service stations has caused inconsistencies between the real price and the one published on platforms by the Energy Regulatory Commission (ERC) in Mexico.

This paper aims to present an automatic price detector for gasoline offered at a service station using a digital camera and a procedure to send the information to the Microsoft Azure cloud. Thus, it will be possible to show automatically the users the exact price of gasoline through a mobile application or a web page. The methodology comprises two stages: i) implement the gasoline price detector, and ii) deploy a function in the Azure cloud and store the price of the gasoline in a database. According to the results obtained, we concluded that the proposed detector recognizes gasoline prices with an accuracy of 90%, which is competitive with the results presented in the literature. Furthermore, transfers to the Azure cloud of the detected prices were 100% successful if the Internet connection is active.

Keywords: Cloud Computing, Optical Character Recognition, Flexibility, Scalability.

1. Introducción

Muy jocoso y antagónicamente cautivador, resulta pensar que “la fábrica del futuro tendrá dos empleados: un humano y un perro. La labor del humano será dar de comer al perro y la del perro, evitar que el humano toque los sistemas automatizados” [1]. En efecto, una de las tendencias actuales de la tecnología se basa en los pilares de la industria 4.0 [2-4]. La industria 4.0 o Cuarta Revolución Industrial, es un conjunto de conceptos con influencia sustancial en el sector manufacturero [5]. Uno de los conceptos más importantes y que las diversas industrias buscan adoptar es la computación basada en “la nube” [3, 4].

La revolución industrial 4.0 ha capturado la atención de industrias en todo el mundo, incluyendo al sector petrolero y de gas [6]. Dicho sector se ha incorporado a la nueva era de la informatización y las comunicaciones. Sin embargo, la transición hacia la industria 4.0 ha sido paulatina y conservadora respecto a los servicios financieros y el sector de las telecomunicaciones [6]. Al parecer, todo apunta a que la transición paulatina se debe fundamentalmente a problemas de seguridad y protección de los datos que aún se presentan como brechas tecnológicas.

Los sistemas de visión han ocupado un puesto importante dentro de los nuevos paradigmas actuales que se han establecido. Los problemas en áreas como la biotecnología, la robótica y la realidad aumentada han protagonizado atractivas soluciones basadas en visión cada vez más complejas y sofisticadas [7, 8]. Una de las industrias que ha implementado soluciones basadas en sistemas de visión es la transportación [9]. En algunos países como Estados Unidos se ha planteado un desafío para contrarrestar el aumento en los accidentes y el incremento en las pérdidas económicas, especialmente las devenidas del aumento en el consumo de combustibles [9]. Una alternativa para combatir dicho desafío es implementar sistemas con sensores en los lugares donde se puede observar mayor actividad del transporte terrestre. Sin embargo ¿qué utilidad, por ejemplo, pudieran tener dichos sensores en las gasolineras que forman parte de la cadena de valor de la industria petrolera más allá de su rol tradicional como complemento para la vigilancia y la supervisión del proceso?

Aun inmersos en una nueva era industrial, en los establecimientos minoristas de distribución de combustibles no se han implementado simultánea y masivamente, las tendencias actuales de la tecnología, especialmente los sistemas de visión y la

computación en la nube. Particularmente, la información relativa al precio del producto en una gasolinera actualmente no está disponible en la red pública de forma automatizada, y el usuario sólo la puede verificar in situ. Queda de parte de propietarios e inversionistas comprender que la descentralización de los precios de la gasolina en muchos países está inclinándose cada vez más la balanza del lado del cliente [10], en donde es posible apostar por una cartera de servicios adyacentes o bien reinvertir en nuevos activos que motiven y estimulen el interés del consumidor.

Los datos que se pueden obtener con un detector ofrecen información relevante a los consumidores como precios y disponibilidad de productos. Es posible incrementar el poder de decisión de los clientes poniendo a su disposición dicha información a través de una aplicación móvil o un sitio web. Con el uso de los recursos que ofrece la nube pública es posible intercambiar información con el usuario rápidamente.

El resto del artículo se encuentra organizado de la siguiente manera: en la Sección 2 se presentan los métodos utilizados para la generación del detector de precios de gasolina. En la Sección 3, se presenta la validación del sistema propuesto utilizando 25 imágenes. Además, se presenta la metodología para almacenar los resultados obtenidos en una base de datos en la nube. Por último, en la Sección 4, se presentan las conclusiones del trabajo de investigación.

2. Materiales y métodos

La metodología para implementar el detector de precios de la gasolina se muestra en la Fig. 1. Además, en la Tabla 1 se presentan los parámetros y los rangos que se utilizaron para validar el sistema. El primer paso consiste en obtener una representación RGB (rojo, verde y azul) de una imagen que representa un cartel de precios real. El preprocesamiento incluye una conversión a escala de grises que reduce a 256 colores la representación matricial de la instantánea. El filtrado se aplica para eliminar ruido convolucionando una matriz cuadrada de Gauss de dimensión $R=5$ (parámetros $blur_1$ y $blur_2$ en la Tabla 1) con cada píxel de la imagen en gris [11]. La imagen resultante del filtrado se pasa al detector de bordes *Canny*, que utiliza la dirección del gradiente de la intensidad de los píxeles, la cual es perpendicular a los bordes [12]. Además, se emplea umbralización por histéresis para determinar qué aristas son realmente bordes. Finalmente, una función basada en la mediana de la intensidad de los píxeles calcula los valores máximo y mínimo del umbral.

La etapa del seccionado de los precios se implementa usando las coordenadas X e Y del punto inicial y final (polígono de cuatro lados que representa la sección de cada precio). El seccionado de la imagen se implementa dentro de un ciclo iterativo *for* y la cantidad de secciones o ciclos depende de la posición de los precios en la imagen. Una vez que se han seccionado los precios, se procede al proceso de binarización adaptativa, por lo que se determina el umbral de un píxel en función de su vecindad. Así es posible obtener umbrales para diferentes regiones de la misma imagen, lo que brinda mejores resultados cuando la iluminación es variable [13]. El resultado permite obtener una imagen cuyo fondo es negro con los caracteres o precios de color blanco (matriz de 1 y 0 con binarización invertida).

Después, se dibuja un marco para eliminar falsos contornos externos. El grosor del marco está definido por el parámetro *lineThickness* de la Tabla 1.

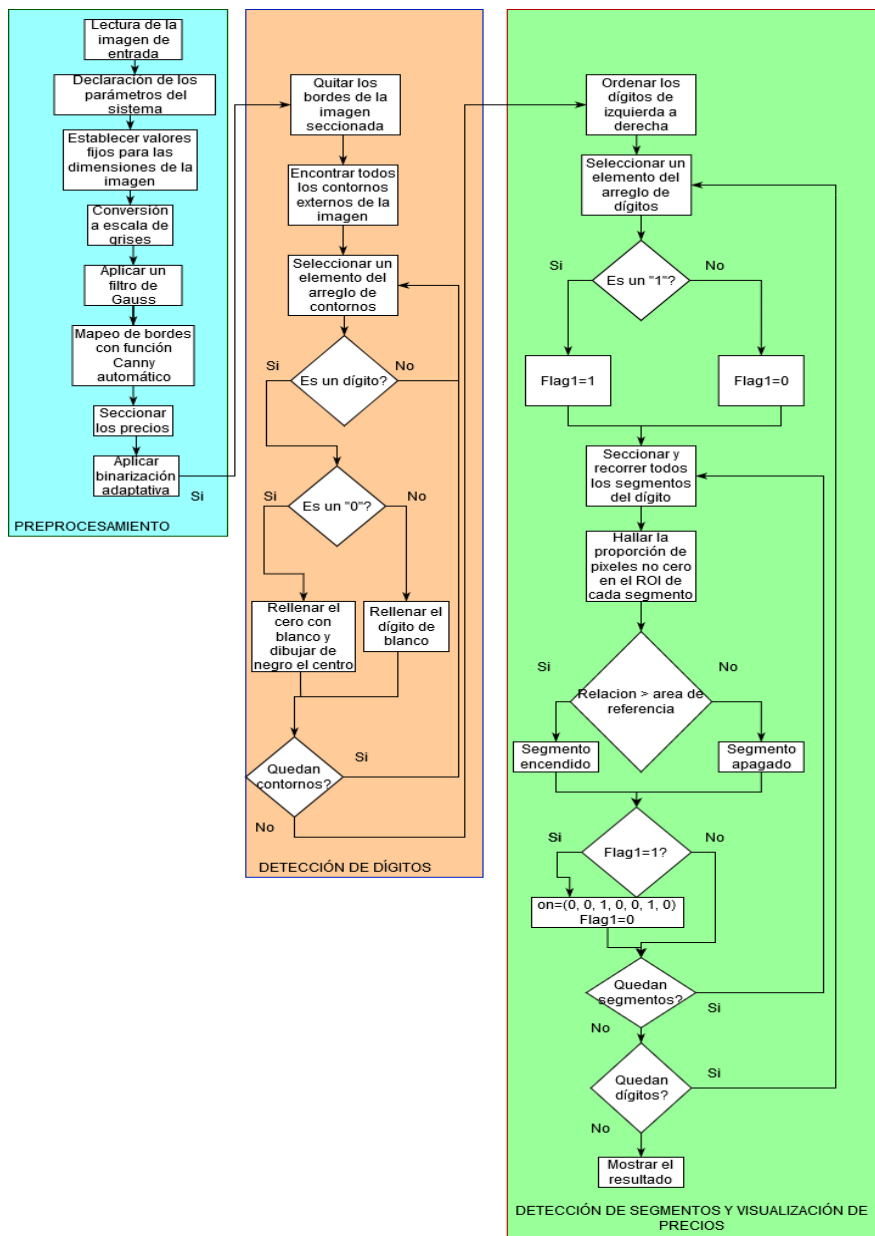


Fig. 1. Diagrama de flujo para el detector de precios de la gasolina.

La etapa de detección y clasificación comienza después de encontrar todos los contornos en la imagen resultante. Posteriormente, se implementó una rutina para encontrar todos los contornos externos y no hacer ninguna aproximación de los puntos que los forman.

Tabla 1. Resumen de parámetros y valores utilizados para implementar el detector de precios.

Parámetros	Condición ajustable	Mínimo valor	Máximo valor
$arearef_{min}$	si	0.31	-
$arearef_0$	si	0.07	-
dw_{ref}	si	0.25	-
dh_{ref}	si	0.15	-
w_1	si	0	25
$flag_1$	no	0	1
		1 1 1 0 1 1 1 : 0	
		0 0 1 0 0 1 0 : 1	
		1 0 1 1 1 0 1 : 2	
		1 0 1 1 0 1 1 : 3	
$Digits_{Lookup}$	no	0 1 1 1 0 1 0 : 4	-
		1 1 0 1 0 1 1 : 5	
		1 1 0 1 1 1 1 : 6	
		1 0 1 0 0 1 0 : 7	
		1 1 1 1 1 1 1 : 8	
		1 1 1 1 0 1 1 : 9	
$thresh_1$	si	3	5
$blur_1$	si	25	45
$blur_2$	si	5	25
$color_{fill}$	no	255,255,255	-
$color_{fill_cero}$	no	0,0,0	-
$w_{descriptor}$	si	8	70
$h_{descriptor}$	si	45	105
$line_{Thickness}$	si	10	20

El cuadro delimitador (*bounding box*) de cada contorno se obtiene por medio de una rutina que aproxima un rectángulo alrededor de la imagen binaria, y por medio de un ciclo *for* se recorren todos los bordes encontrados.

Cada contorno es clasificado en dígitos o no, utilizando un descriptor basado en las dimensiones del delimitador. El descriptor evalúa si las dimensiones de (ancho (w) y alto (h)) están dentro de los valores establecidos en la Tabla 1 ($w_{descriptor}$ y $h_{descriptor}$). Si la condición se cumple se considera que el contorno actual es un dígito. Si el dígito es un cero, se cumplirá que la relación $\frac{\text{píxeles centrales no cero}}{\text{área del segmento central}} > arearef_0 = 0.07$. Se utiliza una rutina para rellenar el espacio del cero con blanco y posteriormente dibujar de negro solo el centro, como se muestra en la Fig. 2. Si el dígito no es un cero, entonces se rellena el interior de todos los contornos de blanco.

Todos los dígitos encontrados son ordenados de izquierda a derecha y son nuevamente seccionados para obtener todos los segmentos correspondientes (seccionado de segmentos del dígito). Para ello se utilizan los parámetros dw_{ref} y dh_{ref} que representan porcentajes del ancho y alto del cuadro delimitador de cada dígito.

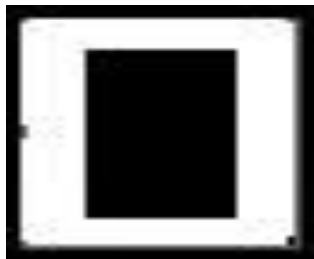


Fig. 2. Resultado del proceso de rellenado del dígito cero.

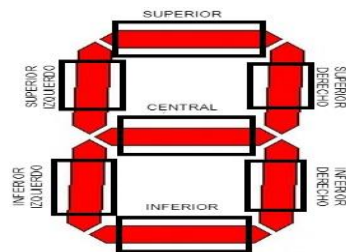


Fig. 3. Seccionado de los segmentos de cada dígito.

El nuevo seccionado permite obtener regiones o subsecciones por separado para cada uno de los siete segmentos del dígito, es decir: superior, superior izquierdo, superior derecho, central, inferior izquierdo, inferior derecho e inferior tal y como se muestra en la Fig. 3. Con cada segmento se implementa un ciclo iterativo para determinar cuáles pixeles tienen valor 0 y cuáles 1. Debe destacarse que como la umbralización previa fue de tipo invertida, un segmento encendido tiene el color blanco, y sus bits son todos uno. A partir de una relación entre la cantidad de pixeles no equivalentes a cero y el área de la subsección, es posible conocer si el segmento está o no encendido. La cantidad de pixeles no equivalentes a cero se obtiene con la operación aritmética de suma de bits que no son cero.

La relación se puede ajustar con el parámetro $arearef_{min}$ (Tabla 1). Cuando la relación es superior al valor del parámetro, significa que más del 31% de los pixeles en la sección tienen valor uno, por lo que se asume que el segmento está encendido y se fuerza un bit a 1 ($on[i]=1$). Particularmente, cuando se trata del dígito 1 se evalúa la condición ($w_1 < 25$ según la Tabla 1) y se establece el parámetro $flag_1 = 1$. Así, en dicho caso es posible además establecer que: $on = 0\ 0\ 1\ 0\ 0\ 1\ 0$ (empaquetado binario del 1 escrito a 7 segmentos).

Luego, se recorre el arreglo de segmentos y se repite el proceso para los 6 segmentos restantes. Cuando se completan todas las iteraciones se busca el resultado en un arreglo de dígitos ($Digits_{Lookup}$ dígitos de siete segmentos empaquetados en formato binario). Una vez que los dígitos son identificados dentro del arreglo se muestra el resultado y se almacena en una variable. En la Fig. 4 se muestra una secuencia de imágenes obtenidas al aplicar el método de detección a un cartel de precios de Pemex. Las imágenes superiores corresponden a la etapa de preprocesamiento y las inferiores muestran el resultado de la detección y visualización de los precios de la gasolina.

2.1. Recolección de imágenes para la validación del detector de precios

Las 25 imágenes utilizadas para validar el detector de precios se obtuvieron de gasolineras en Ciudad Juárez, bajo condiciones de ruido críticas. Las imágenes contienen los precios de distintos proveedores. El número de establecimientos de venta en el municipio es 18. Además, con el grupo de imágenes obtenido, es posible someter al detector de precios a condiciones extremas. Algunas de las instantáneas utilizadas en la validación se muestran en la Fig. 5.

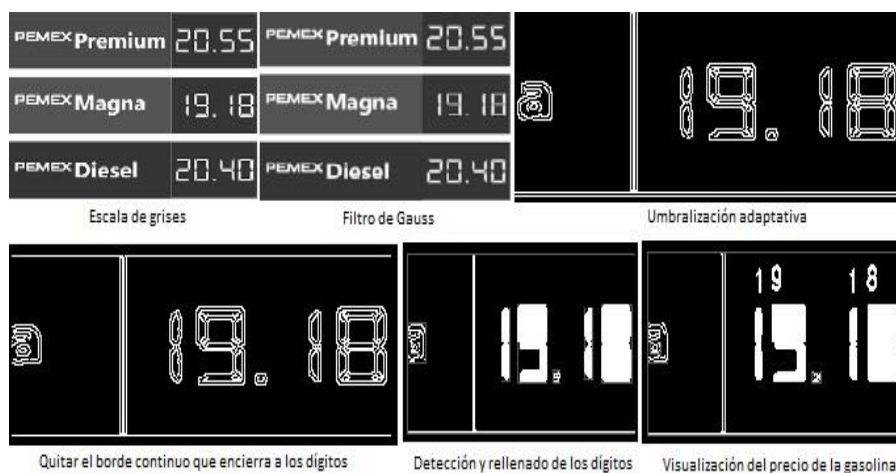


Fig. 4. Resultados del detector de precios.



Fig. 5. Ejemplo de imágenes utilizadas en la validación del detector.

2.2. Detección de los precios de la gasolina para un caso base

Después de implementar el detector se procede a la validación. Para ello se utiliza una construcción digital de un cartel de precios real la cual se utiliza como caso base (ver Fig. 6). Una vez que el detector converge al resultado esperado se valida con 25 imágenes reales y los resultados se muestran en la Sección 3.

El resultado obtenido para los precios de la gasolina Premium y Magna respectivamente se muestra en la Fig. 6, sección b y c. Note que se representan exactamente las cantidades o precios impresos en el cartel de la Fig. 6, sección a. Justo encima de cada dígito se ha representado el valor correspondiente obtenido del arreglo *Digits_{Lookup}*. Los dígitos han sido rellenos con pixeles equivalentes a uno.

Es importante destacar que en ambas figuras ha sido necesario eliminar el marco o borde que encerraba a los precios. Dicho marco hubiese impedido la detección de contornos con la configuración CV_RETR_EXTERNAL. Los precios obtenidos con el detector se asignan a variables para ser posteriormente gestionados y almacenados en una infraestructura de nube completa como Microsoft Azure.

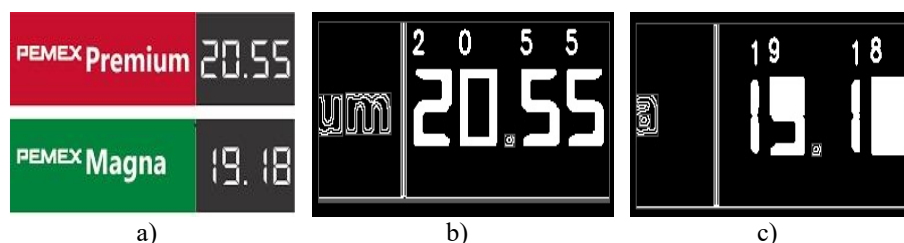


Fig. 6. Ejemplo de detección de precios. a) Cartel usado como base, b) detección del precio para gasolina Premium, c) detección del precio para gasolina Magna.

Tabla 2. Tabla *dbo.gasolinera* dentro de la DB en Azure.

id	PrecioMagna	PrecioPremium
----	-------------	---------------

2.3. Metodología para almacenar datos en la nube Azure

El envío de datos a la nube permite mostrar los precios a clientes potenciales sin la necesidad de crear una infraestructura física. El método de presentación de la información puede ser escalable y automatizado a partir de: i) Crear una cuenta de Azure mediante el vínculo: azure.microsoft.com, ii) Crear una Base de Datos (BD) en Azure usando el editor de consultas SQL. Dentro de la BD se crea una tabla con el formato que se muestra en la Tabla 2, iii) Programar y desplegar una función en lenguaje Java en la nube desde el entorno de desarrollo integrado (IDE) Visual Studio Code (VS Code) para almacenar los precios en la BD, y iv) Programar un método en el cliente remoto para invocar la función desplegada en la nube Azure.

La función en lenguaje Java para almacenar los precios en la nube se programa en VS Code y se despliega usando la línea de comandos en la terminal del IDE. Cabe destacar que la función se despliega en modo solo lectura por lo que los cambios se deben hacer necesariamente desde el VS Code (flexibilidad reducida). Para desplegar la función es necesario instalar Maven en la máquina local. Cuando se ha completado el código de la función es posible desplegarla en Azure escribiendo los comandos: `az login` (paso de autenticación) y `mvn azure-functions:deploy` (despliegue).

Finalmente, se debe programar un método en el cliente remoto (detector de precios) para invocar a la función Java desplegada en la nube Azure. El método utiliza el formato *Java Script Object Notation* (JSON) para enviar los precios de la gasolina a la nube y desencadenar el código de la función. Por ello es importante disponer de la dirección url de la función, la cual puede ser obtenida en el portal de Azure.

3. Experimentación y resultados

Los resultados obtenidos en la validación del detector se muestran en la Tabla 3. Durante la fase de experimentación se realizaron 50 corridas base para obtener el precio de la gasolina Magna y Premium respectivamente. La efectividad del detector corresponde con los aciertos totales en todas las imágenes. En el proceso de detección se empleó la función de relleno de espacios debido a que se detectó que algunos

PrecioMagna	PrecioPremium
12.35	14.57
19.59	20

Fig. 7. Consulta a la DB en Azure.

Tabla 3. Resultados obtenidos en la validación del detector.

No. de imágenes	Aciertos Magna	Aciertos Premium	Efectividad
25	23	22	90%

Tabla 4. Transferencia de datos a la nube.

Envío de paquetes de datos a la nube	Cantidades totales enviadas a Azure	Cantidades recibidas en Azure
Total de transferencia para el detector de precios (paquete Magna+Premium)	23 paquetes (45 precios)	45 precios

dígitos binarizados presentaban “agujeros” que impedían estandarizar el descriptor de segmentos. El caso del número 1 se implementó con la bandera $flag_1$ dado que al obtener el delimitador del contorno, sólo se lograban capturar dos segmentos del dígito. Por lo que, era imposible clasificarlo correctamente. Sin embargo, no fue necesario aumentar el vector de características o completar el delimitador con una reconstrucción para abarcar los 7 segmentos. En la parametrización del detector, las dimensiones de campana y el tamaño de la vecindad de píxeles $thresh_1$ se obtuvieron de forma empírica a partir de la evaluación del conjunto de imágenes. En el caso del detector *Canny*, los valores de los umbrales mínimo y máximo se obtuvieron con una umbralización automática que encuentra el mejor valor del umbral utilizando la mediana de la intensidad de los píxeles.

Finalmente, para validar el despliegue de los precios de la gasolina en la DB en Azure, se enviaron dos pares de precios arbitrarios tal y como se muestra en la Fig. 7. La transferencia de los precios fue exitosa según los resultados obtenidos en el editor de consultas de Azure. Luego y según los datos mostrados en la Tabla 4, se enviaron a la nube los 45 precios detectados de los cuales se almacenaron en la DB el 100%.

3.1. Discusión

Las imágenes utilizadas en la validación contienen ruido y características que complejizan la detección del precio de la gasolina. En los casos que no se pudo detectar el precio de la gasolina interviene la presencia de ruido que no se pudo atenuar con una campana de Gauss de dimensiones superiores a 25 píxeles.

Por otro lado, el color rojo de algunos dígitos hace que desaparezcan cuando se binariza la imagen por lo que no se pudieron detectar los precios. Finalmente, se suscitaron algunos casos donde la presencia del punto decimal cerca del dígito afectó la segmentación y por lo tanto, no se pudo detectar el precio.

Durante la etapa de almacenamiento en la nube se comprobó que las transferencias son exitosas y fiables siempre que exista una conexión activa a Internet. Sin embargo, uno de los inconvenientes del despliegue en la nube es la necesidad de desarrollar localmente la función Java antes de publicarla. Los entornos de ejecución en el portal tienen algunas limitaciones cuando el lenguaje es diferente de C#.

4. Conclusiones

En el presente artículo se mostró una nueva tecnología en el marco de las transformaciones tecnológicas para automatizar la detección y almacenamiento en la nube del precio de la gasolina. El detector permite obtener de forma automatizada el precio con una efectividad del 90% siempre y cuando los dígitos estén escritos a 7 segmentos. La legibilidad del cartel de precios de la gasolina es un factor importante por considerar lo que supone la utilización de una cámara con buenas prestaciones. Las transferencias de datos a la nube son fiables y le proporcionan acceso ubicuo a la información relevante para los clientes.

A futuro se propone utilizar y comparar con otros métodos de detección de caracteres para evaluar el desempeño del sistema. Para mejorar el desempeño del detector se propone como trabajo futuro, probar con otros algoritmos de suavizado como: filtrado bilateral, difusión anisotrópica o filtro kuwahara. Para el caso de las fallas ante dígitos de color rojo se pretende comparar en futuras investigaciones el modelo de colores RGB con otro diferente como por ejemplo el HSV o CIELAB.

Referencias

1. Sáenz, C.C.: *Industria 4.0* (Tesis de maestría), Universidad de la Rioja, España (2016)
2. Dodgson, M., Gann, D.M., Salter, A.: *The management of technological innovation: strategy and practice*. Oxford University Press on Demand (2008)
3. Ibarra, D.: Business model innovation through Industry 4.0: A review. *Procedia Manufacturing*, 22, pp. 4–10 (2018)
4. Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., Harnisch, M.: *Industry 4.0: The future of productivity and growth in manufacturing industries*. Boston Consulting Group, 9, pp. 54–89 (2015)
5. Stock, T., Seliger, G.: Opportunities of sustainable manufacturing in industry 4.0. *Procedia Cirp*, 40, pp. 536–541 (2016)
6. Perrons, R.K., Hems, A.: Cloud computing in the upstream oil & gas industry: A proposed way forward. *Energy Policy*, 56, pp. 732–737 (2013)
7. Ali, L., Khan, W., Chaiyasarn, K.: Damage detection and localization in masonry structure using faster region convolutional networks. *Int. J. Geomate*, 17, pp. 98–105 (2019)
8. Shahid, L., Janabi-Sharifi, F., Keenan, P.: A hybrid vision-based surface coverage measurement method for robotic inspection Robot. *Comput.-Integr. Manuf.*, 57, pp. 138–145 (2019)
9. Walid, B., Refai, H.H.: Intelligent vehicle counting and classification sensor for real-time traffic surveillance. *Transactions on Intelligent Transportation Systems*, 19 (2018)
10. Fernández, G., Almajano, J., García, E., Bludszweit, H., Machín, S., Sanz, J.F.: Control structure for optimal demand-side management with a multi-technology battery storage system. In: *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, pp. 754–759 (2019)

11. Hummel, R.A., Kimia, B., Zucker, S.W.: Deblurring Gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38(1), pp. 66–80 (1987)
12. Bao, P., Zhang, L., Wu, X.: Canny edge detection enhancement by scale multiplication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9), pp. 1485–1490 (2005)
13. Bradley, D., Roth, G.: Adaptive thresholding using the integral image. *Journal of Graphics Tools*, 12(2), pp. 13–21 (2007)

Reentrenamiento continuo de un algoritmo LSTM con un sistema multiagente que implementa el modelo BDI para la predicción de viajes en bicicleta

Ramon A. Briseño, Juan C. López, Rocio Maciel Arellano,
Víctor M. Larios, Raul J. Beltrán, J. Antonio Orizaga T.

Universidad de Guadalajara,
Centro Universitario de Ciencias Económico Administrativas,
Doctorado en Tecnologías de Información,
Centro de Innovación en Ciudades Inteligentes,
México

{alejandrobmartinez, juan.lopez6936}@alumnos.udg.mx,
{raul.beltran, jose.orizaga}@academicos.udg.mx,
{rmaciel, vmlarios}@cucea.udg.mx

Abstract. En este documento se describe el desarrollo un sistema multiagente, con un modelo BDI encargado de mantener entrenado continuamente un algoritmo de redes neuronales recurrentes de corta y larga memoria, el cual predice por día la cantidad de viajes en sistemas de bicicleta pública para el desarrollo del proyecto de la Smart City de la Zona Metropolitana de Guadalajara. Para los algoritmos de aprendizaje automático, es necesario someterlos a reentrenamiento cada que el conjunto de datos aumenta de manera considerable o en tareas en las que se requiere análisis de datos para realizar predicciones en tiempo real. El proceso consiste en la obtención, depuración y estructuración de nueva información, reentrenamiento del algoritmo de aprendizaje automático con este nuevo conjunto de datos y el reporte de resultados. Con algunas adecuaciones al modelo Belief-Desire-Intention (BDI) mediante la plataforma multiagentes SPADE, se implementó un agente que tiene como propósito ejecutar el proceso de aprendizaje automático del algoritmo para la predicción. Este agente calibra de manera automática la longitud de la secuencia de entrada que requiere el algoritmo. Utilizando el modelo BDI, el agente predictor consigue que el algoritmo mantenga de manera sistemática una precisión mayor al 85% en la exactitud, contando con la capacidad de aumentar su eficacia en un 5% con un módulo en otro agente planificador basado BDI. Este sistema multiagente, demostró ser modular, esacleable, reutilizable y con la autonomía suficiente para efectuar una predicción confiable sin necesidad de agregar más desarrollo de software después de su implementación. Además, demostró que el modelo BDI en los sistemas multiagentes, es una herramienta poderosa para interactuar con algoritmos de aprendizaje automático.

Keywords: Datos Masivos, Minería de Datos y Analíticos, Red Neuronal LSTM, Predicción, Sistema Multiagente con Modelo BDI, Movilidad Sustentable, Smart Cities, Tecnologías de Información, E-World.

1. Introducción

Con la urbanización acelerada, uno de los problemas más palpables está asociado al incremento de tráfico de automóviles privados saturando las vialidades y con ello un incremento en la contaminación ambiental. Una estrategia para combatir este problema, es incentivar movilidad limpia en lo posible, a través de infraestructura de ciclovías y con una red de estaciones en un sistema de bicicleta pública. Este es el caso de la Zona Metropolitana de Guadalajara (ZMG). Sin embargo, la adopción nos es fácil, sobre todo si las ciclovías no son seguras y en Guadalajara se ha sufrido un incremento de muertes de ciclistas por atropellamiento aumentando este índice en un 54% comparando el año 2018 con el 2019 [8].

Actualmente, la ZMG cuenta con casi 100 Km de Ciclovías y para que esta solución se integre al proyecto de Smart City de la región [9], es necesario que se puedan simular diferentes escenarios para que el sistema sea más seguro y atractivo. El programa MiBici es como se denomina la red de estaciones de bicicletas públicas ZMG. Las bicicletas están disponibles para su renta por día o por año mediante una membresía. Se realizaron alrededor de 4,658,208 viajes en el 2019, con un total de 85,143 usuarios registrados en el programa hasta junio de 2020 [7]. El programa MiBici publica en su sitio web, datos abiertos sobre la cantidad de viajes que se efectúan por día, por mes y por año. El programa MiBici se toma como objeto de estudio de esta investigación, dada su importancia en el uso de medios de transporte sustentable.

Este trabajo propone escalar este sistema de bicicleta pública en una primera etapa para poder hacer predicciones de operación que lo haga más eficiente y responsivo a las necesidades de los ciudadanos. Para esto, integraremos una solución de simulación con multiagentes que se entrenarán a través de los datos reales producidos en la red de bicicletas públicas para generar optimizaciones. Un agente es una unidad de software autónoma [2], la toma de decisiones y la comunicación son algunas de sus principales características, también se pueden organizar con otras unidades de software, para realizar tareas conjuntas.

Por lo que, un sistema multiagente (MAS) es un entorno donde más de un agente convive y trabaja de manera coordinada para resolver un objetivo en común [3]. Los sistemas MAS han incorporado el aprendizaje automático y al análisis de la movilidad, durante la revisión del estado del arte, destaca un sistema multiagente en el programa de sistema de bicicleta compartida en salamanca [1], este predice la demanda de bicicletas y prevé una visualización en tiempo real de los viajes en curso.

El sistema MAS descrito en este trabajo se encarga de predecir la cantidad de viajes por día que cada estación tendrá como punto de finalización e inicio de un viaje, esta cantidad de viajes resultante de la predicción será analizada para una posterior identificación e inclusión de variables a un modelo de predicción de accidentes de movilidad para ciclo vías de la ZMG. El MAS desarrollado obtiene y estructura información del portal MiBici, reentrena un algoritmo de redes neuronales recurrentes de corta y larga memoria (LSTM) y finalmente postea resultados de la predicción de viajes.

Este sistema utiliza el modelo BDI (Believe, desire and intentions) para dotarlo de información que le permite calibrar la longitud de la secuencia de entrada de manera

autónoma, con el fin de que el sistema ofrezca de manera sistemática una exactitud mayor al 85% en su precisión a lo largo del tiempo y sin la intervención del diseñador del software.

2. Propuesta para la predicción de viajes en bicicleta

Este presente trabajo de investigación describe, la automatización de un proceso de aprendizaje automático para un algoritmo de predicción y la interacción que tiene el MAS con el algoritmo (LSTM) de red neuronal recurrente de corta y larga memoria al ajustar la longitud de la secuencia de entrada, cada agente está diseñado para tomar decisiones de acuerdo a lo establecido en su modelo BDI.

En primer término, el sistema obtiene la información necesaria mediante *scraping*¹, después ejecuta tareas de normalización y depuración de datos no relevantes para generar un conjunto de datos estructurados, este conjunto de datos es sometida a entrenamiento de un algoritmo, el resultado es la predicción de la cantidad de viajes por día que tendrán las estaciones del programa MiBici de la ZMG, el sistema multiagente calibra la longitud de la secuencia de entrada del algoritmo de aprendizaje automático y algunos parámetros del algoritmo como la cantidad de neuronas y épocas, esto se logra a partir del modelo BDI de su arquitectura, La autonomía de un agente generalmente significa que un agente opera sin ntervención u orientación humana directa (u otra) [12], este modelo hace que la toma de decisiones de los agentes sea más cercana a la que tomaría el diseñador del sistema, ya que este plasma su actuar en los planes del modelo [4], lo que le da una mayor sensibilidad al sistema.

Para la predicción el MAS, utiliza un algoritmo de red neuronal recurrente de corta y larga memoria (LSTM), utilizado generalmente para el análisis y predicción de series temporales donde se observa un mejor desempeño en la precisión comparado con otros algoritmos de aprendizaje automático [10]. El experimento contempló una serie de tiempo de una sola variable “la cantidad de viajes por día”. Para la predicción, el algoritmo recibe una secuencia de valores de viajes por día como entrada y arrojará un solo valor de salida, mismo que representa la cantidad de viajes que tendrá el día posterior a los días introducidos como entrada.

El MAS se desarrolló en la plataforma SPADE [5], un entorno de desarrollo en Python que además permite escalar de forma distribuida conectando nodos de procesamiento según sea necesario. El modelo BDI desarrollado funciona para la plataforma SPADE 3 y Python 3, mientras que el algoritmo LSTM es ejecutado con Keras y Tesorflow.

2.1. Arquitectura del sistema multiagente

El sistema multiagente consta de 5 agentes, un agente para la recolección de datos, uno para la depuración, uno para el reporte y dos para gestionar la predicción mediante

¹ *Scraping*, o web scraping es una técnica para obtener código html de paginas web de manera automática, se procesan los documentos html obtenidos, se navegan por las etiquetas y se guardar la información requerida.

el modelo BDI. El agente scraper, el agente depurador, el agente predictor y el agente reporteador siempre están vivos a la espera que suceda un evento o de recibir una notificación para realizar su tarea.

La secuencia de eventos se describe a continuación:

1. El agente scraper monitorea el portal web del programa Mibici de manera periódica (una vez al día), una vez que encuentra nuevos datos abiertos de viajes, los descarga a una carpeta del servidor de donde se ejecutó MAS. Cada que el agente scraper descarga nueva información notifica al agente depurador.
2. Después de recibir la notificación del agente scraper, el agente depurador se encarga de formar un conjunto de datos con la información en la estructura que el algoritmo LSTM necesita. Si ya existe un conjunto de datos el agente depurador solo depura la información nueva y la agrega al conjunto de datos ya existente. Una vez que el conjunto de datos contiene la información nueva el agente depurador notifica al agente predictor.
3. Después de recibir notificación del agente depurador, el agente predictor crea al agente BDI.
 - (a) El agente BDI se encarga de entrenar y encontrar la longitud de la secuencia de entrada indicada para obtener una buena predicción. Una vez entrenado el algoritmo y encontrado la longitud de la secuencia de entrada el agente BDI escribe los resultados de la predicción en diferentes archivos en la carpeta del sistema, después notifica al agente reporteador y muere, el agente predictor sigue a la espera de nuevas notificaciones.
4. Después de recibir notificación del agente BDI, el agente reporteador muestra los resultados de la predicción en un sitio web, donde se muestran entre otra información estadística, la gráfica de validación con la que se calculó el MAPE y la predicción correspondiente al día posterior a los datos iniciales.

En la figura 1 se observa el diagrama de funcionamiento del MAS. A continuación, se detalla el funcionamiento de cada agente.

Agente scraper: Los navegadores web son útiles para ejecutar código javascript, mostrar imágenes y presentar la información en un formato de fácil lectura para las personas, sin embargo, los web scrapers son excelentes para obtener y procesar largas cantidad de información rápidamente, es decir, que en lugar de estar revisando una página web a la vez, es posible estar visualizando los datos de miles de páginas web al mismo tiempo [11]. La búsqueda de información se realiza de manera periódica a la web informativa del programa MiBici, dentro de su sección de blog se publican en ligas de descarga archivos csv con los datos organizados por año y meses. El agente scraper ejecuta una función de consulta a esa web en busca de nueva información. Para obtener las ligas de decarga se utiliza un complemento html para la separación del documento en sus etiquetas.

Si el agente encuentra uno o más posts con datos de viajes que aún no se encuentran en la carpeta datasets del sistema multiagente, los descarga y los guarda. La

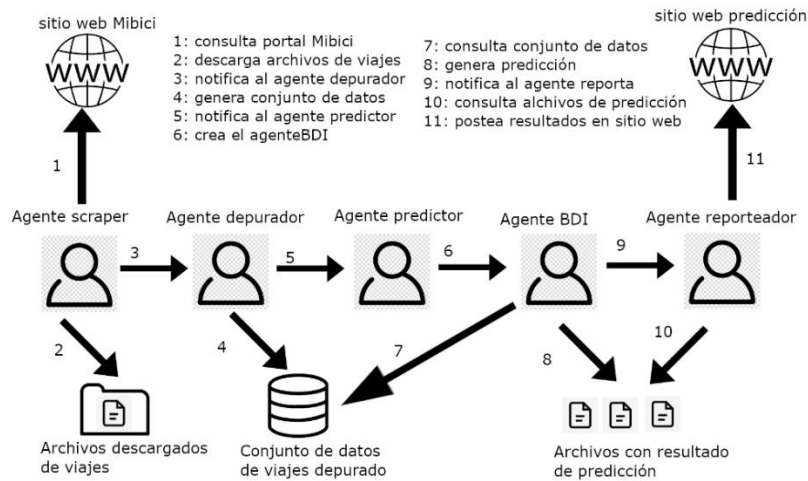


Fig. 1. Ilustración que representa el funcionamiento y la arquitectura del MAS.

información se organiza en carpetas por año y por meses, finalmente los archivos descargados se renombran y se guardan en formato csv.

Agente depurador: Este agente se encuentra todo el tiempo esperando la notificación del agente scraper. Una vez notificado, el agente revisa si hay un conjunto de datos depurado, si este no existe, el agente lo genera con todos los archivos descargados por el agente scraper. Si ya existe un conjunto de datos depurado, el agente verifica si hay archivos nuevos descargados por el agente scraper que no estén incluidos en el conjunto de datos, si se encuentran archivos nuevos, el agente los depura y agrega dicha información al conjunto de datos unificado y depurado. La generación de los archivos finales de inicio y finalización de viajes por estación consiste en realizar un ciclo que revisa cada fila y paralelamente tiene un contador de viajes por cada día y generar un archivo con dos columnas, la primera columna con la fecha en tipo fecha y la segunda de tipo entero misma que contendrá la cantidad de viajes por cada día. El conjunto de datos se guarda como archivo en formato csv con los nombres de columnas fecha y viajes.

Agente predictor: Al igual que el anterior, este agente en todo momento se encuentra a espera de la notificación del agente depurador. Una vez recibida la notificación de que el conjunto de datos está listo para que el algoritmo de aprendizaje automático se entrene, el agente predictor crea al agente BDI. El agente predictor se mantiene en espera hasta que el agente BDI termina su tarea. Cuando el agente BDI termina su tarea también termina su ciclo de vida, por lo que el agente predictor regresa a la espera de una nueva notificación.

Agente BDI: Este es el único agente del sistema MAS que implemente el modelo BDI, de allí fue nombrado “agente BDI”. Las tareas de los demás agentes del sistema pueden

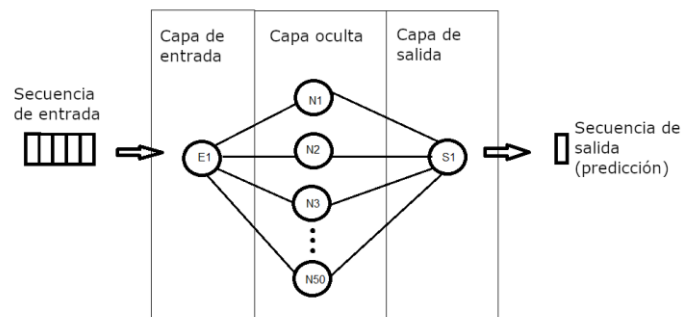


Fig. 2. Diagrama de bloques del algoritmo LSTM utilizado.

ser resueltas fácilmente con los tipos de comportamientos que ofrece la plataforma SPADE, como el comportamiento cíclico, el periódico, el comportamiento de una sola ejecución y el comportamiento de máquina de estados. Sin embargo, este agente necesita un planificador más inteligente para lograr el objetivo de entrenar el algoritmo de aprendizaje automático y encontrar la longitud de la secuencia de entrada indicada para que la predicción tenga un porcentaje de exactitud confiable.

En el modelo BDI implementado contiene 4 conceptos fundamentales. (1) **Belief**: el cual es un conocimiento que el agente tiene sobre su entorno. Se puede agregar, eliminar y consultar beliefs en el momento que se requiera. (2) **Goals**: es el objetivo que el agente perseguirá. En el caso del agente BDI el objetivo es encontrar la longitud de la secuencia de entrada indicada para que la predicción tenga un porcentaje de exactitud mayor al 85%. Originalmente el modelo BDI de la versión de SPADE 2.0 con python 2 [6] acepta múltiples objetivos. (3) **Services**: los servicios son pequeñas tareas que van agregando Beliefs a la base de conocimientos del sistema MAS. (4) **Plans**: estos son planes en los que se define una cantidad de servicios con los que se pretende llegar a cumplir el objetivo. Los servicios se agregan al plan en el orden que se desea que estos ejecuten su tarea. El modelo original pretendía que la tarea ejecutada por el último servicio fuera la que cumpliera con el objetivo. Sin embargo, en la adaptación para este trabajo se flexibilizó el sistema para que cualquier servicio pudiese cumplir el objetivo y omitir los servicios pendientes de un plan.

El sistema BDI implementado funciona ejecutando un ciclo el cual busca los planes y ejecuta los que tienen como finalidad el objetivo del agente. Una vez encontrado un plan este ejecuta todos los servicios que el plan contiene, dichos servicios van agregando beliefs a la base de conocimientos. Si alguno de los servicios de un plan alcanza el objetivo definido en el goal, el sistema para y el agente BDI notifica al agente reportador. Si ningún servicio de un plan logra el objetivo, el agente continúa con la ejecución de los servicios del siguiente plan encontrado, los servicios de los planes en ejecución utilizan los conocimientos que dejaron los servicios de los planes que fallaron. De esta forma cada plan que falla ayuda a que el siguiente plan tenga mayor probabilidad de llegar al objetivo y así de manera sistemática. El modelo BDI

adaptado para este trabajo se configuro para que en el momento que todos los planes fallen, se identifique el servicio obtuvo el mejor acercamiento al objetivo en términos porcentuales.

La primera implementación del algoritmo de aprendizaje automático, se definieron como parámetros los 11 últimos meses del conjunto de datos para validación, 20 épocas de procesamiento, 50 neuronas activas en una capa oculta, MAPE como métrica para medir confiabilidad de la predicción y una secuencia de valores de entrada con tamaño entre 1 a 50 valores, donde la primera secuencia utilizada fue de tamaño 5. En el agente BDI se consideraron 4 planes, 13 servicios y un objetivo. Con el objetivo de alcanzar un MAPE menor al 15%, la descripción de los planes, servicios y el objetivo quedaron de la siguiente forma:

- Goal: El objetivo es que al tener datos nuevos y reentrenar el algoritmo LSTM el MAPE sea menor al 15%.
- Plans: El agente cuenta con 4 planes. El primer plan cuenta con un servicio, los tres posteriores cuentan con 4 servicios cada uno. Al ser un agente con un solo objetivo todos los planes tienen el mismo objetivo en común.
- El primer plan consiste en entrenar el algoritmo con la longitud de la secuencia de entrada que se tiene definida por el diseñador del sistema. La longitud de entrada definida solo cambiará si este plan falla y se comienza con la búsqueda de un nuevo tamaño de secuencia de entrada. Este plan solo cuenta con un servicio. El servicio entrena el algoritmo con la longitud de la secuencia de entrada definido y obtiene la métrica MAPE. Si el MAPE obtenido es menor a 15% se alcanza el objetivo y termina la ejecución del agente BDI. Si el objetivo no es alcanzado, el servicio guarda los valores de MAPE obtenido y tamaño utilizado, así como un belief que confirma que el servicio se ejecutó con éxito a pesar de no cumplir el objetivo.
- El segundo plan consiste en entrenar el algoritmo aumentando y disminuyendo longitud de la secuencia de entrada definida por el diseñador del sistema. Este plan es clave, ya que en caso de no conseguir el objetivo dará información vital a los planes posteriores. El plan cuenta con 4 servicios. Los servicios consisten en entrenar el algoritmo con la longitud de la secuencia de entrada definida aumentada y disminuida en 10 y 20 unidades y obteniendo la métrica MAPE. Si el MAPE obtenido por algún servicio es menor al 15% se alcanza el objetivo y termina la ejecución del agente BDI. Si el objetivo no es alcanzado y los valores de MAPE y longitud de la secuencia de entrada de alguno de estos servicios dieron mejor resultado que el MAPE y longitud de la secuencia de entrada de los servicios anteriores estos se guardan. Proceso realizado de manera sistemática con cada servicio.
- Tanto el tercero como el cuarto plan consisten en entrenar el algoritmo con la mejor longitud de la secuencia de entrada hasta el momento, aumentada y disminuida en 6, 3, 2 y 1 unidades. Al igual que en el Segundo plan cada servicio revisa si cumple el objetivo y en caso de no hacerlo, verifica si sus resultados son los mejores hasta el momento con el fin de que estos sean usados por los servicios posteriores. Si ninguno de los servicios ofrecidos por los 4 planes logra

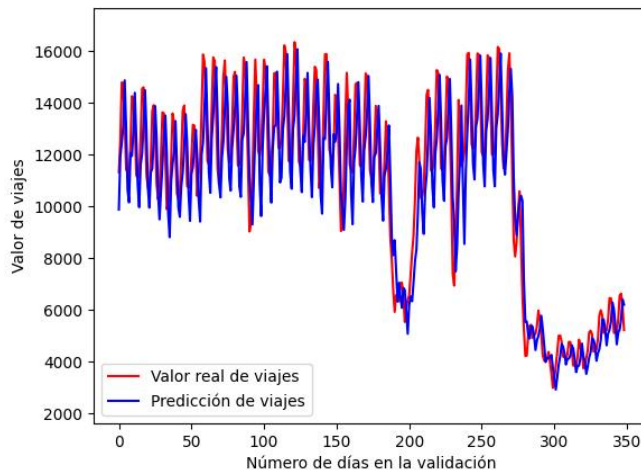


Fig. 3. Gráfica de los valores de predicción en contraste con los meses de validación del conjunto de datos (obtención de 11% en MAPE).

el objetivo de obtener un MAPE menor al 15% el agente entrena el algoritmo con la longitud de la secuencia de entrada que dio el resultado más acercado al objetivo y termina su ejecución. Cada que el agente termina su ejecución el valor de la longitud de la secuencia de entrada definida por el diseñador del sistema cambió al valor que alcanzo el objetivo o al que dio el mejor resultado, lo que provocará un incremento sistemático en la aproximación al objetivo para el siguiente agente BDI.

Agente reporteador: Este agente se encuentra en todo momento a espera de la notificación del agente predictor. Una vez recibida la notificación de que la predicción ha sido realizada ejecuta su tarea. La tarea del agente reporteador es presentar en una página web la predicción de viajes por estación y las gráficas generadas por el agente BDI donde se ilustra la confiabilidad de la predicción.

3. Pruebas y resultados

El conjunto de datos está integrado de 1917 registros correspondientes al periodo del 1 de diciembre del 2014 al 31 de mayo del 2020. Para disminuir el impacto negativo en la predicción causada por la dispersión y los datos atípicos que presenta la serie de tiempo, se le aplicó un suavizado con una media móvil con un tamaño de ventana en 3 valores. Se utilizó el 83% de los datos para el entrenamiento y el 17% de los datos para pruebas y validaciones. Para la reproducción del experimento, tanto el código fuente del modelo BDI creado y la arquitectura multiagente se encuentra en un repositorio Git donde los investigadores interesados pueden pedir acceso al mismo contactando a cualquier autor de este artículo.

Para encontrar la longitud de la secuencia de entrada definida con el que comenzara a correr el MAS en producción, el agente BDI se ejecutó 11 veces con longitudes de secuencias de entradas predefinidas con valores de 20 hasta 30 unidades. La longitud de la secuencia de entrada que arrojó la mejor métrica fue 5, obteniendo un MAPE del 11%.

4. Conclusión y trabajos futuros

El sistema multiagente demostró ser eficiente en la automatización del proceso de predicción de la cantidad de viajes por estación del programa MiBici. Aunado a eso, el MAS complementado con el modelo BDI tuvo la capacidad de interactuar con uno de los parámetros del algoritmo LSTM para buscar una predicción estable a lo largo del tiempo, obteniendo una exactitud en la precisión del 89% en las pruebas realizadas. El MAS, con la autonomía de cada uno de sus agentes, facilitó la ejecución continua de un proceso complejo sin la intervención del diseñador del sistema después de su implementación.

Al calibrar la longitud de la secuencia de entrada el agente BDI aumenta su posibilidad de alcanzar el objetivo de tener un MAPE menor al 15% en la precisión de la predicción, Sin embargo, no asegura que el objetivo sea alcanzado. Por otra parte, es muy útil, ya que en las pruebas realizadas para encontrar la longitud de la secuencia de entrada que tendría definida el MAS en su puesta en producción, se observó que de entre 1 a 50 unidades que pudo tomar la longitud de la secuencia de entrada, la predicción tiene una variación promedio del 5% en el MAPE, por lo que se concluye que si el valor definido de la longitud de la secuencia de entrada en algún momento no llegase a cumplir el objetivo, existe una ventana de aproximadamente el 5% de mejoramiento del MAPE ejecutando los planes del agente BDI.

Cabe remarcar que esta es la primera etapa de este proyecto para que el sistema público de bicicletas de las ZMG converja con otras soluciones del proyecto de Smart City logrando identificar las estaciones de la red y predecir como será su carga de uso. Esto puede servir para eficientar la logística de balanceo de cargas en estaciones para que siempre haya bicicletas disponibles. También para predecir densidad de uso de ciclovías y tener medidas para resguardar la seguridad en ellas cuando estas tienen más demanda mediante mayor supervisión de cámaras de video y agentes de tránsito. La vida de los ciudadanos no tiene costo y poder predecir es una contribución. Si bien estos dos ejemplos se pueden resolver con algoritmos de grafos y optimización tradicional, la solución propuesta se implementa más rápido y el aprendizaje automático de los agentes genera una solución más simple y de menor complejidad computacional como lo describe [13].

En cuanto a la técnica desarrollada, como trabajo futuro se pretende ampliar este trabajo donde el modelo BDI utilizado calibre de manera automática más parámetros del algoritmo de aprendizaje automático. Aunado a ello, se pretende utilizar una ontología que describa del funcionamiento del algoritmo de aprendizaje automático LSTM. Se busca que el agente BDI utilice dicha ontología para que las tareas que

realizan los servicios estén fundamentadas en el conocimiento del algoritmo otorgado por la ontología y se puedan cumplir objetivos de manera más eficiente e inteligente.

Referencias

1. Lozano, A., et al.: Multi-agent system for demand prediction and trip visualization in bike sharing systems. *Applied Sciences* 8.1, 67 (2018)
2. Oja, M., Tamm, B., Taveter, K.: Agent-based software design. *Proc. Estonian Acad. Sci. Eng.* 7(1), pp. 5–21 (2001)
3. Lee, L.C., et al.: The stability, scalability and performance of multi-agent systems. *BT Technology Journal*, 16(3), pp. 94–103 (1998)
4. Araiza-Illan, D., Pipe, T., Eder, K.: Model-based testing, using belief-desire-intentions agents, of control code for robots in collaborative human-robot interactions. *arXiv preprint arXiv:1603.00656* (2016)
5. SPADE: SPADE, Smart Python Agent Development Environment. <https://spade-mas.readthedocs.io/en/latest/readme.html> (2020)
6. Aranda, G., Palanca, J., Escriva, M., Criado, N., Garcia-Pardo, J.A.: SPADE User's Manual. <http://www.javierpalanca.com/spade/manual/> (2020)
7. MiBici: Datos abiertos. Explora, experimenta y comparte los datos de uso de MIBICI. <https://www.mibici.net/es/datos-abiertos/> (2020)
8. Carapia, F.: Periódico MURAL, Matan a más ciclistas pese a las ciclo vías. <https://www.mural.com/matan-a-mas-ciclistas-pese-a-las-ciclovias/ar1861611?referer=-7d616165662f3a3a6262623b727a7a7279703b767a783a--> (2020)
9. Gobierno de Zapopan: Presenta Pablo Lemus estrategia de ciclovías emergentes en sesión de Junta de Coordinación Metropolitana. <https://www.Guadalajara.gob.mx/v3/noticias/presenta-pablo-lemusestrategia-de-ciclovias-emergentes-en-sesion-de-junta-de-coordinacion> (2020)
10. Han, Z., et al.: A review of deep learning models for time series prediction. *IEEE Sensors Journal* (2019)
11. Mitchell, R.: *Web scraping with Python: Collecting more data from the modern web*. O'Reilly Media, Inc. (2018)
12. Wooldridge, M., Jennings, N.R.: Agent theories, architectures, and languages: a survey. In *International Workshop on Agent Theories, Architectures, and Languages*, Springer, Berlin, Heidelberg, pp. 1–39 (1994)
13. Chiariotti, F., et al.: A dynamic approach to rebalancing bike-sharing systems. *Sensors*, vol. 18(2), pp. 512 (2018)

Explicación visual de la predicción de un clasificador de imágenes

Tonantzin M. Guerrero Velázquez, Juan Humberto Sossa Azuela

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

tguerrero_a13@sagitario.cic.ipn.mx,
hsossa@cic.ipn.mx

Abstract. El aprendizaje automático se ha convertido en una herramienta necesaria en la industria actual. La mayoría de los métodos comúnmente usados, más allá de las métricas y pruebas realizadas en realidad no se comprenden las razones detrás de sus predicciones. En el trabajo se propone un nuevo método para entender de manera visual el trasfondo de una predicción realizada.

Keywords: Explainability, Classifier, XAI, Machine Learning.

Visual Explanation of the Prediction of an Image Classifier

Abstract. Machine learning has become a necessary tool in today's industry. Most of the commonly used methods, beyond the metrics and tests performed, don't actually understand the reasons behind their predictions. In the work, a new method is proposed to understand visually the background of a prediction made.

Keywords: Explainability, Classifier, XAI, Machine Learning.

1. Introducción

Actualmente la Inteligencia Artificial Explicable (XAI, por sus siglas en inglés) es un área de gran interés dentro del aprendizaje automático. Aunque es un campo relativamente nuevo su atractivo radica en la usabilidad que se le puede adjudicar. La XAI se refiere a mejorar el entendimiento humano y a justificar las decisiones hechas por un modelo de aprendizaje automático. La explicabilidad en el contexto de la Inteligencia Artificial denota cualquier acción o proceso tomado por un modelo con la intención de aclarar o detallar su funcionamiento. Por otro lado, la interpretabilidad es un concepto que muchas veces es tomado de la misma manera, sin embargo, se refiere

al nivel en el que un modelo dado tiene sentido para un ser humano, esta característica también se puede expresar como la transparencia de un modelo [1].

Un modelo se considera ser transparente si por sí mismo es entendible, por ejemplo, un modelo de regresión logística o lineal, un árbol de decisiones o un clasificador basado en reglas [2]. Existen diferentes enfoques de explicabilidad, uno de ellos se basa en entender mejor los datos a través de técnicas que permitan conocer información relevante de los mismos, tales como, mapas salientes, análisis de sensibilidad, etc. y de esta manera poder entender qué características de estos llevan a que un modelo realice una predicción u otra. El otro enfoque se basa en entender mejor el funcionamiento de un modelo, por ejemplo, visualizando la activación de las capas de una red neuronal [3].

Actualmente existen diversas técnicas que permiten obtener una explicabilidad de un modelo de aprendizaje profundo. Una de ellas es el marco de interpretabilidad LIME (*Local Interpretable Model-Agnostic Explanations*). Para explicar la predicción de un clasificador de imagen, LIME crea un conjunto de imágenes resultantes al perturbar la imagen de entrada, dividiéndola en componentes interpretables (*superpíxeles*) para así obtener una probabilidad de pertenencia para cada una de esas instancias perturbadas. Con este proceso, LIME genera una explicación visual con base en la clasificación de esos nuevos datos perturbados dando como resultado un área de la imagen de entrada que denota aquello en lo que se fijó el modelo para llevar a cabo la predicción [4].

Una de las desventajas más importantes de las técnicas de explicación visuales existentes es la subjetividad de los resultados. Estos se encuentran sujetos a la interpretabilidad de quien los utiliza, es de esperarse que no se genere la suficiente confianza en los mismos. Otra desventaja radica también en el hecho de que los resultados resultan ser aleatorios, ya que incluso bajo las mismas condiciones, es decir, la misma entrada y los mismos parámetros de configuración, los resultados obtenidos siempre son distintos y esto hace también que las herramientas de explicación sean poco confiables.

En el presente trabajo se da solución a los problemas descritos anteriormente (subjetividad y aleatoriedad de los resultados), además de usar a el mismo modelo clasificador como herramienta de explicación. La solución propuesta para resolver el problema consiste en generar una explicación visual de la predicción basada en la caracterización de ciertas regiones de la imagen según su importancia para la predicción, por lo tanto, el principio del método propuesto consta de un algoritmo de búsqueda de regiones que puedan ser caracterizadas y utilizadas finalmente como explicación visual.

La contribución principal de este trabajo radica en disminuir la subjetividad de las explicaciones. Esta cuestión se ataca desde dos frentes: 1) No se utiliza ningún parámetro de configuración para el algoritmo, lo que asegura que no depende de la persona que lo implemente y, por tanto, siempre se llegará al mismo resultado, y 2) el resultado es claro y fácil de interpretar a la vez que intuitivo dadas las categorías presentadas, de tal forma que cualquier persona que conozca el código de colores será capaz de dar una explicación y esta será muy similar de manera inequívoca a la de cualquier otra persona.

2. Método

El método aquí propuesto consiste, de manera general, en realizar una búsqueda selectiva de regiones candidatas como regiones de interés, las cuales serán evaluadas usando el mismo clasificador para encontrar las regiones útiles, además de una función escalonada que indica el grado de relevancia de esa región para predecir la clase en cuestión.

2.1. Búsqueda de regiones útiles

Se busca generar una explicación visual de la predicción basada en la caracterización de regiones según su importancia para el proceso de predicción, por lo tanto, el principio del método consta de un algoritmo de búsqueda de regiones que puedan ser caracterizadas y usadas como explicación visual.

Si bien se podría implementar un tipo de búsqueda exhaustiva usando segmentación o algún algoritmo de ventana deslizante, esto además de contar con un espacio de búsqueda enorme, también generaría regiones indeseables debido a su tamaño o forma que no aportan en sí ninguna información útil a la explicación de la predicción.

En este trabajo se utiliza el algoritmo de búsqueda selectiva [5], pensado para abarcar regiones de distintas escalas lo que garantiza primero, que todos los objetos sean tomados en cuenta y segundo, que exista una distinción marcada en el tamaño de las regiones para así realizar una mejor caracterización de estas.

Para llevar a cabo la búsqueda de regiones, se usa el método de segmentación basada en grafos [6], donde se considera la entrada como un grafo: $G = (V, E)$, con n el número de vértices y m el número de aristas de G . El objetivo es segmentar V en r componentes, que corresponden a las primeras regiones propuestas, las cuáles se agrupan usando un algoritmo voraz basado en la similitud de las regiones con un enfoque de abajo hacia arriba (*bottom-up*). Esta similitud entre regiones debe ser propagada de manera jerárquica, para esto se usa la siguiente formulación:

$$S_{colour}(r_i, r_j) = \sum_{k=1}^n \min(c_i^k, c_j^k), \quad (1)$$

como similitud de color para cada par de regiones r_i, r_j usando el histograma de intersección donde c_i y c_j son los histogramas de las regiones respectivamente.

Si se obtiene el histograma de texturas para cada región entonces se puede calcular la medida de similitud de texturas como sigue:

$$S_{texture}(r_i, r_j) = \sum_{k=1}^n \min(t_i^k, t_j^k). \quad (2)$$

Para obligar a las regiones pequeñas a juntarse a regiones más grande, se utiliza la siguiente medida de similitud:

$$S_{size}(r_i, r_j) = 1 - \frac{size(r_i) + size(r_j)}{size(im)}, \quad (3)$$

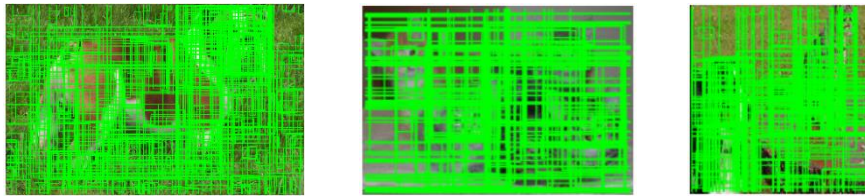


Fig. 1. Regiones encontradas con el algoritmo de búsqueda selectiva para tres imágenes distintas cuyas clases son a) *French_Bulldog*, b) *Egyptian_cat* y c) *Egyptian_cat* respectivamente.

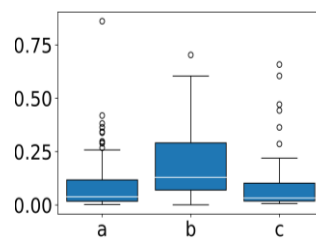


Fig. 2. Cuartiles y valores poco comunes para las imágenes a), b) y c) y de la Fig. 1.

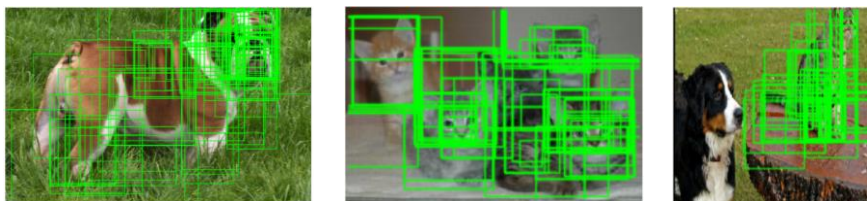


Fig. 3. Conjunto de regiones útiles R_u para las imágenes a, b y c respectivamente.



Fig. 4. Resultado de la explicación visual (abajo) para tres imágenes distintas (arriba).

donde $size(im)$ es el tamaño de la imagen en píxeles. Para tomar en cuenta diferentes escenas y condiciones de luz el proceso anterior se realiza para diferentes espacios de color, RGB y HSV.

Por último, del conjunto de regiones encontradas $R = \{r_1, \dots, r_n\}$, donde $r_i = \{x, y, w, h\}$, es decir, que cada región r_i representa un recuadro delimitador con el par (x, y) representando su posición y el par (w, h) su tamaño, como se muestra en la Fig. 1, es indispensable conocer cuáles de ellas son útiles, es decir, tienen mayor influencia en la predicción. Para esto, dado el clasificador $C(x)$ del cual se desea obtener la explicación visual de la predicción, se obtiene $P = \{p_1, \dots, p_n\}$ donde $p_i = C(r_i)$, como el conjunto de predicciones para el conjunto de regiones obtenido, entonces, el conjunto de regiones útiles se define como:

$$R_u \subset R \mid p_i \in R \wedge p_i > Q_1 + \frac{(Q_3 - Q_1)}{2}, \quad (4)$$

es decir, que el conjunto de regiones útiles se compone de aquellos valores mayores al rango semi-intercuartil, esto debido a que las distribuciones de probabilidad son sesgadas como se muestra en la Fig. 2.

Una vez hecho esto se tiene ya reducido el conjunto R a R_u con las regiones más relacionadas o útiles para el clasificador al momento de realizar la predicción, como se observa en la Fig. 3.

2.2. Caracterización de regiones y visualización

Sea el conjunto de regiones útiles $R_u = \{r_1, \dots, r_m\}$ encontrado, según lo descrito en la sección anterior, habrá que evaluar cada una de estas regiones y clasificarla en una de las tres categorías propuestas según la importancia de está en la toma de decisión del clasificador para la predicción. Estas categorías son *Significativa*, *Relevante* y *Fútil*, y posteriormente marcarlas visualmente en la imagen.

Dado el clasificador C del cual se desea obtener la explicación visual de la predicción, entonces la categoría de cada región en R_u estará dada por $F(C(r_i))$ donde F es una función escalón definida cómo:

$$F(x) = \begin{cases} \textit{Significativa}, & Z(x) \geq 2, \\ \textit{Relevante}, & Z(x) < 2 \wedge x \geq Q_1, \\ \textit{Fútil}, & x < Q_1, \end{cases}$$

donde $Z(x)$ es el puntaje z .

De acuerdo con la regla práctica del intervalo se puede conocer si un valor es infrecuente o poco común si está a más de dos desviaciones estándar de la media, es decir, un valor realmente *Significativo* para la predicción tiene puntuaciones z mayores que 2.

Ahora bien, para saber que valores son *Fútiles* se requiere otro concepto diferente ya que los valores de las predicciones restantes son de hecho relevantes y no se espera que ninguno tenga un valor $Z(x) < -2$, entonces en este caso aquellas regiones que no son significativas y estén por encima del cuartil Q_1 pertenecerán a la categoría de *Relevante* y las que están por debajo de ese valor, aunque si influyen en la predicción, se pueden considerar *Fútiles*.

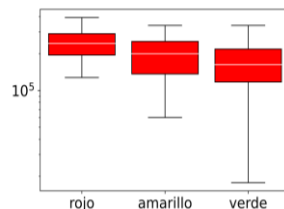


Fig. 5. Comparativo entre las regiones útiles dadas por el algoritmo de explicación propuesto.

Tabla 1. Tabla en donde se detalla el comportamiento del algoritmo de explicación propuesto para 15 ejemplos de imágenes distintas pertenecientes a las clases *perro* y *gato*.

Clase Predicha	RP	RU	AT (px)	ARU(px)	AR (px)	AA (px)	AV (px)	T
Basset	2217	165	306,560	238,920	238,920	60,268	66,196	140.25
Maltese_dog	805	141	242,181	242,181	242,181	199,733	114,114	120.65
Boxer	2281	172	346,977	312,768	312,768	221,001	120,460	238.69
French_bulldog	1010	153	307,200	200,736	200,736	117,912	114,260	170.88
Border_collie	3464	720	404,480	394,368	394,368	293,056	258,300	448.85
MiniSchnauzer	1226	194	307,200	153,120	127,600	112,608	153,120	147.87
English_setter	1145	174	350,080	339,840	338,560	339,840	339,840	216.91
JapaneseSpaniel	2556	1125	332,800	213,720	213,720	213,720	164,400	187.43
Persian_cat	1477	202	307,200	307,200	307,200	307,200	203,010	236.04
Tabby	1826	325	339,200	339,200	324,480	339,200	325,950	224.25
Tabby	2937	201	307,200	291,840	270,408	278,400	233,211	334.58
Tabby	1175	217	272,000	189,975	188,700	189,975	189,550	323.36
Egyptian_Cat	1080	155	307,200	307,200	268,800	206,816	235,520	266.19
Egyptian_Cat	800	55	307,200	143,397	137,572	99,029	92,208	111.13
Tiger_cat	1842	141	272,640	191,268	165,680	184,828	162,756	192.29

Finalmente, se colorean las regiones según su categoría con los colores verde, amarillo y rojo para *Significativa*, *Relevante* y *Fútil* respectivamente dentro de la imagen original como se muestra en la Fig. 4.

3. Discusión y resultados

Para el análisis de los resultados obtenidos durante el presente trabajo se utilizó un modelo pre-entrenado llamado *InceptionResNet*. Se trata de una red neuronal convolucional (CNN) entrenada con más de un millón de imágenes tomadas de la base de datos *ImageNet*. Al usar *InceptionResNet* se puede asumir que la predicción es correcta y así concentrar el análisis únicamente en la explicación. Como ejemplo del funcionamiento del algoritmo propuesto de explicación, en la Fig. 4, se muestran tres imágenes totalmente diferentes (arriba) y su explicación visual (abajo) para las clases predichas *French_Bulldog*, *Egyptian_cat* y *Egyptian_cat* respectivamente (clases dadas por el modelo *InceptionResnet*), con la cual se puede observar claramente qué regiones



Fig. 6. Perro de la clase *Bernese_mountain_dog*.

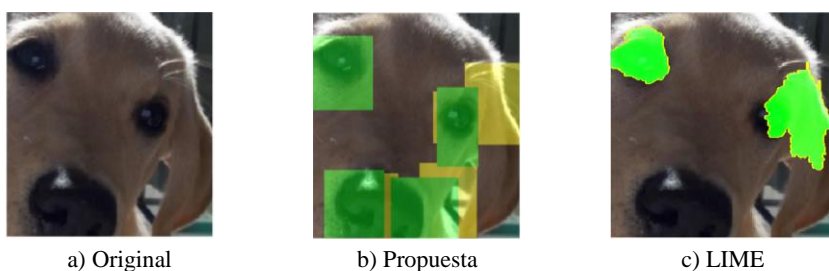


Fig. 7. Explicación del algoritmo propuesto para la clase *Labrador_retriever*, comparada contra la explicación dada por LIME.

son consideradas por el clasificador para determinar a qué clase pertenece cada imagen, y se explica además de forma clara que ponderación tiene cada región en la predicción.

En las tres imágenes están claramente resaltadas como regiones útiles aquellas que están más cercanas a lo que nos fijaríamos como seres humanos, ahora bien, en cuanto a la relevancia es muy interesante observar por ejemplo, en la primera imagen de la Fig. 4 (abajo), que el algoritmo propuesto explica que sí bien, todo el cuerpo del perro es relevante para indicar que pertenece a esa clase (pues está marcado con los colores rojo y amarillo), resulta ser que el hocico, los ojos y las orejas son de vital importancia y son el principal factor discriminante para indicar que en efecto es un perro pero del tipo *French_Bulldog*.

Las regiones útiles se traslapan constantemente antes de ser delimitadas y es interesante estudiar el tamaño real o la sumatoria de todas estas regiones ya que demuestran el proceso para llegar al resultado final mostrado por el algoritmo de explicación propuesto. Esta información fue analizada para 200 imágenes.

En la Fig. 5 se puede observar que las regiones *Relevantes* (amarillas) predominan sobre las otras dos regiones *Significativas* y *Fútiles* (verde y roja) como era de esperarse dado el criterio de caracterización de regiones descrito en la sección 2.2. Sin embargo, se puede observar que las regiones *Fútiles* (rojas) se distribuyen en un área mayor dentro de la imagen a explicar.

En la Tabla 1. se detallan los datos del análisis para 15 imágenes. Donde, RP es el número de regiones propuestas, RU el número de regiones útiles, AT el área total de la imagen, ARU el área total de las regiones útiles, AR el área de las regiones *Fútiles*, AA el área de las regiones *Relevantes*, AV el área de las regiones *Significativas*, y T el tiempo de ejecución del algoritmo de explicación propuesto dado en segundos.

3.1. Explicación multiclase

En la tercera imagen de la Fig. 4 (abajo) se usa el algoritmo de explicabilidad propuesto para resaltar a que le da importancia el modelo para predecir que la imagen pertenece a la clase *Egyptian_cat*, ya que, aunque se visualiza en la imagen también un perro, el gato tiene más relevancia pues es más grande y se muestra todo su cuerpo, como se explica con el algoritmo propuesto.

Sin embargo, aprovechando la complejidad del modelo *InceptionResnet*, y dado que este también reconoce perros y que de hecho existe una probabilidad de predicción para el perro de la imagen perteneciente a la clase *Bernese_mountain_dog*, entonces, se puede explicar de igual manera la predicción de esta clase, como se muestra en la Fig. 6. Con esto se demuestra que pueden explicarse las clases que detecte un modelo dentro de una imagen dada, como en este caso para perro o gato.

3.2. Relevancia de regiones

Es de vital importancia entender que este algoritmo busca dentro de la imagen las regiones de mayor importancia para la predicción realizada sin confundirse con la búsqueda de algún objeto. En el presente trabajo se utiliza el modelo *InceptionResnet* útil para la búsqueda de objetos, sin embargo, el funcionamiento del algoritmo aquí propuesto se adapta a cualquier problema de clasificación de imágenes haciendo visible la explicación de la predicción en cualquier contexto.

En la Fig. 7 se muestra un ejemplo claro de esto, donde aun usando *InceptionResnet*, las regiones marcadas por el algoritmo propuesto Fig. 7 b) muestran claramente las regiones útiles y más aún explica que el modelo se ha fijado en los ojos y nariz del perro para realizar la predicción, que es en realidad lo que se esperaría.

Además, se hace uso de esta misma imagen para hacer una comparación con los resultados obtenidos por LIME Fig. 7 c) que, a pesar de obtener también una buena explicación, pasa por alto la nariz y el resultado tiene claramente una interpretación que depende del punto de vista del observador lo que podría crear desconfianza en la explicación.

3.3. Confiabilidad

Para verificar la eficiencia del algoritmo se compara la región útil obtenida por el algoritmo de explicación propuesto, contra lo que una persona se fijaría para decir que una imagen dada pertenece a una clase predicha, sin embargo, ¿Qué nivel de confianza ofrece el algoritmo para asegurar que funcionará en todos los ejemplos?, pues bien usando imágenes del conjunto de datos COCO que contiene más de 200 mil imágenes con objetos etiquetados y marcados por seres humanos, y que en el presente trabajo se usará como un conjunto de imágenes con cierta pertenencia a una clase (en este caso un objeto dentro de la imagen que un ser humano marcó), es decir, que indicó la región en la que se fijaría para determinar que esa imagen pertenece a la clase en cuestión, se trata de resolver este cuestionamiento.

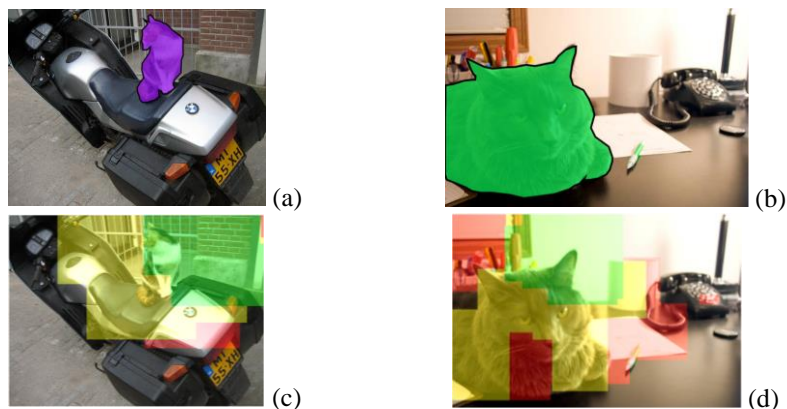


Fig. 8. Ejemplo de comparación entre explicaciones marcadas por un ser humano (arriba) y el algoritmo de explicación visual propuesto (abajo).

En la Fig. 8 a) y b) se muestran dos ejemplos de imágenes del conjunto de datos COCO marcadas y clasificadas por una persona real y se comparan con la explicación del algoritmo aquí propuesto Fig. 8 c) y d). Se puede observar que la región útil marcada por el algoritmo de explicación propuesto rodea efectivamente toda la región marcada por el ser humano, lo que demuestra que funciona adecuadamente, esto siempre y cuando el modelo se encuentre bien entrenado, de lo contrario la utilidad del algoritmo de explicación cambiaría.

Se seleccionaron 200 imágenes diferentes del conjunto de datos COCO con perros y gatos como clase principal, y se realizó una comparación con el algoritmo de explicación propuesto. Para esto se utilizó la técnica de *Intersection Over Union* dada por $IoU = At/A$, donde At es el área de traslape del área marcada en COCO con el área de las regiones útiles del algoritmo de explicación propuesto, y A es el área marcada solamente en COCO.

La explicación visual siempre debería abarcar lo que un humano marcaría si es que el modelo está bien entrenado, como es el caso del modelo *InceptionResnet*. El resultado fue favorable en todos los casos ya que el área marcada en COCO siempre está dentro de la región útil encontrada por el algoritmo de explicación propuesto, demostrando así que es eficaz en encontrar la explicación visual de la región útil que el modelo usa para la predicción, además de mostrarlo en categorías *Significativa*, *Relevante* y *Fútil*.

4. Conclusión

El algoritmo de explicabilidad aquí propuesto mostro ser eficiente, además tiene una ventaja primordial, la cuál es el objetivo del presente trabajo, que consiste en disminuir la subjetividad en la explicación, ya que en los algoritmos más populares (LIME, mapas de calor, etc.), según la persona que realice la implementación e interprete el resultado es la conclusión a la que se llegará.

Esta cuestión de subjetividad aquí se ataca por dos frentes, primero no existe ningún parámetro de configuración para el algoritmo lo que asegura que no depende de la persona que lo implemente y siempre se llegara al mismo resultado; segundo, el resultado es claro y fácil de interpretar a la vez que intuitivo dadas las categorías presentadas, de tal forma que cualquier persona que conozca el código de colores será capaz de dar una explicación y esta será muy similar de manera inequívoca a la de cualquier otra persona.

De manera general, se obtuvo un algoritmo funcional que habrá que seguir trabajando, así como aplicarlo a problemas concretos en donde se demuestre su utilidad y su potencial frente a otros algoritmos. También habrá que trabajar más adelante en la reducción del tiempo de ejecución, ya que en promedio para una imagen de 300px cuadrados tarda hasta 2 minutos en obtener su explicación, sin embargo, esto varía y depende de la cantidad de regiones encontradas o de la complejidad de la imagen de entrada, así como de la complejidad del modelo y del área de interés a explicar.

Agradecimientos. Los autores desean agradecer al Instituto Politécnico Nacional por el apoyo económico brindado a través del proyecto SIP 20200638. Tonantzin M. Guerrero agradece al Consejo Nacional de Ciencia y Tecnología por la beca brindada para realizar sus estudios de doctorado.

Referencias

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Chatila, R.: Explainable artificial intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion*, 58, pp. 82–115 (2020)
2. Lipton, Z.C.: The myths of model interpretability, *Queue*, 30(31), pp. 30–57 (2018)
3. Patrick, H.: On Explainable Machine Learning Misconceptions & A More Human-Centered Machine Learning (2019)
4. Tulo-Ribeiro, M.: Why should I trust you? Explaining the predictions of any classifier. *arXiv:1602.04938v3 [cs.LG]* 9 (2016)
5. Uijlings, J.R., Van De-Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. *International Journal of Computer Vision*, 104(2), pp. 154–171 (2013)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), pp.167–181 (2004)
7. Ghorbani1, A., Zou, J.Y.: What is your data worth? Equitable Valuation of Data (2019)
8. Robnik-Sikonja, M.: Explanation of prediction models with explain prediction. *Informatica*. 42, pp. 13–22 (2018)
9. Szegedy, C., Yangqing-Jia, W.L., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going Deeper with Convolutions (2014)
10. Botchkarev, A.: Performance metrics (Error Measures) in machine learning regression. *Forecasting and Prognostics: Properties and Typology* (2018)
11. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized Input Sampling for Explanation of Black-box Models (2018)
12. Papernot, N., McDaniel, P.: Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning (2018)

13. Hall, P.: On Explainable Machine Learning Misconceptions & A More Human-Centered Machine Learning (2019)
14. Doran, D., Schulz, S., Besold, T.R.: What Does Explainable AI Really Mean? A New Conceptualization of Perspectives (2017)
15. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building Machines That Learn and Think Like People (2016)
16. Tulio-Ribeiro, M., Singh, S., Guestrin, C.: Model-Agnostic Interpretability of Machine Learning (2016)
17. Wang, C., Xi, Y.: Convolutional neural network for image classification. Johns Hopkins University Baltimore, MD (2018)
18. Robnik-Sikonja, M.: Explanation of prediction models with explain prediction (2018)
19. Krause, J.: Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models (2016)
20. van Lent, M.: An Explainable Artificial Intelligence System for Small-unit Tactical Behavior (2004)
21. Gunning, D.: Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA) (2017)

Reconstrucción autónoma de un mapa métrico de una pista de competencias usando visión artificial

Esther Diaz Sarmientos¹, Humberto Sossa Azuela²,
Alberto Petrilli Barceló³

¹ Universidad Tecnológica de la Mixteca,
División de Estudios de Posgrado,
México

² Instituto Politécnico Nacional,
Centro de Investigación en Computación,
México

³ Tokyo University of Science,
Faculty of Science and Technology,
Japan

dise900301@hotmail.com, hsossa@cic.ipn.mx,
petrilli@rs.tus.ac.jp

Resumen. El presente artículo describe el desarrollo de un algoritmo para la reconstrucción de un mapa métrico a partir de las imágenes capturadas por la cámara Intel Real Sense a bordo del robot AutoNOMOS mini. Las etapas que componen el algoritmo constan de la corrección de perspectiva de las imágenes, detección de líneas de carril, asociación de puntos clave y la reconstrucción del mapa. Se presentan los resultados obtenidos en una pista real, realizando el procesamiento a bordo y completamente en la plataforma, y de un ambiente simulado en Gazebo en un ambiente virtual.

Palabras clave: Mapeo, Visión Artificial, Gazebo, ROS, autoNOMOS, Proyectividad.

Autonomous Reconstruction of a Metric Map of a Competition Track Using Computer Vision

Abstract. This article describes the development of an algorithm for the reconstruction of a metric map from the images captured by the Intel Real Sense camera on board the AutoNOMOS mini robot. The stages that make up the algorithm consist of the perspective correction of the images, detection of lane lines, association of key points and the reconstruction of the map. The results obtained are presented on a real track, carrying out the processing on board and

completely on the platform, and from a simulated environment in Gazebo in a virtual environment.

Keywords: Mapping, Artificial Vision, Gazebo, ROS, autoNOMOS, Projectivity.

1. Introducción

En robótica los mapas constituyen una herramienta importante en la representación del entorno y permiten planeación de rutas, localización y navegación. Generalmente se dividen en topológicos y métricos [1]. En esta clasificación, los de tipo métrico se distinguen por ser menos complejos en cuanto a la representación de la información espacial con formas básicas como líneas y curvas.

Las representaciones del entorno dependen completamente de la información proporcionada por los sensores, presentes tanto en el robot como en el ambiente. Estos sensores pueden ser de distintos tipos como láseres, cámaras, GPS, posición y orientación, entre otros.

Los mapas a nivel de carreteras o pistas de competencias contienen información relevante sobre el camino, señales u obstáculos presentes en el entorno. Estas representaciones permiten la planeación de mejores rutas y ahorro de recursos, entre otros beneficios. En las pistas de competencias, contar con información gráfica de carriles permite realizar el reto de recorrer cualquier circuito planeando la trayectoria con anticipación.

En este trabajo se describe la metodología utilizada por medio de visión y asociación de datos para la reconstrucción gráfica de una pista de competencias a nivel de líneas de carril, procesando en tiempo real y a bordo de un vehículo a escala.

2. Trabajos relacionados

La construcción de mapas está basada en la información adquirida por medio de sensores y según las características de relevancia es el tipo de sensor a utilizar. En los coches autónomos, las cámaras permiten la adquisición de información de tipo visual, la cual es base para la detección de líneas de carril. Para esta aplicación por medio de imágenes se usan métodos como redes neuronales y filtros estadísticos [2], [3], [4], [5] y [6].

En el estado del arte de mapas a nivel de carreteras se utilizan distintos sensores a bordo de los vehículos como GPS [7], cámaras [8], sensores láser LiDAR [9], entre otros. El trabajo de [1] realiza una revisión en trabajos en temas de coches autónomos. Dentro de estos destacan las aplicaciones realizadas en mapas tanto métricos como topológicos.

Los vehículos a escala constituyen una oportunidad importante en el desarrollo de aplicaciones para coches autónomos. En este ámbito destacan las competencias internacionales como la copa Carolo [10] de la Technische Universität Braunschweig en Alemania, como nacionales con el Torneo Mexicano de Robótica [11] o Talent Land [12] por la oportunidad que ofrecen de desarrollar algoritmos para resolver los retos.



Fig. 1. AutoNOMOS mini V2.

En el trabajo de [13] hacen un recuento de los algoritmos utilizados por cada equipo en la competencia del 2016 para resolver los distintos desafíos de los que consta la copa Carolo. [14] desarrolló una metodología para la reconstrucción de una pista de competencias utilizando un mapa métrico basado en ocupación de rejillas. Su sistema realiza la construcción de un mapa local por medio de la estimación de movimiento y la extracción de características de imágenes. Los mapas locales son agregados a un mapa global por medio de un algoritmo de asociación de datos.

En el ámbito nacional, [15] desarrolló un sistema de navegación para la plataforma AutoNOMOS mini V1, donde realiza detección de carriles, generación y seguimiento de rutas, detección y evasión de obstáculos y estacionamiento en paralelo. [16] diseñó una metodología basada en redes neuronales convolucionales para el control de dirección de la plataforma AutoNOMOS mini V1 en ambientes reales y virtuales.

3. Desarrollo

La implementación de este trabajo se realizó a bordo de la plataforma “AutoNOMOS Mini”, mostrada en la figura 1. Esta plataforma [17] es un modelo de vehículo (escala 1:10) desarrollado para fines educativos por la Freie Universität Berlín en el entorno de trabajo ROS (Robot Operating System) y cuenta con distintos sensores, entre ellos una cámara Intel Real Sense y un IMU.

ROS es un entorno de trabajo que provee librerías y herramientas para ayudar a desarrollar aplicaciones para robots [18]. Dentro de sus prestaciones importantes está la simulación de robots y entornos de forma gráfica utilizando Gazebo. Gran cantidad de estos modelos se ponen a disposición del público en repositorios en línea. La implementación de este trabajo en un ambiente virtual se realizó con el simulador diseñado por [19]. La metodología consta de cuatro etapas principales, desde la corrección de perspectiva hasta la reconstrucción del mapa. Cada una de estas se describe en las siguientes secciones.

3.1. Calibración de cámaras y corrección de perspectiva

Debido al efecto de la perspectiva en las imágenes obtenidas con la plataforma es necesario realizar una corrección de perspectiva o transformación proyectiva. Una transformación proyectiva o proyectividad [20] es una transformación invertible dada por $H = \mathbb{P}^2 \rightarrow \mathbb{P}^2$ de manera tal que una línea recta es transformada a una línea recta. La corrección de perspectiva realizada está basada en transformaciones geométricas y

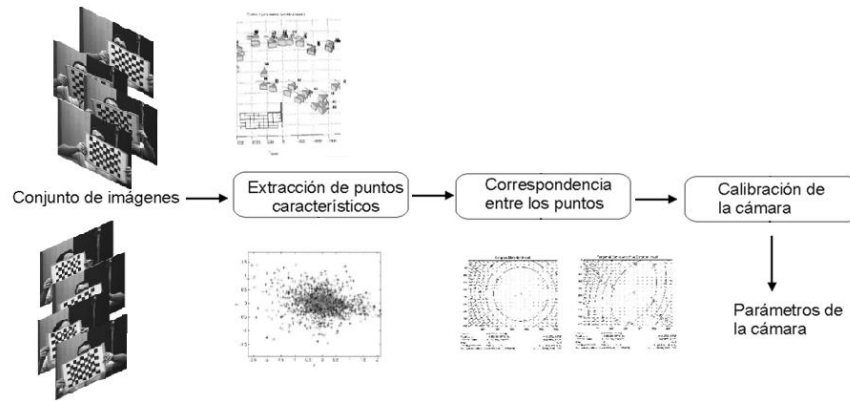


Fig. 2. Proceso de calibración de cámaras [21].

depende de parámetros propios de la cámara obtenidos de la calibración y ubicación de la misma en la plataforma AutoNOMOS mini.

La calibración de una cámara [22] es el proceso que permite obtener los parámetros que definen las condiciones de formación de una imagen, incluyendo la geometría interna y la óptica de la cámara (parámetros intrínsecos), así como su posición y orientación respecto a un patrón de calibración (parámetros extrínsecos). En general el proceso de calibración de cámaras consta de las etapas mostradas en la figura 2.

A fin de encontrar los parámetros intrínsecos se realizó la calibración de la cámara Intel Real Sense con un patrón tipo tablero de ajedrez, con un nodo propio de ROS y OpenCV basado en el método de Zhang. En el caso de los parámetros extrínsecos, se definieron dos, la orientación del campo de visión y la distancia del centro óptico respecto al suelo.

Los parámetros usados en la corrección de perspectiva corresponden a una rotación de 286° y una distancia de 298 mm en la plataforma AutoNOMOS mini. En el ambiente virtual se propuso una cámara ideal en cuanto a los parámetros extrínsecos y los valores intrínsecos se ajustaron a una rotación en X de 273° , una traslación en Z de 325 y en Y de 20. Las matrices de transformación obtenidas tanto en el robot real, como en el ambiente virtual están denotadas por las expresiones 1 y 2 respectivamente:

$$H_{real} = \begin{bmatrix} 594.651 & -294.279 & -18985.060 \\ 0 & -71.718 & 90245.466 \\ 0 & -0.961 & 532.637 \end{bmatrix}, \quad (1)$$

$$H_{sim} = \begin{bmatrix} 640 & -319.561 & -24105.252 \\ 0 & -206.176 & 140282.259 \\ 0 & -0.999 & 564.671 \end{bmatrix}. \quad (2)$$

En la figura 3 se puede ver el resultado de la aplicación de la corrección de perspectiva en la plataforma real con una imagen de la pista. El ancho de cada carril es de 40cm y la línea discontinua central tiene una medida de 20cm en cada segmento al igual que los espacios, lo que puede observarse con ayuda de la cuadrícula verde.

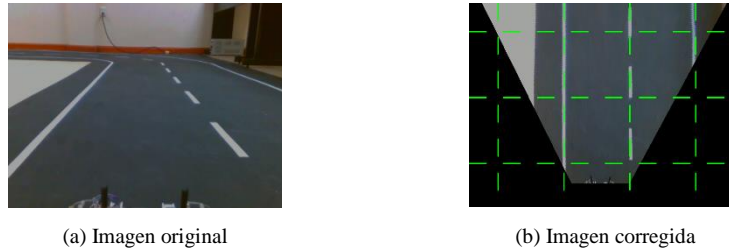


Fig. 3. Aplicación de corrección de perspectiva.

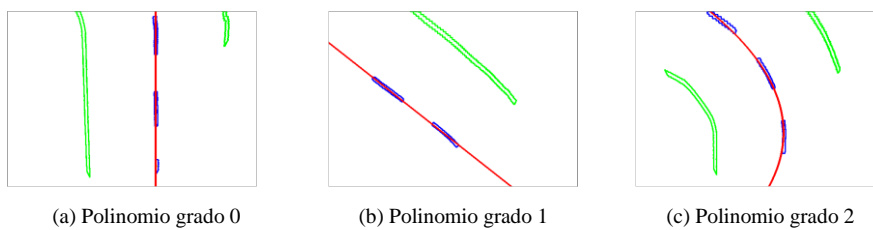


Fig. 4. Aproximación polinomial.

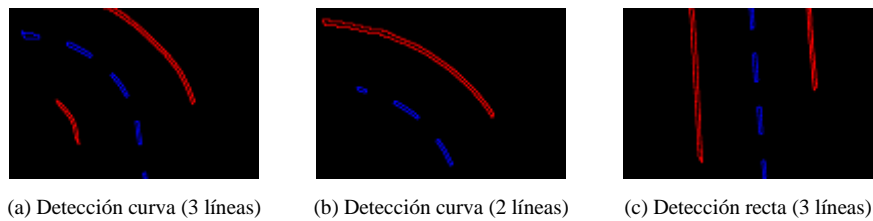


Fig. 5. Detección de líneas de carril en 3 posiciones diferentes.

3.2. Detección de líneas de carriles

La detección de líneas de carriles está enfocada en identificar la línea central y con base en esta clasificar las líneas laterales. La imagen corregida en perspectiva pasa por una etapa de eliminación de píxeles que no son de interés por medio de una operación de binarización con un umbral de 75. Posteriormente se aplica el detector de contornos propio de OpenCV [23] a fin de identificar posibles elementos de las líneas de carril. De estos contornos se eliminan los que tengan áreas muy grandes que corresponden a la porción del suelo visible.

Dentro de esta etapa se realiza una aproximación polinomial, con los centroides de los contornos de la línea central. El polinomio obtenido de esta operación es usado para evaluar los centroides de los contornos nuevos en la imagen siguiente a fin de clasificarlos o no como parte de la línea central. Estos polinomios pueden ser de grado 0, 1 y 2, dependiendo de la ubicación y número de contornos encontrados, como se muestra en la figura 4.

Por ultimo las líneas laterales se identifican según la ubicación respecto a la línea central. La figura 5 muestra los resultados obtenidos en la detección de las líneas de carril, identificando la línea central en color azul y las líneas laterales en color rojo.

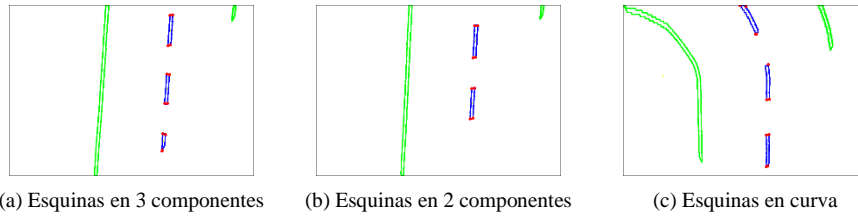


Fig. 6. Detección de esquinas.

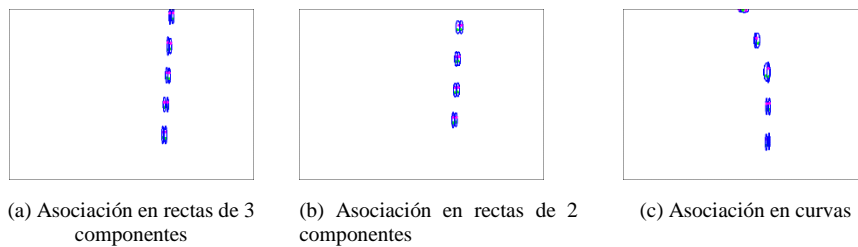


Fig. 7. Asociación de esquinas.

3.3. Asociación de puntos clave

En esta etapa de la reconstrucción del mapa se busca utilizar las características invariantes en las imágenes consecutivas y que permiten encontrar una relación entre las mismas. En este caso se optó por las esquinas de las componentes de la línea central las cuales, dado el movimiento, no tienen un cambio brusco entre cada imagen sucesiva.

La primera parte de esta etapa corresponde a encontrar las esquinas de las componentes de la línea central. El primer paso es encontrar los coordenadas (x,y) de la imagen que encierran todos los puntos de un contorno en el rectángulo más pequeño posible. Para esto se realiza la búsqueda de las coordenadas (x, y) mayores y menores de cada componente.

Posteriormente se buscan los puntos más cercanos a la combinación de las coordenadas x y y mayores y menores para definir las esquinas de la siguiente forma:

- Esquina 0: es el punto más cercano a x_{menor} y y_{menor} .
- Esquina 1: es el punto más cercano a x_{mayor} y y_{menor} .
- Esquina 2: es el punto más cercano a x_{menor} y y_{mayor} .
- Esquina 3: es el punto más cercano a x_{mayor} y y_{mayor} .

La figura 6 muestra el resultado de esta etapa en tres situaciones distintas, con presencia de componentes centrales completos e incompletos.

Una vez detectadas las esquinas de cada contorno central, lo siguiente es relacionarlas con las esquinas de la imagen anterior. Para este paso, se tiene como consideración el hecho de que entre dos imágenes consecutivas el desplazamiento y rotación no aumentan, o disminuyen según el caso, de manera considerable.

La primera comprobación que se realiza es que el componente central este completo. Dado que estos pueden ser rectos o curvos, se optó por usar la distancia entre la Esquina 0 y la Esquina 3. Si esta distancia es aproximadamente igual a la distancia de un componente completo se utilizan las cuatro esquinas. Si este no es el caso solo se guardan las dos superiores o las dos inferiores, según la ubicación de la componente. Se verifica la distancia entre cada una de las componentes del frame anterior para relacionarlo con el nuevo. En la figura 7 se pueden observar las esquinas relacionadas entre sí. Los óvalos encierran a las esquinas presentes en dos imágenes sucesivas.

3.4. Reconstrucción del mapa

Encontradas las esquinas presentes en dos imágenes consecutivas, así como su relación, el paso siguiente es encontrar la matriz de transformación entre ellas. Para esta etapa se utilizó una variación del método descrito en [24], donde realizan el ajuste de dos colecciones de puntos tridimensionales por medio de una transformación de similitud. La variación del método consiste en su implementación con dos colecciones de puntos 2D, como se muestra en el Algoritmo 1.

Después de varias pruebas se decidió utilizar la medición de la orientación proporcionada por el sensor IMU. De esta forma cada vez que se captura una imagen nueva se estima la nueva rotación a partir de la resta de la orientación nueva menos la anterior. Esta información se publica en forma de mensajes en tiempo real en el tópico de ROS `/model_car/yaw` propio de la plataforma.

Obtener el incremento en la orientación por cada frame capturado requiere de ciertas consideraciones. Esto es debido a la forma en que el sensor IMU otorga la medición del ángulo. Los ángulos medidos están dados en grados y van de 0° a -180° y de 180° a 0° . Debido a esto se realiza una conversión entre el valor otorgado por el sensor y la medición real, además de que la posición del IMU al iniciar el recorrido no es cero.

Algoritmo 1. Calculo de matriz de transformación entre dos conjuntos de puntos 2D.

```

Entrada: Matriz de puntos  $A^{2 \times n}$  y  $B^{2 \times m}$ 
Salida : H: Matriz de transformación
1  $P_{m1}$  = punto medio de A.
2  $P_{m2}$  = punto medio de B.
3  $A_m = A - P_{m1}$ 
4  $B_m = B - P_{m2}$ 
5  $D = A_m \cdot B_m^T$ 
6  $(U, S, v_t) = SVD(D)$ 
7  $R = v_t^T \cdot U^T$ 
8  $t = -R \cdot P_{m1} + P_{m2}$ 
9  $H = \begin{bmatrix} R & t \\ 0^{1 \times 2} & 1 \end{bmatrix}$ 

```

4. Resultados

La reconstrucción de mapas se llevó a cabo tanto en el ambiente virtual en Gazebo con tres pistas diferentes y en una sola pista real. En esta última se usaron las imágenes

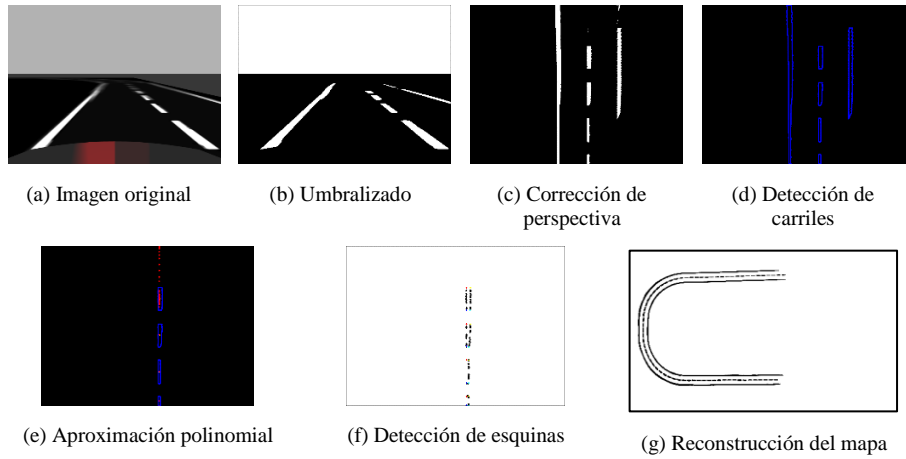


Fig. 8. Proceso de construcción del mapa, frame 2000.

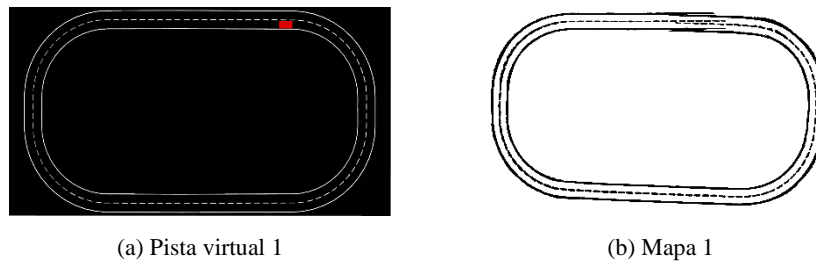


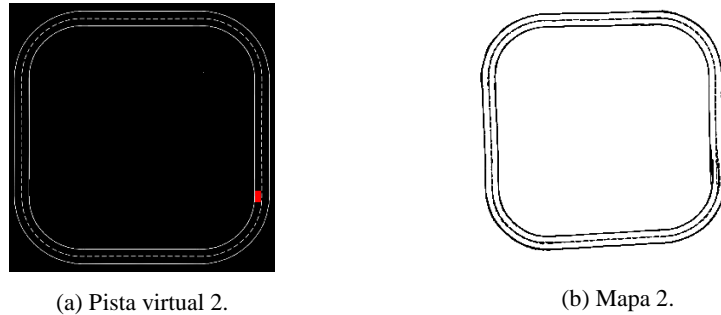
Fig. 9. Reconstrucción del mapa del ambiente virtual 1.

capturadas por la cámara Intel Real Sense RGB de tamaño 640×480 a 60 fps, a bordo de la plataforma AutoNOMOS mini. El vehículo fue teleoperado con un nodo en ROS con una velocidad máxima de 300 rpm, equivalente a 0.177 m/s. Las mediciones de la orientación fueron obtenidas del sensor IMU 6050 a bordo del robot.

En la figura 8 se observan las etapas del procesamiento, así como la porción reconstruida del mapa al recorrer aproximadamente la mitad de la pista virtual con 2000 imágenes procesadas en tiempo real.

En los escenarios virtuales la reconstrucción del mapa tiene menos errores de segmentación con respecto a la pista real. El error final del ángulo de salida en orientación en la pista real fue menor a 7° lo que es notorio en el emparejamiento del inicio y fin del recorrido.

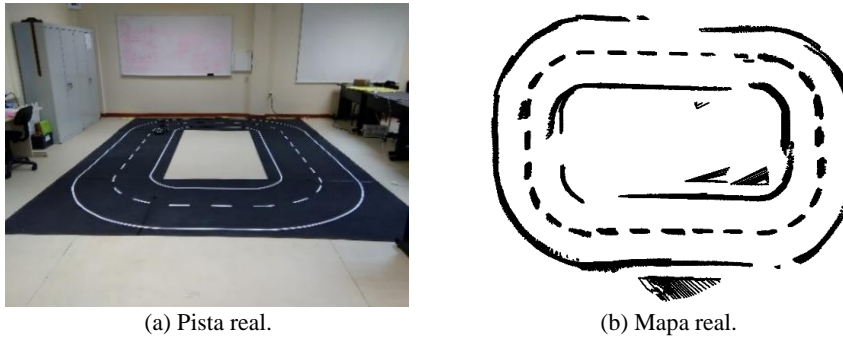
La reconstrucción completa de dos pistas virtuales se muestra en la figura 9 y 10, donde el recuadro rojo marca el inicio y fin de cada recorrido. De igual forma en la figura 11 se observa la reconstrucción de la pista real. En esta última se observan discontinuidades en las líneas laterales que corresponden a pérdida de información por el campo de visión más pequeño de la cámara de la plataforma real.



(a) Pista virtual 2.

(b) Mapa 2.

Fig. 10. Reconstrucción del mapa del ambiente virtual 2.



(a) Pista real.

(b) Mapa real.

Fig. 11. Reconstrucción del mapa de la pista real.

Al detectar al menos dos componentes de la línea central la asociación de puntos es invariante a cambios de carril. Los cambios bruscos de dirección al conducir la plataforma influyen en los errores en la medición de la orientación, lo cual afecta a las transformaciones entre imágenes.

En el procesamiento de imágenes de la pista real se tienen errores de segmentación debido al fondo en el que está montada la pista, cuyo color es muy cercano al blanco. Estos errores agregan píxeles erróneos a la reconstrucción, sin embargo visualmente son pocos en comparación a los correctos. Como se puede observar en la 11(b) en algunas partes de las líneas derecha e izquierda hay discontinuidades. Estas se deben a errores de clasificación en líneas rectas y a pérdida de información por el campo de visión de la cámara.

5. Conclusiones

Utilizando imágenes capturadas en conjunción con la medición de orientación a bordo de la plataforma AutoNOMOS mini se obtiene una representación gráfica de una pista de competencias. Esto es posible usando técnicas de procesamiento de imágenes y asociación de datos. La corrección de perspectiva tiene un correcto funcionamiento en los dos ambientes, real y virtual. Se implementó un detector de esquinas basado en la forma de las componentes centrales, el cual permite relacionar frames consecutivos.

La asociación de datos se basa en la medida de distancia recorrida calculada en píxeles a partir de las imágenes. Los mapas obtenidos son considerablemente mejores en el ambiente virtual, donde no hay cambios de iluminación y las tonalidades del suelo no difieren de la pista.

Agradecimientos. E. Diaz-Sarmientos agradece a CONACYT por la beca otorgada para la realización de sus estudios de maestría. Agradece también al Centro de Investigación en Computación del IPN por la oportunidad para realizar una estancia de investigación en el Laboratorio de Robótica y Mecatrónica. H. Sossa agradece al Instituto Politécnico Nacional por el apoyo financiero para la realización de esta investigación en el marco del proyecto 20200630.

Referencias

1. Badue, C., Guidolini, R., Carneiro, R.V., Azevedo, P., Cardoso, V.B., Forechi, A., Jesus, L., Berriel, R., Paixão, T.M., Mutz, F., et al.: Self-driving cars: A survey. *Expert Systems with Applications* (2020)
2. Aly, M.: Real time detection of lane markers in urban streets. In: *Intelligent Vehicles Symposium*, pp. 7–12 (2008)
3. Lopez, A., Serrat, J., Canero, C., Lumbreras, F., Graf, T.: Robust lane markings detection and road geometry computation. *International Journal of Automotive Technology*, 11, pp. 395–407 (2010)
4. Guo, C., Mita, S., McAllester, D.: Lane detection and tracking in challenging environments based on a weighted graph and integrated cues. In: *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 5543–5550 (2010)
5. Bae, J.H., Song, J.B.: Monocular vision-based lane detection using segmented regions from edge information. In: *Ubiquitous Robots and Ambient Intelligence (URAI), 8th International Conference on*, pp. 499–502 (2011)
6. Küçüküydiz, G., Ocak, H.: Development and optimization of a DSP-based real-time lane detection algorithm on a mobile platform. *Turkish Journal of Electrical Engineering & Computer Sciences*, 22, pp. 1484–1500 (2014)
7. Jo, K., Sunwoo, M.: Generation of a precise roadway map for autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*, 15, pp. 925–937 (2014)
8. Guo, C., Kidono, K., Meguro, J., Kojima, Y., Ogawa, M., Naito, T.: A low-cost solution for automatic lane-level map generation using conventional in-car sensors. *IEEE Transactions on Intelligent Transportation Systems*, 17, pp. 2355–2366 (2016)
9. Gwon, G.P., Hur, W.S., Kim, S.W., Seo, S.W.: Generation of a precise and efficient lane-level road map for intelligent vehicle systems. *IEEE Transactions on Vehicular Technology* 66, pp. 4517–4533 (2017)
10. Carolo cup: <https://wiki.ifr.ing.tu-bs.de/carolocup/en/carolo-cup> (2020)
11. Talent Land Competencias: <https://robotics.talent-network.org/edicion-2019/> (2019)
12. AutomodelCar Torneo Mexicano de Robótica 2019: <https://www.femexrobotica.org/tmr2019/portfolio-item/autos-autonomos/> (2019)
13. Karouach, I., Ivanov, S.: Lane detection and following approach in self-driving miniature vehicles (2016)
14. Rehder, E., Albrecht, A.: Submap-based slam for road markings. In: *2015 IEEE Intelligent Vehicles Symposium (IV), IEEE*, 1393–1398 (2015)

15. Bravo Conejo, C.G.: Navegación autónoma de un robot tipo automóvil en pista de carreras con obstáculos. Master's thesis, CIC IPN (2018)
16. Vanegas Sánchez, T.D.: Navegación autónoma de un vehículo a escala por medio de redes neuronales profundas. Master's thesis, CIC IPN (2018)
17. AutoNOMOS: [https://github.com/automodelcar/automodelcarwiki/wiki/hardware-\(autonomos-model-v2\)](https://github.com/automodelcar/automodelcarwiki/wiki/hardware-(autonomos-model-v2)) (2017)
18. ROS: <http://www.ros.org/about-ros/> (2018)
19. ITAM: Ek autonomos sim. <https://github.com/eagleknights/autonomos> (2019)
20. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
21. de la Escalera, A., Armingol, J.M., Pech, J.L., Gómez, J.J.: Detección automática de un patrón para la calibración de cámaras. Revista Iberoamericana de Automática e Informática Industrial RIAI 7, 83–94 (2010)
22. Isern Gonzalez, J.: Estudio experimental de métodos de calibración y autocalibración de cámaras (2003)
23. Suzuki, S. et al.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32–46 (1985)
24. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Transactions on pattern analysis and machine intelligence, pp. 698–700 (1987)

Diseño y desarrollo de un sistema de monitoreo remoto implementando Internet de las cosas

Y. Aguilar Molina¹, S.A. Sosa Ramírez¹, J.E. Galindo Morales¹,
A.N. Morales Reyes², R. Gallardo Sánchez², I.V. Villaseñor García³

¹ Universidad de Guadalajara,
México

² Instituto Tecnológico de Toluca,
México

³ Universidad Michoacana de San Nicolás de Hidalgo,
México

yehoshua@valles.udg.mx,
{sergio.sosa0939, josue.galindo4506}@alumnos.udg.mx,
{amoralesr1, rgallardos}@toluca.tecnm.mx

Resumen: La constante transformación de la industria, nuevas tendencias tecnológicas y el desarrollo de nuevos procesos de manufactura suponen nuevas necesidades que mejoran la calidad de los procesos y transforman por completo, la interacción humana con las máquinas. El *Internet de las cosas* se presenta como solución a estas áreas de oportunidad, dando como resultado un concepto llamado *Industria 4.0*. Actualmente existe un interés creciente por medir y monitorear numerosas variables del entorno. Con el fin de proveer un mayor flujo de información haciendo uso del Internet de las cosas (IoT, por sus siglas en inglés), el cual está cambiando la forma en que las personas se organizan, interactúan y participan en diversas facetas de la sociedad. *Internet of things* se considera una integración perfecta de usuarios y dispositivos para converger el ámbito físico con entornos virtuales derivados de humanos. El presente artículo presenta un sistema de monitoreo remoto implementado, con base en una Raspberry Pi 3B+, un servidor web y sensores de temperatura, humedad y presión atmosférica. Para recopilar y presentar datos medidos, se desarrolló un sitio web, donde el usuario accede para observar los datos recopilados.

Palabras clave: Monitoreo remoto, IoT.

Design and Development of a Remote Monitoring System Implementing the Internet of Things

The constant transformation of the industry, new technological trends and the development of new manufacturing processes pose new needs that improve the quality of processes and completely transform human interaction with machines. The Internet of Things is presented as a solution to these areas of opportunity, resulting in a concept called Industry 4.0. Currently there is a growing interest in

measuring and monitoring numerous variables in the environment. In order to provide a greater flow of information by making use of the Internet of Things (IoT), which is changing the way in which people organize, interact and participate in various facets of society. Internet of things is considered a seamless integration of users and devices to converge the physical realm with virtual environments derived from humans. This article presents a remote monitoring system implemented, based on a Raspberry Pi 3B +, a web server and temperature, humidity and atmospheric pressure sensors. To collect and present measured data, a website was developed, where the user accesses to observe the collected data.

Keywords: Remote Monitoring, IoT.

1. Introducción

El Internet de las cosas (IoT), denominado también “Internet de los objetos”, representan un parteaguas entre el hoy y ayer. Tomando ventaja de las capacidades que nos otorga el Internet (como herramienta de comunicación por excelencia), el Internet de las cosas es el máximo exponente del mismo; un concepto con un extenso margen de aplicación y un incuestionable impacto en los procesos auténticos modernos. Mismo que se encuentra presente en sectores como la Industria 4.0, domótica, medicina, agricultura, ganadería, hotelería, entretenimiento, solo por mencionar algunos. “La Industria 4.0 hace referencia a una nueva fase en la revolución industrial caracterizada por la automatización, la digitalización de los procesos y el uso de las tecnologías de la electrónica y de la información en la manufactura. El concepto fue acuñado en Alemania en 2011, dentro del contexto de una política económica gubernamental basada en estrategias de alta-tecnología (C. Cortés, J. Landeta and J. Chacón, 2020). La medición de los cambios meteorológicos ha sido siempre indispensable, la obtención y conocimiento de valores como temperatura, humedad, viento, presión atmosférica, entre otros; en un lugar específico representa a ventaja para el desarrollo de las actividades comerciales y de producción. La elaboración de este sistema de monitorio ambiental permitirá la medición de datos confiables para realizar una validación, manipulación y análisis de la información recolectada y está servir como base para futuras predicciones meteorológicas.

En la literatura puede encontrarse la propuesta de Palacios, P., Córdova, A., 2018, quien propone un sistema de IoT basado en Raspberry Pi, un sensor de temperatura, una base de datos (cloud DataBase) enfocada a las casas inteligentes con actuadores para interactuar con el sistema medido. El problema con esta propuesta es que limita al usuario a usar una aplicación de Android para hacer uso de las funciones; mientras que nuestra propuesta permite al usuario acceder desde cualquier dispositivo conectado a internet. Además, el sistema propuesto en este artículo es modular, lo que permite agregar más estaciones de monitoreo ambiental, sensores y actuadores; así como otorgar permisos de administrador a ciertos usuarios registrados.

Además, también se encuentra la propuesta de Danita, M., Blessy, M., Nithila, S., Namrata, S., Paul, J., 2018, en donde un sistema de control de condiciones (intensidad lumínica, humedad, etc.) se implementa en un invernadero. En dicha propuesta no se

otorga al usuario la capacidad de interactuar ni modificar el estado de los actuadores, lo cual puede ser preciso en caos muy particulares.

El desarrollo de este sistema de monitoreo remoto precisó la implementación de la metodología cascada, donde se realiza un estudio de los requerimientos técnicos para dar pie al inicio del proyecto. Definiendo el alcance del proyecto según los recursos disponibles. Durante el análisis se desarrolló un servidor web basado en Apache2, MariaDB, PHP 7.3, MySQL y PHPMyAdmin; basándonos bajo la filosofía LAMP (Linux, Apache, MySQL, PHP), los cuales son recursos completamente Open Sources. También se diseñó la base de datos y sitio web con conexión a la estación de monitoreo con sensores y microcontroladores. Para el diseño del sitio web se realizaron archivos .PHP, .HTML y .CSS donde se visualiza la información de la base de datos. Asimismo, el sitio web fue realizado siguiendo un diseño web responsivo. Para el diseño electrónico se utilizó la tarjeta Raspberry Pi 3B+, a los que se le conectan los sensores de: temperatura, humedad y presión atmosférica. Mismos que fueron programados y controlados con la Raspberry Pi y el lenguaje de programación Python; el cual utiliza para la lectura e inserción de datos en la página web.

2. Materiales y equipos

Para el desarrollo del proyecto, se utilizaron elementos de software y hardware para la implementación del sistema de internet de las cosas, en los cuales se implementa el desarrollo de su arquitectura.

2.1. Descripción del hardware utilizado

El hardware utilizado en el presente artículo incluye una tarjeta Raspberry Pi, una tarjeta Arduino Uno, un circuito integrado ATMEGA328P, los sensores BME280, YL-69, ML8511 y el receptor Neo-6, los cuales se describen a continuación:

Raspberry Pi: Es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo, el cual estimula la enseñanza de informática. El modelo original se convirtió en más popular de lo que se esperaba, hasta incluso vendiéndose fuera del mercado objetivo para usos como robótica.

Arduino UNO: Es una placa de microcontrolador basada en ATmega328P. Tiene 14 pines de entrada / salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz (CSTCE16M0V53-R0), una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio.

ATMEGA328P: Es un circuito integrado de alto rendimiento basado en un microcontrolador RISC, combinando 32 KB (ISP) flash una memoria con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interfaz serial orientada a byte de 2 cables, SPI puerto serial, 6-canales 10-bit Conversor A/D (canales en TQFP y QFN/MLF packages), temporizador "watchdog" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software.

Sensor BME280: Es un sensor de humedad especialmente desarrollado para aplicaciones móviles y wearables donde el tamaño y el bajo consumo de energía son parámetros clave de diseño. La unidad combina sensores de alta linealidad y precisión y es perfectamente factible para un bajo consumo de corriente, estabilidad a largo plazo y alta robustez de EMC. (C. A. Merlo, M. Antonio, R. Silva, C. Y. Sosa y M. Marciano, 2014).

Sensor YL-69: El sensor de humedad del suelo, se usa generalmente para detectar la humedad en la tierra o arena o cualquier sustrato permeable a la humedad y que se pueda enterrar. Por lo tanto, es perfecto para construir un sistema de riego automático o para controlar la humedad de tus plantas en un sistema “indoor”. El sensor está conformado por dos piezas o placas y un cable que une: la placa electrónica YL-38 (a la izquierda) y la sonda con dos patas, que detecta el contenido de agua YL-69 (a la derecha).

Sensor ML8511: Es un sensor UV, adecuado para adquirir intensidad UV en interiores o exteriores. El ML8511 está equipado con un amplificador interno, que convierte la fotocorriente en voltaje dependiendo de la intensidad de los rayos UV. Esta característica única ofrece una interfaz sencilla para circuitos externos como ADC. En el modo de apagado, típico corriente de espera es 0.1 A permitiendo una mayor duración de la batería.

Receptor Neo-6: Familia de receptores GPS independientes con el u-blox 6 de alto rendimiento motor de posicionamiento. Estos receptores flexibles y rentables ofrecen numerosas opciones de conectividad en una miniatura” (U-blox, 2020). Su arquitectura compacta y sus opciones de alimentación y memoria hacen que los módulos NEO-6 ideal para dispositivos móviles.

2.2. Descripción del software utilizado

Para la implementación de este artículo se utilizaron los lenguajes de programación PHP y Python, además de HTML, SQL y JavaScript; así como Fritzing y Proteus para simulación electrónica.

PHP: Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. (PHP, 2020).

Python: Lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. (Martelli, 2007).

HTML: Es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web. Otras tecnologías además de

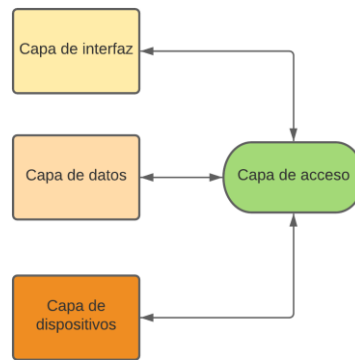


Fig. 1. Diagrama de capas.

HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). (HTML, 2020).

Fritzing: Esta plataforma “brinda apoyo a diseñadores y artistas para el desarrollo de dispositivos electrónicos mediante la simulación de los mismos” (Knöring, A., Wettach, R., Cohen, J., 2009), facilitando la conexión entre los mismo y permitiendo corregir errores y observar el comportamiento de nuestro circuito.

3. Arquitectura de un sistema de IoT

El éxito del IoT depende en gran medida de una arquitectura que permita que sea una tecnología dinámica y segura, por ello, la arquitectura debe cumplir ciertos requerimientos para que esta tecnología sea viable. A gran escala, debe permitir que la tecnología sea distribuida, donde los objetos puedan interactuar entre ellos, escalable, eficiente y segura (Hernández, 2020). La tecnología distribuida, implica que la información adquirida pueda provenir de diferentes lugares y, a su vez, procesada por máquinas o servidores diferentes. La arquitectura debe ser capaz de mostrar todos los componentes como un único sistema a los ojos de los usuarios y desarrolladores.

La interacción entre objetos conectados, hace referencia a que, muchas veces, los dispositivos no se pueden comunicar entre ellos; en el IoT, el intercambio de datos debe ser entre cualquier objeto y de forma bidireccional. Se requiere que entre los diferentes dispositivos exista interacción. Creando un proceso de estandarización de protocolos de comunicación.

La arquitectura escalable necesita de una arquitectura que permita la escalabilidad, es decir, que de si hoy se conectan 10,000 y mañana 1,000,000 el sistema funcione exactamente igual. Dando lugar a que el sistema o el entorno pueda crecer continuamente. La eficiencia energética consiste en tener una autonomía máxima, no se puede depender completamente de baterías de corta duración y, ya que están diseñadas para no apagarse, que su consumo no implique un aumento considerable del consumo de energía. Asimismo, una de las ideas fundamentales es integrarlo con energías renovables e inagotables.

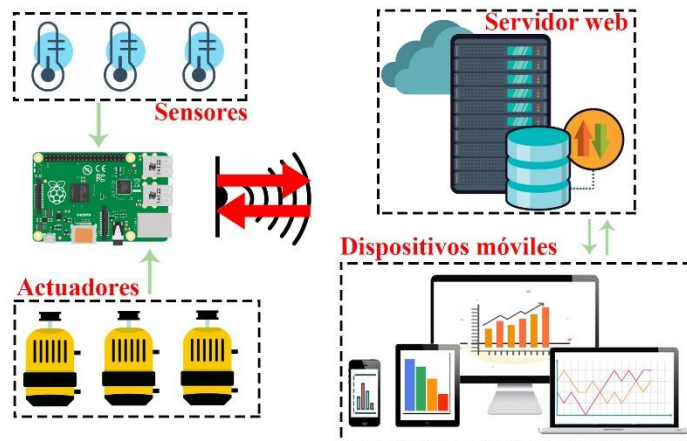


Fig. 2. Diagrama de funcionamiento.

Por otro lado, no se puede construir una arquitectura totalmente impenetrable. La seguridad en el IoT es algo que se debe tener en cuenta desde el principio, ya que no solo están en peligro nuestros datos, inclusive podrían estar vulnerable nuestra integridad física.

La arquitectura del sistema IoT utilizada, está conformado por la parte lógica y la física; que, a su vez, se clasifican en cuatro capas: capa de interfaz (página web), capa de datos (base de datos), capa de acceso (protocolos de comunicación) y la capa de dispositivos que integra los controladores, actuadores y sensores, ver Figura 1.

La capa de datos se encarga del almacenamiento, captura y consulta de los datos recolectados por la capa de dispositivos (sensores y actuadores). En la captura, los datos son obtenidos desde la capa acceso, la cual recibe los datos de los sensores y actuadores para, posteriormente, mostrarlos en la capa de interfaz y de ahí ser capturados y enviados vía WiFi a la base de datos para su almacenamiento y consulta por la página web en la capa interfaz.

La capa de acceso se encarga de transmitir la información entre la capa de interfaz, capa de datos y capa de dispositivos. Es, básicamente, el conjunto de protocolos que permiten la comunicación en nuestro sistema y que este dé un funcionamiento correcto. Asimismo, se encarga en enviar las acciones requeridas por el usuario al microcontrolador para que los actuadores cumplan la acción de usuario sobre el sistema.

La capa de interfaz se encarga de brindar al usuario la interacción con todo el sistema, permitiéndole consultar los registros almacenados en la base de datos. De igual manera, se permite a este recibir, en tiempo real, los datos enviados por los sensores y actuadores desde los microcontroladores y, por último, le otorga al usuario control sobre los actuadores.

La capa de dispositivos, se encarga de enviar datos o recibir órdenes desde y hacia el o los microcontroladores, para que estos lean los datos de los sensores y ordenen a los actuadores cumplir su respectiva función.

Como se puede ver en la Figura 2., se presentan los componentes divididos separados en “Hardware” y “Software” intercomunicados entre sí. Por el lado del hardware

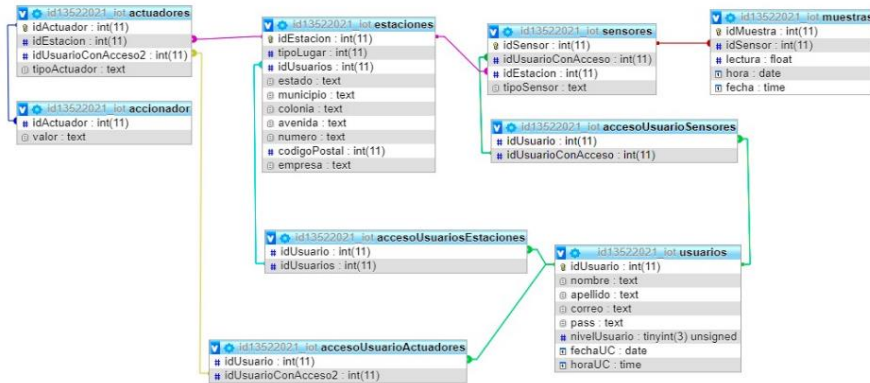


Fig. 3. Modelo de base de datos relacional.

contamos con una Raspberry Pi 3B+ funcionando como microcontrolador de los sensores y actuadores conectados a la misma. Por el lado del software tenemos un servidor web con la base de datos y la interfaz web con la que el usuario podrá interactuar con el sistema. Conectados ambos por medio de internet.

4. Desarrollo de un prototipo de estación para monitoreo remoto

La naturaleza multidisciplinaria de este sistema de monitoreo remoto y la complejidad de éste precisa de cuatro pilares fundamentales para el proceso de elaboración y desarrollo del mismo. Dichos pilares que son:

- Diseño de base de datos.
- Diseño del sitio web.
- Diseño electrónico.
- Diseño mecánico.

4.1. Diseño de la base de datos

Modelo entidad-relación: Es un tipo de diagrama de flujo que ilustra cómo las “entidades”, como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas WE se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación.

Modelo relacional: Utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. Se escogió el modelo relacional para el diseño de la base de datos.

Las tablas pertenecientes a la base de datos diseñada, así como su relación se muestra en la Figura 3.

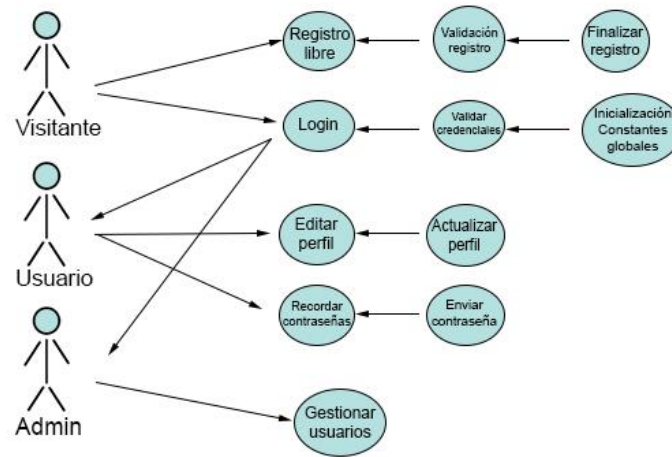


Fig. 4. Diagrama de jerarquías de usuarios.



Fig. 5. Interfaz web.

El modelo relacional presentado corresponde al modelo desarrollado para este proyecto. Como se puede observar, se presentan diversas tablas, donde se almacena información específica como en la tabla “usuarios”, “sensores” y “estaciones”; asimismo se muestra las tablas de control que tienen la finalidad de facilitar la interacción entre dos tablas para así generar una relación entre las tablas y almacenar la información referente al acceso de cada usuario, de esta forma, es posible controlar que puede y que no puede verse y/o modificar un usuario en particular, manteniendo así el control en la interfaz.

Inicio de Sesion

Usuario:

Contraseña:


No soy un robot 
reCAPTCHA
Privacidad - Condiciones

Fig. 6. Inicio de sesión.

Vista... Período... Recargar

Tablas Últimas 24 horas

Buscar

Fecha	Temperatura	Humedad
2020-08-07 11:00:00	24	65
2020-08-06 11:00:00	55	12
2020-08-05 10:58:10	12	1
2020-08-04 10:58:10	12	1
2020-08-03 11:08:10	12	11
2020-08-02 11:08:10	12	11
2020-08-01 10:58:10	28	10
2020-07-31 10:58:10	28	10
2020-07-30 11:00:58	22	15
2020-07-29 11:00:10	19.1	22
2020-07-28 11:00:01	21.5	33
2020-07-27 11:00:10	23	33

Fig. 7. Tablas de datos.

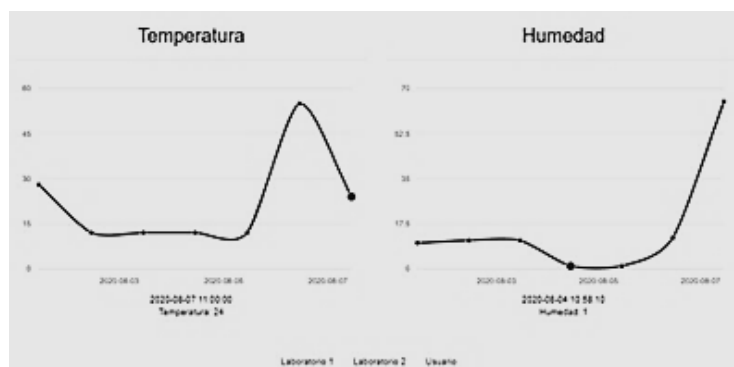


Fig. 8. Gráficas de datos.

Esta base de datos se encarga de almacenar todos los datos tanto de usuarios como de los sensores, actuadores, historiales de muestreo y de accionamiento; posteriormente, como se puede apreciar en el siguiente segmento, esta base de datos interactúa con una página web ya montada en un dominio.

Adicional a las bases de datos diseñadas y desarrolladas, es necesario también generar las jerarquías de usuarios, además de los métodos que son posibles realizar en cada una de su asignación.

4.2. Diseño del sitio web

Para el diseño de la página web se trabajó con los lenguajes de programación: HTML, PHP y JavaScript. Inicialmente se realizaron pruebas en servidor local dentro de la Raspberry Pi y, posteriormente, se trasladó a un dominio web que cuenta con su propio servidor. La navegación del sitio web fue diseñada para ser intuitiva y fácil de usar por el usuario. Contando con cuatro botones que redireccionan al servicio o información requerida: Inicio (para regresar a la página principal), Servicios, Galería y Contacto; así mismo se presentan datos generales. Ver Figura 4. El sitio web desarrollado puede consultarse en la siguiente url: <http://iot-udg.000webhostapp.com/>.

Para iniciar sesión se utiliza un formulario (Figura 5). Al iniciar sesión se personaliza a cada usuario, además, permite otorgar privilegios de administrador a ciertas cuentas. Esto se logra mediante una consulta a la base de datos mediante una “query” (consulta) que se crea a partir del ingreso del formulario hecho por el usuario. Esto mediante el método POST para así proteger la información del usuario, esto implica mayor seguridad en el manejo de información. Una vez iniciada la sesión, al usuario se le otorgan permisos y acceso a las distintas secciones del sitio web.

En la navegación como usuario registrado y con privilegios, es posible acceder a la página principal de este proyecto la cual es la encargada de mostrar los datos que se obtiene del hardware.

Como podemos apreciar en la Figura 6 y 7, se agrega un indicador, el cual presenta el espacio físico que se está monitoreando. En este caso tenemos el laboratorio 1. Asimismo, se presenta un par de menús desplegables encargados de controlar el tipo de vista y el periodo a mostrar. El tipo de vista se refiere a las dos posibles maneras de recuperar los datos de las tablas de muestras en la base de datos que hace referencia al historial de muestras realizados, para poder verlas como gráficas o tablas; mientras que el menú de periodo corresponde a un filtro que permite recuperar los datos de las últimas 24 horas, los últimos 7 días o el último mes.

4.3. Diseño electrónico

Haciendo uso del software de simulación Fritzing y Proteus; se presenta la conexión para cada uno de los sensores y el GPS.

En términos generales, el software Fritzing se utilizó con el propósito de ilustrar, de una manera sencilla, las conexiones entre los componentes del sistema; así como conseguir una simulación más rápida para el prototipado del mismo. Los diagramas hechos en Fritzing resultan ser más fáciles de entender e interpretar no solo para los expertos, sino para los usuarios en general.

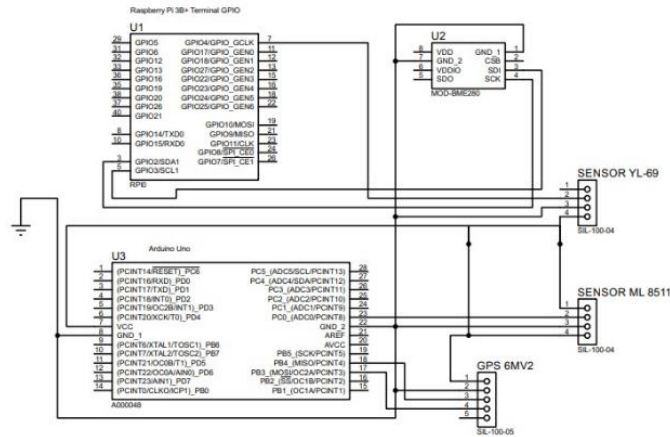


Fig. 9. Diagrama Proteus.

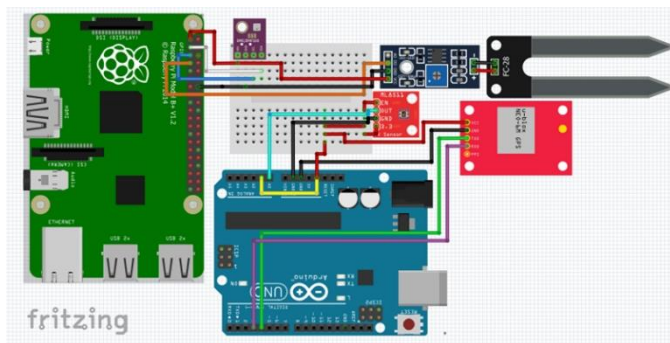


Fig. 10. Diagrama Fritzing.

Por su parte, Proteus nos ofrece un entorno más profesional y detallado del proyecto. Para el diseño, Proteus nos ofrece un entorno más profesional y detallado del proyecto.

Para el diseño, se utilizaron un total de 9 piezas independientes y 3 preensambles para lograr el ensamble completo de esta estación. Cada elemento del sistema cuenta con una carcasa externa para protección contra el ambiente, por lo que tanto los sensores como la propia Raspberry Pi estarán protegidos.

5. Resultados

Esta estación de monitoreo remoto desarrollada es un proyecto multidisciplinar que cumple con la función de mostrar información en tiempo real de datos (inicialmente) atmosféricos recolectados y procesados por una tarjeta Raspberry Pi 3B+ con ayuda de un Arduino UNO. Se diseñó una página web montada en su propio dominio con características responsivas, misma que permite la interacción con el usuario y el acceso a la información recolectada por los sensores. Dicho sitio web también permite otorgar privilegios de administrador a usuarios específicos y seleccionar el laboratorio (estación

de monitoreo) al que se quiera acceder; permitiéndonos medir variables en más de un solo lugar. Toda la información proveniente del exterior es capaz de ser organizada y proceda gracias a una base de datos relacional. Por lo que los alcances de este proyecto son bastos.

6. Conclusiones

Debido a la naturaleza multidisciplinaria de este proyecto y al auge de estas nuevas tecnologías de intercomunicación, los alcances de este se amplían a muchos más sectores de la economía (como la ganadería y agricultura), sectores en donde se puede mejorar la calidad de la producción y mejorar la experiencia de los trabajadores y clientes. Sin embargo, debe considerarse el estudio posterior con el fin de visualizar el diseño, desarrollo del comportamiento y tendencias del mercado. Las herramientas y tecnologías de hardware y software “Open Source” utilizados en este sistema de monitoreo remoto demostraron su alta capacidad, tanto en alcance como en trabajos previos realizados. Estas herramientas muestran se alta capacidad, tanto en alcance como en trabajos previos realizados. Estas herramientas muestran un alto potencial de escalabilidad y crecimiento futuro en trabajos de IoT e Industria 4.0.

Referencias

1. Morteo, F., Bocalandro, N.: Un enfoque práctico de SQL. Ediciones cooperativas (2004)
2. Knowlton, J.: Python. Fernández-Vélez, M.J. (ed.), Anaya Multimedia-Anaya Interactiva (2009)
3. Marteli, A.: Python. Guía de referencia. Gorjón-Salvador, B. (ed.) Anaya Multimedia-Anaya Interactiva (2007)
4. Merlo, C.A., Antonio, M., Silva, R., Sosa, C.Y., Marciano, M.: Sistemas mecánicos subactuados pendulares, Boletín UPIITA, 42(30) (2014)
5. Block, D.J.: Mechanical design and control of the pendubot. University of Illinois (1996)
6. Andrade-Santamaría, R.G., Fuentes-Castillo, R.C.: Análisis, diseño y construcción del Pendubot. Escuela Politécnica Nacional de Ecuador (2000)
7. Cortés, C., Landeta, J., Chacón, J.: Implicaciones y perspectivas futuras. El entorno de la industria 4.0. Dialnet (2020)
8. Html: Lenguaje de etiquetas de hipertexto. Documentación web de MDN (2020)
9. PHP: ¿Qué puede hacer PHP? - Manual. Php.net (2020)
10. Hernández, L.: ¿soy?, Arquitectura IoT, prototipando los dispositivos del futuro. Programar fácil con Arduino (2020)
11. BBC - dot.Rory: A 15 pound computer to inspire young programmers. Bbc.co.uk (2020)
12. Palacios, P., Córdova, A.: Approximation and Temperature Control System via an Actuator and a Cloud: An Application Base don the IoT for Smart Houses. In: International Conference on eDemocracy and eGovernment (ICEDEG), pp. 241–245 (2018)

13. Danita, M., Blessy, M., Nithila, S., Namrata, S., Paul, J.: IoT based Automated Greenhouse Monitoring System. In: Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1933–1937 (2018)
14. Knöring, A., Wettach, R., Cohen, J.: Fritzing – A tool for advancing electronic prototyping for designers. In: Proceedings of the Third International Conference on Tangible and Embedded Interaction (TEI'09) (2009)
15. Silberschatz, A., Korth, H., Sudarshan, S.: Fundamentos de bases de datos. McGraw-Hill/Interamericana de España, S.A.U. (2002)

Time2Vec Embedding on a Seq2Seq Bi-directional LSTM Network for Pedestrian Trajectory Prediction

Victor Peñaloza

Universidad Galileo, Research Laboratory in Information
and Communication Technologies (RLICT),
Guatemala

victorsergio@galileo.edu

Abstract. This paper presents a description of a proposed architecture for the pedestrian trajectory prediction task on the TrajNet dataset. This task aims to determine future final agent position and future agent movement trajectory using previous agent positional information. The proposed architecture is a Seq2Seq bi-directional LSTM (Long short-term memory) network with a Time2Vec input embedding layer. After experimenting and evaluating, our experimental results show that adding a Time2Vec input-embedding layer improves the performance of a Seq2Seq bi-directional LSTM recurrent model on the pedestrian trajectory prediction task.

Keywords: Pedestrian, Trajectory Prediction, Time2vec, Embedding, Sequence to Sequence, Trajnet.

1 Introduction

Many researchers have studied and discussed human mobility analysis due to its applications in robotics, autonomous driving systems, and mobile telecommunications systems. With the growing size of collected data captured by sensors and smartphones at present, it is possible to gather much data about human mobility to be processed later to identify patterns in data [30]. However, some applications need to have a system capable of anticipating and determining a future human position in an online way, such as the case of autonomous driving systems, in which it is helpful to know in advance the pedestrian movements in streets to avoid accidents.

In this paper, we focused on predicting the future path for an agent, using previous movement information to anticipate the future movement. The contributions of this work can be summarized as follows:

1. Show a simple Seq2Seq architecture to predict future pedestrian movement, using an encoder-decoder bi-directional LSTM (Long short-term memory) recurrent model with an input embedding layer.

2. Compare the performance between a model using an input embedding layer and a model without it. We show experimentally that for pedestrian path prediction on TrajNet [31] dataset, the use of Time2Vec [20] embedding layer improves the Seq2Seq model results.

This paper comprises eight sections: the first one presents an introduction to this task and study. The second section describes some previous work in the pedestrian trajectory prediction task. The third section describes the problem statement, metrics, and used dataset. The fourth section presents some informational background of blocks used to build the proposed system. The fifth section describes the system architecture proposed. The sixth section describes the experiments performed. The seventh section shows the results achieved in the model comparisons. The last section presents some conclusions and future work to continue the experiments on this task.

2 Related Work

Many deep learning models have been proposed for human trajectory prediction with data-driven approaches to learning the human motion. LSTM [18] based architectures have been shown to be capable of learning general human movement and can predict their future trajectories [2]. Simple encoder-decoder architectures have been shown promising results in pedestrian trajectory prediction, using only position information about pedestrians [6]. Besides, more complex approaches using Generative Adversarial Networks [15] have been used to predict the motion of pedestrians interacting with others [4].

3 Problem Statement

3.1 Problem Formulation

The goal of this task is to predict a future agent trajectory without using human-human interaction data. We formulate this problem as a sequence generation problem, where we assume agent input trajectories as a sequence $X = (X_1, X_2, \dots, X_8)$ to predict a future trajectory $X_{pred} = (X_9, X_{10}, \dots, X_{20})$. Whereas $X_i = (x_i, y_i)$ are (x, y) coordinates in a 2D plane for any time-instant i .

3.2 Dataset Description

For the experiments, we used the TrajNet Benchmark Dataset [31]. This dataset bundles the most common datasets used for pedestrian trajectory analysis. TrajNet challenge also provides a common platform to compare path prediction approaches. We train our models using the World H-H TRAJ dataset, in which the pedestrian trajectories are formed by x, y world coordinates in meters.

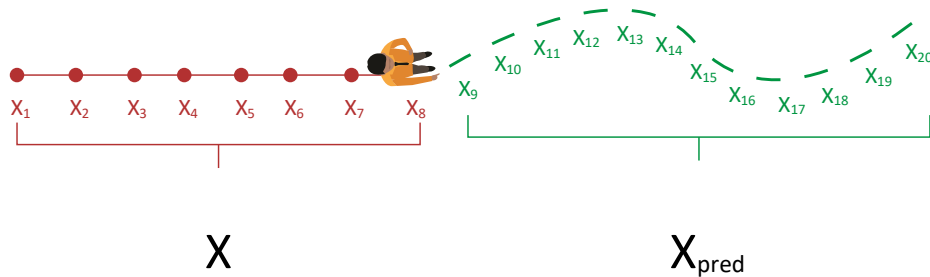


Fig. 1. This task's main goal is to use eight previous steps to predict a future agent trajectory comprised of 12 steps. The agent position on each step is defined by $X_i = (x_i, y_i)$ a coordinate in a 2D plane.

3.3 Metrics and Evaluation

We use the following error metrics to evaluate the models' performance over a test dataset: average displacement error (ADE) and the final displacement error (FDE). ADE is defined as the average of Euclidian distances between ground truth and the prediction overall predicted time steps. FDE is defined as the Euclidian distance between the predicted final position and the ground truth final position. We observed eight sequential picture frames (2.8 seconds) to form an input sequence to predict a future sequence comprised of 12 sequential picture frames (4.8 seconds).

4 Background

For better understanding, this section introduces some fundamental concepts of the blocks used to build our model. The following concepts are presented: LSTM [18], bi-directional LSTM [33], Seq2Seq learning, embedding, and Time2Vec [20].

4.1 LSTM, Long Short-Term Memory

Neural Recurrent Networks (RNN) has been proved to be an effective architecture for diverse learning problems with sequential data [19]. LSTM networks were designed to deal with vanishing gradients problem in learning long-term dependencies [18]. However, LSTMs can still suffer from instability with improper initialization [27]. Additionally, LSTM architectures are highly complex, having about four times more parameters than a simple RNN with an equal number of hidden layers [29]; therefore, LSTM architectures have higher memory requirements [32].

LSTM architecture is mainly comprised of a memory cell and unit gates. The network can use these gates to decide when to keep or overwrite information in the memory cell and decide when to prevent perturbing other cells by a memory cell output [18].

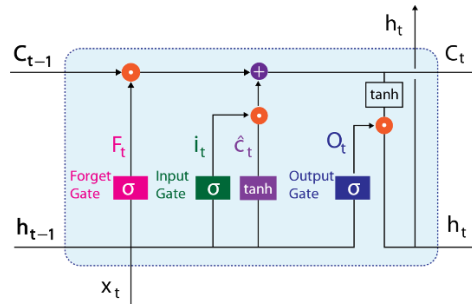


Fig. 2. The architecture of a LSTM block.

4.2 BLSTM, Bi-directional LSTM

A bi-directional RNN (BRNN) [33] has the complete information from the past and the future for every point in a given sequence. A bi-directional RNN (BRNN) is comprised of two separate recurrent neural nets, both of which are connected to the same output layer. Each training sequence forwards and backward feeds the bi-directional RNN (BRNN). Thus for every point in a given sequence, a bidirectional RNN (BRNN) has complete, sequential information about all points before and after it [16]. One limitation of the bi-directional RNN(BRNN) is that it is not appropriate for the online setting, as it is implausible to know sequence elements that have not been observed [25]. Nevertheless, we must distinguish between tasks that require an output after every input (true online tasks) and those where outputs are only needed at the end of some input segment [16]. This paper’s pedestrian trajectory task is in the second class; therefore, the bidirectional RNN (BRNN) can be used without a problem. The LSTM and BRNN are compatible, and the conjunction of these two ideas is termed as BLSTM (bidirectional LSTM) [16].

4.3 Recurrent Sequence-to-Sequence Learning

Seq2Seq is a structure designed to modeling sequential problems in which input sequences and outputs sequences are variable in length [36].

The most common structure associated with Seq2Seq learning is an encoder-decoder structure with recurrent neural networks [34,5]. Being an LSTM [18] the most used neural recurrent unit in the encoder and decoder section.

The encoder processes an input sequence of m elements and returns representations of z in a fixed-length vector. The decoder takes z and generates an output sequence element at a time [13].

Despite the success of this type of architecture in machine translation tasks, due to the fact of compressing all the necessary information of a source sentence into a fixed-length vector, it could be hard for the neural network to cope with long sentences, deteriorating the performance of the model as the length of an input sentence increases [5,8]. Different attention-based models were successfully applied to overcome this [5,26]. Attention-based models are an extension for the encoder-decoder architecture

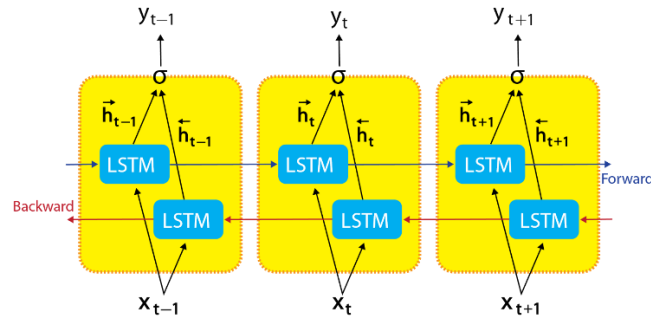


Fig. 3. Bi-directional LSTM architecture.

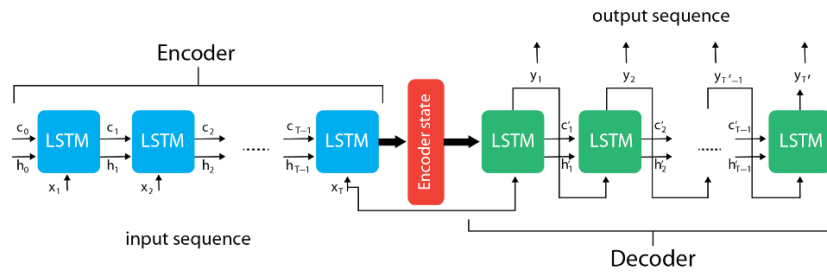


Fig. 4. Encoder-Decoder structure from a Seq2Seq model.

that allows the model to automatically search for relevant parts of a source sentence to predict a target element without encoding a whole input sentence into a single fixed-length vector [5].

4.4 Embedding

Many factors affect the Machine Learning model's performance; among these factors, the principal factor is the representation and quality of the data used for training [21].

For example, in the Natural Language Processing (NLP) area, sequences of sentences, words, or characters are used as input to recurrent neural networks based models; generally is desired to provide a compact representation of these features instead of sparse representation [35].

Diverse techniques have been proposed for feature representation in NLP as one-hot encoding representation [38], continuous bag of words (CBOW) [28], and word-level embedding [14].

Word level embedding goal is to learn a representation in which ideally words semantically related are highly correlated in the representation space [35]; in this way, good results have been obtained in classification [37,24,12] and sentiment analysis NLP tasks [10,3].

4.5 Time2Vec Embedding

In many applications with sequential data, time is an important feature. In many models using recurrent neural networks, the time is not taken in count as a feature. When time is observed as an essential factor, this is generally added as an additional feature [9,11,23].

Time2Vec is a vector representation (*embedding*) to time that it can be combined easily with other architectures [20].

Time2Vec as a time representation was designed with three essential properties in mind: 1- periodicity, the capability to capture periodic and non-periodic patterns; some events can occur periodically(e.g., the weather over different seasons), but others events may be non-periodic and only happens after a point in time, 2- being invariant to time rescaling, since time can be measured in different scales(e.g., days, hours, seconds), and 3- simplicity, should be easy to be added to existent architectures [20].

For a given scalar notion of time τ , Time2Vec of τ , denoted as $\mathbf{t2v}(\tau)$, is a vector of size $k + 1$, defined as follows:

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \phi_i, & \text{if } i = 0, \\ F(\omega_i \tau + \phi_i), & \text{if } 1 \leq i \leq k, \end{cases}$$

where $\mathbf{t2v}(\tau)[i]$ is the i^{th} element of $\mathbf{t2v}(\tau)$, F is a periodic activation function, and ω_i s and ϕ_i s are learnable parameters.

When using $F = \text{sine}$, for $1 \leq i \leq k$, ω_i and ϕ_i are the frequency and the phase-shift of the *sine* function.

This function permits learn periodic behaviors. The linear term section can be used to represent no periodic behaviors [20].

5 System Description

The main idea of the proposed system is the combination of already existing techniques that have shown improvements in other research areas as Natural Language Processing (NLP), and apply these techniques to the pedestrian trajectory prediction task.

We proposed the use of a recurrent Seq2Seq architecture that uses bi-directional LSTM [33,18] layers to build the encoder and decoder with a Time2Vec embedding layer on encoder inputs to feed a better positional sequence representation to the model. Fig. 5 shows the proposed architecture for pedestrian trajectory prediction.

We choose a Seq2Seq multi-step architecture because it has been observed that single-step prediction models that use windowing to predict a complete future path are prone to error accumulation, and Seq2Seq models have been showing better results in similar tasks [36].

Although Time2Vec originally was proposed for a better time representation, we can see the sequences of coordinates (*positions in the space*) as signals that include an implicit time relation and use Time2Vec embedding to create an embedding that represents this signal with a learnable frequency and phase shifts of *sine* function.

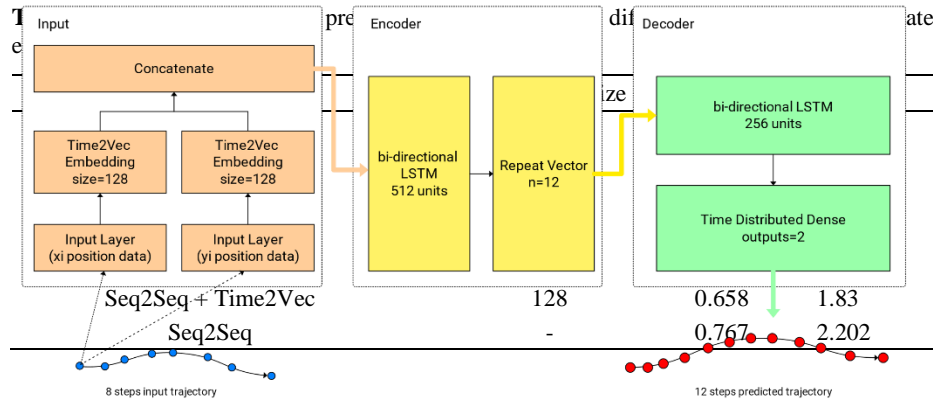


Fig. 5. Proposed Seq2Seq + Time2Vec embedding architecture.

6 Experiments

For performance comparison purposes, we build a Seq2Seq bi-directional LSTM architecture that not uses Time2Vec embedding and compares it against the proposed Seq2Seq bi-directional LSTM model with Time2Vec embedding to observe if the use of a Time2Vec embedding is beneficial to the performance of the Seq2Seq model in trajectory prediction results.

For a better comparison, all models were trained for 81 epochs (*one epoch is a full training pass over the entire dataset such that each example has been seen once*).

The models were trained using a learning rate equal to $1e-5$, a batch size equal to 8, and MSE as a loss function.

For comparison, we used the error metrics: average displacement error (ADE) and the final displacement error (FDE) in meters.

We run various experiments using different embedding vector sizes, 128, 256, and 512, respectively, to compare the effect of embedding size in the model's performance.

All the models were created using TensorFlow [1], running on Google Colab environment, and using Weight and Biases [7] for monitoring training metrics.

7 Results

Table 1 shows our experimental results; it can be observed that using Time2Vec embedding improves the Seq2Seq model in predicting more accurate trajectories. Time2Vec embedding helps the model feeding the Seq2Seq model with a better representation of positional trajectories sequences.

Notice that runs with bigger embedding vector sizes improves both final displacement error (FDE) and average displacement error (ADE) results.

All results from Table 1 were calculated using a test set provided by the TrajNet Official Benchmark website [31].

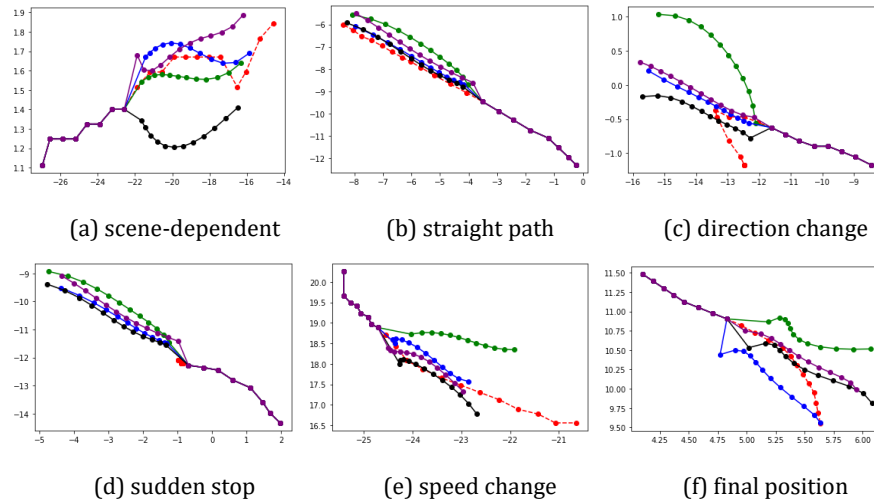


Fig. 6. Some examples of hard to predict trajectories for the models. The ground truth trajectories are shown in red, and the model's predictions are: 128 embedding dimension (blue), 256 embedding dimension (black), 512 embedding dimension (purple), and no-embedding (green). The scale of the graph axis is in meters.

Additionally, the Table 1 shows results achieved by state of the art methods for comparison. We compare our model with the best models with references found on Trajnet Benchmark Dataset [31] for a fair comparison.

However, this comparison must be taken carefully since other methods as Social LSTM [2] and Social Forces (EWAP) [17] use pedestrian interaction information, while our proposed method only uses pedestrian positional information. The closest and fair comparison is against RNN-Encoder-Dense (RED) predictor [6] due to its simplicity and similarity, using a recurrent-encoder with a dense multi-layer perceptron (MLP) layer stacked architecture and using only positional information.

Notice that, although our results are lower than the state of the art models in the pedestrian trajectory prediction task, this work aimed to research better representations for positional data in pedestrian trajectories.

Fig. 6 shows trajectories hard to predict for the models. In Fig. 6c, it can be observed a sudden change of direction; the pedestrian decides to return on his path; this type of behavior is hard to predict even for humans. Fig. 6d depicts another hard to predict case; in this scenario, the pedestrian stop could be caused for a red traffic light. To predict this trajectory correctly, additional information for the model as traffic light state information could be beneficial.

For the case shown in Fig. 6e, it can be observed that the pedestrian decides to increment its speed suddenly; for example, due to the pedestrian was in a hurry, it is hard to predict this behavior due to the prediction was made using the previous constant speed. This problem could be addressed by creating models that could predict trajectories using small future step windows. It should be noticed that it is easy for the models to predict the movement when the speed is constant and follows a commonly used trajectory in the scene; for example, as shown in Fig. 6b the pedestrian follows a

Table 2. Average displacement error (ADE) and final displacement error (FDE) for the special cases prediction shown in Fig. 6.

Embedding Size	Scene Dependent		Straight Path		Direction Change		Sudden Stop		Speed Change		Final Position	
	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
none	0.625	1.739	0.517	0.563	1.003	3.497	1.618	4.912	0.786	2.190	0.306	1.054
128	0.533	1.285	0.503	0.444	0.785	3.346	1.292	4.271	0.799	2.44	0.138	0.010
256	0.759	1.956	0.327	0.157	0.815	3.382	1.469	4.685	0.617	2.049	0.120	0.516
512	0.577	1.639	0.381	0.682	0.941	3.638	1.502	4.529	0.78	2.449	0.126	0.537

sidewalk and maintains a constant speed. In these types of cases, all the models exhibit similar results in the predictions.

It is worth to notice that the models learn scene dependent features, as shown in Fig. 6a, and different models could predict different paths due to obstacles found in the scene. It could be worthy of continuing research this problem with multi-modal models that could output a probability distribution over different possible paths. At last, in some cases, the model gives more importance to predict a correct final position instead of predicting intermediate steps correctly, as shown in Fig. 6f.

In Table 2 are the average displacement error (ADE) and final displacement error (FDE) for the mentioned special cases. Note that in these special cases, the use of bigger embedding does not mean better prediction results. However, as observed in Table 1, on average, evaluating over the complete test set, the use of bigger embedding achieves better prediction results. This contradiction makes us think that the dataset comprises a large variety of cases, in which some of them are easy to predict for some models, and others are hard for them. A possible solution could be to use an ensemble of models to select the best prediction among different models that could be successful in specialized cases.

8 Conclusions and Future Work

This work described an architecture that uses a Seq2Seq bi-directional LSTM recurrent model with an input embedding layer to be compared with a similar architecture without an input embedding layer to show the benefits of having an adequate input data representation for models on the pedestrian trajectory prediction task.

According to our experimental results, although embedding improves the proposed architecture's performance in this task, now the embedding vector size hyperparameter appears. It can be observed that the size of this vector has a model's performance impact and must be adjusted correctly.

Additionally, it can be worth testing other architectures as Generative Adversarial Network (GAN) [15] based, transformed based, and Convolutional Neural Network (CNN) [22] based on comparing embedding performance in the pedestrian trajectory prediction task.

Acknowledgments: This work was supported by Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación (FISICC) and Research Laboratory in Information and Communication Technologies (RLICT), both part of Universidad Galileo from Guatemala. We thank Jean-Bernard Hayet, PhD, for helpful advice and research directions related to this work.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems. <https://www.tensorflow.org/>, software available from tensorflow.org (2015)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: Human trajectory prediction in crowded spaces. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–971 (2016)
3. Altowayan, A.A., Tao, L.: Word embeddings for Arabic sentiment analysis. In: IEEE International Conference on Big Data (Big Data) pp. 3820–3825 (2016)
4. Amirian, J., Hayet, J.B., Pettré, J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 2964–2972 (2019)
5. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2015)
6. Becker, S., Hug, R., Hübner, W., Arens, M.: Red: A simple but effective baseline predictor for the trajnet benchmark. In: ECCV Workshops (2018)
7. Biewald, L.: Experiment tracking with weights and biases. <https://www.wandb.com/>, software available from wandb.com (2020)
8. Cho, K., Merriënboer, B.V., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: SSST@EMNLP (2014)
9. Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J.: Predicting clinical events via recurrent neural networks. Proceedings of machine learning research 56, pp. 301–318 (2016)
10. Deho, B.O., Agangiba, A.W., Aryeh, L.F., Ansah, A.J.: Sentiment analysis with word embedding. In: IEEE 7th International Conference on Adaptive Science & Technology (ICAST), pp. 1–4 (2018)
11. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1555–1564. KDD '16, Association for Computing Machinery, New York (2016)
12. Ge, L., Moh, T.: Improving text classification with word embedding. In: IEEE International Conference on Big Data (Big Data), pp. 1796–1805 (2017)
13. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.: Convolutional sequence to sequence learning. In: ICML (2017)

14. Goldberg, Y.: Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1), pp. 1–309 (2017)
15. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: *NIPS* (2014)
16. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5), pp. 602–610 (2005)
17. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Phys. Rev. E.*, 51, pp. 4282–4286 (1995)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Computation*, 9, pp. 1735–1780 (1997)
19. Karpathy, A., Johnson, J.E., Fei-Fei, L.: Visualizing and understanding recurrent networks. *ArXiv abs/1506.02078* (2015)
20. Kazemi, S., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., Brubaker, M.A.: Time2vec: Learning a vector representation of time. *ArXiv abs/1907.05321* (2019)
21. Kotsiantis, S.B., Kanellopoulos, D., Pintelas, P.E.: Data preprocessing for supervised learning. *World Academy of Science, Engineering and Technology. International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 1, pp. 4104–4109 (2007)
22. LeCun, Y., Haffner, P., Bottou, L., Bengio, Y.: Object recognition with gradient based learning. In: *Shape, Contour and Grouping in Computer Vision* (1999)
23. Li, Y., Du, N., Bengio, S.: Time-dependent representation for neural event sequence prediction (2018)
24. Lilleberg, J., Zhu, Y., Zhang, Y.: Support vector machines and word2vec for text classification with semantic features. In: *IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 136–140 (2015)
25. Lipton, Z.C.: A critical review of recurrent neural networks for sequence learning. *ArXiv abs/1506.00019* (2015)
26. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. *ArXiv abs/1508.04025* (2015)
27. Mehdipour-Ghazi, M., Nielsen, M., Pai, A., Modat, M., Cardoso, M., Ourselin, S., Sørensen, L.: On the initialization of long short-term memory networks. *ArXiv abs/1912.10454* (2019)
28. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
29. Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., Ranzato, M.: Learning longer memory in recurrent neural networks. *CoRR abs/1412.7753* (2015)
30. Morzy, M.: Mining frequent trajectories of moving objects for location prediction. In: *Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition*, pp. 667–680, Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
31. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint* (2018)
32. Salehinejad, H., Baarbe, J., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. *ArXiv abs/1801.01078* (2018)
33. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), pp. 2673–2681 (1997)

34. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. ArXiv abs/1409.3215 (2014)
35. Torfi, A., Shirvani, R.A., Keneshloo, Y., Tavaf, N., Fox, E.A.: Natural language processing advancements by deep learning: A survey (2020)
36. Wang, C., Ma, L., Li, R., Durrani, T.S., Zhang, H.: Exploring trajectory prediction through machine learning methods. *IEEE Access* 7, pp. 101441–101452 (2019)
37. Seyhmus Y., Toklu, S.: A deep learning analysis on question classification task using word2vec representations. *Neural Computing and Applications* 32, pp. 2909–2928 (2020)
38. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*, (1), pp. 649–657. NIPS'15, MIT Press, Cambridge, MA, USA (2015)

Metodología para controlar un robot móvil con lógica difusa

David Luna, Ana C. Zenteno, María del C. Santiago, Yeiny Romero,
Judith Pérez, Gustavo T. Rubín

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

divanso@hotmail.com, {ana.zenteno,
marycarmen.santiago, yeiny.romero, judith.perez,
gustavo.rubin }@correo.buap.mx

Resumen. En este documento se describe el proceso para lograr el control de un robot móvil Scribbler con lógica difusa. Inicialmente se desarrolló un simulador para este proyecto y se implementaron los algoritmos en amplitud y primero el mejor. Se utilizan estos algoritmos al ser algoritmos de búsquedas sin información y con información del SLAM. Paso siguiente, se postula una poda de los nodos cercanos a los objetos del mapa utilizando una función triangular. Posteriormente se utilizan las variables independientes distancia y velocidad para crear un par de conjuntos difusos que se implementan en el modelo de Sugeno [13], se obtiene la variable dependiente frenar y finalmente se realizó el proceso del controlador difuso para desarrollar un algoritmo de búsqueda con lógica difusa.

Palabras clave: Controlador Difuso, Lógica Difusa, SLAM, Robot Scribbler.

Methodology for Control a Mobile Robot with Fuzzy Logic

Abstract. This document describes the process to achieve control of a Scribbler mobile robot with fuzzy logic. Initially a simulator was developed for this project and the algorithms were implemented in amplitude and first the best. These algorithms are used as they are search algorithms without information and with information from the SLAM. Next step, a pruning of the nodes near the map objects is postulated using a triangular function. Subsequently, the independent variables distance and speed are used to create a pair of fuzzy sets that are implemented in Sugeno's model [13], the dependent variable brake is obtained and finally the fuzzy controller process was carried out to develop a search algorithm with diffuse logic.

Keywords: Fuzzy Controller, Fuzzy Logic, SLAM, Robot Scribbler.

1. Introducción

Los primeros robots móviles se desarrollaron como una estructura mecánica anclada en uno de sus extremos. La robótica es la ciencia que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el ser humano, esta rama computacional trabaja en la autonomía de los móviles. Los robots móviles tienen como precedente los dispositivos electromecánicos; tales como el denominado “micro mouse”, creado para desarrollar funciones como descubrir caminos en laberintos [7]. En los años setentas aparecieron robots móviles dotados de autonomía, basados en sistemas de visión [8]. Posteriormente se crearon para interiores por parte de Bares, realizados en el Carnegie Mellon University de Pittsburg EEUU [9]. El SLAM se presentó por primera vez en un panfleto sobre robots móviles en el simposio internacional de investigación robótica de 1995 [10]. Se alcanzó un acercamiento entre el SLAM basado en el filtro Kalman y los métodos probabilísticos para la localización y mapeo introducidos por S. Thurn, en la conferencia internacional de robótica y automatización (ICRA) del Institute of Electrical and Electronics Engineers (IEEE) de 1999. El trabajo con SLAM atrajo a 15 investigadores que buscaban soluciones para la complejidad, la asociación de la información y retos de la implementación [11]. En el año de 2012 se empezó a usar el Kinect en el SLAM, en el proyecto localización y generación de mapas del entorno de un robot utilizando este dispositivo de la Universidad Politécnica de Valencia [12]. El concepto de conjuntos difusos fue propuesto por el matemático Lofti Ali Asker Zade en la Universidad de California, Berkeley, la lógica clásica es precisa, a diferencia de la lógica difusa, esta última se basa en la percepción. El objetivo de esta técnica es imitar la toma de decisiones del cerebro humano [4].

John McCarthy en Dartmouth, acuñó por primera vez el concepto de inteligencia artificial como la ciencia de hacer máquinas inteligentes [1], la cual es la base de la tercera revolución industrial formulada por Jeremy Rifkin [2]. Por otro lado, la lógica difusa se fundamenta en determinar el razonamiento y pensamiento humano basándose en la percepción [3], con el objetivo de imitar la toma de decisiones del cerebro humano [4], este conjunto se caracteriza por la función de pertenencia de la ecuación 1:

$$\mu_A : U \rightarrow [0,1], \quad (1)$$

donde a cada elemento x de U se le asocia un número $\mu_A(x)$ del intervalo $[0,1]$, que representa el grado de pertenencia de x al conjunto difuso A y U es la *universo de discurso*, que se utiliza para ser implementado en el controlador difuso compuesto por

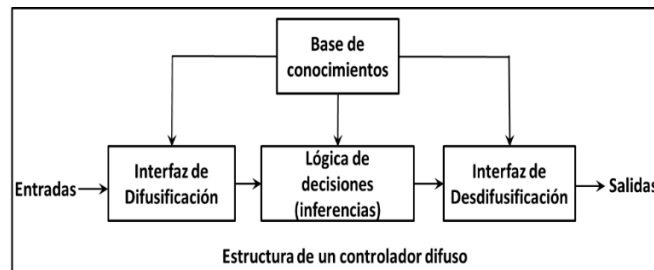


Fig. 1. Estructura de un controlador difuso.

interfaz de defusificación, base de conocimiento, lógica de decisiones e interfaz de desdifusificación (Fig. 1).

2. Trabajos relacionado

En 2007 se describe la navegación de robots móviles mediante comportamientos utilizando lógica difusa. Para la implementación se utilizó el toolbox de lógica difusa de Matlab y el modelo cinemático de la plataforma P-METIN del Grupo GIROPS [17]. En 2009 se presenta el diseño e implementación de un sistema de visión por computadora y control difuso orientado a la navegación autónoma de un robot móvil Lego NXT con capacidades de localización y mapeo simultáneo en un ambiente controlado mediante un algoritmo de navegación con atareas como la búsqueda de la ruta más cercana mediante el algoritmo de Dijkstra, la detección y evasión de los obstáculos presentes en su entorno y la alineación en el desplazamiento de la ruta basada en visión y lógica difusa [18]. Otro trabajo que hace referencia al navegador difuso de un robot móvil, utiliza lógica difusa para controlar el avance del robot hacia una ubicación específica con obstáculos [19]. Un trabajo más es una plataforma robótica móvil autónoma con ruedas mecanum para estudios en ambientes de carga, se analizan los posibles movimientos y se eligen los que van a ser estandarizados dentro del sistema. Se utilizan ecuaciones cinemáticas y de lógica difusa a bajo nivel para controlar la velocidad de los motores de las ruedas [10].

Otros trabajos que combina la lógica difusa con la Inteligencia Artificial son: La aplicación de técnicas de inteligencia artificial para la navegación de robots móviles utilizando el simulador player/stage [14] y la localización basada en lógica difusa y filtros de Kalman para robots con patas [15]. De la misma forma en la inteligencia artificial se han realizado trabajos de investigación utilizando los algoritmos genéticos, como se puede observar en el proyecto de Lozano Márquez Aplicación de técnicas basadas en lógica difusa para la mejora del comportamiento de los algoritmos genéticos con codificación real [16].

3. Metodología

Para esta investigación se estudiaron las variables físicas del Robot, como el frenado del robot, la forma de frenado, la distancia y velocidad en el paro del móvil y la forma de actuar de sus sensores, en la siguiente etapa se desarrolló un simulador con el objetivo de implementar los algoritmos en amplitud y primero el mejor, en este código se analizaron ambos algoritmos, se formuló una poda de los nodos que no se pueden explorar por el robot; posteriormente se plantea un par de variables independientes a las cuales se les dio el nombre de distancia y velocidad que se utilizaron para formular un par de conjuntos difusos, que se estudiaron para el desarrollo del controlador difuso, lo anterior fue para lograr obtener una variable dependiente que se le llamó *Frenar* que se utiliza para lograr que el móvil reduzca su velocidad, en la última etapa de este trabajo de investigación se utilizaron los resultados del control difuso para formular un

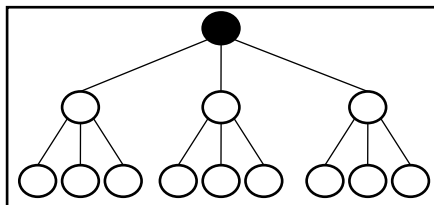


Fig. 2. Árbol con tres hijos.



Fig. 3. Interfaz de simulador.

nuevo algoritmo que ayudará en el mapeo de una región y que es un aporte para el SLAM.

4. Desarrollo

Para el desarrollo de este trabajo se utilizó el modelo del frenado del robot Scribbler® obtenido en trabajos previos [7]. Posteriormente, tomando como base las características físicas del Robot, él cuenta con tres sensores de luz colocados de la siguiente forma, uno colocado en la parte izquierda, parte central y el último en la parte derecha del móvil [8], por lo cual la estructura de datos llamada árbol 3-ario representada en Fig. 2 es la adecuada para representar sus sensores de luz con los que cuenta el robot móvil. La raíz del árbol es el punto de partida donde se colocará el móvil; para el desarrollo de los algoritmos fue necesario se implementara una estructura de datos llamada pila, en donde se guardarán los nodos visitados.

El proyecto se implementa en java por ser un lenguaje de alto nivel, (se puede utilizar cualquier lenguaje para desarrollar el simulador) para desarrollar un simulador cuya interfaz gráfica se ha representado en la Fig. 3. En el simulador se implementó la búsqueda en amplitud, Fig. 4, y la búsqueda el primero el mejor, Fig. 5.

Se realizó un estudio de los algoritmos de búsqueda sin información e información, con el objetivo de mejorar sus resultados. Se puede observar en la Fig. 3 que existen nodos que se encuentran muy cerca de los objetos del mapa, los cuales al realizar un mapeo crean colisiones como se pueden estudiar en la Fig. 6. Enseguida se formula la solución de esta problemática.

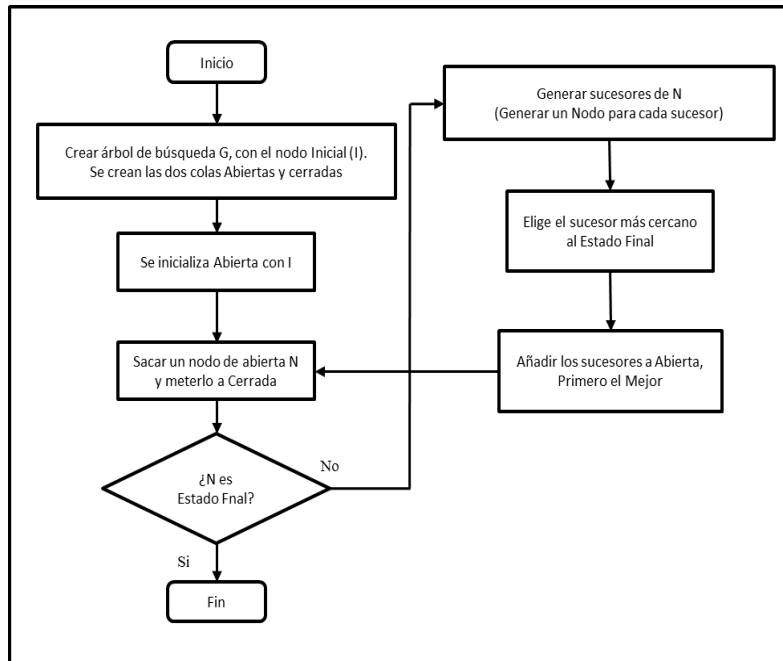


Fig. 4. Búsqueda por Amplitud.

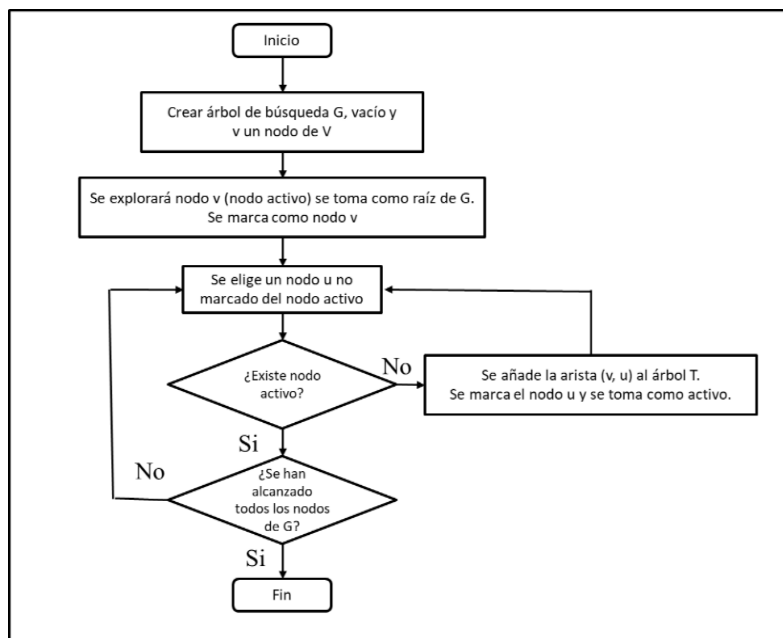


Fig. 5. Búsqueda el primero el mejor.



Fig. 6. Simulador con amplitud.

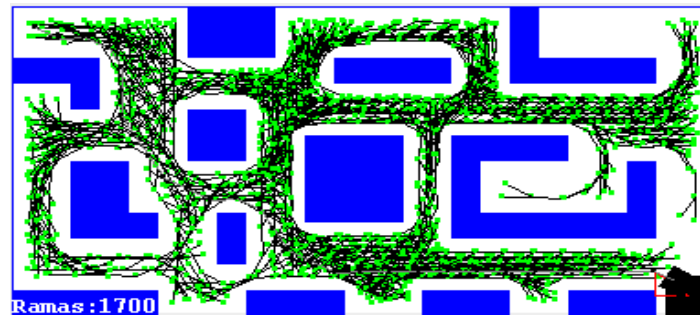


Fig. 7. Árbol de búsqueda con poda.

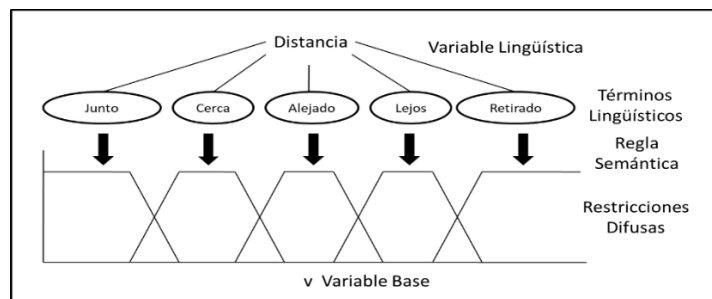


Fig. 8. Variable Lingüística distancia.

Como se puede observar en la Fig. 6 existen nodos que el árbol de búsqueda ha generado, los cuales se encuentran próximos a los objetos del mapa, han rodeado el perímetro de los objetos del mapa, para solucionar este problema se propone realizar una poda de los nodos, para lograr dicho proceso se formuló una función de membrecía triangular (Ecuación 2) cuyo objetivo es determinar los nodos que el robot puede explorar sin tener colisiones.

La Ecuación 2 es una función triangular, la cual tiene 4 casos, el primero $x < a$, la ecuación esta en $0 \mu_A(x)$, la segunda etapa es $\frac{x-a}{b-a} a \leq x < b \mu_A(x)$ aqui se presenta la

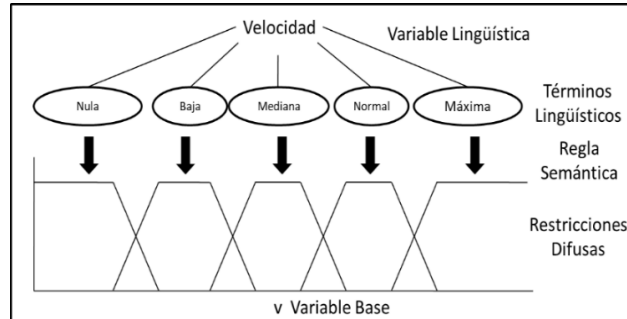


Fig. 9. Variable lingüística velocidad.

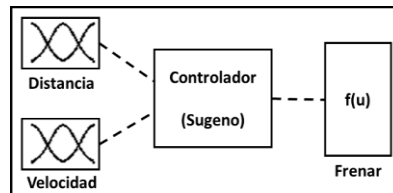


Fig. 10. Controlador difuso.

Tabla 1. Variables distancia, velocidad vs frenar.

		Distancia				
		Junto	Cerca	Alejado	Lejos	Retirado
Velocidad	Nula	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Baja	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Media	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Alta	F(n)	F(m-)	F(m)	F(m+)	F(r)
	Máxima	F(n)	F(m-)	F(m)	F(m+)	F(r)

pendiente de la función, la tercera etapa $\frac{c-x}{c-b} \mathbf{b} \leq \mathbf{x} \leq \mathbf{c} \mu_A(\mathbf{x})$ es la pendiente que decrece de la función y la cuarta parte es cuando $\mu_A(\mathbf{x})$ es *cero*.

$$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c < x \end{cases} \quad (2)$$

Al ser implementada la poda de los nodos, se puede observar en el simulador un espacio entre el árbol de búsqueda y los objetos del mapa como se observa en la Figura 7.

Para la implementación de la Lógica Difusa se realizó el procedimiento para obtener un par de conjuntos difusos, utilizando las variables independientes de velocidad y distancia, las cuales se utilizaron para ser transformadas en variables lingüísticas, las cuales son representadas en las Figuras 8 y 9.

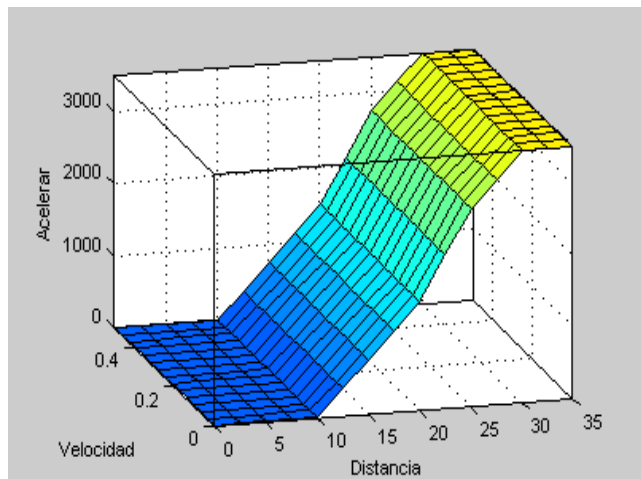


Fig. 11. Curva de control.

Tabla 2. Reglas de control.

1. If (Distancia is Junto) and (Velocidad is Nula)	then (Frenar is Nulo) (1)
2. If (Distancia is Junto) and (Velocidad is Baja)	then (Frenar is Nulo) (1)
3. If (Distancia is Junto) and (Velocidad is Media)	then (Frenar is Nulo) (1)
4. If (Distancia is Junto) and (Velocidad is Alta)	then (Frenar is Nulo) (1)
5. If (Distancia is Junto) and (Velocidad is Maxima)	then (Frenar is Nulo) (1)

Se diseñó el controlador difuso utilizando las variables de distancia y velocidad, al realizar este proceso se genera la variable dependiente *Frenar* (F), que es necesaria para realizar la exploración del móvil. (Fig. 10).

En la Tabla 1 se puede observar la interacción entre las variables distancia y velocidad con la variable *Frenar*, note las acciones de la última variable que son F(n): *Frenar* (nulo), F(m-): F(media-), F(m): F(media), F(m-): F(media-) y F(r): F(rápida).

Partiendo de la tabla anterior, se implementa el controlador difuso, describiendo las acciones de la variable *Distancia* con el valor de nula y los valores en la variable de *Velocidad* (nula, baja, media, alta, máxima). En la Tabla 2 se describen las acciones en la variable *Frenar*.

5. Resultados

En la Fig. 11 se muestra la respuesta del controlador difuso y la estabilidad del sistema, en la misma figura se puede analizar la interacción entre las variables independientes *Velocidad* y *Distancia*, con la variable dependiente *Frenar*. Al realizar el proceso del control difuso se logró el objetivo de este trabajo.

5.1 Algoritmo difuso

Para lograr el diseño del control difuso se formuló el nuevo algoritmo utilizando las reglas de control. A continuación, se describen las acciones de la variable Velocidad con el valor nulo y la variable Distancia.

Algoritmo

0. Inicio

1. Detectar distancia de los tres sensores.

I. Si central es igual a 0 rotar el móvil e ir a 3)

- | | | | |
|----------------------------------|---|-----------------------------|---|
| 2. a) Si velocidad = nula | y | distancia = junto | entonces frenar = nulo. |
| b) Si velocidad = nula | y | distancia = cerca | entonces frenar = mediana menos. |
| c) Si velocidad = nula | y | distancia = alejado | entonces frenar = mediana. |
| d) Si velocidad = nula | y | distancia = lejos | entonces frenar = mediana más. |
| e) Si velocidad = nula | y | distancia = retirado | entonces frenar = rápido. |
| f) Si velocidad = baja | y | distancia = junto | entonces frenar = nulo. |
| g) Si velocidad = baja | y | distancia = cerca | entonces frenar = mediana menos. |
| h) Si velocidad = baja | y | distancia = alejado | entonces frenar = mediana. |
| i) Si velocidad = baja | y | distancia = lejos | entonces frenar = mediana más. |
| j) Si velocidad = baja | y | distancia = retirado | entonces frenar = rápido. |
| k) Si velocidad = media | y | distancia = junto | entonces frenar = nulo. |
| l) Si velocidad = media | y | distancia = cerca | entonces frenar = mediana menos. |
| m) Si velocidad = media | y | distancia = alejado | entonces frenar = mediana. |
| n) Si velocidad = media | y | distancia = lejos | entonces frenar = mediana más. |
| o) Si velocidad = media | y | distancia = retirado | entonces frenar = rápido. |
| p) Si velocidad = alta | y | distancia = junto | entonces frenar = nulo. |
| q) Si velocidad = alta | y | distancia = cerca | entonces frenar = mediana menos. |
| r) Si velocidad = alta | y | distancia = alejado | entonces frenar = mediana. |
| s) Si velocidad = alta | y | distancia = lejos | entonces frenar = mediana más. |
| t) Si velocidad = alta | y | distancia = retirado | entonces frenar = rápido. |
| u) Si velocidad = máxima | y | distancia = junto | entonces frenar = nulo. |
| v) Si velocidad = máxima | y | distancia = cerca | entonces frenar = mediana menos. |
| w) Si velocidad = máxima | y | distancia = alejado | entonces frenar = mediana. |
| x) Si velocidad = máxima | y | distancia = lejos | entonces frenar = mediana más. |
| y) Si velocidad = máxima | y | distancia = retirado | entonces frenar = rápido. |

3. Si no ha llegado a la meta ir a 1).

4. Salir.

6. Conclusiones

Resumamos, se estableció un par de conjuntos difusos con las variables independientes *distancia* y *velocidad*, al utilizarlas en el controlador difuso se logró obtener una variable dependiente *frenar*, además, se utilizaron las reglas del controlador difuso para formular un nuevo algoritmo. Resultando que la lógica difusa influye de manera importante para lograr la exploración de un mapa, ya que el controlador difuso es más rápido con respecto al largo proceso matemático del control tradicional.

Referencias

1. McCarthy, J.: Padre de la Inteligencia Artificial. <https://www.bbvaopenmind.com/tecnologia/inteligencia-artificial> (1956)
2. Rifkin, J.: *The Third Industrial Revolution*. New York: Palgrave Macmillan (2011)
3. Pérez, G.G. : Percepción y valores: Conceptos fecundos e incomprensidos. Ciudad de México: Alternativas cubanas en psicología (2013)
4. Asker-Zadeh, L.A.: Fuzzy sets. *Information and Control*. pp. 338–353 (1965)
5. Morales-Jiménez, F.J.: Reporte de Robot Scribbler® (2012)
6. Robot Scribbler®. Parallax Inc. <https://www.parallax.com/product/28333> (2019)
7. Baturone, A.O.: Robótica manipuladores y robots móviles. Barcelona: Marcombo, S.A. (2001)
8. Moravec, H.P.: Robots and Intelligence. <https://frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1983/robage.mss> (1983)
9. John-Bares, W.: cyberneticzoo.com. <http://cyberneticzoo.com/walking-machines/1988-91-ambler-john-bares-william> (1988)
10. Durrant-Whyte, H., Rye, D., Nebot, E.: Localization of automatic guided vehicles. In: *The 7th International Symposium (ISRR'95)* (1995)
11. Corke, P.I., Trevelyan, J.: *DBLP computer science bibliography*. Sidney, Australia (1999)
12. Pons, J.V.: Localización y generación de mapas del entorno (SLAM) de un robot por medio de una Kinect. Valencia, España: Universidad Politécnica de Valencia (2012)
13. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), pp. 116–132 (1985)
14. Marinelli, M., Kuna, H., Puente, F., Mounier, M., Kornuta, C., De Silvestre, E., Kornuta, C., Kolb, M.: Aplicación de técnicas de inteligencia artificial para la navegación de robots móviles utilizando el simulador player/stage. Universidad Nacional de Misiones Felix de Azara (2013)
15. Martín, F., Matellán, V., Barrera, P., Cañas, J.M.: Localización basada en lógica difusa y filtros de Kalman para robots con patas. Madrid España (2006)
16. Lozano, M.: Aplicación de técnicas basadas en lógica difusa para la mejora del comportamiento de los algoritmos genéticos con codificación real. Universidad de Granada (1997)
17. Parra, H., Ríos, L.: Navegación de Robots móviles mediante comportamientos utilizando lógica difusa. *Scientia et Technica*, Año XIII, 34, Universidad Tecnológica de Pereira, pp. 79–84 (2007)
18. Ibarra, M.: Navegación autónoma de un robot con técnicas de localización y ruteo. INAOE (2009)
19. Caballero, M., Cruz, L.: Navegador Difuso de un Robot Movil. <http://www.mecamex.net/anterior/cong02/papers/art29.pdf> (2019)
20. Costella, V., Rodríguez, H.: Sistema de control con lógica difusa y diseño de plataforma autónoma con ruedas mecanum a partir de su arquitectura. *RIDTEC*, 9(2), pp. 30–38 (2013)

Handcraft and Automatic Approaches for the Recognition of Leukemia Images

Rocio Ochoa-Montiel^{1,2}, Leonardo Martínez Ibarra³, Humberto Sossa^{1,4},
Gustavo Olague⁵

¹Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

²Universidad Autónoma de Tlaxcala,
Facultad de Ciencias Básicas, Ingeniería y Tecnología.
Mexico

³Instituto Politécnico Nacional,
Unidad Interdisciplinaria en Ingeniería y Tecnologías Avanzadas,
Mexico

⁴Tecnológico de Monterrey,
Escuela de Ingeniería y Ciencias.
Mexico

⁵CICESE Research Center,
EvoVision Laboratory Ensenada,
Mexico

{ma.rocio.ochoa,humbertosossa,gustavo.olague}@gmail.com,
leonardo.mtz96@outlook.es

Abstract. In this paper, we analyze how handcraft and automatic approaches frequently can be used in the recognition of leukemia images. The visual recognition of subtypes of leukemia cells is a difficult problem in medical area. For this reason, the visual analysis is a feasible option when advanced techniques for diagnosis are not available. Additionally, the problem of determining a logic sequence of the process of recognition is equal or more important that the solution itself, which implies to know the advantages and drawbacks of the available approaches to solve a particular problem of visual recognition. Thus, in this paper we present two approaches that allow us become aware of the use of particular techniques in problems such as leukemia cell recognition.

Keywords: Machine Vision, Image Recognition, Leukemia Cells.

1 Introduction

Leukemia is type of malignant illness that represents one of the main causes of dead in people less than 15 years. Studies suggest that the incidence of a specific leukemia type named Acute Lymphoblastic (AL) in countries like Mexico is among the highest in the world [1]. Although advanced methods exists for detection of this illness, these are very expensive and inaccessible for the most of affected people. Typically, the centers where these methods are available are located in big cities, which increases the cost and time of detection and treatment. For these reasons is a common practice to carry out a visual analysis of blood smears to detect some abnormalities in blood cells, like in leukemia.

On the other hand, an important problem in blood cells recognition is the inherent subjectivity at visual identification of abnormal cells. This is due to factors such as the lack of experience of clinical personal, variability in the smears preparation, or the physical conditions of people who realize the analysis, to mention a few. From the above discussion, it is worth mentioning that a method is needed for doing an adequate recognition of blood cells which minimizes the differences in the observations from different medical specialists and allow us to obtain more accurate diagnosis.

In this regard, the problem of blood cell recognition has been addressed using handcraft approaches [2, 3, 4, 5], where standard techniques of image processing and machine learning are used. Particularly, these techniques use color spaces like CIELab and YUV, morphologic operators and classifiers as k-means and SVM or variations of them. The same problem can be treated with last generation artificial neural networks (ANNs) to automatically extract features and perform classification [6]. These models do not need a previous image processing or explicit feature extraction. Instead they use the image to find adequate descriptors for image classification. Although these models reach adequate results, the process to obtain the features and their meaning is hidden to the user, whereby they are considered black box models. Moreover, they require a lot of computational resources and a great amount of images with medium or low resolution.

1.1 Research Contribution

An important contribution of this work is the comparison between handcraft and automatic approaches to address the problem of recognition of leukemia images.

Since in the problem of leukemia cell recognition it is of utmost importance to know how the features are derived in a natural way, as well as their meaning and their importance for the recognition task, the analysis shown in this paper is useful to depict the advantages and drawbacks of handcraft and automatic approaches to solve the recognition particular problem of this paper.

The remainder of this paper is organized as follows. Section 2 describes the materials and the methods used. Section 3 presents the experiments and the results obtained through this research. Finally, Section 4 provides the conclusions.

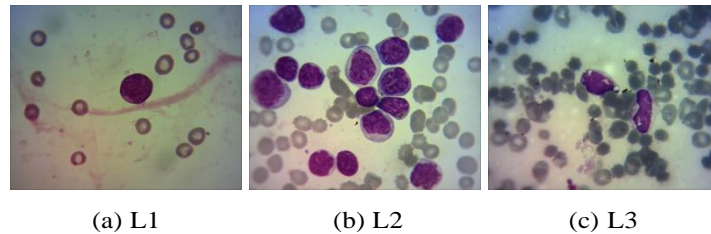


Fig. 1. Types of lymphocytic leukemia cells.

2 Materials and Methods

In visual analysis of blood white cells, an important criterion for pathology recognition in blood tissue is the cell nucleus, as in leukemia [7]. In blood smears, the nucleus appears as a prominent region of dark purple hues into the cell. Typically, the nuclear features are similar for a specific subtype of blood cell. Fig. 1 shows images of three subtypes of leukemia. Notice that the region of nucleus is particularly different for each one of these subtypes. From this, in this work the nucleus region is used particularly to assessment handcraft techniques for the recognition of leukemia images.

This work is divided into two parts, both carried out independently. A part addresses the use of handcraft approaches, the other part includes tests with automatic approaches. In the former, we use standard image processing techniques to obtain the region of nucleus from the input image. We then consider properties of shape and color from extracted region taking into account the criteria of medical area to recognize these subtypes of cells. Later, the obtained descriptors are feed to two standard classifiers for the recognition of leukemia cells. The second part of the work uses as automatic approaches the convolutional neural networks.

We used a dataset composed by 651 bone marrow smear image from three subtypes of leukemia: L1, L2 and L3, 217 images per class in which expert hematologist provide the classification. Images are in BMP format in color with a resolution of 1280×1024 pixels. Image acquisition was done by means a Luzeren XSZ-148S optical microscopic with a magnification of 1250 times and a camera coupled to the microscope with a resolution of 1.3 megapixels. For this work, images were resized to 256×320 pixels using bicubic interpolation. According the subtype of leukemia, the image contains one or more cells of interest as in Fig. 1.

2.1 Handcraft Approach

Visual properties of blood cells suggested from literature focused to hematological diseases are considered in this work [7]. Particularly, nucleus traits as shape, color and the distribution of some elements into it are meaningful for cell recognition. From this, the aim in this stage is to extract the nucleus of the bone marrow smear image to obtain adequate descriptors for the recognition of three subtypes of leukemia, L1, L2 and L3.

Selection of Nucleus. Since color is a prominent feature to identify the nucleus in images, we use saturation from the HSI color space [8] to obtain an image I_s , in which the nucleus is highlighted from the background. Later, a contrast enhancement using a

square function $I_s^2/255$ emphasizes the regions of nucleus preserving its appearance and fading some not useful areas in the image. To select this function, we tried other functions such as inverse, cubic, square root [9]; however, in the output image the region of interest was not properly highlighted.

Output image from the previous step is thresholded by means of Otsu's Method [10]. To remove the remainder noise two criteria are proposed, the area and the Hue (from HSI color space) of nucleus. The first criterion is suggested since the area size of nucleus is typically greater than 200 pixels, this value is obtained by experimentation. Hence, those regions with an area < 200 are removed from image.

It is worth noting that this value is applicable to images with a resolution of 256×320 pixels. Additionally, Hue is useful to discard those pixels that are not removed with the first criteria. Thus, a threshold T is proposed in a range $250 < T < 350$. These thresholds are obtained from the minimum and maximum values founded in the component Hue from nucleus in color images. As a result of applying the last process to each image in dataset, the output binary images contains the regions of the cell nucleus.

Features extraction. The selection of features is based on the morphologic traits given by the World Health Organization (WHO) Classification of Tumors and Lymphoid Tissues, for the leukemia classification [7]. According to this classification, important properties to identify types of leukemia cells are the size, color, shape and composition of nucleus. Thus, we propose two sets of features *ReducF* and *FullF* to measure these properties, see Tables 1 and 2.

To measure these kind of properties, function *regionsprops* from Matlab Toolbox of Image processing is used. In last phase of the handcraft approach, both sets of features are used to feed two classifiers commonly used for the recognition of the images: a Multi-layer Perceptron (MLP) and Random forest (RF) [12]. The experiments are independently carried out by each classifier.

2.2 Automatic Approach

We select two classical convolutional neural network to test the automatic approaches for the cell recognition. LeNet and AlexNet models realize feature extraction and classification automatically from images. In both models, the learning occurs by using the position of patterns directly from the input image data exclusively, without considering previous knowledge about the image. LeNet is a standard CNN [13], while AlexNet [14] use transfer learning. The input is a color image for both nets.

The LeNet architecture used is composed by four sets of convolutional and pooling layers, followed by a flattening convolutional layer, two fully-connected layers, and a softmax classifier. The fully connected softmax output layer produces three possible values that correspond to each class.

AlexNet is a pre-trained net that is eight layers deep and requires an image input size of 227 by 227 by 3 , where 3 is the number of color channels. Because the last three layers of the net are configured originally for 1000 classes, these layers are fined-tuned for our problem of classification, replacing them with a fully connected layer, a softmax layer, and a classification output layer.

Table 1. *ReducF* Features.

Number of Feature	Name	Description
1	Num. cells	Number of nucleus in image
2	Ratio of occupied area	Area Nucleolus / area Cell
3	Circularity	Ratio of circularity
4	Euler number	Euler number
5	Hue	Average of Hue
6 – 12	Hu's 7 invariants moments	Hu's moments [11]

Table 2. *FullF* Features.

Number of Feature	Name	Description
1 – 12	Features 1-12 from Table 1	-Number of pixels in nucleus
13	Area	Number of pixels in 'ConvexImage'
14	Convex area	Distance between the ellipse foci/major axis length
15	Eccentricity	Diameter of a circle with the same area as the region
16	EquivDiameter	Area of the bounding box.
17	Extent	Distance around the boundary of the region
18	Perimeter	$Area/ConvexArea$
19	Solidity	Average of S
20	Saturation (from HSI)	Average of M
21	M (fromCMYK)	

3 Experimental Results

Experiments were executed on a CPU from Intel Core i9- 7900X CPU 3.31Ghz, 64GB RAM, 222Gb hard drive, 64-bit Windows10 Enterprise Edition operating system, graphics processing unit (GPU) GeForceGTX 1080, and MATLAB R2018a.

To address the problem of recognition of leukemia images, first we use a handcraft approach. Thus, to obtain the region of nucleus in the images the process described in section 2.1 is applied.

Then, using the saturation I_s from the HSI color space and a contrast enhancement through a square function $I_s^2/255$, the regions of nucleus are emphasized and some regions not useful in the image are dimmed.

The remainder noise from last step is removed using a thresholding with Otsu's Method and a manual thresholding in the component H from HSI image. In this last, the range of hue H for the nucleus region is obtained considering that the minimum and maximum values found in the component Hue from nucleus in color images are in a range of $250 < T < 350$, hence these values are used for the manual thresholding. Fig. 2 shows the result of this processing for an image of leukemia L2.

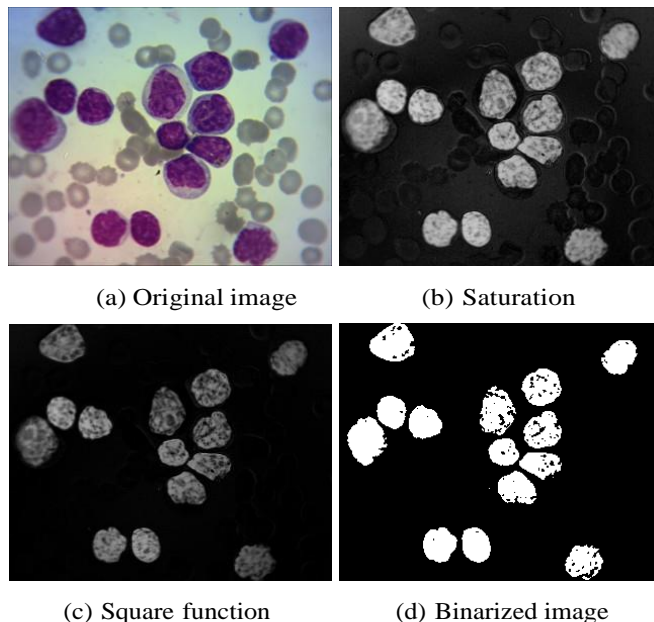


Fig. 2. First preprocessing sequence for an image of leukemia L2.

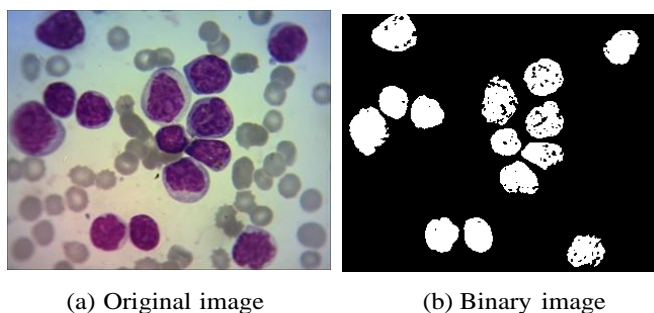


Fig. 3. Images with extracted nucleus region.

Finally, to eliminate the small regions that remain after thresholding, those regions whose area is less than 200 pixels are removed. This criterion is suggested because the area of nucleus is typically greater than 200 pixels, which was determined with a visual and statistical analysis of the images.

Considering that the images were resized to 256×320 pixels, the value of area size should be adjusted if images have a different resolution. As a result of applying last process to each image in dataset, the output binary images contain only the regions of the cell nucleus, see Fig. 3.

The binary image obtained in previous stage is used to extract the sets of features proposed in Tables 1 and 2. It is worth mentioning that features 1 and 2 from Table 1 are computed for the full image, while the remainder are obtained calculating the average from the corresponding feature for each region into the image. Most of the

Table 3. Results of classification accuracy from 30 executions using MLP

<i>Set ReducF</i>				<i>Set FullF</i>			
Statistics	Acc-Val	Acc-Test	Time (s)	Statistics	Acc-Val	Acc-Test	Time (s)
Without normalizing							
Min	95.35	92.82	0.44	Min	95.35	92.31	0.47
Max	99.22	97.44	2.68	Max	100.00	97.95	5.86
Std	1.11	1.03	0.40	Std	1.37	1.47	0.97
Avg	97.36	96.03	0.56	Avg	97.93	96.21	0.73
Normalization minmax							
Min	89.92	91.28	0.44	Min	93.80	93.33	0.47
Max	97.67	96.92	2.70	Max	100.00	97.95	2.99
Std	1.59	1.18	0.40	Std	1.35	0.92	0.45
Avg	95.92	95.61	0.56	Avg	98.24	96.46	0.63
Normalization Zscore							
Min	95.35	94.36	0.45	Min	95.35	94.36	0.50
Max	99.22	97.44	2.71	Max	100.00	98.46	2.84
Std	1.03	0.79	0.41	Std	1.09	0.81	0.42
Avg	97.49	96.09	0.58	Avg	98.27	96.82	0.63

features were obtained using function regions props from the Toolbox of Image processing of Matlab. In all experiments, we used 50% of data for training, 20% for validation and 30% for testing. To valid the classification is used k-FCV (k- Fold Cross Validation) with $k = 5$, and each experiment is carried out 30 times.

The two sets of features are used separately to compare the performance between both groups. Additionally, the experiments consider two types of normalization: minmax and Zscore, and two classifiers: MLP and Random forest. Thus, for each set of features we have three experiments with each classifier, this is using features without normalizing, features normalized with minmax, and features normalized with Zscore. Tables 3 and 4 show the results for these experiments, where is worth to note that normalization Zscore and a set greater of feature is useful in the most of cases. The time for all experiments is low and they do not require a great computing power.

On the other hand, convolutional neural networks LeNet and AlexNet are used as automatic approach. To prevent overfitting, we use image data augmentation including transformations of reflection XY. The input for the nets are the original images in RGB. Accuracy classification we use is the ratio of number of correct predictions to the total number of input examples. This manner, the average of accuracy for testing is 98.36% for LeNet, and 99.98% for AlexNet, however the time is considerably upper to that used by the handcraft approach. This is 188.47 secs. for LeNet, and 836.76 secs. for AlexNet.

Finally, it is important to mention that in the case of the handcraft approach, the visual identification of the leukemia cells could not be clear when the images are similar for the three classes of leukemia as illustrates the Fig. 4. This is because the considered

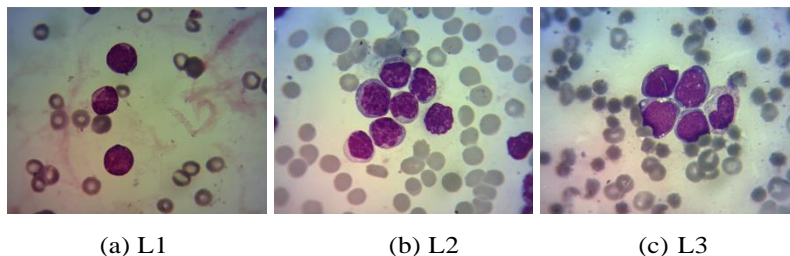


Fig. 4. Types of leukemia cells visually similar.

Table 4. Results of classification accuracy from 30 executions using Random forest

<i>Set ReducF</i>				<i>Set FullF</i>			
Statistics	Acc-Val	Acc-Test	Time (s)	Statistics	Acc-Val	Acc-Test	Time (s)
Without normalizing							
Min	97.67	94.36	5.89	Min	96.90	95.38	6.08
Max	98.45	97.44	8.52	Max	98.45	97.95	8.75
Std	0.38	0.69	0.59	Std	0.25	0.78	0.59
Avg	97.96	95.81	6.28	Avg	97.67	97.08	6.75
Normalization minmax							
Min	97.67	94.36	5.88	Min	96.9	93.85	6.11
Max	98.45	96.92	8.65	Max	99.22	98.46	8.80
Std	0.32	0.66	0.62	Std	0.40	1.01	0.62
Avg	97.83	95.78	6.25	Avg	97.62	96.77	6.42
Normalization Zscore							
Min	97.67	94.87	5.94	Min	96.90	95.90	6.10
Max	98.45	97.44	8.64	Max	98.45	98.46	8.83
Std	0.39	0.66	0.61	Std	0.52	0.61	0.60
Avg	98.06	96.19	6.29	Avg	97.70	97.04	6.47

features are related to the number, morphology and distribution of cells of interest into the image.

Hence, considering that the dataset contains very few cases of this type (approximately 1%), the handcraft approach is useful in part because the RNA learns the patterns of the features derived from the expert knowledge available in the medical area [7].

4 Conclusions

This work shows an analysis of the handcraft techniques and automatic techniques for the recognition of leukemia images. It is to note in experiments that although both techniques types offer adequate solutions to the problem of recognition of leukemia

images, handcraft techniques provide explainable knowledge of the way to select the useful features and the logical process to solve the problem.

Instead, using the automatic techniques is not possible to know the way to derive the salient features for addressing the problem of recognition of leukemia images.

In the handcraft techniques, the canonical methods of image processing generated the descriptors that feed to classifiers MLP and random forest for the recognition of cells. This implies that the feature extraction is driven by the human reasoning for the task of recognition of leukemia cells.

On the other hand, results of automatic techniques are competitive, however the process to generate the predictions of model is hidden, which is a notable drawback of these models. Additionally, exist serious limitations regard the image size and the hardware resources required to prove these models.

In brief, the handcraft techniques can provide useful issues for the development of models more robust and explainable learning techniques.

In future work, we would like to use of expert knowledge and symbolic learning to address the problem of recognition of leukemia images.

Acknowledgements. This research was supported in part by the Instituto Politécnico Nacional, Mexico under project 20200630. Rocio Ochoa thanks the Autonomous University of Tlaxcala, Mexico for the support. Leonardo Martínez thanks the Delfín program for the economic support to undertake his research stay. All authors thank the Applied Computational Intelligence Network (RedICA).

Referencias

1. Colunga-Pedraza, P., Gomez-Cruz, G., Colunga-Pedraza, J., Ruiz-Argüelles, G.: Geographic hematology: Some observations in Mexico. *Acta Haematológica*, 140, pp. 114–120 (2018)
2. Acharya, V., Kumar, P.: Detection of acute lymphoblastic leukemia using image segmentation and data mining algorithms. *Medical and Biological Engineering and Computing*, 57, pp. 1783–1811 (2019)
3. Rege, M.V., Gawali, B.W.: Performance evaluation of image segmentation process for recognition of leukemia. In S.S.A.J., eds.: *Information and Communication Technology for Intelligent Systems*. 107, Springer, pp. 499–509 (2018)
4. Uslu, F., Icoz, K., Tasdemir, K., Yilmaz, B.: Automated quantification of immuno-magnetic beads and leukemia cells from optical microscope images. *Biomedical Signal Processing and Control*, 49, pp. 473–482 (2019)
5. Negm, A.S., Hassan, O.A., Kandil, A.H.: A decision support system for acute leukaemia classification based on digital microscopic images. *Alexandria Engineering Journal*, 57(4), pp. 2319 – 2332 (2018)
6. Vogado, L.E.S., Veras, R.M.S., Araujo, F.H.D., Silva, R.R.V., Aires, K.R.T.: Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification. *Engineering Applications of Artificial Intelligence*, 72, pp. 415–422 (2018)
7. Rodak, B., Carr, J.: *Clinical Hematology Atlas*. Elsevier (2016)
8. González, R.C., Woods, R.E.: *Digital Image Processing*. Prentice Hall (2007)

9. De La Escalera, A.: *Visión por computador. Fundamentos y métodos* (2001)
10. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), pp. 62–66 (1979)
11. Hu, M.K.: Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8, pp.179–187 (1962)
12. Breiman, L.: Random forest. *Machine learning*. 45(1), pp. 5–32 (2001)
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp. 2278–2324 (1998)
14. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25 (2012)

Diseño de un órgano artificial con aplicación a la enfermería en el centro universitario UAEM Valle de Chalco

Maribel Hernández Arroyo, Marco Alberto Mendoza Pérez

Universidad Autónoma del Estado de México, Centro Universitario UAEM Valle de Chalco,
México

maribell1991hernan@gmail.com,
marco_alberto83@hotmail.com

Resumen. La importancia que tiene la tecnología en el mundo actual que trasciende a diferentes ramas de la ciencia, se enfoca en el desarrollo de prototipos o diseños de órganos artificiales en 3D o impresiones de estos. En este trabajo se analizó y diseñó el órgano riñón en 3D, para el aprendizaje de sus componentes, relacionado con temas de las Unidades de Aprendizaje de Anatomía y Fisiología, que cursan los estudiantes de la Licenciatura en Enfermería, en el Centro Universitario (CU) Valle de Chalco perteneciente a la Universidad Autónoma del Estado de México (UAEM). Se diseñó el modelo de un riñón en 3D, posteriormente se desarrolló una aplicación con realidad aumentada, que los estudiantes de segundo semestre utilizaron y observaron, que es un apoyo en su proceso de aprendizaje, siendo una experiencia duradera y útil para estas Unidades de Aprendizaje, haciendo el aprendizaje más entretenido y significativo para su educación profesional, ya que es otro paso más para el desarrollo de este tipo de tecnología educativa. Además de lo anterior, otras ventajas son que el estudiante puede conocer y comprender el funcionamiento de cada una de las partes del riñón en 3D. La aplicación se presentó a un grupo de 20 estudiantes de segundo semestre de la Licenciatura en Enfermería. Al finalizar la demostración, llenaron una encuesta que arrojó resultados satisfactorios, los más relevantes fueron: El 95% le pareció creativa, novedosa, e interesante y al 100% le fue fácil aprender con este tipo de aplicaciones que sirven de apoyo en el proceso de enseñanza y aprendizaje.

Palabras clave: Aprendizaje Significativo, Órgano 3D, Realidad Aumentada, Riñón.

Design of an Artificial Organ with Application to Nursing at the UAEM Valle De Chalco University Center

Abstract. The importance of technology in the current world that transcends different branches of science, focuses on the development of prototypes or

designs of artificial organs in 3D or prints of these. In this work, the kidney organ was analyzed and designed in 3D, for the learning of its components, related to themes of the Anatomy and Physiology Learning Units, which are studied by students of the Bachelor of Nursing, at the University Center (CU) Valle de Chalco belonging to the Autonomous University of the State of Mexico (UAEM). The model of a kidney in 3D was designed, later an application with augmented reality was developed, which the second semester students used and observed, which is a support in their learning process, being a lasting and useful experience for these Learning Units , making learning more entertaining and meaningful for your professional education, as it is another step in the development of this type of educational technology. In addition to the above, other advantages are that the student can know and understand the operation of each of the parts of the kidney in 3D. The application was presented to a group of 20 second-semester students of the Nursing Degree. At the end of the demonstration, they filled out a survey that yielded satisfactory results, the most relevant were: 95% found it creative, novel, and interesting and 100% found it easy to learn with this type of applications that serve as support in the process of teaching and learning.

Keywords: Meaningful Learning, 3D Organ, Augmented Reality, Kidney.

1. Introducción

El aprendizaje es un proceso de habilidades y valores, que cada persona fortalece durante su vida cotidiana. A continuación, se presentan los conceptos relacionados a la nefrología, al modelo 3D, a la realidad aumentada y a la metodología de prototipos, que se utilizaron para diseñar y desarrollar la aplicación de realidad aumentada.

La Nefrología es el estudio de la anatomía y la función de los riñones [1]. Los riñones son órganos que se encargan de filtrar la sangre y producir la orina [2].

A continuación, se describen las partes del riñón [3 y 1]:

- La médula renal está compuesta entre 8 y 18 pirámides renales de forma cónica. Cada pirámide está dirigida hacia la corteza renal.
- La corteza renal es el arca de textura lisa que se extiende desde la cápsula hasta las bases de las pirámides renales y hacia los espacios entre estas. Esas porciones de la corteza renal que se extienden entre las pirámides renales se llaman columnas renales.
- Cálices menores y mayores, los conductos papilares drenan en estructuras en forma de copa.
- La sangre abandona el riñón a través de una única vena renal.
- La cápsula renal es una envoltura grasa del riñón constituida por tejido adiposo. Está envoltura de grasa perirrenal recubre la cápsula.
- Nefrona, es la unidad estructural y funcional del riñón.
- Arteria renal, proporciona la vascularización al riñón, a las glándulas suprarrenales y a los uréteres.
- Pelvis renal, es una estructura anatómica en forma de embudo que drena la orina del riñón al uréter.
- Uréter, conduce la orina desde los riñones a la vejiga.

La tecnología se basa fundamentalmente en conocimientos que se utilizan para lograr un objetivo y dar soluciones de un problema en específico. Es una terminología que abarca una variedad de aspectos que pueden ir desde la robótica hasta la medicina, etc.

Los modelos 3D describen un proceso que representa un objeto tridimensional, los cuales se relacionan con escenas y animaciones etc. Se utiliza un software especializado en la creación y modificación de modelos 3D. Existen aplicaciones que permiten al usuario diseñar modelos en 3D, por ejemplo, blender, sketchup, etc.

A continuación, se presenta una breve historia de los modelos 3D, se remonta a 1976, cuando se inventó la impresora de inyección de tinta. En 1984 se hicieron algunos avances sobre la inyección de tinta que se transformó a impresión de materiales. A lo largo de las últimas décadas que se fueron desarrollando a través de varias industrias. Charles Hull, más tarde, el cofundador de 3D Systems, inventa la estereolitografía, un proceso de impresión que permite que un objeto en 3D se cree a partir de datos digitales. Se utiliza la tecnología para crear un modelo 3D a partir de una imagen y permite que los usuarios prueben un diseño antes de que este invierta en la fabricación del modelo definitivo [4].

En el 2002, Los científicos diseñan un riñón 3D en miniatura completamente funcional y con la capacidad de filtrar sangre y producir orina diluida en un animal. El desarrollo llevó a la investigación en el Instituto de Wake Forest de Medicina Regenerativa el objetivo de imprimir los órganos y tejidos con tecnología de impresión 3D [4].

La realidad aumentada (RA) es un entorno físico que tiene la enorme capacidad de combinar el mundo real y virtual a través de un dispositivo móvil. A continuación, se presentan las herramientas que se utilizaron para diseñar los modelos en 3D y desarrollar la aplicación en RA:

- Sketchup es un software que desarrolla modelos en 3D y es ideal para crear escenarios, objetos, etc. Tiene herramientas que podemos utilizar para la creación de cualquier tipo de proyecto que son animaciones o texturas.
- Vuforia es un SDK que se basa en el desarrollo de aplicaciones móviles y utiliza tecnología que tiene una gran capacidad de reconocer imágenes y objetos 3D orientados a la realidad aumentada.
- Unity 3d es una plataforma para el desarrollo de videojuegos y apps.

Se desarrolló la aplicación utilizando como base la realidad aumentada, sobre los diseños de los modelos 3D del riñón, para que los estudiantes de segundo semestre de la Licenciatura en Enfermería puedan visualizarlos de manera amigable e interactiva. Esta aplicación, se enfoca como apoyo a su aprendizaje o enseñanza de estos contenidos, para que los estudiantes o docentes deseen ampliar o reforzar sus conocimientos.

La aplicación con realidad aumentada es innovadora, creativa y útil para los estudiantes de la Carrera de Enfermería del Centro Universitario UAEM Valle de Chalco, además de que su uso es sencillo, cómodo y fácil de aprender por estos, que anteriormente no contaban con una herramienta tecnológica de este tipo.

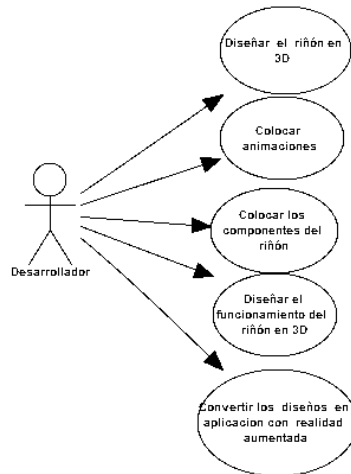


Fig. 1. Diagrama de caso de uso del desarrolló de la aplicación.

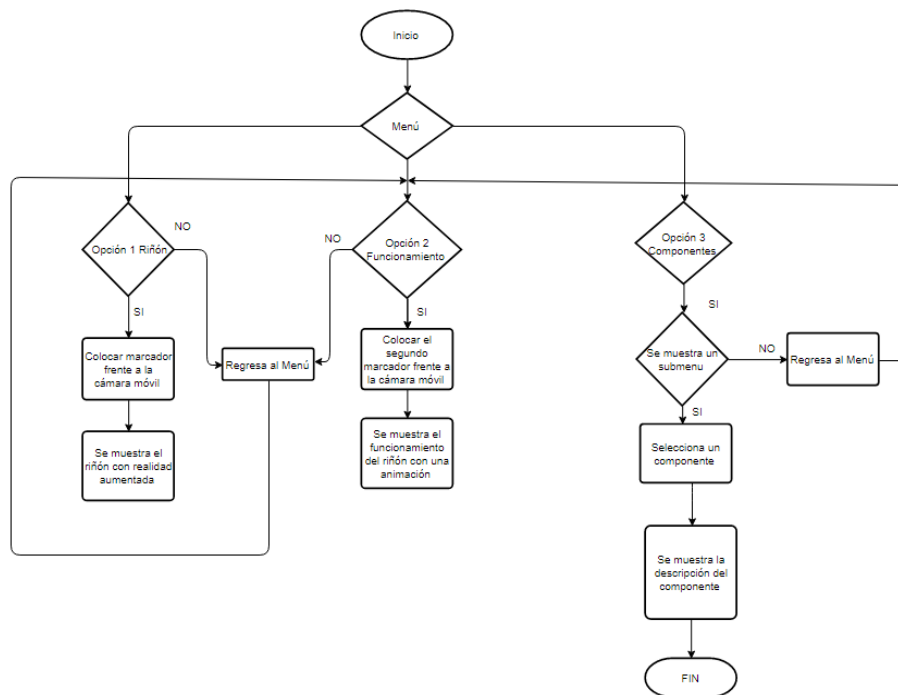


Fig. 2. Diagrama de flujo de la aplicación.

Lo anterior busca que el estudiante pueda visualizar y comprender como funciona el riñón, así como cada uno de sus componentes junto con su descripción. Otro de los

beneficios de desarrollar la aplicación con realidad aumentada, es apoyar a los estudiantes con el uso de la tecnología educativa, porque hoy en día, la tecnología es un factor importante que sirve para combinar conocimientos multidisciplinarios, como es la anatomía del riñón, así como el desarrollo de su aplicación para la enseñanza y aprendizaje de este tipo de contenidos. Este es otro paso más para el desarrollo de este tipo de aplicaciones que involucran la enseñanza de diferentes tipos de disciplinas.

La metodología utilizada para el diseño y desarrollo de la aplicación es la de prototipos, sirve para identificar requerimientos que surgen durante el desarrollo de la aplicación, algunos prototipos se construyen para ser desechables, otros son evolutivos; es decir, poco a poco se transforman en el sistema real [5]. Además, el prototipo se debe realizar en poco tiempo, una vez construido el prototipo se puede poner inmediatamente a prueba, por lo cual se deben utilizar los programas adecuados [6]. Sus etapas se describen a continuación [7]:

- Recolección y refinamiento de requisitos: Se recopilaron datos que se identificaron, para el desarrollo del software, son las especificaciones que requiere el usuario para el funcionamiento del sistema.
- Diseño rápido: una vez que se tiene las especificaciones del software se pasa a la implementación del software.
- Construcción del prototipo: se desarrolló el proyecto y se observa si cumple con las necesidades del usuario.
- Despliegue, entrega y retroalimentación: se muestra la aplicación al usuario para que valide, si le es útil la aplicación.

2. Método

El diseño y desarrollo de una aplicación con realidad aumentada para el aprendizaje y la enseñanza de los elementos del riñón y su funcionamiento como apoyo en las clases de anatomía y fisiología que cursan los estudiantes de la Licenciatura en Enfermería, está basado en la metodología de prototipos [5]. A continuación, se muestran sus etapas finalizadas:

2.1. Recolección y refinamiento de requisitos

Desde el comienzo se busca crear el diseño de un riñón en 3D y su funcionamiento, transformando los modelos en una aplicación con realidad aumentada, para el aprendizaje y la enseñanza de sus componentes para los estudiantes de la carrera de enfermería. Con esta información se describen los casos de uso y el funcionamiento de la aplicación.

El diagrama de caso de uso que se elaboró, para tener una representación clara y concisa de la identificación de los requisitos del desarrollo de la aplicación, sirve para facilitar lo que el desarrollador realizará (ver Figura 1).

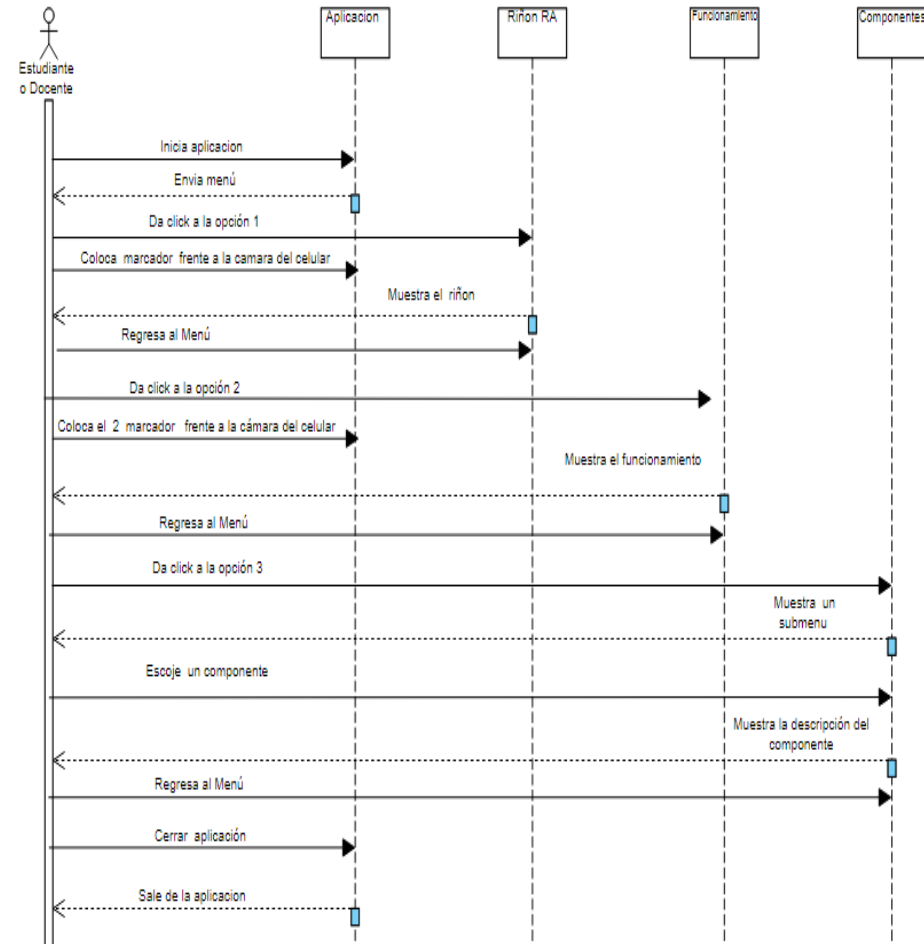


Fig. 3. Diagrama de secuencia.

2.2. Diseño rápido

En esta etapa se describe como se realizó el diagrama de flujo, el diagrama de secuencia y la creación de los modelos en 3D, cómo se observan en las Figuras 2 y 3.

El diagrama de flujo de la Figura 2, se muestra cómo se describe el proceso que realiza la aplicación a través de una serie de procesos entre estudiantes y docentes.

En la Figura 3, se describe el ingreso del estudiante y docente a la aplicación por medio del diagrama de secuencia, se observa a detalle la descripción del comportamiento de la aplicación e interacción con el usuario.

En la Figura 4, se muestra el modelo del riñón en 3D, utilizando el software sketchup. Para la creación del segundo modelo se basa en el funcionamiento del riñón.

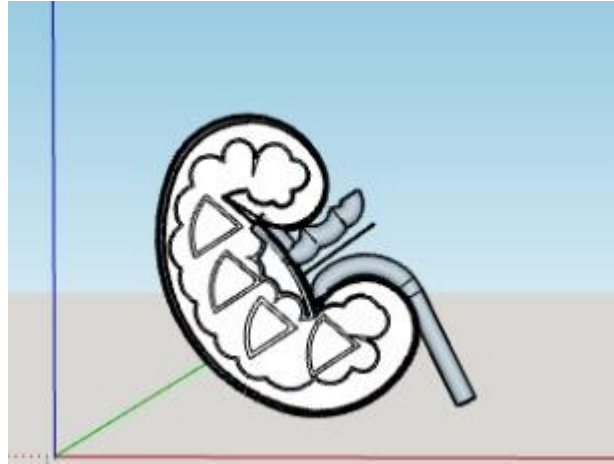


Fig. 4. Riñón 3D.

```
1 using System.Collections;
2 using UnityEngine.SceneManagement;
3 using UnityEngine;
4
5 public class Segundo1 : MonoBehaviour {
6
7
8     public void CambiarEscenas(string riñon)
9     {
10         SceneManager.LoadScene(riñon);
11     }
12     public void CambiarEscenas1(string principal)
13     {
14         SceneManager.LoadScene(principal); // regresa a la escena
15     }
16
17     public void CambiarEscenas2(string Corteza)
18     {
19         SceneManager.LoadScene(Corteza);
20     }
21     public void CambiarEscenas3(string riñon)
22     {
23         SceneManager.LoadScene(riñon); // regresa a la escena
24     }
25     public void CambiarEscenas4(string Cáliz)
26     {
27         SceneManager.LoadScene(Cáliz);
28     }
29
30     public void CambiarEscenas5(string riñon)
31     {
32     }
33 }
```

Salida
Mostrar salida de: Compilación
1> ---- Operación Compilar iniciada: proyecto: Riñon, configuración: Debug Any CPU ----
1> Riñon -> C:\Users\MARIBEL\Desktop\GVPROYECTO\Riñon\TempUnityVS_bin\Debug\Assembly-CSharp.dll
===== Compilar: 1 correctos, 0 incorrectos, 0 actualizados, 0 omitidos =====

Fig. 5. Código fuente.

2.3. Construcción del prototipo

La construcción de la aplicación móvil muestra la configuración del programa unity como la importación de los modelos 3D, se diseñó la interfaz de la aplicación

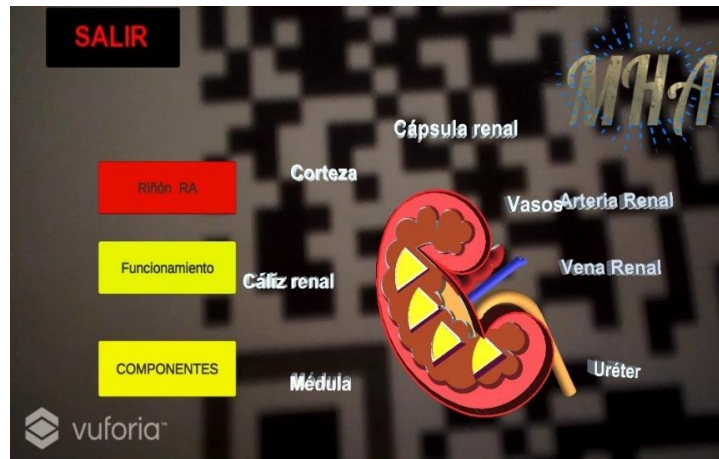


Fig. 6. Riñón con realidad aumentada.

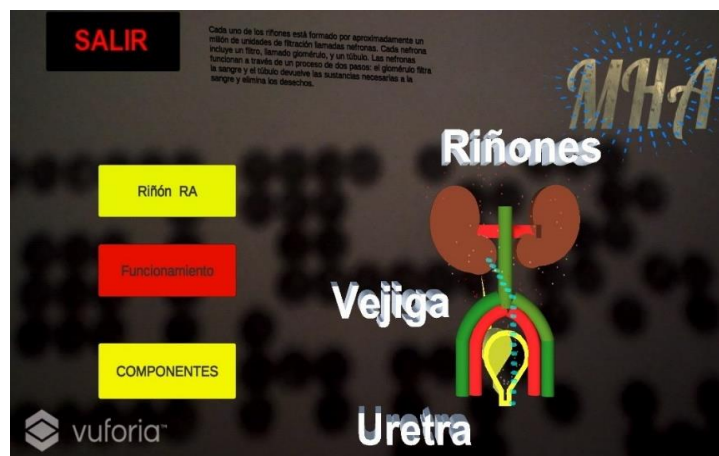


Fig. 7. Funcionamiento del riñón con realidad aumentada.

convirtiendo los modelos en realidad aumentada, como la descripción de los componentes del riñón. Se diseñó la interfaz del menú.

De tal forma que el estudiante o docente al pulsar click en las opciones del riñón o funcionamiento, el botón cambiara a color rojo y se visualizara la realidad aumentada. Para que realice esta función el estudiante pondrá el QR de acuerdo al modelo 3D que quiera visualizar, frente a la cámara móvil. La tercera opción llamada components, nos direcciona a un submenú.

En la Figura 5, se muestra el código, donde se crearon las escenas utilizando la librería SceneManager, se programaron los botones de regresar y de salir, para la creación de la aplicación.

En la Figura 6, se muestra el funcionamiento de la aplicación, mostrando los componentes del riñón con realidad aumentada. En la Figura 7, se muestra el



Fig. 8. Explicación de la aplicación con realidad aumentada.



Fig. 9. Utilización de la aplicación.

funcionamiento de la aplicación, el estudiante visualizara con realidad aumentada, el funcionamiento del riñón por medio de una animación.



Fig. 10. Resultados de la encuesta.

3. Resultados

3.1. Despliegue, entrega y retroalimentación

Se presentó la aplicación con realidad aumentada para la enseñanza del riñón a 20 estudiantes que forman parte de la Licenciatura en Enfermería, perteneciente al Centro Universitario UAEM Valle de Chalco. Posteriormente se les aplicó una encuesta.

En la Figura 8, se visualiza la explicación del funcionamiento junto con la utilización de la aplicación, donde los estudiantes utilizaron la aplicación para saber si fue de utilidad como apoyo a su aprendizaje. En la Figura 9, se observa la interacción del estudiante con la aplicación.

Una vez terminada la demostración de la aplicación, junto con la encuesta, se representaron las seis respuestas de los estudiantes de enfermería, por medio de tabulaciones y gráficas. Se obtuvieron los siguientes resultados: En la figura 10, se observa que para la primera pregunta, el 95% de los estudiantes les agrado la aplicación y solo un 5% no fue de su agrado.

Para la segunda pregunta, el 95% de los estudiantes les pareció interesante y a un 5% les fue indiferente la aplicación. Para la tercera pregunta, el 100% de los estudiantes les fue fácil utilizar la aplicación.

Para la cuarta pregunta, se observa que el 70% de los estudiantes no ha utilizado este tipo de aplicaciones con realidad aumentada y un 30% si ha utilizado una aplicación con realidad aumentada. Para la quinta pregunta, el 95% de los estudiantes les pareció creativa y novedosa la aplicación. Y para la sexta pregunta, se observa que el 100 % de los estudiantes les fue fácil aprender este tipo de contenido con realidad aumentada.

La aplicación cuenta con una buena aceptación por parte de los estudiantes de enfermería, al calificarla como creativa, funcional, novedosa e interesante. Esta

aplicación sirve de apoyo a los estudiantes y docentes en su proceso de enseñanza y aprendizaje de este tipo de contenidos.

4. Conclusiones

Para el análisis y diseño de los modelos 3D, y posteriormente la construcción de la aplicación con realidad aumentada, basada en la enseñanza y aprendizaje, se utilizó la metodología de desarrollo de prototipo de software. Esta aplicación es un recurso tecnológico educativo, que sirve de apoyo para el docente y el estudiante en la adquisición de conocimientos teóricos sobre el riñón y la comprensión del funcionamiento de este.

Esta aplicación le será de gran utilidad, tanto al docente como a los estudiantes en sus procesos de enseñanza y aprendizaje, les ayudará a conocer y a comprender el funcionamiento del riñón, así como la descripción de cada uno de sus componentes; permitiéndoles adquirir conocimientos en su educación profesional.

Este proyecto cumplió su objetivo propuesto y se logró mostrar la aplicación en el Centro Universitario UAEM Valle de Chalco. Al término de las pruebas, se aplicaron 20 encuestas a los estudiantes, llegando a la conclusión que la aplicación es creativa e innovadora, útil, interesante, fácil de utilizar y de aprender cualquier tipo de contenido; logrando que estudiantes y docentes la puedan utilizar como apoyo en su enseñanza y aprendizaje. Así mismo esta aplicación puede ser utilizada por los estudiantes de medicina, para ampliar o repasar los conocimientos ya adquiridos anteriormente. Se diseñó el órgano virtual en 3D para la enseñanza y aprendizaje de este tipo de temas, además se puede utilizar una impresora 3D, para que el estudiante observe de forma física el órgano, ya que las impresoras 3D son dispositivos capaces de reproducir cualquier tipo de modelo 3D.

Por último, se observó que los estudiantes reaccionaron con mucho interés y entusiasmo al interactuar con este tipo de tecnología, como lo es la realidad aumentada como apoyo en los procesos de enseñanza y aprendizaje de forma significativa, ya que relacionaron sus conocimientos anteriores con la utilización de la aplicación para comprender o retroalimentar el contenido ya existente. Los estudiantes comentaron que les gustaría utilizar esta clase de aplicaciones más seguido, porque mantienen su atención e interés visual en ubicar y conocer cada uno de los componentes del riñón junto con su funcionamiento.

Referencias

1. Diccionario de Medicina de Mosby. Barcelona, España: Océano. pp. 854–1319 (1995)
2. Guzmán-López, S., Elizondo-Omaña, R.: Anatomía humana en casos clínicos. Medica Panamericana, pp. 230–276 (2015)
3. Tortora, G., Derrickson, B.: Principios de anatomía y fisiología. Médica Panamericana, pp. 1022–1025 (2006)
4. Impresoras 3d: Breve historia de la impresión 3D. <https://www.impresoras3d.com/breve-historia-de-la-impresion-3d/> (2018)

Maribel Hernández Arroyo, Marco Alberto Mendoza Pérez

5. Pressman, R.: Ingeniería del software: Un enfoque práctico. Mc Graw Hill (2010)
6. EcuRed: Conocimiento con todos y para todos. Modelo de prototipos. https://www.ecured.cu/Modelo_de_Prototipos/ (2012)
7. Sommerville, I.: Ingeniería del Software. Pearson Educación (2005)
8. EcuRed: Conocimiento con todos y para todos. SketchUp. <https://www.ecured.cu/SketchUp/> (2017)
9. EcuRed: Conocimiento con todos y para todos. Realidad Aumentada. https://www.ecured.cu/Realidad_aumentada/ (2016)
10. Unity: Technologies, U. <https://unity.com/es> (2020)

Optimización de los coeficientes del modelo predictivo del número de casos diarios de coronavirus Covid-19 en México

Ana C. Zenteno, María del C. Santiago, Yeiny Romero, Judith Pérez,
Gustavo T. Rubín, Antonio E. Álvarez

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

{ana.zenteno, marycarmen.santiago, yeiny.romero,
judith.perez, gustavo.rubin }@correo.buap.mx,
antonio.alvarez@alumno.buap.mx

Resumen. El coronavirus covid-19 ha sorprendido a todos, millones de personas en todos los países y regiones han cambiado sus vidas con consecuencias inimaginables además de las lamentables muertes que ha cobrado. Al ser una pandemia muy grande, se requieren modelos matemáticos para simular escenarios y proporcionar a los tomadores de decisiones información más precisa basada en las variables de comportamiento actuales. En este trabajo partimos del análisis y procesamiento de los reportes oficiales diarios del número de casos positivos confirmados, cada uno se aproxima a un polinomio, se optimizan los coeficientes de éste y se retroalimenta el polinomio inicial. Con esta metodología se encontró un mecanismo para predecir el número de casos positivos hasta por 3 días con un error menor de 19.9%, con el fin de mejorar esta estimación se aplicará metodología para procesar el número de casos en estados y municipios, y contar con un modelo que brindará una predicción de casos hasta 2 semanas después del último reporte de casos y se actualizará diario incrementando la base de datos, disminuyendo el error.

Palabras clave: Covid-19, Modelo, Estadística.

Optimization of the Coefficients of the Predictive Model of the Number of Daily Cases of Coronavirus Covid-19 in Mexico

Abstract. The coronavirus covid-19 has surprised everyone, millions of people in all countries and regions have changed their lives with unimaginable consequences in addition to the unfortunate deaths it has claimed. Being a very large pandemic, mathematical models are required to simulate scenarios and provide decision makers with more accurate information based on current behavioral variables. In this work we start from the analysis and processing of the daily official reports of the number of confirmed positive cases, each one

approaches a polynomial, its coefficients are optimized and the initial polynomial is fed back. With this methodology, a mechanism was found to predict the number of positive cases for up to 3 days with an error of less than 19.9%, in order to improve this estimate, a methodology will be applied to process the number of cases in states and municipalities, and have a model that will provide a prediction of cases up to 2 weeks after the last case report and will be updated daily increasing the database, decreasing the error.

Keywords: Covid-19, Model, Statistics.

1. Introducción

El surgimiento de casos de virus en china en diciembre de 2019, particularmente en Wuhan (Hubei) y vinculado a un mercado mayorista de marisco, pescado y animales vivos ha propiciado una ola de investigaciones del desarrollo de la pandemia en todos los países. El 31 de diciembre de 2019, la Organización Mundial de la Salud (OMS) recibió reportes de varios casos de neumonía de etiología desconocida. A fecha de 3 de enero de 2020, las autoridades nacionales de China notificaron a la OMS que, en total, existían 44 pacientes con neumonía de etiología desconocida, de entre cuales, 11 pacientes estaban gravemente enfermos, mientras que los 33 pacientes restantes se encontraban en situación estable.

Según informaciones difundidas en los medios de comunicación, el mercado implicado en Wuhan se cerró el 1 de enero de 2020 por saneamiento y desinfección ambiental [1]. El día 8 de enero se Tailandia detectó un primer caso (fuera de China), siendo el 10 de enero el día que se presenta el primer fallecimiento causado por el virus. El incremento de los casos que aparecen en China y en otros países pone en evidencia la gravedad de la situación y la OMS el 10 de enero publica orientaciones técnicas y recomendaciones para todos los países sobre el modo de detectar casos, realizar pruebas de laboratorio y gestionar los posibles casos. Para el 30 de enero la OMS señala la existencia de un total de 7818 casos confirmados en todo el mundo, la mayoría de ellos en China y 82 en otros 18 países. La OMS evalúa el riesgo en China como muy alto y el riesgo mundial como alto [2].

Diferentes esfuerzos se están llevando a cabo en la intención de modelar el número de casos sospechosos, confirmados, decesos, ocupación de instalaciones hospitalarias, formas y patrones de contagio, entre otras variables. Un equipo de trabajo del Centro de Investigación y Docencia Económicas (CIDE) liderado por el profesor Alarid-Escudero desarrollaron un modelo matemático de proyecciones sobre los efectos de las distintas prácticas de mitigación sobre COVID-19. El modelo denominado SC-COSMO (Stanford-CIDE CORonavirus Simulation MOdel) es un modelo matemático epidemiológico de cómo evoluciona la enfermedad y modela también los mecanismos en los que los individuos interactúan entre sí [3]. Incorpora análisis demográficos para considerar a los individuos susceptibles, a los expuestos, los infectados y los recuperados, los analiza conforme a los patrones de contacto para la transmisión. A partir de este análisis, se puede calcular la evolución de cómo las personas se contagian y desarrollan la enfermedad.

Al revisar las cifras de casos confirmados, se miden día a día los casos reales de infectados y la capacidad para detectarlos por parte de los gobiernos [4]. El tiempo es

un factor que permite observar los casos que mejoran, los decesos y la evolución en duplicación y las tendencias que siguen los casos a nivel local y global.

El “Covid-19 Modelo numérico de casos de infección y estimaciones epidémicas modelo asimétrico -Gompertz” [5] que por medio de la ecuación Gompertz obtiene una estimación de la demanda hospitalaria incluso para casos de terapia intensiva en España. Es un claro ejemplo de los beneficios del confinamiento y que permite a la sociedad tener una mejor atención médica en caso de ser contagiado.

Sitios en internet como Worldometer analizan, validan y agregan datos de miles de fuentes en tiempo real y proporcionan estadísticas globales COVID-19 para una amplia audiencia en todo el mundo [6]. Los análisis que se pueden realizar generan proyecciones distintas para casos locales.

En Korea, un estudio pretende identificar el patrón de transmisión local de COVID-19 utilizando modelos matemáticos para predecir el tamaño de la epidemia y el momento del final de la propagación, teniendo como resultados una estimación de que el número de transmisiones por paciente infectado era aproximadamente 10 veces mayor en el área de Daegu / Gyeongbuk que el promedio de todo el país [7].

Predecir el desarrollo de una epidemia no es tarea fácil, y menos cuando se tiene un virus emergente. La validez de la mayoría de los modelos predictivos se basa en numerosos parámetros, que involucran características biológicas y sociales a menudo desconocidas o altamente inciertas [8].

El análisis de los datos es relevante ya que ayuda a los gobiernos de los países a tomar decisiones sobre el proceso de transmisión de la Enfermedad COVID-19 por medio de predicción hacia adelante y de inferencias hacia atrás de la situación epidémica [9].

Existen fuentes que confirman que el aislamiento de personas expuestas e infectadas, y la reducción de la transmisión y la tasa de retorno de la estadía en el hogar pueden mitigar las pandemias [10]. Al igual que con todos los modelos matemáticos, la capacidad predictiva del modelo está limitada por la precisión de los datos disponibles y por el llamado nivel de abstracción utilizado para modelar el problema [11].

En el inicio de la pandemia en México la información del aumento de casos avanzaba lentamente lo cual no permitía generar modelos que describieran el comportamiento estatal y nacional, a partir de que a nivel nacional se alcanzaron los 500 casos confirmados positivos fue posible aplicar diversas estrategias metodológicas para extraer de la información reportada diariamente, modelos y tendencias de la información.

Aunque el número de casos acumulados aumenta diariamente se observaron comportamientos característicos en los reportes presentados, además se analizó el porcentaje de casos distribuido con respecto a las fechas de inicio de síntomas, las de ingreso y defunciones, de donde se encuentra que un gran porcentaje de los casos confirmados fallecen pocos días después de su ingreso a los servicios hospitalarios y más alarmante aún es que de estas lamentables defunciones la mayoría presentó síntomas mucho tiempo antes de su ingreso. Esta información es muy importante para nosotros porque reafirma el hecho de que el número de casos acumulados se alimenta con casos ocurridos varios días antes del reporte.

En este trabajo se propone una metodología para generar predicciones del número de casos diarios confirmados positivos mediante una doble regresión polinomial, hasta

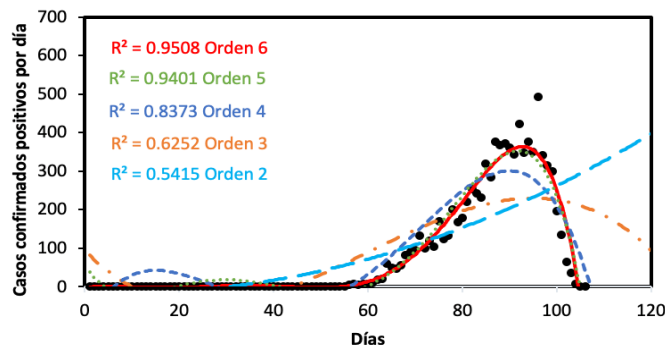


Fig. 1. Comparación de las curvas y coeficientes R^2 de los polinomios de ordenes 6,5,4,3 y 2 para describir los casos diarios confirmados positivos en el reporte oficial del día 20 de abril de 2020.

Tabla 1. Ecuaciones de ajuste y coeficientes de correlación.

Polinomio de ajuste	Coefficiente de correlación
$y = -3E-08x^6 + 8E-06x^5 - 0.0006x^4 + 0.0241x^3 - 0.4045x^2 + 2.5168x - 3.3367$	0.9508
$y = -3E-06x^5 + 0.0006x^4 - 0.0466x^3 + 1.5115x^2 - 18.575x + 55.462$	0.9401
$y = -1E-04x^4 + 0.0196x^3 - 1.1642x^2 + 23.104x - 102.98$	0.8373
$y = -0.0017x^3 + 0.3007x^2 - 12.113x + 93.385$	0.6252
$y = 0.0336x^2 - 0.6276x - 11.42$	0.5415

la fecha no se conoce ningún trabajo de esta naturaleza aplicado al análisis de la pandemia COVID-19.

El modelo utiliza los reportes oficiales publicados diariamente y genera los modelos matemáticos para cada reporte, de los cuales se obtienen sus coeficientes o variables numéricas que se optimizan y a partir de ellos generar una predicción del número de casos confirmados positivos diarios y de esta forma generar un modelo que describa mejor los escenarios futuros.

2. Metodología

Los datos que se obtuvieron de los informes diarios que provee el gobierno de México a través de la Secretaría de Salud y publicados en el portal oficial.

2.1. Modelo matemático

La información obtenida del reporte diario se filtra para considerar solamente los casos confirmados positivos, aunque la metodología se aplicara posteriormente a los casos negativos y a los casos pendientes de confirmación.

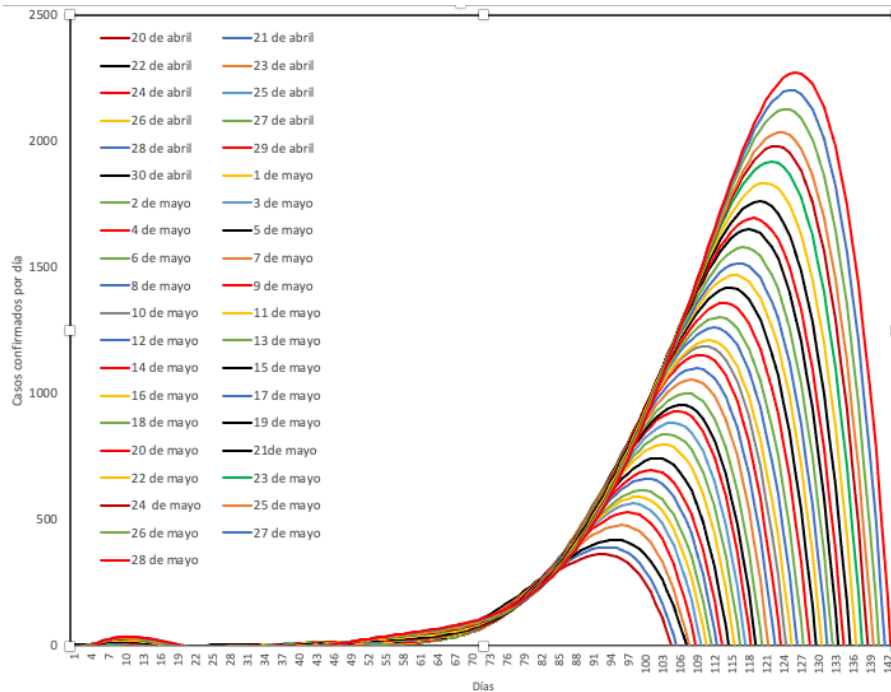


Fig. 2. Simulación de la curva reportada para cada día con el polinomio del orden que maximiza el coeficiente de correlación.

2.2. Metodología

A partir de un reporte oficial se filtra la información de casos confirmados positivos por fecha de inicio de síntomas y con una regresión polinomial se obtienen las ecuaciones que maximicen el coeficiente de correlación que se calcula a partir de la ecuación 1. Este proceso se realizó a partir del reporte del 20 de abril con un total de 8772 casos confirmados positivos y 108 días transcurridos desde el primer caso, como se muestra en la Tabla 1 y la figura 1:

$$R^2 = \frac{\sum_{t=1}^T (\hat{Y}_t - \bar{Y})^2}{\sum_{t=1}^T (Y_t - \bar{Y})^2}. \quad (1)$$

Una vez que se identifica que la regresión polinomial de orden 6 es el menor orden que maximiza el coeficiente de correlación en el mayor número de reportes se aplica este orden a todos los reportes y se obtiene la figura 2.

Los coeficientes de las curvas de la figura 2 se procesan y se obtiene ahora el polinomio que mejor describe su comportamiento, como tenemos polinomios de orden 6, serán 7 coeficientes que se procesan como 7 vectores de información correspondiente a las fechas de reporte. En este punto no es suficiente contar con el máximo coeficiente de correlación vector, ya que como se observa en la Figura 3 aunque la ecuación de mayor orden describe mejor la cantidad de puntos suministrada, sin embargo, esta puede tener cambios de pendiente muy fuertes después del último punto reportado y

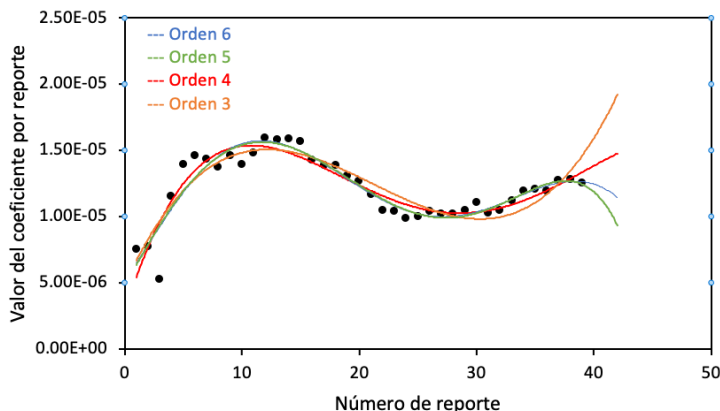


Fig. 3. Elementos del vector de coeficientes A_5 de los polinomios de orden 6 de la figura 2 y sus respectivos ajustes polinomiales de orden 6, 5, 4 y 3. Se puede observar que después del último punto reportado los polinomios de orden 4 y 5 presentan una pendiente similar a la que se encuentra en la zona de información reportada.

Tabla 2. Ecuaciones de ajuste y coeficientes de correlación por coeficiente.

Polinomio de ajuste	Coficiente de correlación
$y = 9E-14x^6 - 1E-11x^5 + 8E-10x^4 - 2E-08x^3 + 1E-07x^2 + 1E-06x + 5E-06$	0.8502
$y = -4E-12x^5 + 3E-10x^4 - 8E-09x^3 - 7E-09x^2 + 2E-06x + 5E-06$	0.8487
$y = -4.96E-11x^4 + 5.73E-09x^3 - 2.15E-07x^2 + 2.89E-06x + 2.72E-06$	0.821
$y = 2E-09x^3 - 1E-07x^2 + 2E-06x + 5E-06$	0.7703
$y = 0.0336x^2 - 0.6276x - 11.42$	0.5415

adoptar un comportamiento abrupto. Para resolver esto se debe analizar la pendiente de los polinomios de regresión obtenidos en puntos posteriores a los reportados a fin de describir escenarios futuros más confiables es decir menor error en la predicción.

Como se observa en la figura 3, los polinomios de regresión obtenidos tienen una pendiente en los puntos desconocidos similar a la pendiente en los puntos conocidos son los de orden 5, 4, 6 y 3 respectivamente, lo cual no se puede concluir del coeficiente de correlación, en cuyo caso nos indicaría 6, 5, 4 y 3, para el orden del polinomio más confiable.

De esta forma se obtiene una ecuación para calcular cada uno de los coeficientes de acuerdo con el día de reporte y utilizar la ecuación obtenida al inicio que maximizaba el coeficiente de correlación, con los nuevos coeficientes.

Finalmente, este proceso se valida en días previos con la reproducibilidad de la información inicial, no se espera que esta sea idéntica, porque ha pasado por una optimización en los coeficientes de la familia de curvas a cada reporte diario, como se muestra en los resultados.

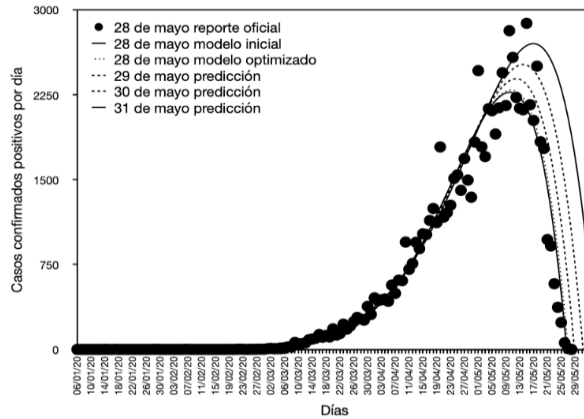


Fig. 4. Simulación con los coeficientes optimizados para días posteriores al reporte oficial del 28 de mayo.

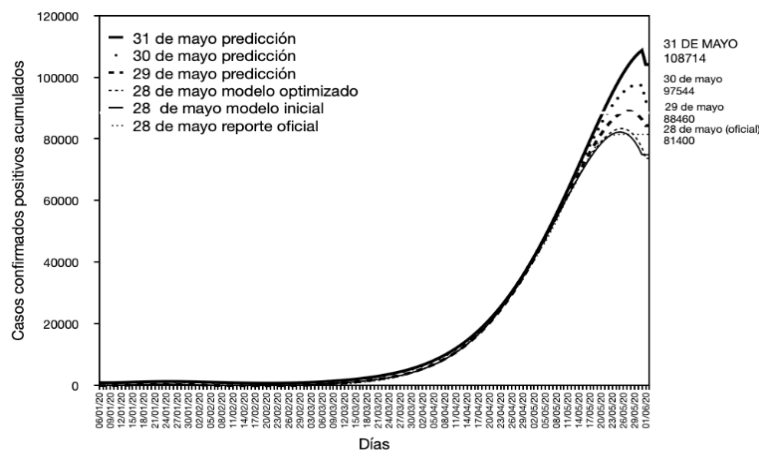


Fig. 5. Simulaciones utilizando los coeficientes de orden 5 optimizados con la pendiente más cercana a la presentada en los puntos reportados.

3. Resultados

Uno de los primeros resultados es la validación de los coeficientes calculados para cualquier día de los reportados, lo cual, como es de esperarse, no muestra diferencia notable con respecto a los previamente obtenidos y mostrados en la figura 2, posterior a este proceso se obtienen del mismo modelo los días posteriores a los reportados, como se muestran a continuación en las figuras 4 y 5.

Como se observa en las simulaciones de la figura 4 la validación con el modelo sin los coeficientes optimizados es muy buena y para días posteriores muestra una buena confiabilidad en la predicción las cuales son utilizadas para generar la gráfica de casos

Tabla 3. Porcentaje de error en las estimaciones a 3 días con Regresiones Polinomiales de orden 6 en el reporte diario y de orden 5 en los vectores de coeficientes.

FECHA	ERROR
28 de mayo de 2020	2.345%.
29 de mayo de 2020	5.118 %
30 de mayo de 2020	11.463 %
31 de mayo de 2020	19.9%

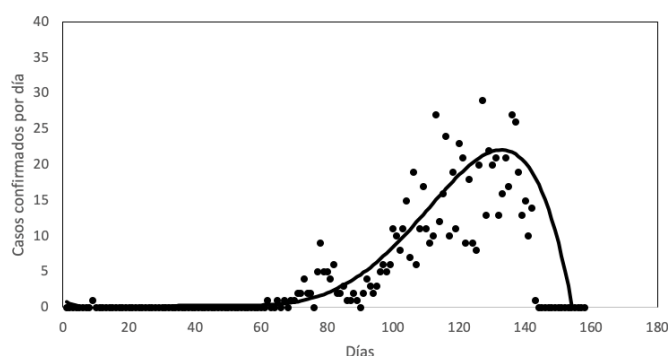


Fig. 6. Modelo para simular los casos confirmados positivos por día en Aguascalientes.

acumulados de la figura 5, en la que la validación presenta un error con respecto al reporte oficial de 0.971% el modelo simulado inicial y el modelo optimizado con 2.345%.

De acuerdo con los reportes de días posteriores se tienen los siguientes porcentajes de error de la tabla 3.

Los resultados encontrados al optimizar los coeficientes muestran un error creciente debido al ajuste en la optimización de los coeficientes. Sin embargo, se ha encontrado que muchos estados presentan comportamientos matemáticos distintos lo cual origina que el error empiece a crecer conforme se busca información en más días posteriores al reporte.

Este análisis por estados ya se está realizando debido a la alta complejidad de aplicar esta metodología a 32 estados por al menos 15 días, en cuyo caso la solución implica un sistema aproximado de 500X500 ya que el algoritmo lleva a cabo 5 regresiones polinomiales a cada estado y posteriormente otras 5 a cada uno de los 6 coeficientes de las anteriores así como el cálculo de los respectivos R^2 y las pendientes, lo cual ya está generando resultados que retroalimentarán el sistema aquí presentado.

En seguida mostramos algunos resultados, en la figura 6 se muestra el caso de Aguascalientes que muestra como el modelo de orden 4 maximiza el coeficiente de correlación, pero aun así la alta dispersión de la información reportada genera que el modelo presente porcentajes de error altos en los modelos optimizados. En la figura 7 se observa el modelado con polinomios de orden 5 de casos diarios para Baja California, Ciudad de México y Estado de México.

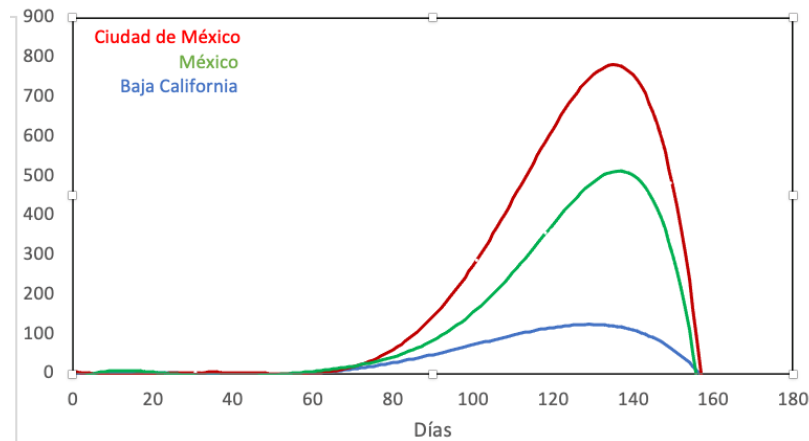


Fig. 7. La optimización de los coeficientes en estados como Ciudad de México soporta hasta un orden 6 a diferencia de estados como Aguascalientes donde el máximo orden es 4.

4. Conclusiones

Se desarrolla una propuesta metodológica mediante una regresión polinomial de orden 6 para cada reporte diario de casos confirmados positivos seguida de otra de orden 5 para cada uno de los 7 coeficientes de los polinomios anteriores, donde el orden 6 es el menor orden que maximiza R^2 en la mayoría de los reportes y el orden 5 se obtiene en base al análisis de la pendiente en los coeficientes. La aplicación de esta metodología brindó resultados con un porcentaje de error del 5.1% a un día, 11.4% a dos días y 19.9% a tres días en el número de casos confirmados positivos diarios y aunque el error aumenta, esta metodología brinda muy buenos resultados en la mayoría de los estados y a nivel nacional, ya se está desarrollando el análisis por municipios para retroalimentar esta matriz a fin de localizar a tiempo los puntos de crecimiento acelerado y su contribución en el crecimiento global de casos confirmados positivos a nivel nacional. Debido a las características de la metodología utilizada, se utilizará también para generar el escenario de defunciones, casos negativos y pendientes a fin de integrar todos en un análisis con modelos más confiables para aplicar otras técnicas de pronóstico.

Referencias

1. OMS: Neumonía de Causa desconocida – China. <https://www.who.int/csr/don/05-january-2020-pneumonia-of-unkown-cause-china/es/> (2020)
2. Novel Coronavirus (2019-nCoV): Situation Report-10. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200130-sitrep-10-ncov.pdf?sfvrsn=d0b2e480_2 (2020)
3. CIDE: CIDE y Stanford desarrollan modelo matemático de proyecciones sobre COVID-19. <https://www.cide.edu/saladeprensa/cide-y-stanford-desarrollan-modelo-matematico-de-proyecciones-sobre-covid-19/> (2020)

Ana C. Zenteno, María del C. Santiago, Yeiny Romero, Judith Pérez, Gustavo T. Rubín, et al.

4. El País: Así evoluciona la curva del coronavirus en México, Colombia, Chile, Argentina y el resto de Latinoamérica. https://elpais.com/sociedad/2020/04/07/actualidad/1586251212_090043.html (2020)
5. Burgos, P.: COVID-19 Modelo numérico de casos de infección y estimaciones epidémicas Modelo Asimétrico -GOMPERTZ. 10.13140/RG.2.2.19440.40969 (2020)
6. Worldometers: COVID-19 Coronavirus Pandemic. <https://www.worldometers.info/coronavirus/> (2020)
7. Prediction of COVID-19 transmission dynamics using a mathematical model considering behavior changes in Korea. <https://pesquisa.bvsalud.org/portal/resource/es/mdl-32375455> (2020)
8. Chaos theory applied to the outbreak of COVID-19: an ancillary approach to decision making in pandemic context. <https://pesquisa.bvsalud.org/portal/resource/es/mdl-32381148> (2020)
9. Propagation analysis and prediction of the COVID-19. <https://www.sciencedirect.com/science/article/pii/S2468042720300087> (2020)
10. Model the transmission dynamics of COVID-19 propagation with public health intervention. <https://www.medrxiv.org/content/10.1101/2020.04.22.20075184v1> (2020)
11. Mathematical Modeling of Epidemic Diseases: A Case Study of the COVID-19 Coronavirus. <https://arxiv.org/pdf/2003.11371.pdf> (2020)

Plataforma de gestión IoT mediante técnicas de industria 4.0 para agricultura de precisión

Alí Pérez Gómez, Adriana Carolina Sánchez Cahuich,
Janeth Jiménez Gómez

Instituto Tecnológico Superior de Coatzacoalcos,
México

aperezg@itesco.edu.mx

Resumen. El desarrollo tecnológico y su implementación para la resolución de problemas concretos de la vida cotidiana ha ido creciendo de manera más acelerada en los últimos años. A pesar de ello, es evidente que este avance no ha sido gradual para todos los sectores, como por ejemplo en el caso del sector económico primario en donde existen muchas áreas de oportunidad que no se han atendido. Por lo anterior, a partir de una problemática analizada, se plantea la creación de un Sistema de Gestión inteligente por la implementación de machine learning a través de una celda que con el uso de TI permiten la automatización en el análisis de una muestra del proceso de cultivo, en este caso del cultivo del maíz en el sector primario que se ve afectado por los cambios climáticos y que debido a la pandemia por el COVID-19 ha descendido sus posibilidades de ventas por la falta de acceso a otras ciudades afectando de forma económica y social no solo a los que producen sino a los involucrados en la cadena de suministro. Por ello, se propone proveer de una herramienta a este sector para la mejora del proceso con la ayuda de recursos tecnológicos para manipular y gestionar la información utilizando técnicas de Industria 4.0 mediante un sistema IoT (Internet de las Cosas).

Palabras clave: Industria 4.0, Machine Learning, IoT.

IoT Management Platform using Industry 4.0 techniques for precision Agriculture

Abstract. Technological development and its implementation for solving specific problems of daily life has been growing more rapidly in recent years. Despite this, it is evident that this progress has not been gradual for all sectors, as for example in the case of the primary economic sector, where there are many areas of opportunity that have not been addressed. Therefore, based on an analyzed problem, the creation of an intelligent Management System is proposed by the implementation of machine learning through a cell that with the use of IT allows automation in the analysis of a sample. tra of the cultivation process, in this case the cultivation of corn in the primary sector that is affected by climatic

changes and that due to the COVID-19 pandemic has decreased its sales possibilities due to the lack of access to other cities affecting economically and socially not only those who produce but those involved in the supply chain. Therefore, it is proposed to provide a tool to this sector to improve the process with the help of technological resources to manipulate and manage information using Industry 4.0 techniques through an IoT (Internet of Things) system.

Keywords: Industry 4.0, Machine Learning, IoT.

1. Introducción

La escasez de alimentos es una problemática que ha permanecido a nivel mundial por distintas causas entre las que se encuentran pobreza extrema, inestabilidad laboral alrededor del mundo, desperdicio de alimentos, entre otras, que en momentos críticos como los que se viven hoy en día con la pandemia por el COVID-19, se hacen todavía más presentes y tienen mayores repercusiones.

A medida que los países avanzan en la adopción de medidas destinadas a detener la expansión de la pandemia del COVID-19, debe tenerse cuidado de reducir al mínimo las posibles repercusiones en el suministro de alimentos o las consecuencias imprevistas en el comercio mundial y la seguridad alimentaria. [1].

En la situación particular de México uno de los sectores con mayor impacto en la economía nacional y con más afectaciones a su vez es el sector primario que se refiere a las actividades como la agricultura, ganadería y pesca.

De acuerdo con la Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA), los sectores agrícola y pecuario son los más dinámicos del país, representando cerca del 10% del PIB. Por lo que en el presente artículo se plantea una propuesta del diseño una celda inteligente para la mejora del proceso de producción del sector agrícola como parte de la integración de este sector del país a los nuevos rubros como el término agricultura de precisión relacionado con la tecnología IoT a implementar en la celda. Cabe destacar que dentro de estos nuevos rubros existen limitantes que se van a considerar ya que, en la agricultura de precisión, el IoT está bien establecido, pero solo se están implementando soluciones patentadas, lo que genera problemas de compatibilidad y conexión entre diferentes dispositivos. [2] Sin embargo, en contraste con el problema actual de la creciente demanda de alimentos, tanto en cantidad como en calidad, ha planteado la necesidad de intensificar e industrializar el sector agrícola, sobre todo en estos momentos por motivos de la pandemia, motivo por el cual el "Internet de las cosas" (IoT) que es una familia de tecnologías muy prometedora, es capaz de ofrecer muchas soluciones para la modernización de la agricultura [3].

Como parte de la delimitación la investigación se centra en el sector dedicado en específico al cultivo del maíz, que se ve afectado por los cambios climáticos lo que afecta el abastecimiento a través de la cadena de suministro en conjunto con el problema de la pandemia que imposibilita y/o ha disminuido las oportunidades de comercio del producto hacia lugares más alejados de quienes lo producen afectándolos de forma económica.

Uno de los productos clave de la dieta de la población mexicana es el maíz grano, sin embargo, se ha analizado que, así como en el estado de Sinaloa, gracias a la infraestructura de riego y su avanzada capacidad tecnológica se le considera líder productor de este grano a nivel nacional, existen estados menos favorecidos no sólo por las condiciones de infraestructura sino también por las condiciones ambientales como es el caso del estado de Oaxaca, la problemática maicera adquiere mayor relevancia, tanto por las condiciones sociodemográficas prevalecientes como por el predominio de las actividades económicas orientadas a la agricultura de subsistencia. [4] Esta situación adquiere mayor relevancia al verse afectada aún más por la cuestión sanitaria actual, generando en primer instancia mayor, desfavorecimiento económico para la región y al mismo tiempo la imposibilidad de abastecimiento a zonas cercanas llega a contribuir a la escasez.

En la actualidad el término industria 4.0 figura cada vez con mayor aceptación en las investigaciones dentro de diversas áreas del conocimiento también cuando hablamos de desarrollo sostenible, se considera imprescindible describir las generaciones anteriores, sus objetivos, retos y desafíos dentro de la industria inteligente ya que las tecnologías han ido avanzando e integrándose a los objetivos que logran ese esperado desarrollo sostenible en cada uno de los procesos de la industria.

Las ciudades enfrentan grandes desafíos debido a la presión económica, ambiental y de urbanización. Las ciudades inteligentes usan el poder de las redes integradas, sensores inalámbricos y sistemas inteligentes de administración para resolver los desafíos.

Se investigó anteriormente la Integración de Tecnología Educativa para sensores ambientales, hacia ciudades inteligentes en zonas urbanas [5]. Sin embargo, para el caso de las zonas rurales los objetivos y esfuerzos son diferentes, aunque es claro que el uso de elementos como IoT puede mejorar cualquier proceso existente para su análisis.

Por lo anterior, es que para el proyecto propuesto se tomó la opción de integrar nuevas tecnologías en procesos que por su manera tradicional oponen resistencia, mostrando a través de la investigación y las pruebas los beneficios que puede generar al hacer uso de ellas para favorecer a un sector que es vital para la economía del país. Todo ello con apoyo del uso de tecnología IoT mostrada en un Dashboard para el análisis del proceso que se lleva a cabo desde el cultivo hasta el momento que se cosecha para llevarlo al siguiente sector, lo que genera datos importantes a partir de las pruebas que se pueden usar para la implementación del machine learning.

2. Metodología

2.1. Sector productivo de maíz en el estado de Oaxaca

Para hacer un análisis más específico sobre las afectaciones de condiciones ambientales al cultivo del maíz, se ha tomado como referente el estado de Oaxaca en donde hay mayor desfavorecimiento por la zona, además, se pretende hacer la propuesta con la finalidad de apoyar a los productores a la comercialización ya que la producción es principalmente para autoconsumo representando un importante acervo de maíz en México. Como antecedente se investigaron los procesos de crecimiento y las



Fig. 1. Representación de la subdivisión del crecimiento de la planta de maíz. Fuente: Propia.

Tabla 1. Etapas de crecimiento de la planta de maíz y sus características [6].

Etapa	DAS*	Características
VE	5	El coleoptilo emerge de la superficie del suelo
V1	9	Es visible el cuello de la primera hoja.
V2	12	Es visible el cuello de la segunda hoja.
Vn		Es visible el cuello de la hoja número "n". ("n" es igual al número definitivo de hojas que tiene la planta; "n" generalmente fluctúa entre 16 y 22, pero para la floración se habrán perdido las 4 a 5 hojas de más abajo.)
VT	55	Es completamente visible la última rama de la panícula.
R0	57	Antesis o floración masculina. El polen se comienza a arrojar.
R1	59	Son visibles los estigmas.
R2	71	Etapa de ampolla. Los granos se llenan con un líquido claro y se puede ver el embrión.
R3	80	Etapa lechosa. Los granos se llenan con un líquido lechoso blanco.
R4	90	Etapa masosa. Los granos se llenan con una pasta blanca. El embrión tiene aproximadamente la mitad del ancho del grano.
R5	102	Etapa dentada. La parte superior de los granos se llena con almidón sólido y, cuando el genotipo es dentado, los granos adquieren la forma dentada. En los tipos tanto cristalinos como dentados es visible una "línea de leche" cuando se observa el grano desde el costado.
R6	112	Madurez fisiológica. Una capa negra es visible en la base del grano. La humedad del grano es generalmente de alrededor del 35%.

* **DAS:** número aproximado de días después de la siembra en tierras bajas tropicales, donde las temperaturas máxima y mínima pueden ser de 33°C y 22°C, respectivamente. En los ambientes más frío, se amplían estos tiempos.

condiciones ideales del ambiente que deben ser las idóneas. Posteriormente se muestra el sistema creado y el análisis como resultado de las pruebas.

2.2. Etapas del proceso de crecimiento

Con apoyo de herramientas Lean se planea la mejora de la metodología del proceso que se lleva a cabo para el cultivo del maíz, sin embargo, antes de entrar en lo propuesto acerca de las metodologías y los recursos técnicos se consideró necesario tener



Fig. 2. Entre los días mojados, distinguimos entre los que tienen solamente lluvia. El tipo más común de precipitación durante el año es solo lluvia, con una probabilidad máxima del 54 % el 21 de junio [10].



Fig. 3. Celda de monitoreo de adquisición de datos en tiempo real para el monitoreo de cosechas de maíz. Fuente: Propia.

conocimiento del proceso evolutivo estandarizado de este producto, para tener un panorama más amplio y obtener como resultado del análisis la etapa en la que mejor se adaptarían los procedimientos para implementarlas.

Los investigadores asumen que el cultivo alcanza una etapa específica cuando al menos el 50% de las plantas presentan las características correspondientes.

La normalización de las definiciones permite que los investigadores se refieran a los problemas de las etapas de crecimiento específicas. [6] De manera general, los investigadores establecieron dos clasificaciones que son Vegetativa (V) y Reproductiva (R), pero a partir de estas, se hicieron subdivisiones más específicas que son: Crecimiento de las plántulas (etapas VE y V1), Crecimiento vegetativo (etapas V2, V3...Vn), Floración y la Fecundación (etapas VT, R0, y R1), Llenado de grano y la madurez (etapas R2 a R6), se muestra su representación en la Figura 1.

Las etapas descritas en las clasificaciones se refieren a las de la Tabla 1 que deriva de la información de las clasificaciones hechas por los investigadores mencionados, en la cual se muestran más detalles de los segmentos de crecimiento como su evolución de acuerdo a los días que pasan después de la siembra.



Fig. 4. Esquema de herramientas implementadas en el desarrollo de la celda. Fuente: Propia.

2.3. Condiciones climáticas

El seguimiento de los factores ambientales no es la solución completa para incrementar el rendimiento de los cultivos. La forma en que los agricultores de forma tradicional llevan este seguimiento es a través de anotaciones o simplemente por costumbre, lo que en la actualidad les trae consecuencias debido a los cambios climáticos que han surgido en los últimos años. Hay otros factores que disminuyen la productividad en mayor medida. Por lo tanto, la automatización debe implementarse en la agricultura para superar estos problemas [7].

Temperatura. El maíz es un cultivo que se desarrolla mejor en suelos ligeramente ácidos, con alta materia orgánica y buen drenaje. En relación al clima, el maíz se desarrolla mejor en regiones con temperaturas entre 25 y 30 °C. Dependiendo de la variedad y del clima [8]. En cuanto a la zona istmo según la Secretaría de Agricultura, Ganadería y Desarrollo Rural, la temperatura media anual en Oaxaca fluctúa entre 26°C y 45°C; en el mes más frío, la temperatura varía entre 3°C (en algunas microregiones) y 18°C [9].

Humedad y época. El maíz requiere de 500 a 800 mm de agua. Los períodos críticos de humedad son la germinación, las primeras tres semanas de desarrollo y dos semanas antes y después de la floración que corresponden a la formación de espiga, mazorca y llenado de grano [8]. En la siguiente grafica (Figura 2) se puede observar los meses en los cuales la zona istmo de Oaxaca tiene mayores probabilidades de lluvia son a mediados de mayo y mediados de octubre.

2.4. Tecnología implementada en la celda

El desarrollo de la celda consta de la fusión de un sistema de adquisición de datos (sensores), tecnología IoT (Internet de las cosas) que se refiere a la interconexión de

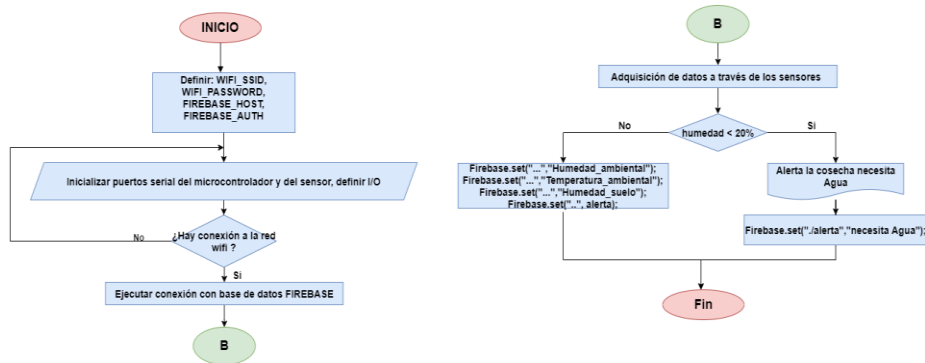


Fig. 5. Diagrama del funcionamiento de la lógica del sistema para la adquisición de datos. Fuente: Propia.

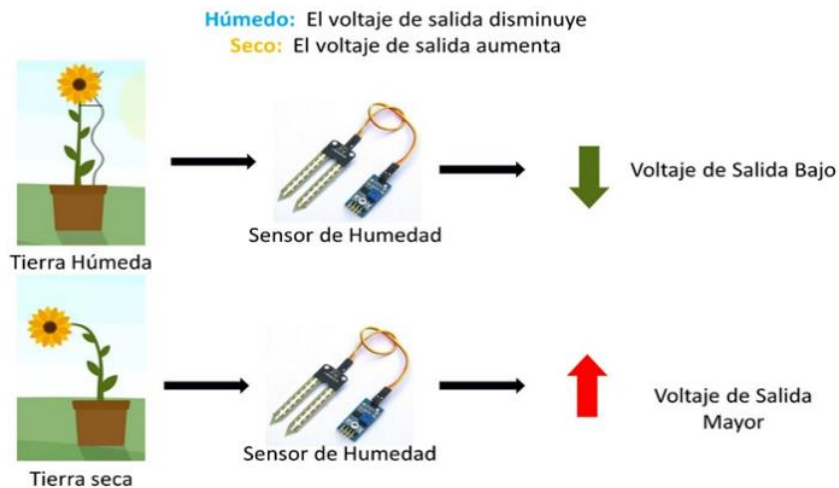


Fig. 6. Funcionamiento del sensor de humedad HL-69. Fuente: Propia.

dispositivos físicos, vehículos (también denominados dispositivos conectados y "dispositivos inteligentes"), edificios y otros elementos integrados con electrónica, software, sensores, actuadores, y conectividad de red que permite a los objetos recopilar e intercambiar datos [11].

Para este caso, se le da el uso al IoT como una herramienta para gestión de datos, pues por el rubro del sector agrícola para tomar decisiones rentables, los agricultores necesitan información durante todo el ciclo de cultivo.

La información requerida se encuentra dispersa en varios lugares que incluye información en tiempo real, como precios de mercado y estadísticas de nivel de producción actual, junto con el conocimiento disponible de cultivos primarios [12].

Y también, consta de un Dashboard que genera: el sistema integral de adquisición de datos para poder llegar al análisis. En la figura 4 se muestran todos los elementos empleados para el desarrollo del sistema integral de adquisición de datos.

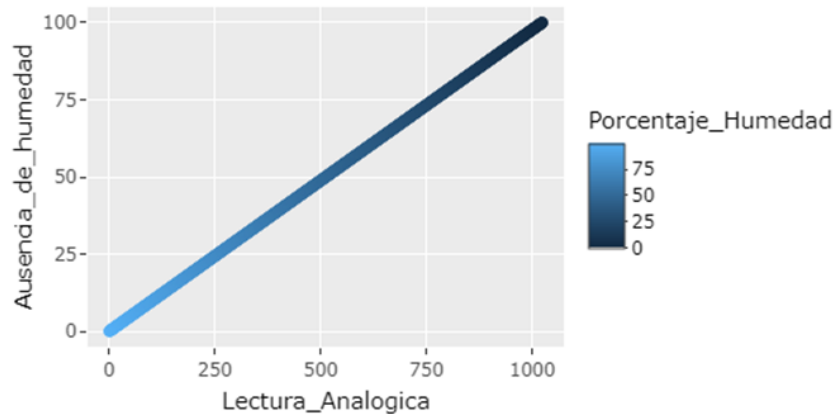


Fig. 7. Gráfica de la ausencia de humedad y el porcentaje de humedad con relación a la lectura analógica proporcionada por el sensor. Fuente: Propia.

Tabla 2. Relación lectura analógica con porcentajes de humedad obtenidas a través del sensor HL-69

Estado del suelo	AnalogRead	Porcentaje de Humedad
Húmedo	0	100%
	256	75%
	512	50%
Seco	768	25%
	1024	0%

2.5. Adquisición de datos

Al implementar tecnología IoT es necesario trabajar con un microcontrolador que cuente con wifi, en esta investigación se trabajó con la ESP8266 por su bajo costo, además de cumplir con las características necesarias para su implementación en la celda. Los sensores implementados son el DHT22 (temperatura y humedad ambiental) y HL-69 (humedad del suelo) los cuales son alimentados por el mismo microcontrolador con un voltaje de 3.3V. La ESP8266 fue programada en el IDE de Arduino, la programación funciona en base al siguiente diagrama (Figura 5).

El sistema de adquisición se desarrolla a través del microcontrolador este se encarga de activar I/O (input /Output) para obtener los datos de los sensores. El sensor DHT22 cuenta con una librería en el IDE de Arduino el cual permite obtener por medio de una función las lecturas de humedad y temperatura ambiental.

El sensor HL-69 requiere de una entrada del tipo analógica y un pequeño cálculo interno para mostrar el porcentaje de humedad. El sensor se rige bajo la siguiente lógica (Figura 6) esta toma en cuenta al suelo como “húmedo” cuando da un voltaje de salida menor y “seco” cuando arroja un voltaje mayor. Utilizando la función analogRead en



Fig. 8. Dashboard del sistema de adquisición de datos en tiempo real. Fuente: Propia.

el IDE de Arduino se obtienen valores que van de 0 ~ 1024 esto debido al conversor analógico digital (ADC) que posee el microcontrolador.

A través de una regla de tres se hizo un cálculo para obtener el porcentaje de ausencia de humedad que posee la tierra, se denomina ausencia de humedad debido a que al análisis de la Figura 6. Se generó una escala propia (Tabla 2) la lectura del sensor proporciona como tal los valores del suelo seco.

Se considera suelo seco a valores en donde la humedad tenga un porcentaje menor al 50% o sea lecturas analógicas mayores a 512 como indica la Tabla 2.

$$\frac{1024}{X} = \frac{100}{Y} \text{ regla de tres aplicada al análisis ,} \quad (1)$$

$$Y=100/1024 X.$$

Siendo “Y” el porcentaje de ausencia de humedad y “X” el valor obtenido por la entrada analógica del microcontrolador.

Para poder obtener el porcentaje de humedad, tendremos que restarle al 100 por ciento el valor de “Y” (ausencia de humedad). Dichos cálculos generan la gráfica siguiente (Figura 7).

Pasando (1) a código en el IDE de Arduino queda de la siguiente forma:

```
float humedad_tierra_analogico = analogRead(A0);
float humedad_analogica= ((0.09765625)*humedad_tierra_analogico);
if(humedad_tierra_analogico>=1023)
{
    porcentaje_humedad = 0;
}else{
    porcentaje_humedad = 100 - humedad_analogica;
}

```

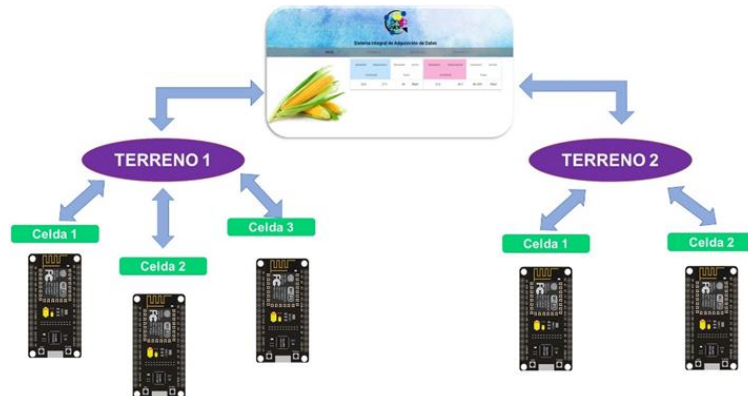


Fig. 9. Diagrama del funcionamiento de los microcontroladores con el Dashboard. Fuente: Propia.

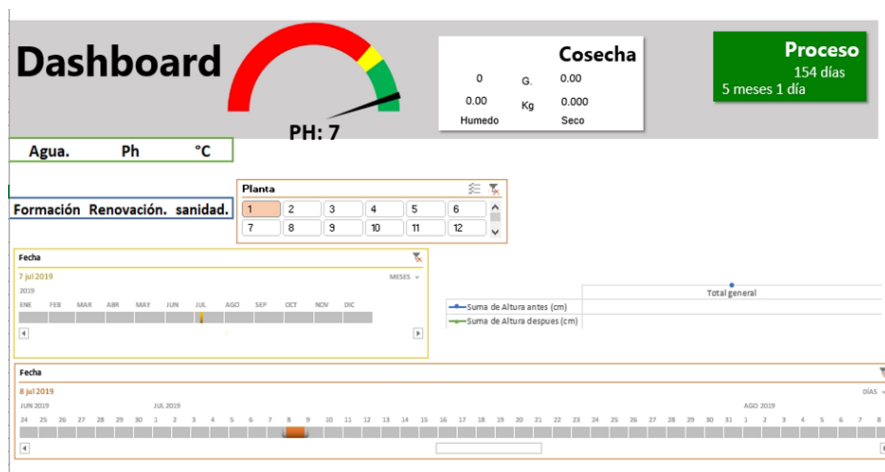


Fig. 10. Dashboard para la generación de una muestra de datos para una prueba t para medias de dos muestras emparejadas. Fuente: Propia.

Se optó por poner una condicional cuando la lectura analógica del sensor arroje datos mayores o iguales a 1023 ya que como indica la gráfica de la Figura 7 este tiende a tener un 100% de ausencia de humedad lo cual deja con 0% de humedad de la tierra, y así evitar errores en el análisis.

Para poder enviar los datos hacia el Dashboard se requiere de una base de datos del tipo no relacional, en esta investigación se trabajó con Firebase.

Firebase es una plataforma para el desarrollo de diversas aplicaciones web, cuenta una base de datos en tiempo real. Se requirió la librería de Firebase para poder enviar los datos obtenidos a través de los sensores hacia la base de datos. Para ello se utiliza la función `Firestore.set`.

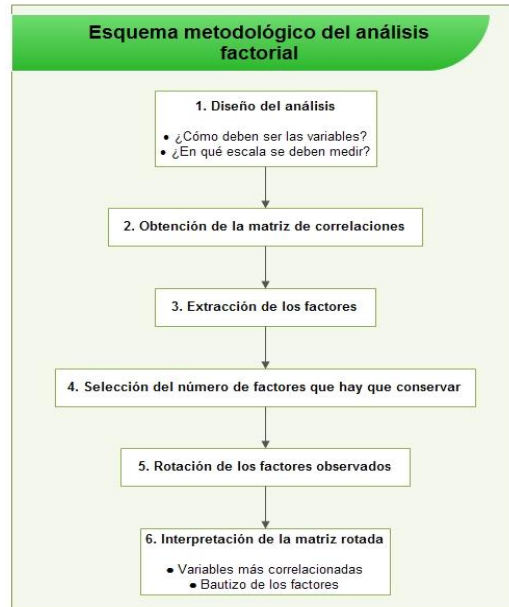


Fig. 11. Esquema metodológico del análisis factorial para la prueba t para medias de dos muestras emparejadas. Fuente: Propia.

Table 3. Prueba t para medias de dos muestras emparejadas.

	Altura antes (cm)	Altura después (cm)
Media	13.85	12.2625
Varianza	1.839130435	0.072880435
Observaciones	24	24
Coefficiente de correlación de Pearson	-0.988654763	
Diferencia hipotética de las medias	0	
Grados de libertad	23	
Estadístico t	4.790191986	
P(T<=t) una cola	3.92721E-05	
Valor crítico de t (una cola)	1.713871528	
P(T<=t) dos colas	7.85443E-05	
Valor crítico de t (dos colas)	2.06865761	

2.6. Dashboard

El desarrollo del Dashboard se llevó a cabo con un Frameworks que facilita el diseño del Dashboard en la parte del frontend modificando la parte estética y visual. El Frameworks implementado fue Material Desing Lite, al ejecutarse la función

Firestore.set en el código del microcontrolador a través de la red wifi envía los datos obtenidos de los sensores hacia el hosting utilizado en el Dashboard para poder reflejar los datos en tiempo real. Figura 8.

El Dashboard del sistema de adquisición tiene la capacidad de monitorear diversos terrenos a la vez cada uno con un máximo de tres celdas de investigación como se muestra en la Figura 9.

3. Resultados

La investigación sobre elementos y las condiciones ideales para el cultivo de la planta del maíz, así como el análisis para la creación del sistema de gestión a través de la Celda que lleva a cabo la recepción de datos por medio de los sensores conectados por IoT, permitió obtener como resultado la adquisición de datos en tiempo real sobre las condiciones ambientales con las que interactúa el cultivo muestra en la celda y gracias a esta recolección de datos es como se pudo generar el Dashboard en la Figura 8 que brindó información para el tratamiento de datos posterior en otro Dashboard para la interpretación de la información recolectada en tiempo real.

Para poder generar el tratamiento de los datos anteriores gestionados en el Dashboard, se siguió el esquema metodológico del análisis factorial Figura 11.

Siguiendo con el esquema metodológico se sacó una correlación entre las variables de distancia de las semillas, tomando como muestra un total de 30 semillas.

Los cálculos de las muestras dieron como resultado la media, la varianza y los grados de libertad mostrados en la Tabla 3. Que nos indica la variabilidad de las estimaciones y una distribución específica.

Esto da pauta a un mayor aprovechamiento para el sector primario, ya que la finalidad es optimizar el tiempo que requiere el conocimiento del estado de los cultivos que realizan por lo tanto la implementación del sistema les permitiría visualizar en tiempo real a través de las muestras colocadas en la celda, cómo se encuentra lo cultivado y qué factores podrían llegar a afectarlo de acuerdo a los parámetros establecidos y la semaforización que se puede implementar a través del uso de los datos para machine learning y el tratamiento de los datos por medio de muestras para permitir una toma de decisiones anticipada que prevenga las pérdidas de los cultivos.

4. Conclusiones

De acuerdo con el sistema realizado en primera instancia la modificación del proceso convencional para este sector significa un gran avance social ya que el hecho de permitir el análisis remoto a través de las muestras y cuantificación de variables en los cultivos para conocer en qué instancia para generar un Dataset y las condiciones que provocan afectaciones y pérdidas que significan un gran apoyo para el sector y una mejora en sus cosechas para posibles incorporaciones de algoritmos de entrenamiento de Machine Learning en la toma de decisiones del cambio climático.

Referencias

1. Dongyu, Q., et al.: Mitigar los efectos del Covid-19 en el comercio y los mercados de alimentos, Organización Mundial de la Salud (2020)
2. Ojha, T., Misra, S., Raghuwanshi, N.: Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges, *Computers and Electronics in Agriculture*, 118, pp. 66–84 (2015)
3. Tzounis, A., Katsoulas, N., Bartzanas, T., Kittas, C.: Internet de las cosas en la agricultura, avances recientes y desafíos futuros. *Ingeniería de biosistemas*, 164, pp. 31–48 (2017)
4. Martínez, E.B.: El maíz en Oaxaca: la cosecha de contradicciones. *Estudios Agrarios*, Universidad Autónoma Benito Juárez de Oaxaca (2006)
5. Maxwell, J.C.: *A Treatise on Electricity and Magnetism*, 2, Oxford: Clarendon, pp. 68–73 (1892)
6. CIBIOGEM: Maíz, Conacyt.gob.mx (2019)
7. Suma, N., Samson, S.R., Saranya, S., Shanmugapriya, G., Subhashri, R.: Sistema de monitoreo de agricultura inteligente basado en IOT. *Revista internacional sobre tendencias recientes e innovadoras en informática y comunicación*, 5(2), pp. 177–181 (2017)
8. Jurado-Guerra, P., Lara-Macías, C.R., Saucedo-Terán, R.A.: Paquete tecnológico para la producción de maíz forrajero en Chihuahua (2014)
9. Balamurugan, C., Satheesh, R.: Desarrollo de dispositivos de monitoreo y control basados en Raspberry pi y IoT para la agricultura. *Revista de Ciencias Sociales, Tecnológicas y Ambientales*, 6(2), pp. 207–15 (2017)
10. Instituto Nacional de Estadística, Geografía e Informática: El maíz en el estado de Oaxaca (1997)
11. WeatherSpark: El clima promedio en Santo Domingo Tehuantepec México. WeatherSpark (2020)
12. Mohanraj, I., Ashokumar, K., Naren, J.: Field monitoring and automation using IOT in agriculture domain. *Procedia Computer Science*, 93, pp. 931–939 (2016)

Sistema IoT para el seguimiento y análisis de la intensidad de luz en plantas de interiores

Gabriel Elías Chanchí Golondrino¹,
Manuel Alejandro Ospina Alarcón¹, Wilmar Yesid Campo Muñoz²

¹ Universidad de Cartagena,
Colombia

² Universidad del Quindío,
Colombia

{gchanchig, mospinaa}@unicartagena.edu.co,
wycampo@uniquindio.edu.co

Resumen. En este trabajo proponemos como aporte el desarrollo de un sistema IoT que permita la captura, almacenamiento, análisis y visualización de datos de luminancia en plantas de interior. A partir de la amplia difusión de temas como internet de las cosas (IoT), su uso se ha hecho posible en contextos como: agricultura de precisión o monitoreo de variables ambientales. A nivel de cuidado de las plantas de interior, existen variables como la luminancia que influyen directamente en el proceso de fotosíntesis, y por tanto en el crecimiento y desarrollo de las plantas. Si bien existen dispositivos de medición comerciales que permiten la captura y transmisión de estas variables de interés, entre sus desventajas se encuentran el almacenamiento limitado y la posibilidad de aplicar algoritmos de aprendizaje automático a los datos.

Palabras clave: Plantas de Interior, Internet de las Cosas, Sistema IoT, Luminancia.

IoT System for Monitoring and Analysis of Light Intensity in Indoor Plants

Abstract. In this paper, we propose as a contribution the development of an IoT system that allows the capture, storage, analysis, and visualization of luminance data in indoor plants. Based on the wide dissemination of topics such as the internet of things (IoT), its use has been made possible in contexts, such as precision agriculture or the monitoring of environmental variables. At the level of care for indoor plants, there are variables such as luminance that have a direct influence on the photosynthesis process, and therefore the growth and development of plants. Although there are commercial measurement devices that allow the capture and transmission of these variables of interest, among

their disadvantages are limited storage and the possibility of applying machine-learning algorithms to data.

Keywords: Indoor Plants, Internet of Things, Iot System, Luminance.

1. Introducción

Internet de las cosas (IoT) puede ser entendido como la conexión de objetos físicos o dispositivos en una red abierta que tiene la capacidad de auto gestionarse, compartir información, datos y recursos, reaccionar y actuar frente a situaciones y cambios en el entorno [1, 2]. Dentro de los desafíos de IoT en diferentes contextos de aplicación, se encuentra el aprovechamiento de los grandes volúmenes de datos de cara a la obtención de información útil para la toma de decisiones [3, 4].

Con la amplia difusión que ha tenido la temática de internet de las cosas (IoT), se ha posibilitado la captura, seguimiento y análisis de variables de interés en diferentes contextos, siendo la agricultura de precisión y la monitorización de variables ambientales un caso específico de aplicación [5, 6]. En este sentido, a nivel del cuidado de las plantas de interiores, es posible hacer uso de las ventajas provistas por IoT en cuanto a la monitorización y análisis de un conjunto de variables que tienen una influencia significativa en el crecimiento y desarrollo de las plantas [7].

Tanto en el caso de las plantas de interiores, como de exteriores la intensidad de la luz es uno de los parámetros fundamentales en el proceso de fotosíntesis, el cual tiene una influencia directa en el crecimiento de las plantas [8]. En el caso particular del cuidado de las plantas de interiores, existen especies que requieren diferentes niveles de intensidad luminosa a lo largo del día, por lo que la monitorización y análisis de esta variable puede ser de gran interés para la fisiología de las plantas [9]. Aunque se ha evidenciado en la literatura la existencia de varios trabajos en cuanto al cuidado de huertas urbanas y al seguimiento de variables de interés en plantas, se ha observado que dichos trabajos se enfocan principalmente en variables como la temperatura y la humedad [7, 10].

Del mismo modo, a partir de la difusión de IoT, se han desarrollado diferentes dispositivos comerciales como es el caso de los luxómetros, los cuales permiten la captura, visualización local y eventualmente la transmisión vía bluetooth de los datos de la intensidad de la luz (en *luxes*). Una de las ventajas de estos dispositivos comerciales es la calibración y la precisión de los datos capturados, mientras que dentro de las desventajas se encuentran el almacenamiento limitado, el uso de formatos propios en los datos transmitidos y la ausencia en el uso de algoritmos de *machine-learning* sobre el histórico de los datos [11]. En este sentido se hace necesario enriquecer las características de estos dispositivos, vinculando las funcionalidades de las diferentes capas de la arquitectura convencional de IoT [5, 12].

En este artículo se propone el diseño y desarrollo de un sistema para el seguimiento y análisis de la intensidad de luz o luminancia en plantas de interiores, teniendo en cuenta para ello la arquitectura convencional a cuatro capas de IoT (captura, almacenamiento, análisis, visualización) [5]. Así, en la capa de captura del sistema se cuenta con un luxómetro comercial *bluetooth* el cual tiene asociado un sensor de intensidad de luz que permite la captura de las mediciones en *luxes*. Los datos

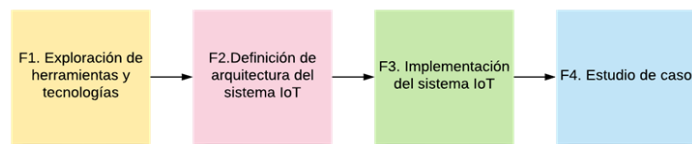


Fig. 1. Fases de la metodología.

capturados son enviados vía *bluetooth* a un sistema *software* que almacena los datos en una base de datos no relacional. En la capa de análisis el sistema *software* se encarga de procesar el histórico de los datos capturados aplicando métodos de estadística descriptiva y modelos predictivos basados en aprendizaje no supervisado. Finalmente, en la capa de visualización se puede hacer un seguimiento sobre los datos capturados en tiempo real y se pueden visualizar los resultados de los diferentes modelos de análisis aplicados al histórico de los datos. De este modo, el sistema IoT propuesto permite extender las funcionalidades de los medidores comerciales, al solventar el problema del almacenamiento limitado y al posibilitar tanto la vinculación de modelos de analítica de datos, como la visualización en tiempo real de los mismos.

El resto del artículo está organizado de la siguiente forma: en la sección 2 se presenta la metodología considerada en el desarrollo propuesto; en la sección 3 se describen un conjunto de trabajos relacionados que se tuvieron en cuenta para el desarrollo de la presente investigación; en la sección 4 se describe la arquitectura y la implementación del sistema IoT diseñado; en la sección 5 se muestra un estudio de caso que permite verificar la utilidad del sistema IoT propuesto; finalmente en la sección 6 se muestran las conclusiones y trabajos futuros derivados de la presente investigación.

2. Metodología considerada

Para el desarrollo del sistema IoT se definieron las siguientes cuatro fases metodológicas: i) caracterización de niveles de intensidad luminosa, ii) definición de la arquitectura del sistema IoT, iii) implementación del sistema IoT de seguimiento y análisis y iv) estudio de caso (ver Fig. 1). En la fase 1 se exploraron y seleccionaron un conjunto de tecnologías a nivel de captura, almacenamiento, análisis y visualización. Dentro de la fase 2 de la metodología se definieron las capas de la arquitectura del sistema, teniendo en cuenta las tecnologías identificadas en la fase 1. En la fase 3, se realizó la implementación del sistema IoT, teniendo en cuenta las capas de la arquitectura diseñadas en la fase 2. Finalmente, en la fase 4 se valida el diseño mediante el desarrollo de un estudio de caso, verificando la funcionalidad del sistema IoT propuesto.

3. Trabajos relacionados

En esta sección se presentan un conjunto de trabajos relacionados que se tuvieron en cuenta para el desarrollo e implementación del sistema IoT.

En [5] se presenta una arquitectura IoT basada en herramientas de *hardware* y *software* libre para el soporte de agricultura de precisión. La propuesta tomó en consideración las capas de la arquitectura Lambda: captura de datos, almacenamiento, procesamiento de datos y consulta. En la capa de captura se hace uso de sensores de temperatura y humedad compatibles con la tarjeta de adquisición de datos Arduino, en la capa de almacenamiento se emplea la base de datos en la nube *Ubidots*, en la capa de análisis se hace uso de un clasificador bayesiano y finalmente en la capa de consulta se hace uso de un aplicativo web desarrollado sobre el *framework Spark* de *Java*. Si bien este trabajo no toma en cuenta dentro de las variables a medir el nivel de intensidad luminosa, se tuvieron en cuenta las capas de la arquitectura consideradas para el diseño del sistema IoT propuesto.

En [13] se presenta un dispositivo electrónico para la adquisición de datos, mediante diferentes tipos de sensores, con el fin de estudiar la evolución de la humedad del suelo y su relación con diferentes parámetros tales como: temperatura ambiental, temperatura al sol, humedad relativa e intensidad de la lluvia. El dispositivo fue construido haciendo uso de la plataforma de *hardware* libre Arduino y sensores compatibles con esta.

Los datos capturados mediante el dispositivo son almacenados en una memoria *flash* y son enviados por puerto serial a un computador, en donde pueden ser visualizados a partir de un *software* de conexión serial. El dispositivo propuesto no permite el seguimiento remoto, ni la aplicación de técnicas de analítica de datos.

En [14] se presenta el diseño e implementación de un sistema de monitoreo en tiempo real de las variables de temperatura y humedad en losas de concreto convencional, haciendo uso para ello de la plataforma de *hardware* libre Arduino y sensores electrónicos compatibles con esta. El sistema construido es útil en el campo de la ingeniería civil de cara a la identificación de la durabilidad de un elemento estructural. El sistema propuesto captura los datos de las variables de temperatura y humedad para luego almacenarlos en una memoria SD articulada a la tarjeta Arduino, de tal modo que luego dichos datos puedan ser analizados. El sistema desarrollado en este artículo no permite la monitorización remota, ni tiene articulado de manera directa la aplicación de técnicas de análisis de datos sobre las capturas realizadas.

En [15] se presenta un sistema para la medición y seguimiento de la intensidad lumínica y la temperatura bajo el mar, realizando mediciones cada metro de profundidad hasta los 20 metros. El sistema está conformado por un total de 20 nodos sensores de luz y temperatura los cuales son gestionados por una tarjeta Arduino, desde donde los datos son enviados a un modem 3G, el cual los transmite a una estación terrestre, donde existe un instrumento virtual desarrollado en *Labview* que permite la visualización en tiempo real de los datos. El sistema desarrollado no evidencia la implementación de la capa de almacenamiento, ni de la capa de análisis de los datos mediante algoritmos de aprendizaje supervisado y no supervisado.

En [10] se propone un sistema de monitorización de humedad del suelo en huertos, el cual incluye la funcionalidad de irrigación automática a través de una electroválvula. El sistema fue implementado a través del uso de la plataforma de *hardware* libre Arduino y el sensor de humedad del suelo compatible con esta. Los datos capturados por los sensores son procesados en la tarjeta Arduino para determinar mediante un algoritmo si el suelo requiere ser irrigado o no. Del mismo modo a medida que los datos son recibidos por la tarjeta Arduino, son enviados

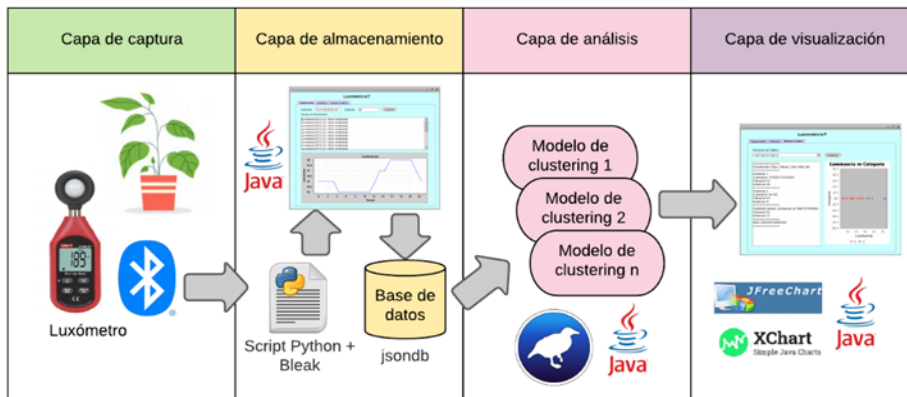


Fig. 2. Arquitectura del Sistema IoT.

mediante el uso de un módulo bluetooth a una aplicación móvil que muestra en tiempo real los datos que van siendo capturados. Aunque el sistema permite el seguimiento remoto, no permite el uso de algoritmos de analítica sobre el histórico de los datos.

En [7] se presenta un sistema para el monitoreo de plantas ornamentales haciendo uso de la plataforma de *hardware* libre Arduino. El sistema de monitoreo permite la obtención de las variables de temperatura y humedad del suelo, las cuales son enviadas por la placa Arduino mediante *bluetooth* a una aplicación móvil que permite la visualización de los resultados en tiempo real. El sistema adicionalmente cuenta con una minibomba de agua, la cual se activa dependiendo de los resultados obtenidos en tiempo real de temperatura y humedad. Aunque este sistema cuenta con diferentes aportes en la capa de captura y visualización, no se evidencian los aportes en cuanto a las funcionalidades de almacenamiento y análisis del histórico de los datos mediante métodos de aprendizaje supervisado y no supervisado.

Los anteriores trabajos permiten evidenciar la importancia de que los sistemas de monitorización basados en IoT tengan en cuenta las capas de captura, almacenamiento, análisis y visualización.

De acuerdo con lo anterior, se observa que varios de los sistemas presentados, incluyen la mayoría de las veces las capas de captura, almacenamiento y visualización, teniendo como gran desafío el enriquecimiento de la capa de análisis. En este sentido en el contexto de la agricultura y el cuidado de las plantas ornamentales las variables más comunes empleadas son temperatura y humedad, sin embargo, se hace necesaria la integración de otras variables de interés tales como la intensidad de la luz, la cual determina también el crecimiento de las plantas de interiores.

4. Sistema IoT propuesto

En esta sección se presenta la descripción de la arquitectura y la implementación del sistema de seguimiento y análisis de la intensidad luminosa en plantas de

Tabla 1. Categorías de luminancia consideradas.

Rango	Categoría
Luminancia<=807	Insuficiente
807<Luminancia<=2152	Bajo
2152<Luminancia<=5382	Medio
5382 <Luminancia<=10764	Alto
Luminancia>10764	Muy Alto

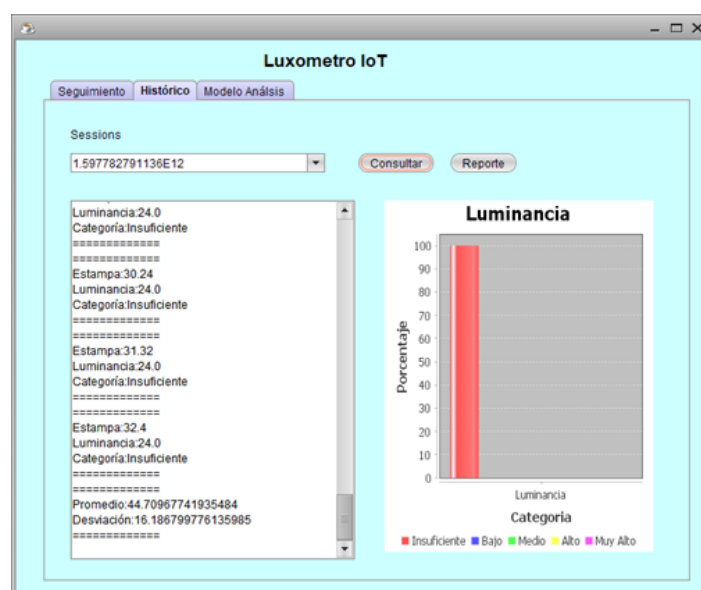


Fig. 3. Pestaña de “Historico”.

interiores. A partir del análisis de la literatura anterior y de la metodología considerada en la Fig. 1, en la Fig. 2 se presenta la arquitectura del sistema IoT propuesto.

En la Fig. 2 se muestra como en la capa de captura de hace uso de un luxómetro comercial IoT UT383 BT, el cual permite la obtención de la intensidad luminosa o luminancia en *luxes* y la transmisión en tiempo real de los datos en un formato de trama propio mediante el protocolo *bluetooth* LE. En la capa de almacenamiento los datos de luminancia son obtenidos de manera asincrónica mediante un *script* de *Python* que hace uso de la librería *Bleak*, la cual posibilita la conexión a los servicios y características de los dispositivos *bluetooth*. El *script* de *Python* es invocado en segundo plano por un aplicativo desarrollado en *Java*, el cual a medida que obtiene los datos, los categoriza en 4 niveles (Insuficiente, Bajo, Medio, Alto), genera una estampa de tiempo y los almacena en la base de datos no relacional *jsondb* mediante un *id* de sesión. A medida que los datos son presentados en pantalla, también son visualizados en función del tiempo mediante una gráfica generada a partir de la librería *XChart* de *Java*.

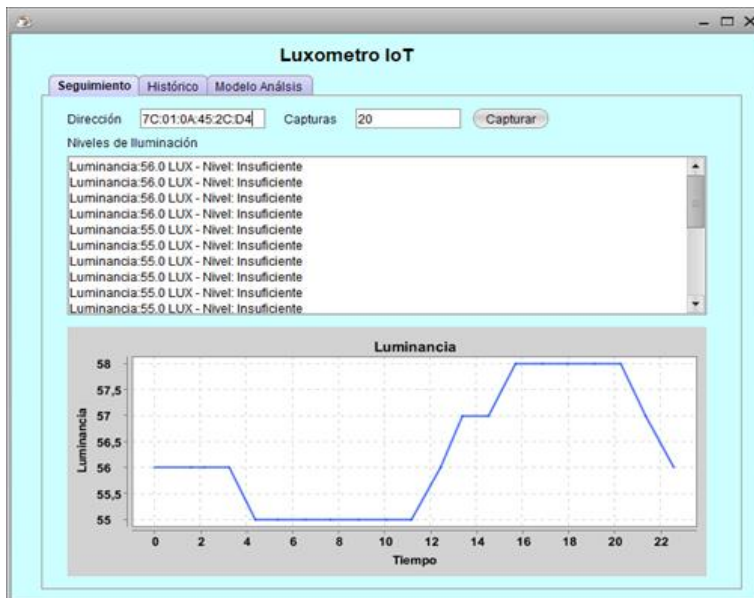


Fig. 4. Pestaña de “Seguimiento”.

Una vez los datos son almacenados, en la capa de análisis es posible realizar cálculos estadísticos básicos (promedio y desviación estándar) sobre los datos de cada sesión de captura, así como aplicar modelos de aprendizaje no supervisado o *clustering* sobre los mismos. De manera específica el sistema IoT implementa mediante el uso de la librería *weka* el algoritmo de agrupamiento *K-means*, el cual permite obtener un número de terminado de centroides alrededor de los cuales se concentran los datos de luminancia capturados. Finalmente, en la capa de visualización son presentados los valores de los *clusters* obtenidos a partir de la aplicación del modelo, así como la gráfica que muestra los datos asociados a cada *cluster*.

Considerando las funcionalidades y tecnologías presentadas en la arquitectura del sistema IoT mostrado en la Fig. 2, en la Fig. 3 se presenta la interfaz principal del sistema IoT para el seguimiento y análisis de la luminancia en plantas de interiores, el cual fue desarrollado en el lenguaje *Java*. La interfaz del sistema está formada por tres pestañas a saber: “Seguimiento”, “Histórico” y “Modelo Análisis”. En la pestaña “Seguimiento”, el usuario del sistema ingresa la dirección *bluetooth* del luxómetro y el número de capturas asincrónicas a obtener a partir de este, de tal modo que al presionar el botón “Capturar”, los datos son obtenidos a partir del luxómetro. Cada captura realizada es categorizada por el sistema, teniendo en cuenta los niveles de luminancia presentados en la Tabla 1 y obtenidos a partir de lo presentado en [9]. Del mismo modo en esta pestaña a medida que los datos son obtenidos se presenta una gráfica que muestra los valores de luminancia en función del tiempo.

Por otra parte, en la Fig. 4 se presenta la pestaña “Histórico”, a través de la cual al presionar el botón “Consultar”, es posible obtener los datos almacenados en cada una de las sesiones de captura, junto con el promedio y la desviación estándar de dichos

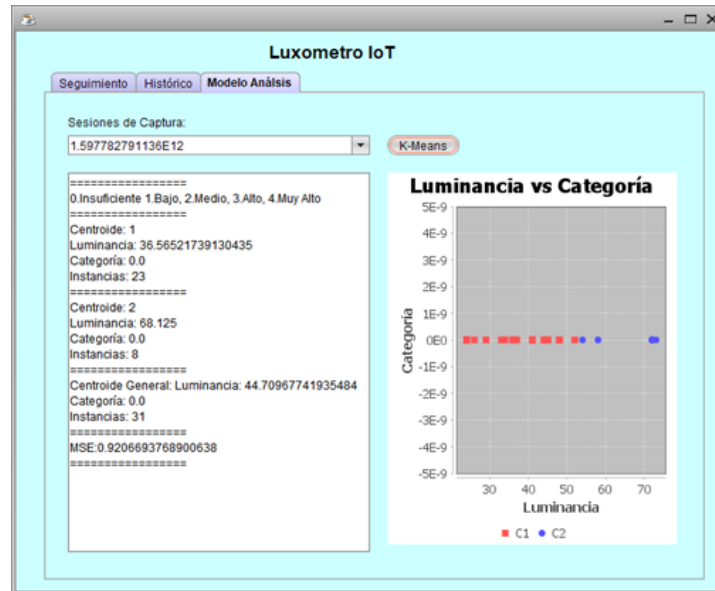


Fig. 5. Pestaña de “Modelo de análisis”.

datos. Del mismo modo, se genera un diagrama de barras con la distribución de cada una de las capturas en los niveles definidos en la Tabla 1.

Finalmente, esta pestaña cuenta con la opción de generar un reporte .csv de cada una de las sesiones de captura almacenadas. La sesión presentada en la Fig. 4 cuenta con un total de 31 capturas (estampa de tiempo, luminancia y categoría), las cuales tienen un promedio de 44.70 *luxes*, una desviación estándar de 16.1876 y han sido categorizados en el nivel de luminancia “Insuficiente”.

En la Fig. 5 se presenta la pestaña “Modelo Análisis”, la cual permite la aplicación de algoritmos de aprendizaje no supervisado o *clustering* a los datos de las diferentes sesiones de captura. En específico el algoritmo utilizado en el sistema IoT propuesto es el de *K-Means*, el cual permite la obtención de un número predefinido de centroides alrededor de los cuales se concentran los datos capturados. Para lo anterior, el modelo fue configurado para trabajar con dos atributos: “luminancia” y “categoría”. Así, en la Fig. 5 se aplica el modelo de *clustering* a la sesión presentada en la Fig. 4, obteniendo dos *clusters* con centroides C1 y C2, donde $C1 = \{\text{luminancia} = 36.5652 \text{ luxes}, \text{categoría} = 0\}$, $C2 = \{\text{luminancia} = 68.125 \text{ luxes}, \text{categoría} = 0\}$, la categoría 0 corresponde al nivel “Insuficiente”. El *cluster* con centroide C1 cuenta con un total de 23 instancias asociadas, mientras que el *cluster* con centroide C2 tiene asociadas un total de 8 instancias, lo que quiere decir que el 74.19% de las capturas están concentrados alrededor del centroide C1 y el 25,81% de las capturas se concentran alrededor del centroide C2. Además de lo anterior, se muestra un centroide general sobre todas las capturas: $C_{\text{general}} = \{\text{luminancia} = 44.71 \text{ luxes}, \text{categoría} = 0\}$. Finalmente, en la pestaña presentada en la Fig. 5 es posible distinguir los centroides y su distribución mediante la gráfica de la derecha, en la cual las capturas del *cluster* 1 se presentan en color azul, mientras que las capturas del *cluster* 2 se muestran en color rojo.

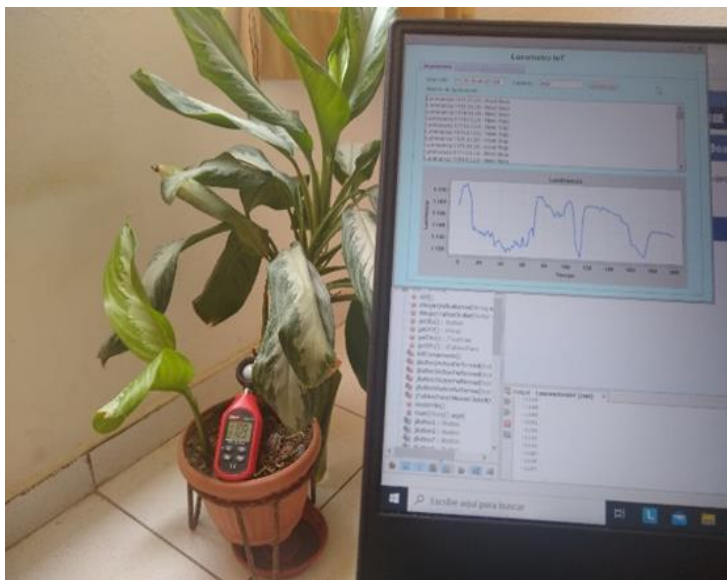


Fig. 6. Estudio de caso.

5. Estudio de caso

Para verificar la funcionalidad del sistema IoT propuesto, se desarrolló un estudio de caso en una zona residencial de la ciudad de Cartagena de Indias Colombia, entre las 12pm y la 1pm, realizando capturas sobre una especie de planta conocida como: dieffenbachia (ver Fig. 6). En total se lograron obtener 365 capturas clasificadas en la categoría de baja luminancia, las cuales tienen un promedio de 1141.51 *luxes* y una desviación estándar de 70.75. Así mismo a partir del reporte generado, se obtuvo como valor mínimo de la luminancia 1081 *luxes* y como valor máximo 1522 *luxes*.

Con respecto a la aplicación del modelo de agrupamiento o *clustering* sobre las 365 capturas realizadas, se obtuvieron dos *clusters* con centroides C1 y C2, donde $C1 = \{\text{luminancia} = 1126.988 \text{ luxes}, \text{ categoría} = 1\}$, $C2 = \{\text{luminancia} = 1421.444 \text{ luxes}, \text{ categoría} = 1\}$, la categoría 1 corresponde al nivel bajo de luminancia de acuerdo a lo presentado en la Tabla 1 (ver Fig. 7). El *cluster* con centroide en C1 cuenta con un total de 347 instancias, mientras que el *cluster* con centroide en C2 tiene un total de 18 instancias, lo que quiere decir que el 95.06% de los datos están concentrados alrededor del centroide C1, mientras que el 4.94% de los datos restantes se concentran alrededor del centroide C2.

La distribución de los datos en los centroides C1 y C2 puede apreciarse de manera más clara en la Fig. 7, en donde las capturas asociadas al cluster 1 aparecen en color rojo, mientras que las capturas en color azul corresponden al cluster 2. Del mismo modo, el centroide general alrededor de los cuales se concentran los datos capturados es: $C_{\text{general}} = \{\text{luminancia} = 1141.51 \text{ luxes}, \text{ categoría} = 1\}$. Así mismo, en la Fig. 7, se puede apreciar que el error cuadrático medio del modelo de clustering aplicado es de 1.767 *luxes*.

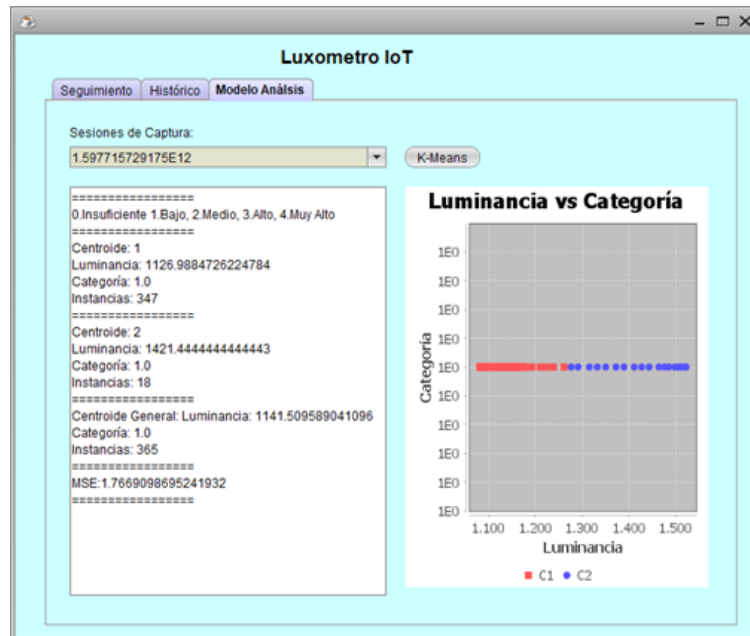


Fig. 7. Modelo de análisis aplicado al estudio de caso.

Finalmente, de acuerdo con lo presentado en [9], para el correcto desarrollo y crecimiento de la *dieffenbachia* se requiere un nivel de luz medio, por lo que según los resultados en el caso de estudio es necesaria la reubicación de la planta a un lugar que cuente con una mayor luminancia.

6. Conclusiones y trabajos futuros

Se propuso un sistema IoT para el seguimiento y análisis de luminancia, el cual está basado en las cuatro capas de la arquitectura convencional de IoT: captura, almacenamiento, análisis y visualización. El sistema propuesto es una alternativa a los problemas de los equipos de medición convencionales, de tal modo que permite solventar el almacenamiento limitado, da soporte a la monitorización remota y posibilita la aplicación de algoritmos de aprendizaje supervisado y no supervisado, centrándose en este caso en los algoritmos de agrupamiento o *clustering*.

Las herramientas y tecnologías empleadas en el desarrollo del sistema demostraron ser adecuadas para dar cumplimiento a los requisitos de cada una de las capas de la arquitectura IoT. En cuanto a la capa de captura se destaca el uso de la librería *Bleak*, la cual permite la conexión a los servicios y características disponibles en los dispositivos *bluetooth*. En la capa de almacenamiento se destaca el uso de la base de datos portable y no relacional *dbjson*, la cual permite el mapeo de objetos a documentos *json*. Finalmente, en la capa de análisis se destaca el uso de la librería *weka*, la cual permite la implementación de algoritmos de aprendizaje supervisado y no supervisado.

El estudio de caso desarrollado permitió verificar la utilidad del sistema propuesto, al posibilitar el seguimiento de la luminancia en la especie de plantas *dieffenbachia*. El estudio realizado demostró que las capturas realizadas se concentran en el nivel de luminosidad baja, mientras que la planta requiere un nivel de luminosidad media.

Como trabajo futuro derivado de esta investigación, se pretende ampliar el desarrollo del sistema IoT propuesto de manera que permita la integración, el seguimiento y análisis de diferentes variables de interés tales como la temperatura, humedad y presión, teniendo en cuenta no solo la arquitectura presentada en este artículo (a cuatro capas), sino también la arquitectura de soluciones comerciales tales como Cayenne IoT.

Referencias

1. Shruti, S., Hedge, G.: Study of IoT: Understanding IoT architecture, applications, Issues and Challenges. *Int. J. Adv. Netw. Appl.*, pp. 477–482 (2016)
2. Conti, M., Dehghantanha, A., Franke, K., Watson, S.: Internet of things security and forensics: Challenges and opportunities, *Futur. Gener. Comput. Syst.*, 78, pp. 544–546 (2018)
3. Firouzi, F., Farahani, B., Weinberger, M., DePace, G., Aliee, F.S.: IoT Fundamentals: Definitions, architectures, challenges, and promises. In: *Intelligent Internet of Things*, Springer International Publishing, pp. 3–50 (2020)
4. Firouzi, F., Farahani, B., Ye, F., Barzegari, M.: Machine Learning for IoT. In: *Intelligent Internet of Things*, Springer International Publishing, pp. 243–313 (2020)
5. Quiroga, E., Jaramillo, S., Campo, W., Chanchí, G.: Proposal of an Architecture for Precision Agriculture Supported in IoT. *RISTI Rev. Ibérica Sist. e Tecnol. Informação*, 24, pp. 39–56 (2017)
6. Gómez, J.E., Marcillo, F.R., Triana, F.L., Gallo, V.T., Oviedo, B.W., Hernández, V.L.: IoT for environmental variables in urban areas. *Procedia Comput. Sci.*, 109, pp. 67–74 (2017)
7. Mendoza, R., Carrillo, I., Juárez, C.: Sistema autónomo de monitoreo y suministro de agua en plantas de ornato. *Tianguistenco* (2016)
8. Melgarejo, P., Navarro, A., Legua, P., Lidón, V.: La iluminación en los invernaderos. Orihuela, Universidad Miguel Hernández (2019)
9. Pennisi, B.: *Growing indoor plants with success* (2009)
10. Guijarro-Rodríguez, A., Cevallos-Torres, L., Preciado-Maila, D., Zambrano-Manzur, B.: Sistema de riego automatizado con Arduino. *Rev. Espac.*, 39(37), pp. 27 (2018)
11. Chanchí, G.E., Ospina, M.A., Pérez, J.L.: Sistema IoT para la monitorización de la variabilidad del ritmo cardíaco en pruebas de usabilidad. *Rev. Espac.*, 41(25) (2020)
12. Chanchí, G., Sierra, L., Campo, W.: Propuesta de una plataforma académica portable para la construcción de microservicios en entornos de IoT. *RISTI Rev. Ibérica Sist. e Tecnol. Informação*, E27, pp. 1–13 (2020)
13. De Pablo, M., De Pablo, C.: Ardudrop 1.0: dispositivo electrónico para el estudio de la humedad del suelo Permasnow: Snow cover effects on the ground thermal regime in Livingston and Deception Islands, Antarctica. View project Permafrost and Climate Change in the Antarctic Peninsula. *Tecnol. y Desarro.*, 7, pp. 3–24 (2010)
14. Velasco-Núñez, E., Abidán, J., Salas, R.: Instrumentación alternativa con Arduino y la reconfiguración de capitales: un caso en la ingeniería civil. *RITI J.*, 7(13), pp. 103–111 (2019)
15. Jiménez, F., Nogueras, M., Artero, C., Del Río, J.: Sistema de medida de intensidad lumínica para columna de agua basada en plataforma Arduino. *Instrum. Viewp.*, pp. 44–48 (2014)

Electronic edition
Available online: <http://www.rcs.cic.ipn.mx>



ISSN: 1870-4069
<http://rcs.cic.ipn.mx>



Centro de Investigación
en Computación