

# EDUCACIÓN

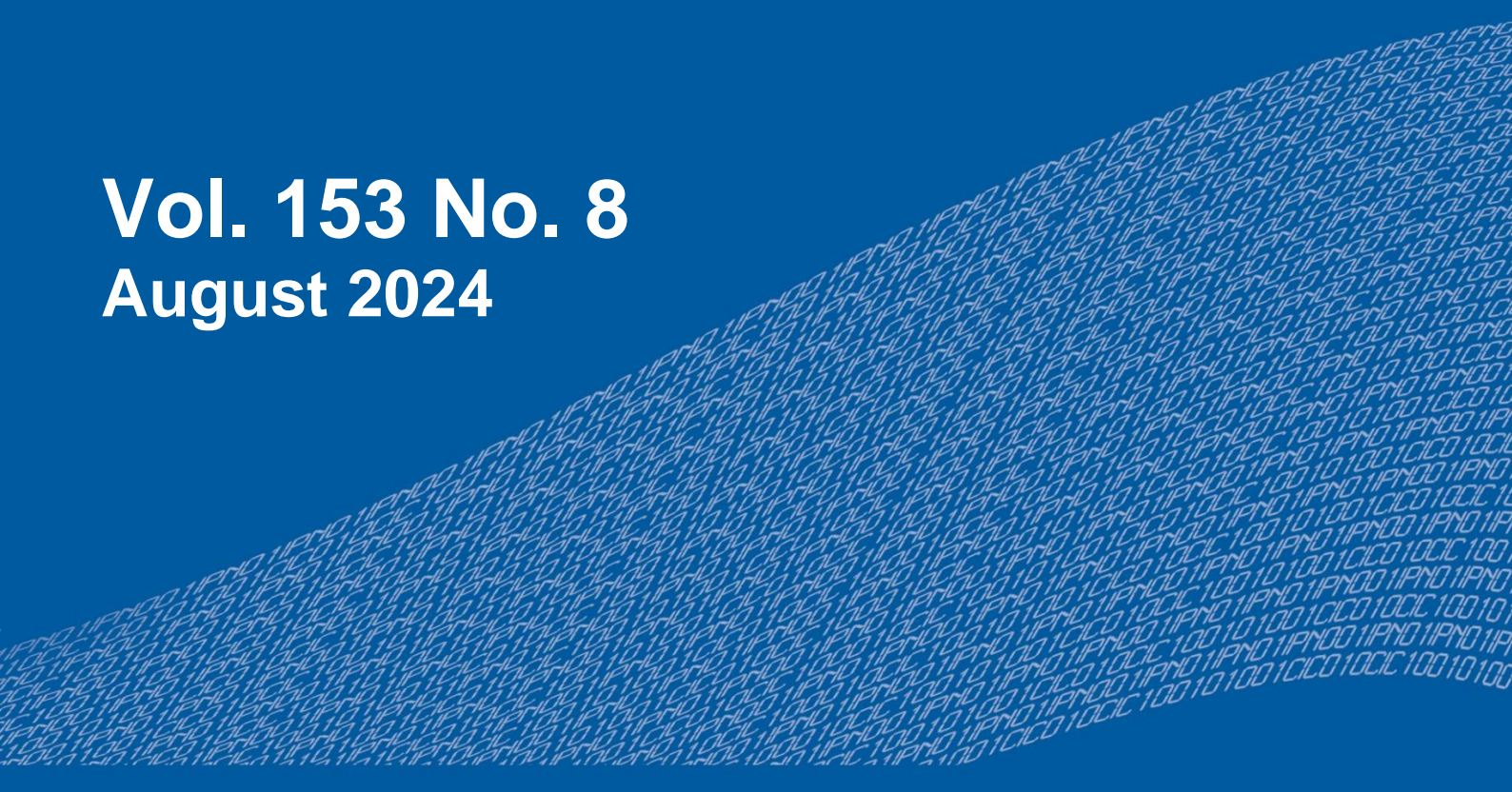
SECRETARÍA DE EDUCACIÓN PÚBLICA



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

# Research in Computing Science

**Vol. 153 No. 8**  
**August 2024**



# Research in Computing Science

---

## Series Editorial Board

### Editors-in-Chief:

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

### Associate Editors:

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

### Editorial Coordination:

Griselda Franco Sánchez

*Research in Computing Science*, Año 23, Volumen 153, No. 8, agosto de 2024, 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 agosto de 2024.

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 23, Volume 153, No. 8, August 2024, 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.

# Advances in Artificial Intelligence

**Roberto A. Vázquez Espinoza de los Monteros**  
(ed.)



Instituto Politécnico Nacional, Centro de Investigación en Computación  
México 2024

## ISSN: in process

---

Copyright © Instituto Politécnico Nacional 2024  
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

## Table of Contents

	Page
Desempeño de algoritmos de inteligencia artificial en la clasificación de objetos astronómicos en Gaia DR3.....	7
<i>Orestes Javier Pérez-Cruz, Cynthia Alejandra Martínez-Pinto, Silvana Guadalupe Navarro-Jiménez, Luis José Corral-Escobedo, Marco Antonio Meza Aguilar</i>	
Implementación de modelos preentrenados de procesamiento de lenguaje natural para la tarea de pregunta-respuesta en un Chatbot para consulta de información sobre notas clínicas de diabetes.....	21
<i>Jonathan Zavala-Díaz, Juan C. Olivares-Rojas, Jennifer Páramo-Mascote, J. Guadalupe Ramos-Díaz, Johan W. González-Murueta</i>	
Detección temprana de degeneración macular asociada con la edad mediante arquitecturas basadas en transformadores de visión: Un estudio comparativo .....	35
<i>Augusto Javier Reyes-Delgado, Jorge Ernesto González-Díaz, José Luis Sánchez-Cervantes, Yara Anahí Jiménez-Nieto, Adolfo Rodríguez-Parada, José Luis Rodríguez-Loaiza</i>	
Análisis comparativo de técnicas de aprendizaje automático para la detección de la enfermedad de Parkinson mediante el reconocimiento de voz .....	49
<i>Justo Francisco León Pastor Baak, Giner Alor Hernández, Nancy Aracely Cruz Ramos, José Luis Sánchez Cervantes, Lisbeth Rodríguez Mazaha</i>	
Modelo secuencial de lógica difusa para la predicción de riesgo de muerte súbita cardíaca en niños con predisposición a SQTl .....	63
<i>Oscar Vélez-Mora, Ulises Juárez-Martínez, Alberto Alfonso Aguilar-Lasserre, Ignacio López-Martínez, Beatriz Alejandra Olivares-Zepahua, Norma Alicia Balderrábano-Saucedo</i>	
Caracterización de las propiedades eléctricas de un vegetal, aplicación en plantas de chile carnaval.....	77
<i>Oscar Josué Barrón López, Juan Prado Olivarez, Juan Carlos Gómez Cortés, José Alfredo Padilla, Alejandro Israel Barranco Gutiérrez, Jessica Alejandra Araujo Rodríguez</i>	

Predicción de expansión urbana de las ciudades Zacatecas-Guadalupe usando máquinas de soporte vectorial .....	89
<i>A. Carmina Llamas-Valenzuela, José I. de la Rosa, G. Moreno-Chávez, Efrén Gonzales-Ramírez, Jesús Villa, José M. Celaya-Padilla</i>	
Identificación de variables causales de la congestión de tráfico mediante la prueba de Granger .....	103
<i>Ernesto De la Cruz-Nicolás, Hugo Estrada-Esquivel, Alicia Martínez-Rebollar, Odette Alejandra Pliego-Martínez</i>	
Clasificación binaria y multiclase de retinopatía diabética con características profundas de imágenes mejoradas en LAB y YCbCr.....	117
<i>Angel Salazar Alvarez, Pedro Arguijo, José Antonio Hiram Vázquez López, David Lara Alabazares</i>	
Análisis y desarrollo de un plugin de moodle con enfoque en aprendizaje basado en retos.....	129
<i>Gerardo Valdés Picos, Ricardo Rafael Quintero Meza</i>	
Diagnóstico de fallas en generadores de potencia utilizando algoritmos de machine learning.....	137
<i>Quetzalli Amaya Sánchez, Alberto Alfonso Aguilar Lasserre, Marco Julio del Moral Argumedo, Gustavo Arroyo Figueroa</i>	
Desarrollo de un modelo inteligente para predecir el precio marginal local (PML) en un nodo de la red eléctrica nacional .....	149
<i>Marcos Fidel Guzmán Escobar, Alberto Alfonso Aguilar Lasserre, Marco Julio Argumedo del Moral, Nicasio Hernández Flores, Gustavo Arroyo Figueroa</i>	
Algoritmos de aprendizaje automático para identificación de patrones SERS de pesticidas organofosforados .....	163
<i>Mateo Frausto-Avila, Monserrat Ochoa-Elias, Gonzalo Ramírez-García, Mario Alan Quiroz-Juarez</i>	
Mejoramiento del agrupamiento de datos mezclados e incompletos mediante algoritmos bioinspirados .....	171
<i>Claudia C. Tusell-Rey, Yenny Villuendas-Rey, Oscar Camacho-Nieto, Viridiana Salinas-García</i>	
Mecanismo de generación pseudoaleatorio para el diseño automático de filtros analógicos pasivos.....	187
<i>Braulio Jesús Montoya Padilla, Aurora Torres Soto, María Dolores Torres Soto</i>	

Modelo de aprendizaje profundo para la inspección de tarjetas de circuitos impresos.....	199
<i>José Antonio Lara Chávez, Carlos Avilés Cruzand, Miguel Magos Rivera</i>	
Traductor inglés-español por reconocimiento de texto.....	213
<i>Miriam Calderón Reyes, Daniela Jiménez Pano, Norma Angélica Márquez Sulca, Luis Carlos Padierna García</i>	
Análisis de ataques de denegación de servicio distribuido (DDoS) en un conjunto de datos usando aprendizaje automático.....	221
<i>Jesus Barranco Castillo, Arturo Zúñiga-López, Carlos Avilés-Cruz, Cesar Benavides-Álvarez</i>	
Sistema de entrenamiento personalizado para equipos de programación competitiva mediante aprendizaje supervisado.....	231
<i>Juventino Aguilar-Correa, Aldonso Becerra-Sánchez, Roberto Solís-Robles, Gustavo Zepeda-Valles</i>	
Detección de micro fraudes en transacciones de tarjetas de crédito utilizando aprendizaje automático: Generación de la base de datos y métodos de detección.....	247
<i>Jaime Alberto Quiñones-Beltran, Karina M. Figueroa-Mora, Juan Pablo Maldonado-López</i>	
Técnicas de aprendizaje automático para agrupar planetas similares a la Tierra.....	259
<i>Marcos Macías-Juárez, Edgar Moyotl-Hernández</i>	
Revisión de los criterios y análisis de diseño de GUI desde una perspectiva de genero.....	273
<i>Paulo Cesar Portilla-Tirado, Juan Villegas-Cortez, Marco Vinicio Ferruzca-Navarro, Román Anselmo Mora-Gutiérrez</i>	





## Desempeño de algoritmos de inteligencia artificial en la clasificación de objetos astronómicos en Gaia DR3

Orestes Javier Pérez-Cruz<sup>1</sup>, Cynthia Alejandra Martínez-Pinto<sup>1</sup>,  
Silvana Guadalupe Navarro-Jiménez<sup>2</sup>, Luis José Corral-Escobedo<sup>2</sup>,  
Marco Antonio Meza Aguilar<sup>1</sup>

<sup>1</sup> Instituto Tecnológico de Ciudad Guzmán,  
Tecnológico Nacional de México,  
México

<sup>2</sup> Universidad de Guadalajara,  
Instituto de Astronomía y Meteorología,  
México

{orestesperez1995, silvananj}@gmail.com,  
cynthia\_amp@hotmail.com, luis.corral@academicos.udg.mx,  
marco.ma@cdguzman.tecnm.mx

**Resumen.** Se realiza un análisis del desempeño de una serie de algoritmos de Machine Learning (ML) para la clasificación de objetos astronómicos utilizando los datos del catálogo DR3 de la misión espacial Gaia. Los modelos de aprendizaje automático fueron entrenados con la información espectral proveniente de los espectrofotómetros rojo y azul del satélite. El propósito es lograr una clasificación precisa de los siguientes tipos de objetos: Estrellas Simbióticas, Nebulosas Planetarias y Gigantes Rojas. Se evalúan diferentes algoritmos de clasificación, incluyendo Random Forest (RF), Máquina de Soporte Vectorial (SVM), Redes Neuronales Artificiales (RNA) y Gradient Boosting. Se comparan los resultados obtenidos usando diversas métricas (Precision, Recall, F1-Score, el índice Kappa) y se comprueba la efectividad de clasificar las estrellas antes mencionadas, utilizando solamente la información de sus espectros. Los modelos que obtuvieron los mejores resultados fueron los entrenados con Redes Neuronales Artificiales y Random Forest, con un porcentaje de precisión superior al 94.67%.

**Palabras claves:** Clasificación automática, machine learning, Gaia DR3, espectroscopia, objetos astronómicos, nebulosas planetarias, estrellas simbióticas, gigantes rojos.

### Performance of Artificial Intelligence Algorithms in the Classification of Astronomical Objects in Gaia DR3

**Abstract.** An analysis is conducted on the performance of a series of Machine Learning (ML) algorithms for the classification of astronomical objects using

data from the DR3 catalog of the Gaia space mission. The machine learning models were trained using spectral information from the satellite's red and blue spectrophotometers. The purpose is to achieve accurate classification of the following types of objects: Symbiotic Stars, Planetary Nebulae, and Red Giants. Different classification algorithms, including Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Gradient Boosting, are evaluated. The results are compared using various metrics such as Precision, Recall, F1-Score, and the Kappa index. The effectiveness of classifying the stars using only their spectral information is demonstrated. The models trained with Artificial Neural Networks and Random Forest achieved the best results, with a precision percentage exceeding 94.67%.

**Keywords:** Automatic classification, machine learning, Gaia DR3, spectroscopy, astronomical objects, planetary nebulae, symbiotic stars, red giants.

## 1. Introducción

El catálogo Gaia DR3 (GDR3), liberado el 13 de junio de 2022, incluye por primera vez, los espectros calibrados de los objetos astronómicos: uno que cubre la región azul del espectro electromagnético (BP) abarcando longitudes de onda de 330 a 680 nm, y otro en la región roja (RP) que cubre el rango de 640 a 1050 nm [1]. Los espectros medios BP/RP observados son de baja resolución, llegando a magnitudes de  $G < 17.65$  [2].

Dentro del catálogo, existen estrellas con características peculiares que resultan difíciles de clasificar utilizando los métodos convencionales. Además, el enorme volumen de información que un telescopio moderno puede generar impide a los astrónomos procesar dichos datos de manera individual. Por lo tanto, la clasificación automática es un imperativo en los tiempos actuales, donde se manejan una gran cantidad de datos. Gaia, en su catálogo DR3, liberó alrededor de 470 millones de fuentes con parámetros astrofísicos de espectros BP/RP [3].

La presente investigación se enfoca en tres tipos de objetos astronómicos: estrellas simbióticas, nebulosas planetarias y gigantes rojas. Estas estrellas en particular son de interés debido a que provienen de la evolución de cuerpos celestes de baja masa, y debido a la similitud de sus espectros, provoca que, en ocasiones, se clasifiquen erróneamente.

- **Gigante Roja (RG).** Son estrellas de masa baja e intermedia (entre 0.8 y 8  $M_{\odot}$ ) que han evolucionado. A medida que envejecen, van agotando el Hidrógeno que sostiene la fusión nuclear que ocurre en el centro de la estrella. Esto hace que el núcleo se contraiga, aumentando la temperatura y la presión, provocando la expansión de las capas exteriores. Este proceso culminará en la formación de una enana blanca, que representa la etapa final del ciclo de vida de estas estrellas. Al liberar una gran cantidad de elementos producto de la fusión nuclear al medio interestelar, las enanas blancas desempeñan un papel fundamental en la evolución química de las galaxias en las que se encuentran [4].

- **Estrellas simbióticas (SS)**. Son sistemas estelares compuestos por dos estrellas separadas que orbitan entre sí. Estos sistemas están constituidos por una gigante roja evolucionada (tipo espectral K o M) que pierde y transfiere masa a su segundo componente. Por lo general, esta segunda componente es una enana blanca, caracterizada por su alta temperatura y emisión de una gran cantidad de fotones ionizantes. Se habla de simbiosis estelar debido a que cada una de las estrellas depende e influye en la evolución de la otra. La comprensión de los procesos de transferencia de masa y de acreción en estos sistemas es importante para entender la evolución de las estrellas y cualquier interacción binaria que involucre a gigantes evolucionados [5].
- **Nebulosa Planetaria (PN)**. Es una nube circunestelar ionizada en expansión que fue expulsada durante la fase de la rama asintótica gigante (GRB) de su estrella progenitora, una estrella por debajo de 8 o 9 masas solares [6]. De la estrella queda un residuo en forma de enana blanca, el cual se encuentra a una gran temperatura. Estas nebulosas, en general, forman anillos o burbujas, pero, debido a las características del material circundante o al carácter binario del progenitor (como en el caso de las simbióticas), pueden ser también elipsoidales, bipolares o hasta cuadrupolares [7].

De los objetos astronómicos antes mencionados, las PNs y SS son difíciles de distinguir entre ellas, debido a las características que poseen y su baja representación con respecto a otros objetos celestes. Estas estrellas se caracterizan por presentar intensas líneas de emisión en su espectro visible.

Debido al volumen de estos objetos, se requiere el uso de algoritmos de aprendizaje automático (ML) para su clasificación automatizada. Después de examinar investigaciones anteriores, se decidió emplear los modelos que han demostrado los mejores resultados, como: Random Forest (RF), Máquina de Soporte Vectorial (SVM), Redes Neuronales Artificiales (RNA) y Gradient Boosting. Estos modelos se entrenaron utilizando la información espectral del catálogo Gaia DR3.

## **2. Metodología**

### **2.1. Adquisición y tratamiento de los datos**

En este estudio, se empleó la información suministrada por SIMBAD<sup>1</sup> para identificar las estrellas analizadas. SIMBAD se destaca por su capacidad de ofrecer información detallada sobre objetos astronómicos presentes en artículos científicos [8].

Posteriormente se procedió a realizar una búsqueda cruzada en la tabla *xp\_continuous\_mean\_spectrum* de la base de datos de Gaia DR3 para determinar los objetos astronómicos por tipo de estrella. Esta tabla proporciona la media de los espectros BP y RP basados en una representación continua en funciones base [9].

Para evitar posibles pérdidas de información al muestrear los espectros, los espectros calibrados se encuentran representados como una combinación lineal de funciones base en lugar de utilizar la convencional tabla de flujos y longitudes de onda [10].

---

<sup>1</sup> SIMBAD. Base de datos dinámica de objetos astronómicos (<https://simbad.cds.unistra.fr/simbad/>)

Los espectros descargados originalmente se encontraban internamente calibrados en los rangos de longitud de onda BP y RP. Para su procesamiento, se utilizó la librería GaiaXPy<sup>2</sup>, la cual permitió calibrar y muestrear cada espectro en una cuadrícula de longitud de onda uniforme predeterminada mediante la rutina *calibrate*. De esta manera, se obtuvo un único espectro que abarcaba el rango completo de longitud de onda cubierto por BP y RP. El proceso de calibración anterior generó un total de 343 valores de flujo por espectro. Se utilizó un muestreo predeterminado que abarcó un rango de longitud de onda de 336 a 1020 nm, con un incremento de 2 nm entre cada punto de muestreo.

En total, se obtuvieron los siguientes recuentos de espectros por tipo de estrellas del catálogo Gaia DR3, disponible en el sitio Gaia Archive<sup>3</sup>: 201 espectros correspondientes a estrellas Simbióticas, 574 a Nebulosas Planetarias y 69,146 a Gigantes Rojas. Es notable que el número de espectros de Gigantes Rojas es significativamente superior en comparación con los otros tipos de estrellas, lo cual se debe a su mayor presencia en el universo. Con el fin de obtener una distribución más uniforme, se seleccionó solamente una muestra de 1200 espectros de Gigantes Rojas para su inclusión en este estudio.

## 2.2. Preprocesamiento de los datos

Con el objetivo de mejorar el rendimiento y la estabilidad de los algoritmos de aprendizaje automático durante el entrenamiento y la inferencia, se normalizaron los valores de flujo de cada espectro en una escala de 0 a 1. De esta manera, todos los valores del espectro quedaron expresados como una fracción del valor máximo, lo que permitió establecer una escala común entre los distintos espectros (Fig. 1).

El proceso de normalización permite, aún a simple vista, distinguir mejor entre los diferentes tipos de espectros. El espectro de la PN consta casi exclusivamente de líneas de emisión, en cambio las RG poseen principalmente un continuo con gran número de líneas y/o bandas de absorción. Por su parte las SS resultan una combinación de ambos tipos de espectros, mostrando líneas en emisión y bandas en absorción en la región IR del espectro (700 a 1000 nm).

Posterior a la normalización, se conformó el conjunto de datos que se utilizaría como entrada para los algoritmos de ML. Este conjunto de datos consta de 1,975 registros que representan los espectros de las estrellas de interés. Cada registro está compuesto por 343 características, que corresponden a los valores de flujo normalizados en el rango de longitud de onda de 336 a 1020 nm. Además, se incluyó una columna adicional en el conjunto de datos con la etiqueta correspondiente al tipo de estrella.

Como se puede evidenciar los datos presentan un desbalanceo notable, debido a que existe una diferencia significativa en la cantidad de muestras por cada tipo de estrella. El desbalanceo de datos puede tener un impacto negativo en el rendimiento de los algoritmos de aprendizaje automático, debido a que pueden presentar dificultades para

---

<sup>2</sup> GaiaXPy: Biblioteca Python para el análisis de datos astronómicos de la misión Gaia (<https://gaia-dpci.github.io/GaiaXPy-website/>)

<sup>3</sup> Gaia Archive: Repositorio en línea de datos astronómicos de la misión espacial Gaia (<https://gea.esac.esa.int/archive/>)

**Tabla 1.** Número de estrellas que conforman los conjuntos de datos desbalanceados y balanceados respectivamente.

Estrellas	Conjunto de Datos	
	Datos desbalanceados	Datos balanceados
Estrellas Simbióticas	201	1000
Nebulosas Planetarias	574	1000
Gigantes Rojas	1200	1000

**Tabla 2.** Descripción de los parámetros utilizados en el entrenamiento del algoritmo Random Forest.

Parámetro	Selección
Número de árboles	[10, 500]
Profundidad máxima	Sin restricción
Criterio	gini (impureza de Gini)
Máximo de características	Total de características
Número máximo de hojas	Sin límite

**Tabla 3.** Descripción de los parámetros utilizados en el entrenamiento del algoritmo SVM.

Parámetro	Selección
Kernel	['linear', 'poly', 'rbf']
C (Parámetro de regularización)	1.0
Grado usando el kernel polinómico	3

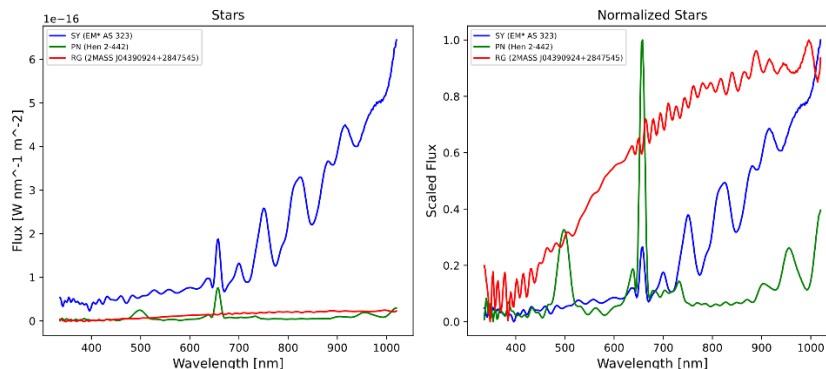
aprender patrones y tomar decisiones precisas para las clases minoritarias. Esta afirmación se pudo comprobar en la sección de resultados del estudio.

En consecuencia, se llevó a cabo la creación de un conjunto de datos balanceado, garantizando que cada tipo de estrella estuviera representado por 1,000 muestras.

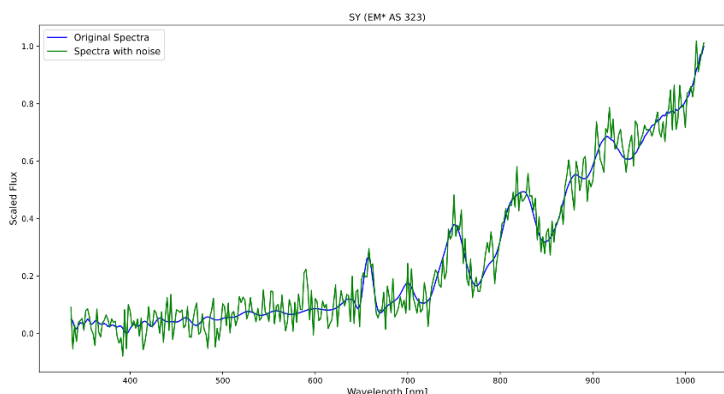
En el caso de las Gigantes Rojas, no se encontraron dificultades debido a que la cantidad recuperada superaba esta cifra. Por lo tanto, se seleccionaron muestras aleatoriamente hasta obtener la cantidad deseada.

Sin embargo, en el caso de las estrellas simbióticas y las nebulosas planetarias, la cantidad de muestras era insuficiente, por lo que se decidió generar nuevos espectros a partir de los originales. Para lograr esto, se empleó el método de adición de ruido blanco, en el cual se generó una secuencia de números aleatorios siguiendo una distribución normal con media 0 y desviación estándar variable, en este caso, en el rango de 0.01 a 0.05. El proceso de generación de nuevos espectros implicó combinar los datos originales con el ruido blanco generado (Ver Fig. 2).

Este nuevo conjunto de datos balanceados, al igual que el anterior estaría conformado por 343 características que representan los valores de flujo y la columna adicional que representa la etiqueta del espectro. En este caso, se logró alcanzar una cantidad final de 3,000 espectros en total, distribuidos equitativamente con 1,000 espectros por cada tipo de estrella. Este conjunto de datos equilibrado garantiza que cada tipo de estrella esté representado de manera adecuada y proporcional en el conjunto. Sin embargo, es importante tener precaución al interpretar los resultados



**Fig. 1.** Las gráficas corresponden a tres tipos de estrellas diferentes, (*azul*) Estrella simbiótica [EM\* AS 323], (*rojo*) Gigante Roja [2MASS J04390924+2847545], (*verde*) Nebulosa Planetaria [Hen 2-442]. En la primera gráfica los valores de flujo están expresados en Vatios por nanómetro por metro cuadrado ( $W/nm/m^2$ ), resultado de calibrarlos externamente usando la librería GaiaXpy. En la gráfica de la derecha los valores de flujo se encuentran escalados en un rango de 0 a 1.



**Fig. 2.** En la gráfica se muestra al espectro de la estrella simbiótica original (*azul*). Y al espectro resultado de la adición de ruido (*verde*), siguiendo una distribución normal con media 0 y desviación estándar 0.05.

obtenidos mediante el sobremuestreo. Los ejemplos sintéticos generados pueden introducir sesgos y afectar la capacidad del modelo de generalización en nuevos datos [11]. La Tabla 1 muestra como quedaron conformados los conjuntos de datos resultantes.

### 2.3. Análisis y selección de los algoritmos

Cada uno de los dos conjuntos de datos creados fue dividido en dos subconjuntos. El primer subconjunto, con una representación del 80% del total de muestras, se utilizó para entrenar los diferentes algoritmos de ML. El otro subconjunto, que representaba

el 20% restante, se reservó para las pruebas. Este conjunto de datos se empleó exclusivamente para la evaluación, con el objetivo de determinar si los algoritmos fueron capaces de aprender y generalizar adecuadamente sin llegar a un sobreajuste.

Para el análisis se utilizaron los siguientes algoritmos supervisados de ML para clasificación: Random Forest, Máquina de Soporte Vectorial, Redes Neuronales Artificiales y Gradient Boosting. La elección de estos algoritmos proporciona una diversa combinación de enfoques de clasificación, lo que permite evaluar su rendimiento y comparar sus resultados en el conjunto de pruebas, para determinar cuál de ellos se adapta mejor a nuestro problema.

### **Random Forest**

Random Forest es un algoritmo de aprendizaje automático supervisado que combina múltiples árboles predictores. Cada árbol se construye basándose en un vector aleatorio muestreado de forma independiente y con la misma distribución para todos los árboles del bosque [12]. Los árboles de decisión tienden a sobreajustarse, lo que significa que aprenden con precisión los datos de entrenamiento, pero tienen dificultades para aplicar ese conocimiento a nuevos datos.

La capacidad de generalización del algoritmo puede mejorarse al combinar múltiples árboles en un conjunto, utilizando la técnica conocida como *ensemble*. Sin embargo, al aplicar Random Forest, es importante tener en cuenta que puede aumentar la complejidad de la interpretación del modelo. Además, este algoritmo es sensible a datos altamente correlacionados, lo cual puede disminuir su capacidad de generalización.

Se realizaron varias pruebas de entrenamiento, cambiando los parámetros que recibía el algoritmo en cada una (ver Tabla 1.).

### **Máquina de soporte vectorial**

La Máquina de Soporte Vectorial (SVM) es un algoritmo de aprendizaje automático supervisado que se utiliza principalmente para la clasificación de datos. En lugar de trabajar directamente en los datos originales, la SVM los representa como puntos en un espacio de múltiples dimensiones, lo que facilita la visualización y el análisis de las relaciones entre las variables. Sin embargo, la interpretación de las decisiones de clasificación es limitada, ya que se enfoca en encontrar un hiperplano óptimo de separación [13]. La comprensión completa del modelo y las relaciones entre las características se dificulta debido a que la atención se centra en los vectores de soporte, que representan solo una pequeña fracción del conjunto de datos total.

Se realizaron pruebas con diferentes parámetros, utilizando distintos *kernels* en cada uno. En la Tabla 3 se muestran las configuraciones analizadas.

### **Redes Neuronales Artificiales**

Las redes neuronales artificiales (ANN) son un subconjunto de herramientas de aprendizaje automático y forman parte de los algoritmos de aprendizaje profundo (deep learning). Su nombre y estructura están inspirados en el cerebro humano, imitando la forma en que las neuronas biológicas envían señales entre sí. Poseen altas velocidades de procesamiento y la capacidad de aprender la solución a un problema a partir de un conjunto de ejemplo [14]. Sin embargo, es importante considerar las limitaciones en la interpretación de las redes neuronales artificiales. Estas limitaciones abarcan su

**Tabla 4.** Descripción de los parámetros utilizados en el entrenamiento del algoritmo Redes Neuronales Artificiales.

Parámetro	Selección
Capas	(343 - 64 - 32 - 32 - 3)
Función de activación	ReLU
Función de pérdida	sparse_categorical_crossentropy
Optimizador	Adam

**Tabla 5.** Descripción de los parámetros utilizados en el entrenamiento del algoritmo Gradient Boosting.

Parámetro	Selección
Función de pérdida	('log_loss', 'deviance', 'exponential')
Cantidad de estimadores	[100-500]
Taza de aprendizaje	[0.1-0.5]
Profundidad máxima	3
Cantidad de características	Total de características

naturaleza de “caja negra”, su sensibilidad a los datos de entrada, el riesgo de sobreajuste y la dificultad de interpretar características abstractas aprendidas.

La red neuronal diseñada tiene la siguiente topología: una capa de entrada de 343 neuronas, seguida de una capa oculta de 64 neuronas y dos capas ocultas de 32 neuronas cada una. Todas las capas son densas, lo que significa que todas las neuronas están completamente conectadas, además utilizan la función de activación *ReLU* para introducir no linealidad en los datos. Después de cada capa densa, se añade una capa *Dropout*, que desactiva aleatoriamente el 10% de las neuronas durante el entrenamiento. Esto ayuda a prevenir el sobreajuste (*overfitting*) y mejora la capacidad de generalización del modelo. La capa de salida consta de 3 neuronas y utiliza la función de activación *softmax*, comúnmente empleada en problemas de clasificación multiclase. En la tabla 4 se muestra la configuración de la red neuronal.

### Gradient Boosting

Gradient Boosting es un algoritmo que se enfoca en la optimización numérica del espacio de funciones en lugar del espacio de parámetros. Trabaja de manera iterativa, donde en cada etapa se añade un nuevo componente a la aproximación existente, ajustándolo en función del gradiente de la función de pérdida. Esto permite mejorar gradualmente la aproximación y obtener mejores resultados tanto en problemas de regresión como de clasificación [15]. Sin embargo, este algoritmo puede presentar limitaciones en términos de interpretabilidad. Estas incluyen la complejidad del modelo, la captura de relaciones no lineales, la dificultad para identificar interacciones entre variables y las características de alta dimensionalidad.

Se probaron diferentes combinaciones de parámetros, utilizando distintas funciones de pérdida, diferentes valores de tasa de aprendizaje, entre otros, quedando las siguientes configuraciones (Ver Tabla 5).



**Matriz de Confusión [Random Forest]**

Datos desbalanceados	Datos balanceados
$\begin{bmatrix} 30 & 6 & 4 \\ 4 & 97 & 14 \\ 0 & 5 & 235 \end{bmatrix}$	$\begin{bmatrix} 191 & 5 & 4 \\ 5 & 187 & 8 \\ 4 & 6 & 190 \end{bmatrix}$

**Fig. 3.** Matriz de confusión resultado de la ejecución del algoritmo Random Forest.**Tabla 6.** Métricas obtenidas del algoritmo de Random Forest.

Métricas	Datos desbalanceados	Datos balanceados
Precision	0.9031	0.9467
Recall	0.8575	0.9467
F1-Score	0.8780	0.9467
Kappa	0.8401	0.9200

### 3. Resultados

#### 3.1. Definición de métricas

Para comparar la precisión de los algoritmos presentados anteriormente, se utilizaron las siguientes métricas: Precisión (Precision), Puntuación F1 (F1-score), Recall (Sensibilidad) y el coeficiente Kappa de Cohen. Estas métricas se calcularon luego de que los algoritmos evaluaran el conjunto de datos de prueba utilizando la matriz de confusión. Esta matriz muestra el número de verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN) para cada clase.

F1-score es una medida que combina la Precisión y el Recall en un solo valor. Proporciona una medida equilibrada entre la precisión y la capacidad de recuperación del clasificador [16]. Es especialmente útil cuando el conjunto de datos está desequilibrado en términos de clases. La fórmula es la siguiente:

$$F1 - Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

El coeficiente Kappa de Cohen, es una medida que expresa el nivel de acuerdo entre dos anotadores en un problema de clasificación [17]. Se define de la siguiente manera:

$$\kappa = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)}$$

En nuestro estudio de clasificación multiclase, se adaptaron las métricas de evaluación utilizadas en problemas de clasificación binaria para su aplicación en un contexto multiclase. Para evaluar el rendimiento de nuestro modelo de clasificación se utilizó la técnica *macro-averaged* [18].

**Matriz de Confusión [SVM]**

Datos desbalanceados	Datos balanceados
$\begin{bmatrix} 22 & 7 & 11 \\ 4 & 93 & 18 \\ 3 & 1 & 236 \end{bmatrix}$	$\begin{bmatrix} 189 & 9 & 2 \\ 13 & 180 & 7 \\ 3 & 3 & 194 \end{bmatrix}$

**Fig. 4.** Matriz de confusión resultado de la ejecución del algoritmo SVM.

**Tabla 7.** Métricas obtenidas del algoritmo SVM.

Métricas	Datos desbalanceados	Datos balanceados
Precision	0.8567	0.9384
Recall	0.7807	0.9383
F1-Score	0.8111	0.9382
Kappa	0.7818	0.9075

La elección de este enfoque se basó en la necesidad de tratar todas las clases de manera equitativa durante la evaluación, independientemente de su tamaño o distribución de los datos. En primer lugar, se calcularon las métricas de manera binaria para cada clase de forma individual, y posteriormente se obtuvo el promedio de estas métricas para obtener una evaluación global del modelo.

**3.2. Modelos**

**Random Forest**

La Fig. muestra la matriz de confusión generada al ejecutar el algoritmo Random Forest. Se observa que la clasificación de las estrellas simbióticas y las nebulosas planetarias tuvieron una mayor imprecisión. Sin embargo, en el conjunto de datos balanceados, a pesar de la presencia de errores, estos son menores, en comparación con el conjunto de datos desbalanceados. Esto se refleja en el valor de Recall, que mejoró de 0.8575 a 0.9467, indicando una mayor capacidad del modelo para identificar correctamente las estrellas. La Tabla 6 muestra los valores obtenidos de todas las métricas evaluadas usando el algoritmo Random Forest.

**Máquina de Soporte Vectorial**

El algoritmo SVM mostró un rendimiento deficiente en las pruebas realizadas con el conjunto de datos desbalanceados. En este escenario, el modelo tendió a sobreajustarse y no logró generalizar correctamente, lo que resultó en una clasificación incorrecta del 12 % del total de muestras usando el conjunto de datos desbalanceados (ver Fig. 4). No obstante, al utilizar el conjunto de datos balanceados, se observó una mejora significativa en el rendimiento, ya que se redujo considerablemente el error de clasificación (6.16%), pasando de un valor de sensibilidad de 0.7807 a 0.9383. La Tabla 7 muestra los valores obtenidos de todas las métricas evaluadas usando el algoritmo SVM.

**Matriz de Confusión [Gradient Boosting]**

Datos desbalanceados	Datos balanceados
$\begin{bmatrix} 31 & 3 & 6 \\ 6 & 93 & 16 \\ 0 & 7 & 233 \end{bmatrix}$	$\begin{bmatrix} 186 & 9 & 5 \\ 8 & 180 & 12 \\ 1 & 8 & 191 \end{bmatrix}$

**Fig. 5.** Matriz de confusión resultado de la ejecución del algoritmo Gradient Boosting.

**Tabla 8.** Métricas obtenidas del algoritmo Gradient Boosting.

Métricas	Datos desbalanceados	Datos balanceados
Precision	0.8848	0.9286
Recall	0.8515	0.9283
F1-Score	0.8666	0.9283
Kappa	0.8158	0.8925

**Gradient Boosting**

El algoritmo Gradient Boosting demostró un rendimiento similar al obtenido por Random Forest en nuestro estudio. Sin embargo, existe la presencia de falsos positivos, debido al desequilibrio presente en los datos de entrenamiento. No obstante, se logra obtener mejoras significativas al emplear el conjunto de datos balanceado (Ver Fig. 5). El valor de F1-Score aumentó de 0.8666 a 0.9283. Este incremento indica una mayor capacidad del algoritmo para identificar correctamente las estrellas en cuestión, reduciendo así los falsos positivos. La Tabla 8 muestra los valores obtenidos de todas las métricas evaluadas usando el algoritmo Gradient Boosting.

**Redes Neuronales Artificiales**

El algoritmo de Redes Neuronales Artificiales mostró resultados superiores en comparación con los algoritmos anteriores, logrando los valores más altos de precisión en general. Esto se evidencia en la matriz de confusión mostrada en la Fig. 6, donde se observa una disminución en el número de falsos positivos tanto en el conjunto de datos balanceados como en los desbalanceados. El valor de F1-Score respalda esta afirmación, con un valor de 0.9012 y 0.9533 para el conjunto de datos desbalanceados y balanceados respectivamente (Ver Tabla 9). Su capacidad para capturar relaciones complejas entre los datos y adaptarse a diferentes patrones lo posiciona como una opción favorable, especialmente en casos de conjuntos de datos desbalanceados.

**4. Conclusiones**

El estudio se centró en la clasificación de los siguientes objetos astronómicos: Estrellas Simbióticas, Nebulosas Planetarias y Gigantes Rojas. Se observó que las Gigantes Rojas eran más numerosas y representaban una proporción mayor en comparación con las restantes clases. Esta desigualdad generó dificultades en la clasificación, ya que los modelos tendieron a confundir parte de la clase minoritaria

<b>Matriz de Confusión [RNA]</b>					
Datos desbalanceados			Datos balanceados		
33	4	3	197	2	1
2	97	16	5	186	9
1	1	238	1	10	189

**Fig. 6.** Matriz de confusión resultado de la ejecución del algoritmo RNA.

**Tabla 9.** Métricas obtenidas del algoritmo RNA.

Métricas	Datos desbalanceados	Datos balanceados
Precision	0.9312	0.9532
Recall	0.8867	0.9533
F1-Score	0.9067	0.9532
Kappa	0.8686	0.9300

(Estrellas Simbióticas y Nebulosas Planetarias) con la clase mayoritaria (Gigantes Rojas).

Se pudo comprobar que los algoritmos Random Forest y Redes Neuronales Artificiales mostraron resultados satisfactorios, con valores de precisión de 0.9467 y 0.9532, respectivamente, para el conjunto de datos balanceado. Por otro lado, en el conjunto de datos desbalanceado, Random Forest alcanzó una precisión de 0.9031, mientras que la RNA alcanzó una precisión de 0.9312. Por lo tanto, estos algoritmos demuestran una mayor efectividad en conjuntos de datos desbalanceados en comparación con otros enfoques.

Las Redes Neuronales Artificiales (RNA) demostraron ser altamente efectivas al trabajar con conjuntos de datos desbalanceados debido a su capacidad para aprender patrones complejos y adaptarse a diferentes distribuciones de clases.

Estos modelos de clasificación se presentan como una herramienta valiosa y de gran utilidad para los astrónomos, al brindarles un apoyo efectivo en la clasificación de estrellas peculiares. Su aplicación contribuye significativamente a una mejor comprensión del ciclo de vida de las estrellas.

## 5. Trabajo futuro

Como trabajo futuro, se sugiere realizar pruebas y evaluaciones de estos modelos de clasificación utilizando conjuntos de datos de estrellas candidatas a Estrellas Simbióticas y Nebulosas Planetarias. Esto permitiría comprobar la capacidad de los modelos para identificar y clasificar de manera precisa estos tipos de estrellas.

Además, se propone la expansión de los modelos para incluir y clasificar una mayor variedad de tipos de estrellas, como: Estrellas Be, Variables Cataclísmicas, Estrellas Mira, Estrellas AeBe, Estrellas post-AGB, Estrellas K-gigantes, entre otras.

Esta ampliación permitirá una aplicación más versátil del modelo, facilitando la identificación de estrellas que no se encuentran clasificadas dentro del catálogo de Gaia.

## Referencias

1. Prusti, T., De Bruijne, J.H.J., Brown, A.G., Vallenari, A., Babusiaux, C., Bailer-Jones, C.A.L., and Navascués, D.B.: The Gaia mission. *Astronomy and Astrophysics*, vol. 595, pp. 36 (2016). DOI: 10.1051/0004-6361/201629272.
2. Vallenari, A., Brown, A.G., Prusti, T., De Bruijne, J.H., Arenou, F., Babusiaux, C., and Bianchi, L.: Gaia data release 3—summary of the content and survey properties. *Astronomy and Astrophysics*, vol. 674 (2022). DOI: 10.1051/0004-6361/202243940.
3. Gaia Data Release 3 contents summary—Gaia-Cosmos: <https://www.cosmos.esa.int/web/gaia/dr3> (2022)
4. Jastrow, R.: *Red Giants and White Dwarfs*. vol. 269 (1990)
5. Mikolajewska, J.: Symbiotic Stars: Observations confront theory. *Baltic Astronomy*, vol. 21, pp. 5–12 (2012). DOI: 10.1515/ASTRO-2017-0352.
6. Frankowski, A., Soker, N.: Very late thermal pulses influenced by accretion in planetary nebulae. *New Astronomy*, vol. 14, no. 8, pp. 654–658 (2009). DOI: 10.1016/J.NEAST.2009.03.006.
7. Kwok, S.: *The origin and evolution of planetary nebulae. The Origin and Evolution of Planetary Nebulae/Sun Kwok*, Cambridge University Press (2000)
8. Wenger, M., Ochsenbein, F., Egret, D., Dubois, P., Bonnarel, F., Borde, S., and Monier, R.: The SIMBAD Astronomical Database—The CDS Reference Database for Astronomical objects. *Astronomy and Astrophysics Supplement Series*, vol. 143, no. 1, pp. 9–22 (2000). DOI: 10.1051/AAS:2000332.
9. Babusiaux, C., Fabricius, C., Khanna, S., Muraveva, T., Rey, C., Spoto, F., and Weiler, M.: Gaia Data Release 3—Catalogue validation. *Astronomy and Astrophysics*, vol. 674, no. A32, pp. 1–25 (2023). DOI: 10.1051/0004-6361/202243790.
10. Carrasco, J.M., Weiler, M., Jordi, C., Fabricius, C., De Angeli, F., Evans, D.W., and Montegriffo, P.: Internal Calibration of Gaia BP/RP Low-Resolution Spectra. *Astronomy and Astrophysics*, vol. 652, no. A86 (2021). DOI: 10.1051/0004-6361/202141249.
11. Santos, M.S., Soares, J.P., Abreu, P.H., Araujo, H., and Santos, J.: Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [research frontier]. In: *IEEE Computational Intelligence Magazine*, vol. 13, no. 4, pp. 59–76 (2018). DOI: 10.1109/MCI.2018.2866730.
12. Breiman, L.: Random forests. *Machine Learning* 2001, vol. 45, pp. 5–32 (2001). DOI: 10.1023/A:1010933404324.
13. Noble, W.S.: What is a support vector machine?. *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567 (2006). DOI: 10.1038/nbt1206-1565.
14. Bishop, C.M.: *Neural Networks and their Applications. Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832 (1998). DOI: 10.1063/1.1144830.
15. Friedman, J.H.: Greedy Function Approximation: A Gradient Boosting Machine. *Annals of statistics*, vol. 29, no. 4, pp. 1189–1232, (2001). DOI: 10.1214/AOS/1013203451.
16. Powers, D.M.: Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *arXiv:2010.16061* (2020)
17. Chicco, D., Warrens, M.J., and Jurman, G.: The Matthews Correlation Coefficient (MCC) is more Informative than Cohen’s Kappa and Brier Score in Binary Classification Assessment. *IEEE*, vol. 9, pp. 78368–78381 (2021). DOI: 10.1109/ACCESS.2021.3084050.
18. Farhadpour, S., Warner, T.A., and Maxwell, A.E.: Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and best Practices. *Remote Sensing*, vol. 16, no. 3, pp. 533 (2024). DOI: 10.3390/RS16030533.



# Implementación de modelos preentrenados de procesamiento de lenguaje natural para la tarea de pregunta-respuesta en un Chatbot para consulta de información sobre notas clínicas de diabetes

Jonathan Zavala-Díaz<sup>1</sup>, Juan C. Olivares-Rojas<sup>1</sup>,  
Jennifer Páramo-Mascote<sup>1</sup>, J. Guadalupe Ramos-Díaz<sup>1</sup>,  
Johan W. González-Murueta<sup>1</sup>

Tecnológico Nacional de México,  
División de Estudios de Posgrado e Investigación, Morelia, Michoacán,  
México

{dl9123006, juan.or, l19120215, jose.rd,  
johan.gm}@morelia.tecnm.mx

**Resumen.** La diabetes, una enfermedad con un impacto global significativo en la salud, plantea desafíos considerables en su diagnóstico y tratamiento. Este artículo aborda la necesidad de mejorar la accesibilidad a información precisa sobre la diabetes mediante la aplicación de procesamiento de lenguaje natural (PLN) en notas clínicas. En esta investigación, los investigadores desarrollaron un Chatbot especializado utilizando la API de ChatGPT, con el objetivo de proporcionar respuestas pertinentes a consultas relacionadas con esta enfermedad. Además, los investigadores realizaron un análisis comparativo, que incluyó la evaluación de otros modelos de Hugging Face. Este proyecto propone una metodología para el desarrollo de un chatbot que responde preguntas sobre notas clínicas, lo cual constituye una contribución valiosa tanto al ámbito de la salud como al desarrollo de aplicaciones mediante el procesamiento de lenguaje natural.

**Palabras clave:** Procesamiento de lenguaje natural, API ChatGPT, Chatbot.

## Implementation of Pretrained Natural Language Processing Models for the Question-Answer Task in a Chatbot for Querying Information on Diabetes Clinical Notes

**Abstract.** Diabetes, a disease with a significant global health impact, poses considerable challenges in its diagnosis and treatment. This article addresses the need to improve accessibility to accurate diabetes information by applying natural language processing (NLP) to clinical notes. In this research, the researchers developed a specialized Chatbot using the ChatGPT API, with the aim of providing relevant answers to queries related to this disease. Additionally, the researchers conducted a comparative analysis, which included evaluating

other Hugging Face models. This project proposes a methodology for the development of a chatbot that answers questions about clinical notes, which constitutes a valuable contribution both to the field of health and to the development of applications through the natural language process.

**Keywords:** Natural language processing, ChatGPT API, Chatbot.

## 1. Introducción

La diabetes pertenece a un grupo de enfermedades metabólicas y es consecuencia de la deficiencia en el efecto de la insulina, causada por una alteración en la función endocrina del páncreas o por la alteración en los tejidos efectores, que pierden su sensibilidad a la insulina [1]. La diabetes representa un grave problema de salud pública. Su incidencia oscila entre el 1-2% de la población mundial. El tipo más frecuente es la diabetes no insulino dependiente o tipo 2 [2]. Los registros médicos electrónicos (EHR) contienen datos cruciales de los pacientes en notas clínicas. A medida que estas notas crecen en volumen y complejidad, la extracción manual se vuelve un desafío [3]. En este contexto superar las limitaciones de tiempo en entornos médicos y revolucionar cómo los profesionales acceden y manejan los datos clínicos, prometiendo no solo optimizar los procesos sino también transformar radicalmente la interacción con la información médica se ha vuelto crucial.

El Procesamiento de Lenguaje Natural (PLN) se refiere a un campo apasionante dentro del ámbito de la ciencia de la información que se ocupa de analizar el lenguaje natural en sus diversas variantes. Con el progreso en el procesamiento del lenguaje natural (PNL), la extracción de información valiosa de la literatura biomédica ha ganado popularidad entre los investigadores, y el aprendizaje profundo ha impulsado el desarrollo de modelos eficaces de minería de textos biomédicos [4].

Su integración en entornos hospitalarios representa un hito trascendental con el potencial de mejorar significativamente la eficiencia y accesibilidad a información crucial, particularmente en el contexto de enfermedades crónicas como la diabetes. En particular, una aplicación del PLN en el contexto de la diabetes se anticipa a proporcionar respuestas contextualizadas y precisas, alterando fundamentalmente la gestión y comprensión de los datos clínicos de esta enfermedad significativa.

Este estudio explora la aplicación pionera del Procesamiento de Lenguaje Natural (PLN) para la extracción de datos específicos sobre la diabetes a partir de notas clínicas, utilizando modelos preentrenados en tareas de pregunta-respuesta mediante un chatbot y la API de ChatGPT como componente fundamental.

Este enfoque examina cómo integrar tecnologías avanzadas para abordar desafíos prácticos en la atención médica, abriendo al mismo tiempo nuevas vías para tomar decisiones más informadas y efectivas. Al analizar detalladamente el impacto potencial que el PLN podría tener en el acceso a información diabética en el ámbito hospitalario, y complementándolo con un estudio comparativo que evalúa otros modelos de Hugging Face, este trabajo emerge como un paso adelante esencial para mejorar la accesibilidad y la calidad de la información sobre la diabetes.

Esta contribución es relevante tanto para el campo de la salud como para el desarrollo de aplicaciones mediante el procesamiento de lenguaje natural, resaltando la importancia de acceder a información precisa sobre la diabetes a través del desarrollo



de un chatbot especializado y la realización de un análisis comparativo. Este proyecto se posiciona como una iniciativa transformadora para el manejo futuro de la información médica.

## **2. Trabajos relacionados**

En el ámbito clínico, se los investigadores han explorado el uso de chatbots para apoyar decisiones clínicas. Un estudio destacado [5] exploró cómo un chatbot interactivo, basado en los criterios de idoneidad del American College of Radiology (ACR) y utilizando procesamiento de similitud semántica, podría ofrecer recomendaciones personalizadas de imágenes médicas. Este sistema, denominado accGPT, utilizó con 209 documentos de criterios de ACR y se integró con tecnologías como LlamaIndex y ChatGPT-3.5-turbo, lo que permitió su conexión con bases de datos externas y el procesamiento avanzado del lenguaje. En pruebas comparativas con cincuenta casos clínicos, accGPT demostró un rendimiento superior al de radiólogos de diferentes niveles de experiencia y versiones genéricas de ChatGPT, resaltando así el potencial de los algoritmos basados en ChatGPT para optimizar la selección de estudios de imágenes clínicas siguiendo las directrices del ACR y evidenciando la importancia de adaptar tecnologías de inteligencia artificial a necesidades médicas específicas.

En [6] se evaluó la precisión de listas de diagnóstico diferencial creadas por ChatGPT-3 para notas clínicas basadas en síntomas comunes (incluyendo dolor abdominal, fiebre, dolor de pecho, dificultad respiratoria, dolor articular, vómitos, ataxia/dificultades para caminar, dolor de espalda, tos y mareos). Médicos especializados en medicina interna general diseñaron casos clínicos, identificaron los diagnósticos correctos y propusieron cinco diagnósticos diferenciales para cada una de las diez principales quejas. Este estudio demuestra la notable precisión de ChatGPT-3 al generar listas de diagnóstico diferencial para quejas clínicas comunes, sugiriendo que chatbots de IA como ChatGPT-3 pueden ofrecer listas de diagnóstico diferencial bien fundamentadas para síntomas frecuentes.

Los investigadores han llevado a cabo estudios sobre el uso de chatbots en el campo médico, centrándose particularmente en patologías específicas, como el cáncer. En Xu y colaboradores [7] revisan y buscan esclarecer los progresos recientes y las tendencias actuales en la aplicación de tecnología de chatbots en el ámbito médico, centrándose especialmente en el cáncer. Los chatbots examinados se categorizan según su área de aplicación, tales como la detección de síntomas, recomendaciones para el tratamiento de pacientes, monitorización remota de pacientes, apoyo emocional, promoción de una alimentación saludable, entre otros aspectos relevantes.

En el contexto de la diabetes, existen investigaciones destacadas, como la señalada en [8], que presentan el desarrollo de un chatbot diagnóstico. Este chatbot interactúa con los pacientes mediante conversaciones, empleando técnicas avanzadas de comprensión del lenguaje natural para ofrecer predicciones personalizadas. Utiliza datos generales de salud y síntomas específicos proporcionados por el paciente para predecir una variedad de enfermedades de manera genérica y detallada. En casos donde la predicción general indica diabetes, el sistema profundiza sus análisis tomando en cuenta atributos específicos relacionados con esta enfermedad.

En [9] se muestra un trabajo en el cual desarrollan un chatbot de recomendación de alimentos personalizado para pacientes con diabetes.

Existen diversas investigaciones relacionadas con la implementación de chatbots clínicos para enfermedades específicas, utilizando una variedad de técnicas [10]–[14]. Sin embargo, nuestro trabajo se enfoca en responder preguntas específicas sobre las notas clínicas de pacientes diabéticos utilizando modelos de lenguaje de pre-entrenados existentes, con el objetivo de evaluar su aplicabilidad en este dominio.

### **3. Marco teórico**

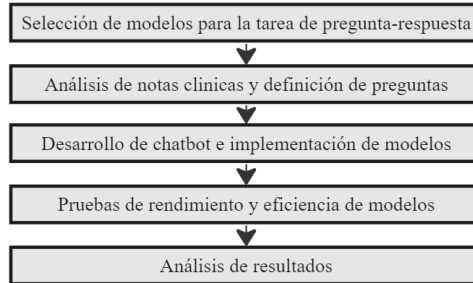
#### **3.1. Modelos de lenguaje preentrenados**

Los modelos de lenguaje previamente entrenados han logrado un éxito sorprendente en el procesamiento del lenguaje natural (PNL), lo que ha llevado a un cambio de paradigma del aprendizaje supervisado al entrenamiento previo seguido de un ajuste fino [15]. Los investigadores están realizando esfuerzos significativos en la creación de modelos preentrenados destinados a tareas específicas, tales como clasificación de textos, pregunta-respuesta, traducción, generación de texto, extracción de características, entre otras. Sin embargo, ha surgido la necesidad de ajustar estos modelos para dominios especializados, como el entorno clínico. El entrenamiento de un modelo en un contexto particular contribuirá a una comprensión más profunda de los términos específicos de ese dominio. En este sentido, hay trabajos en curso que se enfocan en la creación de modelos preentrenados específicamente para el ámbito clínico [16-20]. Además, la utilización de tecnologías como GPT (Generative Pre-trained Transformer) ha demostrado ser especialmente prometedora en esta área, ofreciendo herramientas poderosas para la comprensión y generación de texto en contextos clínicos específicos.

#### **3.2. API Chat-GPT**

Las avanzadas capacidades en el campo del procesamiento del lenguaje natural (PLN) han visto progresos notables, siendo ChatGPT la herramienta de PLN que ha alcanzado un éxito sin precedentes desde finales del 2022. Los hallazgos subrayan el valor de la Interfaz de Programación de Aplicaciones (API) de ChatGPT como un recurso significativo en el ámbito del desarrollo de software, proveyendo respuestas inteligentes y eficaces para un amplio espectro de usos. Esta herramienta se basa en un entrenamiento con una vasta biblioteca de conocimientos, consistente en 570 gigabytes de texto y un modelo con más de 175 millones de parámetros, de acuerdo con información de la Universidad de Stanford. Al igual que otros servicios modernos, OpenAI ofrece a sus usuarios una API que facilita el acceso a sus variados servicios, lo cual permite a los desarrolladores y a las empresas incorporar estas capacidades en sus propias aplicaciones [21].

Los Modelos de Lenguaje Grandes como ChatGPT demuestran la capacidad de brindar consejos precisos, informativos y seguros en escenarios clínicos, al tiempo que enfatizan la importancia de involucrar a colegas médicos de alto nivel en las primeras etapas del proceso de toma de decisiones [22].



**Fig. 1.** Metodología.

### 3.3. Chat Bot

Los chatbots pueden imitar la conversación humana utilizando un campo de inteligencia artificial (IA) conocido como procesamiento del lenguaje natural. Los chatbots ahora se utilizan ampliamente en varias formas como agentes basados en voz, como Siri (Apple), Google Now (Google), Alexa (Amazon) o Cortana (Microsoft). Los chatbots basados en texto están disponibles como agentes de Messenger (Facebook) o como aplicaciones web o móviles independientes. Proporcionan información y crean una interacción dinámica entre el agente y el usuario, sin intervención humana de fondo [22].

Los chatbots se han convertido en la plataforma de referencia para que los usuarios reciban respuestas a sus consultas. Pero cuando se trata de entablar un diálogo con un usuario, los chatbots existentes tienen varias deficiencias, con problemas como no proporcionar una respuesta significativa al usuario, ofrecer información semánticamente incorrecta, etc. [23].

## 4. Metodología

La Fig.1 muestra las etapas de la metodología empleada en esta investigación. Inicialmente, la primera etapa se enfoca en la selección de los modelos a utilizar en este estudio, incluyendo tanto los provenientes de la API de Chat GPT como aquellos de la plataforma Hugging Face. La segunda etapa implica el análisis de notas clínicas de pacientes con diabetes y la definición de las preguntas que podrán ser respondidas a partir de estas notas.

La tercera etapa se dedica al desarrollo de un chatbot que permite seleccionar la nota clínica y la pregunta a responder, además de su integración con un modelo de lenguaje preentrenado específico para la tarea de pregunta-respuesta. La cuarta etapa comprende la realización de pruebas y la evaluación de la eficacia de los modelos seleccionados para responder las preguntas identificadas previamente.

Este proceso se describe con más detalle en la sección 4.4. Posteriormente, se lleva a cabo el análisis de los resultados obtenidos.

**Tabla 1.** Modelos HuggingFace.

Modelo	Clasificación
mdeberta-v3-base-squad2 [24]	1
distill-bert-base-spanish-wwm-cased-finetuned-spa-squad2-es [25]	2
bert-base-spanish-wwm-cased-finetuned-spa-squad2-es [26]	3
roberta-large-bne-sqac [27]	4
xlm-roberta-base-finetuned-squad2 [28]	5
longformer-base-4096-spanish-finetuned-squad [29]	6
ixambert-finetuned-squad [30]	7

#### 4.1. Selección de modelos para la tarea de pregunta-respuesta

La selección de modelos de procesamiento de lenguaje natural en esta fase implicó un proceso de evaluación y análisis. Esta selección incluyó modelos tanto de Hugging Face como de la API de ChatGPT, centrándose particularmente en aquellos especializados en preguntas y respuestas para el idioma español.

En el caso de los modelos de Hugging Face, se llevó a cabo un análisis preliminar de veinte modelos, agrupándolos en dos conjuntos de diez según criterios específicos. El primer grupo se organizó según el número de descargas, mientras que el segundo se basó en la popularidad, medida en términos de la cantidad de "likes" recibidos.

Después de esta evaluación inicial, se procedió a una selección más detallada, identificando los siete modelos más destacados que sobresalieron en ambas categorías. La Tabla 1 muestra el resultado del proceso de evaluación de modelos de Hugging Face, enfocado en tareas de preguntas y respuestas para el idioma español en el ámbito del procesamiento de lenguaje natural. Estos modelos, varían en arquitecturas—tales como DeBERTa, DistilBERT, BERT, RoBERTa, y XLM-RoBERTa—.

Al mismo tiempo, se decidió seleccionar cinco modelos destacados en la API de ChatGPT basándose específicamente en su reconocida capacidad para proporcionar respuestas.

La Tabla 2 presenta los modelos GPT utilizados en este artículo, así como sus costos de uso por token. Además, para una mejor comprensión del límite de tokens, se puede concebir a estos como fragmentos de palabras, donde aproximadamente 1000 tokens equivalen a 750 palabras.

#### 4.2. Análisis de notas clínicas y definición de preguntas

Se llevó a cabo un análisis con el propósito de identificar las preguntas que podrían ser abordadas a partir de las notas médicas de pacientes con diabetes disponibles. Para ello, se seleccionaron aleatoriamente 10 notas médicas de pacientes diabéticos, las cuales forman parte de un conjunto de notas clínicas específicas del proyecto de doctorado al que pertenecen los autores. Luego, la información hallada en estas notas fue clasificada en 11 tópicos pertinentes, los cuales se detallan en la Tabla 3 acompañados de una pregunta asociada a cada tópico.

**Tabla 2.** Modelos GPT.

Modelo	Precio
"davinci-002"/GPT Base [31]	\$0.0020 dólares/1000 tokens
"babbage-002"/GPT Base [32]	\$0.0004 / 1000 tokens
"text-davinci-002"/GPT-3.5 [33]	\$0.0200 dólares/1000 tokens
"text-davinci-003"/GPT-3.5 [34]	\$0.0200 dólares/1000 tokens
"gpt-3.5-turbo-instruct"/GPT-3.5 Turbo [35]	Entrada: \$0.0015 dólares/1000 tokens Salida: \$0.0020 dólares/1000 tokens

**Tabla 3.** Tópicos y preguntas.

No.	Tópicos en las notas clínicas	Preguntas por tópico
1	Datos del Paciente	¿Cuál es la edad y género de la paciente?
2	Antecedentes Médicos	¿Cuál es el historial médico del paciente?
3	Estado de Salud Actual	¿Cuál es el motivo de consulta actual de la paciente?
4	Exploración Física	¿Qué se observa en la exploración física de la paciente?
5	Diagnóstico	¿Cuáles son los diagnósticos registrados para la paciente?
6	Plan de Tratamiento y Recomendaciones	¿Qué medidas higiénico-dietéticas y recomendaciones de educación en salud se le han proporcionado a la paciente?
7	Tratamiento farmacológico	¿Qué medicamentos se han prescrito para el tratamiento de la paciente?
8	Fecha de la Próxima Cita Médica	¿Cuándo está programada la próxima cita médica para la paciente?
9	Resultados de laboratorio	¿Cuáles son los resultados de laboratorio?
10	Pronóstico	¿Cuál es la perspectiva de mejora para la paciente?
11	Referencias a Otros Servicios de Salud	¿Habrá una remisión a otros servicios médicos o especialistas para la paciente?

**Tabla 4.** Tópicos encontrados

Nota Clínica	Tópicos en las notas clínicas										
	1	2	3	4	5	6	7	8	9	10	11
1	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No
2	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	No
3	Sí	No	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	No
4	No	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí
5	No	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	No
6	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No
7	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No
8	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No
9	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No
10	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No

En la Tabla 4 se muestran los tópicos identificados en las 10 notas médicas. Se observa que el tópico 11, "Referencias a Otros Servicios de Salud", solo está presente en 1 de las 10 notas clínicas, por lo que se descartará para el resto de este trabajo.

### **4.3. Desarrollo de Chatbot e implementación de modelos**

En este proyecto se desarrollaron dos chatbots distintos, uno de ellos está diseñado para integrarse con los modelos de la API de OpenAI, mientras que el otro está configurado para trabajar con modelos seleccionados de Hugging Face. Ambos chatbots siguen una lógica similar en su implementación y su programación fue en python.

Para implementar los modelos preentrenados, utilizamos la biblioteca Transformers, la cual es un componente fundamental de Hugging Face y es reconocida por su amplia selección de modelos afinados para diversas tareas en Procesamiento de Lenguaje Natural (PLN). El uso de la función "pipeline" en Hugging Face Transformers simplificó notablemente la activación de modelos para tareas específicas de PLN. En particular, se optó por la función "pipeline" con el parámetro "question-answering", estableciendo de esta manera un método de trabajo eficaz enfocado en la respuesta a preguntas, una vertiente práctica y esencial del PLN. Para integrar los modelos de ChatGPT, se recurrió a la biblioteca de OpenAI, diseñada para facilitar la interacción con la API de OpenAI y permitir el acceso a avanzados modelos de PLN. Esta herramienta posibilita el envío de consultas a la API de OpenAI y la recepción de las respuestas generadas, abriendo un canal directo para la explotación de estas tecnologías en aplicaciones específicas.

Para la implementación del proyecto, se desarrollaron diversas funciones categorizadas en tres áreas principales:

**Funciones de Procesamiento de Entrada:** Esta fase se enfoca en la preparación de las notas clínicas para su análisis subsiguiente. Implica un meticuloso proceso de limpieza de texto, que busca eliminar caracteres no deseados y normalizar el formato, asegurando que los datos estén listos para un procesamiento eficiente.

**Funciones de Interacción:** En esta etapa, el usuario elige un modelo de Hugging Face o de la API ChatGPT previamente seleccionado para responder preguntas. Ingresar el nombre de la nota clínica y elige entre diez preguntas predefinidas. Después de responder las diez preguntas, puede cambiar de modelo o consulta para comenzar de nuevo.

**Funciones de Salida:** Tras obtener las respuestas del modelo de Hugging Face o de la API ChatGPT seleccionado, se evalúa su puntaje de confianza para determinar la precisión y relevancia. Las respuestas que no alcanzan un umbral de confiabilidad establecido son filtradas. Se agrega un comentario personalizado a cada respuesta del modelo para enriquecer el contenido. Las preguntas y respuestas se guardan en archivos CSV etiquetados con el nombre del modelo y la nota clínica consultada, lo que facilita el análisis posterior de la información procesada.

### **4.4. Pruebas de rendimiento y eficiencia de modelos**

Se llevó a cabo una evaluación exhaustiva de 12 modelos, que incluían 5 de la API de ChatGPT y los otros 7 de Hugging Face, con el objetivo de determinar cuál sería el

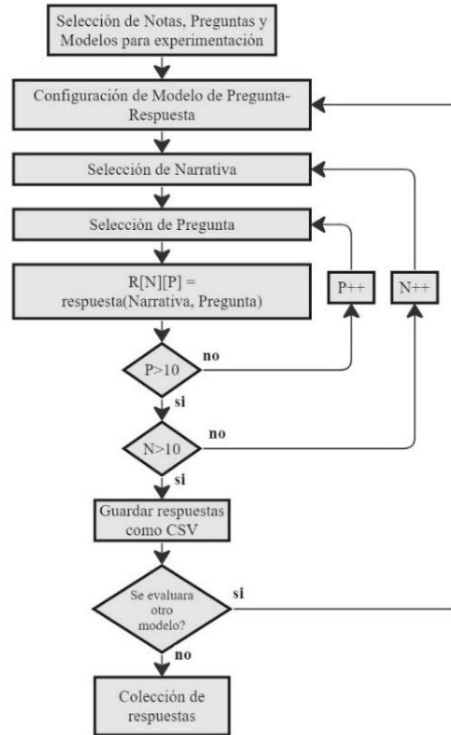


Fig. 2. Metodología de la etapa experimental.

más adecuado para el proyecto. Se probó los 12 modelos utilizando 10 notas clínicas específicas de la base de datos. Estas pruebas incluyeron las 10 preguntas del Chatbot. Posteriormente, se guardó cada una de las respuestas de cada modelo para cada nota en archivos de texto, con el propósito de realizar un análisis manual exhaustivo y examinar las respuestas proporcionadas por cada modelo.

Para cada nota, se registró cada una de las respuestas correctas e incorrectas en un archivo de CSV que abarcaba todos los modelos. Luego, se calculó el promedio total de cada modelo para evaluar cuál era el mejor dentro de Hugging Face, y de la API de ChatGPT, esta metodología se muestra en la Fig. 2.

## 5. Resultados

Se llevó a cabo un análisis minucioso de cada respuesta proporcionada por los modelos, tanto Hugging Face como la API de ChatGPT. Este análisis se centró en determinar si las respuestas eran correctas o incorrectas, asignándoles valores de 1 y 0 respectivamente. El propósito de esta evaluación fue calcular un promedio final de respuestas correctas para cada modelo en una escala del 0 al 10.

Para evaluar la precisión de las respuestas, se contó con la asesoría de un médico especialista. Este experto recibió las notas clínicas pertinentes junto con las preguntas planteadas a los modelos. El rol del médico no consistió en comparar directamente sus

respuestas con las de los modelos, sino en validar y asesorar sobre la precisión de las respuestas generadas por los modelos.

La validación del médico se basó en su conocimiento y experiencia clínica, así como en la información proporcionada en las notas clínicas. Se evaluó si las respuestas de los modelos eran coherentes y precisas en el contexto de la información médica disponible. Esto implicaba que las respuestas debían estar alineadas con el diagnóstico médico y las recomendaciones clínicas, siendo consideradas correctas si coincidían con esta información y incorrectas si se desviaban o contradecían la misma.

Una vez validadas por el médico, las respuestas fueron comparadas con las generadas por los modelos. Esta comparación permitió determinar la precisión de las respuestas automáticas en relación con la validación médica. De este modo, se pudo evaluar con mayor precisión el desempeño de cada modelo en términos de su capacidad para proporcionar respuestas médicamente precisas y útiles.

La tabla 5 presenta los resultados de la evaluación de los modelos en cada una de las 10 notas clínicas, mostrando el número de respuestas evaluadas correctamente por cada nota y calculando el promedio de cada modelo. Se observa que el modelo gpt-3.5-turbo-instruct tuvo el mejor rendimiento en comparación con los demás modelos, los cuales presentaron un rendimiento inferior.

Destaca especialmente el modelo mdeberta-v3-base-squad2, el cual obtuvo el mejor rendimiento con una puntuación de 4.9 entre los modelos de Hugging Face. Se puede apreciar que los modelos de la API ChatGPT exhiben una precisión superior en comparación con otros modelos, destacando su mayor precisión en la tarea de pregunta respuesta.

La Fig. 3 muestra la pantalla del chatbot desarrollado, el chatbot empieza con la elección de un modelo por parte del usuario. Una vez elegido dicho modelo, se solicita al usuario que ingrese el nombre de la nota clínica que desea utilizar, verificando de que la nota esté en formato .txt, para luego mostrar al usuario las diez preguntas predefinidas, y así el usuario debe elegir una pregunta marcando su número correspondiente, el ChatBot procesa la pregunta utilizando el modelo seleccionado y genera una respuesta. La pregunta seleccionada junto con la respuesta generada se muestra en pantalla.

## **6. Discusión de los resultados**

Tras revisar detenidamente los resultados anteriores, los cuales ofrecen una visión integral del rendimiento de los modelos evaluados, podemos extraer valiosa información y análisis detallados.

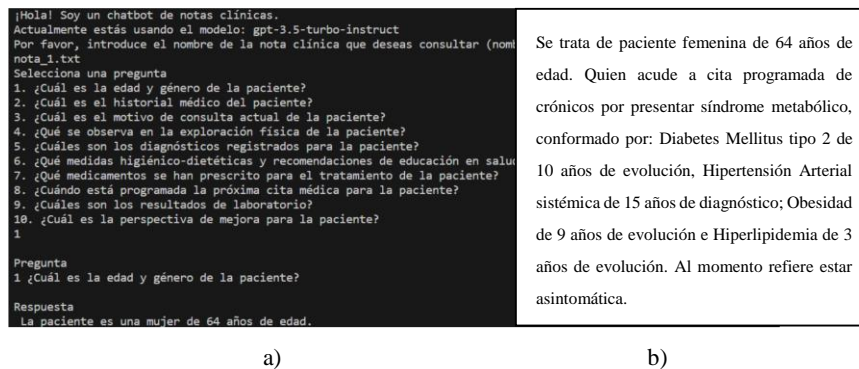
Al evaluar el rendimiento de los 12 modelos, se observa que los modelos de la API de ChatGPT, superan a los demás en términos de respuestas correctas. Esto sugiere una alta efectividad en la comprensión y respuesta de preguntas relacionadas con notas clínicas de diabetes.

Los modelos GPT de la API de ChatGPT, incluyendo "text-davinci002", "text-davinci-003" y "gpt-3.5-turbo-instruct", exhibieron un rendimiento destacado, con un promedio de respuestas correctas cercano al 10. Estos modelos demuestran una capacidad excepcional para proporcionar respuestas precisas y coherentes.



**Tabla 5.** Resultados evaluación de modelos.

Modelo	Notas Clínicas										Prom.
	1	2	3	4	5	6	7	8	9	10	
mdeberta... [24]	8	3	3	4	2	4	6	3	8	8	4.9
distill-bert... [25]	4	0	2	1	1	3	4	1	6	6	2.8
bert-base... [26]	4	0	2	2	1	1	4	2	4	3	2.3
roberta-large... [27]	6	1	3	0	0	1	1	1	5	5	2.3
xlm-roberta... [28]	6	2	3	2	2	3	3	2	6	6	3.5
longformer-base... [29]	1	1	2	1	1	2	4	1	2	2	1.7
ixambert-finetuned... [30]	4	1	2	2	1	2	1	1	4	4	2.2
davinci-002 [31]	2	2	3	3	3	2	2	2	2	0	2.1
babbage-002 [32]	3	2	1	1	1	3	3	3	2	2	2.1
text-davinci-002 [33]	10	6	8	7	8	8	8	9	7	9	8
text-davinci-003 [34]	10	8	9	9	9	9	9	10	8	9	9
<b>gpt-3.5-turbo-instruct [35]</b>	<b>10</b>	<b>9</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>9.9</b>



**Fig. 3.** a) Ejemplo de uso de Chatbot, b) Nota clínica asociada a la consulta del ejemplo.

Entre los modelos de Hugging Face, el "timpa01/mdeberta-v3-base-squad2" se destacó como el mejor, con una calificación promedio de respuestas correctas de 4.9. Aunque los modelos de Hugging Face muestran un rendimiento sólido, los modelos de la API de ChatGPT superan en efectividad según la métrica elegida.

En términos generales, el "gpt-3.5-turbo-instruct" se destacó como el mejor modelo, con una impresionante efectividad del 9.9. Este modelo de la API de ChatGPT exhibe un rendimiento excepcional, con apenas un 0.1% de error en las respuestas evaluadas.

Considerando la elección del modelo "gpt-3.5-turbo-instruct" como el mejor entre los 12 evaluados, es necesario abordar aspectos adicionales antes de su implementación, como los costos asociados. Este modelo, destacado por su rendimiento excepcional, presenta particularidades que deben sopesarse cuidadosamente en el contexto de nuestro proyecto. A continuación, se examinan las ventajas y desventajas en comparación con el modelo "timpa01/mdeberta-v3-base-squad2", que se destacó como la mejor opción en el ámbito gratuito de Hugging Face.

El ChatBot desarrollado con la API de OpenAI ha demostrado tener una capacidad sobresaliente para comprender y responder las preguntas. Su rendimiento en términos de respuestas correctas es muy alto. Esto resalta la capacidad sobresaliente de algunos

modelos que alcanzan puntajes perfectos en múltiples casos. Sin embargo, se reconocen que algunos modelos tienen un mal desempeño, esto a que son modelos más antiguos teniendo una arquitectura de redes neuronales menos avanzadas. Además, este ChatBot ha demostrado ser escalable, con una enorme capacidad de manejar un alto volumen de consultas de manera eficiente, lo que podría adaptarse a las necesidades cambiantes de los usuarios sin comprometer la calidad del servicio.

El ChatBot desarrollado con modelos de Hugging Face también ha demostrado ser competente en el ámbito de respuestas para preguntas. Aunque su rendimiento no es tan bueno como los modelos de la API de OpenAI, se llegó a observar que las respuestas generadas pueden variar en consistencia y precisión, y aún pueden ser útiles y más con algunos modelos como el de “mdeberta-v3-base-squad2”. Esto sugiere una comprensión media en el contexto y las consultas realizadas por parte de los modelos de Hugging Face.

## **7. Conclusiones**

La evaluación revela que, al manejar datos médicos, donde la precisión y exactitud son críticas, el modelo de Hugging Face, a pesar de ser gratuito, proporciona solo la mitad de las respuestas correctas. Esta constatación subraya la prioridad absoluta que se otorga a la precisión en la generación de respuestas para garantizar la calidad y fiabilidad de la información médica.

El excepcional rendimiento de la API de ChatGPT en la generación de respuestas contextualmente relevantes superó nuestras expectativas, desencadenando una evaluación exhaustiva del costo-beneficio frente a otros modelos. Aunque la elección del modelo “gpt-3.5-turbo-instruct” implica costos adicionales, su rendimiento excepcional y capacidad para cumplir con los estándares médicos lo posicionan como la opción preferida. La priorización de la precisión en la interpretación de datos médicos justifica la inversión, enfocándonos en la calidad de la información generada. Este enfoque respalda nuestra decisión de adoptar una solución que destaca tanto en eficacia como en la satisfacción de los rigurosos requisitos médicos.

El ChatBot desarrollado con la API de OpenAI como el de Hugging Face han demostrado competencia en la generación de respuestas para preguntas relacionadas con notas clínicas de diabetes. El ChatBot de OpenAI destaca por su rendimiento excepcional y escalabilidad, lo que lo hace una buena elección para proyectos que requieren respuestas precisas y coherente. Por otro lado, el ChatBot de Hugging Face también tiene una habilidad en la generación de respuestas para preguntas, aunque puede necesitar mejoras en consistencia y precisión.

Por lo tanto, la elección entre un modelo u otro dependerá de las necesidades específicas del proyecto, y la disponibilidad de recursos. Ambos ChatBots ofrecen opciones buenas. Este estudio proporciona una visión general sobre la aplicabilidad de herramientas disponibles para su integración en el desarrollo de aplicaciones clínicas. Esto sugiere que su uso puede replicarse en diversas aplicaciones clínicas en futuros proyectos, abarcando distintos tipos de notas clínicas o tareas variadas.

## Referencias

1. Polonsky, K. S.: The Past 200 Years in Diabetes. *New England Journal of Medicine*, vol. 367, no. 14, pp. 1332–1340 (2012). DOI: 10.1056/NEJMra1110560.
2. Cervantes-Villagrana, R.D., Presno-Bernal, J.M.: Fisiopatología de la diabetes y los mecanismos de muerte de las células  $\beta$  pancreáticas. *Revista de Endocrinología y Nutrición*, vol. 21, no. 3, pp. 98–106 (2013).
3. Elgedawy, R., Srinivasan, S., Danciu, I.: Dynamic Q&A of Clinical Documents with Large Language Models (2024). DOI: 10.48550/arXiv.2401.10733.
4. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240 (2020). DOI: 10.1093/bioinformatics/btz682.
5. Rau, A., Rau, S., Zoeller, D., Fink, A., Tran, H., Wilpert, C., Nattenmüller, J., Neubauer, J., Bamberg, F., Reiser, M., Russe, M.F.: A context-based chatbot surpasses radiologists and generic ChatGPT in following the ACR appropriateness guidelines. *Radiology*, vol. 308, no. 1, pp. e230970 (2023). DOI: 10.1148/radiol.230970.
6. Hirose, T., Harada, Y., Yokose, M., Sakamoto, T., Kawamura, R., Shimizu, T.: Diagnostic accuracy of differential-diagnosis lists generated by generative pretrained transformer 3 chatbot for clinical vignettes with common chief complaints: A pilot study. *International journal of environmental research and public health*, vol. 20, no. 4, pp. 3378–3378 (2023). DOI: 10.3390/ijerph20043378.
7. Xu, L., Sanders, L., Li, K., Chow, J.C.: Chatbot for health care and oncology applications using artificial intelligence and machine learning: Systematic review. *JMIR cancer*, vol. 7, no. 4, pp. e27850 (2021). DOI:10.2196/27850.
8. Bali, M., Mohanty, S., Chatterjee, S., Sarma, M., Puravankara, R.: Diabot: A predictive medical chatbot using ensemble learning. *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 6334–6340 (2019). DOI: 10.35940/ijrte.B2196.078219.
9. Thongyoo, P., Anantapanya, P., Jamsri, P., Chotipant, S.: A Personalized Food Recommendation Chatbot System for Diabetes Patients. In: Luo, Y. (eds) *Cooperative Design, Visualization, and Engineering. CDVE '20. Lecture Notes in Computer Science*, pp. 19–28 (2020). DOI: 10.1007/978-3-030-60816-3\_3.
10. Battineni, G., Chintalapudi, N., Amenta, F.: AI chatbot design during an epidemic like the novel coronavirus. *Healthcare*, vol. 8, No. 2, pp. 154 (2020). DOI: 10.3390/healthcare8020154.
11. Bhayana, R.: Chatbots and large language models in radiology: A practical primer for clinical and research applications. *Radiology*, vol. 310, no. 1, pp. e232756.310 (2024)
12. Ahmed, S.T., Fathima, A.S., Nishabai, M., and Sophia, S.: Medical ChatBot assistance for primary clinical guidance using machine learning techniques. *Procedia Computer Science*, vol. 233, pp. 279–287 (2024)
13. Platz, J.J., Bryan, D.S., Naunheim, K.S., Ferguson, M.K.: Chatbot reliability in managing thoracic surgical clinical scenarios. *The Annals of Thoracic Surgery*. vol. 118, no. 1, pp. 275–281 (2023). DOI: 10.1016/j.athoracsur.2024.03.023.
14. Gracias, D., Siu, A., Seth, I., Dooremeah, D., Lee, A.: Exploring the role of an artificial intelligence chatbot on appendicitis management: an experimental study on ChatGPT. *ANZ Journal of Surgery*, vol. 94, no. 3, pp. 342–352 (2024)
15. Wang, H., Li, J., Wu, H., Hovy, E., Sun, Y.: Pre-trained language models and their applications. *Engineering*, vol. 25, pp. 51–65 (2023)
16. Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D.: Med-BERT: Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ Digital Medicine*, vol. 4, no. 1, pp. 86 (2021). DOI: 10.1038/s41746-021-00455-y.

17. Li, Y., Wehbe, R.M., Ahmad, F.S., Wang, H., Luo, Y.: A comparative study of pretrained language models for long clinical text. *Journal of the American Medical Informatics Association*, vol. 30, no. 2, pp. 340–347 (2022). DOI: 10.1093/jamia/ocac225.
18. Carrino, C.P., Llop, J., Pàmies, M., Gutiérrez-Fandiño, A., Armengol-Estapé, J., Silveira-Ocampo, J., Villegas, M.: Pretrained biomedical language models for clinical NLP in Spanish. In: *Proceedings of the 21st Workshop on Biomedical Language Processing*, pp. 193–199, (2022). DOI: 10.18653/v1/2022.bionlp-1.19.
19. Liu, H., Zhang, Z., Xu, Y., Wang, N., Huang, Y., Yang, Z., Chen, H.: Use of BERT (bidirectional encoder representations from transformers)-based deep learning method for extracting evidences in Chinese radiology reports: development of a computer-aided liver cancer diagnosis framework. *Journal of medical Internet research*, vol. 23, no. 1, pp. e19689 (2021). DOI:10.2196/19689.
20. Zhou, S., Wang, N., Wang, L., Liu, H., Zhang, R.: CancerBERT: A cancer domain-specific language model for extracting breast cancer phenotypes from electronic health records. *J. Am. Med. Informatics Assoc.*, vol. 29, pp. 1208–1216 (2022). DOI: 10.1093/jamia/ocac040.
21. Paredes, C.M.G., Machuca, C., Claudio, Y.M.S.: ChatGPT API: Brief overview and integration in Software Development. *International Journal of Engineering Insights*, vol. 1, no. 1, pp. 25–29 (2023).
22. Bibault, J.E., Chaix, B., Guillemassé, A., Cousin, S., Escande, A., Perrin, M., and Brouard, B.: A chatbot versus physicians to provide information for patients with breast cancer: Blind, randomized controlled noninferiority trial. *Journal of medical Internet research*, vol. 21, no. 11, pp. e15787 (2019). DOI: 10.2196/15787.
23. Nuruzzaman, M., Hussain, O.K.: IntelliBot: A Dialogue-based chatbot for the insurance industry. *Knowledge-Based Systems*, vol. 196, pp. 105810 (2020)
24. Isbister, T.: mdeberta-v3-base-squad2. at <https://huggingface.co/timpal01/mdeberta-v3-base-squad2> (2024)
25. Romero, M.: BETO (Spanish BERT) + Spanish SQuAD2.0 + distillation using ‘bert-base-multilingual-cased’ as teacher. <https://huggingface.co/mrm8488/distill-bert-base-spanish-wm-cased-finetunedspa-squad2-es> (2021)
26. Romero, M.: BETO (Spanish BERT) + Spanish SQuAD2.0. at <https://huggingface.co/mrm8488/bert-base-spanish-wm-cased-finetuned-spa-squad2-es> (2020)
27. Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C.P., and Villegas, M.: Maria: Spanish language models. *arXiv preprint arXiv:2107.07253*, (2021). DOI: 10.26342/2022-68-3.
28. Chaudhari, S.: XLM-roberta-base-finetuned-squad2. <https://huggingface.co/IProject-10/xlm-roberta-base-finetuned-squad2> (2024)
29. Romero, M.: Spanish Longformer fine-tuned on SQAC for Spanish QA. <https://huggingface.co/mrm8488/longformer-base-4096-spanish-finetuned-squad> (2021)
30. Brun, M.: ixambert-base-cased finetuned for QA. <https://huggingface.co/MarcBrun/ixambert-finetuned-squad> (2022)
31. OpenIA: davinci-002. <https://platform.openai.com/docs/models/gpt-base> (2023)
32. OpenIA: babbage-002. <https://platform.openai.com/docs/models/gpt-base> (2023)
33. OpenIA: text-davinci-002. <https://platform.openai.com/docs/models/gpt-3-5-turbo> (2024)
34. OpenIA: text-davinci-003. <https://platform.openai.com/docs/models/gpt-3-5-turbo> (2024)
35. OpenIA: gpt-3.5-turbo-instruct. <https://platform.openai.com/docs/models/gpt-3-5-turbo> (2024)

## **Detección temprana de degeneración macular asociada con la edad mediante arquitecturas basadas en Transformadores de Visión: Un estudio comparativo**

Augusto Javier Reyes-Delgado<sup>1</sup>, Jorge Ernesto González-Díaz<sup>1</sup>,  
José Luis Sánchez-Cervantes<sup>1</sup>, Yara Anahí Jiménez-Nieto<sup>2</sup>,  
Adolfo Rodríguez-Parada<sup>2</sup>, José Luis Rodríguez-Loaiza<sup>3</sup>

<sup>1</sup> Tecnológico Nacional de México,  
México

<sup>2</sup> Universidad Veracruzana,  
Facultad de Negocios y Tecnologías, Campus Ixtaczoquitlán,  
México

<sup>3</sup> Instituto de Oftalmología Conde de Valenciana,  
Departamento de retina,  
México

{M17010207,D04010291,jose.sc}@orizaba.tecnm.mx  
{yjimenez.adrodriguez}@uv.mx,  
jose.rodriguez@institutodeoftalmologia.org

**Resumen.** La Degeneración Macular Asociada con la Edad (DMAE) es una de las principales causas de pérdida de visión en personas mayores a nivel mundial y catalogada entre las primeras seis afecciones visuales en México. La dificultad de diagnosticar la DMAE en etapas iniciales debido a las sutiles características patológicas en las imágenes retinianas motiva el uso de métodos avanzados de Deep Learning, los cuales ofrecen un potencial significativo para mejorar la precisión del diagnóstico. Recientemente, las arquitecturas basadas en transformadores de visión, como Vision Transformer (ViT), Swin Transformer BERT Pre-training of Image Transformers (BEiT), han emergido, ofreciendo una nueva perspectiva en el análisis de imágenes al aprender relaciones espaciales complejas. Este estudio presenta un análisis comparativo de estas arquitecturas de transformadores de visión aplicadas a la detección de la DMAE, enfocándose en la capacidad de cada modelo para identificar y clasificar las etapas tempranas de la enfermedad. A pesar de los desafíos asociados con el tamaño reducido de los conjuntos de datos de imágenes médicas. Nuestros resultados sugieren que las arquitecturas basadas en ViT y sus derivados logran un rendimiento significativo en la detección de la DMAE, siendo el BEiT particularmente destacado por su consistencia en el desempeño superior. No obstante, es importante resaltar que el ViT mantiene una eficacia notable en la clasificación multiclase de la DMAE, con la ventaja adicional de requerir un menor consumo de recursos computacionales.

**Palabras clave:** Clasificación multiclase, degeneración macular asociada con la edad (DMAE), detección temprana, transformadores de visión.

## Early Detection of Age-related Macular Degeneration Using Vision Transformer-based Architectures: A Comparative Study

**Abstract.** Age-related macular degeneration (AMD) is one of the leading causes of vision loss in older adults worldwide and is among the top six visual impairments in Mexico. The difficulty in diagnosing AMD in its early stages, due to subtle pathological features in retinal images, motivates the use of advanced deep learning methods that offer significant potential to improve diagnostic accuracy. Recently, vision transformer architectures such as Vision Transformer (ViT) and Swin Transformer BERT Pre-training of Image Transformers (BEiT) have emerged, providing a novel perspective in image analysis by learning complex spatial relationships. This study presents a comparative analysis of these Vision Transformer architectures applied to the detection of AMD, focusing on each model's capability to identify and classify the early stages of the disease. Despite the challenges associated with the small size of medical image datasets, our results suggest that ViT-based architectures and their derivatives achieve significant performance in AMD detection, with BEiT in particular notable for its consistently superior performance. Nevertheless, it is important to highlight that ViT retains remarkable effectiveness in multi-class classification of AMD, with the added advantage of requiring fewer computational resources.

**Keywords:** Multiclass classification, age-related macular degeneration (AMD), early detection, vision transformers.

### 1. Introducción

Según la Organización Mundial de la Salud [1], la DMAE afecta significativamente la calidad de vida de las personas mayores, impactando su independencia y capacidad para realizar actividades diarias. En México la DMAE ha sido catalogada por la Secretaría de Salud como el tercero de los seis principales problemas oculares que afectan a la población [2]. La detección temprana de la DMAE es fundamental para prevenir la progresión de la enfermedad y preservar la visión. La literatura sugiere que la integración de métodos de Deep Learning, especialmente las redes neuronales convolucionales (CNN), ha mejorado el rendimiento en la detección y clasificación de imágenes médicas, superando en algunos casos a las evaluaciones hechas por especialistas [3].

El avance en el procesamiento de imágenes y la incorporación de técnicas de Deep Learning han ofrecido nuevas perspectivas para su uso en ciencias médicas. El procesamiento de imágenes utilizando CNN y sus métodos automáticos de análisis han demostrado ser herramientas de alta eficiencia, proporcionando sistemas inteligentes y amigables para el escaneo y diagnóstico de enfermedades, incluida la DMAE, fuera de un entorno clínico [4]. Además, se han propuesto diferentes enfoques para detectar las características patológicas de la DMAE utilizando imágenes de alta resolución, analizando patrones de color y textura [5]. La clasificación automática del nivel de

progresión de la enfermedad enfrenta desafíos, especialmente con características sutiles o similares a condiciones no patológicas.

Esto se complica por las altas resoluciones de imagen y la intensidad de recursos necesarios, lo que da la posibilidad de afectar la precisión del diagnóstico y ralentizar el proceso [6]. En este contexto, con base en el análisis comparativo que abordamos en este artículo consideramos que las arquitecturas de transformadores de visión como ViT [7], Swin Transformer [8] y BEiT [9] sugieren una evolución prometedora en la detección de la DMAE, ofreciendo la capacidad de entender las complejas relaciones espaciales en las imágenes retinianas. Este estudio presenta un análisis comparativo de estas arquitecturas de transformadores de visión aplicadas a la detección de la DMAE, enfocándose en la capacidad de cada modelo para identificar y clasificar las etapas de la enfermedad en No DMAE, Leve, Moderada y Avanzada. Estos modelos pueden ser clave para mejorar la detección temprana y la precisión diagnóstica de la DMAE, lo que es crucial para el tratamiento efectivo y la preservación de la visión en las poblaciones envejecidas incluyendo la de nuestro país.

El trabajo se organiza con base en las siguientes secciones: La Sección 2 muestra los trabajos relacionados. En la Sección 3 se describen los detalles sobre las arquitecturas comparadas, ViT, Swin Transformer y BEiT. En la Sección 4 se detallan los experimentos realizados y la evaluación comparativa del rendimiento de las arquitecturas incluidas en el estudio. Finalmente, la Sección 5 presenta conclusiones y trabajo futuro.

## **2. Trabajos relacionados**

Se han identificado en la literatura algunos trabajos relacionados, que, si bien no realizan una comparativa entre diferentes arquitecturas de transformadores de visión para DMAE, destacan la eficiencia de estos sobre las CNN. Entre los de mayor relevancia se encuentra el presentado en [10], que abordó la aplicación de ViT para detectar glaucoma mediante el uso de imágenes de fondo del ojo. Se evaluaron varios modelos: ViT, Swin Transformer, Twins-PCPVT y Atención de clase en transformadores de imágenes (CaiT) con algoritmos de aprendizaje con pocas imágenes y se analizó el impacto de las técnicas de aumento de datos.

Los resultados del estudio mostraron que ViT, combinado con ProtoNets, superó a las contrapartes basadas en CNN y logró un rendimiento competitivo en conjuntos de datos de referencia. Por otro lado, en [11] se evaluó la eficacia de las CNN y Sistemas Basados en ViT para detectar glaucoma en imágenes de fondo de ojo. Los autores probaron diversas arquitecturas de CNN como VGG19, ResNet50, InceptionV3 y Xception, junto con variantes de ViT como Swin Transformer y Twins-PCPVT, así como sistemas híbridos como CaiT, Transformadores de imágenes con eficiencia de datos (DeiT), Transformador de imagen mejorada por convolución (CeiT) y Transformador de visión convolucional (ConViT), y la arquitectura ResMLP. Los resultados mostraron un rendimiento similar entre CNN y ViT en el conjunto de pruebas, pero las CNN demostraron mejor generalización en conjuntos externos. Asimismo, en [12] se exploró el potencial de las arquitecturas ViT en aplicaciones de imagen médica. Se compararon las capacidades de ViT con las de las CNN en tareas como segmentación, reconocimiento y clasificación de imágenes médicas. Se

destacaron arquitecturas como Conformer, U-Net Transformer y Multi-transSP, mostrando su eficacia en la mejora de la precisión y la eficiencia en diversas aplicaciones médicas.

Los resultados mostraron que ViT superó a las CNN en la segmentación de imágenes médicas, gracias a su capacidad para modelar dependencias a largo plazo y su escalabilidad.

Además, un método para clasificar enfermedades retinianas mediante imágenes de tomografía de coherencia óptica (OCT) fue introducido en [13], utilizando una red Swin-Poly Transformer. Los hallazgos indicaron que el método propuesto facilitó una clasificación retiniana precisa y eficiente, subrayando el valor de la inteligencia artificial en diagnósticos oftalmológicos y el potencial de las redes ViT en este ámbito.

Del mismo modo, en [14] los autores se centraron en la comparación de CNNs y ViTs para la clasificación de radiografías de tórax (CXR) en casos de COVID-19, neumonía viral y casos sanos. Los autores utilizaron el conjunto de datos COVID-QU-Ex, dividiendo aleatoriamente el 80% para entrenamiento y el 20% para pruebas. Evaluaron la efectividad en casos balanceados y desbalanceados, implementaron modelos ViT como Twins, Swin y Segformer.

Los resultados mostraron que los modelos basados en CNN y ViT tenían un rendimiento comparable, con una precisión máxima del 99.82% para EfficientNetB7 (CNN) y un rendimiento destacado para SegFormer (ViT).

De manera similar en [15], se evaluó el rendimiento de las arquitecturas de ViT, específicamente ViT-B y Swin-B, en clasificación de imágenes médicas, contrastando su efectividad con los modelos basados en CNNs para diagnosticar enfermedades como enfermedades torácicas, embolia pulmonar y tuberculosis usando radiografías y tomografías computarizadas. Aquí plantearon que la inicialización adecuada es esencial para los Transformadores de Visión en el ámbito médico, y que los enfoques de autoaprendizaje que utilizan información mutua generan representaciones más precisas para la clasificación médica.

En el mismo sentido, los autores de [16] exploraron el uso de ViT, Swin Transformer y ConvNext, aplicando técnicas de transfer learning para la detección de Glaucoma a partir de imágenes del fondo de ojo. Este esfuerzo buscó crear un método automatizado que permitió identificar el Glaucoma en fases tempranas, con el fin de prevenir la ceguera.

Finalmente, en [17] Wassel et al., reportaron un estudio centrado en la clasificación de condiciones oculares glaucomatosas utilizando modelos de ViT en imágenes de fondo de ojo completas y recortadas en el disco óptico. Evaluaron las arquitecturas ViT, Swin, CaiT, crossViT, XciT, ResMlp y DeiT, tanto de forma individual como en ensamblés. Además del glaucoma, abordaron otras enfermedades oftalmológicas como diabetes, cataratas, hipertensión, miopía patológica y otras anomalías.

Los resultados mostraron que Swin y CaiT obtuvieron altos niveles de precisión, sensibilidad y especificidad en la validación y prueba de los conjuntos de datos combinados, destacando su eficacia en la detección de glaucoma en imágenes oftalmológicas, lo que sugiere su potencial utilidad en la práctica clínica.

La tabla 1 muestra los datos resumidos de los trabajos relacionados identificados en la literatura para este trabajo de investigación.



**Tabla 1.** Trabajos relacionados de comparativas de arquitecturas ViT en medicina.

Año	Autor	Arquitecturas ViT	Enfermedades	Tipo Imágenes
2023	Nurgazin M. et al [10]	Variantes del ViT clásico: ViT_tiny ViT_small ViT_base	Melanoma, Carcinoma basocelular, Carcinoma espinocelular, Nevus, Queratosis actínica, Dermatofibroma, Quiste epidermoide, Psoriasis, Dermatitis atópica, Rosácea, Cáncer de mama	Lesiones cutáneas. Biopsias de tejido mamario. Citologías cervicales
2023	Alayon S. et. al [11]	ViT, Swin Transformer, Twins-PCPVT, CaiT	Glaucoma	Fondo de Ojo
2023	Li J. et al [12]	Conformer, U-Net Transformer, Módulo Residual Transformer Multi-transSP, TransPath, i-ViT BabyNet	Predicción de peso fetal. Detección de retinopatía diabética. Segmentación de cartílago de rodilla	Ultrasonido. Resonancia magnética, Tomografía computarizada, Rayos X, Histopatología
2023	He J. et al [13]	ViT Swin Transformer	Retinopatía diabética, Edema macular diabético, Glaucoma, Anormalidades oculares	Tomografía de coherencia óptica (OCT)
2023	Nafisah S. et al [14]	Twins, Swin, Segformer	COVID-19 Neumonía	Radiografías de tórax (CXR)
2022	Ma D. et al [15]	ViT-B, Swin-B.	Enfermedades torácicas, Embolia pulmonar, Tuberculosis.	Radiografías de tórax, Tomografías computarizadas.
022	Mallick S. et al [16]	ViT, Swin Transformer, ConvNext	Glaucoma	Fondo de Ojo
2022	Wassel M. et al [17]	Cait, crossViT, XciT, ResMlp, DeiT, ViT	Glaucoma, Diabetes Cataratas, Hipertensión, Miopía Patológica	Fondo de Ojo

La Tabla 1 muestra investigaciones recientes centradas en diversas aplicaciones de arquitecturas de ViT en el campo médico, abarcando desde enfermedades de la piel hasta condiciones oculares como glaucoma y retinopatía diabética.

Sin embargo, no se observan trabajos específicos que exploren el rendimiento de los modelos ViT, Swin Transformer y BEiT para la detección o clasificación de la DMAE. Esto permite inferir que la aplicación de BEiT en el diagnóstico de DMAE es un área poco explorada, lo cual destaca la oportunidad y necesidad de investigar esta dirección, dada la importancia de la DMAE como causa significativa de pérdida de visión en la población mayor.

Con base en lo anterior, realizar una comparativa entre ViT, Swin Transformer y BEiT ofrece una perspectiva poco abordada en la literatura para el análisis de imágenes,

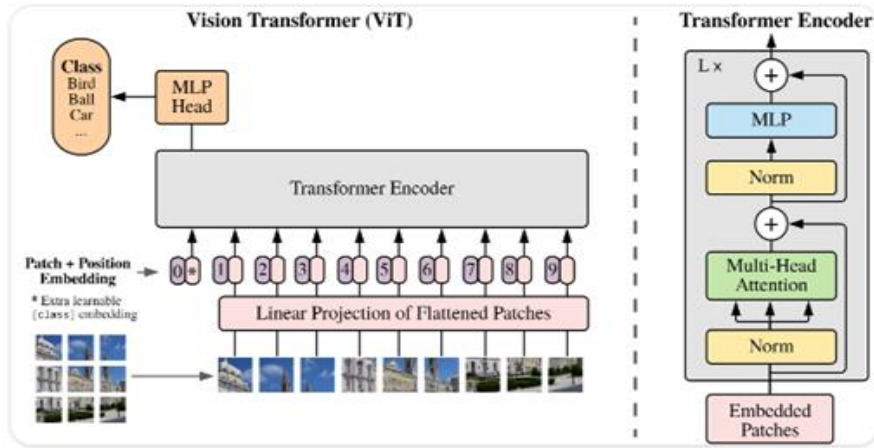


Fig. 1. Arquitectura ViT original (tomada de [7]).

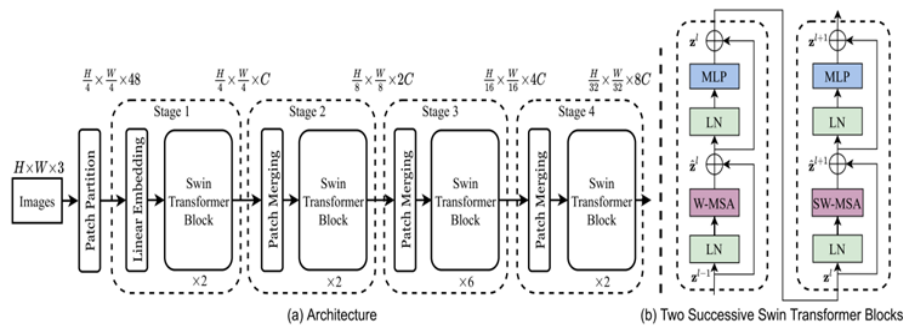


Fig. 2. Arquitectura de Swin Transformer (tomada de [8]).

aprovechando la atención global y las características jerárquicas, lo que podría mejorar significativamente la detección y clasificación de la DMAE.

### 3. Métodos

En este estudio se realiza una comparativa detallada entre las arquitecturas de ViT, Swin Transformer y BEiT para la detección y clasificación multiclase de la DMAE en imágenes de fondo del ojo. Se manejan cuatro categorías de clasificación: No DMAE, DMAE moderada, intermedia y avanzada.

Estas tecnologías de Deep Learning se seleccionaron por su potencial para procesar de manera eficaz las características visuales intrincadas, fundamentales para discernir los distintos estadios de la DMAE, lo que resulta clave para lograr un diagnóstico temprano y preciso.

La elección de estas arquitecturas se justifica por su avanzada capacidad para capturar patrones globales y locales en las imágenes, lo cual es esencial para una detección fiable y una clasificación precisa de la progresión de la DMAE.

### **3.1. Vision Transformer (ViT)**

ViT [7] es una innovadora arquitectura que aplica el mecanismo de transformadores, usual en el procesamiento del lenguaje, al campo de la visión por computadora. ViT parte las imágenes en parches y los procesa como si fueran tokens en una secuencia.

Utiliza la atención para ponderar la importancia de diferentes partes de la imagen, permitiendo al modelo captar patrones complejos y relaciones a larga distancia (Fig. 1). Su enfoque en las relaciones globales lo hace especialmente adecuado para identificar patrones en imágenes médicas, como las relacionadas con la DMAE, donde las manifestaciones de la enfermedad pueden estar sutiles y distribuidas por toda la imagen.

### **3.2. Swin Transformer**

Swin Transformer fue introducido por Ze Liu et al. en su trabajo "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" [8] en 2021. El Swin Transformer surge como una respuesta a algunas limitaciones de los modelos de transformadores puros como ViT, especialmente en términos de eficiencia computacional y la capacidad para manejar tamaños de imagen variables. Aunque ViT demostró que los transformadores podían ser poderosos para tareas de visión, su enfoque de tratar la imagen como una secuencia de parches fijos planteaba desafíos en términos de escalabilidad y adaptabilidad a diferentes resoluciones y tamaños de imagen.

El Swin Transformer introduce varios conceptos innovadores para abordar estas limitaciones:

- **Ventanas Desplazadas (Shifted Windows):** Una de las innovaciones clave del Swin Transformer es su uso de ventanas desplazadas. Divide la imagen en ventanas no superpuestas para la atención local, lo cual reduce la complejidad computacional.
- **Jerarquía:** Al igual que en las CNN, el Swin Transformer procesa las imágenes en varias resoluciones. Comienza con una alta resolución y va reduciéndola progresivamente, permitiendo al modelo capturar características a diferentes escalas y mejorar la eficiencia al reducir la resolución en las capas más profundas.
- **Flexibilidad y Generalidad:** A diferencia de ViT, que utiliza parches de tamaño fijo, el Swin Transformer puede manejar de manera más efectiva diferentes tamaños de imagen y resoluciones, lo que lo hace más flexible y adaptable para diversas aplicaciones en visión por computadora.

La selección de Swin Transformer se justifica por su diseño que aborda eficientemente la jerarquía y la localidad en imágenes. A diferencia del ViT, que considera toda la imagen de manera global, Swin Transformer procesa las imágenes en ventanas locales, lo que permite una representación más detallada de las características locales (Fig.2).

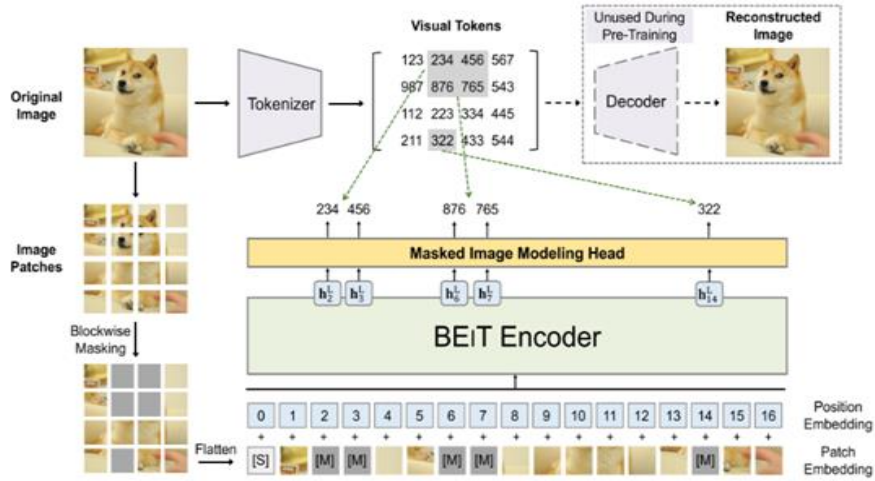


Fig. 3. Arquitectura de BEiT (tomada de [9]).

### 3.3. BEiT (BERT Pre-training of Image Transformers)

Fue presentado en un trabajo por Bao et al. [9] en 2021, titulado “BEiT: BERT Pre-training of Image Transformers”.

BEiT se inspira en el éxito de BERT (Bidirectional Encoder Representations from Transformers) en el campo de Procesamiento de lenguaje natural (NLP). BERT revolucionó NLP mediante el preentrenamiento de transformadores en grandes corpus de texto usando tareas de predicción de palabras ocultas, donde el modelo aprende a predecir partes del texto que han sido intencionalmente ocultadas. BEiT traslada este enfoque de preentrenamiento al dominio de las imágenes (Fig.3). En lugar de predecir palabras ocultas, BEiT se entrena para predecir partes ocultas de una imagen. Este proceso implica dos etapas principales:

- **Tokenización de Imágenes:** BEiT convierte una imagen en un conjunto de tokens visuales utilizando un modelo de tokenización de imágenes (como un VQ-VAE, un autoencoder variacional cuantizado). Esto resulta en una representación de la imagen en forma de tokens, similar a cómo se tokeniza un texto en NLP.
- **Preentrenamiento de Modelo:** El modelo se preentrena con la tarea de predecir los tokens visuales de partes de la imagen que han sido ocultadas, similar a la predicción de palabras faltantes en BERT. Esto enseña al modelo a entender y predecir la estructura y el contenido visual basándose en el contexto proporcionado por las partes visibles de la imagen.

El uso de BEiT en este estudio está justificado por su enfoque en el aprendizaje de representaciones visuales mediante la predicción de píxeles ocultos, lo cual representa una innovación para el análisis de imágenes de fondo de ojo en DMAE. BEiT es capaz de captar sutilezas en las texturas y patrones de las imágenes, aspectos cruciales para identificar las etapas de la DMAE.

## **4. Resultados experimentales**

Para la comparativa, se entrenaron tres modelos seleccionados, mediante un proceso de transferencia de entrenamiento (fine tuning), utilizando el mismo conjunto de datos. Se llevaron a cabo múltiples iteraciones para ajustar los hiperparámetros de cada modelo. A continuación, se detallan los aspectos más específicos de la implementación.

### **4.1. Detalles de implementación**

La implementación de los modelos ViT y Swin Transformer se realizó en Google Colab®, utilizando aceleración por GPU. En contraste, el modelo BEiT se entrenó en una computadora de escritorio con un GPU Nvidia RTX 3070 de 8GB, debido a limitaciones de recursos en Colab®. Para el entrenamiento, se empleó la biblioteca PyTorch Transformers en todos los modelos.

Para determinar los hiperparámetros óptimos, se realizaron múltiples experimentos, concluyendo que 42 épocas y lotes de 32 imágenes resultan ser los más adecuados. Se observó que incrementar el número de épocas más allá de 42 no generaba mejoras significativas en el rendimiento, identificándose una meseta en el desempeño cerca de las 20 épocas. Adicionalmente, la tasa de aprendizaje se estableció en  $5e^{-05}$  después de exhaustivas evaluaciones.

### **4.2. Conjunto de datos**

Para construir el conjunto de datos, se utilizaron inicialmente 305 imágenes, repartidas en 185 para entrenamiento y 60 para validación y pruebas, siguiendo una distribución de 60%/20%/20%. Para potenciar la generalización del modelo y reducir el sobreajuste, se aplicaron técnicas de aumento de datos al lote de entrenamiento, incluyendo cambios en tamaño, rotaciones y ajustes de brillo y contraste. Esto incrementó el conjunto a 1,094 imágenes, con 974 dedicadas al entrenamiento. Las imágenes se obtuvieron del conjunto de datos iChallenge-AMD [18] y de un conjunto en Kaggle publicado por Mujib [19], que incluye imágenes de DMAE extraídas de varios conjuntos de imágenes de fondo de ojo con patologías retinianas.

La clasificación de las imágenes se basó en la literatura existente [20,21], y posteriormente fue validada por expertos en el área médica [22].

### **4.3. Resultados**

Al concluir el entrenamiento de cada modelo, se registró la exactitud, observando un incremento hasta la 40ª época. Este desempeño fue monitoreado y documentado utilizando la plataforma Weights & Biases® (W&B), como se ilustra en la Fig.4.

Todos los modelos alcanzaron una exactitud superior al 0.7500. Para fortalecer la evaluación de los resultados, se calcularon métricas adicionales de rendimiento, incluyendo precisión, sensibilidad y F1 Score, permitiendo un análisis más profundo de las capacidades de cada modelo. Estas métricas se obtuvieron utilizando el conjunto de validación y el conjunto de pruebas previamente separado del conjunto de datos inicial. Los resultados de estas métricas se presentan en la tabla 2.

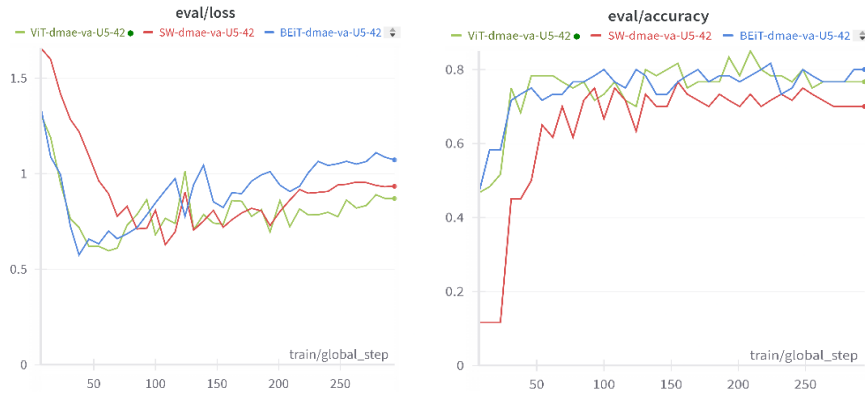


Fig. 4. Resultados obtenidos por los modelos durante el proceso de entrenamiento.

Los resultados presentados en la tabla 2 muestran diferencias notables en el rendimiento de los modelos. Al promediar los resultados de precisión de los modelos para los conjuntos de validación y prueba sobre los que fueron evaluados, se obtiene que ViT alcanza una exactitud del 0.7833, Swin Transformer obtiene el 0.7583, y BEiT llega hasta el 0.8166.

Es importante destacar que, para todos los modelos, el rendimiento en la clasificación de imágenes de fondo de ojo de las clases "leve" y "moderada" es consistente y muestra mejores resultados en comparación con las clases "No dmae" y "Avanzada", las cuales presentan una mayor varianza. Las Figs. 5 y 6 muestran las gráficas comparativas de las principales métricas aplicadas a los modelos.

#### 4.4. Discusión

En nuestro estudio comparativo, se evaluaron tres modelos avanzados: ViT, Swin Transformer y BEiT. Los resultados indican variaciones significativas en el rendimiento de cada modelo, subrayando la importancia de la selección de la arquitectura en aplicaciones clínicas.

El modelo ViT exhibió una elevada precisión en el conjunto de validación, aunque se observó una disminución en su rendimiento al evaluarlo en el conjunto de pruebas. Destacó particularmente en la clasificación de casos avanzados de degeneración macular dentro del conjunto de validación, lo cual sugiere que esta arquitectura posee una aptitud específica para la identificación de manifestaciones severas de la enfermedad. No obstante, se registró una sensibilidad reducida en la detección de casos en etapas tempranas, lo que podría reflejar una predisposición hacia la sobre clasificación en las fases más graves.

En contraste, el Swin Transformer presentó una precisión general ligeramente inferior en comparación con el ViT, especialmente notable en el conjunto de validación. En el conjunto de pruebas, esta arquitectura enfrentó retos al clasificar de manera acertada los casos avanzados, evidenciado por su baja sensibilidad y puntuación F1 en dicha categoría. Sin embargo, mostró un desempeño competitivo en la identificación de etapas iniciales de la enfermedad, lo que indica su potencial utilidad en la detección precoz de la misma.

**Tabla 2.** Resultados en las métricas de evaluación de los diferentes modelos.

Modelo	Exactitud	Conjunto	Clase	Precisión	Sensibilidad	F1-Score
ViT	0.8500	Validación	No dmae	0.8000	0.6666	0.7272
			Leve	0.8333	0.9259	0.8771
			Moderada	0.8500	0.8500	0.8500
			Avanzada	1.0000	0.7142	0.8333
	0.7166	Pruebas	No dmae	1.0000	0.5000	0.6666
			Leve	0.7187	0.8518	0.7796
			Moderada	0.6666	0.7000	0.6829
			Avanzada	0.7500	0.4285	0.5454
Swin Transformer	0.7500	Validación	No dmae	0.5555	0.8333	0.6666
			Leve	0.7777	0.7777	0.7777
			Moderada	0.8235	0.7000	0.7567
			Avanzada	0.7142	0.7142	0.7142
	0.7666	Pruebas	No dmae	0.8333	0.8333	0.8333
			Leve	0.9583	0.8518	0.9019
			Moderada	0.6666	0.7000	0.6829
			Avanzada	0.4444	0.5714	0.5000
BEiT	0.8166	Validación	No dmae	0.8000	0.6666	0.7272
			Leve	0.7931	0.8518	0.8214
			Moderada	0.8333	0.7500	0.7894
			Avanzada	0.8750	1.0000	0.9333
	0.8166	Pruebas	No dmae	1.000	0.6666	0.8000
			Leve	0.8518	0.8518	0.8518
			Moderada	0.7391	0.8500	0.7906
			Avanzada	0.8333	0.7142	0.7692

Por su parte, el BEiT demostró ser el modelo más eficaz en el conjunto de pruebas, superando a los modelos ViT y Swin Transformer en términos de precisión general. A pesar de un rendimiento inicialmente inferior al ViT en el conjunto de validación, el BEiT evidenció una consistencia notable entre ambos conjuntos y una mejora significativa en la detección de todas las etapas de la enfermedad en comparación con el ViT durante las pruebas.

Esto revela una capacidad de generalización y robustez superior, posicionando al BEiT como una alternativa prometedora para la detección práctica de la degeneración macular en sus diversas etapas.

La variabilidad en el rendimiento entre estos modelos destaca la complejidad de aplicar Deep Learning para diagnósticos médicos. Mientras que ViT y Swin Transformer ofrecen ventajas en la detección de etapas específicas de la enfermedad, BEiT muestra un balance entre sensibilidad y precisión en un rango más amplio de condiciones.

Lo anterior subraya la necesidad de considerar múltiples factores, como la exactitud, sensibilidad y especificidad, al seleccionar un modelo de Deep Learning para la detección de enfermedades oftalmológicas.

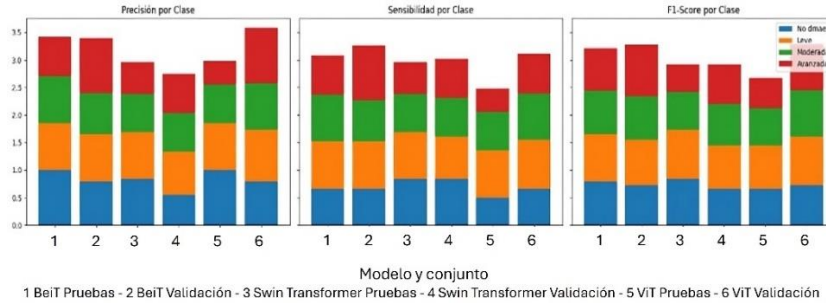


Fig. 5. Comparativa de precisión, sensibilidad y F1-score por clase de diferentes modelos.

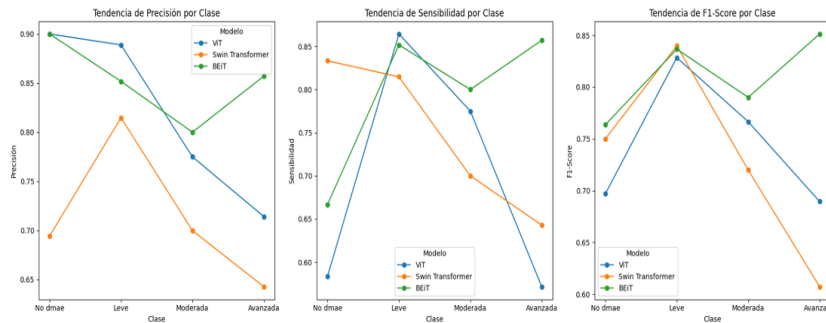


Fig. 6. Comparativa de tendencias por clase y modelo de las métricas de evaluación de los modelos.

## 5. Conclusiones

Aunque ViT sigue presentando un alto desempeño, BEiT se presenta como una alternativa más consistente en el presente caso de estudio.

Los resultados promedio de precisión para los conjuntos de validación y prueba muestran que el BEiT lidera con una exactitud del 81.66%, seguido por ViT con el 78.33%, y finalmente Swin Transformer con el 75.48%. Estas cifras reflejan no solo la capacidad de generalización de cada modelo sino también su fiabilidad en el reconocimiento de patrones asociados con condiciones oculares específicas.

Resulta particularmente interesante que los modelos muestren una consistencia en el rendimiento en la clasificación de condiciones clasificadas como “leve” y “moderada”. Este fenómeno indica que las características visuales presentes en estas etapas de la enfermedad son más distintivas y, por lo tanto, más fácilmente reconocibles por los modelos de aprendizaje profundo. Por otro lado, las categorías “No dmae” y “Avanzada” exhiben una mayor variabilidad en los resultados, lo que sugiere que las manifestaciones visuales de estas etapas pueden ser más sutiles o menos diferenciadas, dificultando así la clasificación precisa.

Los hallazgos subrayan la importancia de la selección de modelo en aplicaciones de diagnóstico médico basadas en inteligencia artificial. Aunque el BEiT supera en rendimiento general, por su balance entre precisión y capacidad de generalización, ViT



continúa mostrando un desempeño superior en determinados casos. El Swin Transformer, a pesar de tener un rendimiento ligeramente inferior, aún podría ser valioso en un contexto clínico cuando se combina con otras modalidades o como parte de un sistema de ensamble.

Nuestros resultados sugieren que, aunque no existe una solución única para la detección de todas las etapas de la degeneración macular asociada con la edad, la selección cuidadosa de la arquitectura de Deep Learning puede mejorar significativamente los resultados de diagnóstico. Futuras investigaciones deberán explorar la integración de estas arquitecturas con otras modalidades de datos y técnicas de aprendizaje, para desarrollar sistemas de diagnóstico más precisos y confiables.

## Referencias

1. Organización Mundial de la Salud: Informe mundial sobre la visión, vol. 214, no. 14 (2020)
2. Día Mundial de la Visión 2020: Secretaría de Salud. Available: <https://www.gob.mx/salud/es/articulos/dia-mundial-de-la-vision-2020?idiom=es> (2024)
3. He, T., Zhou, Q., Zou, Y.: Automatic Detection of Age-related Macular Degeneration based on deep Learning and Local outlier Factor Algorithm. *Diagnostics*, vol. 12, no. 2, pp. 532 (2022). DOI: 10.3390/DIAGNOSTICS12020532.
4. Abd-El-Khalek, A.A., Balaha, H.M., Alghamdi, N.S., Ghazal, M., Khalil, A.T., Abo-El-soud, M.E.A., El-Baz, A.: A Concentrated Machine Learning-based Classification System for Age-related Macular Degeneration (AMD) Diagnosis using Fundus Images. *Scientific Reports*, vol. 14, no. 1, pp. 2434 (2024). DOI: 10.1038/s41598-024-52131-2.
5. Leingang, O., Riedl, S., Mai, J., Reiter, G.S., Faustmann, G., Fuchs, P., Bogunović, H.: Automated deep Learning-based AMD Detection and Staging in Real-World OCT Datasets (PINNACLE study report 5). *Scientific Reports*, vol. 13, no. 1, pp. 1–13 (2023). DOI: 10.1038/s41598-023-46626-7.
6. Retina-Specialist: Deep learning for AMD screening and detection. Available: <https://www.retina-specialist.com/article/deep-learning-for-amd-screening-and-detection> (2024)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929 (2020)
8. Heigold, G., Gelly, S., Uszkoreit, J., Houtsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR'21, 9th International Conference on Learning Representations (2020). Available: <https://arxiv.org/abs/2010.11929v2>.
9. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows (2020). DOI: 10.48550/arXiv.2103.14030.
10. Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT Pre-Training of Image Transformers. ICLR'22 - 10th International Conference on Learning Representations (2021). <https://arxiv.org/abs/2106.08254v2>.
11. Nurgazin, M., Tu, N.A.: A Comparative Study of Vision Transformer Encoders and Few-shot Learning for Medical Image Classification. In Proceedings IEEE/CVF International Conference on Computer Vision Workshops, ICCVW'23, pp. 2505–2513 (2023). DOI: 10.1109/ICCVW60793.2023.00265.
12. Alayón, S., Hernández, J., Fumero, J.F., Sigut, F.J., Díaz-Alemán, T.: Comparison of the Performance of Convolutional Neural Networks and Vision Transformer-Based Systems for Automated Glaucoma Detection with Eye Fundus Images. *Applied Sciences*, vol. 13, no. 23, pp. 12722 (2023). DOI: 10.3390/app132312722.

13. Li, J., Chen, J., Tang, Y., Wang, C., Landman, B.A., Zhou, S.K.: Transforming Medical Imaging with Transformers? A Comparative Review of Key Properties, current Progresses, and Future Perspectives. *Medical Image Analysis*, vol. 85 (2023). DOI: 10.1016/j.media.2023.102762.
14. He, J., Wang, J., Han, Z., Ma, J., Wang, C., Qi, M.: An Interpretable Transformer Network for the Retinal Disease Classification using Optical Coherence Tomography. *Sci Rep*, vol. 13, no. 1 (2023). DOI: 10.1038/s41598-023-30853-z.
15. Nafisah, S.I., Muhammad, G., Hossain, M.S., AlQahtani, S.A.: A Comparative Evaluation between Convolutional Neural Networks and Vision Transformers for COVID-19 Detection. *Mathematics*, vol. 11, no. 6 (2023). DOI: 10.3390/math11061489.
16. Ma, D. A., Hosseinzadeh-Taher, M.R., Pang, J., Islam, N.U., Haguigui, F., Gotway, M.B., Liang, J.: Benchmarking and Boosting Transformers for Medical Image Classification, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, pp. 12–22 (2022). DOI: 10.1007/978-3-031-16852-9\_2.
17. Mallick, S., Paul, J., Sengupta, N., Sil, J.: Study of Different Transformer based Networks For Glaucoma Detection. In *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, Institute of Electrical and Electronics Engineers Inc. (2022). DOI: 10.1109/TENCON55691.2022.9977730.
18. Wassel, M., Hamdi, A.M., Adly, N., Torki, M.: Vision Transformers Based Classification for Glaucomatous Eye Condition. In *Proceedings International Conference on Pattern Recognition*, Institute of Electrical and Electronics Engineers Inc., pp. 5082–5088 (2022). DOI: 10.1109/ICPR56361.2022.9956086.
19. Baidu: Research Open-Access Dataset - download. Available: <https://ai.baidu.com/broad/download> (2024)
20. Mujib, R.: ARMD curated dataset 2023, Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/rakhshandamujib/armd-curated-dataset-2023> (2023)
21. National Eye Institute: U.S. Department of Health and Human Services, Age-related macular degeneration (AMD). <https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/age-related-macular-degeneration> (2024)
22. Al-Zamil, W., Yassin, S.: Recent Developments in Age-related Macular Degeneration: A review. *Clin Interv Aging*, vol. 12, pp. 1313–1330 (2017). DOI: 10.2147/CIA.S143508.
23. CONDE: Investigación – Unidad de Investigación. Available: <https://www.condeinvestigacion.org/> (2023)

## **Análisis comparativo de técnicas de aprendizaje automático para la detección de la enfermedad de Parkinson mediante el reconocimiento de voz**

Justo Francisco León Pastor Baak, Giner Alor Hernández,  
Nancy Aracely Cruz-Ramos, José Luis Sánchez Cervantes,  
Lisbeth Rodríguez Mazahua

Tecnológico Nacional de México, Instituto Tecnológico de Orizaba,  
División de Estudios de Posgrado e Investigación,  
México

{M22011021, giner.ah, jose.sc.dci,  
lisbeth.rm}@orizaba.tecnm.mx, ncruz@ito-depi.edu.mx

**Resumen.** La enfermedad de Parkinson es una de las enfermedades neurodegenerativas más prevalentes a nivel mundial siendo aproximadamente el 1% en mayores de 60 años; su progresión es gradual y sin cura conocida. En este trabajo se presenta un análisis comparativo de los algoritmos de inteligencia artificial usados para la detección temprana de la enfermedad de Parkinson mediante el reconocimiento de voz. Para este análisis se consideraron 5 conjuntos de datos y 11 algoritmos de aprendizaje automático. Como resultado de este análisis, se obtuvo que los algoritmos Bagged SVM, Random Forest y Gradient Boosting presentaron los mejores resultados en sensibilidad, siendo aplicados en el conjunto de datos Parkinson's Disease Classification.

**Palabras claves:** Bagged SVM, enfermedad de Parkinson, procesamiento de lenguaje natural, random forest, reconocimiento de voz.

### **Comparative Analysis of Machine Learning Techniques for Detecting Parkinson's Disease Using Speech Recognition**

**Abstract.** Parkinson's disease is one of the most prevalent neurodegenerative diseases worldwide being approximately 1% in people over 60 years old; its progression is gradual and without known cure. This paper presents a comparative analysis of artificial intelligence algorithms used for early detection of Parkinson's disease using speech recognition. For this analysis, 5 datasets and 11 machine learning algorithms were considered. As a result, Bagged SVM, Random Forest and Gradient Boosting algorithms presented the best results in recall, being applied to the Parkinson's Disease Classification dataset.

**Keywords:** Bagged SVM, natural language processing, Parkinson's disease, random forest, speech recognition.

## 1. Introducción

La enfermedad de Parkinson es un proceso neurodegenerativo complejo de aparición en la edad adulta, constituye la segunda enfermedad neurodegenerativa más frecuente por detrás de la enfermedad de Alzheimer [1], los síntomas abarcan, desórdenes motores como; temblores, bradicinesia, rigidez, inestabilidad postural, deterioro cognitivo, bradifenia, hiponimia, disartria, y otros [2].

También se presentan desórdenes del discurso y lenguaje como: mudez, afasia, velocidad rápida, volumen reducido, monótono, mono sonoridad, disartria hipocinética, tartamudeo adquirido, y otros desordenes [3].

Algunas estadísticas como la de la Organización Mundial de la Salud, informan que la enfermedad de Parkinson, está en constante aumento, las estimaciones mundiales indican que en el 2019 más de 8,5 millones de personas fueron afectadas por la enfermedad de Parkinson, lo que provocó el fallecimiento de 329 000 de personas en el año 2020, esto destaca la urgente necesidad de estrategias más efectivas para su diagnóstico temprano [4].

Martínez *et al.* [5] señalan que, en México, la enfermedad de Parkinson aumentó la incidencia de 6.7 en 2014 a 14.9 por 100 000 habitantes en 2023.

Hoy en día esta enfermedad se puede diagnosticar implementando algoritmos de inteligencia artificial, mediante el análisis del movimiento de un paciente [6], análisis de magneto encefalografías [7], análisis de electroencefalogramas [8] e irregularidades en la señal de voz analizando fonemas del paciente [9-19].

Por lo que la importancia de las técnicas de inteligencia artificial en la detección de enfermedades es ampliamente aceptada en la comunidad científica y médica, ya que también, estas herramientas posibilitan la automatización y optimización de los procesos diagnósticos y terapéuticos lo que conduce a una presentación de servicios de salud más eficiente y efectiva, dentro de estas técnicas se encuentran los algoritmos de aprendizaje automático, que es la ciencia (y arte) de programar computadoras para que puedan aprender de datos [10].

Actualmente existen diversos estudios sobre la aplicación del aprendizaje automático en el diagnóstico de la enfermedad de enfermedades con altos porcentajes de éxito en la precisión de su clasificación.

En este trabajo, se presenta un análisis de los algoritmos de aprendizaje automático para el diagnóstico de la enfermedad de Parkinson a través del reconocimiento de voz.

La estructura de este artículo es la siguiente: en la sección 2 se presenta el estado del arte de los trabajos relacionados referente a los algoritmos y técnicas de Procesamiento de Lenguaje Natural para la extracción de elementos vocales para la detección de la temprana de la enfermedad de Parkinson. La sección 3 especifica la descripción de los (*Data Set*, Conjunto de datos) utilizados y sus características para este análisis. En la sección 4 se profundiza en los resultados del análisis y evaluación realizada a los conjuntos de datos con técnicas de inteligencia artificial previamente seleccionadas. En la sección 5 se encuentra la discusión de los resultados obtenidos y finalmente en la sección 6 se presentan las conclusiones y el trabajo futuro.

## 2. Estado del arte

En esta sección se presentan los estudios previos que utilizaron técnicas de aprendizaje automático en el reconocimiento de voz para la detección de la enfermedad de Parkinson, destacan desarrollos como NeuroSpeech de Orozco-Arroyave *et al.* [11], y modelos LSTM Bidireccionales propuestos por Quan *et al.* [12], también se identificaron herramientas prometedoras con hasta un 92% de precisión.

Orozco-Arroyave *et al.* [11] desarrollaron "NeuroSpeech", un software para evaluar deficiencias del habla en pacientes con Parkinson, con el uso del algoritmo SVM (*Support Vector Machine*, Máquinas de Vector Soporte), se identificaron alteraciones en fonación, articulación, prosodia e inteligibilidad y se demostró una reducción en la inteligibilidad, con un 83% de exactitud en la pronunciación y 50% de similitud, sugiriendo ser una herramienta prometedora en la detección de la enfermedad de Parkinson en el reconocimiento de voz.

Quan *et al.* [12] propusieron un modelo LSTM *Bidirectional* (*Long Short-Term Memory*, Red de Gran Memoria de Corto Plazo Bidireccional) para detectar Parkinson mediante análisis de señales de voz, que reveló cambios específicos como aumento de (*Jitter*, Fluctuación) y (*Shimmer*, Brillo), con el uso de técnicas como MFCC (*Mel Frequency Cepstral Coefficients*, Coeficientes Cepstrales en las Frecuencias de Mel) y PCA (*Principal Component Analysis*, Análisis de Componentes Principales) para clasificación. Klempf *et al.* [13] se enfocaron en la clasificación del diagnóstico de la enfermedad de Parkinson basados en sus señales del discurso, usando los métodos de reconocimiento de patrones AdaBoost, y (*Bagged Trees*, Árboles Embolsados), Quadratic SVM y KNN (*Nearest Neighbors*, k-Vecinos más Cercanos) obteniendo un promedio de 82.3% de exactitud y un promedio de 0.88 en el AUC (*Area Under Curve*, Área Bajo la Curva).

Rahman *et al.* [14] propusieron un marco de trabajo para diagnosticar la enfermedad de Parkinson utilizando datos de 726 pacientes de EE.UU. y otros lugares, utilizando los algoritmos SVM, XGBoost (*eXtreme Gradient Boosting*, Potenciación de Gradientes Extremo), LightGBM (*Light - Gradient Boosting Machine*, Luz - Máquina de Potenciación de Gradiente) y RF (*Random Forest*, Bosque Aleatorio) obteniendo 0.753 de AUC con XGBoost. Almeida *et al.* [15] adoptaron un enfoque, centrado en características como el lenguaje nativo y la recolección de datos mediante pruebas grabadas en lituano. Se obtuvo una clasificación de la enfermedad de Parkinson por voz con 92% de exactitud. Younis Thanoun *et al.* [16] investigaron un enfoque para identificar enfermos de Parkinson usando el clasificador SMOTE en una distribución de clases desequilibradas, obteniendo un 96%.52 de exactitud. Zhang *et al.* [17] desarrollaron una aplicación móvil con enfoque en voz, donde SVM y SVR (*Support Vector Regressor*, Regresión de Vectores de Soporte) mostraron el mejor rendimiento para diagnosticar y evaluar la severidad del Parkinson, lograron un 97.03% de sensibilidad en la clasificación. Karan *et al.* [18] propusieron un nuevo enfoque para detectar la enfermedad de Parkinson en la señal de voz basado en descomposición en modo intrínseco tomando en cuenta los bancos de filtros diádicos que capturan eficazmente la dinámica en la señal de la voz, obtuvieron un 100 % de exactitud utilizando el algoritmo de clasificación RF. Quan *et al.* [19] propusieron un modelo de aprendizaje profundo para detectar la enfermedad de Parkinson a partir de señales de voz. Se utilizaron *2Dimensional CNN* (*Convolutional Neural Networks*, Redes

Neuronales Convolucionales) para extraer características dinámicas de series de tiempo y 1Dimensional CNN para capturar dependencias, logrando el 92% de precisión. Hoq *et al.* [20] propusieron 2 modelos híbridos basados en SVM con un PCA y 1 con codificador automático escaso para detectar pacientes con la enfermedad de Parkinson basándose en sus atributos vocales, donde se obtuvo el 93% en exactitud. Khan *et al.* [21] desarrollaron un modelo que clasifica la enfermedad de Parkinson a través de la escala UPDRS (*Unified Parkinson's Disease Rating Scale*, Escala Unificada para la Evaluación de la Enfermedad de Parkinson) aplicando el algoritmo de SVM utilizando la escala Guttman para realizar sistemáticamente la correlación entre variables cualitativas y su orden en el índice UPDRS, obteniendo un 92% de exactitud.

Wroge *et al.* [22] aplicaron diferentes modelos de aprendizaje automático para clasificar la enfermedad de Parkinson usando el conjunto de datos llamado mPower y la biblioteca de Python, PyAudio, de esta manera obtuvieron como resultado 89% de AUC con RF. Soumaya *et al.* [23] decodificaron la señal del habla de la vocal /a/ para clasificar pacientes con la enfermedad de Parkinson, el método propuesto fue un sistema de clasificación que consistió en las 5 fases: conversión de señal, preprocesamiento, extracción de atributos, selección de atributos, y clasificación, en donde el mejor resultado fue obtenido por SVM con un 91.8% de exactitud.

Hyder *et al.* [7] investigaron la conectividad funcional en las redes corticales durante el procesamiento de lenguaje hablado, utilizando MNE-Python 0.19 para analizar magnetoencefalogramas, y LR (*Logistic Regression*, Regresión logística) para la clasificación, logrando un 78% de precisión en la detección de Parkinson mediante el análisis de verbos de acción. Mostafa *et al.* [24] evaluaron el desempeño de 3 métodos para diagnosticar la enfermedad de Parkinson: DT (*Decision Tree*, *Árbol de Decisión*), NV (*Naive Bayes*, Bayes Ingenuo) y NN (*Neuronal Network*, Red Neuronal), el estudio encontró que el algoritmo DT tuvo el mejor desempeño, con un 91,63% de precisión, los resultados sugieren que DT es el método más preciso para diagnosticar la enfermedad de Parkinson. Nissar *et al.* [9] evaluaron el rendimiento del modelo de aprendizaje automático XGBoost para el diagnóstico de la enfermedad de Parkinson mediante la voz, el estudio utilizó un conjunto de datos de voz de 756 pacientes y 64 sujetos sanos, los resultados mostraron que XGBoost obtuvo la exactitud más alta de 88.15%, seguido de RF y DT con 84.86% de exactitud, el modelo de clasificación NB obtuvo la exactitud más baja del 74.34%, en general.

A continuación, se describen los conjuntos de datos de reconocimiento de voz para la detección de la enfermedad de Parkinson reportados en la literatura para su análisis.

### 3. Conjuntos de datos para la identificación del Parkinson

En esta sección se describen los distintos conjuntos de datos utilizados para el análisis en la identificación de la enfermedad de Parkinson mediante el reconocimiento de voz. Se utilizaron 5 conjuntos de datos, cada uno con distintos enfoques y características:

- **Parkinson's Disease Data Set** [25]: Es un conjunto de datos, que se diseñó en [26] para determinar si un paciente se encuentra sano o tiene enfermedad de Parkinson tomando en cuenta los atributos relacionados con la detección de la disfonía, estos

atributos se conforman, 195 registros de fonaciones vocales de 31 hombres y mujeres, de los cuales 23 fueron diagnosticados con la enfermedad de Parkinson.

- **Parkinson Speech Dataset with Multiple Types of Sound Recordings** [27]: Se diseñó en [28] para clasificar el registro de voz de un paciente como sano o con la enfermedad de Parkinson utilizando 2 algoritmos de clasificación: SVM y KNN, siendo el algoritmo de SVM el que obtuvo un mejor rendimiento general en la clasificación, contiene atributos relacionados con la disfonía, los datos de entrenamiento pertenecen a 20 pacientes con la enfermedad de Parkinson y 20 sujetos sanos, de todos los sujetos, se tomaron 26 tipos de grabaciones de sonidos, de las cuales se extrajo un grupo de 26 características lineales y basadas en frecuencia de tiempo, puntuación UPDRS, en total el conjunto de datos contiene 1040 instancias de enteros reales.
- **Parkinson's Disease Classification** [29]: Se diseñó en [30] con el objetivo de usar la TQWT (*Tunable Q-factor Wavelet Transform*, Transformada Ondícula de Factor Q Sintonizable) a las señales de voz de pacientes con la enfermedad de Parkinson para la extracción de características, que tiene una resolución de frecuencia más alta que la clásica transformada ondícula discreta. La extracción de características del conjunto de datos se basó en los métodos usados en el diagnóstico de los desórdenes vocales en la enfermedad de Parkinson. Los datos se recopilaron de 188 pacientes con enfermedad de Parkinson (107 hombres y 81 mujeres) con edades comprendidas entre 33 y 87 años ( $65,1 \pm 10,9$ ) en el Departamento de Neurología de la Facultad de Medicina de CerrahpaSa, Universidad de Estambul, el grupo control estuvo formado por 64 individuos sanos (23 hombres y 41 mujeres) con edades que oscilan entre los 41 y los 82 años ( $61,1 \pm 8,9$ ), durante el proceso de recolección de datos, el micrófono se configuró a 44,1 KHz y luego del examen del médico, se recolectó la fonación sostenida de la vocal /a/ de cada sujeto con 3 repeticiones, el conjunto de datos contiene 758 instancias de enteros reales, organizados por base (26), Intensidad (3), Ancho de banda y Formantes (8), MFCC (84), WT (*Wavelet transform, Transformada Wavelet*) aplicada a F0 (182), Pliegue vocal (22), TQWT (432) con un total de 756 tributos.
- **Parkinson Replicated Acoustic Features Dataset** [31]: Se utilizó en [32] con el objetivo de ser usado por un sistema capaz de discriminar automáticamente personas sanas de personas con la enfermedad de Parkinson partiendo de múltiples grabaciones de la voz de los mismos sujetos. Los atributos se utilizaron en un enfoque de BLR (*Bayesian Linear Regression*, Regresión Lineal Bayesiana) para manejar mediciones replicadas y tiempo. Estos atributos se identificaron como los mejores para alimentar el modelo de predicción por el método LASSO (*Least Absolute Shrinkage and Selection Operator*, Operador de Selección y Contracción Mínima Absoluta), que minimiza el error cuadrado sujeto a la restricción no diferenciable expresada en términos de la norma L1 de los coeficientes. El conjunto de datos contiene características extraídas de 3 grabaciones de voz replicadas, por lo que no se puede utilizar cada fila de forma independiente, ya que es 1 de las 3 réplicas de 1 individuo, existen 240 instancias, pero solo para 80 sujetos, 40 de ellos con enfermedad de Parkinson, por lo que no son independientes.

**Tabla 1.** Comparación de los conjuntos de datos para el diagnóstico de la enfermedad de Parkinson.

Conjunto de datos	# de atributos	# de registros	Descripción de los datos
Parkinson's Data Set	24	195	Datos de amplitud, frecuencia y ruido en los componentes tonales de la voz
Parkinson Multiple Sound Recording	29	1040	Datos de irregularidad en la frecuencia, amplitud, y número de frenos en la voz
Parkinson's Disease Classification	756	758	Datos de irregularidad en la frecuencia, amplitud, y de la transformada ondícula de factor Q sintonizable.
Parkinson Replicated Acoustic Features Dataset	240	48	Datos de irregularidad en la frecuencia, amplitud, y de la relación de excitación glótica / ruido
Parkinson's telemonitoring	22	5875	Datos de irregularidad en la frecuencia, amplitud, e índice UPDRS

- **Parkinson's telemonitoring** [33]: Contiene atributos relacionados con el diagnóstico de la disfonía en sujetos con la enfermedad de Parkinson. Este diagnóstico se realizó a través del uso de fonaciones de vocales sostenidas durante el desarrollo del artículo [34], donde se logró determinar si un registro contiene rasgos de síntomas vocales pertenecientes a la enfermedad de Parkinson, a través de analizar la señal, extraer rasgos representando las características de la señal, y mapear estos rasgos a la escala UPDRS usando métodos de regresión, el estudio se inició asumiendo que el deterioro en el rendimiento del habla es debido a la enfermedad de Parkinson y no a otra patología, los datos se obtuvieron de 5923 señales, usando un micrófono montado en la cabeza a 5 cm de los labios del paciente, el conjunto de datos se compone de una serie de mediciones de la voz de 42 personas con enfermedad de Parkinson en fase temprana, reclutadas para un ensayo de 6 meses, de un dispositivo de telemonitorización para la monitorización remota de la progresión de los síntomas, en donde las grabaciones se capturaron automáticamente en los hogares de los pacientes, las columnas de la tabla contienen el número de sujeto, la edad del sujeto, el sexo del sujeto, el intervalo de tiempo desde la fecha de reclutamiento inicia, UPDRS motor, UPDRS total y 16 medidas de voz biomédicas en donde cada fila corresponde a una de las 5,875 grabaciones de voz de estas personas siendo el objetivo principal de los datos predecir las puntuaciones motoras y totales de UPDRS a partir de las 16 medidas de voz, los datos están en formato ASCII CSV y las filas del archivo CSV contienen una instancia correspondiente a una grabación de voz.

En la tabla 1 se muestran características como: nombre del conjunto de datos, número de atributos, cantidad de registros y una breve descripción de los datos.

A continuación, se muestran los algoritmos de aprendizaje automático, de tipo (*Boosting*, Impulso) y (*Bagging*, Ensamble) utilizados en el análisis de los conjuntos de datos, así como su configuración y el rendimiento de las métricas seleccionadas.



**Tabla 2.** Comparación de las métricas de los algoritmos implementados en el conjunto de datos Parkinson's Data Set.

Algoritmo	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score	AUC
SVM	87.18	90.63	93.55	62.50	92.06	0.8185
KNN	76.92	92.31	77.42	75.00	84.21	0.8972
NB	61.54	100.00	51.61	100.00	68.09	0.8790
LR	82.05	90.00	87.10	62.50	88.52	0.8347
DT	76.92	95.83	74.19	87.50	83.64	0.8085
AdaBoost	87.18	93.33	90.32	75.00	91.80	0.9637
Gradient Boosting	82.05	92.86	83.87	75.00	88.14	0.9194
XGBoost	87.18	96.43	87.10	87.50	91.53	0.9395
RF	79.49	89.66	83.87	62.50	86.67	0.8931
Bagged DT	76.92	89.29	80.65	62.50	84.75	0.8992
Bagged SVM	84.62	87.88	93.55	50.00	90.63	0.8306

**Tabla 1.** Comparación de las métricas de los algoritmos implementados en el conjunto de datos Parkinson Multiple Sound Recording.

Algoritmo	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score	AUC
SVM	63.46	60.17	71.00	56.48	65.14	0.6925
KNN	59.13	56.64	64.00	54.63	60.09	NA
NB	53.85	51.47	70.00	38.89	59.32	NA
LR	58.17	55.75	63.00	53.70	59.15	0.6519
DT	61.06	59.79	58.00	63.89	58.88	0.6094
AdaBoost	62.02	60.82	59.00	64.81	59.90	0.6769
Gradient Boosting	63.94	62.89	61.00	66.67	61.93	0.6996
XGBoost	66.35	65.31	64.00	68.52	64.65	0.7339
RF	65.38	64.89	61.00	69.44	62.89	0.7133
Bagged DT	60.10	58.95	56.00	63.89	57.44	0.5994
Bagged SVM	63.46	60.17	69.00	59.26	64.79	0.6897

#### 4. Análisis comparativo de técnicas de aprendizaje automático

En esta sección se detalla el análisis realizado a los conjuntos de datos revisados con distintas técnicas de inteligencia artificial, las cuales aplican técnicas de aprendizaje automático. Dentro de los algoritmos de aprendizaje automático se utilizaron SVM, KNN, NB, LR, DT, Dentro de los algoritmos de ensamble de tipo Boost: AdaBoost, (*Boostraping*, Autoensamble), (*Gradient Boosting*, Potenciación de Gradiente), XGBoost, y dentro de los algoritmos de ensamble de tipo *Bagging* se utilizaron RF, Bagged DT, Bagged SVM. Para llevar a cabo el análisis, se utilizó el lenguaje de programación Python en su versión 3.12 para realizar los scripts.

Como primer paso se importó la biblioteca Pandas, para manipular los datos de forma organizada y eficiente, después, se determinó la forma del df (Dataframe,

**Tabla 4.** Comparación de resultados obtenidos en el conjunto de datos Parkinson’s Disease Classification.

Algoritmo	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score	AUC
SVM	80.92	88.99	85.09	68.42	87.00	0.8802
KNN	86.18	86.61	96.49	55.26	91.29	0.9535
NB	73.03	90.11	71.93	76.32	80.00	0.7946
LR	85.53	89.66	91.23	68.42	90.43	0.8996
DT	76.92	95.83	74.19	87.50	83.64	0.8085
AdaBoost	90.13	91.60	95.61	73.68	93.56	0.9148
Gradient Boosting	87.50	87.40	97.37	57.89	92.12	0.9578
XGBoost	90.13	90.24	97.37	68.42	93.67	0.9663
RF	90.13	89.60	98.25	65.79	93.72	0.9675
Bagged DT	88.82	89.43	96.49	65.79	92.83	0.9481
Bagged SVM	88.16	86.36	100.00	52.63	92.68	0.9224

**Tabla 5.** Comparación de resultados obtenidos en el conjunto de datos Parkinson Replicated Acoustic Features Dataset.

Algoritmo	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score	AUC
SVM	68.75	70.83	68.00	69.57	69.39	0.8035
KNN	85.42	87.50	84.00	86.96	85.71	0.8904
NB	87.50	88.00	88.00	86.96	88.00	0.8765
LR	75.00	76.00	76.00	73.91	76.00	0.8383
DT	77.08	85.00	68.00	86.96	75.56	0.7748
AdaBoost	77.08	76.92	80.00	73.91	78.43	0.7878
Gradient Boosting	83.33	83.33	83.33	82.61	83.33	0.8591
XGBoost	85.42	87.50	84.00	86.96	85.71	0.8852
RF	87.50	88.00	88.00	86.96	88.00	0.8983
Bagged DT	81.25	81.27	81.25	78.26	81.23	0.8583
Bagged SVM	75.00	75.09	75.00	69.57	74.91	0.7965

Estructura de Datos Bidimensional), se obtuvo la información sobre el total de columnas y nombres de las columnas, se verifico si existían valores nulos y cuáles eran los tipos de datos dentro del df.

Como segundo paso, se filtraron las características importantes de uno de cada conjunto de datos, excluyendo aquellas características cuya presencia no aporta valor al resultado, después se asignaron los atributos relacionados con la predicción y el atributo target.

Continuando, como tercer paso, los conjuntos de datos se dividieron en un 80% de entrenamiento y un 20% de pruebas, para después estandarizar los valores con la

**Tabla 2.** Comparación de las métricas de los algoritmos implementados en el conjunto de datos Parkinson's telemonitoring.

Algoritmo	Exactitud	Precisión	Sensibilidad	Especificidad	F1-Score	AUC
SVM	57.28	60.71	27.52	84.01	37.87	0.6488
KNN	70.72	70.46	65.65	75.28	67.97	0.7727
NB	54.38	56.94	14.75	89.98	23.43	0.5943
LR	57.96	58.24	39.39	74.64	47.00	0.6470
DT	65.28	64.12	60.43	69.63	62.22	0.6503
AdaBoost	53.19	58.33	3.78	97.58	07.09	0.5635
Gradient Boosting	67.15	68.89	55.76	77.38	61.63	0.7575
XGBoost	70.64	70.81	64.57	76.09	67.54	0.7943
RF	71.66	72.34	64.93	77.71	68.44	0.7933
Bagged DT	68.51	71.04	56.47	79.32	62.93	0.7604
Bagged SVM	57.87	60.00	32.91	80.29	42.51	0.6504

función `StandardScaler()` de la biblioteca `Scikit-Learn`, la cual se utiliza para preprocesar datos antes de ajustar un modelo de aprendizaje automático, estandarizando características, eliminando la media y escalando a la varianza unitaria. Hasta este paso la configuración es la misma para todos los algoritmos de inteligencia artificial, en el cuarto paso, se crearon los modelos de aprendizaje automático, de *Boosting* y *Bagging*. Para todos los algoritmos, se obtuvieron las siguientes métricas: (Exactitud, *Accuracy*), (*Precision* Precisión), (*Recall*, Sensibilidad), (*Specificity*, Especificidad), F1 score y el AUC.

Para evaluar el rendimiento del modelo de clasificación al momento de predecir a que clase pertenece la instancia de datos, se utilizó la matriz de confusión y otras métricas por ser ampliamente aceptadas en la literatura.

A continuación, las tablas 2-6 presentan una comparación de las métricas obtenidas por los algoritmos de inteligencia artificial implementados a los conjuntos de mayor uso en los trabajos relacionados: Parkinson's Data Set, Parkinson Multiple Sound Recording, Parkinson's Disease Classification, Parkinson Replicated Acoustic Features Data Set y Parkinson telemonitoring, en donde los resultados de las métricas, se representan como valores de porcentaje, con excepción del AUC ya que es una medida entre 0 y 1.

En la tabla 2 se presenta una evaluación comparativa de 11 algoritmos de Inteligencia Artificial aplicados al conjunto de datos Parkinson's Disease Data Set, en donde se analizan métricas clave como exactitud, precisión, sensibilidad, especificidad, F1-Score y AUC para determinar el algoritmo con mejor desempeño en la detección de la enfermedad de Parkinson.

Con base en los resultados de la tabla 2, se concluye que el algoritmo XGboost presenta el mejor desempeño general para la detección de la enfermedad de Parkinson. La tabla 3 presenta una evaluación comparativa al conjunto de datos Parkinson Speech Dataset with Multiple Types of Sound Recordings.

**Tabla 7.** Comparación de los resultados obtenidos por los algoritmos de aprendizaje automático en cada conjunto de datos.

Dataset	Algoritmo destacado	Exactitud	Precisión	Sensibilidad	Especificidad	Índice F1	AUC
Parkinson's Data Set	SVM	87.18	90.63	93.55	62.50	92.06	0.8185
	Bagged SVM	84.62	87.88	93.55	50.00	90.63	0.8306
	AdaBoost	87.18	93.33	90.32	75.00	91.80	0.9637
Parkinson Multiple Sound Recording	SVM	63.46	60.17	71.00	56.48	65.14	0.6925
	NB	53.85	51.47	70.00	38.89	59.32	NA
	Bagged SVM	63.46	60.17	69.00	59.26	64.79	0.6897
Parkinson's Disease Classification	Bagged SVM	88.16	86.36	100.00	52.63	92.68	0.9224
	RF	90.13	89.60	98.25	65.79	93.72	0.9675
	Gradient Boosting	87.50	87.40	97.37	57.89	92.12	0.9578
Parkinson Replicated Acoustic Features	RF	87.50	88.00	88.00	86.96	88.00	0.8983
	NB	87.50	88.00	88.00	86.96	88.00	0.8765
	XGBoost	85.42	87.50	84.00	86.96	85.71	0.8852
Parkinson's telemonitoring	KNN	70.72	70.46	65.65	75.28	67.97	0.7727
	RF	71.66	72.34	64.93	77.71	68.44	0.7933
	XGBoost	70.64	70.81	64.57	76.09	67.54	0.7943

En la tabla 3, se observa que el algoritmo XGBoost presenta el mejor desempeño general en términos de exactitud (66.35%), precisión (65.31%), F1-Score (64.65%) y AUC (73.39%), estos resultados sugieren que XGBoost es el algoritmo más efectivo para la detección de la enfermedad de Parkinson a partir de grabaciones de voz. En la tabla 4 continúa la evaluación de los algoritmos y el análisis de los resultados del conjunto de datos Parkinson's Disease Classification.

En la tabla 4 se observa un rendimiento notable de varios algoritmos, XGBoost, RF y Bagged SVM destacan por su alta precisión (90.24%, 89.60% y 86.36% respectivamente), F1-Score (93.67%, 93.72% y 92.24%) y AUC (0.9663, 0.9675 y 0.9224). En la tabla 5 se presenta la evaluación del conjunto de datos Parkinson Replicated Acoustic Features Dataset.

En la tabla 5 RF, NB y KNN destacan con un 88% y 88-86% F1-Score, NB y *Gradient Boosting* sobresalen con 88%-83% de sensibilidad, mientras que NB, KNN y XGBoost presentan un buen equilibrio entre precisión 88-87% y sensibilidad 88-84% según el AUC 89-88%. En la tabla 6 se muestran los resultados de la evaluación del conjunto de datos Parkinson's telemonitoring.

En la Tabla 6, se observa un rendimiento notable del algoritmo RF con 72.34% de precisión y 68.44% de F1-Score, NB presenta la sensibilidad más alta del 89.98%, pero con una precisión baja del 56.94%. La tabla 7 muestra una comparación de las métricas obtenidas por los algoritmos de aprendizaje automático destacados en la métrica de sensibilidad, implementados en todos los conjuntos de datos.

En el conjunto de datos Parkinson's Disease Data Set, AdaBoost sobresale con un 90.32% de sensibilidad, seguido de SVM con un 93.55%. Para Parkinson Speech Dataset with Multiple Types of Sound Recordings, SVM (71.00%) y NB (70.00%) son los más sensibles, para Parkinson's Disease Classification, Bagged SVM (100.00%) y RF (98.25%) lideran en sensibilidad. RF (88.00%) y NB (88.00%) nuevamente se destacan en Replicated Acoustic Features Dataset, mientras que en Parkinson's telemonitoring, KNN (65.65%) y RF (64.93%) son los más sensibles.

## 5. Discusión

Los resultados obtenidos durante el análisis realizado arrojaron un rango de sensibilidad de 50% al 100% de sensibilidad, la sensibilidad de los algoritmos de inteligencia artificial, es crucial para diagnosticar la enfermedad de Parkinson porque determina la capacidad del modelo para detectar correctamente los casos positivos de la enfermedad, independientemente, el rendimiento de cada modelo varía significativamente según el conjunto de datos utilizado. Este resultado podría deberse a la calidad o naturaleza de los datos, la cantidad de datos de entrenamiento y la complejidad del modelo utilizado. Ahora, profundizando más en los resultados obtenidos, se tiene que los 3 algoritmos con mejor rendimiento en la métrica de sensibilidad fueron, Bagged SVM 100%, RF 98%, y *Gradient Boosting* 97%, el rendimiento de exactitud en la clasificación de paciente con enfermedad de Parkinson obtuvo un 87%, 88% y 90% respectivamente, en el conjunto de datos Parkinson's Disease Classification, mientras que para el conjunto de datos Parkinson's Data Set tuvo valores arriba del 84% de exactitud con los algoritmos SVM, Bagged SVM, y AdaBoost; para el resto de los conjuntos de datos (Parkinson Multiple Sound Recording, Parkinson Replicated Acoustic Features, Parkinson's telemonitoring) fueron los menos sensibles, arrojando resultados entre 90-93%. Por último, considerando los resultados de este análisis, se determina que los atributos más óptimos para la detección de la enfermedad de Parkinson son la combinación de atributos con, 1) datos de irregularidad en la frecuencia ya que se estima que con un porcentaje de irregularidad superior a 1.040 % ya existiría patología y 2) Datos de irregularidad en la

amplitud (Shimmer) ya que se estima que una voz es patológica a partir de un porcentaje de irregularidad del 3.810.

## 6. Conclusiones

La falta de un diagnóstico temprano y accesible para la detección de la enfermedad de Parkinson evita que las personas con esta enfermedad reciban el tratamiento adecuado a tiempo, generando problemas en su entorno social, laboral y personal. En este trabajo se analizaron diferentes conjuntos de datos con distintas técnicas de aprendizaje automático, de *Boosting* y *Bagging*, para identificar que algoritmos ofrecen los mejores resultados, así como también los atributos de la voz más relacionados para la identificación de la enfermedad de Parkinson, durante el desarrollo de este análisis se detectó que la exactitud en la detección puede variar significativamente según el conjunto de datos utilizado y las técnicas de inteligencia artificial utilizadas en este trabajo. Los algoritmos con mayor puntaje obtenido para esta métrica fueron Bagged SVM, RF, *Gradient Boosting*, aplicados sobre el conjunto de datos Parkinson's Disease Classification, donde los resultados arrojaron un rango entre un 87% y 90% de exactitud. El análisis de la voz con aprendizaje automático y colaboración médica permite la detección temprana del Parkinson de forma precisa y eficiente. En resumen, este análisis presenta resultados prometedores, demostrando que las técnicas de aprendizaje automático, *Boosting* y *Bagging* son útiles en la detección de la enfermedad de Parkinson, pero se necesita un conjunto de datos con suficiente información y características específicas para realizar una correcta identificación. Como trabajo a futuro se considera el uso de más conjuntos de datos similares o diferentes para complementar la información, también la evaluación de métricas en otros lenguajes de programación como R.

**Agradecimientos.** Este trabajo de investigación fue patrocinado por el Consejo Nacional de Humanidades, Ciencia y Tecnología de México (CONAHCYT) y la Secretaría de Educación Pública (SEP) de México a través del programa PRODEP. Adicionalmente, se agradece al Tecnológico Nacional de México (TecNM) por apoyar este proyecto.

## Referencias

1. Martínez-Fernández, R., Gasca-Salas, C., Sánchez-Ferro, Á., Obeso, J.Á.: Actualización en la enfermedad de Parkinson. *Revista Médica Clínica Las Condes*, vol. 27, pp. 363–379 (2016). DOI: 10.1016/j.rmcl.2016.06.010.
2. Jankovic, J.: Parkinson's disease: Clinical Features and Diagnosis. *Journal of neurology, Neurosurgery & Psychiatry*, vol. 79, no. 4, pp. 368–376 (2008)
3. Archive: Bradley's neurology in clinical practice. <https://archive.org/details/bradleysneurolog00unse> (2024)
4. WHO: Enfermedad de Parkinson. <https://www.who.int/es/news-room/fact-sheets/detail/parkinson-disease> (2024)
5. Martínez-Ramírez, D., Rodríguez-Violante, M., Velázquez-Ávila, E.S., Cervantes-Arriaga, A., González-Cantú, A., Corona, T., Velásquez-Pérez, L.: Incidencia y distribución

- geográfica de la enfermedad de Parkinson en México. *Salud Pública de México*, vol. 62, no. 6, pp. 873–875 (2020). DOI: 10.21149/11750.
6. Jiang, H., Lim, W.Y.B., Ng, J.S., Wang, Y., Chi, Y., Miao, C.: Towards Parkinson's Disease Prognosis using Self-supervised Learning and Anomaly Detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing-Proceedings*, pp. 3960–3964 (2021). DOI: 10.1109/ICASSP39728.2021.9414840.
  7. Hyder, R., Jensen, M., Højlund, A., Kimppa, L., Bailey, C.J., Schaldemose, J.L., Shtyrov, Y.: Functional Connectivity of Spoken Language Processing in Early-stage Parkinson's disease: An MEG Study. *NeuroImage: Clinical*, vol. 32, pp. 102718 (2021). DOI: 10.1016/j.nicl.2021.102718.
  8. Nogales, A., García-Tejedor, Á.J., Maitín, A.M., Pérez-Morales, A., Del Castillo, M.D., Romero, J.P.: BERT Learns from Electroencephalograms about Parkinson's Disease: Transformer-based Models for Aid Diagnosis. *IEEE Access*, vol. 10, pp. 101672–101682 (2022). DOI: 10.1109/ACCESS.2022.3201843.
  9. Nissar, I., Rizvi, D.R., Masood, S., Mir, A.N.: Voice-based Detection of Parkinson's Disease through Ensemble Machine Learning Approach: A Performance Study. *EAI endorsed Transactions on Pervasive Health and Technology*, vol. 5, no. 19, pp. e2–e2 (2019). DOI: 10.4108/eai.13-7-2018.162806.
  10. Géron, A.: *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc. (2022)
  11. Orozco-Arroyave, J.R., Vásquez-Correa, J.C., Vargas-Bonilla, J.F., Arora, R., Dehak, N., Nidadavolu, P.S., Noeth, E.: NeuroSpeech: An Open-source Software for Parkinson's Speech Analysis. *Digital Signal Processing*, vol. 77, pp. 207–221 (2018). DOI: 10.1016/j.dsp.2017.07.004.
  12. Quan, C., Ren, K., Luo, Z.: A Deep Learning based Method for Parkinson's Disease Detection using Dynamic Features of Speech. *IEEE* vol. 9, pp. 10239–10252 (2021). DOI: 10.1109/ACCESS.2021.3051432.
  13. Klempíř, O., Krupička, R.: Machine Learning using Speech Utterances for Parkinson Disease Detection. *Lékař a technika-Clinician and Technology*, vol. 48, no. 2, pp. 66–71 (2018)
  14. Rahman, W., Lee, S., Islam, M.S., Antony, V.N., Ratnu, H., Ali, M.R., Hoque, E.: Detecting Parkinson Disease using a Web-based Speech Task: Observational Study. *Journal of Medical Internet Research*, vol. 23, no. 10, pp. e26305 (2021). DOI:10.2196/26305.
  15. Almeida, J.S., Rebouças-Filho, P.P., Carneiro, T., Wei, W., Damaševičius, R., Maskeliūnas, R., de Albuquerque, V.H.C.: Detecting Parkinson's Disease with Sustained Phonation and Speech Signals using Machine Learning Techniques. *Pattern Recognition Letters*, vol. 125, pp. 55–62 (2019). DOI: 10.1016/j.patrec.2019.04.005.
  16. Thanoun, M.Y., Yaseen, M.T., Aleesa, A.M.: Development of Intelligent Parkinson Disease Detection System based on Machine Learning Techniques using Speech Signal. *International Journal on Advanced Science Engineering and Information Technology*, vol. 11, no. 1, pp. 388–392 (2021). DOI: 10.18517/ijaseit.11.1.12202.
  17. Zhang, L., Qu, Y., Jin, B., Jing, L., Gao, Z., Liang, Z.: An Intelligent Mobile-enabled System for Diagnosing Parkinson Disease: Development and Validation of a Speech Impairment Detection System. *JMIR Medical Informatics*, vol. 8, no. 9, pp. e18689 (2020). DOI: 10.2196/18689.
  18. Karan, B., Sahu, S.S., Mahto, K.: Parkinson Disease Prediction using Intrinsic Mode Function based Features from Speech Signal. *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 249–264 (2020). DOI: 10.1016/j.bbe.2019.05.005.

19. Quan, C., Ren, K., Luo, Z., Chen, Z., Ling, Y.: End-to-end deep Learning Approach for Parkinson's disease Detection from Speech Signals. *Biocybernetics and Biomedical Engineering*, vol. 42, no. 2, pp. 556–574 (2022). DOI: 10.1016/j.bbe.2022.04.002.
20. Hoq, M., Uddin, M.N., Park, S.B.: Vocal Feature Extraction-based Artificial Intelligent Model for Parkinson's Disease Detection. *Diagnostics*, vol. 11, no. 6, pp. 1076 (2021). DOI: 10.3390/diagnostics11061076.
21. Khan, T., Westin, J., Dougherty, M.: Classification of Speech Intelligibility in Parkinson's disease. *Biocybernetics and Biomedical Engineering*, vol. 34, no. 1, pp. 35–45 (2014). DOI: 10.1016/j.bbe.2013.10.003.
22. Wroge, T.J., Özkanca, Y., Demiroglu, C., Si, D., Atkins, D.C., Ghomi, R.H.: Parkinson's disease Diagnosis using Machine Learning and Voice. In *IEEE Signal Processing in Medicine and Biology Symposium*, pp. 1–7 (2018). DOI: 10.1109/SPMB.2018.8615607.
23. Soumaya, Z., Taoufiq, B.D., Benayad, N., Yunus, K., Abdelkrim, A.: The Detection of Parkinson disease using the Genetic Algorithm and SVM classifier. *Applied Acoustics*, vol. 171, pp. 107528 (2021). DOI: 10.1016/j.apacoust.2020.10752.
24. Mostafa, S.A., Mustapha, A., Khaleefah, S.H., Ahmad, M.S., Mohammed, M.A.: Evaluating the Performance of three Classification Methods in Diagnosis of Parkinson's disease. In *Recent Advances on Soft Computing and Data Mining: Proceedings of the Third International Conference on Soft Computing and Data Mining*, Springer International Publishing, vol. 700, pp. 43–52 (2018). DOI: 10.1007/978-3-319-72550-5\_5.
25. Kaggle: Parkinson's Disease Data Set, <https://www.kaggle.com/datasets/vikasukani/parkinsons-disease-data-set> (2024)
26. Little, M., McSharry, P., Hunter, E., Spielman, J., Ramig, L.: Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's disease. *Nature Proceedings*, pp. 1015–1022 (2009)
27. ICS: Parkinson Speech Dataset with Multiple Types of Sound Recordings - UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/301/parkinson+speech+dataset+with+multiple+types+of+sound+recordings> (2024)
28. Sakar, B.E., Isenkul, M.E., Sakar, C.O., Sertbas, A., Gurgen, F., Delil, S., Kursun, O.: Collection and Analysis of a Parkinson Speech Dataset with Multiple Types of Sound recordings. *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 828–834 (2013). DOI: 10.1109/JBHI.2013.2245674.
29. ICS: Parkinson's Disease Classification - UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/470/parkinson+s+disease+classification> (2024)
30. Sakar, C.O., Serbes, G., Gunduz, A., Tunc, H.C., Nizam, H., Sakar, B.E., Apaydin, H.: A Comparative Analysis of Speech Signal Processing Algorithms for Parkinson's Disease Classification and the Use of the Tunable Q-factor Wavelet Transform. *Applied Soft Computing*, vol. 74, pp. 255–263 (2019). DOI: 10.1016/j.asoc.2018.10.022.
31. ICS: Parkinson Dataset with replicated acoustic features - UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/489/parkinson+dataset+with+replicated+acoustic+features> (2024)
32. Naranjo, L., Pérez, C.J., Martín, J.: Addressing Voice Recording Replications for Tracking Parkinson's disease Progression. *Medical & Biological Engineering & Computing*, vol. 55, pp. 365–373 (2017). DOI: 10.1007/s11517-016-1512-y.
33. ICS: Parkinsons Telemonitoring - UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/189/parkinsons+telemonitoring> (2024)
34. Tsanas, A., Little, M., McSharry, P., Ramig, L.: Accurate Telemonitoring of Parkinson's Disease Progression by Non-invasive Speech Tests. *Nature preceding* (2010). DOI: 10.1038/npre.2009.3920.1.



## Modelo secuencial de lógica difusa para la predicción de riesgo de muerte súbita cardíaca en niños con predisposición a SQTL

Oscar Vélez-Mora<sup>1</sup>, Ulises Juárez-Martínez<sup>1</sup>,  
Alberto Alfonso Aguilar Lasserre<sup>1</sup>, Ignacio López-Martínez,  
Beatriz Alejandra Olivares Zepahua<sup>1</sup>,  
Norma Alicia Balderrábano-Saucedo<sup>2</sup>

<sup>1</sup> Tecnológico Nacional de México, Instituto Tecnológico de Orizaba,  
División de Estudios de Posgrado e Investigación,  
México

<sup>2</sup> Laboratorio de Investigación en Cardiopatías Congénitas y Arritmias,  
Hospital Infantil de México Federico Gómez,  
México

{M08011407, ulises.jm, alberto.al, ignacio.lm,  
beatriz.oz}@orizaba.tecnm.mx, nbalderrabano@himfg.edu.mx

**Resumen.** La Muerte Súbita Cardíaca (MSC) se refiere a la interrupción brusca e inesperada de toda actividad cardíaca. En niños, dos grupos de enfermedades aumentan la probabilidad de MSC: las estructurales y las eléctricas, siendo el Síndrome de QT Largo (SQTL) recurrente en la población infantil mexicana. Este estudio busca diagnosticar el riesgo de MSC en niños mediante un modelo de lógica difusa secuencial. Se utilizan 4 variables en el primer modelo y 11 en el segundo, relacionadas con electrocardiogramas, síntomas e historial del paciente. Las salidas en el modelo ECG (electrocardiograma) son alto, medio y bajo riesgo; en el modelo clínico, son probable, poco probable y definitivo. Para la validación se utilizó una muestra de 69 pacientes del Hospital Infantil de México Federico Gómez, alcanzando una precisión del 97.7%.

**Palabras clave:** Lógica difusa, muerte súbita cardíaca, inteligencia artificial.

### Sequential Fuzzy Logic Model for Predicting the Risk of Sudden Cardiac Death in Children Predisposed to LQTS

**Abstract.** The Sudden Cardiac Death (SCD) refers to the abrupt and unexpected cessation of all cardiac activity. In children, two groups of diseases increase the likelihood of SCD: structural and electrical, with Long QT Syndrome (LQTS) being recurrent in the Mexican pediatric population. This study aims to diagnose the risk of SCD in children using a sequential fuzzy logic model. Four variables

are used in the first model and eleven in the second, related to electrocardiograms, symptoms, and patient history. The outputs in the ECG (electrocardiogram) model are high, medium, and low risk; in the clinical model, they are probable, unlikely, and definitive. For validation, a sample of 69 patients from the Federico Gómez Children's Hospital in Mexico was used, achieving an accuracy of 97.7%.

**Keywords:** Fuzzy logic, sudden cardiac death, artificial intelligence.

## 1. Introducción

La Muerte Súbita Cardíaca (MSC) es un padecimiento que afecta a millones de personas en el mundo y se demostró que es la causa del 20% al 30% de las muertes de origen cardíaco en personas adultas [1]. Sin embargo, en México no se reporta un organismo encargado de llevar estadísticas precisas sobre este padecimiento y mucho menos en la población pediátrica. En muchas ocasiones, las muertes por este suceso se catalogan como “causas desconocidas”. En 2015, el INEGI [2] reportó 3,842 decesos por causas no definidas (R95-R99), de estos números 514 corresponden a MSC, lo cual genera una problemática en cuanto al registro y seguimiento del padecimiento. Diversos estudios demuestran que las muertes por este padecimiento en niños están asociadas a dos grupos de enfermedades, las cuales se clasifican como padecimientos estructurales y eléctricos [1], siendo la MCD (Miocardiopatía Dilatada) y SQT (Síndrome de QT Largo) las más frecuentes [1]. No obstante, la falta de conocimiento por parte de los médicos de primer contacto y las personas en general hace que estos padecimientos pasen desapercibidos y los niños no reciban un tratamiento adecuado. Por otro lado, las investigaciones demuestran que la predicción de la muerte súbita o la identificación de los padecimientos que predisponen a la misma es posible mediante técnicas de inteligencia artificial, siendo la lógica difusa [3] y el aprendizaje automático [4] aplicado a electrocardiogramas electrónicos dos de las más destacadas [5]. Por lo anteriormente mencionado, en este trabajo se propone crear un algoritmo basado en lógica difusa para la predicción de riesgo de muerte súbita cardíaca en la población infantil de México.

## 2. Trabajos relacionados

En [6] los autores presentaron un modelo de predicción de riesgo de arritmias cardíacas mediante una combinación de lógica difusa y redes neuronales. Las variables de entrada fueron colesterol, presión arterial, actividad física, tabaquismo, edad, índice de masa corporal y la diabetes. Alcanzando una precisión del 99,3 % en la detección de enfermedades coronarias superando a los clasificadores tradicionales KNN y SVM.

En [7] se propuso el desarrollo de un sistema experto aplicando lógica difusa para respaldar el diagnóstico de enfermedades coronarias en Nigeria, se emplearon 12 variables de entrada, como edad, presión arterial, glucosa, colesterol, ritmo cardíaco, entre otras, la variable resultante fue el diagnóstico de riesgo, categorizado como bajo, medio, moderado y severo alcanzando una precisión, sensibilidad y especificidad del 94,55%, 95,35% y 95,00% respectivamente.

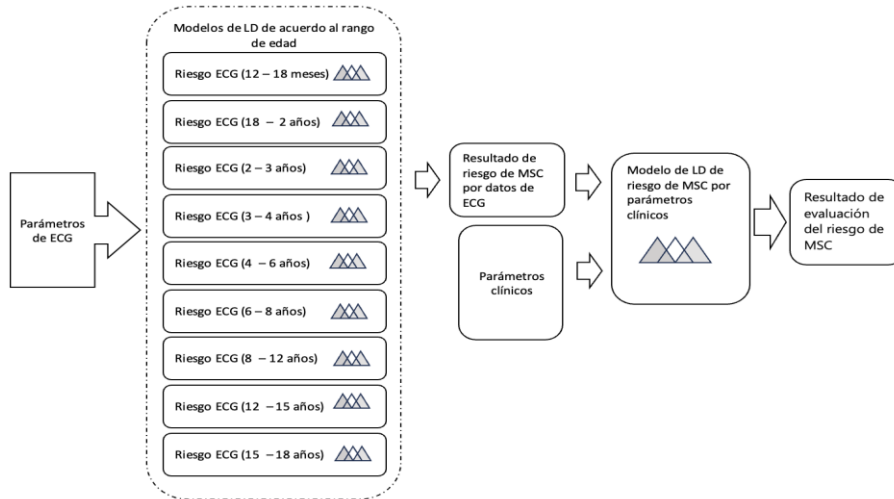


Fig. 1. Modelo secuencial de lógica difusa propuesto.

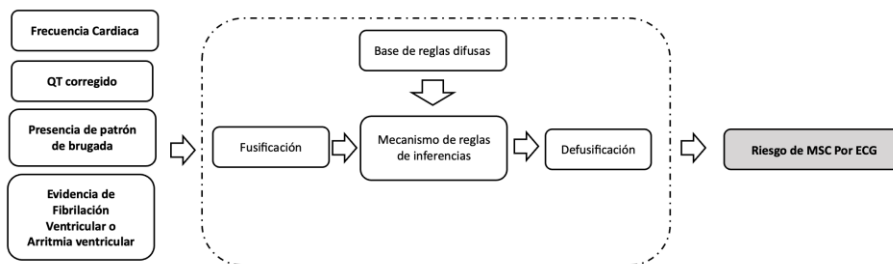


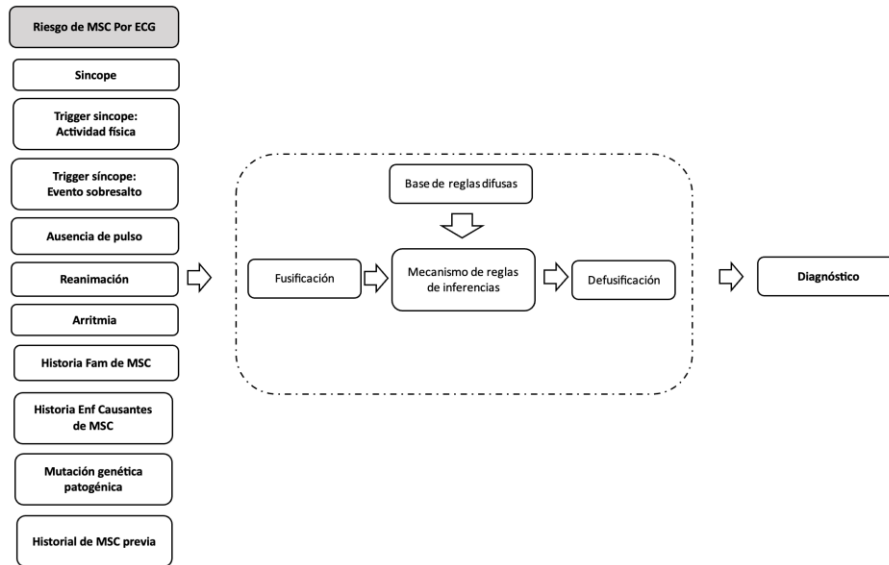
Fig. 2. Modelo secuencial de lógica difusa para predecir el riesgo de MSC por datos de ECG.

Por otro lado, Arslankaya et al. [8] diseñaron un modelo de lógica difusa que incorpora parámetros como la edad, género, niveles de colesterol, concentración de glucosa, presencia de angina inducida por el ejercicio, entre otros.

Se emplearon un total de 576 reglas de inferencia en la construcción del modelo. La efectividad del modelo se puso a prueba en un grupo de 10 pacientes y, mediante un análisis de regresión múltiple, se determinó que la relación entre las variables de entrada y salida.

En [9], los autores determinaron los factores relacionados con eventos de arritmias peligrosas en un estudio multicéntrico que involucró a un total de 13 instituciones. En este estudio se incluyeron a 72 pacientes con Tetralogía de Fallot (TOF) con taquicardia ventricular sostenida, una arritmia vinculada a la muerte súbita cardíaca. Los resultados finales utilizando Random Forest mostraron una sensibilidad del 54% (IC del 95%: del 44% al 63%) y una especificidad del 86% (IC del 95%: del 83% al 89 %).

Asimismo, Kasbe et al. [10] presentaron un modelo de diagnóstico para la predicción de enfermedades cardíacas utilizando Lógica Difusa. En este estudio, se utilizaron 13 parámetros de entrada, destacando la presión arterial, el colesterol, el nivel de azúcar



**Fig. 3.** Modelo secuencial de lógica difusa para predecir el riesgo de MSC por datos clínicos e historial del paciente.

**Tabla 1.** Relación de frecuencias cardiacas representado por rango de edades.

Edad	Frecuencia cardiaca
12 a 18 meses	88 - 156
18 a 24 meses	82 -149
2 – 3 años	76 - 142
3 – 4 años	70 - 136
4 – 6 años	65 - 131
6 – 8 años	59 - 123
8 – 12 años	52 - 115
12 – 15 años	47 - 108
15 – 18 años	43 - 104

en la sangre, el tipo de dolor torácico y la frecuencia cardíaca máxima, entre otros. Este modelo logró una precisión del 93.33%.

En [11] los autores propusieron un Sistema de Inferencia Neuro-difusa Adaptativa (ANFIS) para identificar seis afecciones cardíacas diferentes, que incluyen el bloqueo de rama derecha, contracción auricular prematura, contracción ventricular prematura, entre otras. Las variables de entrada utilizadas incluyeron la amplitud de la onda Q, el intervalo QRS, entre otras, logrando una especificidad media del 96.67%.

De igual forma Chugh et al. [12] presentaron un modelo de aprendizaje híbrido que combina algoritmos de aprendizaje automático como Random Forest, Gradient Boosting, eXtreme Gradient Boosting, entre otros, junto con un sistema experto difuso

**Tabla 2.** Variables de entrada del modelo de lógica difusa de datos de ECG (Modelo de niños de 12 meses).

Atributo (Factor de riesgo)	Conjuntos difusos	Valores
Frecuencia cardiaca (lpm)	Taquicardia	>140
	Normal	112 – 132
	Bradicardia	<103
QT corregido (ms)	Corto	<330
	Normal	370 – 470
	Largo	>480
Presencia de patrón de Brugada	Si	1
	No	0
FV o AV	Si	1
	No	0

**Tabla 3.** Variables de salida del modelo de lógica difusa por datos de ECG.

Nivel de riesgo por parámetros	Puntaje
Alto	>70
Medio	45 – 65
Bajo	<40

para la detección de enfermedades vasculares. Lograron una precisión del 91,2% en la UCI de Cleveland y del 91,7% en Framingham.

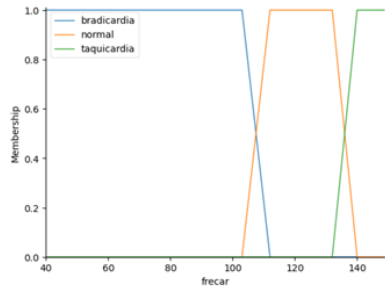
### 3. Software y herramientas utilizados

Para el diseño del modelo de lógica difusa se utilizó Python 3.11.0, JupyterLab 4.0.11, la biblioteca scikit-fuzzy 0.2, pandas 2.2.1, matplotlib y NumPy. Para el gráfico de superficie de respuesta, se empleó la biblioteca Plotly. Los requerimientos de hardware recomendados son: un procesador x86 con una velocidad de reloj de al menos 1.5 GHz y al menos dos núcleos físicos, junto con 4 GB de memoria RAM.

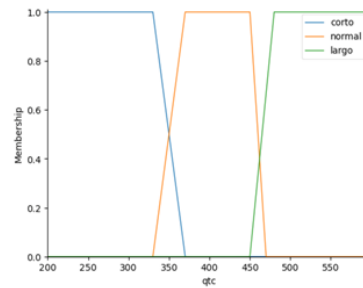
### 4. Modelo de lógica difusa propuesto

La lógica difusa (LD) permite la toma de decisiones basada en términos subjetivos expresados a través de variables lingüísticas [13]. Estas variables se valoran, modelan, fuzzifican, analizan mediante un mecanismo de inferencia y finalmente se defuzzifican. En el contexto de este estudio, se presenta un modelo secuencial de lógica difusa tipo Mamdani que recopila el conocimiento de diferentes expertos en muerte súbita cardiaca en edad pediátrica, a partir de dos modelos especializados en dos áreas de interés.

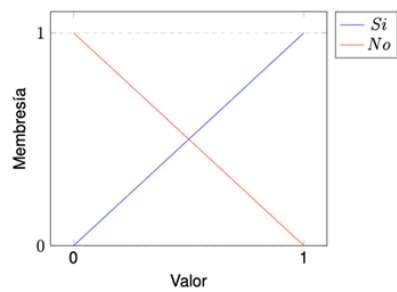
El primer modelo se enfoca en la recepción de variables relacionadas con electrocardiogramas, ya que diversos estudios [14,15] han demostrado que el ECG es una herramienta de gran utilidad en el diagnóstico de enfermedades asociadas a la MSC. Por otro lado, el segundo modelo utiliza el resultado del primer modelo sumado a una



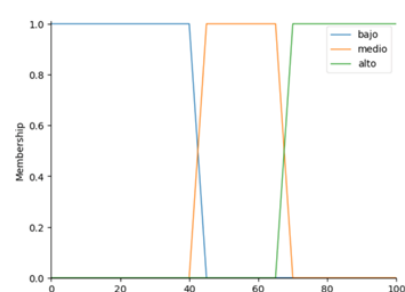
**Fig. 4.** Variable de frecuencia cardiaca (niños de 18 meses).



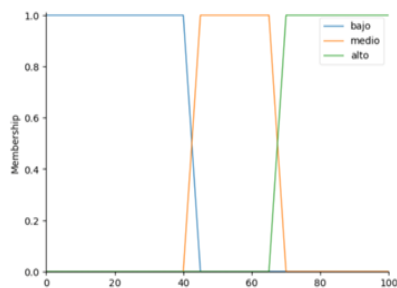
**Fig. 5.** Variable QTc.



**Fig. 6.** Variable patrón de Brugada y evidencia de FV o AV.



**Fig. 7.** Variable de salida Riesgo por ECG.



**Fig. 8.** Variables por riesgo por ECG.



**Fig. 9.** Variables binarias.

serie de parámetros clínicos identificados por especialistas en el diagnóstico de niños con enfermedades que predisponen a la MSC, incluyendo el síncope [16], taquicardia, historia familiar de MSC, entre otros. Dado que el objetivo principal del primer modelo es predecir el riesgo de MSC en la población infantil, es necesario contar con los valores de frecuencia cardíaca asociados a la edad del niño, ya que la estructura anatómica del corazón varía de acuerdo a la edad [17].

Para abordar este problema, se propone replicar este modelo en rangos específicos (universos del discurso) que se detallan en la Tabla 1. Es por ello que para la predicción

de riesgo de MSC por medio de ECG se necesitan un total de 9 modelos. En la Figura 1 se muestra el proceso general del modelo secuencial de lógica difusa propuesto, en la figura 2 se presenta a detalle el modelo por parámetros de ECG y en la figura 3 el modelo de lógica difusa por parámetros clínicos.

#### **4.1. Selección de variables de entrada y salida para el modelo de predicción por valores de ECG**

Las variables asociadas al modelo de predicción mediante electrocardiogramas se presentan en la tabla 2, las cuales fueron seleccionadas y revisadas con el respaldo de un panel de expertos en enfermedades cardíacas pediátricas y muerte súbita cardíaca. Entre estas variables se destacan el QT corregido, el patrón de Brugada y la presencia de Fibrilación Ventricular o Arritmia Ventricular debido a su valor predictivo en otras investigaciones [18-20].

#### **4.2. Variables de salida**

En la tabla 3 se muestran las variables de salida asociadas al modelo de predicción por ECG el cual tiene un puntaje acorde al nivel de riesgo.

#### **4.3. Selección de variables de entrada y salida por parámetros clínicos**

Los parámetros clínicos utilizados en este modelo abarcan síntomas asociados a la MSC en niños. Estas variables fueron seleccionadas con base en la experiencia de un panel de expertos en enfermedades asociadas a la MSC y se presentan en la Tabla 4.

#### **4.4. Variables de salida**

En la Tabla 5 se muestran las variables de salida relacionadas con el modelo de predicción basado en parámetros clínicos, el cual asigna un puntaje correspondiente al nivel de riesgo.

#### **4.5. Función de membresía de los datos de electrocardiograma**

La función de membresía de cada variable se describió en detalle en la tabla 2.

1. Frecuencia cardíaca: Esta variable se representa con 3 valores lingüísticos que incluyen taquicardia, frecuencia normal y bradicardia. Estos conjuntos difusos se muestran en la figura 4.
2. QT corregido. Es una variable que presenta tres valores representados por términos lingüísticos: corto, normal y largo. Estos conjuntos difusos se visualizan en la figura 5.

**Tabla 4.** Variables de entrada del modelo de lógica difusa por parámetros clínicos.

Variable Lingüística	Conjuntos difusos	Valores
Riesgo de MSC por ECG	Bajo	< 40
	Medio	45 - 65
	Alto	>70
Síncope	Si	1
	No	0
Ausencia de pulso	Si	1
	No	0
Reanimación o desfibrilación	Si	1
	No	0
Taquicardia o Arritmia	Si	1
	No	0
Historia Familiar de MSC	Si	1
	No	0
Historia Familiar de enf. asociadas a MSC	Si	1
	No	0
Diagnóstico de enfermedad asociada a MSC	Si	1
	No	0
Mutación genética patogénica	Si	1
	No	0
Evento de MSC en el pasado	Si	1
	No	0

**Tabla 5.** Variables de salida del modelo de lógica difusa parámetros clínicos.

Nivel de riesgo por parámetros clínicos	Puntaje
<b>Definitivo</b>	<b>&gt;70</b>
Probable	45 – 65
Poco probable	<40

3. Presencia de patrón de Brugada y Evidencia de FV o AV. Tienen 2 valores representados por un valor numérico 1 para sí y 0 para no. Los conjuntos mencionados se observan en la figura 6.
4. Variable de salida Riesgo de MSC por ECG consta de 3 conjuntos difusos alto, medio y bajo, representados en la figura 7.

#### 4.6. Función de membresía de los datos clínicos

Variables de entrada del modelo de LD de datos clínicos.

1. Variable de entrada: riesgo por ECG, consta de 3 conjuntos difusos: alto, medio y bajo. Se describe a detalle en la tabla 4, y los conjuntos difusos se aprecian en la figura 8.



**Tabla 6.** Ejemplo de reglas de inferencia del modelo de lógica difusa por datos de ECG.

No de regla	AV o FV	Patrón de Brugada	QTc	FC	Riesgo ECG
R1	si	si	corto	bradicardia	alto
R2	si	si	corto	normal	alto
R3	si	si	corto	taquicardia	alto
R4	si	si	normal	bradicardia	alto
R28	no	no	normal	bradicardia	medio
R29	no	no	normal	normal	bajo

**Tabla 7.** Ejemplo de reglas de inferencia del modelo de lógica difusa por datos clínicos.

NR <sup>a</sup>	R <sup>b</sup>	S <sup>c</sup>	A <sup>d</sup>	R <sup>e</sup>	A <sup>f</sup>	H <sup>g</sup>	HE <sup>h</sup>	D <sup>i</sup>	TA <sup>j</sup>	T <sup>k</sup>	M <sup>l</sup>	MP <sup>m</sup>	D <sup>n</sup>
R1	alto	sí	sí	sí	sí	sí	sí	sí	sí	sí	si	sí	definitivo
R2	alto	sí	sí	sí	sí	sí	sí	sí	sí	sí	si	no	definitivo
R3	alto	sí	sí	sí	sí	sí	sí	sí	sí	sí	no	sí	definitivo
R4	alto	no	sí	sí	sí	sí	sí	sí	sí	no	no	sí	probable
R5	alto	sí	no	no	no	no	no	no	no	no	no	no	probable
R6	bajo	no	no	no	sí	no	no	no	no	no	no	no	poco probable
R7	bajo	no	no	no	no	no	no	no	no	no	no	no	poco probable

2. Variables binarias. La función de membresía de cada variable se describe en detalle en la tabla 4. Las variables de entrada son de tipo binario, correspondiendo a 1 para sí y 0 para no, y se representan en la figura 9.

Variable de salida del modelo de LD de datos clínicos.

El diagnóstico consta de 3 conjuntos difusos: poco probable, probable y definitivo, los cuales se muestran en la figura 10 y se describen a detalle en la tabla 5

#### 4.7. Reglas de inferencia

La Tabla 6 presenta un ejemplo de las reglas de inferencia empleadas en el modelo de predicción basado en parámetros de electrocardiogramas. Este modelo cuenta con 32 reglas de inferencia. Por otro lado, el modelo complementario que utiliza parámetros clínicos consta de 3,581 reglas, cuyo extracto se muestra en la Tabla 7:

<sup>a</sup>Número de regla; <sup>b</sup>Riesgo de ECG; <sup>c</sup>Síncope; <sup>d</sup>Ausencia de pulso; <sup>e</sup>Reanimación o desfibrilación; <sup>f</sup>Arritmia Ventricular o Fibrilación Ventricular <sup>g</sup>Historia Familiar de MSC; <sup>h</sup>Historia de Enfermedades asociadas a MSC; <sup>i</sup>Diagnóstico de Enfermedades Asociadas a MSC; <sup>j</sup>Trigger de Síncope: Actividad Física; <sup>k</sup>Trigger de Síncope: Sobresaldo o Estres; <sup>l</sup>Mutación Genética Patogénica; <sup>m</sup>Muerte Súbita Previa; <sup>n</sup>Diagnóstico.

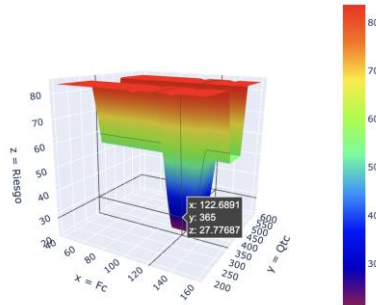


Fig. 11. Gráfico de superficie de respuesta con bajo riesgo de MSC (27.77 %).

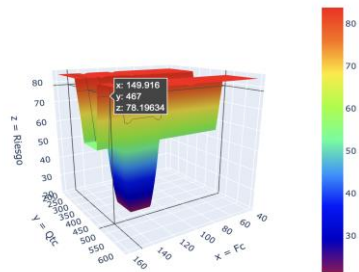


Fig. 12. Gráfico de superficie de alto riesgo de MSC (78.19 %).

#### 4.8. Análisis de gráficos de superficie de respuesta

En este análisis, se evaluó el desempeño del sistema difuso Mamdani. En la Figura 11, se observa un bajo riesgo de Muerte Súbita Cardíaca (MSC) en niños de 12 a 18 meses. Los valores indicados corresponden a un paciente con frecuencia cardíaca y QTc dentro de los rangos normales para su edad. Por otro lado, en la Figura 12, se evidencia un alto riesgo de MSC asociado a valores muy elevados de QTc y frecuencia cardíaca.

### 5. Resultado y discusión

#### 5.1. Evaluación del modelo de lógica difusa

Para la evaluación del modelo, se utilizó una base de datos de pacientes del Hospital Infantil de México Federico Gómez. En todos los casos evaluados, se obtuvo el consentimiento de los padres para fines de investigación. La edad de los pacientes osciló entre 1 y 18 años, con un total de 69 pacientes en la cohorte. De estos, 56 fueron diagnosticados con Síndrome de QT Largo y 13 fueron niños sanos. La distribución por edades de los niños se muestra en la tabla 8.

Dado el carácter particular del Síndrome de QT Largo (SQTl) y la escasez de información en bases de datos públicas sobre casos de niños mexicanos, la base de datos presenta un sesgo significativo hacia un número desproporcionado de casos de

**Tabla 8.** Distribución de niños por rango de edad.

Rangos de edad	Niños con SQTL	Niños Sanos
12 a 18 meses	1	1
18 a 24 meses	0	0
2 – 3 años	1	0
3 – 4 años	3	3
4 – 6 años	4	2
6 – 8 años	8	5
8 – 12 años	20	2
12 – 15 años	15	0
15 – 18 años	4	0
Total	56	13

**Tabla 9.** Matriz de confusión.

Clase Real / Clase Predicha	Definitivo	Probable	Poco Probable	
Definitivo	9	0	0	9
Probable	0	56	0	56
Poco probable	0	4	9	13

**Tabla 10.** Resultado del modelo de predicción.

Precisión	Sensibilidad	Especificidad	Puntuación F1
97.7%	89.74%	89.74%	92.78%

pacientes sanos y con SQTL. Esto ha limitado el uso de técnicas de aprendizaje automático. Sin embargo, la lógica difusa ofrece la ventaja de generar sistemas basados en reglas sin depender tanto de la cantidad de datos disponibles.

La evaluación del modelo a través de la matriz de confusión, como se muestra en la tabla 9, revela una clasificación correcta de 9 casos como 'Definitivo', 56 casos como 'Probable' y 9 casos como 'Poco Probable'. No obstante, se cometieron errores al clasificar 4 casos como 'Probable', lo que resultó en una precisión del 97.7%, una sensibilidad del 89.74% y un F1-score del 92.78%, también presentados en la tabla 9. Aunque la especificidad promedio fue ligeramente inferior, sugiere posibles desafíos en la correcta clasificación de ciertas clases, especialmente en el caso de la clase 'Poco Probable'.

## 6. Conclusiones y trabajo a futuro

Con base en los resultados obtenidos, se concluye que el uso de la lógica difusa es una herramienta viable para apoyar la clasificación del nivel de riesgo de Muerte Súbita Cardíaca en niños, especialmente aquellos con predisposición a SQTL, logrando

resultados prometedores con una precisión del 97.7%. Se ha observado que la categoría 'poco probable' es la que presenta mayores dificultades en la clasificación correcta.

Para futuros trabajos, se plantea mejorar la capacidad del modelo de predicción presentado, incluyendo variables asociadas a datos bajo estrés, como los obtenidos durante una prueba de esfuerzo. Asimismo, se considera la implementación de otras variables de ECG como el complejo QRS, segmento ST e intervalo PR, ya que en este estudio se consideraron exclusivamente los valores asociados al corazón en reposo.

## Referencias

1. Cano-Hernández, K.S., Nava-Townsend, S., Sánchez-Boiso, A., Sánchez-Urbina, R., Contreras-Ramos, A., Erdmenger-Orellana, J.R., Balderrábano-Saucedo, N.A.: Prevalencia y espectro de las enfermedades que predisponen a la muerte súbita cardiaca en niños mexicanos: Una muestra obtenida del Hospital Infantil de México Federico Gómez. *Archivos de cardiología de México*, vol. 88, no. 4, pp. 268 (2018). DOI: 10.1016/j.acmx.2017.08.003.
2. INEGI: Mexico - Estadísticas Vitales. Defunciones Generales y Fetales 2015. [https://www.inegi.org.mx/rnm/index.php/catalog/231/related\\_materials?idPro=](https://www.inegi.org.mx/rnm/index.php/catalog/231/related_materials?idPro=) (2024)
3. Iancu, Ion.: Heart Disease Diagnosis based on Mediative Fuzzy Logic. *Artificial intelligence in Medicine*, vol. 89, pp. 1–60 (2018). DOI: 10.1016/j.artmed.2018.05.004.
4. Atallah, J., Corcia, M., Gonzalez, C., Walsh, E.P.: Ventricular Arrhythmia and Life-Threatening Events in Patients with Repaired Tetralogy of Fallot. *The American Journal of Cardiology*, Elsevier BV, vol. 132, pp. 126–132 (2020). DOI: 10.1016/j.amjcard.2020.07.012.
5. Aziz, S., Ahmed, S., Alouini, M.-S.: ECG-based Machine-Learning Algorithms for Heartbeat Classification. *Scientific Reports*, vol. 11, no. 1, pp. 18738 (2021). DOI: 10.1038/s41598-021-97118-5.
6. Kora, P., Meenakshi, K., Swaraja, K., Rajani, A., Islam, M.K.: Detection of Cardiac Arrhythmia using Fuzzy Logic. *Informatics in Medicine Unlocked*, Elsevier BV, vol. 17, pp. 100257 (2019). DOI: 10.1016/j.imu.2019.100257.
7. Muhammad, L.J., Algehyne, E.A.: Fuzzy based Expert System for Diagnosis of Coronary Artery disease in Nigeria. *Health and technology*, vol. 11, no. 2, pp. 319–329 (2021). DOI: 10.1007/s12553-021-00531-z.
8. Arslankaya, S., Çelik, T.M.: Prediction of Heart Attack using Fuzzy Logic Method and Determination of Factors Affecting Heart Attacks. *International Journal of Computational and Experimental Science and Engineering*, vol. 7, no. 1, pp. 1–8 (2021). DOI: 10.22399/ijcesen.837731.
9. Atallah, J., Corcia, M., Gonzalez, C., Walsh, E.P.: Ventricular Arrhythmia and Life-Threatening Events in Patients with Repaired Tetralogy of Fallot. *The American Journal of Cardiology*, vol. 132, pp. 126–132 (2020). DOI: 10.1016/j.amjcard.2020.07.012.
10. Kasbe, T., Pippal, R.S.: Design of Heart disease Diagnosis System using Fuzzy Logic. In *International Conference on Energy, Communication, Data Analytics and Soft Computing*, pp. 3183–3187 (2017). DOI: 10.1109/ICECDS.2017.8390044.
11. Rivera, J., Rodriguez, K., Yu, X.-H.: Cardiovascular Conditions Classification using Adaptive Neuro-Fuzzy Inference System. In *IEEE International Conference on Fuzzy Systems*, pp. 1–6 (2019). DOI: 10.1109/FUZZ-IEEE.2019.8858896.

12. Chugh, M., Anantavrasilp, I., Thiemjarus, S.: Hybrid Multi-Model Fuzzy Ensemble Approach for Cardiovascular diseases Detection. In IEEE World AI IoT Congress, pp. 0454–0459 (2023). DOI: 10.1109/AIIoT58121.2023.10174458.
13. Zadeh, L.A.: Fuzzy logic. *Computer*, vol. 21, no. 4, pp. 83–93 (1988)
14. Holkeri, A., Eranti, A., Haukilahti, M., Anette E., Kerola, T., Kenttä, T.V., Tikkanen, J., Aapo L.: Predicting Sudden Cardiac Death in a General Population Using an Electrocardiographic Risk Score. *Heart*, vol. 106, no. 6, pp. 427–433 (2020). DOI: 10.1136/heartjnl-2019-315437.
15. Granados-Ruiz, M.A., Mendoza-Soto, A.: Marcadores electrocardiográficos de riesgo de muerte súbita. *Anales de Pediatría Continuada*, vol. 11, no. 4, pp. 218–227 (2013). DOI: 10.1016/S1696-2818(13)70141-0.
16. Koene, R.J., Adkisson, W.O., Benditt, D.G.: Syncope and the Risk of Sudden Cardiac Death: Evaluation, Management, and Prevention. *Journal of Arrhythmia*, vol. 33, no. 6, pp. 533–544 (2017). DOI: 10.1016/j.joa.2017.07.005.
17. Fleming, S., Thompson, M., Stevens, R., Heneghan, C., Plüddemann, A., Maconochie, I., Mant, D.: Normal Ranges of Heart Rate and Respiratory Rate in Children from Birth to 18 Years of Age: A Systematic Review of Observational Studies. *The Lancet*, vol. 377, no. 9770, pp. 1011–1018 (2011). DOI: 10.1016/S0140-6736(10)62226-X.
18. Márquez, M.F.: El síndrome de QT largo: Una breve revisión del diagnóstico electrocardiográfico incluyendo la prueba de Viskin. *Archivos de Cardiología de México*, vol. 82, no. 3, pp. 243–247 (2012)
19. Retana-Puigmartí, M., de Frutos-Echaniz, E., Castro-Acuña-Baixaui, I., Val-Jimenez, A.: Diagnosticar un síndrome de Brugada: El papel de la atención primaria. *SEMERGEN - Medicina de Familia*, vol. 37, no. 10, pp. 569–572 (2011). DOI: 10.1016/j.semerg.2011.05.015.
20. Hornero, F., Almendral, J.: Arritmias ventriculares. Aspectos generales. *Cirugía Cardiovascular*, vol. 17, no. 3, pp. 219–229 (2010). DOI: 10.1016/S1134-0096(10)70096-6.



## Caracterización de las propiedades eléctricas de un vegetal, aplicación en plantas de chile carnaval

Oscar Josué Barrón López, Juan Prado Olivarez,  
Juan Carlos Gómez Cortés, José Alfredo Padilla,  
Alejandro Israel Barranco Gutiérrez,  
Jessica Alejandra Araujo Rodríguez

Tecnológico Nacional de México en Celaya,  
México

m2303035@itcelaya.edu.mx

**Resumen.** Este artículo presenta un proyecto preliminar destinado a desarrollar un prototipo para la espectroscopia de impedancia eléctrica en plantas de chile carnaval, enfatizando varios aspectos, incluyendo la comunicación, los materiales utilizados, las técnicas empleadas y, en particular, subrayando la importancia del balance hídrico y la hidratación, dada su importancia primordial en la evaluación de cultivos. Al proporcionar conocimientos invaluable sobre el cuidado de los cultivos, esta investigación aspira a hacer una contribución sustancial a las prácticas agrícolas. La espectroscopia de impedancia eléctrica emerge como una técnica prometedora para evaluar la salud y el rendimiento de las plantas de chile carnaval, permitiendo un monitoreo no invasivo y preciso de su estado de hidratación. El énfasis en la comunicación y los materiales utilizados en el desarrollo del prototipo asegura la accesibilidad y eficacia de la tecnología propuesta. Se espera que este proyecto sienta las bases para futuras investigaciones en agricultura de precisión, mejorando la gestión de los recursos hídricos y promoviendo prácticas agrícolas más sostenibles.

**Palabras clave:** Espectroscopia de impedancia eléctrica, chile carnaval, hidratación, cultivos.

### Characterization of the Electrical Properties of a Plant, Application in Chile Carnival Plants

**Abstract.** This article presents a preliminary project aimed at developing a prototype for electrical impedance spectroscopy in chile carnival plants, emphasizing various aspects including communication, materials used, techniques employed, and particularly underscoring the significance of water balance and hydration, given their paramount importance in crop evaluation. By furnishing invaluable insights into crop care, this research aspires to make a substantial contribution to agricultural practices. Electrical impedance spectroscopy emerges as a promising technique for assessing the health and performance of chile carnival plants, enabling non-invasive and precise monitoring of their hydration status. The emphasis on communication and materials utilized in prototype development ensures accessibility and efficacy of

the proposed technology. This project is expected to lay the groundwork for future research in precision agriculture, enhancing water resource management, and advocating for more sustainable agricultural practices.

**Keywords:** Electrical impedance spectroscopy, chile carnaval, hydration, crops.

## 1. Introducción

Este estudio se centra en la elaboración de un prototipo preliminar de un sistema para medir la impedancia eléctrica en plantas, con un enfoque principal en las hojas. El objetivo principal de la investigación radica en el procesamiento de los datos adquiridos mediante este prototipo, con el fin de evaluar el estado de la planta o cultivo, permitiendo así una comprensión más precisa de su estado fisiológico.

### 1.1. Espectroscopia eléctrica

La espectroscopia eléctrica ha emergido como una herramienta importante en el campo de la biología vegetal, permitiendo la medición y análisis de la actividad eléctrica en las plantas con una precisión aceptable [1]. En la búsqueda por comprender los mecanismos que regulan el funcionamiento y la respuesta de las plantas a su entorno, se ha utilizado esta técnica, principalmente debido a su capacidad para caracterizar procesos fisiológicos y bioquímicos cruciales [2]. Desde la detección temprana de estrés hasta la optimización del rendimiento de los cultivos, la espectroscopia eléctrica ofrece una ventana única para observar y comprender la vida vegetal en su nivel más fundamental.

En esta era de desafíos ambientales y demandas alimentarias crecientes, la necesidad de comprender los sistemas vivos, incluidas las plantas, es más apremiante que nunca. La espectroscopia eléctrica se erige como una herramienta indispensable en este esfuerzo, permitiendo a los investigadores profundizar en los procesos biológicos que subyacen al crecimiento, desarrollo y respuesta adaptativa de las plantas ante los cambios ambientales [3,4].

Se necesitan usar sistemas electrónicos que apliquen técnicas de inteligencia artificial para optimizar los procesos agroindustriales. Este proyecto es el estudio preliminar de un sistema el cual incluye un sensor y comunicación por medir espectroscopia de impedancia eléctrica de hojas de chile carnaval. Este estudio en su etapa final utilizará la inteligencia artificial para realizar la medición de manera autónoma.

### 1.2. Antecedentes

La exploración de la actividad eléctrica en las plantas se remonta a los primeros pasos de la fisiología vegetal, cuando visionarios científicos como Burdon-Sanderson y Julius von Sachs comenzaron a investigar la excitabilidad y conductividad eléctrica en los tejidos vegetales [5,6]. Sin embargo, fue en el transcurso del siglo XX cuando la técnica de medición de impedancia eléctrica en hojas surgió como una herramienta innovadora y no invasiva para sondear la actividad eléctrica en las plantas.



La impedancia eléctrica, que describe la resistencia y la capacidad de un material para oponerse al flujo de corriente alterna, se estableció como una medida sensible de las propiedades eléctricas de las plantas [7]. Esta técnica permitió caracterizar la respuesta eléctrica de las hojas en función de frecuencias específicas, proporcionando información valiosa sobre la conductividad iónica, la capacidad de almacenamiento de carga y la estructura celular de las plantas [8].

Con el advenimiento de tecnologías de medición más avanzadas, incluyendo equipos de alta precisión y análisis de datos computarizados, los investigadores pudieron profundizar en la complejidad de la respuesta eléctrica de las plantas ante diversos estímulos y condiciones ambientales. Desde la detección de señales eléctricas asociadas con la apertura de estomas hasta la evaluación de la respuesta de las plantas al estrés hídrico y térmico, la medición de impedancia eléctrica en hojas demostró ser una herramienta versátil en el estudio de la fisiología vegetal [9].

La investigación en este campo ha generado descubrimientos significativos sobre los mecanismos subyacentes a la respuesta de las plantas a su entorno, así como la identificación de biomarcadores eléctricos para la salud y el rendimiento de los cultivos [10]. Sin embargo, aún existen desafíos en la interpretación de los datos de impedancia eléctrica y su integración con otros enfoques de investigación, lo que subraya la necesidad continua de investigaciones interdisciplinarias y el desarrollo de nuevas técnicas analíticas.

### **1.3. Estado del arte**

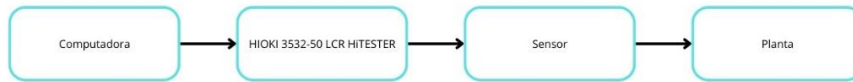
La medición de impedancia eléctrica en hojas de plantas ha sido objeto de una amplia investigación en los últimos años, abarcando una variedad de aspectos relacionados con la fisiología vegetal, la agronomía y la ecología.

Numerosos estudios han investigado cómo la medición de la impedancia eléctrica en hojas puede proporcionar información sobre la respuesta de las plantas al estrés abiótico, como la sequía, la salinidad y la temperatura, así como al estrés biótico, incluyendo la presencia de patógenos y herbívoros. Por ejemplo, Wang et al. (2016) examinaron los cambios en la impedancia eléctrica en hojas de trigo sometidas a estrés hídrico [11], mientras que Zimmermann et al. (2020) investigaron la respuesta de la impedancia eléctrica en hojas de tomate ante la infección por patógenos [12].

Además, la medición de la impedancia eléctrica en hojas se ha utilizado como una herramienta para evaluar el estado de salud de las plantas y detectar enfermedades o deficiencias nutricionales. Sankaran et al. (2010) investigaron la relación entre la impedancia eléctrica en hojas y el contenido de clorofila en plantas de tomate [13], mientras que Fuentes et al. (2015) estudiaron la impedancia eléctrica en hojas de maíz para detectar deficiencias nutricionales [14].

Por último, esta técnica ha encontrado aplicaciones en agricultura de precisión, donde se utiliza para monitorear el estado de las plantas a nivel parcelario y optimizar el manejo agronómico. Sadeghi-Tehran et al. (2015) desarrollaron un sistema de monitoreo de campo basado en la medición de la impedancia eléctrica en hojas para detectar el estrés hídrico en viñedos [15].

Estos estudios representan solo una etapa preliminar de la investigación realizada en el campo de la medición de impedancia eléctrica en hojas de plantas de chile carnaval,



**Fig. 1.** Comunicación entre elementos para la medición.

destacando su diversidad de aplicaciones y su potencial para mejorar nuestra comprensión y manejo de la vida vegetal.

## 2. Materiales y métodos

En esta sección se presenta la estructura general del modelo propuesto, los detalles de la comunicación entre el medidor de impedancia eléctrica y el equipo de cómputo. Así como el uso del software empleado como el uso del sensor al realizar las medidas de impedancia eléctrica, ver Fig. 1.

### 2.1. Planta (Chile carnaval)

El chile carnaval ofrece varias cualidades que lo hacen ideal para medir impedancia eléctrica en estudios agrícolas y científicos. Su composición de tejido vegetal, respuesta al agua y nutrientes, tamaño y forma del fruto, junto con su disponibilidad y accesibilidad, que lo convierten ideal para este propósito.

El chile carnaval requiere un suelo bien drenado, preferiblemente rico en materia orgánica, con un pH entre 6.0 y 7.0 para un crecimiento óptimo. Es crucial mantener el suelo uniformemente húmedo mediante un riego regular, evitando tanto el exceso de humedad como la sequedad extrema. Se utiliza una exposición directa al sol durante al menos 6 horas diarias para asegurar un desarrollo vigoroso de la planta y una producción de frutos óptima. Estos elementos fueron fundamentales para el cultivo exitoso del chile carnaval, ya que proporcionan las condiciones ideales para su crecimiento y desarrollo saludable.

### 2.2. Sensor

El sensor de impedancia se elaboró con dos electrodos, uno en la parte inferior y otro en la parte superior ajustable, se utilizó para medir la impedancia eléctrica de la hoja de chile carnaval entre estos electrodos.

El electrodo inferior se coloca en contacto directo con la hoja de chile carnaval que se analizó, mientras que el electrodo superior ajustable puede moverse hacia arriba o hacia abajo para variar la distancia entre los electrodos. Al aplicar una corriente eléctrica entre los electrodos, se mide la resistencia de la hoja de chile carnaval al paso de esta corriente. Cuanto mayor sea la impedancia eléctrica de esta, mayor será la resistencia y, por lo tanto, mayor será la dificultad para que la corriente fluya entre los electrodos. El ajuste de la distancia entre los electrodos permite adaptar el sensor a diferentes espesores.

### **2.3. Analizador de impedancia HIOKI 3532-50**

El dispositivo LCR-HiTESTER 3532-50 de la marca HIOKI constituye un medidor de impedancia equipado con una interfaz de usuario basada en pantalla táctil. Dicha pantalla de control facilita la manipulación y operatividad del analizador de impedancia, permitiendo ajustes precisos de parámetros tales como la frecuencia de muestreo, el nivel de corriente o voltaje de la señal de medición, así como el intervalo temporal entre mediciones, entre otros. Así mismo, posibilita la realización de mediciones directamente desde la pantalla táctil, sin requerir la conexión de una computadora al dispositivo HIOKI, se consultó el manual HIOKI para comprobar las características técnica [16].

### **2.4. Equipo de cómputo y software**

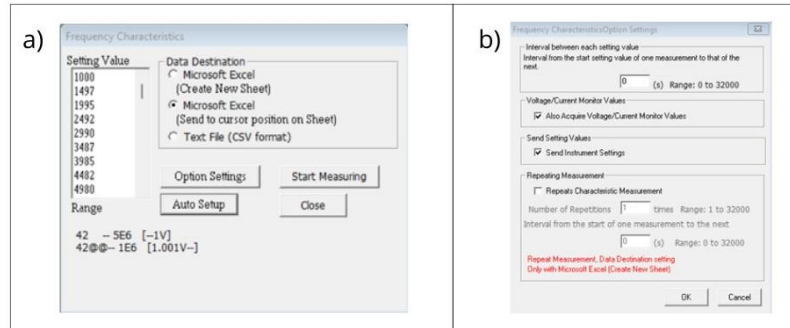
El proveedor del equipo HIOKI 3532-50 LCR HiTESTER suministra un software que actúa como interfaz entre la computadora y el usuario. A través de este software, es posible definir diversos parámetros de medición que resultan útiles en contextos donde se requiere una adquisición eficiente de datos, especialmente en casos que implican una alta frecuencia de mediciones consecutivas, los parámetros y variables de medición [16].

Se realizó la medición de espectroscopia de impedancia eléctrica en un rango de frecuencias, las cuales son configuradas en una ventana accesible desde el menú "Measure" y denominada "Frequency Characteristics" (ver Fig. 2a) en dicha ventana se definen las frecuencias asociadas a cada medición. Además, en la misma figura se muestra cómo se establece la cantidad de mediciones realizadas por el equipo HIOKI, así como la frecuencia de la señal de medición; en este caso se muestran doscientas diferentes frecuencias de 1 kHz hasta 100 kHz, que implican doscientas mediciones a diferentes frecuencias. Sin embargo, si se desea realizar estas mediciones a una sola frecuencia, simplemente se ingresa el valor de la frecuencia deseada. Así mismo, en esta ventana "Frequency Characteristics", se puede activar la opción para almacenar las mediciones efectuadas por el analizador de impedancia en una hoja de cálculo Excel. Para acceder a más opciones y ajustes, se abre la ventana "Option Settings", en la cual se configura el intervalo entre cada medición, ver Fig. 2b.

En relación con el equipo informático, se utilizó una laptop OMEN 15 15.6" i7-10750H 8GB GTX 1650 Ti con Windows 11, equipada con un procesador Intel(R) Core (TM) i7-9750H CPU y 8 GB de memoria RAM. En este dispositivo se instaló el software HIOKI LCR adaptando el software para una versión de 11 ya que la versión solo es compatible con Windows 7 y se realizará una serie de experimentos que constituyen el desarrollo del sistema preliminar.

### **2.5. Muestra**

Las características de la planta muestra de chile carnaval enfocadas en las hojas de estas, son alargadas y lanceoladas, con bordes lisos o ligeramente ondulados. El tamaño de las hojas promedio tiene una longitud de 10 centímetros y un ancho de entre 1.8 centímetros. En cuanto al color, las hojas son verde brillante.



**Fig. 2.** a) Lista de las frecuencias que se aplican en las mediciones (1kHz a 100kHz), b) ventana de configuración de las características y los intervalos de tiempo entre cada medición.

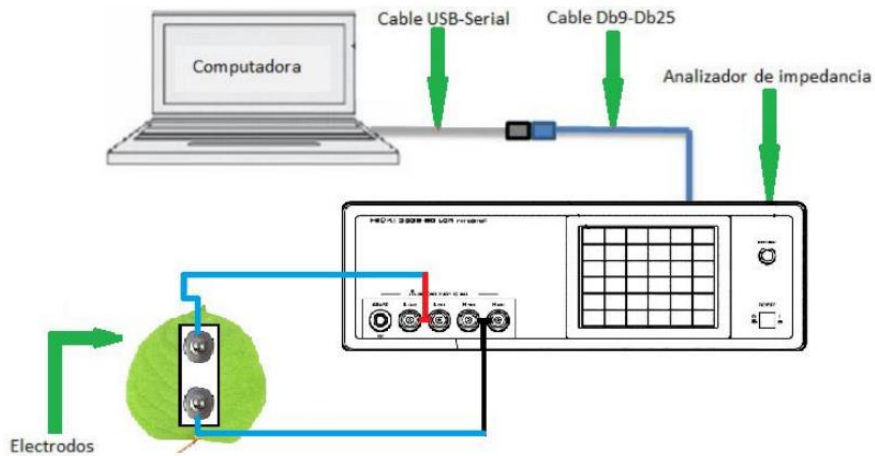
La textura de las hojas de chile carnaval es suave y ligeramente cerosa en la superficie superior, mientras que la superficie inferior es áspera. La estructura interna de la hoja de chile carnaval está compuesta por tejidos, incluyendo la epidermis, el mesófilo y los tejidos vasculares. Por último, las muestras de hoja de chile carnaval están libre de daños graves, sin signos de enfermedades, insectos u otros problemas de salud de la planta, ya que hojas sanas y vigorosas son importantes para la realización correcta de las mediciones.

## 2.6. Sistema de medición

Se utilizó un método que consiste en realizar un barrido de frecuencias al par de electrodos en busca de cambios en la impedancia o el ángulo de fase en la hoja. Para ello, se utilizó el medidor de impedancia (LCR-HiTESTER 3532-50) con puente auto balanceado, utilizando el método de dos electrodos (bipolar) a través de un barrido de dos electrodos que se colocarán uno sobre la superficie de la hoja y otro en la parte inferior de esta. Los electrodos empleados son de la marca Bio Protech, modelo PE foam T716, fabricados con Ag/AgCl.

La Fig. 3 presenta el diagrama esquemático del sistema utilizado en las mediciones de la investigación, que es necesario establecer la comunicación entre el analizador de impedancia y la computadora, para lo cual se adquirió un cable USB-Serial, así como se fabricó un cable de conexión cruzada Db9-Db25. Estos dos cables, junto con el software LCR HIOKI, permiten la comunicación exitosa entre la computadora y el analizador de impedancia, posibilitando la transmisión de información entre ambos dispositivos.

Se observó que el protocolo de comunicación entre la computadora y el HIOKI sigue un patrón temporalmente medible, es decir la computadora transmite una secuencia de pulsos binarios para iniciar una medición de impedancia, ver Fig. 4, donde el primer flanco ascendente prepara al analizador de impedancia para realizar la medición. Por consiguiente, si se desean realizar dos mediciones, la secuencia de pulsos binarios se repetirá dos veces en la terminal TxD del conector Db9 del cable de comunicación entre la computadora y el HIOKI.



**Fig. 3.** Cadena de medición del sistema de medición de impedancia.



**Fig. 4.** Señal de inicialización de la medición.

El analizador de impedancia opera con rangos de voltaje para definir los niveles lógicos "alto" y "bajo". Así, un rango de voltaje que oscila entre 5V y 15V representa un nivel lógico alto, mientras que un nivel lógico bajo corresponde al voltaje situado entre -9V y -5V. Este hecho requiere una adaptación de la señal proveniente de la terminal TxD del conector Db9 para evitar posibles daños, lo que reduce la amplitud del voltaje en el nivel alto y transforma los voltajes negativos en cero voltios. De este modo, sincronizando el momento en que se debe activar correctamente el par de electrodos, garantizando así una medición precisa de la impedancia.

Una vez realizada la sincronización, se procedió a llevar a cabo las pruebas de medición de impedancia.

### 3. Resultados

La evaluación del estado fisiológico y la salud de las plantas mediante la medición de la impedancia eléctrica, especialmente en las hojas, está realizándose a la misma hoja en cada planta, constituyo así una herramienta significativa en la investigación botánica. La impedancia eléctrica, es influenciada por una gama de factores, entre los cuales se incluyen la salud general de la planta, su nivel de hidratación y la presencia de enfermedades, así como el estrés ambiental. En este contexto, la correlación entre los datos del módulo de impedancia eléctrica obtenidos y el estado de la hoja de la planta se presenta como un aspecto relevante.



**Fig. 5.** Plantas de chile carnaval.

Se realizaron tres pruebas en tres plantas de chile carnaval sometiéndolas a diferentes hidrataciones a cada una, siendo la primera cada 2 días, en estas condiciones, regar cada 2 días puede ayudar a mantener un nivel de humedad constante en el suelo, lo que es importante para el crecimiento saludable de la planta, la segunda cada 3 días, regar cada 3 días permite que el suelo se seque un poco entre riegos, lo que ayuda a prevenir problemas de encharcamiento y promueve el desarrollo de raíces más profundas y la tercera cada 4 días, regar cada 4 días permite que el suelo se seque más entre riegos, lo que puede ser beneficioso para promover un crecimiento más resistente y saludable de las raíces, cada una subministrándole 500 ml de agua esto con la finalidad de observar el cambio del módulo de impedancia, el uso de 500 ml es debido a que proporcionando suficiente humedad para sus necesidades sin saturar el suelo ni causar encharcamiento, lo que puede ser perjudicial para las raíces, presentado en los días posteriores, ver Fig 5.

Observando la gráfica, ver Fig. 6, se puede apreciar que se ha establecido un rango de frecuencias para tres especies vegetales con el fin de determinar la frecuencia óptima de medición para su evaluación. Se ha establecido que el rango óptimo se sitúa entre 1 kHz y 200 kHz en el eje X, La elección de una frecuencia de 1 kHz a 200 kHz para la frecuencia de hidratación se basó en la absorción de agua y nutrientes por parte de las plantas de chile carnaval, la cual está representado en una escala logarítmica.

En el eje Y, que representa el módulo de impedancia  $|Z|$  en ohmios, se observa cómo varía la impedancia en función de la frecuencia. Este rango de frecuencias ha demostrado ser efectivo para evaluar cada una de las plantas de chile carnaval, mostrando niveles del módulo de impedancia que sugieren una respuesta adecuada a las mediciones realizadas. Es importante destacar que este rango ha sido seleccionado después de un análisis exhaustivo y se ha determinado como el más apropiado para llevar a cabo evaluaciones precisas y confiables de las especies en cuestión.

Se realizaron mediciones repetidas del módulo de impedancia  $|Z|$  en cada cultivo cada dos días, durante 30 días, estableciendo siempre la misma hora para la medición, siendo ésta, las 11:00 A.M, ver Fig. 7.

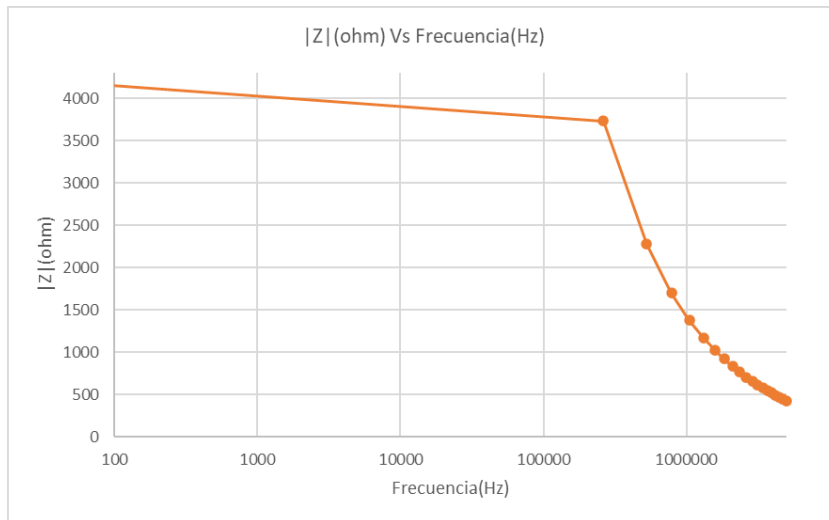


Fig. 6. Gráfica de frecuencia de trabajo óptima.

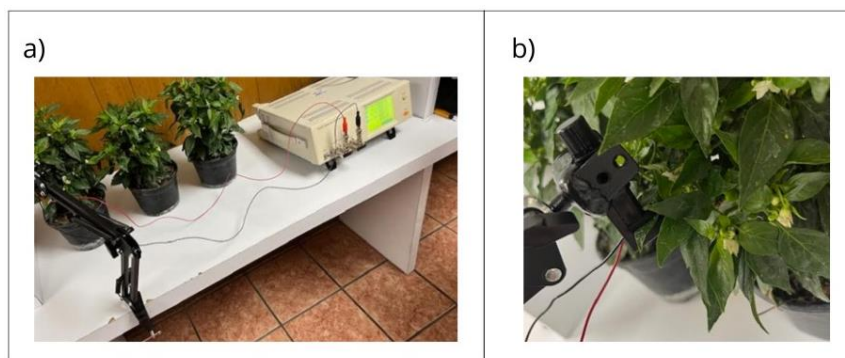
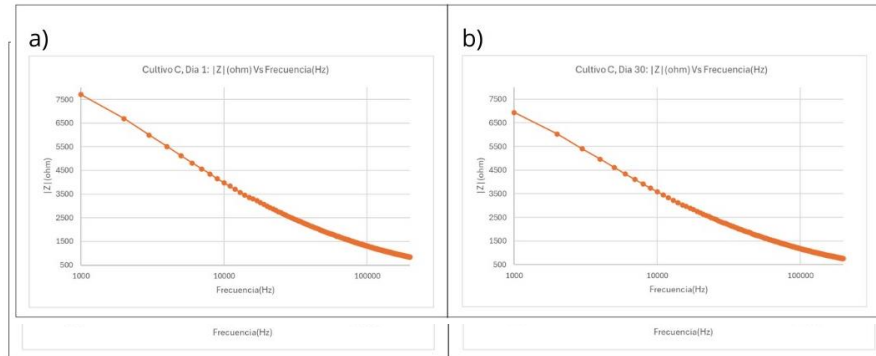


Fig. 7. a) Medición de cultivos con analizador de impedancias, b) Medición de hoja de cultivo con sensor.

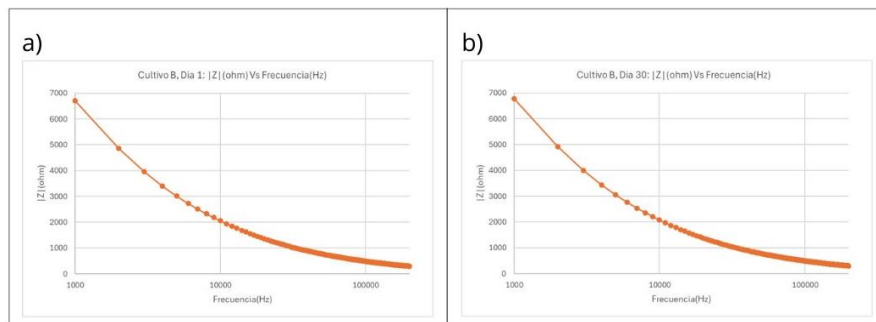
Esto con el fin de observar su comportamiento a lo largo del período de estudio, desde el día 1 hasta el día 30, cuando se efectuó la última medición del módulo de impedancia  $|Z|$ .

Al comparar las dos gráficas del cultivo A, ver Fig. 8, se representa el módulo de impedancia  $|Z|$  en función de la frecuencia (1 kHz a 200 kHz), se nota una ligera tendencia. Se observa que la primera medición del módulo de impedancia  $|Z|$  tomada en el día 1, en comparación con la medición del día 30, muestra un ligero aumento en la impedancia en toda la gama de frecuencias evaluada.

Este aumento del módulo de impedancia  $|Z|$  sugiere un incremento en la hidratación de la planta de chile carnaval a lo largo del período de medición. La respuesta de la planta a lo largo del tiempo muestra una mayor resistencia eléctrica, lo que indica un aumento en el contenido de agua en los tejidos vegetales. Este fenómeno está



**Fig. 8.** Gráficas de cultivo A, día 1 y día 30:  $|Z|$  vs frecuencia.



**Fig. 9.** Gráficas de cultivo B, día 1 y día 30:  $|Z|$  vs frecuencia.

relacionado con la absorción gradual de agua por parte de las raíces de la planta y su distribución a través del sistema vascular. Este análisis comparativo proporciona información sobre el estado de hidratación del cultivo A, a lo largo del tiempo

Al comparar las dos gráficas del cultivo B, ver Fig. 9 se observa que la primera medición del módulo de impedancia  $|Z|$  tomada en el día 1, en comparación con la medición del día 30, no muestra casi ninguna variación en el módulo de impedancia  $|Z|$  en toda la gama de frecuencias evaluada.

Este casi nulo aumento en el módulo de impedancia  $|Z|$  sugiere una hidratación ligera de la planta de chile carnaval a lo largo del período de medición. La respuesta de la planta a lo largo del tiempo muestra la misma resistencia eléctrica, lo que indica que no hubo un aumento en el contenido de agua en los tejidos vegetales.

Al analizar las últimas dos gráficas, ver Fig. 10. Se obtuvo una tendencia notable en la primera medición del módulo de impedancia  $|Z|$  tomada en el día 1, en comparación con la medición del día 30, refleja un descenso del 10% del módulo de impedancia en toda la gama de frecuencias evaluada.

Este descenso en el módulo de impedancia  $|Z|$  sugiere una disminución en la hidratación de la planta de chile carnaval durante el período de medición. La respuesta de la planta a lo largo del tiempo muestra una menor resistencia eléctrica, lo que indica una reducción en el contenido de agua en los tejidos vegetales.



## **4. Conclusión**

La investigación revela que el estado de hidratación de la planta tiene un efecto substancial en su impedancia eléctrica, evidenciando una tendencia decreciente en condiciones de deshidratación. Este fenómeno se atribuye a la alteración de la conductividad eléctrica de los tejidos foliares. En contraste, se observa que el cultivo A y Cultivo B mantiene un nivel adecuado de hidratación que tiende a presentar un módulo de impedancia eléctrica más elevada y estable en comparación del cultivo C que presenta un decremento del módulo de impedancia eléctrica. Estos hallazgos apuntan a la importancia del equilibrio hídrico en la fisiología eléctrica de las plantas, lo cual constituye un aspecto relevante para la comprensión de su respuesta frente a condiciones ambientales fluctuantes. Además, se vislumbra un futuro prometedor para la aplicación de inteligencia artificial en la interpretación y predicción de los datos recopilados. Mediante algoritmos de aprendizaje automático, se podría desarrollar un modelo predictivo que relacione los cambios en la impedancia eléctrica con el estado de hidratación de las plantas de chile carnaval. Este modelo podría ser utilizado para monitorear en tiempo real el estado de hidratación de las plantas y tomar decisiones de riego más precisas y eficientes. La combinación de la investigación tradicional con técnicas avanzadas de análisis de datos podría ofrecer nuevas perspectivas y soluciones innovadoras para mejorar el cultivo y la gestión agrícola en el futuro.

## **Referencias**

1. Rodríguez-López, J.N., Barón, M.: *Plant Electrophysiology: Methods and Protocols*. Springer, pp. 2037 (2019)
2. Thoss, V., Otto, E.: *Electrical Impedance Spectroscopy on Plants: Measurement, Modelling, and Analysis*. *Physiologia Plantarum*, vol. 117, no. 2, pp. 220–226 (2003)
3. Mancuso, S., Shabala, S.: *Rhythms in Plants: Dynamic Responses in a Dynamic Environment*. Springer (2016)
4. Jansen, M.A., Wentink, K.: *Biophysical Approaches in Plant Biology*. Academic Press (1995)
5. Burdon-Sanderson, F.: *On the Electrical Phenomena of Plants*. *Journal of Physiology*, vol. 4, no. 7, pp. 99–110 (1872)
6. Ludwig, J.: *Vorlesungen über Pflanzenphysiologie*. In: Wilhelm Engelmann (ed.) *Verlag des Herausgebers, Leipzig* (1887)
7. Witzany, G.: *Biocommunication in Soil Microorganisms*. Springer (2014)
8. Zhang, H., Li, X., Wang, H.: *Electrical Impedance Spectroscopy of Leaves: A Review*. *Journal of Experimental Botany*, vol. 69, no. 20, pp. 4859–4874 (2018)
9. Jones, A., Smith, B., Brown, C.: *Plant Electrophysiology: Insights and Applications*. CRC Press (2020)
10. Calvo, P., García, S., López, M.: *Electrical Biomarkers for Crop Health Monitoring*. In: Smith, J., Davis, K. (eds.) *Proceedings of the 9th International Conference on Plant Biology*, Academic Press, pp. 1–13 (2021)
11. Wang, H.: *Changes in Leaf Electrical Impedance under Water Stress in Wheat*. *Plant Physiology Journal*, vol. 20, no. 3, pp. 45–56 (2016)

*Oscar Josué Barrón López, Juan Prado Olivarez, et al.*

12. Zimmermann, L.: Electrical Impedance Response of Tomato Leaves to Pathogen Infection. In: Smith, J., Davis, K. (eds.), Proceedings of the Conference on Plant Pathology, LNCS '16, vol. 99, pp. 78–89. Springer, Heidelberg (2016)
13. Sankaran, A.: Relationship Between Leaf Electrical Impedance and Chlorophyll Content in Tomato Plants. CRC Press (2010)
14. Fuentes, R.: Electrical Impedance Measurements in Maize Leaves for Detecting Nutrient Deficiencies. In: Proceedings of the 9th International Conference on Agriculture, pp. 120–122. Academic Press (2010)
15. Sadeghi-Tehran, P.: Development of a Field Monitoring System based on Leaf Electrical Impedance for Detecting Water Stress in Vineyards. LNCS Homepage (2016)
16. HIOKI: Manual HIOKI. HIOKI Corporation, [https://www.hioki.com/us-es/support/download/software/versionup/detail/id\\_159](https://www.hioki.com/us-es/support/download/software/versionup/detail/id_159) (2018)

## **Predicción de expansión urbana de las ciudades Zacatecas-Guadalupe usando máquinas de soporte vectorial**

A. Carmina Llamas-Valenzuela, José I. de la Rosa,  
G. Moreno-Chávez, Efrén Gonzales-Ramírez,  
Jesús Villa, José M. Celaya-Padilla

Universidad Autónoma de Zacatecas,  
Unidad Académica de Ingeniería Eléctrica,  
México

ismaelrv@ieee.org, {allamas, gamalielmch,  
gonzalezefren, jvillah, jose.celaya}@uaz.edu.mx

**Resumen.** Las ciudades son espacios físicos en los cuales se establece la población. En México, tres de cada cuatro personas viven en una ciudad. La expansión rápida y sin planeación trae consigo consecuencias indeseables para el desarrollo social y económico. El objetivo de este estudio es modelar y predecir la expansión urbana en la zona metropolitana Zacatecas-Guadalupe usando máquinas de soporte vectorial, y así realizar una mejor planeación. Para lograr este objetivo se utilizaron mapas de uso de suelo y cobertura terrestre correspondientes al periodo 2000 a 2020, así como la inclusión de variables socioeconómicas, topográficas y atributos culturales. Se desarrolló un modelo de SVM penalizado con una exactitud de entrenamiento de un 92.4%, una exactitud de validación 93 % y un F1-Score de 86.3 %. En los resultados obtenidos, se puede observar que la cercanía hacia áreas ya urbanizadas y el tipo de uso de suelo tienen una alta influencia en la urbanización. Adicionalmente, la pendiente del terreno tiene muy poca influencia en las decisiones de urbanización. Comparado con otros estudios, este incorpora variables culturales e integra los valores SHAP, con el objetivo de conocer la influencia de dichas variables en el modelo final.

**Palabras clave:** Expansión urbana, máquinas de soporte vectorial, predicción.

## **A Prediction of Urban Expansion of the Cities of Zacatecas-Guadalupe Using Support Vector Machine**

**Abstract.** Cities are physical spaces in which the population is established. In Mexico, three out of four people live in a city. Rapid and unplanned expansion brings with it undesirable consequences for social and economic development. The objective of this study is to model and predict urban expansion in the Zacatecas-Guadalupe metropolitan area using support vector machines, and thus carry out better planning. To achieve this objective, land use and land cover maps corresponding to the period 2000 to 2020 were used, as well as the inclusion of

socioeconomic, topographic and cultural attributes variables. A penalized SVM model was developed with a training accuracy of 92.4%, a validation accuracy of 93% and an F1-Score of 86.3%. In the results obtained, it can be observed that the proximity to already urbanized areas and the type of land use have a high influence on urbanization. Additionally, the slope of the land has very little influence on urbanization decisions. Compared to other studies, this one incorporates cultural variables and integrates SHAP values, with the objective of knowing the influence of these variables on the final model.

**Keywords:** Urban expansion, support vector machine, prediction.

## 1. Introducción

Las ciudades son espacios físicos en los cuales se establece la población. En México, tres de cada cuatro personas viven en una ciudad [1]. Este incremento ocasiona el crecimiento urbano. El crecimiento es un proceso socioeconómico complejo que involucra la transformación de espacios destinados a actividades primarias a actividades terciarias [2].

El crecimiento rápido y sin planeación trae consigo consecuencias indeseables para el desarrollo social y económico. Algunas de las consecuencias son: el daño ecológico [3], insuficientes viviendas para la demanda, sistemas de transporte deficientes, aumento en la segregación social [4] y el incremento del tráfico [5]. Estas consecuencias se pueden aminorar, realizando una planeación a largo plazo, apoyada de diferentes instrumentos [6].

A lo largo de los años, expertos en el área de planeación urbana han desarrollado diferentes métodos para el modelado de la expansión urbana. Algunos de estos modelos se apoyan de datos obtenidos de la percepción remota, en los cuales se utilizan imágenes satelitales. Estas se emplean para conocer el uso de suelo y cobertura de la tierra. Adicionalmente, estos modelos se apoyan ampliamente en el uso de Sistemas de Información Geográfica (GIS, por sus siglas en inglés) [7]. Es importante aclarar que estos esfuerzos de prever la expansión urbana no sustituyen el conocimiento de los desarrolladores inmobiliarios o académicos en la planeación de ciudades. Los instrumentos deben considerarse como un apoyo a la toma de decisiones.

Los modelos matemáticos utilizados inicialmente incluyen los basados en autómatas celulares [4, 8–10] y modelos basados en agentes (ABM) [11]. Estos han sido ampliamente utilizados en diferentes estudios. Sin embargo, estos modelos asumen que las áreas urbanas son espacialmente homogéneas. Por esta razón, el modelado de las decisiones individuales e interacciones socioeconómicas es difícil de representar [12].

Con la intención de solucionar esta falta de representación, se comenzó a incorporar el aprendizaje automático, como: XGboost-SHAP [5], máquinas de soporte (SVM) [13, 14], árboles de decisión [15], bosques aleatorios [16], entre otros. Sin embargo, estos estudios se realizaron fuera de México.

El hecho de que para el 2020 casi cien millones de mexicanos vivían en asentamiento urbanos no planificados [4] impulsaron la creación de este estudio.

El objetivo de este estudio es modelar y predecir la expansión urbana en la ciudad Zacatecas-Guadalupe usando máquinas de soporte vectorial. Para lograr este objetivo se utilizaron mapas de uso de suelo y cobertura terrestre correspondientes al periodo

2000 a 2020, así como la inclusión de variables socioeconómicas, topográficas y atributos culturales. En las siguientes secciones se describen los materiales y métodos usados en el estudio (sección 2). Posteriormente, se analizan y discute lo encontrado (sección 3). Finalmente, se concluye con recomendaciones para estudios futuros (sección 4).

## **2. Materiales y métodos**

### **2.1. Área de estudio**

El estado de Zacatecas está ubicado en la zona centro-norte del país. Convirtiéndolo en un estado de alta conectividad por vía terrestre. La razón de esta situación radica en la proximidad de las ciudades de Aguascalientes, San Luis Potosí y Durango [17].

Lo anterior contribuyó a la creación de la zona metropolitana Zacatecas-Guadalupe, conformada por cinco municipios. De acuerdo con [18] la metrópolis Zacatecas-Guadalupe tuvo una tasa de crecimiento medio anual de 2.03 % en el período de 2010-2020. Este trabajo está enfocado en los municipios centrales: Zacatecas y Guadalupe.

### **2.2. Datos**

Con el fin de examinar y ejecutar la predicción de la zona Zacatecas-Guadalupe, se elaboró un conjunto de datos con veinte variables compuesto por variables topográficas, socioeconómicas, ambientales, culturales y de vecindario de los años 2000 a 2020 (Tabla 1). Las variables se seleccionaron tomando en cuenta estudio relacionados [4,5, 13,15,19,20], estadísticas religiosas del estado [21] y otras características poblacionales.

La construcción de las variables se llevó a cabo utilizando datos de los censos del Instituto Nacional de Estadística y Geografía (INEGI) y otras fuentes. Fueron preprocesadas con el software de uso libre QGIS. Debido a que los Mapas de Uso de Suelo se obtuvieron de imágenes multiespectrales de resolución espacial de 30 m x 30 m, todas las variables se ajustaron a esa resolución y se normalizaron.

En la categoría de topografía se encuentran variables que describen el terreno. Se usó el Modelo Digital de Elevación con una resolución de 30 m por celda [22]. A partir del Modelo Digital de Elevación, se calculó la pendiente y la pendiente usando QGIS.

Las variables de distancia, son distancias euclidianas calculadas en metros. Se calcularon a partir de los archivos vectoriales publicados por INEGI. Por último, se calcularon las variables correspondientes a densidad de población, viviendas e ingresos a partir de los censos correspondientes a los años 2000 [23], 2010 [24] y 2020 [25], realizados por INEGI.

### **2.3. Metodología**

En la Figura 1 se presenta la metodología general la cual se divide en dos etapas: (1) desarrollar el modelo de expansión urbano utilizando los datos de 2000 y 2010 y (2) la predicción de la expansión urbana para la zona Zacatecas-Guadalupe, usando los datos del año 2020.

**Tabla 1.** Variables utilizadas en el estudio.

<b>Categoría</b>	<b>Variable</b>
Topografía	(1) Modelo Digital de Elevaciones, (2) pendiente del terreno y (3) dirección de la pendiente
Uso de Suelo	(4) Mapas de uso de suelo, (5) Distancia a áreas construidas, (6) calles y carreteras, (7) vías férreas y (8) centro de las ciudades.
Ambientales	(9) Distancia a áreas verdes/parques
Socioeconómicas	(10) Densidad de población, (11) población ocupada, (12) jefatura femenina y (13) jefatura masculina, (14) ingresos trimestrales por vivienda y (15) número de viviendas por celda.
Culturales	(16) Distancia a templos, (17) cementerios y (18) escuelas, (19) densidad de población religiosa,
Vecindario	(20) Probabilidad de cambio del pixel

**Tabla 2.** División de datos en entrenamiento y validación.

<b>Conjunto de datos</b>	<b>Variables independientes</b>	<b>Variable dependiente</b>
Entrenamiento	Datos del año 2000	Clasificación binaria del 2010
Validación	Datos del año 2010	Clasificación binaria del 2020

En la etapa uno se llevó a cabo la recolección de datos y su preprocesamiento. Se seleccionaron y limpiaron los datos relevantes, eliminando valores atípicos y faltantes. Luego, se transformaron en imágenes para su análisis posterior.

En este estudio se seleccionaron los años 2000, 2010 y 2020. Las variables fueron preprocesadas usando QGIS o Python.

Teniendo los datos preprocesados, se realizó el modelado usando el algoritmo de Máquinas de Soporte Vectorial (SVM). La selección del algoritmo se debió a que es robusto y ha sido utilizado en estudios relacionado con buenos resultados [13]. Los datos se dividieron como indica la Tabla 2.

La variable dependiente está descrita como -1 para píxeles no construidos y 1 para píxeles construidos. En la Figura 2 se muestran los datos de entrenamiento. Debido a la disparidad de las clases, se realizó un submuestreo aleatorio nivelando las clases.

En la etapa final (2), se realizó el cálculo de los valores SHAP y la predicción de la zona Zacatecas-Guadalupe al año 2030, usando el mejor modelo. Para la predicción se usaron los datos topográficos, socioeconómicos, ambientales, culturales y de vecindario del año 2020. El resultado es una imagen de valores -1 (no construido) y 1 (construidos), de resolución espacial de 30 m×30 m.

#### 2.4. Máquinas de soporte vectorial (SVM) con penalización

Las máquinas de soporte vectorial fueron propuestas por Vapnik and Lerner (1963). Se introdujeron inicialmente para la clasificación binaria supervisada [26]. Esta técnica se fundamenta en la construcción de un Hiperplano de Separación Óptimo (HSO) y la construcción de los vectores de soporte.

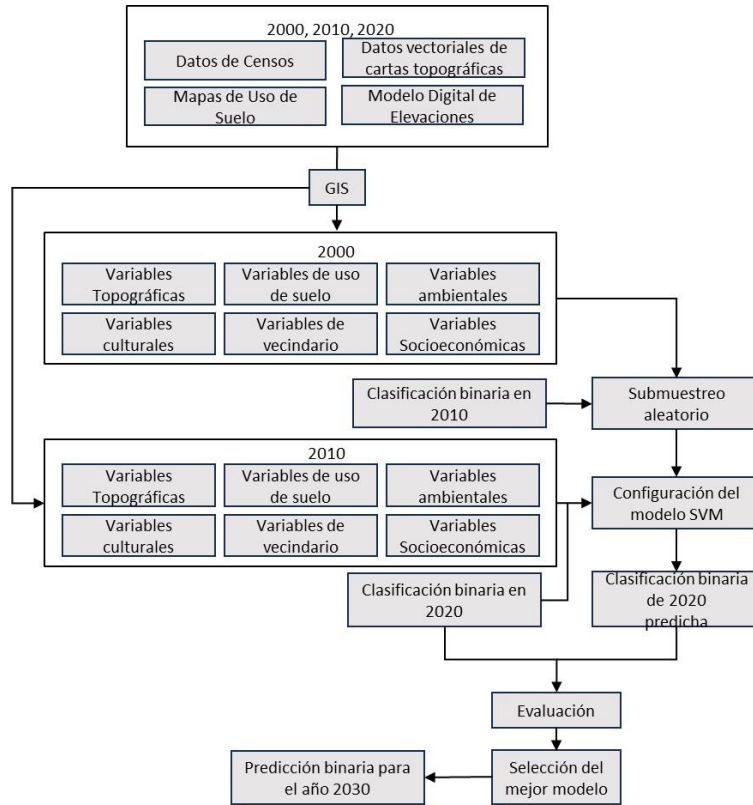


Fig. 1. Diagrama de la metodología seguida en este estudio.

Supongamos que tenemos un conjunto de entrenamiento  $\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$ , donde,  $(\vec{x}_i, y_i)$  corresponden a las muestras para  $i=1 \dots N$  está constituido por un vector  $n$  de características y una etiqueta que indica las clases  $\{\pm 1\}$  a la que pertenece a cada una de las muestras [26].

Debido a que las clases son +1 y -1, el HSO está definido por el margen máximo de separación entre las clases. Con base en esto, los vectores de soporte se definen como:  $\vec{w} \cdot \vec{x} + b = +1$  y  $\vec{w} \cdot \vec{x} + b = -1$ , los cuales son paralelos al HSO el cual se expresa por  $\vec{w} \cdot \vec{x} + b = 0$ . El margen máximo de los vectores de soporte está expresado por  $2/\|\vec{w}\|$ .

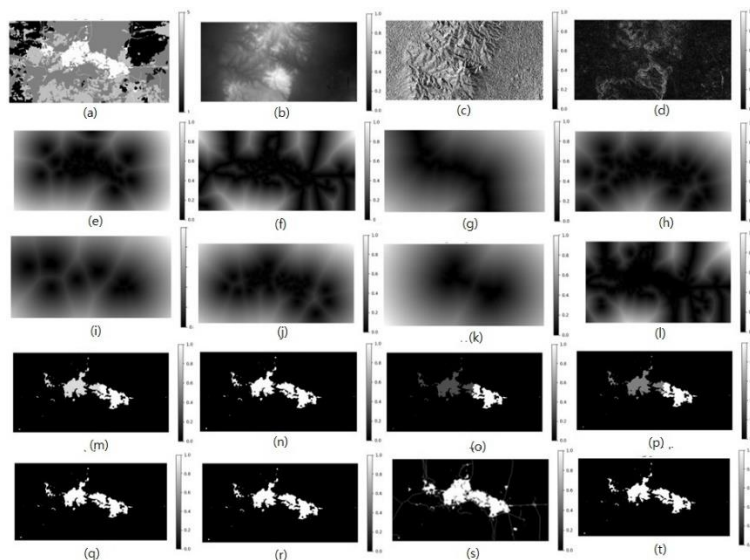
Sin embargo, inicialmente las SVM fueron propuestas de manera rígida, es decir, que los datos son completamente separables.

No obstante, existen datos que no lo son. En estos casos se introduce una variable de holgura  $\xi_i$ , la cual permite errores en la clasificación, pero estos errores son penalizados.

Al incluir este nuevo parámetro la SVM se suaviza. El problema primario cambia de forma y ahora se busca minimizar la expresión:

$$J(w, \xi) = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^n \xi_i, \quad (1)$$

Sujeta a la condición:  $y_i[w_i \cdot x_i + b] + \xi_i - 1 \geq 0, \xi_i \geq 1, i = 1, \dots, n$ ,



**Fig. 2.** Datos de entrenamiento del año 2000 usadas en el estudio. (a) Uso de Suelo, (b) Modelo Digital de elevaciones, (c) Dirección de la pendiente, (d) Pendiente. (e)-(l) variables de distancia y (m)-(t) Variables de densidad.

donde  $\xi_i$  es el parámetro de holgura y  $c$  es el parámetro de penalización a los errores de clasificación [13]. En este nuevo enfoque, el objetivo es encontrar el HSO que minimice los errores de clasificación y maximice la separación entre los vectores de soporte. Mientras el valor de  $c$  se incrementa, se obtiene un margen estrecho, esto minimiza el número de clasificaciones erróneas. Por otro lado, si  $c$  disminuye, se están permitiendo más clasificaciones erróneas [27].

Para abordar la no linealidad, los datos son transportados a un espacio de alta dimensionalidad. Esto se logra realizando una proyección matemática de los datos usando un kernel. Dicho kernel se puede definir como:  $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ . Algunas funciones de kernel populares son la lineal  $k(x, y) = x_i \cdot x_j$ , función de base radial (RBF)  $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ , y la función polinomial  $k(x, y) = (1 + x_i \cdot x_j)^q$ , donde  $\gamma$  y  $q$  son parámetros correspondientes al kernel [13].

### 2.5. Métricas de evaluación

Al entrenar un modelo de aprendizaje automático, es necesario evaluar el rendimiento del modelo. Existen diferentes técnicas de evaluación; la elección depende del problema y su contexto. En la clasificación, algunas de las evaluaciones disponibles son exactitud, precisión, área bajo la curva, entre otros [28]. Estas métricas se calculan usando la matriz de confusión [13], la cual nos introduce a los términos de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN).

Como se puede visualizar en la Tabla 3, los TP se refiere a los píxeles que son construidos y el modelo los clasificó como construidos. FP indica los píxeles que son



no construidos y el modelo lo coloco como construidos. TN, píxeles clasificados como no construidos y en la etiqueta real son no construidos. Por último, FN píxeles que por el modelo se indicaron como no construidos, pero en la etiqueta real son construidos [28].

En este estudio se utilizó: la exactitud, precisión y F1-Score. La exactitud indica la proporción de clasificaciones correctas realizadas por el modelo respecto al total de muestras, se calculó usando (2). La fórmula (3) se utiliza para calcular la precisión, la cual se centra en la proporción de TP respecto al total de muestras clasificadas como positivas. Por último, el F1-Score calculado con la ecuación (5) es la media armónica de la precisión y recall (ecuación 4). El recall se refiere a la proporción de TP respecto a las instancias que son positivas [29]:

$$\text{Exactitud} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

$$\text{Precisión} = \frac{TP}{TP + FP}, \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (4)$$

$$\text{F1-score} = \frac{2 * (\text{Recall} * \text{Precisión})}{(\text{Recall} + \text{Precisión})}. \quad (5)$$

## 2.6. Shapley Additive exPlanations (SHAP)

SHAP fue propuesto por Lundberg y Lee [30], se utiliza para asignar un valor de importancia a cada característica para una predicción específica. Esto se basa en la teoría de los valores Shapley, que proviene de la teoría de juegos cooperativos. Los valores Shapley miden la contribución marginal de cada característica al resultado final utilizando la siguiente fórmula [5]:

$$\Phi_i = \sum_{S \in N} \frac{|S|! (n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)], \quad (6)$$

donde N es el número de instancias que tienen n características, S representa un subconjunto de las características n y v representa las características de entrada del conjunto S.

## 3. Resultados

Antes de desarrollar el modelo predictivo se llevó a cabo un análisis de los píxeles correspondientes a las etiquetas de construido y no construido. Con esto se calculó la tasa de cambio de los píxeles entre los años estudiados (Tabla 4). La tasa de cambio se mantuvo constante en los dos periodos analizados. El aumento está enfocado principalmente al noreste en el municipio de Guadalupe. En el municipio de Zacatecas, el crecimiento está focalizado en el área suroeste de la ciudad.

**Tabla 3.** Matriz de confusión de píxeles construidos-no construidos.

<b>Predicción observada</b>	<b>Construido</b>	<b>No-construido</b>
Construido	Verdaderos Positivos (TP)	Falsos Negativos (FN)
No construido	Falsos Positivos (FP)	Verdaderos Negativos (TN)

**Tabla 4.** Tasa de crecimiento de la zona de estudio en los periodos 2000 a 2010 y 2010 a 2020.

	<b>2000-2010</b>	<b>2010-2020</b>
Cantidad de píxeles que cambiaron	16369	14578
Cantidad de píxeles que no cambiaron	343133	344924
Tasa de cambio	4.55%	4.06%

### 3.1. Modelo SVM

Se realizó una experimentación modificando diferentes valores de los parámetros del SVM penalizado, basado en el trabajo [13]. Los parámetros que tomaron diferentes valores en esta experimentación fueron: parámetro  $c$ , kernel, gama y grado  $q$  del polinomio. Para el parámetro  $c$  se seleccionaron los valores 0.1, 1, 10, 100. Asimismo, se seleccionó el kernel radial (RBF) con el parámetro gama de 1, 2, 3. Finalmente, para el kernel polinomial se eligieron los valores de grado de polinomio de 1, 2, 3, 4 y 5. Las combinaciones se realizaron con el propósito de obtener el modelo predictivo más adecuado.

La selección del modelo se llevó a cabo usando las métricas de exactitud (entrenamiento y validación), F1-Score y Precisión. Estas métricas se calcularon usando la matriz de confusión de los píxeles construidos y no construidos.

Los modelos resultantes tienen una exactitud de entrenamiento y validación elevada (mayores al 90 %). En la Figura 3 se pueden apreciar los resultados. En la exactitud de entrenamiento, los valores más altos son aquellos obtenidos con los modelos con kernel RBF. Por otro lado, en la exactitud de validación, los mejores valores se encuentran en los modelos con kernel polinomial con grado de polinomio de cinco. Sin embargo, debido al desbalance de las clases, se decidió apoyarnos del F1-Score.

Los resultados más elevados en el F1-Score se encuentran con el kernel polinomial y grado del polinomio cinco, similar a la exactitud de validación. Estos resultados están alrededor del 80 %. Los resultados en la precisión son similares al F1-Score, los mejores se encuentran con el kernel polinomial en los valores del 80 %.

En el caso de los valores de  $c$  los mejores resultados se encuentran en los valores de 100 y 0.1. En la exactitud presentada en el entrenamiento y precisión, los mejores valores están con el valor  $c$  de 100. Mientras que la exactitud de validación y F1-Score se encuentran con el valor de 0.1.

El modelo que se eligió como el mejor, se debe a que tiene uno de los mejores F1-Score y un balance con las demás métricas. Este modelo es el de kernel polinomial, con valor  $c$  de 0.1 y un grado de polinomio de cinco, la exactitud de entrenamiento es de un 92.4%, exactitud de validación 93% y F1-Score de un 86.3%. En la Figura 3 en el recuadro rojo se indica el modelo seleccionado. En la Figura 4 se encuentra la comparación del resultado del modelo respecto a la realidad del año 2020.

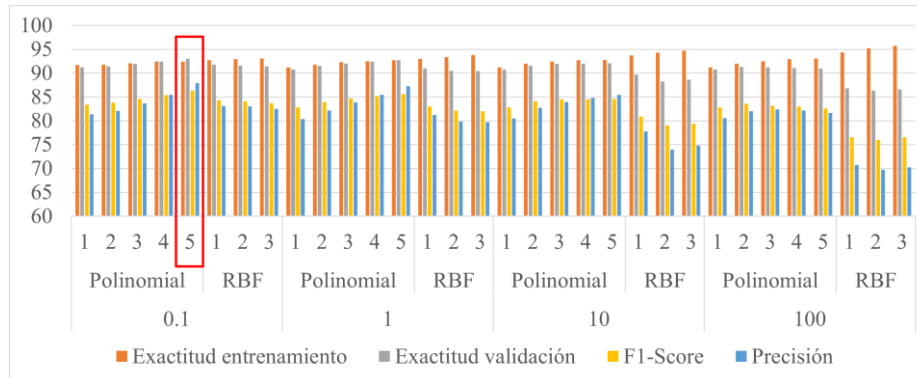


Fig. 1. Resultado de la experimentación, en el cuadro rojo se encuentra el que se considera el mejor modelo.

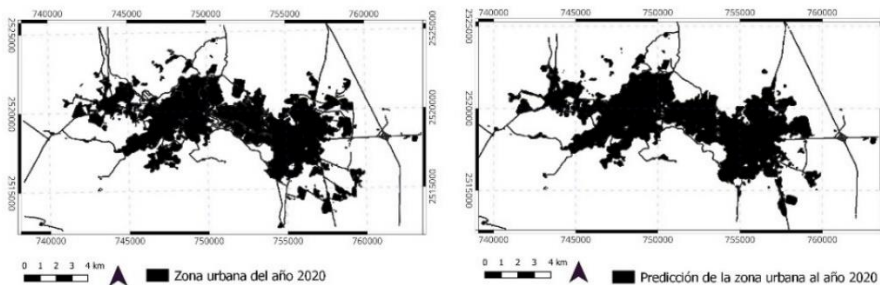


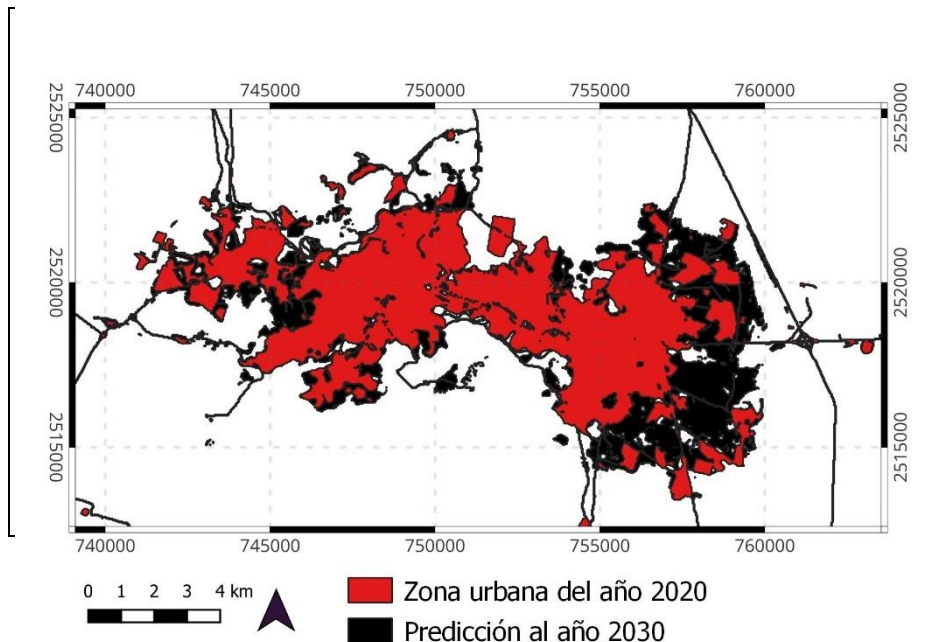
Fig. 4. A la izquierda se encuentra las zonas urbanizadas en el año 2020, en la de la derecha la predicción de urbanización realizada por el modelo SVM penalizado.

### 3.2. Factores de importancia

Los valores SHAP se calcularon seleccionando un subgrupo de los datos. La selección se llevó a cabo aplicando el algoritmo de K-means a los datos y posteriormente seleccionando los puntos más cercanos a los centroides. En la Figura 5 se observan los valores SHAP, lo cual nos ayuda a comprender el comportamiento de las variables independientes que determinan si los píxeles se urbanizan o no.

La variable que más impacta al modelo es el tipo de uso de suelo. Mientras más alto es el valor, es más probable que se urbanice. Esto quiere decir que si ya es urbanizado permanece urbanizado y si es suelo desnudo o vegetación cercana a zona urbana, el clasificador lo coloca como urbanizado.

La segunda variable que tiene mayor influencia es la probabilidad de urbanización. Esto indica que el pixel se urbaniza en dependencia de cuantos píxeles dentro de su vecindario de 3x3 ya es urbanizado. Si el valor de esta probabilidad es alto el modelo lo indica como urbanizado. Esto va de la mano con la tercera variable la cual es la distancia a zona urbana. Si la distancia a la zona urbana es pequeña el modelo tiende a colocarlo como urbanizado. Por otro lado, la distancia a los centros de la ciudad es la



**Fig. 3.** Zona urbanizada en el año 2020 y la predicción al año 2030 resultado del modelo predictivo.

cuarta variable que tiene importancia. Cuanto menor la distancia al centro, es más llamativa a la urbanización.

Por el contrario, las variables que no tuvieron un impacto fueron la elevación del terreno, los ingresos trimestrales, la pendiente y la dirección.

### 3.3. Predicción del crecimiento urbano

Con el modelo de SVM penalizado desarrollado y los datos del año 2020 se realizó la predicción de las zonas urbanas para el año 2030, el cual se muestra en la Figura 6. En el resultado se puede observar que el crecimiento de la zona metropolitana es mayor en la zona de Guadalupe. Esto es algo que ha estado constante a lo largo de los años.

En cuanto a la dirección de la expansión en la zona de Guadalupe es más densa en la zona sureste de la zona de estudio. De forma contraria la expansión en el municipio de Zacatecas esta más concentrada en la zona central-oeste. Disminuyendo la expansión de la zona suroeste de los años anteriores.

La tasa de cambio de los píxeles en esta predicción es del 8.11%. El doble de lo encontrado en el análisis de los periodos anteriores.

## 4. Discusión

La rápida urbanización es la causa principal del cambio de uso de suelo de áreas vegetales, agricultura y suelo desnudo a urbanización [31]. En este estudio, hemos

explorado los patrones y tendencias del crecimiento urbano en la ciudad de Zacatecas-Guadalupe durante el periodo de 2000 a 2020. Utilizando SVM penalizado, el cual es un algoritmo robusto, e integrando los valores SHAP.

Como resultado de los valores SHAP se pudo observar que la cercanía a áreas ya urbanizadas y si el píxel es principalmente suelo desnudo, existe más probabilidad de que este se urbanice. Lo cual es consistente con otros estudios como [13,15]. Sin embargo, a pesar de que algunos autores consideran importante la elevación y pendiente [5], en los valores SHAP encontrados en este estudio indican que estas variables no tienen impacto en la urbanización de la zona, lo cual es algo que encuentran otros autores [31,32].

Las distancias a los centros de la ciudad, a escuelas y a calles también tienen un impacto en la predicción final. Lo cual también se ha visto en otros artículos [13]. En la predicción del 2030 se observó que la distancia a calles tiene una alta influencia. Ya que la predicción está concentrada en zonas que tienen más vías principales de primer y segundo orden, ubicadas en el área de Guadalupe.

Mientras que en el área de Zacatecas se tiene una menor densidad de este tipo de vías, lo que provoca menor expansión en esa dirección.

Hablando de las variables culturales, distancia a templos y densidad de población religiosa no tuvieron un impacto significativo de acuerdo a los resultados. Por lo tanto, se consideraría utilizar otras variables que sean más llamativas para habitar un lugar.

Por último, consideramos que los planeadores de ciudades deben tener en cuenta la importancia de la construcción de calles, ya que estas incrementan la probabilidad de construcción y la cercanía de los nuevos asentamientos al centro de la ciudad.

## 5. Conclusiones

Se analizó la predicción de la expansión urbana de la zona Zacatecas-Guadalupe usando un modelo de SVM penalizado apoyado del método SHAP. Los resultados muestran que el uso de suelo tiene un impacto alto, siendo las áreas de vegetación cercanas a las zonas urbanas y los suelos desnudos más probables de ser urbanizados, al igual que las áreas cercanas a zonas ya urbanizadas. Por otro lado, la pendiente del terreno y su dirección no tienen mucha influencia en el resultado del modelo.

Consideramos que la predicción puede ayudar a la planeación de la ciudad, principalmente evitando el descontrol del tráfico, debido al impacto que tienen las calles dentro del modelo y la forma de la ciudad.

A pesar de que el modelo refleja la interacción de las variables y cómo la ciudad se expande, consideramos que existen variables que no se exploraron, las cuales pueden aumentar la exactitud de los modelos. Como trabajo futuro, se consideraría incluir variables de proyección de población, valor de los predios y otras variables culturales que demuestren una relación más fuerte. Así como realizar un análisis comparativo con otros algoritmos como bosques aleatorios y boosting categórico.

**Agradecimientos.** El autor principal agradece el apoyo recibido por el Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) a través del Programa de Becas para Estudios de Posgrado en México.

## Referencias

1. Zubicaray, G., Brito, M., Ramírez-Reyes, L., García, N., Macías, J.: Las ciudades mexicanas: Tendencias de expansión y sus impactos. London: Coalition for Urban Transitions (2021)
2. Secretaría de Desarrollo Social, Secretaria de Gobernación, Consejo Nacional de Población (México): Catálogo: Sistema Urbano Nacional (2012)
3. Deng, Y., Fu, B., Sun, C.: Effects of Urban Planning in Guiding Urban Growth: Evidence from Shenzhen, China. *Cities*, vol. 83, pp. 118–128 (2018). DOI: 10.1016/j.cities.2018.06.014.
4. Garrocho-Rangel, C., Chávez-Soto, T., Vilchis-Mata, I., Jiménez-López, E.: Un modelo de expansión urbana no estacionario en el espacio: Autómatas celulares y regresión geográficamente ponderada (2022)
5. Kim, M., Kim, G.: Modeling and Predicting Urban Expansion in South Korea Using Explainable Artificial Intelligence (XAI) Model. *Applied Sciences (Switzerland)* Vol. 12 (2022). DOI: 10.3390/app12189169.
6. González-Madriral, J., Solano-Lamphar, H., Ramírez, M.: La contaminación lumínica como aproximación a la planeación urbana de ciudades mexicanas, *Euere*, vol. 46, no. 138, pp. 155–174 (2020). DOI: 10.4067/S0250-71612020000200155.
7. Nugroho, F., Al-Sanjary, OI.: A Review of Simulation Urban Growth Model. *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 17–23 (2018). DOI: 10.14419/ijet.v7i4.11.20681.
8. Wang, S.W., Munkhnasan, L., Lee, W.K.: Land Use and Land Cover Change Detection and Prediction in Bhutan's High-Altitude City of Thimphu, Using Cellular Automata and Markov Chain. *Environmental Challenges*, vol. 2, pp. 100017 (2021). DOI: 10.1016/j.envc.2020.100017.
9. Kara, C., Doratli, N.: Predict and Simulate Sustainable Urban Growth by Using Gis and MCE based CA. Case of Famagusta in Northern Cyprus. *Sustainability (Switzerland)*, vol. 13 (2021). DOI: 10.3390/su13084446.
10. Liang, X., Liu, X., Li, X., Chen, Y., Tian, H., Yao, Y.: Delineating Multi-scenario Urban growth Boundaries with a CA-based FLUS Model and Morphological Method. *Landscape and Urban Planning*, vol. 177, pp. 47–63 (2018). DOI: 10.1016/j.landurbplan.2018.04.016.
11. Chaturvedi, V., de Vries, W.T.: Machine Learning Algorithms for Urban Land use Planning: A review. *Urban Science*, vol. 5, no. 3, pp. 68 (2021). DOI: 10.3390/urbansci5030068.
12. Li, X., Gong, P.: Urban Growth Models: Progress and Perspective. *Sci Bull (Beijing)*, vol. 61, pp. 1637–1650 (2016). DOI: 10.1007/s11434-016-1111-1.
13. Karimi, F., Sultana, S., Shirzadi-Babakan, A., Suthaharan, S.: An Enhanced Support Vector Machine Model for Urban Expansion Prediction. *Comput Environ Urban Systems*, vol. 75, pp. 61–75 (2019). DOI: 10.1016/j.compenvurbsys.2019.01.001.
14. Mirbagheri, B., Alimohammadi, A.: Integration of Local and Global Support Vector Machines to Improve Urban Growth Modelling. *Canadian Historical Review*, vol. 7 (2018). DOI: 10.3390/ijgi7090347.
15. Karimi, F., Sultana, S., Babakan, A.S., Suthaharan, S.: Urban Expansion Modeling using An Enhanced Decision Tree Algorithm. *Geoinformatica*, vol. 25, pp. 715–731 (2021). DOI: 10.1007/s10707-019-00377-8.
16. Asif, M., Kazmi, J.H., Tariq, A., Zhao, N., Guluzade, R., Soufan, W., Aslam, M.: Modelling of Land use and Land Cover Changes and Prediction Using CA-Markov and Random Forest. *Geocarto International*, vol. 38, no. 1, pp. 2210532 (2023). DOI: 10.1080/10106049.2023.2210532.
17. Reyes-Pérez, O., Loredó-Sánchez, M.A., Castañeda-Camacho, X., Aguilar-Robledo, M.: Methodology for Mining Economic Assimilation in Zacatecas, Mexico. *Economía, sociedad y territorio*, vol. 20, no. 62, pp. 871–902 (2020). DOI: 10.22136/est20201415.

18. Garbutt, O.P.: *Metrópolis de México 2020*. Primera edición (2024)
19. Garrocho, C., Soto, T.C., Jiménez-López, E.: Autómata celular metro: Laboratorio experimental de expansión urbana. *Boletín de la Sociedad Botánica de México*, vol. 3, no. 3, pp. 149–175 (2021)
20. Garrocho, C., Jiménez, E., Chávez-Soto, T.: Expansión de la ciudad: Un instrumento de simulación de escenarios para los sectores público y privado. *La situación demográfica de México*, vol. 2, no. 2 (2020)
21. INEGI: Diversidad. Zacatecas. <https://cuentame.inegi.org.mx/monografias/informacion/zac/poblacion/diversidad.aspx> (2024)
22. Abrams, M., Crippen, R.: *ASTER GDEM V3 (ASTER Global DEM), User Guide* (2019)
23. INEGI: XII Censo General de Población y Vivienda 2000. <https://www.inegi.org.mx/programas/ccpv/2000/> (2024)
24. INEGI: Instituto Nacional de Estadística y Geografía Censo de Población y Vivienda 2010. <https://www.inegi.org.mx/programas/ccpv/2010/> (2010)
25. INEGI: Instituto Nacional de Estadística y Geografía Censo de Población y Vivienda 2020. <https://www.inegi.org.mx/programas/ccpv/2020/> (2024)
26. González, R., Barrientos, A., Toapanta, M., Del Cerro, J.: Aplicación de las Máquinas de Soporte Vectorial (SVM) al diagnóstico clínico de la enfermedad de párkinson y el temblor Esencial. *RIAI - Revista Iberoamericana de Automática e Informática Industrial*, vol. 14, pp. 394–405 (2017). DOI: 10.1016/j.riai.2017.07.005.
27. Awad, M., Khanna, R.: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Springer Nature., pp. 268 (2015)
28. Panesar, A.: *Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes*. Apress Media LLC (2020)
29. Pagano, T.P., Loureiro, R.B., Lisboa, F.V., Cruz, G.O., Peixoto, R.M., Guimarães, G.A.D. S., Nascimento, E.G.S.: Context-based Patterns in Machine Learning Bias and Fairness Metrics: A Sensitive Attributes-based Approach. *Big Data and Cognitive Computing*, vol. 7, no. 1 (2023). DOI: 10.3390/bdcc7010027.
30. Lundberg, S.: A Unified Approach to Interpreting Model Predictions. *arXiv preprint arXiv:1705.07874* (2017)
31. Rana, M.S., Sarkar, S.: Prediction of Urban Expansion by Using Land Cover Change Detection Approach. *Heliyon*, vol. 7, no. 11 (2021). DOI: 10.1016/j.heliyon.2021.e08437.
32. Frimpong, B.F., Molkenhain, F.: Tracking Urban Expansion using Random Forests for the Classification of Landsat Imagery (1986–2015) and Predicting Urban/built-up Areas for 2025: A Study of the Kumasi Metropolis, Ghana. *Land*, vol. 10, no. 1, pp. 44 (2021). DOI: 10.3390/land10010044.





## Identificación de variables causales de la congestión de tráfico mediante la prueba de Granger

Ernesto De la Cruz-Nicolás, Hugo Estrada-Esquivel,  
Alicia Martínez-Rebollar, Odette Alejandra Pliego-Martínez

Tecnológico Nacional de México  
Centro Nacional de Investigación y Desarrollo Tecnológico,  
México

{d21ce090, hugo.ee, alicia.mr, d21ce092}@cenidet.tecnm.mx

**Resumen.** La identificación de variables causales de la congestión de tráfico que surgen en las grandes ciudades mediante la prueba de Granger es fundamental para comprender las variables que contribuyen al origen de los puntos de congestión de tráfico vehicular en áreas urbanas. La prueba de Granger evalúa la causalidad entre dos series temporales, lo que permite identificar si una variable precede a otra en el tiempo y, por lo tanto, tiene un efecto causal sobre ella en el contexto de identificar que variables impactan en la congestión del tráfico. Al aplicar la prueba de Granger a datos de tráfico e incidencias viales, es posible determinar qué variables impactan de manera directa e indirecta a la congestión de tráfico, como la velocidad de los vehículos, las condiciones climáticas, la infraestructura vial o eventos específicos, pueden ser considerados como causas significativas de la congestión. Identificar las variables causales de la congestión de tráfico ofrece una serie de beneficios significativos como el desarrollo de estrategias efectivas de gestión del tráfico, promover la sostenibilidad ambiental, seguridad vial. La investigación en cuestión se suma a otras técnicas de la literatura que se dedican a identificar, correlacionar y relacionar variables causales de la congestión de tráfico, con el propósito de perfeccionar la precisión de los modelos predictivos en esta área.

**Palabras clave:** Causalidad, congestión, tráfico, impacto.

### Identification of Causal Variables of Traffic Congestion Using the Granger Test

**Abstract.** The identification of causal variables of traffic congestion that arise in large cities through Granger causality testing is fundamental to understanding the factors contributing to the origin of traffic congestion points in urban areas. The Granger test assesses causality between two time series, allowing for the identification of whether one variable precedes another in time and thus has a causal effect on it in the context of identifying which variables impact traffic congestion. By applying the Granger test to traffic data and road incidents, it is possible to determine which variables directly and indirectly impact traffic congestion, such as vehicle speed, weather conditions, road infrastructure, or

specific events, which can be considered significant causes of congestion. Identifying the causal variables of traffic congestion offers a range of significant benefits such as the development of effective traffic management strategies, promoting environmental sustainability, and road safety. The research in question adds to other techniques in the literature dedicated to identifying, correlating, and relating causal variables of traffic congestion, with the purpose of refining the accuracy of predictive models in this area.

**Keywords:** Causality, congestion, traffic, impact.

## 1. Introducción

La congestión de tráfico es un fenómeno que aparece en las ciudades urbanizadas de todo el mundo, con consecuencias significativas en términos de tiempo de viaje perdido en los desplazamientos, afectación a la salud por el aumento de emisiones de gases de efecto invernadero, contaminación entre otros. En el intento por mitigar el problema de la congestión de tráfico, es fundamental comprender las variables subyacentes que contribuyen al origen de la congestión vehicular.

Existen en la literatura un extenso trabajo relacionado con modelos predictivos que han utilizado variables significativas para predecir congestiones de tráfico. La regresión lineal o múltiple, se han utilizado para predecir la congestión de tráfico en función de variables significativas como el flujo de tráfico, la velocidad promedio, las condiciones meteorológicas, la hora del día, eventos especiales y la densidad poblacional, entre otros. Estos modelos pueden proporcionar estimaciones cuantitativas de la probabilidad o el grado de congestión en diferentes momentos y ubicaciones [1, 2, 3].

Las redes neuronales artificiales utilizan las variables significativas que causan la congestión de tráfico como entradas para predecir la congestión de tráfico en función de múltiples factores, incluyendo el historial de tráfico, datos de sensores, eventos especiales y condiciones ambientales [4, 5, 6].

Los modelos de series temporales, se utilizan para predecir la congestión de tráfico en función de patrones temporales pasados. Estos modelos incorporan variables significativas como la hora del día, el día de la semana, días festivos y eventos especiales para prever futuros niveles de congestión con base en tendencias históricas [7, 8, 9].

Los modelos de simulación de tráfico, como VISSIM o SUMO, simulan la congestión de tráfico en función de diferentes variables, como la geometría de la carretera, la demanda de tráfico, los semáforos y las condiciones meteorológicas. Estos modelos predicen las congestiones de tráfico en tiempo real [10, 11, 12].

En este contexto, la importancia de identificar las variables causales de la congestión de tráfico es fundamental para incrementar la precisión y otras métricas en los modelos predictivos y otras técnicas para la caracterización, pronóstico y predicción de la congestión de tráfico. Las técnicas estadísticas ofrecen un enfoque poderoso para analizar la relación causal entre diferentes variables y la congestión de tráfico e identificar las variables de mayor significancia en el origen de la congestión de tráfico. La prueba de Granger, es una herramienta que permite evaluar si una serie temporal de una variable puede predecir de manera significativa otra serie temporal,

proporcionando así información valiosa sobre las variables causales detrás de la congestión de tráfico.

En este artículo, exploraremos la aplicación de la prueba de Granger para identificar las variables causales de la congestión de tráfico, a partir de una propuesta de un conjunto de variables relacionados con el tráfico e incidencias viales que impactan en el origen de la congestión de tráfico. A través de esta investigación, se contribuye a una comprensión más profunda de los factores que influyen en la congestión de tráfico, lo que a su vez puede proporcionar variables significativas de congestión de tráfico a los modelos predictivos para mejorar su precisión.

## **2. Trabajos relacionados**

La rápida expansión de la urbanización en las grandes ciudades del mundo ha generado un aumento sin precedentes en el origen de múltiples puntos de congestión de tráfico, un problema constante con repercusiones significativas en la salud pública y el medio ambiente, como se destacan los hallazgos en el trabajo de [14]. La congestión de tráfico, al ser un desafío complejo y persistente, ha motivado una amplia gama de investigaciones orientadas al desarrollo de estrategias para mitigar este problema, tal como lo señala Yue [24].

Sin embargo, es fundamental comprender qué variables ejercen una mayor influencia en el origen de la congestión de tráfico, ya que esto no solo mejora la precisión de los modelos predictivos, sino que también optimiza otras técnicas y estrategias destinadas a mitigar la congestión de tráfico y se evita caer en la maldición de la multidimensionalidad que refiere al número de variables de un conjunto de datos, lo que puede dificultar el análisis y el modelado de manera efectiva. Esta maldición puede provocar problemas como el aumento de la complejidad computacional, la dispersión de los datos y el sobreajuste de los modelos de acuerdo al trabajo de [25].

La literatura existente ha llevado a cabo una serie de estudios para evaluar las variables que causan la congestión de tráfico. Por ejemplo, en el trabajo de Kardani-Yazd [13], encuentran relaciones instantáneas entre variables climáticas/no climáticas locales y el flujo de tráfico como variables causales. Por otro lado, en la investigación de Iro [16], identifican que un 66% de la congestión del tráfico se debe a factores humanos, mientras que un 34% se atribuye a condiciones físicas de las calles.

Además, se ha observado que aspectos como los accidentes automovilísticos y la duración de la atención a los mismos, como lo señala Chen [17] son variables que impactan. Mientras el área de calle per cápita, la propiedad de vehículos y las millas recorridas por vehículo, como lo describe Bian [18], juegan un papel importante en la congestión del tráfico. Mahona [19] también identifica a las cruces existentes en las calles como influencia a la congestión de tráfico. Asimismo, Yu [20] ha resaltado la importancia de las intersecciones de calles como una de las principales variables que ocasionan la congestión de tráfico.

El análisis de puntos de atracción como centros comerciales, escuelas, oficinas y tiendas, como se describe en el trabajo de Gullotta [21], también se ha revelado como relevante para comprender el origen de la congestión de tráfico. Rahman [22], utilizando el Modelado de Ecuaciones Estructurales, identificó el agrupamiento de ingresos y empleo como variables predominantes en la congestión del tráfico. Por otro

lado, Pi [23] sugiere que la combinación del diseño de las calles, eventos en las vías, condiciones de la calle y malos hábitos de los conductores, puede contribuir a la congestión del tráfico.

La identificación de variables causales de la congestión de tráfico ha permitido la creación de indicadores que evalúan la calidad de la infraestructura vial y la movilidad vehicular. En el trabajo de Jia [15], se analizan 15 indicadores relevantes y se examinan las variables como la densidad de población urbana, el Producto Interno Bruto, factores sociales, económicos y la oferta de transporte público, entre otros.

La congestión de tráfico es un fenómeno complejo influenciado por múltiples variables. La prueba de Granger permite examinar la relación causal entre variables y comprender cómo influyen en la congestión de tráfico.

La comprensión de las variables causales es crucial para construir modelos predictivos precisos de congestión de tráfico. La prueba de Granger proporciona información sobre qué variables pueden utilizarse como predictores para pronosticar la congestión de tráfico futura. La prueba de Granger permite evaluar la relación causal entre variables y validar hipótesis sobre las variables que contribuyen a la congestión de tráfico. Esto ayuda a confirmar o refutar suposiciones previas y a generar nuevas ideas para investigaciones futuras.

### **3. Metodología para identificar las variables causales de la congestión de tráfico mediante la prueba de Granger**

La metodología para identificar las variables que causan la congestión de tráfico mediante la técnica estadística de Granger consta de cinco fases. La primera fase implica la definición del caso de estudio, donde se selecciona la zona geográfica que será objeto de análisis de la congestión de tráfico. En la segunda fase, se lleva a cabo la recolección de datos de tráfico e incidencias viales, donde se recopila información detallada y precisa sobre diversos aspectos relacionados con el flujo vehicular y los incidentes que afectan la circulación. En la tercera fase, se realiza el preprocesamiento de los datos de tráfico e incidencias viales. Durante esta etapa, los datos se transforman para mejorar su normalidad y se estandarizan debido a la diversidad de variables presentes.

La cuarta fase implica la implementación de la prueba de Granger a las variables de tráfico e incidencias viales con respecto a la congestión de tráfico (jamFactor). Aquí, se utiliza el método estadístico de Granger para identificar posibles relaciones causales entre estas variables. Finalmente, en la quinta fase, se lleva a cabo la validación de las variables causales de la congestión de tráfico encontradas mediante la prueba de Granger. Durante esta fase, se emplean las variables causales de la congestión de tráfico como entrada para el algoritmo predictivo de Random Forest, analizando su precisión en función de las variables de entrada. La metodología propuesta consiste en un conjunto de fases diseñadas para identificar las variables causales más significativas entre las 25 variables propuestas en esta investigación. Este enfoque complementa otras técnicas de identificación, relación y correlación de variables de mayor impacto relacionadas con la congestión de tráfico. A continuación, se detalla cada fase de la metodología.

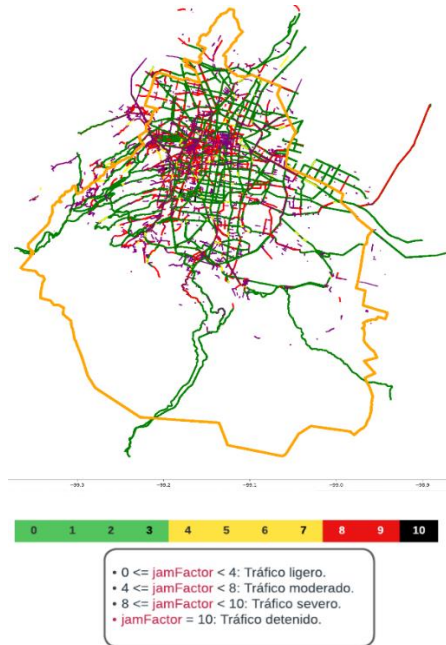


Fig. 1. Niveles de congestión de tráfico (jamFactor) de las calles de la Ciudad de México.

### 3.1. Caso de estudio

Una muestra de 6708 calles de las 16 alcaldías de la Ciudad de México será el caso de estudio para recopilar datos de tráfico e incidencias. En la Figura 1 se muestra la movilidad vehicular en las calles de la Ciudad de México a las 6:25 p.m. de un día martes. Las calles se representan con diferentes colores: verde, amarillo, rojo y negro, indicando distintos niveles de congestión de tráfico. Se destacan algunas calles en color morado, que representan aquellas con incidencias viales. El jamFactor oscila entre 0 y 10, donde un valor cercano a 0 indica una congestión de tráfico mínima, lo que implica un flujo vehicular ligero con una velocidad de desplazamiento adecuada. Por el contrario, un valor de jamFactor cercano a 10 indica una congestión de tráfico severa, con una velocidad de desplazamiento considerablemente reducida, mostrados en la Figura 1.

### 3.2. Recolección de datos de tráfico e incidencias viales

La recolección de datos de tráfico e incidencias viales se llevó a cabo durante un período que abarcó desde el 1 de septiembre hasta el 31 de diciembre de 2023, lo que representó un total de 122 días de extracción de datos. La recolección de datos se hizo mediante la aplicación de Here Maps, la cual fue configurada utilizando un script en Python para recopilar información en intervalos de solicitud de cada 5 minutos a lo largo de las 24 horas del día. En total, se logró reunir un conjunto de datos denominado

**Tabla 1.** Variables de tráfico e incidencias viales (Tráfico-Incidencias).

Conjunto de datos	Variable	Descripción
Tráfico	jamFactor	Describe los niveles de tráfico en un escala del 0 al 10.
	Bbox_Traffic	Cuadro delimitador donde se origina el tráfico.
	Free_Flow	Velocidad de desplazamiento sin ninguna obstrucción.
	Length	Longitud de la calle.
	Number_Segments	Número de segmentos que contiene la calle.
	Speed	La velocidad esperada.
	Speed_Uncapped	Velocidad sin restricción.
	Daily_Traffic	Día en que se monitorea el tráfico.
	Day_Off_Traffic	Indica si el día es festivo en cuanto a tráfico.
	Traffic_Hour	Hora del monitoreo de tráfico.
	Traffic_Minute	Minuto del monitoreo de tráfico.
	Traffic_Day_Number	Día de monitoreo de tráfico (1= lunes, 2=martes...7=domingo).
	Work_Day_Traffic	Monitoreo del tráfico en semana laboral o fin de semana
	Incidencias	Bbox_Incident
Road_Closed		Calle cerrada por incidencia (0 abierto y 1 cerrado).
Criticality		Gravedad de la incidencia.
Type		Describe el tipo de incidencia.
Incident_Day		Día del monitoreo de la incidencia.
Day_Off_Incident		Indica si el día es festivo en cuanto a incidencia.
Incident_Hour		Hora del monitoreo de la incidencia.
Incident_Month		Mes del monitoreo de la incidencia.
Incident_Minute		Minuto del monitoreo de la incidencia.
Incident_Day_Number		Día de incidencia (1= lunes, 2=martes...7=domingo).
Work_Day_Incident		Incidencia por laboral o fin de semana (1 o 0 respectivamente).
Category_Number		Categoría de la incidencia.
Time	Tiempo en minutos.	

Tráfico-Incidencias con una impresionante cantidad de 5, 000, 000 registros de datos, los cuales incluyen 26 variables.

Las 26 variables relacionados con el tráfico e incidencias viales recopiladas se describen en la tabla 1.

### 3.3. Preprocesamiento de datos de tráfico e incidencias viales

El preprocesamiento de datos de tráfico e incidencias viales se llevó a cabo mediante dos pasos para mejorar la calidad y uniformidad de los datos. Esto incluyó tanto la transformación de datos para mejorar la normalidad como la estandarización para abordar las diferentes escalas de cada variable. A continuación, se describen en detalle los pasos realizados:

**Paso 1, transformar los datos:** La transformación se hizo mediante la raíz cuadrada, el z-score y la transformación cúbica, aplicadas mediante las ecuaciones 1, 2 y 3, respectivamente, en el conjunto de datos de Tráfico-Incidencias:

$$\text{Variable normalizada} = \sqrt{\text{Variable original}}, \quad (1)$$

$$z = \frac{(x-\mu)}{\sigma}, \quad (2)$$

dónde:  $x$  Es el valor individual de la variable,  $\mu$  Es la media de la distribución de la variable y  $\sigma$  Es la desviación estándar de la distribución de la variable:

$$\text{Variable normalizada} = \sqrt[3]{\text{Variable original}}. \quad (3)$$

Los hallazgos derivados de la normalización del conjunto de datos de Tráfico-Incidencias señalan que la aplicación de la transformación cúbica resulta en una notable mejora en la normalidad de este conjunto de datos.

*Paso 2, estandarizar los datos:* Se realizó la estandarización el conjunto de datos de Tráfico-Incidencias, en una escala de 0 a 1. Esta estandarización se efectuó utilizando la ecuación número 4:

$$x_{norm} = \frac{(x-x_{min})}{(x_{max}-x_{min})}, \quad (4)$$

dónde:

$x$  Es el valor original que se desea normalizar,  $x_{min}$  Es el valor mínimo en el conjunto de datos original y  $x_{max}$  Es el valor máximo en el conjunto de datos original.

La estandarización produjo que todas las variables del conjunto de datos de Tráfico-Incidencias fueran ajustadas a un intervalo de valores que oscila entre 0 y 1. Este proceso garantiza que todas las variables estuvieran en una escala uniforme y comparable entre sí.

### 3.4. Implementación de la prueba de Granger a las variables de tráfico e incidencias viales

La prueba de causalidad de Granger es una técnica econométrica utilizada para determinar si existe una relación de causalidad entre dos variables de series de tiempo. La idea principal detrás de esta prueba es evaluar si los valores pasados de una variable  $X$  pueden ayudar a predecir los valores futuros de otra variable  $Y$ , más allá de la información contenida en los valores pasados de  $Y$ .

La implementación de la prueba de Granger a las variables del conjunto de datos Tráfico-Incidencias implica varios pasos clave. Aquí se describen estos pasos de manera detallada:

**Paso 1 seleccionar variables:** se utilizan las 25 variables de tráfico e incidencias viales descritas en la tabla 1, exceptuando la variable jamFactor.

**Paso 2 establecer las hipótesis:** para identificar la causalidad entre dos variables mediante la prueba de Granger se establecen las siguientes hipótesis:

**Hipótesis nula ( $H_0$ ):** No hay causalidad entre las dos variables. En este caso la serie temporal de una variable no puede predecir la otra variable.

**Hipótesis alternativa ( $H_1$ ):** Existe causalidad entre las dos variables. La serie temporal de una variable puede predecir la otra variable.

Si el valor  $p$  es menor que el nivel de significancia, en este caso de 0.05, entonces se rechaza la hipótesis nula y concluimos que hay evidencia suficiente para sugerir que la serie temporal de una variable influye en la otra variable.

Si el valor  $p$  es mayor que el nivel de significancia, no se rechaza la hipótesis nula y no hay suficiente evidencia para afirmar que una variable causa la otra.

**Paso 3 obtener el valor p entre las variables con la prueba de causalidad de Granger:** para calcular el valor p entre las variables descritas en la tabla 1 y el factor de atasco (jamFactor), donde las variables de la tabla 1 serán referidas como X y el jamFactor como Y, se emplea la prueba de causalidad de Granger. Inicialmente, se ajusta un modelo de regresión lineal que incorpora los retrasos del par de variables. Luego, se lleva a cabo una prueba de hipótesis para evaluar la significancia de los coeficientes relacionados con los retrasos de una variable en la predicción de la otra. Se aplica la ecuación 5 para calcular el p-valor en la prueba de Granger:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \gamma_1 X_{t-1} + \varepsilon_t, \quad (5)$$

dónde:

$Y_t$  Es el registro actual de la serie temporal Y,  $X_t$  Es el registro actual de la serie temporal X,  $Y_{t-1}$  Es el registro pasado de la serie temporal Y (retraso 1),  $X_{t-1}$  Es el registro pasado de la serie temporal X (retraso 1),  $\alpha$  Es el intercepto,  $\beta_1$  Es el coeficiente asociado al retraso de la serie temporal Y,  $\gamma_1$  Es el coeficiente asociado al retraso de la serie temporal X y  $\varepsilon_t$  Es el término de error.

Se utilizó la clase grangercausalitytests del paquete estadístico statsmodels en Python para llevar a cabo la prueba de causalidad de Granger y determinar el valor de p entre las variables mencionadas en la tabla 1 con la variable jamFactor, la cual representa la congestión de tráfico. La prueba de Granger se implementó con valor de retraso de 3. Se estableció un nivel de significancia de 0.05 para identificar las variables con un valor de p significativamente menor a 0.05. El conjunto de datos "Tráfico-Incidencias" se dividió en dos subconjuntos (TI1 para los meses de septiembre y octubre, y TI2 para noviembre y diciembre), con el fin de realizar dos pruebas separadas y determinar si los resultados muestran consistentemente las mismas variables de mayor impacto. Aquellas variables que cumplen con este criterio se detallan en la tabla 2 como las variables que poseen una influencia estadísticamente significativa en la congestión de tráfico representada por jamFactor.

Las variables presentadas en la tabla 2 muestran valores p inferiores al umbral de significancia de 0.05, en las dos pruebas realizadas, se observan fluctuaciones en los valores de "p" a pesar de que las variables analizadas se mantienen constantes, lo que indica una relación causal directa con la variable jamFactor, de acuerdo a la prueba de causalidad de Granger. Sin embargo, las variables ausentes en la tabla 2 están vinculadas con otras variables que no tienen una conexión directa con jamFactor, pero están interrelacionadas con otras variables, como se muestra en la figura 2. Estas variables ejercen un impacto indirecto en la congestión del tráfico (jamFactor). Por otro lado, las variables no representadas en la figura 2, se consideran independientes dado que no presentan ningún tipo de relación con otras variables.

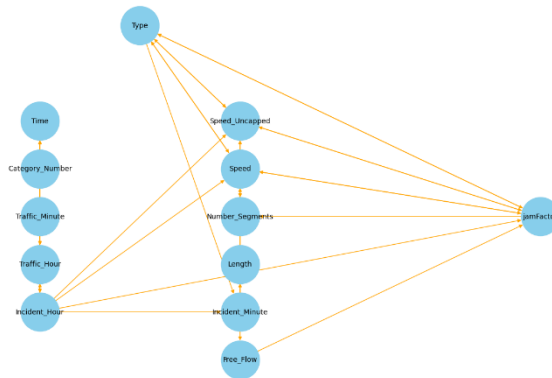
Las variables que inciden en jamFactor muestran una relación causal con la congestión de tráfico, como se detalla en la tabla 2. La similitud en las series de tiempo entre Number\_Segments y jamFactor, así como entre Speed y jamFactor se muestran en la figura 3 y 4 respectivamente.

La aplicación de la prueba de Granger ha revelado que las variables **Type**, **Speed\_Uncapped**, **Speed**, **Number\_Segments** y **Free\_Flow** son las más significativas en la causalidad de la congestión de tráfico, dentro del conjunto de 25 variables consideradas y detalladas en la tabla 1. Este resultado indica que estas variables tienen



**Tabla 2.** Variables con valores de p inferiores al nivel de significancia de 0.05.

X	Y	Valor F de T11	Valor P de T11	Valor F de T12	Valor P de T11
Type	jamFactor	4.5	0.03	3.9	0.045
Speed_Uncapped	jamFactor	9.6	0.01	4.7	0.029
Speed	jamFactor	8.9	0.002	10.5	0.001
Number_Segments	jamFactor	6.1	0.01	18.1	0.002
Free_Flow	jamFactor	6.1	0.01	7.6	0.005



**Fig. 2.** Causalidad entre todas las variables, donde cada variable muestra una relación de causalidad con una o más variables.

un impacto estadísticamente significativo en la predicción de la congestión vehicular, lo que sugiere su relevancia en el análisis y la gestión del tráfico.

### 3.5. Validación de las variables causales de congestión de tráfico identificadas mediante la prueba de Granger

La validación de las variables causales de congestión de tráfico identificadas mediante la prueba de Granger en la sección 3.4 consiste en la implementación del algoritmo Random Forest para evaluar la importancia de las variables identificadas en la predicción de la congestión de tráfico y para validar su causalidad. Se ha seleccionado el algoritmo de Random Forest debido a que es robusto frente al sobreajuste, lo que lo hace adecuado para evitar problemas comunes al trabajar con conjuntos de datos grandes, como es el caso de la congestión vehicular. Además, Random Forest proporciona una medida de importancia para cada variable, lo que facilita la identificación de aquellas que tienen un mayor impacto en la predicción de la congestión del tráfico. Su capacidad para manejar tanto variables categóricas como numéricas sin requerir transformaciones adicionales lo hace particularmente útil en este escenario heterogéneo. A continuación se detallan los pasos:

*Paso 1, establecer el conjunto de datos:* se eligen aleatoriamente 8,983 registros de los datos recopilados de tráfico e incidentes viales correspondientes al mes de enero de 2024, previamente equilibrados mediante la técnica de SMOTE (Técnica de Sobremuestreo Sintético para la Minimización de la Clase Minoritaria). Estos registros contienen información sobre cinco variables específicas: Type, Speed\_Uncapped, Speed, Number\_Segments y Free\_Flow.

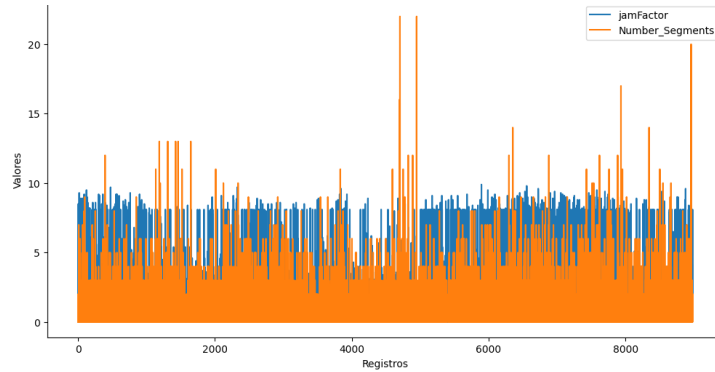


Fig. 3. Causalidad entre las variable Number\_Segments con respecto a jamFactor.

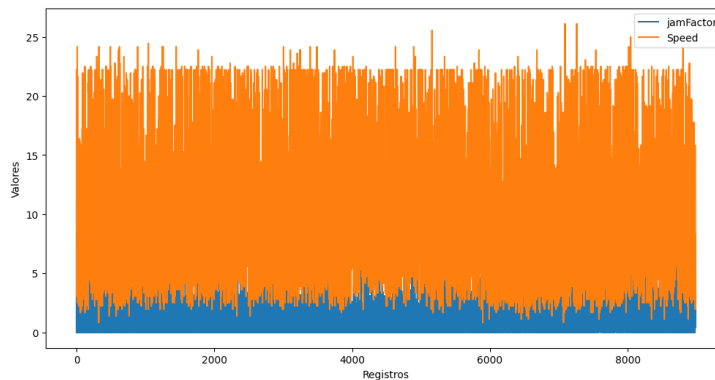


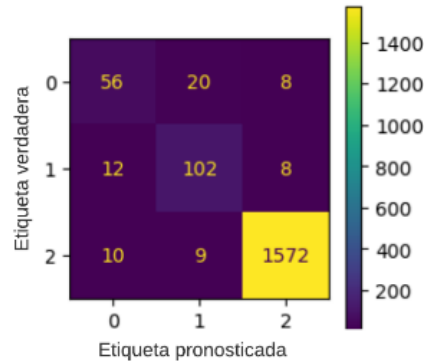
Fig. 4. Causalidad entre las variable speed con respecto a jamFactor.

*Paso 2, establecer la variable objetivo:* la variable objetivo en este trabajo de investigación es la congestión de tráfico representada por la variable jamFactor.

*Paso 3, dividir los datos en conjuntos de entrenamiento y prueba:* de acuerdo a la literatura para el entrenamiento de modelos predictivos se determina el 80% (7186 registros) de los datos para el entrenamiento y el 20% (1797 registros) para la prueba.

*Paso 4, entrenar el modelo:* para entrenar el modelo, importamos la clase RandomForestClassifier de la biblioteca sklearn en Python, específicamente para realizar una tarea de clasificación. Establecemos los valores de los hiperparámetros mediante el uso del algoritmo de Random Search, y estos valores quedaron definidos de la siguiente manera: n\_estimators se fija en 100 para definir el número de árboles en el bosque; max\_depth se establece en None para permitir que los árboles crezcan sin restricciones de profundidad máxima; min\_samples\_split se define como 10, estableciendo el número mínimo de muestras requeridas para dividir un nodo interno; min\_samples\_leaf se fija en 5 para determinar el número mínimo de muestras permitidas en una hoja; y bootstrap se establece en verdadero para permitir el muestreo con reemplazo durante la construcción de los árboles.

Una vez entrenado el modelo se muestra la matriz de confusión en la figura 5 que permite visualizar el desempeño del algoritmo de clasificación entrenado comparando las predicciones del modelo con los valores verdaderos.



**Fig. 5.** Matriz de confusión del algoritmo Random Forest con las variables más relevantes.

La matriz de confusión de la figura 5 proporciona una descripción detallada del desempeño del modelo de clasificación de congestión de tráfico, en este caso, para tres clases: "Amarillo", "Rojo" y "Verde". A continuación se explican los resultados mostrados en la matriz de confusión:

Verdaderos Positivos (TP): para la clase "Amarillo", hay 56 casos en los que el modelo predijo correctamente "Amarillo". Para la clase "Rojo", hay 102 casos en los que el modelo predijo correctamente "Rojo". Para la clase "Verde", hay 1572 casos en los que el modelo predijo correctamente "Verde".

En el caso de los Falsos Positivos (FP): Para la clase "Amarillo", hay 20 casos en los que el modelo predijo incorrectamente "Amarillo" cuando en realidad era "Rojo" (12 casos) o "Verde" (8 casos). Para la clase "Rojo", hay 12 casos en los que el modelo predijo incorrectamente "Rojo" cuando en realidad era "Amarillo" (8 casos) o "Verde" (4 casos). Para la clase "Verde", hay 9 casos en los que el modelo predijo incorrectamente "Verde" cuando en realidad era "Amarillo" (8 casos) o "Rojo" (1 caso).

La exactitud global del modelo en el conjunto de prueba es del 96.27%. Esto indica la proporción de predicciones correctas en general.

## 4. Resultados

Los resultados de la prueba de Granger indicaron que cinco variables específicas, a saber, Type, Speed\_Uncapped, Speed, Number\_Segments y Free\_Flow, mostraron una causalidad significativa con respecto al jamFactor, que se utiliza como variable para medir la congestión de tráfico.

Esto implica que el tipo de incidente (Type) que ocurra en las calles incide directamente en el surgimiento de congestiones de tráfico. Por lo tanto, en las vías, cuando se registra algún percance, la congestión resultante es dinámica, dependiendo del tipo de incidencia.

La velocidad esperada (Speed) en las calles es alta, es decir, cuando el flujo de vehículos es fluido y pueden desplazarse a una velocidad adecuada, la congestión tiende a ser mínima. Sin embargo, cuando esta velocidad disminuye debido a factores como

el aumento del volumen de tráfico y otras variables relacionadas la velocidad esperada es baja.

La velocidad sin restricciones (*Speed\_Uncapped*) describe la velocidad constante que se pueden visualizar de las calles y que ayudan a minimizar los puntos de congestión y mantener la eficiencia de la movilidad vehicular.

El número de segmentos en la calle (*Number\_Segments*) se vuelve significativo, cuando la calle tiene muchas intersecciones y aumenta las probabilidades de que los vehículos deban detenerse o reducir su velocidad para ceder el paso a otros vehículos que ingresan desde diferentes direcciones, lo que contribuye a la congestión.

Además, el flujo libre (*Free\_Flow*) de la calle, que representa la capacidad de los vehículos para moverse sin obstáculos ni restricciones, también juega un papel crucial en la congestión de tráfico. Estas relaciones descritas anteriormente son estadísticamente significativas, según los valores de *p* obtenidos en el análisis y que impactan en la congestión de tráfico.

Por otro lado, las otras variables del conjunto de datos no mostraron una causalidad significativa con el *jamFactor*. Sin embargo, es importante destacar que estas variables se encuentran relacionadas con otras variables que afectan indirectamente la congestión de tráfico, aunque no sean directamente causales con el *jamFactor*.

## **5. Conclusiones y discusión**

La validación de las 5 variables más significativas mediante un algoritmo de clasificación *Random Forest* demuestra la capacidad predictiva de las cinco variables identificadas. El alto nivel de precisión obtenido en la clasificación de diferentes niveles de congestión respalda la relevancia de estas variables en la predicción de la congestión de tráfico.

Identificar las variables clave que influyen en la congestión de tráfico puede permitir a las autoridades de transporte diseñar estrategias más efectivas para mitigar la congestión y mejorar la movilidad en las ciudades.

La selección de variables específicas mediante el uso de la aplicación de mapas (*Here Maps*) propuestas en este trabajo de investigación y el enfoque en un método estadístico particular pueden ser reforzadas con otras técnicas de análisis de datos. Por ejemplo, el estudio podría beneficiarse de la aplicación de técnicas como el *Análisis de Componentes Principales*, el *Análisis Factorial*, la *Correlación de Spearman*, *Modelos de Ecuaciones Estructurales*, entre otros.

Estas técnicas podrían ayudar a explorar más a fondo las relaciones entre las variables y la congestión de tráfico, identificando posibles interacciones complejas o relaciones no lineales que podrían haber sido pasadas por alto en el análisis inicial. Además, la incorporación de variables adicionales que podrían influir en la congestión de tráfico, como datos de cultura de manejo, dimensiones de la infraestructura vial, nivel de estrés de los usuarios, variables socioeconómicas entre otros, podrían proporcionar una comprensión más completa de las variables que contribuyen a la congestión de tráfico.

## Referencias

1. Liu, Y., Liu, C., Zheng, Z.: Traffic congestion and duration prediction model based on regression analysis and survival analysis. *Open Journal of Business and Management*, vol. 08, no. 02, pp. 943–959 (2020). DOI: 10.4236/ojbm.2020.82059.
2. Tamir, T.S., Xiong, G., Li, Z., Tao, H., Shen, Z., Hu, B., Menkir, H.M.: Traffic congestion prediction using decision tree, logistic regression and neural networks. *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 512–517 (2020). DOI: 10.1016/j.ifacol.2021.04.138.
3. Fahs, W., Chbib, F., Rammal, A., Khatoun, R., Attar, A.E., Zaytoun, I., Hachem, J.: Traffic congestion prediction based on multivariate modelling and neural networks regressions. *Procedia Computer Science*, vol. 220, pp. 202–209 (2023). DOI: 10.1016/j.procs.2023.03.028.
4. Nuli, S., Vikranth, N., Gupta, K.A.: Real-time traffic prediction using neural networks. In: *IOP Conference Series. Earth and Environmental Science*, vol. 1086, no. 1, pp. 012029 (2022). DOI: 10.1088/1755-1315/1086/1/012029.
5. Nuli, S., Vikranth, N., Gupta, K.A.: Real-time traffic prediction using neural networks. In: *IOP Conference Series. Earth and Environmental Science*, vol. 1086, no. 1, pp. 012029 (2022). DOI: 10.1088/1755-1315/1086/1/012029.
6. Iyer, P.R., Iyer, S.R., Ramesh, R., Subramanya, K.N.: Adaptive real time traffic prediction using deep neural networks. In: *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 2, pp.107 (2019). DOI: 10.11591/ijai.v8.i2.pp107-119.
7. Rajalakshmi-Vaidyanathan, G.: Hybrid time-series forecasting models for traffic flow prediction. *PROMET - Traffic and Transportation*, vol. 34 no. 4, pp. 537–549 (2022). DOI: 10.7307/ptt.v34i4.3998.
8. Dadashova, B., Li, X., Turner, S., Koeneman, P.: Multivariate time series analysis of traffic congestion measures in urban areas as they relate to socioeconomic indicators. *Socio-Economic Planning Sciences*, vol. 75, no. 100877 (2021). DOI: 10.1016/j.seps.2020.100877.
9. Van der-Bijl, B., Gijsbertsen, B., van Loon, S., Reurich, Y., de Valk, T., Koch, T., Dugundji, E.: A comparison of approaches for the time series forecasting of motorway traffic flow rate at hourly and daily aggregation levels. *Procedia Computer Science*, vol. 201, pp. 213–222 (2022). DOI: 10.1016/j.procs.2022.03.030.
10. Wu, Y., Lin, Y., Hu, R., Wang, Z., Zhao, B., Yao, Z.: Modeling and simulation of traffic congestion for mixed traffic flow with connected automated vehicles: A cell transmission model approach. *Journal of Advanced Transportation*, vol. 2022, pp. 1–20 (2022). DOI: 10.1155/2022/8348726.
11. Hu, W., Wang, H., Qiu, Z., Yan, L., Nie, C., Du, B.: An urban traffic simulation model for traffic congestion predicting and avoiding. *Neural Computing and Applications*, vol. 30, no. 6, pp. 1769–1781 (2018). DOI: 10.1007/s00521-016-2785-7.
12. Dorokhin, S., Artemov, A., Likhachev, D., Novikov, A., Starkov, E.: Traffic simulation: an analytical review. In: *IOP Conference Series. Materials Science and Engineering*, vol. 91, no. 1, pp. 012058 (2020). DOI: 10.1088/1757-899x/918/1/012058.
13. Kardani-Yazd, N., Kardani-Yazd, N., Mansouri-Daneshvar, M.R.: A rapid method for evaluating the variables affecting traffic flow in a touristic road, Iran. *Environmental Systems Research*, vol. 8, no. 34 (2019). DOI: 10.1186/s40068-019-0162-0.
14. Afrin, T., Yodo, N.: A survey of road traffic congestion measures towards a sustainable and resilient transportation system. *Sustainability*, vol. 12, no. 11, pp. 4660 (2020). DOI: 10.3390/su12114660.

15. Jia, X.: Analysis on influencing factors of traffic congestion in Chongqing and study on countermeasures: Empirical analysis based on principal component analysis. Atlantis Press International BV, pp. 814–822 (2023). DOI: 10.2991/978-94-6463-200-2\_84.
16. Iro, S., Pat-Mbano, E.C.: Causes of traffic congestion: A study of owerri municipal area of IMO state. American Journal of Environmental Sciences, vol. 18, no. 3, pp. 52–60 (2022). DOI: 10.3844/ajessp.2022.52.60.
17. Chen, L., Shi, J., Cheng, M., Zhu, H., Sun, L.: Characteristics of urban road non-recurrent traffic congestion based on floating car data. In: 4th International Conference on Electronic Information Technology and Computer Engineering, pp. 120–126 (2020). DOI: 10.1145/3443467.3443740.
18. Bian, C., Yuan, C., Kuang, W., Wu, D.: Evaluation, classification, and influential factors analysis of traffic congestion in Chinese cities using the online map data. Mathematical Problems in Engineering, pp. 1–10 (2016). DOI: 10.1155/2016/1693729.
19. Mahona, J., Mhilu, C., Kihedu, J., Bwire, H.: Factors contributing to traffic flow congestion in heterogenous traffic conditions. International Journal for Traffic and Transport Engineering, vol. 9, no. 2, pp. 238–254 (2019). DOI: 10.7708/ijtte.2019.9(2).09.
20. Yu, J., Wang, L., Gong, X.: Study on the status evaluation of urban road intersections traffic congestion base on AHP-TOPSIS modal. Procedia, Social and Behavioral Sciences, vol. 96, pp. 609–616 (2013). DOI: 10.1016/j.sbspro.2013.08.071.
21. Gullotta, G., Loret, E., Stewart, C., Sarti, F.: Traffic attractors and congestion in the urban context, the case of the city of Rome. Journal of Geographic Information System, vol. 12, no. 6, pp. 545–559 (2020). DOI: 10.4236/jgis.2020.126032.
22. Rahman, M.M., Najaf, P., Fields, M.G., Thill, J.-C.: Traffic congestion and its urban scale factors: Empirical evidence from American urban areas. International Journal of Sustainable Transportation, vol. 16, no. 5, pp. 406–421 (2022). DOI: 10.1080/15568318.2021.1885085.
23. Pi, M., Yeon, H., Son, H., Jang, Y.: Visual Cause Analytics for Traffic Congestion. In: IEEE Transactions on Visualization and Computer Graphics, vol. 27, No. 3, pp. 2186–2201 (2021). DOI: 10.1109/tvcg.2019.2940580.
24. Yue, W., Li, C., Chen, Y., Duan, P., Mao, G.: What is the root cause of congestion in urban traffic networks: Road infrastructure or signal control?. In: IEEE Transactions on Intelligent Transportation Systems: A Publication of the IEEE Intelligent Transportation Systems Council, vol. 23, no. 7, pp. 8662–8679 (2022). DOI: 10.1109/tits.2021.3085021.
25. Bellman, R., Kalaba, R.E.: Dynamic programming and modern control theory. New York: Academic Press, vol. 81 (1965)

# Clasificación binaria y multiclase de Retinopatía Diabética con características profundas de imágenes mejoradas en LAB y YCbCr

Ángel Salazar Alvarez<sup>1</sup>, Pedro Arguijo<sup>1,2,\*</sup>,  
José Antonio Hiram Vázquez López<sup>2</sup>, David Lara Alabazares<sup>1</sup>

<sup>1</sup> Tecnológico Nacional de México,  
Instituto Tecnológico Superior de Misantla,  
División de Estudios de Posgrado e Investigación,  
México

<sup>2</sup> Tecnológico Nacional de México,  
Instituto Tecnológico Superior de Misantla,  
División de Ingeniería en Sistemas Computacionales,  
México

222T0541@itsm.edu.mx, pedroarguijo16@gmail.com

**Resumen.** La Retinopatía Diabética (DR) es una complicación de la diabetes que afecta gradualmente la visión, su identificación y grado de severidad permite tomar medidas preventivas y correctivas. Se realizó la clasificación binaria y multiclase de características profundas extraídas con DenseNet-121 y Ademxapp Model A tanto en las imágenes sin procesar como de imágenes mejoradas en los espacios de color LAB y YCbCr. En la clasificación binaria se determina la presencia o ausencia de DR y en la multiclase el grado de severidad. La clasificación se realizó con Support Vector Machine (SVM), Random Forest (RF) y Gradient-Boosted Trees (GBT). Para la clasificación binaria se obtuvieron resultados de exactitud del 98% mientras que para la multiclase del 84%.

**Palabras clave:** Retinopatía diabética, características profundas, SVM, RF, GBT, DenseNet 121, ademxapp model A, LAB, YCbCr.

## Binary and Multiclass Classification of Diabetic Retinopathy with Deep Features of Enhanced Images in LAB and YCbCr

**Abstract.** Diabetic Retinopathy (DR) is a complication of diabetes that gradually affects vision, its identification and degree of severity allows preventive and corrective measures to be taken. Binary and multiclass classification of deep features extracted with DenseNet-121 and Ademxapp Model A was performed on both raw and enhanced images in LAB and YCbCr color spaces. Binary classification determines the presence or absence of DR and multiclass classification determines the degree of severity. Classification was performed with Support Vector Machine (SVM), Random Forest (RF) and Gradient-

Boosted Trees (GBT). For the binary classification, 98% accuracy results were obtained, while for the multiclass classification, 84% accuracy was obtained.

**Keywords:** Diabetic retinopathy, deep features, SVM, RF, GBT, DenseNet 121, Ademxapp Model A, LAB, YCbCr.

## 1. Introducción

La diabetes o diabetes mellitus, es una enfermedad crónica que altera el metabolismo de la glucosa en la sangre, siendo la diabetes tipo 1 y tipo 2 las más comunes. Mientras que en la primera, el cuerpo no produce suficiente insulina; como consecuencia de la destrucción de células pancreáticas, en la de tipo 2, el cuerpo se vuelve resistente a la insulina; lo que puede provocar hiperglucemia, es decir, un aumento de azúcar en la sangre. La falta de control de este padecimiento puede acarrear serias y diversas complicaciones de salud, tales como enfermedades cardíacas, daño renal, problemas de la visión y neuropatías [1].

Globalmente, 1 de cada 10 adultos de 20 a 79 años padecen algún tipo de diabetes, además, cada vez hay más países en los que 1 de cada 5, incluso más, se encuentran con la misma situación. La Federación Internacional de Diabetes (IDF, por sus siglas en inglés) reportó que en el año 2021 había 537 millones de personas con diabetes. Además, estimó que habrá 643 y 783 millones de personas con este padecimiento hacia los años 2030 y 2045, respectivamente. Así mismo, la IDF señala que la región de Canadá, México, Estados Unidos y algunos países del Caribe presentan la segunda mayor mortalidad por diabetes y el segundo porcentaje más alto de muertes de personas en edad laboral, de igual manera es la región que tiene el mayor gasto relacionado con esta enfermedad [2]. En México, la diabetes se posicionó como la segunda causa de defunciones [3], de ahí la importancia de la concientización, diagnóstico oportuno y control adecuado de esta condición.

Como se mencionó, la diabetes puede acarrear diversas complicaciones médicas en aquellas personas diagnosticadas con este padecimiento. Estas se dividen en microvasculares y macrovasculares en relación con el tamaño de los vasos sanguíneos afectados en el cuerpo. Entre las complicaciones microvasculares se incluyen la nefropatía, la neuropatía y la retinopatía; entre las complicaciones macrovasculares se encuentran las enfermedades cardiovasculares y de arteria periférica, por mencionar algunas [4].

La Retinopatía Diabética (DR por sus siglas en inglés), es una complicación ocular de la diabetes que afecta a los vasos sanguíneos de la retina debido a altos niveles de glucosa en la sangre, lo que provoca que estos se inflamen y en ocasiones se obstruyan o incluso se presenten hemorragias. En etapas prematuras puede ocasionar visión borrosa o pérdida parcial hasta llegar a los estados más avanzados donde puede ocurrir la pérdida total de la visión. Por lo tanto, es importante que las personas diagnosticadas con diabetes se sometan a estudios oftalmológicos periódicos para detectar DR en etapas tempranas, controlar su progreso y evitar el riesgo de ceguera. La DR puede ser no proliferativa (NPDR) y proliferativa (PDR), en la primera aparecen lesiones tempranas progresivas, mientras que la segunda se refiere a un escenario más avanzado y serio de la enfermedad. Estudios realizados muestran que el 34% de la población diagnosticada con diabetes presenta alguno de estos niveles de DR [5].

La detección manual de la DR presenta desafíos considerables, es un proceso que demanda un tiempo considerable, además de ser susceptible a errores de diagnóstico y



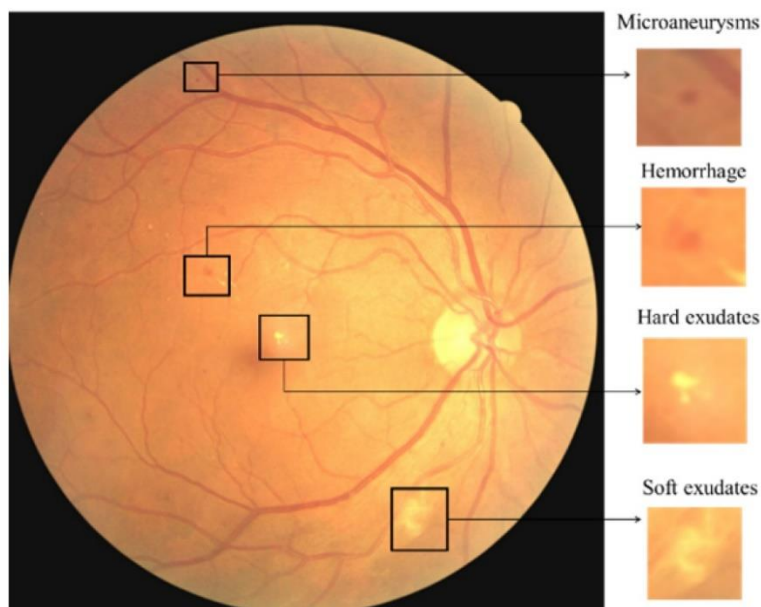


Fig. 1. Tipos de lesiones de DR [5].

Tabla 1. Clasificación de DR.

Grado de Retinopatía Diabética	Lesiones
No DR	No hay lesiones.
DR leve	Solo MA.
DR moderada	Más que MA, pero menos que DR grave.
DR grave	Cualquiera de las siguientes: <ul style="list-style-type: none"> <li>- Mas de 20 HM intrarretinales cada 4 cuadrantes.</li> <li>- Vasos sanguíneos bien definidos en más de 2 cuadrantes.</li> <li>- Anormalidades microvasculares intrarretinales en al menos 2 cuadrantes sin señales de DR proliferativa.</li> </ul>
DR proliferativa	Una o más de las siguientes: <ul style="list-style-type: none"> <li>- Neovascularización.</li> <li>- HM pre-retinales.</li> </ul>

puede resultar costoso. El diagnóstico se realiza mediante la observación a través de lentes especiales y la utilización de equipo especializado para escanear la retina y proveer imágenes detalladas que ayudan al especialista en el análisis, búsqueda y medición de las lesiones oculares (ver Fig. 1), lesiones rojas conocidas como microaneurismas (MA) y hemorragias (HM), lesiones brillantes conocidas como exudados blandos y duros (EX), además es posible observar el crecimiento anormal de nuevos vasos sanguíneos denominado como neovascularización [6].

En la Tabla 1, se muestran las etapas de la DR que depende del tipo y número de lesiones en la retina [7].

El desarrollo e implementación de sistemas de diagnóstico asistido por computadora (CAD, por sus siglas en inglés) surge como una alternativa prometedora para superar

las limitaciones mencionadas en la detección manual de la DR. En la última década, los CAD basados en aprendizaje profundo (DL, por sus siglas en inglés) han demostrado un gran potencial para la detección precisa y eficiente de la DR. El DL se caracteriza por su capacidad para aprender directamente de grandes conjuntos de datos, sin necesidad de una extracción manual de características. Esta propiedad lo convierte en una herramienta ideal para el análisis de imágenes médicas, donde la identificación de patrones complejos es crucial. En contraste, los métodos de aprendizaje automático (ML, por sus siglas en inglés) sí requieren la extracción previa de características, pero necesitan menos datos para su entrenamiento.

En este trabajo realizamos una comparación en la clasificación binaria y multiclase del grado de DR considerando características profundas extraídas con arquitecturas previamente entrenadas. En la clasificación binaria se determina la presencia o ausencia de DR en las imágenes, mientras que en la clasificación multiclase, se especifica el grado de DR presente en las imágenes. Las imágenes se mejoraron con dos técnicas, reportadas en la literatura, las cuales incrementan los detalles significativos de las imágenes. Como extractores de características profundas empleamos Adexapp Model A y DenseNet 121. La clasificación de las características profundas se realizó con Support Vector Machine (SVM), Random Forest (RF) y Gradient-Boosted Trees (GBT).

El resto del manuscrito se organiza como se menciona a continuación. En la siguiente sección se abordan trabajos relacionados con la detección de DR a través de características profundas. Los clasificadores utilizados y los extractores de características profundas se describen en la sección 3. En la sección 4 se aborda la metodología empleada junto con las métricas consideradas. Los resultados se muestran y discuten en la sección 5. Finalmente, en la sección 6 se presentan las conclusiones del desarrollo de la investigación.

## 2. Trabajos relacionados

Para la predicción del grado de DR a partir de imágenes, se han propuesto una amplia gama de algoritmos y técnicas. Entre las más destacadas se encuentran las redes neuronales convolucionales (CNN, por sus siglas en inglés), las cuales han demostrado un alto rendimiento tanto en la detección binaria de DR como en la identificación de la severidad de la enfermedad mediante clasificación multiclase; en diferentes dataset de imágenes, entre los que se encuentran APTOS 2019, MESSIDOR e IDRID, por mencionar algunos; una descripción de estos dataset y otros más se encuentra en [8].

Kassani y colaboradores propusieron una arquitectura modifica de Xception concatenando características de las capas intermedias y compararon su eficiencia contra las arquitecturas entrenadas InceptionV3, MobileNet y ResNet50; obteniendo una exactitud del 83% para la para la clasificación del dataset ICDRSS [9]. Bodapati y colaboradores proponen una red neuronal profunda con un mecanismo de atención entrenada con características de diversas CNN preentrenadas, además utilizan métodos de reducción espacial para lograr representaciones condensas de la información al clasificar obteniendo una exactitud de 82.54% en el dataset APTOS 2019 [10]. Mobeen-ur-Rehman y Khan, utilizaron CNNs preentrenadas, VGG-16, AlexNet y SqueezeNet, además de una red neuronal propia de 5 capas, 2 capas de convolución y 3 completamente conectadas, para la identificación de la presencia de DR en imágenes de retina, obteniendo exactitudes del 98% en cada escenario [11]. Wejdan y Maysoun proponen dos de aprendizaje profundo, CNN512 para la clasificación de imágenes de

DR en sus 5 etapas, y YOLOv3 para la detección y localización de las lesiones. La combinación de ambos modelos resulta en un sistema integral de clasificación y localización de las lesiones de DR con una exactitud de 89% [12].

Bilal y Sun proponen un enfoque diferente a CNNs, un enfoque híbrido para detectar las lesiones de la retina y su clasificación para identificar la etapa de la enfermedad, someten las imágenes a un preprocesamiento aprovechando el espacio de color YCbCr para mejorar la calidad de las muestras y realizar la detección de las lesiones además de extraer sus características, emplean SVM, k-vecinos más cercanos (KNN) y árboles de decisión para llevar a cabo la clasificación, finalmente someten a votación el resultado de los clasificadores obteniendo una exactitud de 98.06% [13]. Makmur y Kwan aplican y comparan dos técnicas de extracción de características de imágenes, Patrones Locales Binarios (LBP) y Matriz de Concurrencia de Niveles de Gris (GLCM), para mejorar el diagnóstico del grado de DR. Comparan el rendimiento de ambas técnicas utilizando SVM, bosque aleatorio y regresión logística y evalúan exactitud de la clasificación de DR obteniendo exactitudes entre el 73% y 74% [14].

### 3. Marco teórico

#### 3.1. Support Vector Machine

Support Vector Machine son un algoritmo propuesto en los años 90 [15] para resolver problemáticas de clasificación binaria, actualmente se utiliza ampliamente para problemas de regresión y clasificación multiclase en diversas áreas, tales como visión, artificial, reconocimiento de objetos, procesamiento de lenguaje natural, entre otras [16]. La idea principal de SVM es encontrar el separador lineal, denominado como hiperplanos, que separe las diferentes clases del conjunto de datos sobre el espacio de características. Los puntos que se encuentran más cerca del hiperplano se denominan vectores de soporte.

#### 3.2. Random Forest

Random Forest es un algoritmo de aprendizaje automático, introducido por Leo Breiman [17], utilizado principalmente para tareas de clasificación y regresión. Es un método de aprendizaje no paramétrico por conjuntos basado en árboles de decisión, combina la idea de los vecinos más cercanos adaptativos con el bagging [18]. Es un grupo de árboles de clasificación o regresión, de acuerdo a su función, que se forman a partir de la selección aleatoria de los datos del conjunto de datos de entrenamiento y cuando se trata de realizar predicciones con nuevos datos, cada árbol en el bosque emite una predicción y la clase o valor más común entre los árboles se elige como la predicción final, es decir, la predicción final depende del voto mayoritario en cuanto a clasificación o el promedio cuando su función es regresión [19].

#### 3.3. Gradient Boosted Trees

Los árboles impulsados por gradiente son una técnica de aprendizaje automático utilizada para problemas de regresión y clasificación, combina múltiples modelos más simples, normalmente árboles de decisión, para mejorar la precisión y la generalización del modelo final [20]. La idea principal de este algoritmo es entrenar árboles de decisión secuencialmente, de tal forma que cada nuevo árbol se enfoca en corregir los

errores del modelo anterior, las funciones de pérdida aplicadas pueden ser arbitrarias y depende del investigador o la tarea específica en cada situación [21].

### 3.4. Ademxapp Model A

Es una CNN presentada por investigadores de la Universidad de Adelaida en 2016, una arquitectura de redes residuales, menos profunda, que supera el rendimiento de modelos más profundos como ResNet-200 en el conjunto de entrenamiento de ImageNet [22].

Crearon 6 modelos, A, B, C, D, E y F de esta arquitectura, con diferentes tamaños de entrada, diferentes kernels y los evaluaron con el dataset de clasificación ImageNet Large Scale Visual Recognition Challenge (ILSVR) 2012 frente a otras arquitecturas, VGG16, ResNet-50, ResNet101, ResNet1-52, ResNet200, Inception demostrando mayor eficiencia desde el punto de vista espacial y mejor rendimiento no solo en clasificación del conjunto de datos sino también en la segmentación de las imágenes. ILSVR es un conjunto de imágenes que consta de 1.2 millones de imágenes de entrenamiento con 1000 clases de objetos.

### 3.5. DenseNet-121

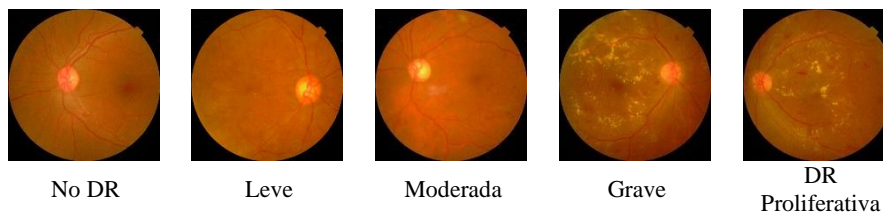
La arquitectura DenseNet ha sido utilizada en diversas áreas debido a su gran rendimiento, se ha empleado para el diagnóstico de varias enfermedades por ejemplo para la detección de Alzheimer, COVID-19, enfermedades de plantas, incluso condiciones climáticas. DenseNet proporciona arquitecturas con diferentes profundidades que pueden mejorar los resultados de los modelos de aprendizaje profundo [23].

Estas redes convolucionales profundas, conectan cada capa con todas las demás de forma feed-forward, donde cada capa utiliza como entrada los mapas de características de todas las capas anteriores, y sus propios mapas de características se utilizan como entrada en todas las capas posteriores [24].

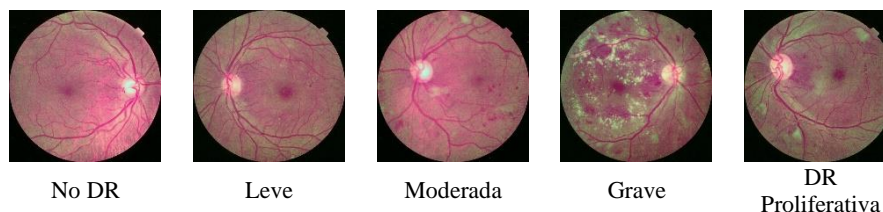
DenseNet-121 logró una exactitud de 75% en el conjunto de validación original de ILSVR 2012.

## 4. Metodología

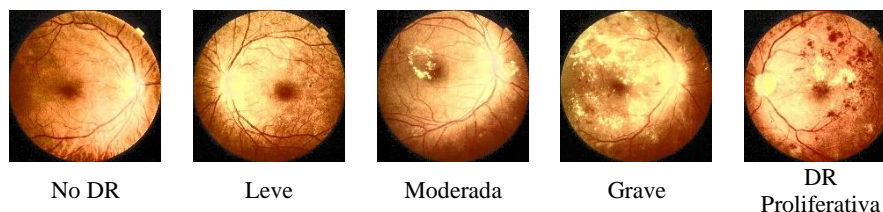
El conjunto de imágenes en este trabajo es APTOS 2019 Blindness Detection [25]. Imágenes representativas del fondo del ojo en el que se indica el grado de DR, o sea, No DR, leve, moderada, grave y DR proliferativa, se muestran en la figura 2. Similar a otros datasets de acceso público, algunas imágenes presentan diversos problemas como: bajo contraste, desenfoque, imágenes con diferente resolución espacial que van desde los 474×358 píxeles hasta 4288×2848 píxeles, entre otras. Es importante destacar que es un conjunto de imágenes desbalanceado, es decir, que las diferentes clases o muestras respectivas a cada grado de DR cuentan con cantidades diferentes de imágenes, que afectan directamente el desempeño de los modelos de clasificación, cuenta con 1805 muestras de clase No DR, 370 de clase leve, 999 de clase moderada, 193 de clase grave y 295 de clase DR proliferativa.



**Fig. 2.** Etapas DR Aptos 2019 en espacio de color RGB. Muestra representativa de cada grado de retinopatía proporcionado en el conjunto de datos [25].



**Fig. 3.** Etapas DR Aptos 2019 mejoradas en espacio de color LAB.



**Fig. 4** Etapas DR Aptos 2019 mejoradas en espacio de color YCbCr.

Con la finalidad de resaltar detalles significativos de las imágenes se mejoraron en los espacios de color LAB [26] y YCbCr [6]. Las imágenes mejoradas en LAB y YCbCr se muestran en las figuras 3 y 4, respectivamente.

Los vectores de características correspondientes a cada arquitectura se obtuvieron a partir de las capas de agregación de ambas redes, DenseNet-121 genera un vector de 1024 características profundas mientras que Ademxapp Model A genera un vector de 4096 características.

Los vectores característicos correspondientes a cada espacio de color se clasificaron en dos escenarios, el primero para evaluar la presencia o ausencia de DR y el segundo para identificar el grado de DR.

Para realizar la clasificación binaria se unieron las muestras de las clases leve, moderada, severa y proliferativa como la presencia de DR y NoDR para representar la ausencia de la enfermedad. En el caso de clasificación multiclase, se consideraron las etiquetas proporcionadas por el autor del dataset.

El desempeño de los modelos entrenados se evalúa con métricas que se obtienen de su respectiva matriz de confusión. Las métricas proporcionan información valiosa sobre el rendimiento de los modelos de aprendizaje automático al momento de realizar predicciones, existen diversas métricas y cada una aborda diferentes aspectos y contextos cuando se pretende evaluar el desempeño. Para conocer el rendimiento de los modelos obtenidos durante este estudio se consideraron tres métricas, exactitud, recall y F1 score:

$$\text{Exactitud} = \frac{TP+TN}{TP+FP+TN+FN}. \quad (1)$$

La exactitud proporciona una medida global de las predicciones correctas que realiza un modelo entre el número total de predicciones, pero cuando las clases no están completamente balanceadas conviene emplear otras métricas de evaluación que consideren la distribución entre las clases:

$$\text{Recall} = \frac{TP}{P} = \frac{TP}{TP+FN}. \quad (2)$$

Recall mide la capacidad de un modelo para predecir correctamente los ejemplos positivos entre el total de los ejemplos positivos. Proporciona una medida específica a cada una de las clases evaluadas, para realizar una medida global del rendimiento del modelo es necesario considerar un promedio ponderado de cada clase, ecuación 3:

$$\text{Recall}_{\text{GLOBAL}} = \frac{\sum_{i=1}^n \text{Recall}_i \cdot m_i}{\sum_{i=1}^n m_i}, \quad (3)$$

donde  $i$  se refiere a la clase,  $n$  es el número de clases y  $m$  indica las muestras por clase:

$$\text{F1 Score} = \frac{2TP}{2TP+FP+FN}. \quad (4)$$

F1 Score combina la precisión y recall una clase para obtener un valor más objetivo, similar a recall, es una medida específica de cada clase, para obtener un F1 Score global se consideró el promedio ponderado, ecuación 5:

$$\text{FS}_{\text{GLOBAL}} = \frac{\sum_{i=1}^n \text{F1 Score}_i \cdot m_i}{\sum_{i=1}^n m_i}, \quad (5)$$

donde *F1 Score* se refiere al F1 Score individual de una clase,  $n$  es el número de clases y  $m$  indica las muestras por clase.

## 5. Resultados

Se clasificaron los vectores de características profundas de las imágenes originales, o sea sin procesar, y de las imágenes mejoradas en los espacios de color LAB y YCbCr con RF, SVM y GBT.

### 5.1. Clasificación binaria

En las tablas 2, 3 y 4 se muestran los resultados de la clasificación binaria, o sea, la presencia o ausencia de DR tanto en las imágenes originales y en las mejoradas. Como se puede observar en las imágenes procesadas en los espacios de color LAB y YCbCr se obtienen desempeños similares en las métricas consideradas en comparación con las imágenes originales. Con los vectores de características extraídas en las imágenes procesadas en el espacio de color LAB se obtienen los mejores resultados sin importar el extractor de características profundas.

**Tabla 2.** Resultados obtenidos en la clasificación binaria en el conjunto de imágenes originales.

	Ademxapp Model A			DenseNet 121		
	ACC	FS	Re	ACC	FS	Re
<b>RF</b>	0.9631	0.9631	0.9631	0.9645	0.9645	0.9645
<b>SVM</b>	0.9781	0.9781	0.9781	0.9740	0.9740	0.9740
<b>GBT</b>	0.9727	0.9727	0.9727	0.9672	0.9672	0.9672

**Tabla 3.** Resultados obtenidos en la clasificación binaria en el conjunto de imágenes mejoradas en el espacio de color LAB.

	Ademxapp Model A			DenseNet 121		
	ACC	FS	Re	ACC	FS	Re
<b>RF</b>	0.9577	0.9577	0.9577	0.9713	0.9713	0.9713
<b>SVM</b>	0.9809	0.9809	0.9809	0.9713	0.9713	0.9713
<b>GBT</b>	0.9645	0.9645	0.9645	0.9618	0.9618	0.9618

**Tabla 4.** Resultados obtenidos en la clasificación binaria en el conjunto de imágenes mejoradas en el espacio de color YCbCr.

	Ademxapp Model A			DenseNet 121		
	ACC	FS	Re	ACC	FS	Re
<b>RF</b>	0.9658	0.9658	0.9658	0.9672	0.9672	0.9672
<b>SVM</b>	0.9768	0.9768	0.9768	0.9754	0.9754	0.9754
<b>GBT</b>	0.9699	0.9699	0.9699	0.9658	0.9658	0.9658

SVM es el algoritmo que obtiene mejores resultados para este tipo de clasificación en cualquiera de los espacios de color propuestos a través de las características extraídas por medio de Ademxapp Model A y DenseNet-121 demuestra un desempeño consistente en cualquiera de los tres algoritmos de clasificación.

Analizando las métricas obtenidas, demostramos que SVM logra una exactitud de 98% en clasificación binaria con Ademxapp Model A como extractor de características con las imágenes mejoradas a través del espacio de color LAB y 97% de exactitud en cualesquiera de los demás escenarios propuestos, ya sea en diferente espacio de color o extractor de características.

## 5.2. Clasificación multiclase

Para la clasificación multiclase, de acuerdo con los resultados mostrados en las tablas 5, 6 y 7, no se demuestra una diferencia significativa en relación con las técnicas de mejoramiento de imágenes planteadas en los espacios de color RGB y LAB, ambas técnicas muestran en general un desempeño similar respecto a la identificación de los grados de DR a través de las arquitecturas de extracción utilizadas. Se observa nuevamente que SVM ofrece el mejor modelo de clasificación siendo estos resultados similares a los reportados en [10].

Por otro lado, Ademxapp Model A proporciona 91% cuando se trata de identificar los casos positivos mediante SVM, sin embargo, RGB es más consistente en recall cuando mediante GBT y SVM.

**Tabla 5.** Resultados obtenidos en la clasificación binaria en el conjunto de imágenes originales.

	Ademxapp Model A			DenseNet 121		
	ACC	FS	Re	ACC	FS	Re
<b>RF</b>	0.7653	0.7163	0.8880	0.7967	0.7698	0.8857
<b>SVM</b>	0.8267	0.8146	0.9071	0.8226	0.8144	0.9023
<b>GBT</b>	0.7994	0.7753	0.9000	0.8158	0.8043	0.9047

**Tabla 6.** Resultados obtenidos en la clasificación binaria en el conjunto de imágenes mejoradas en el espacio de color LAB.

	Ademxapp Model A			DenseNet 121		
	ACC	FS	Re	ACC	FS	Re
<b>RF</b>	0.7735	0.7349	0.8880	0.7858	0.7549	0.8952
<b>SVM</b>	0.8431	0.8327	0.9128	0.8130	0.8015	0.8928
<b>GBT</b>	0.7912	0.7709	0.8952	0.8076	0.7932	0.8833

**Tabla 7.** Resultados obtenidos en la clasificación binaria en el conjunto de imágenes mejoradas en el espacio de color YCbCr.

	Ademxapp Model A			DenseNet 121		
	ACC	FS	Re	ACC	FS	Re
<b>RF</b>	0.7503	0.7047	0.8666	0.7721	0.7352	0.8809
<b>SVM</b>	0.8349	0.8232	0.9095	0.8076	0.7916	0.8833
<b>GBT</b>	0.7748	0.7442	0.8690	0.8062	0.7900	0.8904

## 6. Conclusiones

El diagnóstico del grado de DR es una tarea compleja debido a diversas problemáticas que enfrentan los conjuntos de datos o imágenes a utilizar para su diagnóstico, es necesario desarrollar técnicas eficientes para minimizar los efectos negativos y mejorar el desempeño y por ende los resultados de los modelos de clasificación.

Cuando se trata de identificar la presencia o ausencia de DR en el dataset propuesto, se logran resultados sobresalientes con cada uno los escenarios planteados, SVM demostró ser el mejor modelo de clasificación a través de los vectores generados por Ademxapp Model A a partir de las imágenes mejoradas en el espacio de color LAB con 98% de exactitud frente a los 97% de exactitud de DenseNet-121, siendo RF el clasificador con el desempeño más bajo obteniendo tan solo el 95% de exactitud, pero considerando el costo computacional que conlleva la extracción de las características de Ademxapp Model A y entrenamiento de SVM, DenseNet-121 puede proporcionar una mejora considerable en cuanto al tiempo en general y exactitud.

La identificación de la etapa de DR en la que se encuentran las muestras es más compleja, si bien SVM y Ademxapp Model A proporcionan las métricas más sobresalientes cuando se utiliza el espacio de color LAB, trabajar en el espacio de color RGB a través de la arquitectura DenseNet puede ofrecer prácticamente los mismos resultados con un costo computacional menor, puesto que no requiere mejorar las imágenes y la extracción de características requiere una cantidad de tiempo considerablemente menor.



Es necesario continuar experimentando con más arquitecturas, desarrollar o aplicar otras técnicas de mejoramiento de imágenes para incrementar los resultados de clasificación, así como para discriminar las muestras que no aportan información útil al momento de definir correctamente cualquiera de las clases de DR.

## Referencias

1. American Academy of Ophthalmology: <https://www.aao.org> (2024)
2. International Diabetes Federation: <https://idf.org> (2024)
3. INEGI, <https://www.inegi.org.mx> (2024)
4. Papatheodorou, K., Banach, M., Bekiari, E., Rizzo, M., Edmonds, M.: Complications of Diabetes 2017. *Journal of Diabetes Research* (2018). DOI: 10.1155/2018/3086167.
5. Fung, T.H., Patel, B., Wilmot, E.G., Amoaku, W.M.: Diabetic Retinopathy for the Non-Ophthalmologist. *Clinical Medicine*, vol. 22, no. 2, pp. 112–116 (2022). DOI: 10.7861/clinmed.2021-0792.
6. Bilal, A., Sun, G., Li, Y., Mazhar, S., Khan, A.Q.: Diabetic Retinopathy Detection and Classification using Mixed Models for a Disease Grading Database. *IEEE Access*, vol. 9, pp. 23544–23553 (2021). DOI: 10.1109/ACCESS.2021.3056186.
7. Wilkinson, C.P., Ferris III, F.L., Klein, R.E., Lee, P.P., Agardh, C.D., Davis, M.: Global Diabetic Retinopathy Project Group. Proposed International Clinical Diabetic Retinopathy and Diabetic Macular Edema Disease Severity Scales. *Ophthalmology*, vol. 110, no. 9, pp. 1677–1682 (2003). DOI: 10.1016/S0161-6420(03)00475-5.
8. Nagpal, D., Panda, S.N., Malarvel, M., Pattanaik, P.A., Khan, M.Z.: A Review of Diabetic Retinopathy: Datasets, Approaches, Evaluation Metrics and Future Trends. *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7138–7152. DOI: 10.1016/j.jksuci.2021.06.006.
9. Kassani, S.H., Kassani, P.H., Khazaeinezhad, R., Wesolowski, M.J., Schneider, K.A., Deters, R.: Diabetic Retinopathy Classification Using a Modified Xception Architecture. In: *IEEE International Symposium on Signal Processing and Information Technology*, pp. 1–6 (2019). DOI: 10.1109/ISSPIT47144.2019.9001846.
10. Bodapati, J.D., Shaik, N., Naralasetti, V.: Composite Deep Neural Network with Gated-Attention Mechanism for Diabetic Retinopathy Severity Classification. *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–15 (2021). DOI: 10.1007/s12652-020-02727-z.
11. Mobeen-Ur-Rehman, Sharzil Haris Khan, Zeeshan Abbas, Danish Rivzi, S.M.: Classification of Diabetic Retinopathy Images based on Customised CNN Architecture. In: *Amity International Conference on Artificial Intelligence, AICAI'19*, Institute of Electrical and Electronics Engineers Inc., pp. 244–248 (2019). DOI: 10.1109/AICAI.2019.8701231.
12. Alyoubi, W.-L., Abulkhair, M.F., Shalash, W.M.: Diabetic Retinopathy Fundus Image Classification and Lesions Localization System Using Deep Learning. *Sensors*, vol. 21, no. 11, pp. 3704 (2021). DOI: 10.3390/s21113704.
13. Bilal, A., Sun, G., Li, Y., Mazhar, S., Khan, A.Q.: Diabetic Retinopathy Detection and Classification using Mixed Models for a Disease Grading Database. *IEEE Access*, vol. 9, pp. 23544–23553 (2021). DOI: 10.1109/ACCESS.2021.3056186.
14. Makmur, N.M., Kwan, F., Rana, A.D., Kurniadi, F.I.: Comparing Local Binary Pattern and Gray Level Co-occurrence Matrix for Feature Extraction in Diabetic Retinopathy Classification. *Procedia Computer Science*, vol. 227, pp. 355–363 (2023). DOI: 10.1016/j.procs.2023.10.534.
15. Cortes, C., Vapnik, V.: Support-Vector Networks. *Mach Learn*, vol. 20, pp. 273–297 (1995). DOI: 10.1007/BF00994018.
16. Carmona, E.: Tutorial sobre Máquinas de Vectores Soporte (2016)

17. Breiman, L.: Random Forests. *Machine Learning*, vol. 45, pp. 5–32 (2001)
18. Breiman, L.: Bagging predictors. *Mach Learn*, vol. 24, pp. 123–140 (1996). DOI: 10.1007/BF00058655.
19. Ali, J., Khan, R., Ahmad, N., Maqsood, I.: Random Forests and Decision Trees. In: *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, pp. 272 (2012)
20. Fafalios, S., Charonyktakis, P., Tsamardinos, I.: Gradient Boosting Trees. *Gnosis Data Analysis PC*, pp. 1–3 (2020)
21. Natekin, A., Knoll, A.: Gradient Boosting Machines, a Tutorial. *Frontiers in Neurorobotics*, vol. 7, no. 21 (2013)
22. Wu, Z., Shen, C., Hengel, A.: Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *Pattern Recognition*, vol. 90, pp. 119–133 (2016). DOI: 10.1016/j.patcog.2019.01.006.
23. Albelwi, S.: Deep Architecture based on DenseNet-121 Model for Weather Image Recognition. In: *International Journal of Advanced Computer Science and Applications*, vol. 13 (2022). DOI: 10.14569/IJACSA.2022.0131065.
24. Huang, G., Liu, Z., Van Der-Maaten, L., Weinberger, K.Q.: Densely Convolutional Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708 (2017). DOI: 10.1109/CVPR.2017.243.
25. Karthik, M., Sohler, D.: APTOS 2019 Blindness Detection. *Kaggle* (2019) <https://kaggle.com/competitions/aptos2019-blindness-detection>.
26. Priyadharsini, C., Jagadeesh-Kannan, R.: Retinal Image Enhancement based on Color Dominance of Image. *Scientific Reports*, vol. 13, pp. 7172 (2023). DOI: 10.1038/s41598-023-34212-w.

## **Análisis y desarrollo de un plugin de moodle con enfoque en aprendizaje basado en retos**

Gerardo Valdés Picos, Ricardo Rafael Quintero Meza

Tecnológico Nacional de México,  
División de Estudios de Posgrado e Investigación,  
México

{gerardo.vp, ricardo.qm}@culiacan.tecnm.mx

**Resumen.** El presente artículo describe el desarrollo y la implementación de un plugin para Moodle, diseñado específicamente para fomentar el aprendizaje activo en estudiantes de analítica y ciencia de datos. El plugin se basa en el marco de Aprendizaje Basado en Retos (CBL), el cual promueve la resolución de problemas y la aplicación práctica de los conceptos teóricos. Para empezar, se proporcionan los fundamentos teóricos que gobiernan la estructura de la metodología, desde los pasos que deben seguir los participantes para aplicar el marco apropiadamente, hasta los elementos digitales que sirven para auxiliar en su composición. Se exploran herramientas existentes que utilizan metodologías similares y su eficiencia como medios alternativos y/o complementarios para educar estudiantes. Por otra parte, se analizan las ventajas y desventajas del marco CBL en comparación con otras metodologías educativas, resaltando así sus fortalezas y potencial como sistema predominante en instituciones educativas de nivel superior. Como punto principal, se explica en detalle el proceso de desarrollo del plugin, incluyendo los aspectos técnicos más relevantes, sus características funcionales, y la experiencia de usuario ideal. Finalmente, se incluyen las conclusiones de la investigación, y los resultados que se esperan como producto del trabajo futuro y la fase de pruebas con estudiantes universitarios.

**Palabras clave:** Aprendizaje basado en retos, moodle, plugin, analítica de datos, ciencia de datos

### **Analysis and Development of a Moodle Plugin Focused on Challenge-based Learning**

**Abstract.** The present article describes the development and implementation of a plugin for Moodle, specifically designed to promote active learning in analytics and data science students. The plugin is based on the Challenge-Based Learning (CBL) framework, which encourages problem-solving and the practical application of theoretical concepts. Initially, the theoretical foundations governing the methodology's structure are provided, from the steps participants must follow to apply the framework appropriately, to the digital elements that aid in its composition. Existing tools that use similar methodologies and their efficiency as alternative and/or complementary means to educate students are explored. Furthermore, the advantages and

disadvantages of the CBL framework are analyzed in comparison to other educational methodologies, thus highlighting its strengths and potential as a predominant system in higher education institutions. As the focal point, the process of plugin development is explained in detail, including the most relevant technical aspects, its functional characteristics, and the ideal user experience. Finally, the research conclusions are included, along with the expected results as a product of future work and the testing phase with university students.

**Keywords:** Challenge-based learning, moodle, plugin, data analytics, data science

## 1. Introducción

El Aprendizaje Basado en Retos (CBL) es un atractivo enfoque multidisciplinar de la enseñanza y la adquisición de conocimiento que anima a los estudiantes a aprovechar la tecnología que utilizan en su vida cotidiana para resolver problemas del mundo real. El Aprendizaje Basado en Retos es colaborativo y práctico, solicitando a estudiantes a trabajar con sus colegas, profesores, y expertos en sus comunidades y alrededor del mundo para hacer buenas preguntas, desarrollar conocimientos especializados más profundos, aceptar y resolver retos, tomar acción, y compartir sus experiencias [1].

El concepto de CBL construye su base sobre un marco efectivo de aprendizaje donde se resuelven retos del mundo real simultáneamente [2]. En este marco, el profesor asume el papel de mentor, supervisor, y facilitador del aprendizaje, renunciando así a su papel de única fuente de información para los alumnos implicados en el proyecto. Es justo decir que los estudiantes se convierten en los principales agentes de su proceso de aprendizaje [3].

El CBL surgió del proyecto conocido como “Apple Classrooms of Tomorrow-Today (ACOT<sup>2</sup>)”, un esfuerzo de colaboración con la comunidad educativa para identificar los principios de diseño esenciales para la escuela secundaria del siglo XXI centrándose en las relaciones entre estudiantes, profesores, y el currículo escolar [4].

Poner en marcha este marco aplicándolo a los programas educativos modernos ayudaría a los estudiantes a reforzar sus habilidades de resolución de problemas, su implicación en el proceso de aprendizaje, así como su compromiso social y ético, permitiendo así evaluar no sólo lo que se aprende en el programa, sino también la capacidad de aprender en sí misma [5].

Un extenso repertorio de literatura proporciona pruebas que apoyan el impacto positivo que el CBL tiene tanto en los estudiantes como en los profesores. Para los profesores, el CBL mejora la colaboración, el pensamiento crítico, la comunicación y la creatividad de los alumnos. Además, los alumnos aprenden más de lo previsto y se sienten motivados para trabajar más duro [6].

La aplicación del enfoque CBL en Analítica de Datos y Ciencia de Datos se espera que traiga consigo muchos beneficios. Estos beneficios incluyen una conexión sólida entre lo que se aprende y lo que se percibe en el mundo profesional, el desarrollo de habilidades de comunicación de alto nivel y la familiaridad con el proceso de creación

y compartición de soluciones a través de plataformas sociales dedicadas a la ciencia de datos, como Kaggle [7].

## 2. Estado del arte

En los últimos años se han producido avances en el campo de la educación utilizando tecnologías modernas de comunicación y una variedad de herramientas de aprendizaje, como la minería de datos educativos, la analítica académica o la analítica del aprendizaje [8]. Estos trabajos tecnológicos ofrecen flexibilidad y la ventaja de su uso remoto, así como el potencial de comunicación y cooperación entre los participantes en el desarrollo de un proyecto [9].

El trabajo que más se asemeja a la naturaleza de este proyecto es un plugin móvil de Moodle estructurado en torno al Marco de Aprendizaje Basado en Problemas y la Taxonomía de Bloom [10]. Al igual que CBL, este marco divide a los estudiantes en grupos y trabajan juntos para encontrar la mejor solución a un problema dado. Los autores del artículo en cuestión encontraron que la implementación de una metodología de enseñanza naciente fue un desafío para un entorno donde la educación tradicional basada en el teatro sigue siendo la norma, aunque los comentarios positivos de los estudiantes que encontraron conveniencia en fuentes de información accesibles y una comunicación fiable con otros participantes en el proyecto proporcionan evidencia de que trabajar para actualizar nuestro paradigma educativo actual con marcos más modernos y bien mantenidos es un esfuerzo que vale la pena.

## 3. Marco teórico

Según [challengebasedlearning.com](http://challengebasedlearning.com) [11], un sitio web oficial con amplia documentación sobre la estructura y el flujo de trabajo del marco CBL, este último está compuesto por tres fases interconectadas: Comprometerse, Investigar y Actuar. Lo siguiente es una compilación informativa de conceptos proporcionados por este sitio web.

### *Compromiso*

Durante la fase de Compromiso, los estudiantes pasan de una gran idea abstracta a un desafío concreto y accionable utilizando el proceso de preguntas esenciales. El objetivo es conectar personalmente con el contenido académico identificando, desarrollando y asumiendo un desafío convincente. Es en esta fase donde se exploran y deciden los elementos pivotes del proyecto: Grandes ideas, Preguntas esenciales y Desafíos.

Una **gran idea** es un término o concepto amplio que presenta múltiples posibilidades de exploración y es importante en el contexto del aprendiz y la comunidad más amplia. Ejemplos de grandes ideas incluyen Comunidad, Relaciones, Creatividad, Salud, Sostenibilidad y Democracia.

Por diseño, la gran idea genera **preguntas esenciales** que reflejan los intereses personales y las necesidades de la comunidad (por ejemplo, ¿Por qué esto es importante para mí? ¿Dónde se cruza este concepto con mi mundo? etc.). Al final del

proceso de preguntas esenciales, se identifica una pregunta esencial con significado contextual.

El **desafío** convierte la pregunta esencial en un llamado a la acción para aprender profundamente sobre el tema. Un desafío es accionable y genera entusiasmo. La fase de Compromiso concluye identificando un enunciado de desafío convincente y accionable.

### *Investigación*

Partiendo del desafío, los estudiantes desarrollan experiencias de aprendizaje contextualizadas y llevan a cabo una investigación rigurosa basada en contenido y conceptos para crear una base para soluciones accionables y sostenibles. Los elementos producidos en esta fase servirán como complemento y factor determinante para los resultados en la Fase de Actuar: Preguntas Guía, Actividades y Recursos Guía y Síntesis.

La fase de Investigación comienza con la generación de preguntas relacionadas con el desafío, conocidas como **preguntas guía**. Las preguntas desarrollan el curso de estudio necesario para desarrollar una solución informada al desafío. Las preguntas se categorizan y priorizan, creando un esquema para el viaje del aprendiz.

Los estudiantes identifican y utilizan **recursos o actividades guía** para responder a las Preguntas Guía mientras trabajan hacia soluciones innovadoras, perspicaces y realistas. Ejemplos de recursos guía incluyen contenido y cursos en línea, bases de datos, libros de texto y redes sociales. Ejemplos de actividades guía incluyen simulaciones, experimentos, proyectos, conjuntos de problemas, investigaciones y juegos.

Después de responder a las preguntas guía e identificar ideas, los estudiantes analizan los datos acumulados e identifican temas. La fase de Investigación concluye con informes y presentaciones -en forma de **síntesis**- que demuestran que los aprendices han abordado con éxito todas las preguntas guía y han desarrollado conclusiones claras, estableciendo la base para la solución mientras cumplen con los objetivos de aprendizaje.

### *Actuar*

En la Fase de Actuar, se desarrollan e implementan soluciones basadas en evidencia con una audiencia auténtica y se evalúan los resultados. Los Estudiantes combinan el deseo de hacer una diferencia con una demostración de dominio del contenido. Los elementos conseguidos en este paso serán evaluados como producto final del proceso de aprendizaje de los estudiantes: Conceptos de Solución, Desarrollo de Solución, e Implementación y Evaluación.

Después de completar la fase de Investigación, los estudiantes tienen una base concreta y definida para desarrollar **conceptos de solución**. Los conceptos de solución pueden implicar planes para una campaña informativa o educativa, proyectos de mejora escolar o comunitaria, desarrollo de productos u otras actividades. Una vez que se aprueba el concepto de solución, los aprendices desarrollan prototipos, experimentan y prueban; proceso que puede ser definido como el **desarrollo de la solución**. Este ciclo de diseño iterativo probablemente planteará nuevas preguntas guía que requieran más investigación y puede hacer que los estudiantes vuelvan a la Fase de Investigación.

Después de desarrollar sus soluciones, los estudiantes las implementan, miden los resultados, reflexionan sobre lo que funcionó y lo que no, y determinan su impacto en el desafío.

Cuando la implementación está completa, los estudiantes pueden continuar refinando la solución o desarrollar un informe de finalización y compartir su trabajo con el resto del mundo.

#### 4. Herramienta propuesta

Con el propósito de implementar el marco CBL en el área deseada y así mismo aprovechar el uso de tecnología moderna y convencional para complementar este plan, se propone la institución de un plugin para la popular plataforma de educación digital *Moodle*, la cual ofrecerá un sitio de planeación, distribución de tareas, evaluación, y repositorio de material didáctico. La estructura de esta herramienta será definida por los lineamientos establecidos por el marco CBL. Esto permitirá a educadores y estudiantes familiarizados con la metodología – y posiblemente con Moodle - asimilar el contenido del plugin de manera intuitiva.

Para permitir al usuario consumir e implementar esta herramienta en sus sesiones educativas, será necesario que el mismo cuente con permisos de administración de un servidor de Moodle existente, y subsecuentemente descargue el plugin a través del sitio de distribución oficial de la compañía, *Moodle plugins directory*. Una vez descargado, el plugin deberá ser instalado en el servidor, lo que requerirá un reinicio y actualización breve del mismo. Una vez realizada la instalación con éxito, los administradores de cursos que cuenten con los permisos correspondientes podrán insertar instancias del plugin en su curso sin restricción alguna.

El tipo de plugin que se utilizará es *Activity*, por lo que el contenido de este será visualizado por el usuario como una serie de páginas web interconectadas navegables por medio de hiperenlaces en la apariencia de menús de pestaña y botones. Cada instancia del plugin dentro del curso representará un grupo de proyecto, cuya organización quedará en manos de los administradores del curso.

El contenido de cada instancia del plugin puede ser establecido desde el momento de la creación de la instancia, es decir, durante el paso de su configuración. La modalidad de inserción del contenido podrá ser de carácter preestablecido, personalizado, o vacío.

- **Preestablecido:** Se ofrecerán plantillas de proyectos que el usuario podrá incluir en sus instancias del plugin en donde el contenido ya estará cubierto con información preexistente – el objetivo de estas plantillas proveer al usuario de materiales de referencia para sus propios proyectos personalizados.
- **Personalizado:** El usuario podrá definir todo el contenido de la instancia del plugin en el momento de su creación.
- **Vacío:** El contenido del plugin consistirá en espacios en blanco (con excepción del título del proyecto) que el administrador del curso junto a su grupo de proyecto llenará con el paso del tiempo.

Se espera que el administrador del curso dicte los términos del ritmo con el que se estará integrando nuevo contenido a la instancia del plugin a lo largo del curso

**Tabla 1.** Programa de estudios propuesto como complemento al flujo de trabajo del plugin.

Unidad	Paso	Actividades
1	Comprensión del Marco de Trabajo	<ol style="list-style-type: none"> <li>1. Explorar el Aprendizaje Basado en Retos</li> <li>2. Fundamentos de la Analítica y Ciencia de Datos</li> </ol>
2	Paso: Compromiso	<ol style="list-style-type: none"> <li>1. Formular la Gran Idea</li> <li>2. Generar la Pregunta Esencial</li> <li>3. Plantear Retos</li> </ol>
3	Paso: Investigación	<ol style="list-style-type: none"> <li>1. Plantear Preguntas Guía</li> <li>2. Responder Preguntas Guía mediante Recursos y Actividades</li> <li>3. Generar la Síntesis</li> </ol>
4	Paso: Actuar	<ol style="list-style-type: none"> <li>1. Proponer Conceptos de Solución</li> <li>2. Desarrollar la Solución Seleccionada</li> <li>3. Implementar y Evaluar la Solución</li> </ol>

escolar. Predeterminadamente los estudiantes tendrán acceso a funciones de subida y descarga de contenido; se asume que los estudiantes discutirán sus planes con el administrador del curso con anticipación, por lo que el administrador estará a cargo de la disciplina de los estudiantes con respecto a cómo se manejará el contenido dentro del plugin. Se incluirán opciones de configuración que limiten las capacidades de los estudiantes en este ámbito con tal de aliviar posibles preocupaciones en materia de disciplina y organización.

Se sugiere utilizar como referencia de trabajo una retícula similar a la mostrada en la Tabla 1. Dando continuidad a lo estipulado en la sección anterior, el marco CBL sigue la estructura del contenido desde los puntos de la Unidad 2 hasta la Unidad 4, a lo cual es lo que deberá ajustarse el contenido del plugin por igual.

La Figura 1 muestra un apartado del plugin – en este caso, el contenido del paso de Compromiso, del inglés *engage*. Se puede apreciar que todos los elementos que el paso debe abordar se muestran en la captura, con la opción de editar el contenido de estos elementos puesta a disposición del administrador del curso.

En términos generales, el plugin fungirá como una red social especializada en la educación y el trabajo en equipo. Los estudiantes podrán utilizar esta herramienta como bitácora de todas las actividades que realizarán a lo largo del ciclo escolar, y a su vez capturarán y compartirán experiencias aprendidas con el resto del grupo de proyecto, pues el crecimiento de los estudiantes como agentes de su propio aprendizaje es el criterio más importante para evaluar en el marco CBL.





Fig. 1. Proposición de vista del contenido dentro del plugin.

## 5. Conclusiones y trabajo futuro

La diseminación e influencia del Aprendizaje Basado en Retos sobre los sistemas educativos modernos es vasta y clara. Si bien la familiaridad con el marco CBL yace limitada a ciertas áreas de concentración alrededor del mundo, es cuestión de tiempo para que estas fronteras se expandan, y, por consiguiente, instituciones educativas desde nivel básico hasta nivel superior empiecen a adoptar esta metodología.

Como trabajo a futuro se harán mejoras estéticas y técnicas al plugin, implementando funciones como un foro de conversación para los miembros del grupo de proyecto, y un sistema de votos en los que participantes podrán advocar por la adición, edición, o eliminación de contenido a la plataforma.

Se esperan realizar pruebas de rendimiento del plugin con estudiantes de la asignatura de Analítica y Ciencia de Datos del Tecnológico Nacional de México, Campus Culiacán. Las evaluaciones obtenidas de dichas pruebas serán comparadas con evaluaciones producto de la impartición del curso en su modalidad regular: utilizando una metodología basada en competencias.

El resultado de esta comparación definirá la eficacia del marco de Aprendizaje Basado en Retos – tomando en consideración sesgos generados a partir de la recepción del plugin por parte de los estudiantes – como metodología educativa en un ambiente escolar que ha mantenido estáticos sus modelos de enseñanza desde los albores de este.

## Referencias

1. Nichols, M., Cator, K.: Challenge based learning white paper. Apple Inc., Cupertino, California (2008)
2. Nichols, M., Cator, K., Torres, M.: Challenge based learner user guide. Digital Promise, Redwood City, California, United States (2016)
3. Membrillo-Hernández, J., J. Ramírez-Cadena, M., Martínez-Acosta, M., Cruz-Gómez, E., Muñoz-Díaz, E., Elizalde, H.: Challenge based learning: the importance of world-leading

- companies as training partners. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 13, pp. 1103–1113 (2019)
4. *Challenge Based Learning A Classroom Guide* (2011), [https://www.apple.com/br/education/docs/CBL\\_Classroom\\_Guide\\_Jan\\_2011.pdf](https://www.apple.com/br/education/docs/CBL_Classroom_Guide_Jan_2011.pdf), Retrieved 2024/02/14
  5. Torres-Barreto, M. L., Castro-Castaño, G. P., Alvarez-Melgarejo, M.: A learning model proposal focused on challenge-based learning. *Advances in Engineering Education*, vol. 8, no. 2 (2020)
  6. Perna, S., Recke M. P., Nichols, M. H.: *Challenge based learning: A comprehensive survey of the literature*. The Challenge Institute (2023)
  7. Serrano, E., Molina, M., Manrique, D., Bajo, J.: Challenge-based learning in computational biology and data science. In: *ICTERI Workshops*, pp. 725–733 (2018)
  8. Conde, M. Á., García-Peñalvo, F. J., Fidalgo-Blanco, Á., Sein-Echaluce, M. L.: Can we apply learning analytics tools in challenge based learning contexts? In: *International Conference on Learning and Collaboration Technologies*, Springer International Publishing, pp. 242–256 (2017) doi: 10.1007/978-3-319-58515-4\_19.
  9. Portuguez Castro, M., Gómez Zermeño, M. G.: Challenge based learning: Innovative pedagogy for sustainability through e-learning in higher education. *Sustainability*, vol. 12, no. 10 (2020) doi: 10.3390/su12104063
  10. Peramunugamage, A., Usoof, H., Hapuarachchi, J.: Moodle mobile plugin for problem-based learning (PBL) in engineering education. In: *2019 IEEE Global Engineering Education Conference*, Dubai, United Arab Emirates, pp. 827–835 (2019) doi: 10.1109/EDUCON.2019.8725062
  11. The Challenge Institute. *Challenge Based Learning*, <https://www.challengebasedlearning.org>, Retrieved 2024/02/14

## Diagnóstico de fallas en generadores de potencia utilizando algoritmos de machine learning

Quetzalli Amaya Sánchez<sup>1</sup>, Alberto Alfonso Aguilar Lasserre<sup>1</sup>,  
Marco Julio del Moral Argumedo<sup>1</sup>, Gustavo Arroyo Figueroa<sup>2</sup>

<sup>1</sup> Tecnológico Nacional de México,  
Instituto Tecnológico de Orizaba,  
División de Estudios de Posgrado e Investigación,  
México

<sup>2</sup> Instituto Nacional de Electricidad y Energías Limpias (INEEL),  
México

{quetza\_amaya, albertoaal}@hotmail.com,  
marcojulioarg@gmail.com, garroyo@ineel.mx

**Resumen.** Un generador de potencia es un activo clave en la disponibilidad y confiabilidad de la red eléctrica nacional. Estos activos son muy confiables, sin embargo, debido a la operación estresante que son sometidos pueden presentar fallas recurrentes, por esta razón la detección oportuna del estado de salud es fundamental. Este trabajo presenta el desarrollo de un modelo inteligente de diagnóstico de fallas de generadores basado en algoritmos de Machine Learning (ML). Para el aprendizaje, se utilizó una base de datos de pruebas de índice de salud de generadores asociada al tipo de falla, curada por expertos en equipos eléctricos. Se utilizaron dos algoritmos de clasificación: árbol de decisión y máquina de soporte vectorial, los parámetros de algoritmos se optimizaron mediante procedimientos estándar de aprendizaje automático. Los resultados muestran que el algoritmo más adecuado para el diagnóstico de fallas es el árbol de decisión con una precisión del 87%, una precisión balanceada del 81%, un índice Kappa de 0.75, un índice de Jaccard de 0.67 y un coeficiente de correlación de Matthews de 0.77. El modelo inteligente tiene la capacidad de predecir con gran certeza la presencia de una falla y el tipo de falla con buena exactitud. El proveer información sobre el estado de salud y la falla en un generador permitirá tomar decisiones adecuadas y oportunas sobre la operación y mantenimiento de estos activos.

**Palabras clave:** Generadores de potencia, diagnóstico de fallas, aprendizaje automático, árbol de decisión, máquina de soporte vectorial.

### Fault Diagnosis in Power Generators Using Machine Learning Algorithms

**Abstract.** A power generator is a key asset in the availability and reliability of the national power grid. These assets are very reliable; however, due to the

stressful operation they are subjected to, they can experience recurrent failures. For this reason, timely detection of their health status is essential. This work presents the development of an intelligent fault diagnosis model for generators based on Machine Learning algorithms. For the learning process, a database of generator health index tests associated with the type of failure, curated by experts in electrical equipment, was used. Two classification algorithms were used: decision tree and support vector machine, and the algorithm parameters were optimized using standard machine learning procedures. The results show that the most suitable algorithm for fault diagnosis is the decision tree with an accuracy of 87%, a balanced accuracy of 81%, a Kappa index of 0.75, a Jaccard index of 0.67, and a Matthews correlation coefficient of 0.77. The intelligent model has the ability to predict with high certainty the presence of a fault and the type of fault with good accuracy. Providing information on the health status and faults in a generator will allow for appropriate and timely decisions regarding the operation and maintenance of these assets.

**Keywords:** Power generators, fault diagnosis, machine learning, decision tree, support vector machine.

## 1. Introducción

La energía eléctrica es fundamental para el desarrollo económico y la calidad de vida en la sociedad contemporánea. La energía eléctrica es producida en centrales de generación hidroeléctrica y termoeléctrica, principalmente. En México, la Comisión Federal de Electricidad (CFE) es responsable de generar el 54% de la energía del país, mientras que el restante 46% proviene de empresas privadas [1]. Un generador eléctrico es una máquina encargada de transformar la energía mecánica en energía eléctrica. La disponibilidad y confiabilidad de estos equipos es crucial para el suministro y la estabilidad del sistema eléctrico nacional (SEN) [2]. Los componentes principales de un generador experimentan un deterioro progresivo debido a las tensiones eléctricas, mecánicas, térmicas y ambientales a las que están expuestas durante su operación. Este proceso de envejecimiento se ve acelerado por un mantenimiento deficiente y por operar en condiciones severas o anormales. Cuando un generador experimenta una falla durante su operación, puede provocar inestabilidad en todo el sistema eléctrico y pérdida de suministro eléctrico.

Entre las principales fallas que puede presentar un generador están: el cortocircuito entre las espiras del devanado del estator debido al deterioro del aislamiento representando más del 25% de las fallas; y las descargas parciales, responsables del 22% de las fallas en los devanados [3]. Es de suma importancia el monitoreo y diagnóstico de fallas en un generador para garantizar la disponibilidad y confiabilidad del SEN; así como para prolongar la vida útil del activo [4]. La detección de fallas en generadores se enfoca principalmente en el análisis de descargas parciales, crucial para su fiabilidad. Este análisis cubre varias formas de descargas, como a la ranura, por delaminación, entre bobinas, superficiales/tracking y en huecos del aislamiento principal.

El análisis de descargas parciales es una técnica clave, aunque su interpretación a menudo requiere del conocimiento y experiencia de los expertos en activos eléctricos.

Simplificar esta interpretación y clasificar los generadores según su estado de salud es crucial para llevar a cabo acciones de operación y mantenimiento [5].

Dada la importancia de contar con sistemas de apoyo a la operación y mantenimiento de equipos eléctricos, se han realizado diversos trabajos para la identificación y el análisis de fallos en equipos eléctricos y generadores de potencia, así como en sus partes fundamentales, lo que permite una evaluación más precisa y eficiente de la salud del equipo [6]. Destacan la aplicación de sistemas basados en una metodología no lineal para determinar el índice de salud de generadores que utiliza la información obtenida de las pruebas que se aplican a los componentes principales del generador y diversos criterios de evaluación establecidas en referencias nacionales e internacionales.

Para cada variable y parámetro de condición se define su importancia en el estado del activo y la puntuación máxima que puede obtenerse. El cálculo del Índice de Salud del activo está basado en una formulación no-lineal de tipo exponencial [7]. Estos métodos son empíricos y están basados en la experiencia y conocimiento de los expertos en equipo eléctricos, su principal problema es confiabilidad y disponibilidad de los datos debido al manejo no adecuado de las pruebas eléctricas, se requieren un mínimo de 82 variables para obtener un valor confiable para determinar el índice de salud. La investigación y el desarrollo de sistemas de diagnóstico de fallas en generadores basada en algoritmos de Inteligencia Artificial es incipiente debido principalmente a la disponibilidad y calidad de los datos.

Melih, Veysi, Hakan y Muhamment [8] proponen una metodología novedosa para analizar las fallas en los generadores de barcos. En su enfoque, integran una versión mejorada del método de toma de decisiones multicriterio, enriquecida con Lógica Difusa (LD), lo que les permite desarrollar un mecanismo de decisión para priorizar las fallas del generador. Altaf, Al-Anbuky y GholamHosseini [9], proponen una Red Neuronal Artificial (RNA) distribuida supervisada para identificar múltiples tipos de fallas y determinar la ubicación del evento de falla en una red de motores industriales.

En este trabajo, se propone el uso de algoritmos de aprendizaje automático supervisado: árbol de decisión y máquina de soporte vectorial, para diagnosticar fallas en los generadores de centrales hidroeléctricas y termoeléctricas haciendo uso de los datos de las pruebas de descargas parciales. El objetivo es tener información precisa y oportuna de las fallas, para mejorar la operación, confiabilidad y eficiencia de los sistemas de generación de energía.

## **2. Materiales y métodos**

En esta sección se detalla los materiales y métodos empleados para el desarrollo de los modelos inteligentes para el diagnóstico de fallas de generadores de potencia mediante el uso de algoritmos de aprendizaje automático supervisado, la figura 1 muestra el diagrama de flujo de la metodología utilizada.

### **2.1. Análisis general**

En esta etapa se establecen los límites y alcances del problema tomando en cuenta el conocimiento del dominio, así como un marco de referencia para definir el espacio de soluciones a explorar. El objetivo principal de este trabajo es diagnosticar las fallas

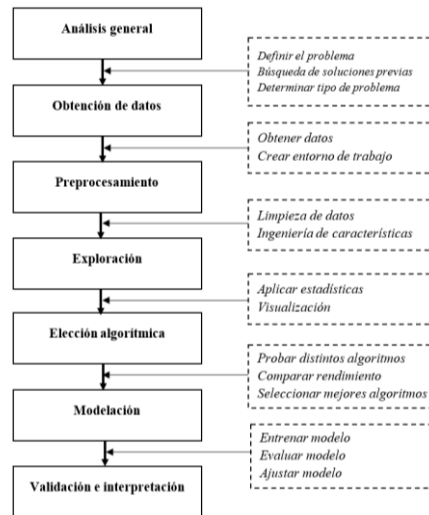


Fig. 1. Diagrama de flujo de la metodología.

en generadores de potencia. Se ha observado en la revisión de la literatura que existen enfoques previos para abordar este problema utilizando en modelos físicos basados en la experiencia de los expertos en estos activos eléctricos. Sin embargo, se opta por desarrollar un modelo inteligente debido a las ventajas de flexibilidad y robustez que ofrecen los algoritmos de aprendizaje automático. El problema asociado al diagnóstico de fallas en generadores de potencia se clasifica como un problema de clasificación multiclase, ya que implica categorizar el diagnóstico del generador en múltiples categorías de fallas.

## 2.2. Obtención de datos

En esta etapa, se comienza identificando y recopilando los datos necesarios para llevar a cabo el diagnóstico de fallas. Se cuenta con una base de datos de pruebas de generadores de potencia, compuesta por 196 filas y 20 columnas. Esta base de datos incluye una variable de salida que indica el tipo de falla y 19 atributos relacionados con pruebas de índice de salud del activo, tales como factores de disipación, monitoreo de descargas parciales, tipo de enfriamiento e inspección visual.

En la Tabla 1 se presentan las variables flotantes de la base de datos, las cuales están representadas por número reales. En la Tabla 2 se presentan las variables categóricas de la base de datos.

## 2.3. Preprocesamiento

La fase de preparación de datos implica una serie de procesos, como transformación de datos, normalización, limpieza y eliminación de ruido. En este caso específico, se realiza ingeniería de características para transformar las variables: *Enfriamiento*, *DP\_simm*, *DP\_maxVal*, *Visual* y *Diagnóstico*, las cuales originalmente se presentan de tipo entero en la base de datos, en variables categóricas. Este proceso es esencial para

**Tabla 1.** Variables categóricas de la base de datos.

Variable	Descripción	Categorías
Enfriamiento	Tipo de enfriamiento del generador	0. NA
		1. Hidroeléctrica
		2. Hidrógeno
		3. Aire
DP_simm	Simetría en semiciclos	0. Simétricos
		1. Descargas parciales en el semiciclo (+) menores al semiciclo (-)
		2. Descargas parciales en el semiciclo (+) mayores al semiciclo (-)
DP_maxVal	Valor máximo de descargas parciales	0. DP < 10
		1. 30 > DP > 10
		2. DP > 30
Visual	Inspección visual	0. Sin evidencias
		1. Ionización externa entre un bastón y el dedo de sujeción del núcleo
		2. Presencia de polvo blanco, principalmente en bobinas de salida de fase
		3. Detección de zonas con descargas externas visualmente o con cámara corona
		4. Evidencia de actividad de descargas en la zona de graduación
		5. Pintura de graduación de campo eléctrico en salida de algunos bastones del núcleo y en puentes de interconexión
		6. Evidencia de tracking
		7. Incipiente contaminación en cabezales
		8. Contaminación en cabezales
		9. Olor a ozono audible/visibles en la zona recta de las bobinas con voltaje aplicado
		10. Ionización audible/visible en la zona recta de las bobinas con voltaje aplicado
		11. Fractura de los capuchones
12. Obstrucción de ductos de ventilación		
Diagnóstico	Tipos de fallas del generador de potencia	0. Descargas a la ranura
		1. Descargas por delaminación entre el aislamiento principal y conductor de cobre
		2. Descargas entre bobinas en cabezales
		3. Descargas superficiales/tracking en cabezales debido a contaminación
		4. Descargas internas en huecos del aislamiento principal
5. Descargas por alto contenido de huecos en el aislamiento principal		

clasificar los datos según valores predefinidos asociados a una cualidad específica. Se selecciona Python como el principal lenguaje de programación para llevar a cabo el desarrollo. Para facilitar la gestión de los paquetes y entornos requeridos, se utiliza Anaconda como gestor de paquetes.

## 2.4. Exploración

La fase de exploración de datos implica un análisis detallado y una visualización exhaustiva para familiarizarse con los datos. Por ello, se lleva a cabo un análisis estadístico para obtener información relevante y detectar patrones que requieran un análisis más profundo. La representación visual de los datos mediante gráficos estadísticos es fundamental para comprender la evolución y las relaciones entre

**Tabla 2.** Variables flotantes de la base de datos.

Variable	Descripción
FD_tand_A	Factor de disipación $\tan \delta$ (0.2Vn) Ø A
FD_tand_B	Factor de disipación $\tan \delta$ (0.2Vn) Ø B
FD_tand_C	Factor de disipación $\tan \delta$ (0.2Vn) Ø C
FD_Dtand_A	Factor de disipación $\Delta \tan \delta$ (0.6Vn-0.2Vn) Ø A
FD_Dtand_B	Factor de disipación $\Delta \tan \delta$ (0.6Vn-0.2Vn) Ø B
FD_Dtand_C	Factor de disipación $\Delta \tan \delta$ (0.6Vn-0.2Vn) Ø C
DP_VnCrA_Ap	Descargas parciales (nC) $Vn/\sqrt{3}$ Ø A semiciclo (+)
DP_VnCrA_An	Descargas parciales (nC) $Vn/\sqrt{3}$ Ø A semiciclo (-)
DP_VnCrB_Bp	Descargas parciales (nC) $Vn/\sqrt{3}$ Ø B semiciclo (+)
DP_VnCrB_Bn	Descargas parciales (nC) $Vn/\sqrt{3}$ Ø B semiciclo (-)
DP_VnCrC_Cp	Descargas parciales (nC) $Vn/\sqrt{3}$ Ø C semiciclo (+)
DP_VnCrC_Cn	Descargas parciales (nC) $Vn/\sqrt{3}$ Ø C semiciclo (-)
DP_diff_A	Diferencia en la magnitud de descargas parciales en semiciclo (+) vs semiciclo (-) Ø A
DP_diff_B	Diferencia en la magnitud de descargas parciales en semiciclo (+) vs semiciclo (-) Ø B
DP_diff_C	Diferencia en la magnitud de descargas parciales en semiciclo (+) vs semiciclo (-) Ø C

variables. Por ejemplo, la distribución de frecuencia de la variable tipo de falla (diagnóstico), ver Fig. 2, muestra que se trata de una base de datos no balanceada.

### 2.5. Elección algorítmica

La elección del algoritmo depende de los objetivos del proyecto y la naturaleza de los datos. En este caso, se busca clasificar fallas en generadores de potencia, lo que implica predecir la categoría de una muestra. Se evaluaron dos algoritmos de aprendizaje supervisado de clasificación multiclase: árbol de decisión y máquina de soporte vectorial. En la Tabla 3 se presenta una descripción de cada algoritmo junto con los hiperparámetros utilizados para su sintonización. Donde el DecisionTreeClassifier se limita la profundidad máxima del árbol a 5 niveles y requiere al menos 2 muestras para dividir un nodo interno. Por otro lado, SVC utiliza un kernel de función de base radial (no lineal) para mapear los datos en un espacio dimensional superior antes de clasificarlos.

### 2.6. Modelación

Una vez seleccionados los algoritmos junto con sus hiperparámetros, se procede a su modelación. Los datos se dividen en conjuntos de entrenamiento y prueba, asignando el 80% de los datos al conjunto de entrenamiento (156 datos) y el 20% restante al conjunto de prueba (40 datos). Esta división se realiza debido a la escasez de datos disponibles.



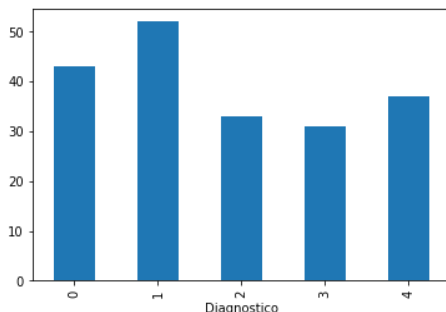


Fig. 2. Distribución de frecuencia de la variable diagnóstico.

Tabla 3. Algoritmos de clasificación de aprendizaje supervisado.

Algoritmo	Descripción	Clasificador
Árbol de decisión	Se representa como un árbol donde cada nodo interno denota una característica o atributo, cada rama indica una regla de decisión, y cada hoja representa el resultado de la clasificación	DecisionTreeClassifier(random_state=42, max_depth=5, min_samples_split=2)
Máquina de soporte vectorial	Busca encontrar el hiperplano que mejor separa los puntos de datos en diferentes clases en un espacio multidimensional	SVC(probability=True, random_state=42, kernel='rbf')

### 2.7. Validación

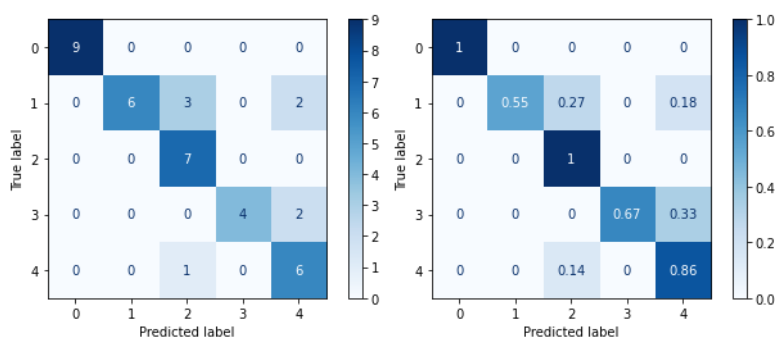
Estos modelos se evalúan utilizando métricas específicas que consideran el desequilibrio en la distribución de las clases. Entre las métricas utilizadas se incluyen: la precisión, que mide la proporción de predicciones correctas entre todas las predicciones realizadas; el recall (sensibilidad), que mide la proporción de casos positivos que fueron correctamente identificados por el modelo; el F1-score, que es la media armónica de precisión y recall; la exactitud, que mide las predicciones correctas en relación con todas las predicciones realizadas; la exactitud balanceada, que evalúa el rendimiento en situaciones de desequilibrio de clases; el índice Kappa de Cohen, que mide la concordancia entre las predicciones y las etiquetas; el índice de Jaccard, que cuantifica la similitud entre dos conjuntos de datos; el coeficiente de correlación de Matthews, que evalúa la calidad de las predicciones; la especificidad, que mide la proporción de casos negativos que fueron correctamente identificados, y la matriz de confusión, que muestra la distribución de las predicciones en comparación con las etiquetas reales.

## 3. Resultados y discusiones

En esta sección, se realiza una comparación del desempeño de los algoritmos de clasificación multiclase. Estos algoritmos son evaluados utilizando métricas específicas que consideran el desequilibrio presente en la distribución de las clases. De esta manera, se proporciona una evaluación más justa y equitativa del rendimiento de los modelos.

**Tabla 4.** Reporte de clasificación del modelo árbol de decisión.

	Clase 0	Clase 1	Clase 2	Clase 3	Clase 4
Precision	1.00	1.00	0.64	1.00	0.60
Recall	1.00	0.55	1.00	0.67	0.86
F1-score	1.00	0.71	0.78	0.80	0.71
Support	9	11	7	6	7
Accuracy	0.80				



**Fig. 3.** Matrices de confusión del modelo árbol de decisión (no normalizada y normalizada).

El análisis de métricas de evaluación en ML es un proceso fundamental para evaluar el rendimiento y la efectividad de un modelo de aprendizaje automático.

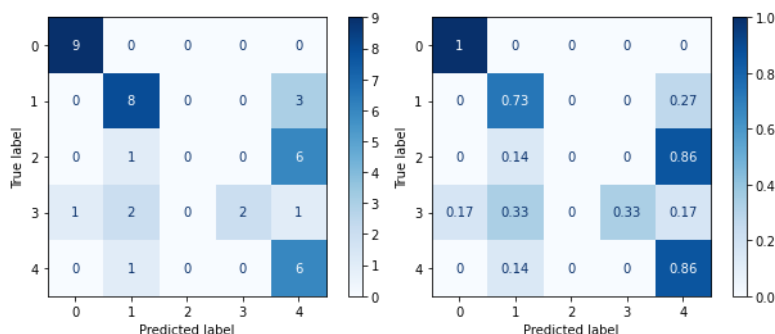
### 3.1. Evaluación del algoritmo de árbol de decisión

En la Tabla 4 se muestran los resultados del reporte de clasificación obtenidos mediante el algoritmo de árbol de decisión. Se observa para la clase 0, se logra una clasificación perfecta para la falla tipo 0. En la clase 1, todas las predicciones positivas de la falla tipo 1 son correctas, pero identifica el 55% de todos los casos reales de la falla tipo 1. En cuanto a la clase 2, el 64% de las predicciones positivas correspondientes a la falla tipo 2 son correctas e identifica todos los casos reales para esa clase. En la clase 3, todas las predicciones positivas de la falla tipo 3 son correctas, pero identifica el 67% de todos los casos reales para esa clase. Para el caso de la clase 4, el algoritmo logra una precisión del 60% en la predicción de la falla tipo 4 e identifica el 86% de todos los casos reales de dicha categoría. En general, este modelo presenta una exactitud del 0.80, indicando que el 80% de todas las predicciones hechas por el modelo son correctas.

Otras métricas para evaluar la clasificación con desequilibrio de clases son las siguientes: El algoritmo de árbol de decisión muestra una exactitud balanceada de 0.81, lo que indica un rendimiento aceptable en general. Sin embargo, el índice Kappa de Cohen muestra un valor de 0.75, sugiriendo una concordancia entre las predicciones del modelo y las etiquetas reales. El índice de Jaccard de 0.67 indica una similitud considerable entre las predicciones y las etiquetas reales. Además, un coeficiente de correlación de Matthews de 0.77 sugiere una estimación sustancial entre las predicciones y las etiquetas reales. Por otra parte, el algoritmo muestra una

**Tabla 5.** Reporte de clasificación del modelo máquina de soporte vectorial.

	Clase 0	Clase 1	Clase 2	Clase 3	Clase 4
Precision	0.90	0.67	0.00	1.00	0.38
Recall	1.00	0.73	0.00	0.33	0.86
F1-score	0.95	0.70	0.00	0.50	0.52
Support	9	11	7	6	7
Accuracy	0.62				



**Fig. 4.** Matrices de confusión del algoritmo de máquina de soporte vectorial (no normalizada y normalizada).

especificidad de 1 en la clase 0, lo que indica que es capaz de identificar correctamente las instancias de la clase negativa. La clase 1 tiene un valor de 0.54, indicando una especificidad baja para esa clase. En la clase 2 un valor de 1, indicando una especificidad alta para esa clase. Para la clase 3, el valor 0.66 indica una especificidad baja para esa clase. Y, la clase 4 tiene una regular especificidad de 0.86.

Con respecto a la métrica matriz de confusión, en la Figura 3 se muestra las matrices de confusión, tanto normalizada como no normalizada, de dicho algoritmo. Se puede observar en la matriz no normalizada que un total de 8 elementos están mal clasificados, de los cuales, tres fallas de tipo 1 es clasificado como falla de tipo 2, dos fallas de tipo 1 clasificado como falla de tipo 4, dos fallas de tipo 3 están clasificadas como falla de tipo 4 y una falla de tipo 4 clasificada como una falla de tipo 2.

### 3.2. Evaluación del algoritmo de máquina de soporte vectorial

En la Tabla 5 se muestran los resultados del reporte de clasificación obtenidos mediante el algoritmo de máquina de soporte de vectorial (SVM). Donde, en la clase 0 el 90% de las predicciones positivas de la falla tipo 0 son correctas e identifica todos los casos reales de la falla tipo 0, lo que significa un rendimiento sólido para esta clase. En la clase 1, el 67% de las predicciones positivas de la falla tipo 1 son correctas e identifica el 73% de todos los casos reales de la falla tipo 1. En la clase 2, el modelo no hace ninguna predicción correcta ni logra identificar ningún solo caso real de la falla tipo 2, lo que significa que el rendimiento es extremadamente deficiente para esa clase. En la clase 3, todas las predicciones positivas de la falla tipo 3 son correctas e identifica el 33% de todos los casos reales de la falla tipo 3. Y en la clase 4, el 38% de las

**Tabla 6.** Resumen de análisis de métricas.

	Árbol de decisión	Máquina de soporte vectorial
Exactitud	0.80	0.62
Exactitud balanceada	0.81	0.58
Precisión	0.87	0.60
Recall	0.80	0.62
F1-score	0.80	0.57
Especificidad	0.82	0.64
Índice de Kappa de Cohen	0.75	0.52
Índice de Jaccard	0.67	0.45
Coefficiente de correlación de Matthews	0.77	0.56

**Tabla 7.** Resumen de métricas por clase.

Modelo	Precisión por clase					Recall por clase					Especificidad por clase				
	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
Árbol de decisión	1.00	1.00	0.64	1.00	0.60	1.00	0.55	1.00	0.67	0.86	1.00	0.54	1.00	0.66	0.86
Máquina soporte vectorial	0.90	0.67	0.00	1.00	0.38	1.00	0.73	0.00	0.33	0.86	1.00	0.73	0.00	0.33	0.86

predicciones positivas de la falla tipo 4 son correctas e identifica el 86% de todos los casos reales de la falla tipo 4. El algoritmo SVM presenta una exactitud global del 0.62, indicando que el 62% de todas las predicciones hechas por el modelo son correctas.

Otras métricas para evaluar la clasificación con desequilibrio de clases son las siguientes: El algoritmo de máquina de soporte vectorial muestra una exactitud balanceada de 0.58, lo que indica que el modelo tiene en general un bajo rendimiento. El índice Kappa de Cohen muestra un valor de 0.52, sugiriendo una concordancia baja entre las predicciones del modelo y las etiquetas reales. El índice de Jaccard de 0.45 indica una similitud baja entre las predicciones y las etiquetas reales. Además, un coeficiente de correlación de Matthews de 0.56 sugiere una estimación moderada entre las predicciones y las etiquetas reales.

Con respecto a la métrica matriz de confusión del algoritmo de máquina de soporte vectorial (ver Fig. 4), se puede observar en la matriz no normalizada que un total de 15 elementos están mal clasificados, de los cuales, una falla tipo 3 es clasificada como falla tipo 0; una falla tipo 2, dos fallas tipo 3 y una falla tipo 4 son clasificadas como fallas tipo 1; y, tres fallas tipo 1, seis fallas tipo 2 y una falla tipo 3 son clasificadas como falla tipo 4.

### 3.3. Análisis de resultados

En la Tabla 6 se condensan los valores globales resultantes de cada algoritmo en relación con las métricas de evaluación mencionadas anteriormente. En la Tabla 7 se presenta el resumen de algunas métricas específicas para cada clase. Se observa que el modelo árbol de decisión logra el mejor rendimiento, prediciendo con mayor precisión

las clases de tipo 3 y 4, que son consideradas por los expertos como las fallas más graves en los generadores de potencia. Por esta razón, es preferible tener una menor tasa de falsos negativos en este tipo de fallas, ya que un falso negativo ocurre cuando una falla grave no es detectada por el modelo, lo cual podría llevar a no tomar las acciones preventivas necesarias, resultando en daños significativos y posibles interrupciones en el suministro de energía.

En el contexto del mantenimiento de generadores de potencia, la precisión del 87% y la especificidad del 82% destacan la efectividad del modelo de árbol de decisión en la identificación de fallas, especialmente las graves. Esta capacidad resulta crucial para prevenir daños significativos y garantizar la continuidad del suministro de energía. Con un recall del 80%, el modelo es capaz de detectar la mayoría de las fallas existentes, lo cual es fundamental para la seguridad y el rendimiento de los generadores. Un menor índice de falsos negativos es esencial, ya que la falta de detección de una falla grave podría ocasionar fallos catastróficos y costosos. Además, la exactitud balanceada del 81% y el coeficiente de correlación de Matthews del 77% demuestran que el modelo logra un equilibrio adecuado entre la clasificación de casos positivos y negativos, lo cual es crucial en situaciones donde las clases están desequilibradas, como es común en los datos de fallas de equipos.

Estos resultados indican que el modelo árbol de decisión representa una herramienta valiosa para mejorar la eficacia y eficiencia de las estrategias de mantenimiento preventivo y predictivo en los sistemas de generación de energía.

#### **4. Conclusiones**

Se presenta el desarrollo de un modelo inteligente para la detección de fallas en un generador de potencia. Se trata de una aplicación industrial de la Inteligencia Artificial donde la calidad y disponibilidad de los datos es limitada. El modelo inteligente se desarrolló utilizando una metodología estándar para la construcción de modelos basados en Machine Learning. Para su entrenamiento, se empleó una base de datos de pruebas convencionales (descargas parciales, factores de disipación, entre otros) utilizadas para determinar el estado de salud de los generadores. Esta base de datos, compuesta por 196 pruebas de generadores asociadas a distintos tipos de fallas y curada por expertos en equipos eléctricos, fue esencial para el proceso. En la etapa de procesamiento de datos, se aplicó ingeniería de características para extraer mayor información y generar un modelo con mayor precisión.

Los resultados indican que el modelo de árbol de decisión ofrece el mejor rendimiento, con una precisión del 87% en la detección de fallas y una exactitud balanceada del 81% en la clasificación del tipo de fallas. Para evaluar el rendimiento del modelo y considerar el desequilibrio en la distribución de las clases, se utilizaron diversas métricas como la exactitud balanceada, el índice de Kappa de Cohen, el índice de Jaccard y el coeficiente de correlación de Matthews. Esto es especialmente importante en este dominio, ya que se requiere predecir con mayor exactitud las fallas de tipo 3 y 4, consideradas por los expertos como las más graves. Estas fallas pueden detener la operación de un generador, lo que hace crucial mantener una menor tasa de falsos negativos.

La capacidad del modelo para identificar de manera precisa estas fallas críticas no solo mejora la seguridad y la eficiencia operativa, sino que también ayuda a prevenir paradas no planificadas y costosas en el sistema de generación de energía.

Como trabajo futuro se plantea utilizar otros algoritmos de aprendizaje automático tanto de tipo clásico, como ensamblados. También se plantea un proceso normado para generar datos de pruebas de índice de salud de generadores confiables que garanticen la disponibilidad y confiabilidad de los datos de prueba. Por último, los resultados obtenidos serán comparados por los procedimientos empíricos utilizados por los expertos en generadores, con la finalidad de mostrar la calidad de la predicción de fallas y desarrollar un sistema inteligente para la determinación del índice de salud y el diagnóstico de fallas de generadores de potencia.

## Referencias

1. Comisión Federal de Electricidad: Historia de la empresa, <https://www.cfe.mx/nuestraempresa/pages/historia.aspx> (2024)
2. CINVESTAV: Sistemas eléctricos de potencia, <https://www.cinvestav.mx/Departamentos/Sistemas-Elctricos-de-Potencia> (2024)
3. Yang, B., Li, F., Ye, J., Song, W.: Condition Monitoring and Fault Diagnosis of Generators in Power Networks. In: IEEE Power and Energy Society General Meeting, pp. 1–5 (2019). DOI: 10.1109/PESGM40551.2019.8973427.
4. CIGRE: <https://www.cigre.org.mx/> (2024)
5. Ramírez-Monguá, J.J., Dávila-Arias, J.Y.: Hacia el mantenimiento basado en condición de los dispositivos inteligentes en la industria 4.0 (2022)
6. Cerón, A.F., Echeverry, D.F., Aponte, G., Romero, A.A.: Índice de salud para transformadores de potencia inmersos en aceite mineral con voltajes entre 69kV y 230kV usando lógica difusa. *Información tecnológica*, vol. 26, no. 2, pp. 107–116 (2015). DOI: 10.4067/S0718-07642015000200013.
7. Campuzano-Martínez, I.R.: Diagnóstico de generadores eléctricos de potencia con técnica de monitoreo en línea y fuera de línea. *Información tecnológica*, vol. 27, no. 2, pp. 11–20 (2016). DOI: 10.4067/S0718-07642016000200003.
8. Yucesan, M., Başhan, V., Demirel, H., Gul, M.: An Interval Type-2 Fuzzy Enhanced best–Worst Method for the Evaluation of Ship Diesel Generator Failures. *Engineering Failure Analysis*, vol. 138, pp. 106428 (2022). DOI: 10.1016/j.engfailanal.2022.106428.
9. Altaf, S., Al-Anbuky, A., Gholam-Hosseini, H.: Fault Diagnosis in a Distributed Motor Network using Artificial Neural Network. *International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, pp. 190–197 (2014). DOI: 10.1109/SPEEDAM.2014.6871946.

## Desarrollo de un modelo inteligente para predecir el precio marginal local (PML) en un nodo de la red eléctrica nacional

Marcos Fidel Guzmán Escobar<sup>1</sup>, Alberto Alfonso Aguilar Lasserre<sup>1</sup>,  
Marco Julio Argumedo del Moral<sup>1</sup>, Nicasio Hernández Flores<sup>2</sup>,  
Gustavo Arroyo Figueroa<sup>2</sup>

<sup>1</sup> Tecnológico Nacional de México, Instituto Tecnológico de Orizaba,  
División de Estudios de Posgrado e Investigación,  
México

<sup>2</sup> Instituto Nacional de Electricidad y Energías Limpias,  
México

{fidel.mfge, marcojulioarg}@gmail.com,  
albertoaal@hotmail.com,  
{nicasio.hernandez, garroyo}@ineel.mx

**Resumen.** Esta investigación se centró en analizar 28 nodos representativos de los tres sistemas de interconexión de México para desarrollar pronósticos precisos del Precio Marginal Local (PML). Se utilizaron cinco modelos: Árboles de Decisión para Regresión, Regresión Polinomial de Segundo Grado, SARIMA, LSTM y una combinación de Prophet con LSTM Secuencial. Los resultados mostraron errores porcentuales medios absolutos (MAPE) en un rango de 42% a 23%. Además, se incluyeron métricas como el Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE) para evaluar la eficacia de los algoritmos en el pronóstico del PML. La metodología desarrollada, combina la integración y análisis de datos climáticos y el uso de algoritmos avanzados de Machine Learning (ML), para erigir un modelo replicable y robusto. Esta investigación proporciona un referente valioso para futuras investigaciones y aplicaciones prácticas en el ámbito del pronóstico energético en México. El proyecto culmina con el desarrollo de una Interfaz de Usuario Gráfica (GUI) interactiva, llamada “PML Explorer”, que permite a los usuarios visualizar de manera intuitiva los datos del Precio Marginal Local (PML). Además, incorpora una funcionalidad clave: la capacidad de ejecutar un código desarrollado en Python que integra cinco algoritmos avanzados. Esta herramienta está diseñada para predecir el PML hasta 48 horas en el futuro, mejorando significativamente la precisión de los pronósticos y facilitando la toma de decisiones basada en datos concretos y análisis predictivo.

**Palabras clave:** Machine learning, precio marginal local, árboles de decisión para regresión, LSTM, regresión polinomial, SARIMA, prophet, interfaz de usuario gráfica (GUI).

## Development of an Intelligent Model to Predict the Local Marginal Price (LMP) at a Node of the National Electric Grid

**Abstract.** This research focused on analyzing 28 representative nodes of the three interconnection systems in Mexico to develop accurate forecasts of the Locational Marginal Price (LMP). Five models were used: Regression Trees, Second Degree Polynomial Regression, SARIMA, LSTM, and a combination of Prophet with Sequential LSTM. The results showed mean absolute percentage errors (MAPE) ranging from 42% to 23%. Additionally, metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) were included to evaluate the effectiveness of the algorithms in forecasting LMP. The developed methodology combines the integration and analysis of climatic data and the use of advanced Machine Learning (ML) algorithms to build a replicable and robust model. This research provides a valuable reference for future research and practical applications in the field of energy forecasting in Mexico. The project culminates with the development of an interactive Graphical User Interface (GUI) called "PML Explorer," allowing users to intuitively visualize Locational Marginal Price (LMP) data. It also incorporates a key functionality: the ability to execute code developed in Python that integrates five advanced algorithms. This tool is designed to predict LMP up to 48 hours into the future, significantly improving forecast accuracy and facilitating decision-making based on concrete data and predictive analysis.

**Keywords:** Machine learning, local marginal price, decision trees for regression, LSTM, polynomial regression, SARIMA, prophet, graphical user interface (GUI).

### 1. Introducción

El Precio Marginal Local (PML) refleja el valor de la energía en un tiempo y lugar determinado. Este se calcula por medio de un modelo de optimización que minimiza los costos de generación mientras se satisface la demanda y se respetan las restricciones físicas de la red eléctrica (Alonso and Gabriel, 2017). Por lo tanto, la importancia de un precio justo mejorará los indicadores de disponibilidad, confiabilidad y eficiencia que se maneja en todos los procesos de potencia que se manejan en Empresa Generadora de Electricidad, que, a su vez, afectarán de manera positiva a los clientes consumidores de electricidad (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, and Cerisola, 2006).

Se presenta un modelo inteligente de pronóstico con redes neuronales para el Precio Marginal Local (PML) considerando el clima como factor a integrar y que influye en el precio de la energía de un nodo en específico. En la realización del modelo se llevó a cabo con el Instituto Nacional de Electricidad y Energías Limpias (INEEL), el cual es un centro de investigación dedicado a dar solución a las necesidades del sector eléctrico y energético. El artículo detalla trabajos relacionados en la materia y continúa describiendo la metodología empleada, detallando las etapas seguidas para desarrollar el modelo inteligente. Se destaca la optimización del modelo para mejorar su



rendimiento, así como la validación mediante métricas que cuantifican los errores en los pronósticos. En la sección dedicada a los resultados, se presentan los hallazgos de los cinco modelos de pronóstico evaluados, resaltando el rendimiento superior del modelo LSTM secuencial. Se ofrecen métricas de errores para comprender la eficiencia del modelo de Machine Learning al implementarlo en 28 nodos de interconexión. Finalmente, el proyecto culmina con el desarrollo de una Interfaz de Usuario Gráfica (GUI) denominada "PML Explorer". Esta herramienta interactiva no solo permite visualizar de manera intuitiva los datos del Precio Marginal Local (PML), sino que también integra la capacidad de ejecutar un código desarrollado en Python, el cual incorpora cinco algoritmos avanzados de aprendizaje automático, basados en datos concretos y análisis predictivo.

## **2. Trabajos relacionados**

La predicción de series temporales se ha destacado por el uso de redes neuronales recurrentes (RNN), las cuales poseen la capacidad inherente de retener una "memoria" que las hace idóneas para la predicción de secuencias extensas. Diversas variantes de RNN han demostrado su eficacia en este ámbito, permitiendo capturar patrones complejos en los datos temporales (SatvikKhuntia, AtharHanif, QadeerAhmed, JohnLahti, and MaartenMeijer, 2022). Por ejemplo, El enfoque de aprendizaje automático bayesiano (BML) se ha aplicado para la predicción de ondas con incertidumbre cuantificada, utilizando métodos de aproximación como la deserción Monte Carlo (MC) y hardware de ML moderno (Jincheng Zhang, Xiaowei Zhao, Siya Jin, and Greaves, 2022). El uso de redes neuronales en la optimización de la producción de energía eólica y la maximización de ingresos ha sido investigado, aplicando técnicas como el algoritmo genético cuántico y la coordinación de parques eólicos con centrales eléctricas de almacenamiento de energía (Xiyun Yang, Liwei Fan, Xiangjun Li, and Meng, 2022). El pronóstico del PML ha sido abordado utilizando diversos enfoques, como redes neuronales artificiales (RNA), el método de Mínimos Cuadrados Generalizados por División de Grupo (GMDH) y sistemas de inferencia difusa basados en redes neuronales adaptativas (ANFIS); donde el día del año fue la variable con mayor impacto en los componentes de pérdida y congestión.

Lo anterior está asociado al déficit de capacidad de transmisión eléctrica, como también los precios de los combustibles impactan significativamente en el componente energético, que está relacionado con el perfil de mezcla de generación de la región peninsular [6]. En el artículo (Jeovani E. Santiago López and García, 2018), se describe y caracteriza la evolución temporal de la serie de tiempo de los PML's horarios – diarios resultantes del MECP en México, analizando simultáneamente su volatilidad a lo largo de los periodos para cada mercado de energía (MDA y MTR) y para cada sistema interconectado (SIN, BCA y BCS) analizando la serie de tiempo no es estacionaria debido a patrones asociados a la frecuencia semanal donde el análisis se hizo enfocándose a datos históricos analizados estadísticamente.

El estudio del artículo (Andrés Ramos, Gonzalo Cortés, Jesus Maria Latorre, and Cerisola, 2006) analiza la relación entre el precio marginal del mercado de electricidad y la demanda mediante el uso de conglomerados analizados matemáticamente, donde se definió que el precio de la electricidad depende fundamentalmente de la demanda y

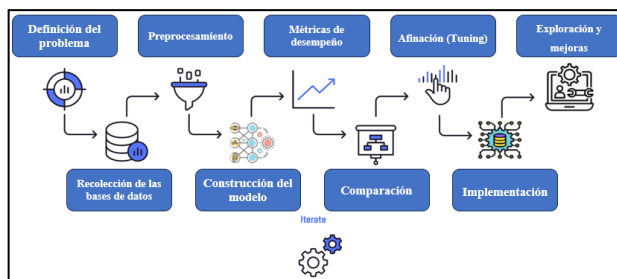


Fig. 1. Proceso de construcción de un modelo de Machine Learning.

la potencia disponible que es cambiante a lo largo del tiempo. Por otra parte, métodos como las redes neuronales de propagación inversa (BP) se han empleado para pronosticar los precios de liquidación del mercado energético diario (Deepak Singhal and Swarup, 2011). Estudios recientes han explorado funciones de pérdida para optimizar la predicción de precios al contado de la electricidad, utilizando redes neuronales con técnicas como K-vecino más cercano (KNN) y redes LSTM con aprendizaje reforzado por retro propagación (Ahmad Amine Loutfi, Mengtao Sun, Ijlal Loutfi, and Solibakke, 2022).

### 3. Metodología

A continuación, se muestra la metodología en la Fig. 1 que simplifica el proceso de construcción del modelo de ML:

El proceso a nivel general, estos son los pasos para seguir:

**Definición del problema:** El objetivo es predecir con precisión el Precio Marginal Local (PML) de nodos interconectados de la red nacional; lo que es un problema de regresión en un contexto de series temporales. Se necesita un enfoque de aprendizaje supervisado para una predicción precisa, con un horizonte de predicción de hasta 48 horas. Esto implica comprender los datos históricos y modelar sus variables y fluctuaciones.

**Recolección de las bases de datos:** Se recopilan datos de dos fuentes principales: registros horarios del PML del Centro Nacional de Control de Energía (CENACE) y variables climatológicas de la Comisión Nacional del Agua (CONAGUA), lo que plantea un desafío debido a la variabilidad en la operatividad de las estaciones climatológicas y la dificultad para obtener datos consistentes y completos.

**Preprocesamiento:** El preprocesamiento de datos es una etapa crítica para garantizar la calidad y utilidad de los datos en la modelización. Se utiliza “Python” y la biblioteca “Pandas” para preparar e identificar las variables en los datos. Esta fase incluye la unificación de archivos, la detección y manejo de valores faltantes, la normalización de datos entre 0 y 1 con “MinMaxScaler” que ayude a que todas las características contribuyan de manera equitativa al modelo sin que una característica domine debido a su escala más amplia, además de una ingeniería de características aplicando una conversión senoidal para que se tenga el patrón temporal de día y noche en los datos;

Por ejemplo, en una escala de tiempo de 24 horas, el salto de 23 a 0 crea una discontinuidad que no refleja la realidad cíclica del tiempo. Las funciones seno y coseno suavizan este salto, representando las 23:00 y 00:00 como puntos cercanos en el ciclo.

**Construcción del Modelo:** Se desarrollan una variedad de modelos de aprendizaje automático para abordar la complejidad de los datos del PML. Estos incluyen árboles de decisión, regresión polinomial, modelo en SARIMA y redes neuronales LSTM, estableciendo los subconjuntos 80% entrenamiento, 10% validación y 10% de prueba. Los pasos clave dentro del modelo LSTM + Prophet requiere:

1. **Preparación de los datos en formato seq2seq:** Los datos se organizan en pares de entrada y salida para permitir al modelo predecir secuencias futuras en lugar de valores individuales. Esta estructura de datos es fundamental para el enfoque seq2seq.
2. **Utilización de una capa LSTM seguida de una capa Time Distributed:** El modelo LSTM se emplea para capturar dependencias temporales complejas en los datos. La capa Time Distributed permite aplicar una capa densa a cada punto de tiempo en la secuencia de salida, lo que facilita la generación de predicciones secuenciales.
3. **Uso de Time Series Split para dividir los datos temporalmente:** La división temporal de los datos con Time Series Split garantiza que el modelo se entrene y valide de manera adecuada, evitando filtrar información del futuro hacia el pasado durante el proceso de entrenamiento.
4. **Preparación de Datos para Prophet:** Esto implicó indicar las columnas “ds” para las fechas y “y” para la variable objetivo (precios de energía) que son los pronósticos del modelo LSTM. Prophet requiere este formato específico para realizar sus predicciones de series temporales. Prophet está diseñado para modelar series temporales univariadas, lo que significa que solo puede manejar una variable objetivo (en este caso, 'y') y una variable de tiempo (en este caso, 'ds').
5. **Modelado y Entrenamiento con Prophet:** Se configuró y entrenó un modelo Prophet, excluyendo la estacionalidad anual (yearly\_seasonality=False) pero incluyendo la diaria (daily\_seasonality=True). Esta configuración se eligió considerando la naturaleza horaria de los datos de precios de energía.
6. **Generación de Predicciones Futuras:** Se generaron predicciones para un horizonte de 48 horas utilizando Prophet. Para esto, se creó un DataFrame “future\_lstm” que contenía las fechas futuras para las predicciones. Prophet produjo predicciones para este período, abordando las tendencias y estacionalidades identificadas.

**Métricas de desempeño:** Se evalúa el rendimiento de los modelos desarrollados utilizando métricas estándar como el Error Absoluto Medio (MAE), el Error Cuadrático Medio (MSE), la Raíz del Error Cuadrático Medio (RMSE) y el Error Porcentual Absoluto Medio (MAPE), así como técnicas de validación temporal con “timeseriesplit” y también con “walkforward-validation” para una evaluación rigurosa apropiada para series de tiempo.



Fig. 2. Localización geográfica de los nodos para la implementación del modelo.



Fig. 3. Menú principal de la interfaz gráfica de usuario "PML Explorer".

**Comparación de los modelos:** Para lograr una comparación objetiva y cuantitativa, se propone una tabla comparativa donde se presentan las métricas de rendimiento clave, incluyendo los errores MAE, MSE, RMSE y MAPE para cada modelo. Esta tabla permite no solo visualizar el rendimiento de cada modelo en términos de precisión de predicción, sino también entender sus fortalezas y limitaciones en el contexto específico del pronóstico de PML.

**Afinación (Tuning):** El objetivo es refinar el modelo LSTM para maximizar su precisión y eficiencia en la predicción de los PML.

- **Análisis de Sensibilidad en Redes LSTM:** Se llevo a cabo en redes LSTM, probando dos arquitecturas diferentes y ajustando manualmente sus hiperparámetros en 10 iteraciones, registrando los cambios en MAPE y MAE para identificar las configuraciones más efectivas.
- **Búsqueda Grid en LSTM Secuencial + PROPHET:** Para el modelo híbrido de LSTM con Prophet, se utilizó una búsqueda grid para explorar sistemáticamente una amplia gama de combinaciones de hiperparámetros y encontrar la configuración óptima para una mayor precisión predictiva.

**Implementación:** Esta fase realiza la aplicación de los cinco algoritmos en 28 nodos de interconexión en México, y se lleva a cabo un análisis utilizando los árboles de decisión para regresión y matrices de correlación.

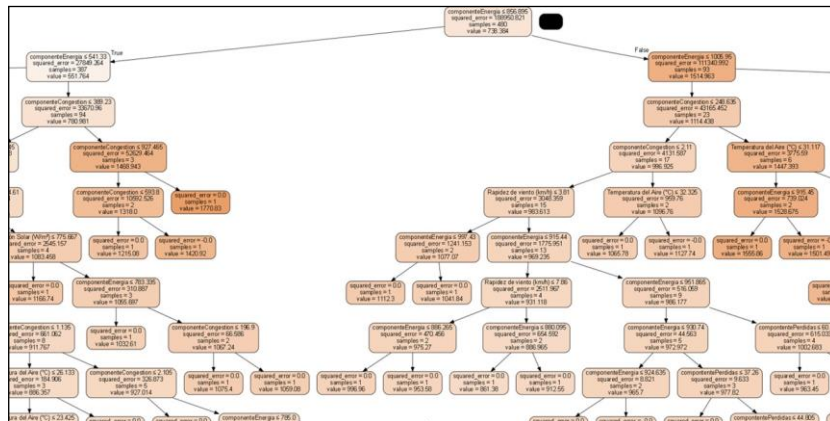
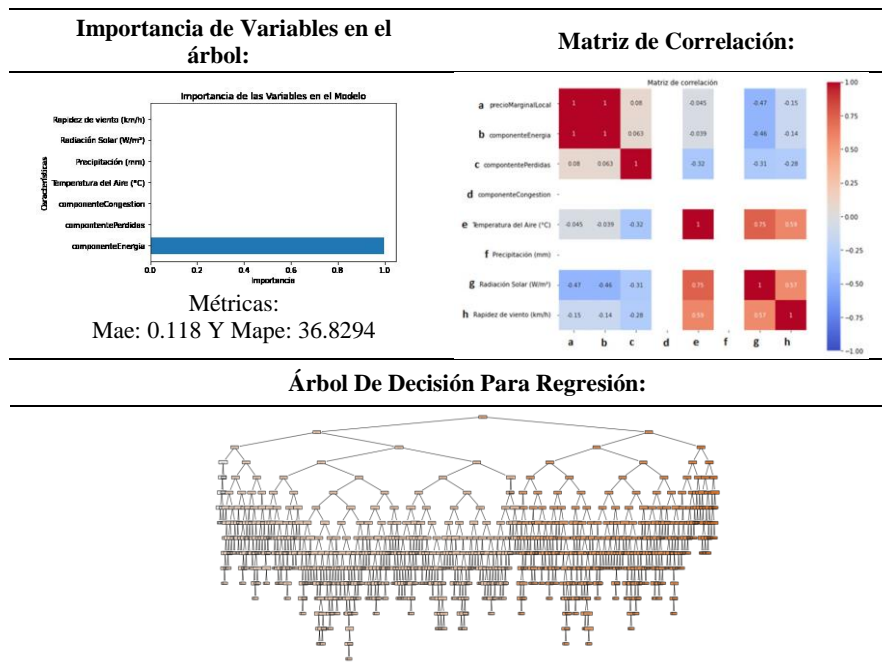


Fig. 4. Diagrama de árbol del nodo 1 de la implementación del modelo.

Tabla 1. Análisis de variables del nodo 14 (07LPZ-115), con árbol de decisión y matriz de correlación.



**Exploración y mejoras:** En esta etapa complementaria, se crea una Interfaz Gráfica de Usuario (GUI); usando “Tkinter” se propone una herramienta para visualizar los datos en una gráfica que le permite seleccionar diferentes periodos y guardarlos, si así se desea.

Una vez visualizados se regresa al menú para correr el código Python al oprimir el botón que ejecuta el código en entorno “Jupyter” notebook.

Incorporar una interfaz de usuario para que los analistas puedan ajustar manualmente la visualización gráfica y dinámica de los datos del PML es una buena herramienta.

#### 4. Resultados

Los árboles de decisión para regresión son modelos predictivos de valores numéricos por lo que el Precio Marginal Local (PML) puede ser tratado como un problema de regresión con funcionalidad de los árboles de decisión; donde cada nodo en el árbol representa una pregunta o una condición sobre las características de los datos, y cada rama representa una posible respuesta a esa pregunta.

Las hojas del árbol contienen las predicciones finales y no se limitó la altura del árbol; en la práctica, se utilizaron 720 registros (30 días de datos horarios), y la profundidad del árbol oscilaba entre 16 y 17 niveles para los diferentes nodos de la etapa de implementación. Al evaluar el modelo mediante gráficos de desempeño y validaciones cruzadas para series de tiempo, se determinó que esta profundidad proporcionaba un buen desempeño.

Esta configuración permitió al árbol capturar adecuadamente la complejidad de los datos sin necesidad de imponer una restricción explícita en su altura. Para la visualización más detallada se muestran en la Fig. y Tabla 1, y se pone a disposición un enlace a una carpeta en Google Drive para acceder al árbol en alta resolución<sup>1</sup>.

Tabla 1 muestra el nodo 07LPZ-115 de La Paz, Baja California Sur, donde el componente de energía es crucial para el pronóstico según el algoritmo de árbol de decisión y la matriz de correlación. Con una base de datos de PML de 28 nodos, el modelo LSTM secuencial puede aprender de las variables y ofrecer pronósticos confiables. Mejoramos el enfoque LSTM con una neurona "Dense" para predecir 48 horas a futuro. Aunque las métricas de prueba fueron menos eficientes que las de entrenamiento, el MSE, MAE y RMSE fueron bajos. A pesar de un MAPE elevado, el modelo generaliza bien, como se muestra en la Fig.

En la Fig. 6, se puede observar que cada división (split) se aproxima a cero, alcanzando un error cuadrático medio (MSE) muy cercano a cero antes de la época 80. Esto indica que el modelo aprendió eficazmente de la base de datos y generalizó bien los pronósticos con datos reales.

La Fig muestra una comparación entre los datos reales y los predichos por el modelo en el mismo período de tiempo. Sin embargo, la Fig. 1 va más allá al demostrar la capacidad del modelo para generalizar tanto con datos reales como para los próximos 48 PML imprevistos.

Otra mejora aplicada se logró al complementarla con Prophet; para ello es necesario que disponga de una neurona de salida llamada "*TimeDistributed*" fundamental para predecir una secuencia que pueda Prophet reconocer y lograr el ensamble de ambas herramientas útiles en series temporales. La razón además fue obtener un intervalo de confianza útil para nuestros pronósticos en LSTM fue Prophet. La proyección a futuro como se muestran en la Fig. 1 permite saber que tan bien logra generalizar los valores pronosticados y resultan ser bastante similares:

<sup>1</sup> [drive.google.com/drive/folders/1pMgxpHxrnHC8b68Ke9jTOLHo7fvLGeA?usp=drive\\_link](https://drive.google.com/drive/folders/1pMgxpHxrnHC8b68Ke9jTOLHo7fvLGeA?usp=drive_link)

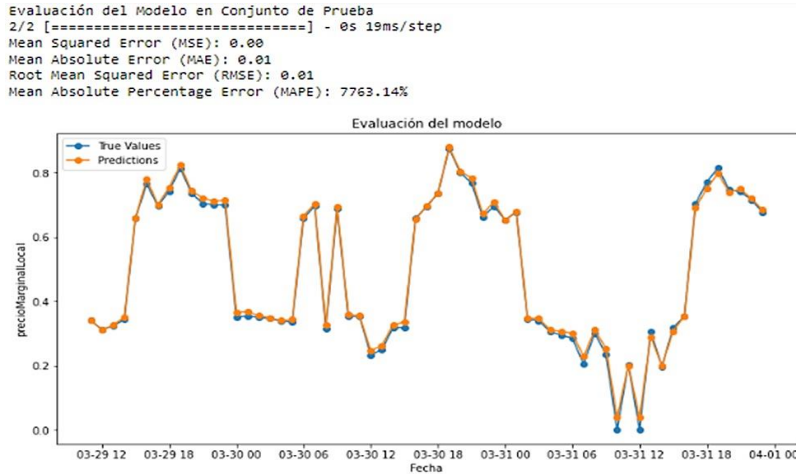


Fig. 5. Validación del modelo en el conjunto de prueba.

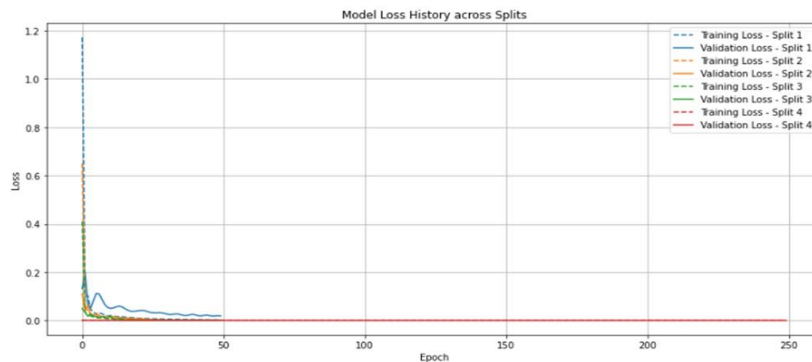


Fig. 6. Validación del modelo LSTM.

Además, se hizo una búsqueda en cuadrícula para asignar días festivos y parámetros de relevancia para la tendencia, que busquen automáticamente el pronóstico eficiente por considerar, por ejemplo, días de alta demanda entre semana, al modelo LSTM + Prophet. Dicha búsqueda abarca 3 parámetros con 4 valores, por lo tanto, el número total de combinaciones es  $4 \times 4 \times 4 = 64$  combinaciones a probar, para encontrar los mejores hiperparámetros para el modelo Prophet en función del MAPE.

Este itera a través de todas las combinaciones de parámetros y entrena un modelo Prophet para cada combinación. A continuación, los resultados del ensamble de dos poderosos algoritmos de pronóstico, véase Tabla 1 y Fig. :

Las predicciones incluyen tres columnas relevantes para los intervalos de confianza, donde se muestran las predicciones generadas por el modelo LSTM (Long Short-Term Memory) en conjunto con el algoritmo Prophet para el Precio Marginal Local (PML) y cada fila representa una predicción para una hora específica, y las columnas proporcionan la fecha correspondiente, la predicción del PML, así como los límites inferior y superior de la predicción. Véase Fig. :

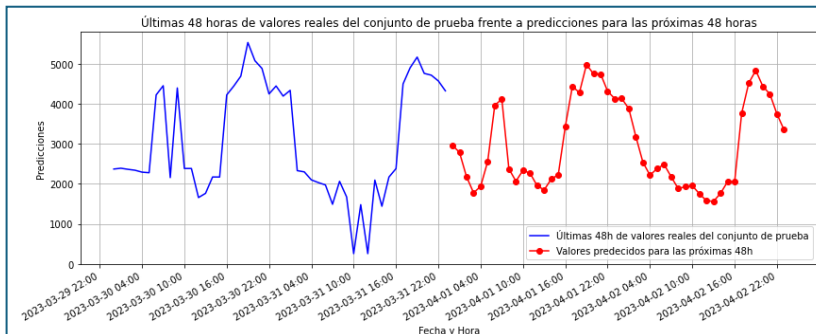


Fig. 1. Gráfica de los pronósticos a futuro del modelo LSTM.

Tabla 1. Lista de predicciones del PML para el nodo 07LPZ-115 La Paz, Baja California Sur, BCS.

	Fecha	Predicción	Limite Inferior	Limite Superior
720	2023-04-01 00:00:00	3112.005631	2193.283494	4038.913784
721	2023-04-01 01:00:00	2770.515998	1844.436235	3736.092496
722	2023-04-01 02:00:00	2517.998256	1660.765271	3393.738719
723	2023-04-01 03:00:00	2440.362533	1537.300040	3325.764496
724	2023-04-01 04:00:00	2572.406561	1680.079793	3476.827607
725	2023-04-01 05:00:00	2813.649044	1853.277718	3698.457952
726	2023-04-01 06:00:00	2963.441176	2103.976816	3896.577917
727	2023-04-01 07:00:00	2861.869027	1912.291430	3788.422816
728	2023-04-01 08:00:00	2512.488324	1604.671978	3448.388620
729	2023-04-01 09:00:00	2073.066336	1205.472347	3027.894809
730	2023-04-01 10:00:00	1724.196337	812.573193	2594.136177
731	2023-04-01 11:00:00	1541.021564	617.754523	2419.545514
732	2023-04-01 12:00:00	1485.305152	514.042457	2402.229779
733	2023-04-01 13:00:00	1511.350722	576.648782	2516.085676

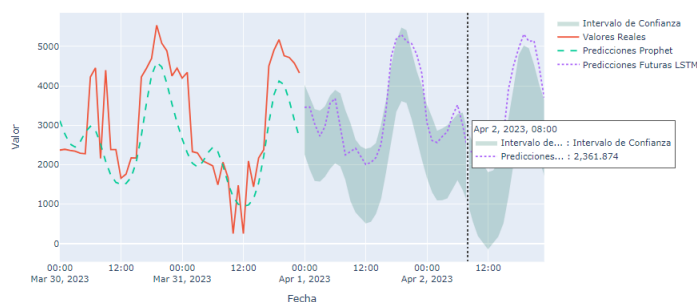


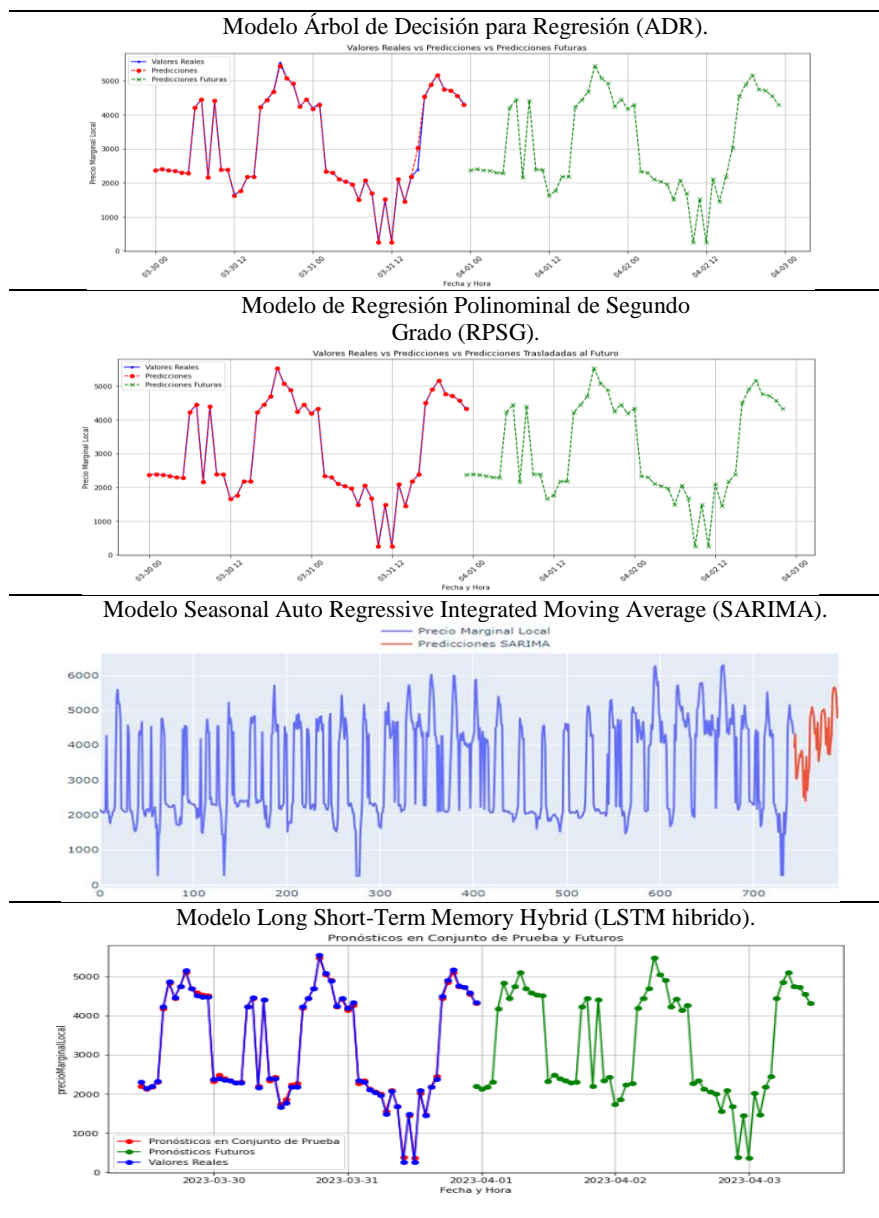
Fig. 8. Gráfica de predicciones del PML del nodo 07LPZ-115 La Paz, Baja California Sur, BCS.

La Tabla 3 muestra los nodos de la implementación del modelo, donde resaltan en colores azules los nodos de los sistemas BCA y BCS, mientras que los que no tienen colores son del sistema SIN. Lo más destacable de la tabla es que resume en orden las correlaciones de las variables que se identificaron de relevancia en el proyecto, están ordenadas de acuerdo con el coeficiente de correlación “r” y además respaldadas por





**Tabla 5.** Gráficas de los pronósticos obtenidos en cada modelo de Machine Learning para el nodo 07LPZ-115, La Paz, Baja Cal., BCS.



Sin embargo, su capacidad predictiva se limita a un solo paso adelante debido a su naturaleza no secuencial, lo que resulta en pronósticos estancados a largo plazo. La regresión polinómica de segundo grado, por su parte, presenta un ajuste excelente a los datos históricos, superando al árbol de decisión en ciertos aspectos. No obstante, su

incapacidad para capturar la temporalidad y generalizar valores volátiles se traduce en predicciones que son meras extrapolaciones de datos pasados. En contraste, el modelo SARIMA muestra un rendimiento inferior en los pronósticos, con errores significativos en comparación con los valores reales. En cuanto a la LSTM, aunque generalmente tiene un buen rendimiento, su elevado MAPE sugiere dificultades para predecir con precisión, posiblemente debido a la volatilidad de los datos o a la presencia de valores cero en variables clave como la precipitación y la congestión. Estas consideraciones son cruciales para interpretar la capacidad predictiva de cada modelo en escenarios dinámicos y realistas.

## **5. Conclusiones y trabajo a futuro**

Los árboles de decisión, utilizados en un contexto de regresión, demuestran ser una herramienta valiosa no solo para predecir precios sino también para comprender las relaciones subyacentes entre las variables y permite identificar las variables más relevantes para la predicción, lo que respalda la elección de árboles de decisión en el análisis de modelos de aprendizaje automático en este proyecto, convirtiéndolo en un referente en la construcción de modelos de pronósticos eficientes. La aplicación de técnicas de Machine Learning fue esencial para abordar la complejidad inherente a los datos de series temporales y mejorar sustancialmente la precisión de los pronósticos de PML en los nodos de energía. El modelo más destacado es la red LSTM "dense", con Prophet el cual demostró ser muy efectivo, especialmente por sus pronósticos razonables a futuro y la provisión de intervalos de confianza. Los resultados obtenidos establecen una base sólida para futuros desarrollos y aplicaciones prácticas en el sector energético de México. Para futuros desarrollos, se propone una mejora continua. Un desafío clave es la adaptación de las redes neuronales para prever “n” datos futuros, lo que requiere flexibilidad en la manipulación de “arrays” desiguales o incompatibles, lo que podría requerir un proyecto especializado para garantizar flexibilidad y realizar pruebas exhaustivas.

## **Referencias**

1. Alonso, G.: Reporte anual del mercado eléctrico mayorista 2016. Monitor independiente del Mercado (2017)
2. Ramos, A., Cortés, G., Latorre, J.M., Cerisola, S.: Análisis de la relación precio marginal y demanda de electricidad mediante conglomerados. In: X Congreso de Ingeniería de Organización (2006)
3. Loutfi, A.A., Sun, M., Loutfi, I., Solibakke, P.B.: Empirical Study of Day-ahead Electricity Spot-price Forecasting: Insights into a Novel Loss Function for Training Neural Networks. Elsevier, vol. 319 (2022). DOI: 10.1016/j.apenergy.2022.119182.
4. Singhal, D., Swarup, K.: Electricity Price Forecasting using Artificial Neural Networks. In: International Journal of Electrical Power and Energy Systems, vol. 33, no. 3, pp. 550–555 (2011). DOI: 10.1016/j.ijepes.2010.12.009.
5. Santiago-López, J.E., García, F.G.: Precios marginales locales: Análisis de volatilidad del MEM. In: IEEE, RVP-AI/2018, pp. 18 (2018)
6. Livas-García, A., May-Tzuc, O., Cruz-May, E., Tariq, R., Jimenez-Torres, M., Bassam, A.: Forecasting of Locational Marginal Price Components with Artificial Intelligence and

- Sensitivity Analysis: A Study under Tropical Weather and Renewable Power for the Mexican Southeast. Elsevier, vol. 206, (2022). DOI: 10.1016/j.epsr.2022.107793.
7. Yang, X., Fan, L., Li, X., Meng, L.: Day-ahead and Real-Time Market Bidding and Scheduling Strategy for Wind Power Participation based on Shared Energy Storage. *Electric Power Systems Research*, vol. 214 (2022). DOI: 10.1016/j.epsr.2022.108903.
  8. Zhang, J., Zhao, X., Jin, S., Greaves, D.: Phase-resolved Real-Time Ocean Wave Prediction with Quantified Uncertainty based on Variational Bayesian Machine Learning. Elsevier, vol. 234 (2022)
  9. Khuntia, S., Hanif, A., Ahmed, Q., Lahti, J., Meijer, M.: Cabin Load Prediction Using Time Series Forecasting in Long-haul Trucks for Optimal Energy Management. Elsevier, pp. 342–347 (2022)

# Algoritmos de aprendizaje automático para identificación de patrones SERS de pesticidas organofosforados

Mateo Frausto-Avila, Monserrat Ochoa-Elias,  
Gonzalo Ramírez-García, Mario Alan Quiroz-Juarez

Universidad Nacional Autónoma de México,  
Centro de Física Aplicada y Tecnología Avanzada,  
México

{mfrausto, maqj}@fata.unam.mx

**Resumen.** El estudio de los pesticidas organofosforados presenta obstáculos sustanciales tanto para el medio ambiente como para la salud pública. Las peligrosas propiedades de estos insecticidas han tenido impactos adversos en las personas y otras criaturas, lo que ha generado importantes desafíos en los países en desarrollo. Las técnicas existentes son complejas y requieren mucho tiempo, y existe un gran interés dentro de la comunidad científica de crear formas más rápidas y precisas. Este estudio utiliza las capacidades de autoaprendizaje de las redes neuronales artificiales para mejorar las capacidades analíticas de los espectros SERS. Los algoritmos de aprendizaje automático se entrenan utilizando patrones SERS que se adquieren directamente de las mediciones experimentales. Los hallazgos sugieren que el aprendizaje automático ofrece una mayor precisión en el análisis de espectros, creando así oportunidades para desarrollar técnicas analíticas más rápidas y eficientes para detectar pesticidas organofosforados en diferentes matrices ambientales.

**Palabras clave:** Redes neuronales artificiales, SERS, paratión, edifenfos.

## Machine Learning Algorithms for Identification of SERS Patterns of Organophosphate Pesticides

**Abstract:** The study of organophosphate pesticides presents substantial obstacles for both the environment and public health. The dangerous properties of these insecticides have had adverse impacts on people and other creatures, creating significant challenges in developing countries. Existing techniques are complex and time-consuming, and there is a great interest within the scientific community to create faster and more accurate ways. This study uses the self-learning capabilities of artificial neural networks to improve the analytical abilities of SERS spectra. Machine learning algorithms are trained using SERS spectra, which are acquired directly from experimental measurements. The findings suggest that machine learning offers greater precision in spectral analysis, thus creating opportunities to develop faster and more efficient analytical techniques to detect organophosphate pesticides in different environmental matrices.

**Keywords:** Artificial neural networks, SERS, parathion, edifenphos.

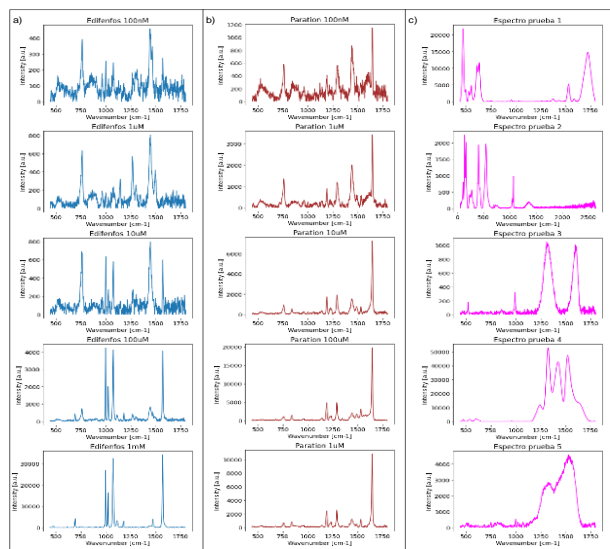
## 1. Introducción

La exposición a pesticidas organofosforados plantea desafíos significativos en términos de salud humana y cuidado del medio ambiente [1]. Estos pesticidas desempeñan un papel crucial en el aumento de la producción agrícola al prevenir y controlar una amplia variedad de plagas, pero estudios han sugerido que su presencia tiene efectos tóxicos en humanos y otros mamíferos [2]. Este hecho ha generado problemas notables en países en desarrollo, donde la falta de conocimiento generalizado sobre los riesgos asociados con estos pesticidas ha resultado en numerosos efectos nocivos entre los trabajadores agrícolas [3].

La determinación de residuos de pesticidas organofosforados en alimentos es de suma importancia debido a su impacto potencial en la salud humana. La mayoría de los métodos disponibles para esta determinación son complicados y laboriosos. Por esta razón, el desarrollo de métodos rápidos y precisos para la determinación de estos pesticidas es de gran interés para la comunidad científica. Existen algunos métodos de determinación rápida, los cuales se dividen en dos categorías principales: métodos electroquímicos [4-7] y espectrales [8-11]. Si bien, los métodos electroquímicos presentan ventajas en términos de tiempos de determinación cortos, alta sensibilidad y bajo costo, la necesidad de sintetizar reactivos intrincados los hace inadecuados para una amplia variedad de aplicaciones. Por otro lado, los métodos espectrales, como la espectrometría de fluorescencia, infrarroja y Raman, son rápidos y emplean procedimientos simples [12].

Entre las técnicas espectrales, la tecnología de dispersión Raman mejorada en superficie (Surface Enhanced Raman Spectroscopy, SERS por sus siglas en inglés) ha surgido como una herramienta poderosa para la determinación rápida de pesticidas [13]. Los sustratos SERS se basan principalmente en nanopartículas de metales nobles (Ag, Au), y son capaces de intensificar las señales Raman con un factor de mejora (EF) de hasta 107 veces, permitiendo identificar los pesticidas con alta sensibilidad y precisión [14,15]. Sin embargo, el análisis de los espectros es complejo y usualmente tardado, impidiendo que esta estrategia analítica se aplique de manera masiva. A este respecto, es necesario desarrollar estrategias que permitan simplificar y automatizar el análisis de datos. Actualmente, el aprendizaje automático (machine learning, ML por sus siglas en inglés) se ha convertido en una potencia innovadora, demostrando su capacidad para detectar patrones complejos y ofrecer nuevas perspectivas en numerosos ámbitos científicos [16,17]. El campo de la investigación ha experimentado una profunda revolución como resultado de la capacidad de los algoritmos de aprendizaje automático para revelar conexiones intrincadas y ocultas dentro de vastos conjuntos de datos [18].

En este trabajo, explotamos las características de autoaprendizaje de redes neuronales artificiales (Neural Networks, NN por sus siglas en inglés) para potenciar la capacidad analítica de los espectros SERS. Estos algoritmos de ML son entrenados con patrones SERS directamente obtenidos del espectrómetro Raman. En este punto, resaltamos que nuestro algoritmo no demanda preprocesamiento de datos además de que permanece compacto, al contar únicamente con una capa oculta de 5 neuronas. Se analizaron espectros SERS con diferentes concentraciones de paratión y edifenfos, dos



**Fig. 1.** Ejemplos de espectros SERS en concentraciones variables (100 nM, 1  $\mu$ M, 10  $\mu$ M, 100  $\mu$ M y 1mM) para: a) Edifenfos, b) Paratión y c) compuestos desconocidos.

pesticidas usados como insecticida y fungicida respectivamente. Mediante un modelo de redes neuronales de clasificación se pudieron identificar exitosamente los espectros SERS de ambas moléculas, así como una tercera molécula de prueba desconocida que sirve como tercera clase en la clasificación. Los resultados indican que ML proporciona una mayor precisión en el análisis de los espectros, lo que abre nuevas posibilidades para avanzar hacia métodos analíticos más rápidos y eficientes que permitan la detección de pesticidas organofosforados en diversas matrices ambientales y alimentos.

## 2. Metodología

### 2.1. Preparación de sustratos SERS

Las nanoestrellas de oro se sintetizaron conforme a la metodología reportada por Panikar et al [19]. Brevemente, una disolución 0.25 de  $\text{HAuCl}_4$ , volúmenes fijos de semillas de oro previamente sintetizadas mediante el método de Turkevich [20], y ácido clorhídrico (HCl) 1 M fueron mezclados con agitador magnético durante 2 minutos. Después, se añadió nitrato de plata ( $\text{AgNO}_3$ ) 3 mM y de ácido ascórbico 0.1M ( $\text{C}_6\text{H}_8\text{O}_6$ ). La reacción se continuó durante 3 minutos antes de agregar 2 mL de solución de bromuro de hexadeciltrimetilamonio (CTAB). Finalmente, se dejó en agitación durante 30 minutos. Por último, se centrifugaron y enjuagaron 2 veces a 600 revoluciones por minuto (rpm) por 10 minutos.

Volúmenes fijos de las nanoestrellas fueron depositados sobre sustratos de aluminio, y se permitió que se secaran a temperatura ambiente. Finalmente, se añadieron concentraciones variables (100 nM, 1  $\mu$ M, 10  $\mu$ M, 100  $\mu$ M y 1mM) de los pesticidas sobre la muestra anterior, y se permitió nuevamente que se secaran a temperatura

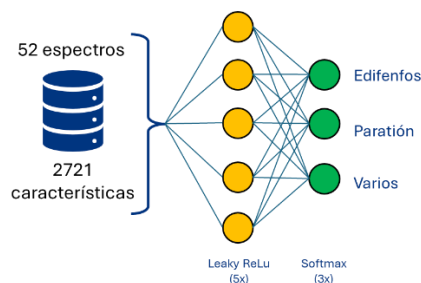


Fig. 2. Modelo de la red neuronal empleada.

ambiente. Los espectros se obtuvieron con el espectrómetro Raman Senterra de Bruker, con un láser de 785 nm. Espectros representativos de las concentraciones analizadas pueden verse en la Fig. 1.

## 2.2. Obtención de los espectros SERS

Los espectros Raman de los pesticidas paratión y edifenfos se obtuvieron mediante la técnica SERS con nanoestrellas de oro, las cuales generan un incremento del campo electromagnético debido a su plasmón de resonancia. En la molécula edifenfos las bandas 1000, 1020 y 1080  $\text{cm}^{-1}$  corresponden a los enlaces C=C, C-H y C-S, respectivamente. En el paratión las bandas 760, 1347 y 1590  $\text{cm}^{-1}$  representan los enlaces de P=S, N-O y C-H.

## 2.3. Preparación de la base de datos

Para la creación de la base de datos, se emplearon 52 espectros SERS; 18 espectros de edifenfos, 19 de paratión y 15 de compuestos desconocidos, estos últimos fueron usados con el fin de propiciar variabilidad a la clasificación de los espectros. Cada espectro está conformado de 2721 filas y 2 columnas (número de onda y amplitud respectivamente). La base de datos fue construida únicamente utilizando la columna de amplitud de cada espectro ya que el vector de número de onda se conserva en cada registro. Este nuevo vector con los datos de amplitud de la medición pasa a ser nuestro vector de características, y con cada registro se construye un dataframe de 52 filas y 2721 columnas.

## 2.4. Modelo de aprendizaje automático

Nuestro modelo de ML fue programado en Python y consiste en una red neuronal de 2 capas (ver Fig. 2) donde la capa oculta está compuesta de 5 neuronas tipo Leaky-ReLU. Finalmente, la capa de salida consta de 3 neuronas tipo Softmax para clasificar si el espectro corresponde a edifenfos, paratión o una molécula diferente. Los pesos sinápticos fueron optimizados a través del método Adam [21], en la dirección que minimiza la función de costo, específicamente la entropía cruzada. En esencia, el método de Adam es una modificación del método del descenso del gradiente



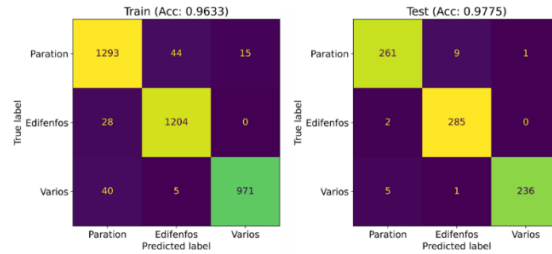


Fig. 3. Matrices de confusión para los datos de entrenamiento y prueba.

estocástico, que se basa en la estimación del primer y segundo momento de la derivada direccional.

### 3. Resultados

Durante el entrenamiento se usó el 70% de los datos. 15% de los datos se reservaron para el segmento de prueba, y finalmente, el 15% restante de los datos se destinaron para validación. Es importante mencionar que se llevaron a cabo 100 realizaciones del entrenamiento para fines estadísticos. Los resultados obtenidos indican que la red neuronal exhibe una precisión del 96.33%, 97.75% y 97.12% en las etapas de entrenamiento, prueba y validación, respectivamente. Las matrices de confusión promedio para el conjunto de datos de entrenamiento y prueba se muestran en la Fig. 3. Con el fin de evitar sobre entrenamiento del modelo propuesto, incorporamos regularización L2 con un factor de penalización del orden de  $10^{-3}$ .

Con el fin de evaluar el rendimiento de nuestra red neuronal, comparamos nuestro modelo con dos algoritmos diferentes de aprendizaje automático: Random Forest (RF) y Maquinas de Soporte Vectorial (SVM por sus siglas en ingles). Estos algoritmos fueron sometidos a 100 realizaciones como en el caso de la red neuronal. El algoritmo RF se compone de un conjunto de árboles predictores, donde las predicciones de cada árbol están influenciadas por los valores de un vector aleatorio. Este vector aleatorio se muestrea de forma independiente y sigue la misma distribución para todos los árboles del bosque. El error de generalización para los bosques converge hacia un límite a medida que aumenta el número de árboles en el bosque. Por su parte, el algoritmo SVM es un método de aprendizaje automático que tiene como objetivo crear hiperplanos en un espacio N-dimensional para distinguir entre conjuntos de datos que pertenecen a clases distintas. Los hiperplanos sirven como límites de decisión que establecen una distancia de margen, lo que permite la clasificación. La Tabla 1 muestra la precisión obtenida en NN, RF y SVM.

### 4. Conclusiones

Los algoritmos de aprendizaje automático permiten analizar los espectros SERS de manera rápida, automatizada, y con gran precisión, abriendo nuevas posibilidades para

**Tabla 1.** Precisiones obtenidas de la red neuronal (NN), random forest (RF) y máquina de soporte vectorial (SVM).

Algoritmo	Precision (%)
NN	97.75%
RF	94.30%
SVM	94.15%

la detección de pesticidas organofosforados en diversas matrices ambientales y de alimentos. El estudio realizado utilizando espectros SERS de paratión y edifentos demostró la efectividad de esta técnica en la identificación de pesticidas. Nuestra red neuronal presenta una precisión de 97.75%, mostrando su capacidad para clasificar con precisión los espectros correspondientes a diferentes pesticidas y compuestos desconocidos.

Para establecer un punto de comparación se entrenó una máquina de soporte vectorial y un algoritmo de random forest. Estos modelos exhiben precisiones cercanas al 94% en ambos casos.

**Agradecimientos.** MAQJ agradece el apoyo de la Dirección General de Asuntos del Personal Académico (DGAPA)-Universidad Nacional Autónoma de México a través del proyecto UNAM-PAPIIT TA101023 y al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT) de México bajo el proyecto CF-2023-I-1496.

## Referencias

1. Phopin, K., Tantimongcolwat, T.: Pesticide Aptasensors—State of the Art and Perspectives. *Sensors*, vol. 20, no. 23, pp. 6809 (2020). DOI: 10.3390/s20236809.
2. Yu, J.J.: Removal of Organophosphate Pesticides from Wastewater by Supercritical Carbon dioxide extraction. *Water Research*, vol. 36, no. 4, pp. 1095–1101 (2002). DOI: 10.1016/S0043-1354(01)00293-7.
3. Rani, L., Thapa, K., Kanojia, N., Sharma, N., Singh, S., Grewal, A.S., Srivastav, A.L., Kaushal, J.: An Extensive Review on the Consequences of Chemical Pesticides on Human Health and Environment. *Journal of Cleaner Production*, vol. 283, pp. 124657 (2021). DOI: 10.1016/j.jclepro.2020.124657.
4. Sidhu, G.K., Singh, S., Kumar, V., Dhanjal, D.S., Datta, S., Singh, J.: Toxicity, Monitoring and Biodegradation of Organophosphate Pesticides: A Review. *Critical Reviews in Environmental Science and Technology*, vol. 49, no. 13, pp. 1135–1187 (2019). DOI: 10.1080/10643389.2019.1565554.
5. Kim, Y.J., Cho, Y.A., Lee, H.S., Lee, Y.T., Gee, S.J., Hammock, B.D.: Synthesis of Haptens for Immunoassay of Organophosphorus Pesticides and Effect of Heterology in Hapten Spacer Arm Length on Immunoassay Sensitivity. *Analytica Chimica Acta*, vol. 475, no. 1-2, pp. 85–96 (2003). DOI: 10.1016/S0003-2670(02)01037-1.
6. Liang, Y., Liu, X.J., Liu, Y., Yu, X.Y., Fan, M.T.: Synthesis of Three Haptens for the Class-Specific Immunoassay of O, O-dimethyl Organophosphorus Pesticides and Effect of Hapten Heterology on Immunoassay Sensitivity. *Analytica Chimica Acta*, vol. 615, no. 2, pp. 174–183 (2008). DOI: 10.1016/j.aca.2008.03.050.

7. Amine, A., Mohammadi, H., Bourais, I., Palleschi, G.: Enzyme Inhibition-based Biosensors for Food Safety and Environmental Monitoring. *Biosensors and Bioelectronics*, vol. 21, no. 8, pp. 1405–1423 (2006). DOI: 10.1016/j.bios.2005.07.012.
8. Rainina, E.I., Efremenco, E.N., Varfolomeyev, S.D., Simonian, A.L., Wild, J.R.: The Development of a New Biosensor based on Recombinant E. Coli for the Direct Detection of Organophosphorus Neurotoxins. *Biosensors and Bioelectronics*, vol. 11, no. 10, pp. 991–1000 (1996). DOI: 10.1016/0956-5663(96)87658-5.
9. Viveros, L., Paliwal, S., McCrae, D., Wild, J., Simonian, A.: A Fluorescence-based Biosensor for the Detection of Organophosphate Pesticides and Chemical Warfare Agents. *Sensors and Actuators B. Chemical*, vol. 115, no. 1, pp. 150–157 (2006). DOI: 10.1016/j.snb.2005.08.032.
10. Simonian, A.L., Good, T.A., Wang, S.S., Wild, J.R.: Nanoparticle-based Optical Biosensors for the Direct Detection of Organophosphate Chemical Warfare Agents and Pesticides. *Analytica Chimica Acta*, vol. 534, no. 1, pp. 69–77 (2005). DOI: 10.1016/j.aca.2004.06.056.
11. Janotta, M., Karlowatz, M., Vogt, F., Mizaikoff, B.: Sol–gel based Mid-Infrared Evanescent Wave Sensors for Detection of Organophosphate Pesticides in Aqueous Solution. *Analytica Chimica Acta*, vol. 496, no. 1-2, pp. 339–348 (2003). DOI: 10.1016/S0003-2670(03)01011-0.
12. Tanner, P.A., Leung, K.H.: Spectral Interpretation and Qualitative Analysis of Organophosphorus Pesticides using FT-Raman and FT-infrared Spectroscopy. *Applied Spectroscopy*, vol. 50, no. 5, pp. 565–571 (1996)
13. Weng, S., Li, M., Chen, C., Gao, X., Zheng, S., Zeng, X.: Fast and Accurate Determination of Organophosphate Pesticides using Surface-Enhanced Raman Scattering and Chemometrics. *Analytical Methods*, vol. 7, no. 6, pp. 2563–2567 (2015). DOI: 10.1039/C4AY03067B.
14. Li, B., Zhang, W., Chen, L., and Lin, B.: A Fast and Low-Cost Spray Method for Prototyping and Depositing Surface-Enhanced Raman Scattering Arrays on Microfluidic Paper based Device. *Electrophoresis*, vol. 34, no. 15, pp. 2162–2168 (2013). DOI: 10.1002/elps.201300138.
15. Liron, Z., Zifman, A., Heleg-Shabtai, V.: Surface-Enhanced Raman Scattering Detection of Cholinesterase Inhibitors. *Analytica Chimica Acta*, vol. 703, no. 2, pp. 234–238 (2011). DOI: 10.1016/j.aca.2011.07.033.
16. Mjolsness, E., DeCoste, D.: Machine Learning for Science: State of the Art and Future Prospects. *Science*, vol. 293, no. 5537, pp. 2051–2055 (2001). DOI: 10.1126/science.293.5537.2051.
17. Device: *Electrophoresis*, vol. 34, no. 15, pp. 2162–2168 (2020)
18. Sarker, I.H.: Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, vol. 2, no. 3, pp. 160 (2021). DOI: 10.1007/s42979-021-00592-x.
19. Panikar, S.S., Ramírez-García, G., Sidhik, S., Lopez-Luke, T., Rodriguez-Gonzalez, C., Ciapara, I.H., De la Rosa, E.: Ultrasensitive SERS Substrate for Label-Free Therapeutic-Drug Monitoring of Paclitaxel and Cyclophosphamide in Blood Serum. *Analytical Chemistry*, vol. 91, no. 3, pp. 2100–2111 (2018). DOI: 10.1021/acs.analchem.8b04523.
20. Ramírez-García, G., Honorato-Colin, M.A., De la Rosa, E., López-Luke, T., Panikar, S.S., de Jesús Ibarra-Sánchez, J., Piazza, V.: Theranostic Nanocomplex of Gold-Decorated up Conversion Nanoparticles for Optical Imaging and Temperature-Controlled Photothermal Therapy. *Journal of Photochemistry and Photobiology A: Chemistry*, vol. 384, pp. 112053 (2019)

*Mateo Frausto-Avila, Monserrat Ochoa-Elias, et al.*

21. Kingma, D.P., Lei-Ba, J.: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)

## Mejoramiento del agrupamiento de datos mezclados e incompletos mediante algoritmos bioinspirados

Claudia C. Tusell-Rey<sup>1</sup>, Yenny Villuendas-Rey<sup>2</sup>,  
Oscar Camacho-Nieto<sup>2</sup>, Viridiana Salinas-García<sup>2</sup>

<sup>1</sup> Instituto Politécnico Nacional,  
Centro de Investigación en Computación,  
México

<sup>2</sup> Instituto Politécnico Nacional,  
Centro de Innovación y Desarrollo Tecnológico en Cómputo,  
México

clautusellrey2014@gmail.com,  
{yvilluendasr, ocamacho, vsalinasg}@ipn.mx

**Resumen.** Mejorar los resultados del agrupamiento de datos mixtos (numéricos y categóricos) y con ausencias de información es fundamental para reconocer patrones y tomar de decisiones en diversos ámbitos. Los algoritmos de agrupamiento tradicionales a menudo tienen problemas con tipos de datos heterogéneos e información incompleta, lo que genera grupos subóptimos y conocimientos potencialmente sesgados. Al abordar estos desafíos, las técnicas avanzadas como los algoritmos bioinspirados, pueden proporcionar grupos más precisos y completos. Recientemente se propuso un método que mejora los resultados obtenidos mediante algoritmos de agrupamiento: el PAntSA; pero este sólo fue diseñado y probado para datos numéricos. Por este motivo, este trabajo analiza la influencia de la aplicación del PAntSA en el rendimiento de algoritmos de agrupamiento mezclados e incompletos. Para ello se comparan los resultados de diferentes algoritmos antes y después de aplicar el PAntSA. El análisis estadístico de los resultados proporciona evidencia experimental que respalda que el algoritmo PAntSA mejora la calidad de los grupos obtenidos mediante métodos tradicionales de agrupamiento de datos mixtos e incompletos.

**Palabras clave:** Agrupamiento bio-inspirado, datos mezclados e incompletos, PAntSA.

### Bioinspired Based Improvement of Mixed and Incomplete Data Clustering

**Abstract.** Enhancing the results of data clustering for mixed (numeric and categorical) and missing data is paramount for robust pattern recognition and decision-making in various fields. Traditional clustering algorithms often struggle with heterogeneous data types and incomplete information, leading to

suboptimal groupings and potentially biased insights. By addressing these challenges, advanced techniques, such as bioinspired algorithms, can provide more accurate and comprehensive clusterings. Recently, a method was proposed that improves the results obtained by clustering algorithms, PAntSA; but this was only designed and tested for numerical data. For this reason, this work analyzes the influence of applying PAntSA on the performance of mixed and incomplete clustering algorithms. To do this, the results of different algorithms are compared before and after applying PAntSA. The statistical analysis of the results provides experimental evidence that supports that the PAntSA algorithm improves the quality of the groups obtained by traditional mixed and incomplete data clustering methods.

**Keywords:** Bioinspired clustering, mixed and incomplete data, PAntSA.

## 1. Introducción

El proceso de agrupar un conjunto de objetos físicos o abstractos dentro de clases con objetos similares se denomina agrupamiento. En este proceso se toma una colección dada de datos no etiquetados y se crea un conjunto de grupos de tal manera que los objetos que pertenecen a un grupo sean homogéneos entre sí [1], buscando además que la heterogeneidad entre los distintos grupos sea lo más elevada posible. Esta técnica ha adquirido gran relevancia en los últimos tiempos debido a su aplicación práctica en la solución exitosa de disímiles problemas de la vida real como: el reconocimiento del habla, la segmentación de imágenes y visión por computadora, la recuperación de información y minería de textos, en biología computacional para el análisis de ADN y muchas otras aplicaciones.

La mayoría de los algoritmos de agrupamiento se han diseñado para trabajar sólo con datos numéricos o con datos categóricos, mientras que, en una gran cantidad de ocasiones, es necesario trabajar con datos mezclados, es decir, con atributos de distintos tipos como: numéricos, binarios, discretos y categóricos. También en muchas ocasiones no es posible conocer el valor de un determinado atributo, por lo que se requiere además desarrollar algoritmos para agrupar datos incompletos [2].

El agrupamiento de datos mezclados e incompletos (DMI) ha sido abordado tradicionalmente siguiendo paradigmas clásicos como el jerárquico y particional, aunque también han aparecido propuestas bio-inspiradas que han tenido también un buen desempeño [3-5]. Por otra parte, el agrupamiento obtenido por un algoritmo dado puede ser mejorado mediante otro algoritmo utilizando un índice de validación interno. Empleando esta idea fue publicado el algoritmo bio-inspirado PAntSA [6] (basado en un árbol de hormigas [7,8]) el cual toma los resultados obtenidos por un algoritmo de agrupamiento previo e intenta perfeccionarlos utilizando el índice de la Silueta [9] y la definición de una atracción entre grupos. PAntSA mejora la calidad de los resultados obtenidos por algoritmos de agrupamiento en datos numéricos, particularmente en la clasificación de documentos; sin embargo, hasta donde sabemos, no existe un estudio de la influencia de PAntSA en el agrupamiento de DMI. Por este motivo, en este trabajo nos proponemos analizar la influencia del PAntSA en el mejoramiento de los resultados de los algoritmos de agrupamiento de DMI.

A partir de aquí el resto del trabajo está organizado como sigue: En la sección 2 mostramos algunos trabajos relacionados con el agrupamiento de DMI. En la sección 3 describimos el algoritmo PAntSA empleado para mejorar los grupos obtenidos por otros algoritmos. En la sección 4 presentamos un análisis experimental sobre la influencia del PAntSA en el agrupamiento de DMI y los resultados obtenidos. Finalmente, en la sección 5 se ofrecen las conclusiones obtenidas.

## 2. Trabajos relacionados

Los algoritmos de agrupamiento existentes para DMI, en su mayoría, son producto de extensiones realizadas a métodos para el manejo de tipos de datos homogéneos (numéricos o no numéricos), estos pueden dividirse en varias categorías según el procedimiento que utilizan para agrupar los objetos.

### 2.1. Algoritmos particionales

La estrategia empleada por los algoritmos particionales para encontrar los grupos es ir reubicando iterativamente objetos entre subconjuntos. En 1967 J. MacQueen propuso un algoritmo clásico que se ha denominado *k-means* el cual se considera arquetipo del modelo particional [10]. El algoritmo *k-means* para su funcionamiento tiene que conocer un número  $k$ , que es la cantidad de grupos que se desean obtener. La idea es ubicar  $k$  centroides y agrupar los objetos por su centroide más cercano según una función de distancia definida a priori. Iterativamente, se van actualizando los centroides como el promedio de los objetos que pertenecen a cada grupo, hasta que los centroides dejen de cambiar.

A pesar de su poca complejidad temporal, estos algoritmos presentan muchas desventajas: por ejemplo, los resultados finales dependen de la inicialización de los centroides. Además, son inválidos ante objetos con atributos categóricos por la necesidad de calcular el promedio y resultan inadecuados para detectar grupos no convexos y *outliers* (objetos ruidosos).

Unas de las primeras extensiones realizadas al *k-means* para tratar con DMI fue *k-Prototypes* publicado por Huang en 1997 [11]. Este autor basa su propuesta en la definición de una nueva función de disimilitud entre los objetos, de forma tal que se permite tratar las descripciones mezcladas.

Sea  $o$  un objeto,  $c_i$  el centro del  $i$ -ésimo grupo y  $X_p$  el  $p$ -ésimo atributo. La disimilitud entre el objeto y el centro se presenta a continuación:

$$d(o, c_i) = \sum_{p \in R_n} (X_p(o) - X_p(c_i))^2 + \gamma_i \sum_{r \in R_c} \Gamma(X_r(o), X_r(c_i)), \quad (1)$$

donde:  $\gamma_i$  es un parámetro del algoritmo,  $R_n$  es el conjunto de rasgos numéricos,  $R_c$  el conjunto de rasgos categóricos, y  $\Gamma(X_r(O_j), X_r(c_i))$  es una función de disimilitud que es igual a cero si  $X_r(O_j) = X_r(c_i)$  y uno en otro caso. Además, realiza una modificación en la forma de obtener los centros de los grupos tomando como valores la media de los atributos numéricos, y la moda de los atributos categóricos.

También Ahmad y Dey propusieron otra modificación al *k-means* [12]. Las modificaciones realizadas consisten en la actualización de la función de disimilitud,

teniendo en cuenta la contribución de cada atributo a cada grupo. Este algoritmo no presenta nombre, solo se enuncia un pseudocódigo bajo el título *modified\_kmean\_subspace\_clustering*, por lo cual decidimos a los efectos de este trabajo, referirnos al algoritmo como AD2011.

## 2.2. Algoritmos jerárquicos

Los algoritmos jerárquicos, como su nombre indica, construyen una jerarquía de agrupamientos, uniendo o dividiendo los grupos de acuerdo con una cierta función de similitud/disimilitud entre los grupos. En otras palabras, construyen un árbol de grupos llamado dendograma. Tal enfoque permite estudiar los datos con diferentes niveles de granularidad.

Los algoritmos de agrupamiento jerárquicos se categorizan en aglomerativos (*bottom-up*) y divisivos (*top-down*). Un agrupamiento aglomerativo, generalmente, comienza con grupos unitarios (*singleton clusters*) y, recursivamente, une dos o más grupos apropiados. El proceso continúa hasta que se alcanza algún criterio de parada (frecuentemente el número  $k$  de grupos).

Entre las ventajas de los algoritmos de agrupamiento jerárquicos se puede mencionar la flexibilidad con respecto al nivel de granularidad, son fáciles de manejar y son aplicables a cualquier tipo de atributo. Entre las desventajas se encuentran el no mejoramiento de los grupos que han sido construidos por la no consideración de los objetos ya asignados y la sensibilidad al ruido. Además, el costo computacional para la mayoría de estos algoritmos es también como mínimo de  $O(m^2)$ , donde  $m$  es el número de objetos, lo que limita su aplicación a grandes conjuntos de datos.

En 1999, Reyes-González y Ruiz-Shulcloper propusieron un nuevo algoritmo aglomerativo para DMI [13]. El algoritmo (al que denominaremos AERE) en cada paso forma un nuevo nivel, hasta que todos los objetos se encuentren en el mismo nivel. Un nivel está definido por la cantidad de grupos presentes en el nivel, el valor de  $\beta_0$  (similitud máxima entre dos grupos del mismo nivel), y el conjunto de particiones posibles. Como elemento distintivo se tiene que es determinista (siempre obtiene la misma solución), y que cada grupo esté formado por elementos que se encuentran en la misma componente  $\beta_0$  conexa en un grafo de máxima similitud. Finalmente, el algoritmo devuelve todas las estructuraciones posibles (conjunto de particiones) para el nivel deseado  $k$  de la jerarquía formada.

También para el agrupamiento de DMI, en 2005, fue propuesto un algoritmo jerárquico denominado HIMIC (*Hierarchical Mixed type data Clustering algorithm*) [14]. El algoritmo se basa en el uso de una función de disimilitud entre dos grupos  $C_i, C_j$  que considera el conjunto de valores categóricos posibles  $D_p$  la cual viene dada por la siguiente ecuación:

$$d(C_i, C_j) = \sum_{p=1}^n S_p(C_i, C_j), \quad (2)$$

donde:



2	1	2	1	1	2	1	2	1	2
---	---	---	---	---	---	---	---	---	---

**Fig. 1.** Ejemplo de un individuo. En este caso, se tienen 10 objetos, agrupados en dos grupos. Cada posición codifica el grupo al que pertenece dicho objeto:

$$S_p(C_i, C_j) = \begin{cases} 1 - \left| \frac{1}{|C_i|} \sum_{o \in C_i} X_p(o) - \frac{1}{|C_j|} \sum_{o \in C_j} X_p(o) \right| & \text{si } p \text{ es numérico,} \\ \sum_{l=1}^{|D_p|} \frac{|\{o \in C_i | X_p(o) = v_l\}|}{|C_i|} * \frac{|\{o \in C_j | X_p(o) = v_l\}|}{|C_j|} & \text{si } p \text{ es categórico.} \end{cases}$$

Posteriormente se aplica un método aglomerativo tradicional, utilizando como criterio de parada la obtención del número deseado de grupos  $k$ .

### 2.3. Algoritmos basados en técnicas de optimización

El problema de agrupar datos puede verse como un problema de optimización que localiza los centroides óptimos de los grupos o encuentra la partición óptima de un conjunto de objetos. Por este motivo se han empleado con éxito diferentes técnicas de optimización que ayudan a encontrar la mejor solución o al menos una solución lo suficientemente buena para un problema en un espacio de búsqueda. En este sentido destacan las metaheurísticas como algoritmos aproximados que intentan resolver estos problemas, sacrificando la garantía de encontrar el óptimo a cambio de encontrar una "buena solución en un tiempo razonable", razón por la cual han sido utilizadas para el agrupamiento de grandes conjuntos de datos.

Se puede decir que una metaheurística es una estrategia de alto nivel que usa diferentes estrategias para explorar el espacio de búsqueda y por su fácil adaptación, simplicidad y eficiencia están entre los métodos aproximados más ampliamente usados. A continuación, describiremos tres propuestas de algoritmos de agrupamiento de DMI que se basan en metaheurísticas.

El algoritmo AKGA propuesto en [15] consiste en la utilización de un Algoritmo Genético (GA) para obtener grupos sin necesidad de aplicar exhaustivamente un algoritmo de agrupamiento. La aplicación de un GA permite salir de óptimos locales, y buscar por un óptimo global, sin un costo computacional demasiado elevado. Como esquema de representación, cada solución consiste en un individuo, representado por una cadena de tamaño igual a la cantidad de objetos, y donde cada elemento  $i$ -ésimo de la cadena representa el grupo al cual el  $i$ -ésimo objeto está asignado (ver Fig. 1).

En el caso de los centros de los grupos, los autores utilizan el esquema de Ahmad y Dey de 2011 [12]. También utilizan la función de disimilitud de los objetos a los centros de [12] como parte de la función de optimización del Algoritmo genético.

Una metaheurística que ha sido empleada con éxito en el agrupamiento de datos numéricos es ABC (*Artificial Bee Colony*) inspirada en el comportamiento natural de las abejas y propuesta por Karaboga [16] para optimización numérica. Una extensión a esta para agrupar DMI lo constituye BECA (*BEE based Clustering Algorithm*) [4]. Este algoritmo genera  $n$  agrupamientos iniciales de forma aleatoria que constituyen las

fuentes de alimento. Luego genera nuevas fuentes empleando la estrategia de mutación definida en [15] y utilizando como disimilitud la HEOM [17] que permite tratar con DMI. Para la evaluación de las fuentes de alimento emplea como función objetivo un índice de validación interno, en este caso el índice de Dunn [9] para obtener grupos compactos y bien separados. Finalmente, después de un número de iteraciones definido a priori se devuelve la fuente de alimento (agrupamiento) que optimizó la función objetivo.

## 2.4. Índices de validación internos

La Silueta (*Silhouette*) [9] es un índice de validación interno que combina dos elementos claves para la calidad de un agrupamiento dado: compacidad y separabilidad. Este índice calcula el promedio, para todos los grupos, del ancho de la silueta de sus puntos. Si  $x$  es un objeto del grupo  $c_i$  y  $n_i$  es el número de objetos en  $c_i$ , entonces la silueta de  $x$  se define por la siguiente ecuación:

$$S(x) = \frac{b(x) - a(x)}{\max\{a(x), b(x)\}}, \quad (3)$$

donde  $a(x)$  es la distancia promedio a todos los otros objetos en  $c_i$ , y  $b(x)$  es el mínimo del promedio de las disimilitudes de  $x$  y los objetos en los otros grupos:

$$a(x) = \frac{1}{n_i - 1} \sum_{\substack{y \in c_i \\ y \neq x}} d(x, y), \quad (4)$$

$$b(x) = \min_{\substack{h=1..k \\ h \neq i}} \left\{ \frac{1}{n_h} \sum_{y \in c_h} d(x, y) \right\}, \quad (5)$$

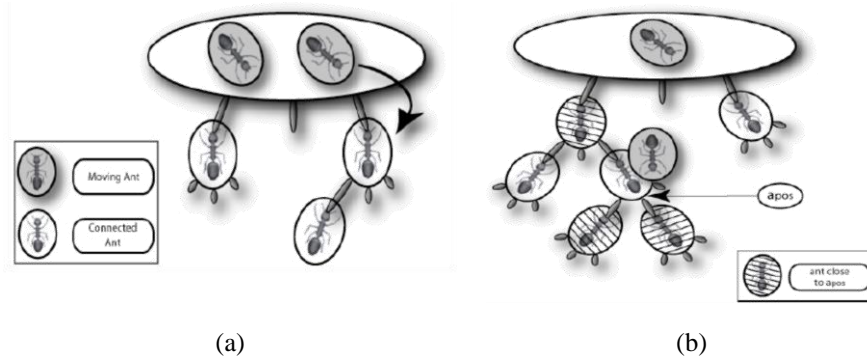
finalmente, la silueta global se define por:

$$S = \frac{1}{k} \sum_{i=1}^k \frac{1}{n_i} \sum_{x \in c_i} S(x). \quad (6)$$

Para un objeto dado  $x$ , el ancho de su silueta varía entre -1 y 1. Si el resultado se encuentra cercano a -1, esto significa que dicho objeto es más similar, como promedio, a un grupo al cual él no pertenece. Por el contrario, si el valor es cercano a 1, quiere decir que la disimilitud promedio entre el objeto que se analiza y los objetos de su grupo, es significativamente menor que la disimilitud promedio con respecto a cualquier otro grupo. Mientras mayor sea la Silueta, más compactos y separados serán los grupos. Debido al rol importante que representa este índice en la validación interna de agrupamientos ha sido empleado en el algoritmo PAntSA para mejorar los resultados un algoritmo de agrupamiento dado.

## 3. Algoritmo PAntSA

El algoritmo PAntSA [6] está basado en el algoritmo AntTree, propuesto en [7,8]. Para el mejor entendimiento del PAntSA, explicaremos en detalle primeramente el



**Fig. 2.** Movimiento de  $a_i$  sobre el árbol. (a) Representación del árbol formado por hormigas conectadas y hormigas en movimiento. (b) Posibilidades de movimiento de una hormiga que se encuentra sobre apos. (figuras tomadas y adaptadas de [6]).

funcionamiento del algoritmo AntTree. Este algoritmo es pionero en la aplicación de la modelación de la construcción de nidos por las hormigas, a problemas de Inteligencia Artificial. El AntTree se basa en modelar la habilidad de las hormigas para construir estructuras vivientes con sus cuerpos [8], para descubrir, de forma distribuida y no supervisada, una estructura arbórea que organice un conjunto de datos. Esta estructura jerárquica puede ser interpretada de varias formas: como una partición de los datos o como una estructuración jerárquica de los mismos [8].

El principio fundamental del AntTree es el siguiente: cada hormiga representa un nodo en el árbol que será construido (es decir, los objetos que serán agrupados) y existe una función de similitud entre dos objetos  $Sim(i,j)$ .

Sobre la base de un nodo raíz ficticio  $a_0$ , que representa el soporte sobre el que se va a construir el árbol, cada hormiga  $a_i$  se va a ir fijando paulatinamente al nodo inicial, y sucesivamente a las hormigas ya fijadas, hasta que todas las hormigas están en la estructura. Todos los movimientos y fijaciones en la estructura van a depender del valor de  $Sim(i,j)$ , y de una vecindad en donde se mueven las hormigas (ver Fig. 2).

Tomando como inspiración el AntTree y particularmente el AntSA, un algoritmo derivado de este, Ingaramo et al. proponen el PAntSA (*Partitioned AntSA*) [6], específicamente para el mejoramiento de los resultados de cualquier algoritmo de agrupamiento de textos. En el PAntSA cada hormiga conectada representa un grupo, y las conectadas a esta forman una lista simple.

Así, cuando una hormiga  $a_i$  se incorpora al grupo de una hormiga  $a_+$  (hormiga más similar a  $a_i$ ) denotado  $G_{a_+}$ , la atracción se implementa simplemente adicionando la hormiga al grupo correspondiente. A continuación, se presenta el pseudocódigo del algoritmo (Fig. 3).

A pesar de que Ingaramo et al. probaron la eficiencia del PAntSA [6] y mostraron que es capaz de mejorar a algoritmos de agrupamiento para datos numéricos, hasta nuestro conocimiento no existe un estudio de si el mismo es capaz de mejorar los resultados de algoritmos de agrupamiento para datos mezclados, lo cual es objetivo de este trabajo. Es importante señalar que la complejidad del PAntSA para el peor caso

---

### Algoritmo PAntSA

---

**Entradas:**

$T$ : conjunto de objetos  
 $A$ : algoritmo de agrupamiento  
 $k$ : cantidad de grupos

**Pasos:**

1. Aplicar el algoritmo de agrupamiento  $A$ , a los objetos  $T$ .
2. Construir  $k$  filas (una para cada grupo obtenido en el paso 1) y ordenarlas decrecientemente de acuerdo al índice de la silueta.
3. Crear un grupo para la 1ra hormiga de cada fila, donde dicha hormiga es la representante del grupo,  $R_j$  (la hormiga corresponde al objeto con mayor silueta).
4. Unir las filas en una fila  $F$  tomando iterativamente la 1ra hormiga de cada fila no vacía, hasta que todas las filas estén vacías.  
Para cada hormiga  $a_i$  en  $F$ :
  - a. Para cada hormiga  $R_j$ :
    - i. Hallar el conjunto de hormigas en el grupo de  $R_j$ , denotadas por  $Ga_+$
    - ii. Calcular la atracción de cada hormiga  $att(a_i, Ga_+) = \frac{\sum_{a \in Ga_+} Sim(a_i, a)}{|Ga_+|}$ .Conectar a la hormiga  $a_i$  al grupo con  $att(a_i, Ga_+)$  máximo.
5. Devolver los grupos obtenidos.

---

**Fig. 3.** Pseudocódigo del PAntSA.

(todos los grupos tienen la misma cantidad de objetos) está acotada por  $O(k * (m/k + (m/k)^2))$  donde  $m$  es la cantidad de objetos, y  $k$  es la cantidad de grupos.

## 4. Resultados y discusión

### 4.1. Configuración experimental

#### Bancos de datos bajo estudio

Para evaluar el desempeño del PAntSA en datos mezclados se realizó una comparación experimental mediante algoritmos de agrupamiento de DMI reportados en la literatura, aplicando el PAntSA a los resultados obtenidos por cada uno de ellos.

Para los experimentos se utilizaron 10 bases de datos mezclados e incompletos del repositorio de la Universidad de California en Irvine (UCI) [18] (ver Tabla 1). La selección de este conjunto de datos descansa en que se trata de bases de datos etiquetadas.

Ello permitió contar con agrupamientos modelo (clases) contra los cuales medir la calidad de los grupos obtenidos por los algoritmos mediante dos índices de validación.

**Tabla 1.** Descripción de los bancos de datos utilizados.

No.	Banco de Datos	Atributos Categóricos	Atributos Numéricos	Clases	Valores perdidos	Instancias
1	Autos	10	16	7	si	205
2	Colic	15	7	2	si	368
3	dermatology	1	33	6	si	366
4	heart-c	7	6	5	si	303
5	hepatitis	13	6	2	si	155
6	Labor	6	8	2	si	57
7	lymph	15	3	4	no	148
8	sponge	44	0	3	si	76
9	tae	2	3	3	no	151
10	zoo	16	1	7	no	101

### Índices de validación utilizados

El primer índice utilizado fue la Entropía, la cual mide el grado de desorden del agrupamiento modelo (AM) en los grupos obtenidos mediante la siguiente ecuación:

$$E = \sum_{k \in AE} \frac{|k|}{N} \left[ \frac{1}{\log(|AM|)} \sum_{m \in AM} \frac{n_k^m}{|k|} \log \frac{n_k^m}{|k|} \right], \quad (7)$$

donde: AE representa el agrupamiento a evaluar,  $|k|$  es el total de objetos del grupo  $k$ ,  $|AM|$  es el total de grupos del agrupamiento modelo y  $n_k^m$  es el total de objetos en el grupo  $k$  que pertenece al grupo de AM. Mientras menor sea la Entropía mejor será la calidad del agrupamiento.

El segundo índice de validación utilizado fue el Error del Agrupamiento (*Cluster Error*), otro de los índices de validación internos más empleados en comparaciones experimentales. Su funcionamiento se basa en minimizar los objetos que estén asignados a un grupo diferente del agrupamiento modelo (AM) en el agrupamiento a evaluar (AE) y su definición viene dada por:

$$CE = \sum_{k \in AE} \frac{1}{N} \min_{m \in AM} \{P_k^m\}, \quad (8)$$

donde  $P_k^m$  es el total de objetos en el grupo  $k$  que no pertenecen el grupo  $m$  de AM, mientras menor sea el Error mayor calidad tendrá el agrupamiento.

### Algoritmos de agrupamiento evaluados y sus parámetros

En la comparación experimental se emplearon seis algoritmos que permiten el agrupamiento de DMI y se evaluaron los resultados obtenidos por estos antes y luego de aplicado el PAntSA [6]. Así se seleccionaron los métodos particionales kPrototypes [11] y AD2011 [12], el jerárquico HIMIC [14] y el método CEBMDC [19] el cual es un algoritmo que utiliza una combinación de otros métodos. Además, se utilizaron los metaheurísticos AGKA [15] y BECA [4].

Un aspecto importante en el diseño de los experimentos son los parámetros con los que se van a ejecutar los algoritmos (ver Tabla 2). Cabe señalar que el PAntSA no tiene parámetros más allá de la función de disimilitud entre objetos.

**Tabla 2.** Parámetros utilizados por cada algoritmo.

No.	Algoritmo	Parámetros
1	AD2011	Parámetro de contribución de los grupos: 20
2	AGKA	Número de generaciones: 10 Probabilidad de mutación: 0.05 Tamaño de la población: 10 Probabilidad de cruzamiento: 1
3	BECA	Cantidad de fuentes de alimento: 10 Cantidad de abejas exploradoras: 1 Número de generaciones: 10 Límite de las fuentes de alimento: 10
4	CEBMDC	Umbral de similitud: 0
5	HIMIC	-
6	k-prototypes	-

Como todos los algoritmos requieren conocer la cantidad de grupos a formar, el valor asignado a este parámetro va a coincidir, para cada base de datos, con la cantidad de clases. Con esto se toma las clases como agrupamiento modelo contra el cual evaluar los agrupamientos resultantes de aplicar los algoritmos.

El resto de los parámetros fueron elegidos en base a la existencia de estudios que la recomendaran determinados valores para un mejor desempeño. Además, a los parámetros comunes de los diferentes algoritmos se les suministró el mismo valor. Ello permitió lograr cierta homogeneidad y disminuir un posible desbalance en el desempeño de un algoritmo frente a otro por el empleo de valores diferentes para un mismo parámetro.

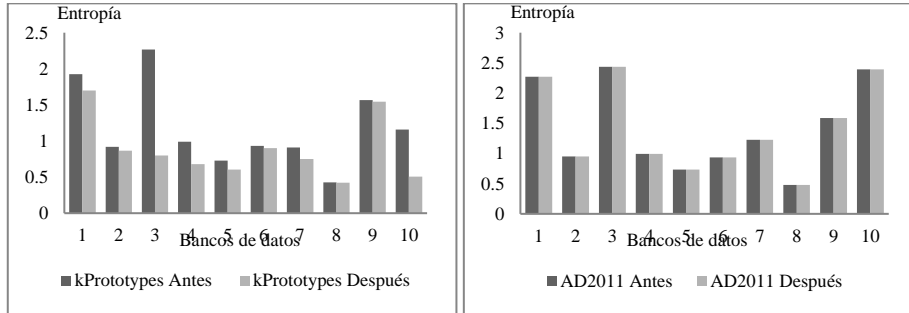
En el caso de la disimilitud se utilizó la función HEOM [17] para todos los algoritmos. La razón de su uso fue sus buenos resultados en el tratamiento de DMI.

## 4.2. Resultados obtenidos

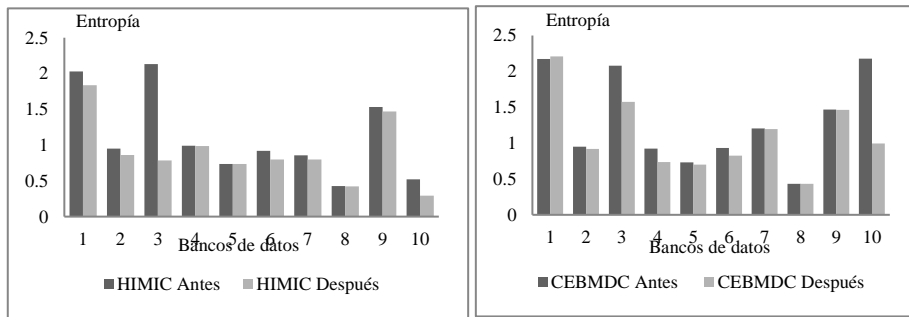
Los experimentos se condujeron de la siguiente forma: Para cada base de datos, se aplicaron los diferentes algoritmos y se calculó la Entropía. Luego se aplicó el PAntSA [6] al resultado de cada algoritmo obteniéndose un nuevo agrupamiento y se calculó igualmente la Entropía a estos nuevos resultados.

En cada caso, se estableció como cantidad de grupos a la cantidad de clases de cada una de las bases de datos. Este procedimiento se repitió para el Error del Agrupamiento. En las figuras 4, 5 y 6 se muestran los resultados de cada algoritmo, antes y después de aplicar el PAntSA. Como puede observarse, en muchos casos la Entropía disminuye.

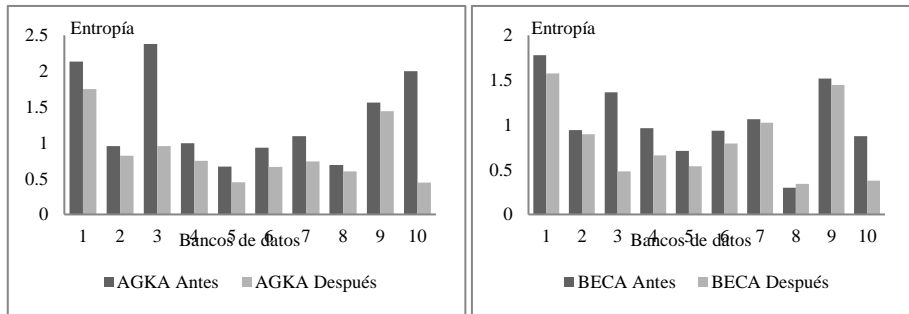
En la mayoría de los casos se obtiene una menor Entropía luego de aplicar el PAntSA. Sin embargo, para establecer si estas diferencias son o no significativas desde el punto de vista estadístico, se aplicó la prueba de Wilcoxon para dos muestras relacionadas. En la Tabla 3 se muestra la significación asintótica bilateral obtenida por la prueba.



**Fig. 4.** Resultados de la Entropía antes y después de aplicado el PAntSA para los algoritmos particionales kPrototypes y AD2011.



**Fig. 5.** Resultados de la Entropía antes y después de aplicado el PAntSA para HIMIC y CEBMDC.



**Fig. 6.** Resultados de la Entropía antes y después de aplicado el PAntSA para los algoritmos metaheurísticos AGKA y BECA.

El test de Wilcoxon permite esclarecer si efectivamente el PAntSA mejora o no los resultados obtenidos por los métodos de agrupamiento. Para ello, se estableció un nivel de confianza del 95%. Así, para valores de probabilidad mayores que 0.05, se considera que no existen diferencias en la calidad de los grupos luego de aplicar el PAntSA. Para

**Tabla 3.** Significación asintótica bilateral de la prueba de Wilcoxon para la Entropía de los agrupamientos obtenidos por los algoritmos antes y después de aplicado el PAntSA.

Algoritmos antes y después de aplicar PAntSA	kPrototypes	AD2011	HIMIC	CEBMDC	AGKA	BECA
Significación asintótica	0.005	1.000	0.008	0.037	0.005	0.009
Decisión	☺	☹	☺	☺	☺	☺

**Tabla 4.** Significación asintótica bilateral de la prueba de Wilcoxon para el Error del Agrupamiento de los resultados obtenidos por los algoritmos antes y después de aplicado el PAntSA.

Algoritmos antes y después de aplicar PAntSA	kPrototypes	AD2011	HIMIC	CEBMDC	AGKA	BECA
Significación asintótica	0.018	0.998	0.028	0.043	0.008	0.018
Decisión	☺	☹	☺	☺	☺	☺

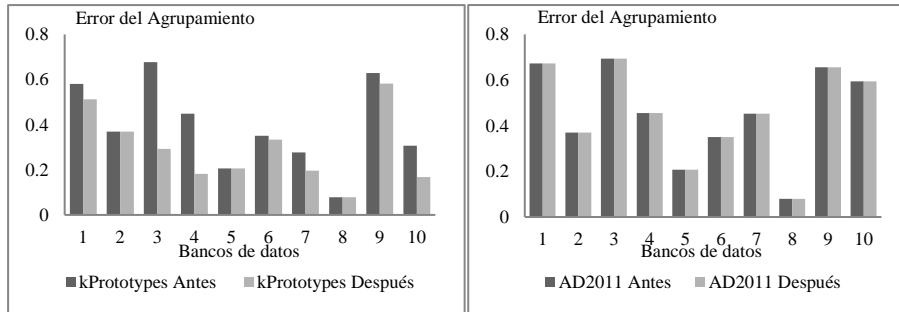
ello se escoge el símbolo ☹ para facilitar la comprensión. Sin embargo, para valores menores a 0.05, es necesario determinar si este método mejora o empeora los resultados obtenidos por el método de agrupamiento, utilizando los símbolos ☺ y ☹, respectivamente.

El símbolo ☺ significa que el PAntSA mejoró significativamente los resultados del algoritmo correspondiente, mientras que el símbolo ☹ significa que no se evidenciaron diferencias entre los resultados del algoritmo antes y después de aplicado el PAntSA. No se encontró evidencias que el PAntSA empeorara los resultados obtenidos. Como se puede apreciar el PAntSA es capaz de mejorar los resultados de los agrupamientos obtenidos por todos los algoritmos a excepción del método AD2011, el cual no evidenció diferencias significativas.

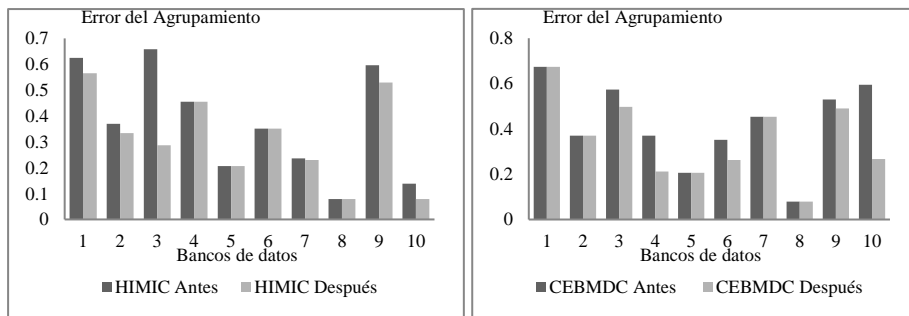
En el caso del Error del Agrupamiento, las figuras 7, 8, y 9 muestran los resultados de los algoritmos comparados antes y después de aplicar el PAntSA.

Como se puede apreciar, al igual que sucedió con la Entropía, en la mayoría de los casos el Error de Agrupamiento disminuye luego de aplicar el PAntSA. Una vez más fue necesario realizar la prueba de Wilcoxon para establecer si existen o no diferencias significativas entre cada algoritmo antes y luego de aplicado el PAntSA. Seguidamente se presentan los resultados de la prueba en la Tabla 4 tomando como nivel de confianza el valor 0.05.

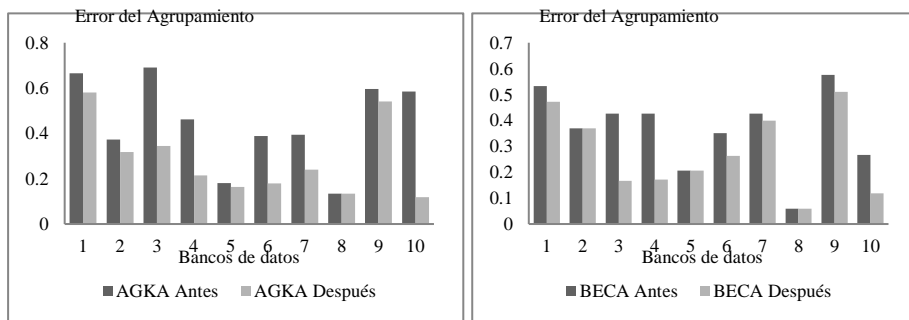




**Fig. 7.** Resultados del Error del Agrupamiento antes y después de aplicado el PAntSA para los algoritmos particionales.



**Fig. 8.** Resultados del Error del Agrupamiento antes y después de aplicado el PAntSA para HIMIC y CEBMDC.



**Fig. 9.** Resultados del Error del Agrupamiento antes y después de aplicado el PAntSA para los algoritmos metaheurísticos AGKA y BECA.

El comportamiento fue similar a lo ocurrido con la Entropía, todos los algoritmos mostraron diferencias significativas favorables a la aplicación del PAntSA, solo AD2011 no evidenció diferencias dado por su significación de 0.998. De forma general

como se pudo comprobar en las experimentaciones analizadas, PAntSA mostró un buen nivel de efectividad en el mejoramiento de los resultados de algoritmos de agrupamiento de DMI. Solamente no se obtuvieron diferencias utilizando el algoritmo AD2011.

## **5. Conclusiones**

La obtención de agrupamientos de elevada calidad en datos mezclados e incompletos reviste especial importancia. El estudio realizado permite aseverar que los resultados obtenidos por métodos de agrupamiento de diversa naturaleza (particionales, jerárquicos, bio-inspirados y otros) pueden ser refinados al aplicar estrategias de post-procesamiento.

El algoritmo PAntSA, en todos los casos mejoró o mantuvo la calidad de los grupos analizados, y en ningún caso su aplicación implicó un detrimento de la misma. Por otra parte, el uso de índices de validación internos, en este caso de la Silueta, abre nuevas líneas de investigación en cuanto a la calidad de los agrupamientos, pues se considera que además de las propiedades de compacidad y separabilidad que tiene en cuenta este índice, otras propiedades pueden ser utilizadas para refinar aún más los agrupamientos de datos mezclados e incompletos.

Las limitaciones de este estudio están dadas por la cantidad de datos y algoritmos estudiados, por lo que en el futuro se pretende realizar un estudio más extenso.

## **Referencias**

1. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
2. Ruiz-Shulcloper, J.: Pattern Recognition with Mixed and Incomplete Data. *Pattern Recognition and Image Analysis*, vol. 18, pp. 563–576 (2008). DOI: 10.1134/S1054661808040044.
3. González-Patiño, D., Villuendas-Rey, Y., Saldaña-Pérez, M., Argüelles-Cruz, A.J.: A Novel Bioinspired Algorithm for Mixed and Incomplete breast Cancer Data Classification. *International Journal of Environmental Research and Public Health*, vol. 20, no. 4, pp. 3240 (2023). DOI: 10.3390/ijerph20043240.
4. Cabrera-Venegas, J.F., Chávez-Castilla, Y.: Clustering Mixed Data using An Artificial Bee Colony. In: II Cuba-Flanders Workshop on Machine Learning and Knowledge Discovery (CFWMLKD'11), Cuba (2011)
5. Errecalde, M., Ingaramo, D., Rosso, P.: A New AntTree-based Algorithm for Clustering Short-Text Corpora. *JCS&T*, vol. 10, no. 1, pp. 1–7 (2010)
6. Ingaramo, D., Errecalde, M., Rosso, P.: A General Bio-Inspired Method to Improve the Short-Text Clustering Task. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 661–672 (2010). DOI: 10.1007/978-3-642-12116-6\_56.
7. Azzag, H., Venturini, G.: A Clustering Model using Artificial Ants. Université François-Rabelais, Tours-France (2004)
8. Azzag, H., Monmarche, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A New Model for Clustering with Artificial Ants. In: CEC'03, IEEE Press, Australia, pp. 2642–2647 (2003). DOI: 10.1109/CEC.2003.1299421.

9. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.R.: Model-based Evaluation of Clustering Validation Measures. *Pattern Recognition*, vol. 40, no. 3, pp. 807–824 (2007). DOI: 10.1016/j.patcog.2006.06.026.
10. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, pp. 281–297 (1967)
11. Huang, Z.: Clustering Large Data Sets with Numeric and Categorical Values. In: *Pacific - Asia Conference on Knowledge Discovery and Data Mining* (1997)
12. Ahmad, A., Dey, L.: A K-Means Type Clustering Algorithm for Subspace Clustering of Mixed Numeric and Categorical Data. *Pattern Recognition Letters*, vol. 32, no. 7, pp. 1062–1069 (2011). DOI: 10.1016/j.patrec.2011.02.017.
13. Reyes-González, R., Ruiz-Shulcloper, J.: Un algoritmo de estructuración restringida de espacios. CIARP, Cuba (1999)
14. Ahmed, R.A., Borah, B., Bhattacharyya, D.K., Kalita, J.K.: HIMIC: A Hierarchical Mixed Type Data Clustering Algorithm. Department of Computer Science and Information (2005)
15. Roy, D.K., Sharma, L.K.: Genetic K-Means Clustering Algorithm for Mixed Numeric and Categorical Datasets. *International Journal of Artificial Intelligence & Applications (IJAIA)*, vol. 1, no. 2, pp. 23–28 (2010). DOI: 10.5121/ijaia.2010.1203.
16. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06. Computer Engineering Department, Erciyes University (2005)
17. Wilson, R.D., Martinez, T.R.: Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, vol. 6, pp. 1–34 (1997). DOI: 10.1613/jair.346.
18. He, Z., Xu, X., Deng, S.: Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach. <http://arix.org/abs/cs/0509011> (2005)



## Mecanismo de generación pseudoaleatorio para el diseño automático de filtros analógicos pasivos

Braulio Jesús Montoya Padilla<sup>1</sup>, Aurora Torres Soto<sup>1</sup>,  
María Dolores Torres Soto<sup>2</sup>

<sup>1</sup> Universidad Autónoma de Aguascalientes,  
Departamento de Ciencias de la Computación,  
México

<sup>2</sup> Universidad Autónoma de Aguascalientes,  
Departamento de Sistemas de Información,  
México

al188428@edu.uaa.mx, {atorres, mdtorres}@correo.uaa.mx

**Resumen.** Este trabajo presenta un innovador mecanismo de generación pseudoaleatorio desarrollado específicamente para el diseño automático de filtros analógicos pasivos, el cual tiene como principal ventaja disminuir el costo computacional asociado con la simulación de los circuitos generados aleatoriamente. Ya que el mecanismo garantiza que cualquier circuito generado se va a poder simular, eliminando la necesidad de descartar diseños no viables. El mecanismo propuesto utiliza una serie de reglas y restricciones cuidadosamente definidas que guían el proceso de generación, asegurando que los componentes y las conexiones resultantes cumplan con los requisitos básicos de un filtro analógico pasivo funcional. Para demostrar la eficiencia del mecanismo, se incluye un ejemplo detallado en el que se analiza el comportamiento del circuito y observa que el circuito generado tiene el comportamiento esperado por un filtro, validando así la efectividad del método de generación pseudoaleatoria. Este mecanismo tiene el potencial de acelerar significativamente el proceso de diseño y optimización de filtros analógicos pasivos en diversas aplicaciones de ingeniería eléctrica.

**Palabras clave:** Generación de circuitos, diseño automático, circuitos analógicos, filtros.

### Pseudorandom Generation Mechanism for the Automatic Design of Passive Analog Filters

**Abstract.** This work presents an innovative pseudo-random generation mechanism developed specifically for the automatic design of passive analog filters, which has the main advantage of reducing the computational cost associated with the simulation of randomly generated circuits. The mechanism guarantees that any generated circuit can be simulated, eliminating the need to discard unfeasible designs. The proposed mechanism uses a set of carefully

defined rules and constraints to guide the generation process, ensuring that the resulting components and connections meet the basic requirements of a functional passive analog filter. To demonstrate the efficiency of the mechanism, a detailed example is included in which the behavior of the circuit is analyzed, and it is observed that the generated circuit has the expected behavior for a filter, thus validating the effectiveness of the pseudo-random generation method. This mechanism has the potential to significantly accelerate the design and optimization process of passive analog filters in various electrical engineering applications.

**Keywords:** Circuit generation, automatic design, analog circuits, filters.

## 1. Introducción

El diseño de circuitos es una tarea que se realiza partiendo de circuitos simples que posteriormente se les añaden ciertas características, se encuentran fallas y se corrigen errores hasta encontrar el funcionamiento deseado. Por esto los diseñadores de circuitos necesitan de su experiencia, su conocimiento y su habilidad para diseñar circuitos que cumplan un funcionamiento deseado [1].

Claramente esta tarea llega a consumir mucho tiempo, dinero y esfuerzo, por lo que existe gran esfuerzo para automatizar el diseño de circuitos eléctricos para disminuir los costos de producción. Pero esta también es una tarea que requiere de muchos esfuerzos, ya que los algoritmos deben contener la experiencia y los conocimientos de los diseñadores [2].

En el presente artículo se propone un mecanismo de generación pseudoaleatorio de filtros analógicos pasivos. Se dice pseudoaleatorio ya que los circuitos no se construyen considerando todas las combinaciones posibles, sino que parten de topologías que garantizan que el circuito generado va a simular correctamente. Debido a que la simulación ayuda a evaluar el funcionamiento del circuito, y es esta una tarea que consume más recursos computacionales.

Este mecanismo de generación pseudoaleatorio es el primer paso en el diseño automático de filtros analógicos pasivos. Ya que este puede ser utilizado para generar soluciones iniciales, las cuales pueden ser tomadas por algoritmos de optimización para mejorar los circuitos y aproximarlos a un filtro ideal.

## 2. Filtro analógico pasivo

En general un filtro se describe como un dispositivo que de un modo específico puede modificar una señal. En otras palabras, estos dispositivos permiten el paso de una señal en ciertas bandas de frecuencia y la bloquean en otras [3]. Y éstos se pueden clasificar según el tipo de señal que procesan:

- **Analógicos:** Los filtros analógicos son diseñados para tratar señales analógicas, y durante el procesamiento de la señal esta puede tomar cualquier valor en un intervalo de tiempo.

- **Digitales:** Los filtros digitales son diseñados para tratar señales digitales, y durante el procesamiento de la señal esta solo puede tomar valores discretos en un intervalo de tiempo.

O bien, se pueden clasificar por su ganancia:

- **Activos:** Un filtro activo tiene el mismo propósito que un filtro pasivo, con la diferencia de que en su construcción se suelen usar tantos elementos pasivos como elementos activos, por eso estos filtros suelen contar con amplificadores operacionales, lo que hace que sean de mayor factor de calidad.
- **Pasivos:** Un filtro pasivo está conformado por componentes pasivos, como lo son las resistencias, conductancias e inductancias; estos dos últimos son los elementos reactivos del filtro, es decir, son elementos que hacen que cambie la fase de la corriente que pasa por ellos. El número de componentes reactivos determina el orden del filtro.

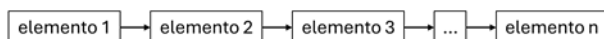
La función de un filtro analógico pasivo consiste en alterar una señal de entrada para obtener sus propiedades aprovechables en ciertas circunstancias, eliminando las propiedades que se quieren ignorar. Estos filtros son usados para seleccionar un conjunto determinado de frecuencias de una señal para amplificarlas, atenuando el conjunto excluido, siendo de especial utilidad en el mundo del audio y las comunicaciones [4].

El principio de funcionamiento de un filtro pasivo se basa en que la impedancia de algunos elementos es variable con la frecuencia, de manera que cuando se disponen de la manera apropiada algunas frecuencias permanecen inalteradas, mientras que otras sufren distintos grados de atenuación. Los elementos de impedancia variable son la capacitancia y la inductancia. Estos pueden emplearse con una resistencia formando filtros de primer orden.

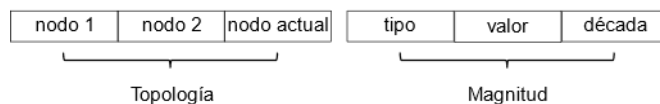
## **2.1. Aplicaciones de los filtros**

A continuación, se describen algunas de las aplicaciones de los filtros analógicos pasivos [5].

- Los filtros del tipo pasa bajas son utilizados para proteger componentes electrónicos de repentinos picos de frecuencia.
- Los filtros del tipo pasa altas son utilizados para reducir el desplazamiento entre etapas de amplificación de una señal.
- Los filtros del tipo pasa bandas son utilizados en dispositivos de medición de comunicación y de procesamiento de señales.
- Los filtros del tipo rechaza bandas son utilizados para eliminar los rangos de frecuencias que generan ruido en una señal, en otras palabras eliminan las interferencias de un señal.



**Fig. 1.** Codificación de un circuito mediante una lista.



**Fig. 2.** Codificación de un elemento del circuito.

**Tabla 1.** Nodos reservados.

Conexión	Nodo
Inicio	1
Final	1002
Tierra	0

**Tabla 2.** Series de valores comerciales.

Valor	E12	E6	E3
0	10	10	10
1	12	15	22
2	15	22	47
3	18	33	
4	22	47	
5	27	68	
6	33		
7	39		
8	47		
9	56		
10	68		
11	82		

**Tabla 3.** Codificación de la magnitud para cada elemento.

Elemento	Tipo	Valor	Década
Capacitancia	0	E6 (0 – 5)	$10^{-5} - 10^{-9}$ (0 – 4)
Resistencia	1	E12 (0 – 11)	$10^1 - 10^6$ (0 – 5)
Inductancia	2	E12 (0 – 11)	$10^{-1} - 10^{-6}$ (0 – 5)

### 3. Representación de circuitos

A fin de generar circuitos analógicos pasivos automáticamente, se debe establecer como se van a representar en la computadora. Existen distintas formas de representar un circuito, pero la mayoría de las representaciones coinciden en que una buena forma de representar un circuito en la computadora es mediante una lista encadenada.



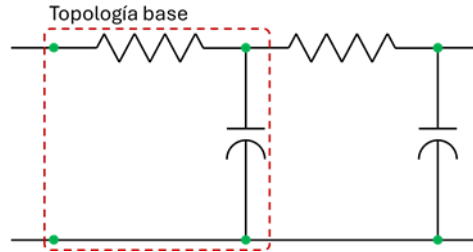


Fig. 3. Ejemplo de circuito con una topología base que se replica dos veces.

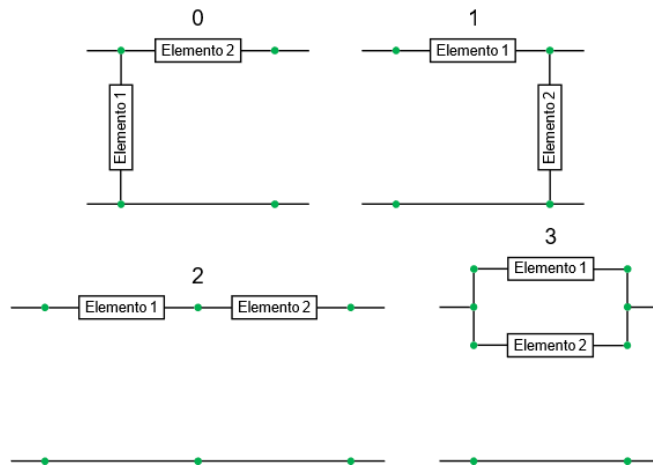


Fig. 4. Topologías base para circuitos con dos elementos.

Tabla 4. Disposición de elementos para circuitos con dos elementos diferentes.

Disposición	Elemento 1	Elemento 2
0	Capacitancia	Resistencia
1	Capacitancia	Inductancia
2	Resistencia	Capacitancia
3	Resistencia	Inductancia
4	Inductancia	Capacitancia
5	Inductancia	Resistencia

Tal como lo proponen los autores en el artículo Mecanismo de representación para la evolución automática de circuitos analógicos [6].

El circuito se codifica como una lista de n elementos, es decir, cada componente de la lista contiene la información del n-ésimo elemento, como se muestra en la Fig. 1.

Mientras que cada componente de la lista (o elemento del circuito) se codifica considerando su topología y su magnitud. En donde la topología es la información sobre la conexión del elemento mediante un nodo 1, un nodo 2 y un nodo actual, y la magnitud

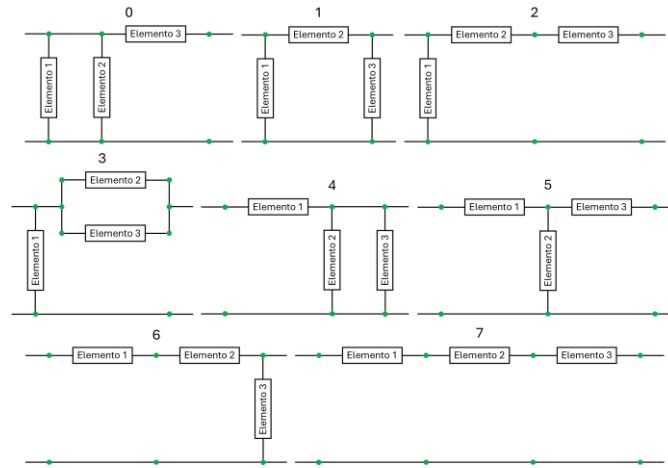


Fig. 5. Topologías base para circuitos con tres elementos (Parte 1).

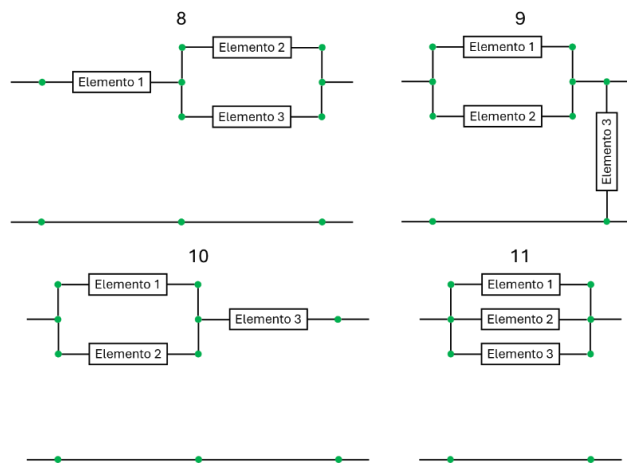


Fig. 6. Topologías base para circuitos con tres elementos (Parte 2).

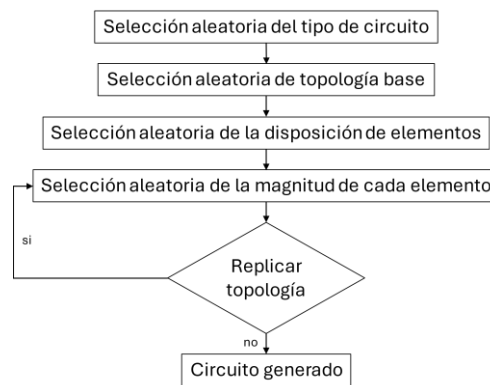
Tabla 5. Disposición de elementos para circuitos con tres elementos diferentes.

Disposición	Elemento 1	Elemento 2	Elemento 3
0	Capacitancia	Resistencia	Inductancia
1	Capacitancia	Inductancia	Resistencia
2	Resistencia	Capacitancia	Inductancia
3	Resistencia	Inductancia	Capacitancia
4	Inductancia	Capacitancia	Resistencia
5	Inductancia	Resistencia	Capacitancia

está compuesta del tipo de elemento, un valor y un multiplicador del valor. Como se observa en la Fig. 2.

**Tabla 6.** Disposición de elementos para circuitos con dos elementos iguales y uno diferente.

Disposición	Elemento 1	Elemento 2	Elemento 3
0	Capacitancia	Resistencia	Capacitancia
1	Capacitancia	Inductancia	Capacitancia
2	Resistencia	Capacitancia	Resistencia
3	Resistencia	Inductancia	Resistencia
4	Inductancia	Capacitancia	Inductancia
5	Inductancia	Resistencia	Inductancia



**Fig. 7.** Diagrama del mecanismo de generación pseudo aleatorio de circuito.

**Tabla 7.** Disposición de elementos para cada topología base de circuitos con dos elementos diferentes.

Topología base	Disposición de elementos
0	0 – 5
1	0 – 5
2	0, 1, 3
3	0, 1, 3

Cabe indicar que dentro de la topología existen nodos reservados que sirven para identificar el inicio y el final del circuito, así como un componente conectado a tierra. Estos nodos reservados se muestran en la Tabla 1.

Para codificar la magnitud del elemento, se considera para el valor una serie de valores comerciales (véase la Tabla 2) y para la década una serie de valores que determina si se trata de cantidades en el orden nano, micro, mili, kilo, etc. Según estas consideraciones un elemento se puede codificar en base a la Tabla 3 [7].

#### 4. Mecanismo de generación pseudoaleatorio

El mecanismo para la generación automática de filtros analógicos se presenta a continuación. Este mecanismo garantiza que los circuitos generados se puedan simular

correctamente y además no contengan demasiados elementos. Ya que se han estudiado distintas topologías, que sin importar los elementos que contengan, no presentan problemas en la simulación. Por lo que se obtienen soluciones con un buen desempeño, lo que ayuda a que un algoritmo encuentre buenas soluciones en pocas generaciones [8]. El mecanismo parte de la idea de que los circuitos están formados por topologías base que se van replicando, como se muestra en la Fig. 3 [9]. De esta forma se evita generar circuitos abiertos o con elementos que estén conectados en corto.

Una topología base está formada por dos o tres elementos que forman un circuito en serie, en paralelo o mixto. En donde, según el tipo de elementos, se consideran tres tipos de circuitos según su topología base:

- 1 Circuitos con dos elementos diferentes.
- 2 Circuitos con tres elementos diferentes.
- 3 Circuitos con dos elementos iguales y uno diferente.

Para los circuitos con dos elementos diferentes se contemplan las topologías mostradas en la Fig. 4, en donde sus elementos toman una disposición en base a la Tabla 4. Para los otros dos tipos de circuitos se contemplan las topologías base mostradas en la Fig. 5 y 6. En donde los circuitos con tres elementos diferentes sus elementos toman una disposición en base a la Tabla 5, mientras que los circuitos con dos elementos iguales y uno diferente toma una disposición en base a la Tabla 6.

Una vez que se ha seleccionado el tipo de circuito, la topología base y la disposición de los elementos que contendrá, se le asigna una magnitud (según la codificación de la Tabla 3) a cada elemento. Para finalmente decidir las veces que se va a replicar el circuito, en donde los elementos pueden tener la misma magnitud o cualquier otra.

Es importante mencionar que los circuitos con dos elementos iguales y uno diferente, presenta algunas condiciones para los casos en que la disposición de los elementos es 4 o 5, ya que en estos casos no se puede replicar cuando la topología base es 0, 1, 3, 4 o 9 y no se puede seleccionar la 11. Por lo que el mecanismo considera estas condiciones al momento de generar un circuito.

Ya que, bajo estas condiciones, estos circuitos no simulan porque las inductancias generan corto en las conexiones en paralelo.

En general, el mecanismo de generación funciona en base al diagrama mostrado en la Fig. 7.

### **Disminución de la disposición de los elementos**

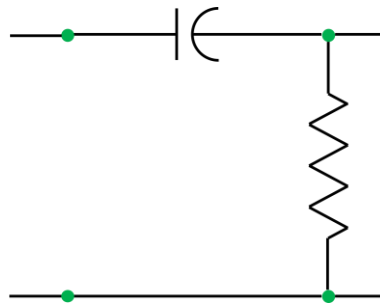
En los circuitos en serie o en paralelo la disposición de los elementos tiene equivalencias. Ya que, en los circuitos en serie, el voltaje es la suma del voltaje de todos los elementos, y en los circuitos en paralelo, el voltaje permanece constante [10].

Por ejemplo, si se va a un construir un circuito con dos elementos diferentes y se selecciona la topología base 2, en donde los dos elementos se encuentran conectados en serie (véase Fig. 4). Y se selecciona la disposición de elementos 0, es decir, una capacitancia seguido de una resistencia, es equivalente a seleccionar la disposición de elementos 2, una resistencia seguido de una capacitancia.

Por lo que se puede disminuir las distintas disposiciones de elementos en las topologías base en serie o en paralelo. Para el caso de los circuitos con dos elementos

**Tabla 8.** Disposición de elementos para cada topología base de circuitos con tres elementos diferentes.

Topología base	Disposición de elementos
0	0, 1, 3
1	0 – 5
2	0, 2, 4
3	0, 2, 4
4	0, 2, 4
5	0 – 5
6	0, 1, 3
7	0
8	0, 2, 4
9	0, 1, 3
10	0, 1, 3
11	0



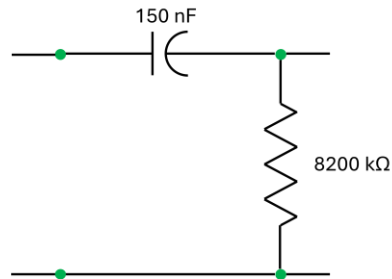
**Fig. 8.** Circuito generado hasta el paso tres del mecanismo.

diferentes, a las diferentes topologías base le corresponde la disposición de elementos mostradas en la Tabla 7.

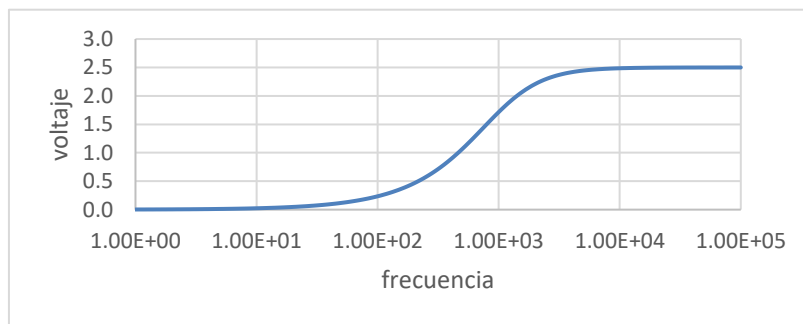
Para los circuitos con tres elementos diferentes, les corresponde la disposición de elementos, mostradas en la Tabla 8. En el caso de los circuitos con dos elementos iguales y uno diferente, se puede seleccionar cualquiera de las 6 diferentes disposiciones de elementos, ya que no se cuentan con disposición equivalentes cuando existen elementos conectados en serie o en paralelo.

### Ejemplo del funcionamiento del mecanismo

Siguiendo el diagrama del mecanismo (véase Fig. 7), como primer paso se selecciona el tipo de circuito de forma aleatoria, en este caso se selecciona el tipo de circuito 1 (circuito con dos elementos diferentes). Después se selecciona una topología base, correspondiente al tipo de circuito (véase Fig. 4), en este caso se selecciona la topología base 1.



**Fig. 9.** Circuito generado al asignarle una magnitud a cada elemento.



**Fig. 10.** Gráfico de voltaje contra frecuencia del circuito generado.

Como tercer paso, selecciona aleatoriamente la disposición de los elementos según el tipo de circuito (véase Tabla 3), en este se selecciona la disposición 0. Hasta el momento se tiene el circuito mostrado en la Fig. 8.

Posteriormente se selecciona aleatoriamente una magnitud para cada elemento, en base a la Tabla 3. En este caso, se le asigna una magnitud de  $15 \times 10^{-8} \text{ F}$  o 150 nF a la capacitancia y una de  $82 \times 10^5 \text{ } \Omega$  o 8200 kΩ a la resistencia. Entonces se tiene el circuito mostrado en la Fig. 9.

Finalmente se decide si el circuito se va a replicar, para este no será el caso. Por lo que el circuito final es el mostrado en la Fig. 9. Al simular el circuito y analizar su voltaje de salida, se ha obtenido el grafico de voltaje contra frecuencia mostrado en la Fig. 10. Se puede observar que el comportamiento esperado de un filtro pasa altas, ya que esta atenuando las frecuencias bajas.

Se observa que el mecanismo genera circuitos con buenas aproximaciones a filtros, lo que ayuda a que un algoritmo de diseño automático de circuitos inicie con buenas soluciones. Cabe resaltar que el circuito de ejemplo fue generado aleatoriamente usando el mecanismo propuesto, y no se ha sometido a un algoritmo de optimización de parámetros.

## 5. Conclusiones

El mecanismo de generación propuesto garantiza que los circuitos generados se pueden simular, ya que éstos se generan en base a topologías que evitan la generación de circuitos abiertos o con conexiones en corto. Esto ayuda a que un algoritmo de diseño

automático de circuitos inicie con soluciones factibles para obtener el mejor rendimiento.

Adicionalmente, este mecanismo de generación puede ser usado en algoritmos de optimización para encontrar la mejor combinación de valores de cada elemento del circuito. O bien, para encontrar la mejor topología con los mejores valores, según el tipo de filtro que se quiere construir.

Al haber probar el mecanismo en un algoritmo genético, se observó que este obtiene mejores resultados cuando se generan filtros pasa bajas y pasa altas, a comparación de filtros pasa bandas y rechaza bandas. Por lo que se está trabajando en una mejora del mecanismo generación para obtener mejores filtros pasa bandas y rechaza bandas.

Cabe mencionar, que también se está utilizando este mecanismo con otras metaheurísticas para analizar su comportamiento en cada una de estas. De forma que se pueda encontrar la mejor metaheurística para este mecanismo.

## **Referencias**

1. Brito, M.A., Giraldo, J.S.: Metodologías para diseño de circuitos ladder con base en sistemas secuenciales y combinatoriales. Universidad Tecnológica de Pereira (2011)
2. Torres, J.C.: Diseño asistido por ordenador. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, vol. 20 (1998)
3. Haver, J.J.: Análisis comparativo de filtros activos y filtros pasivos en sistemas de comunicación analógica. Biblioteca Virtual SEMISUD (2018)
4. Alvarez, F.O.: Estudio y programas digitales para el diseño de filtros pasivos analógicos. Escuela Politécnica Nacional (1984)
5. López D.A.: Historia, definición, descripción, tipos y aplicaciones de filtros electrónicos (2003)
6. Soto, A., Soto, M.A., Ponce, E.E.: Mecanismo de representación para la evolución automática de circuitos analógicos. *Research in Computing Science*, vol. 149, no. 8, pp. 253–265 (2020)
7. Chouza, M., Rancan, C., Clúa, O., García, R.: Diseño de filtros analógicos pasivos basados en programación genética. En: XIV Congreso Argentino de Ciencias de la Computación, (2008)
8. Soto, A., Soto, M.A., Ponce, E.E., and Montoya, B.J.: Diseño automático de filtros analógicos pasivos mediante un algoritmo genético. *Research in Computing Science*, vol. 151, no. 5, pp. 75–90 (2022)
9. Miyara, F.: Filtros activos, Cátedra de Electrónica III, FCEIA-UNR (2004)
10. Pozuelo, J., Cascarosa, E.: Intensidad, Resistencia, Circuitos en Serie y en Paralelo. ¿Quién es quién? Una actividad práctica para trabajar los circuitos en 2o de ESO (2022)





## Modelo de aprendizaje profundo para la inspección de tarjetas de circuitos impresos

José Antonio Lara Chávez, Carlos Avilés Cruz, Miguel Magos Rivera

Universidad Autónoma Metropolitana, Unidad Azcapotzalco,  
México

{jalch, mrm, caviles}@azc.uam.mx

**Resumen.** En este artículo, se presenta el desarrollo de un modelo específico de aprendizaje profundo para la detección de errores en el montaje de componentes electrónicos sobre una tarjeta de circuito impreso. El diseño de este modelo se compone por dos etapas: extracción de atributos y clasificación. La primera de estas se basa en redes neuronales convolucionales, utilizadas bajo un enfoque inspirado en la visión humana: primero se perciben estructuras gruesas, luego medianas y finalmente detalles finos. Por su parte, la clasificación se lleva a cabo mediante una red neuronal totalmente conectada. Para el entrenamiento y evaluación correspondiente, fue construida una base de datos propia, compuesta por 2376 imágenes que corresponden a seis tipos de tablillas. Finalmente, el modelo fue implementado en un equipo de cómputo, el cual recibe una imagen digital de la placa de circuito impreso que se requiere inspeccionar para, en tiempo real, indicar si presenta algún tipo de defecto.

**Palabras Clave:** Aprendizaje profundo, inspección, tarjeta de circuito impreso.

### Deep Learning Model for Printed Circuit Board Inspection

**Abstract.** In this article, the development of a specific deep learning model for detecting errors in the assembly of electronic components on a printed circuit board is presented. The design of this model consists of two stages: feature extraction and classification. The first of these is based on convolutional neural networks, used under an approach inspired by human vision: first, coarse structures are perceived, then medium ones, and finally fine details. The classification is carried out using a fully connected neural network. For the corresponding training and evaluation, a proprietary database was constructed, consisting of 2376 images corresponding to six types of PCBs. Finally, the model was implemented on a computing device, which receives a digital image of the PCB to be inspected and, in real time, indicates if it presents any type of defect.

**Keywords:** Deep learning, inspection, printed circuit boards.

## **1. Introducción**

La detección de defectos en productos ensamblados y/o empacados, es una tarea importante en muchos procesos industriales. Su realización en forma manual la convierte en una actividad con un alto grado de incertidumbre, además de complicada y tediosa para el personal encargado [1]. Lo anterior puede ocasionar un impacto negativo, tanto en la calidad de los productos, como en los costos de producción, lo que podría traducirse en pérdidas significativas para las empresas.

Por tal motivo, desde hace algunos años se han reportado diversos trabajos en los cuales se realizan inspecciones automáticas, utilizando generalmente, visión por computadora [2]. Las primeras metodologías empleadas en este tipo de tareas se basan principalmente en el procesamiento digital de imágenes, complementadas en ocasiones con técnicas de optimización, algoritmos genéticos, etc. [3].

La principal desventaja de este tipo de procedimientos, radica en la dependencia frente a condiciones externas, tales como: iluminación, posición de la cámara, enfoque, color de fondo, etc. [4].

Por su parte, estas deficiencias suelen no presentarse cuando la inspección visual automática se realiza mediante modelos computacionales creados a partir de aprendizaje automático y aprendizaje profundo [5].

Por tal motivo, en el último quinquenio se han estudiado diferentes métodos basados en este enfoque para realizar inspecciones autónomas [6]. Las máquinas de soporte vectorial, redes neuronales, redes neuronales convolucionales y redes neuronales recurrentes, han sido aplicadas en control de calidad a nivel industrial. En este artículo, se presenta el desarrollo de un modelo específico de aprendizaje profundo para la detección y clasificación de errores en el montaje de componentes electrónicos sobre una tarjeta de circuito impreso (PCB, por sus siglas en inglés).

Este modelo se divide en dos principales etapas: extracción de atributos y clasificación. La primera de estas, se basa en redes neuronales convolucionales (CNN, por sus siglas en inglés), utilizadas bajo el esquema grueso-mediano-fino [7]. Este enfoque está inspirado en la visión humana: primero se perciben estructuras gruesas, luego medianas y finalmente detalles finos. La etapa de clasificación, se llevó a cabo mediante una red neuronal totalmente conectada.

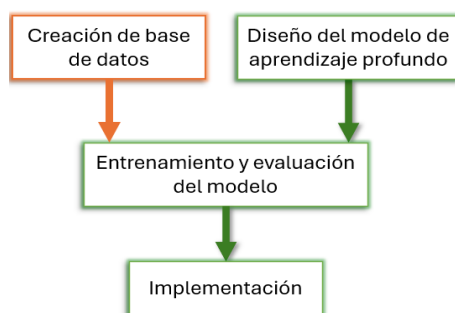
Posterior al diseño de dicho modelo, éste se entrenó y evaluó a partir de una base de datos propia, compuesta por 2376 imágenes. Una de estas placas no tiene ningún defecto, mientras que el resto presenta uno de cinco distintos errores de montaje, ver Tabla 1.

El modelo desarrollado, se implementó en un equipo de cómputo, al cual se le envía una imagen digital de la tablilla de circuito impreso para detectar si presenta defectos. Un diagrama de flujo de la metodología seguida se muestra en la Figura 1.

En la sección dos del artículo, se describen los trabajos relacionados con el mismo. Posteriormente se abordan la construcción de la base de datos utilizada y el diseño del modelo. Enseguida se detalla el entrenamiento y evaluación, para después describir su implementación. El artículo finaliza con las conclusiones y la bibliografía del mismo.

**Tabla 1.** Tipos de tarjetas y sus errores correspondientes.

Tipo de tarjeta	Defecto
Tarjeta correctamente ensamblada	No presenta defecto
Tarjeta con error A	Circuito integrado ausente
Tarjeta con error B	Circuito integrado montado en sentido opuesto
Tarjeta con error C	Diodo con la polaridad invertida
Tarjeta con error D	Circuito integrado en lugar erróneo
Tarjeta con error E	Resistor con valor erróneo



**Fig. 1.** Metodología utilizada.

## 2. Estado del arte

En la mayoría de los casos, los modelos de aprendizaje profundo empleados para inspecciones automáticas se componen por redes neuronales convolucionales [8]. Estos pueden generarse, modificando alguno ya existente, o bien, mediante la creación de uno específico para un trabajo en particular [9]. En Kulkarni et al. [10], utilizaron y evaluaron por separado, algoritmos de seguimiento de línea y agrupamiento, algoritmos de detección de objetos y redes neuronales convolucionales, para encontrar errores de fabricación en tapas de botellas de plástico. Un trabajo similar es presentado por Rajan et al. [11], quienes desarrollaron un sistema de identificación y clasificación de pernos dañados, utilizando redes neuronales convolucionales y una Raspberry Pi4. En Tran et al. [12], emplearon YOLO v4, como parte fundamental de un sistema de inspección física de redes eléctricas aéreas de alto voltaje. En la industria farmacéutica, Ficzer et al. [13], mediante ajustes a la versión 5 de YOLO, en combinación con técnicas de procesamiento de imágenes, desarrollaron un sistema identificador y clasificador de imperfectos en la superficie de píldoras médicas.

Las empresas de fabricación y ensamble de tarjetas de circuito impreso, son un ejemplo de área industrial en la cual la inspección manual resulta ser bastante complicada [14]. Lo anterior se debe a la gran cantidad, y tamaño reducido de los elementos que éstas contienen. Los tipos de errores que una PCB presentan, se pueden clasificar como: defectos superficiales, defectos en soldaduras y defectos de montaje.

Se consideran defectos superficiales a aquellos errores en las pistas de los circuitos impresos, tales como: cortocircuitos y circuitos abiertos, por mencionar solo algunos.

Un método de inspección automática para localizar este tipo de imperfecciones, es presentado en Kim et al. [15], en donde se desarrolló un modelo de aprendizaje profundo al que los autores llamaron: “Skip-Connected Autoencoder”. Este se basa en una CNN que codifica y decodifica una imagen de la placa a inspeccionar, posteriormente, el resultado se resta a la de una imagen sin defecto, para así dar con el error.

De forma similar, Park et al. [16] proponen un método para detectar desperfectos en la superficie de una PCB a través de un Auto-Encoder convolucional variable, en combinación con un modelo de aprendizaje profundo. Para este último, se implementaron, evaluaron y compararon resultados, utilizando por separado los modelos: ResNet 101, EfficientNet\_b3 y DenseNet 121.

Así mismo, Austria et al [17], lograron encontrar y clasificar 10 tipos distintos de defectos superficiales, utilizando el modelo ResNet-101, como extractor de características, y Mask R-CNN como delimitador semántico.

Los defectos en soldaduras, están asociados, entre otros, a escasa cantidad de estaño o exceso de la misma en las terminales de los componentes, o a cortocircuitos generados por una mala técnica de soldadura. Kim et al. [18], desarrollaron un modelo de aprendizaje profundo al que denominaron “Dual-Stream Convolutional Networks”, el cual se caracteriza por tener dos CNN que trabajan en paralelo y que después se fusionan. Ambas redes reciben imágenes seccionadas de la tablilla a inspeccionar, detectando en cuál de éstas se presenta un error.

De forma semejante, Sezer et al. [19] presentan un método, en el cual, mediante una red convolucional 2D diseñada para distintas clases, se ubican elementos mal soldados. Esta red fue entrenada en tres ocasiones, utilizando respectivamente, las funciones de optimización: Adam, AdaMax y RMSprop, posteriormente, los resultados obtenidos en cada caso, fueron analizados para determinar la mejor función.

Por último, los defectos de montaje corresponden a un incorrecto ensamblaje de algún componente electrónico en una PCB, por ejemplo: elementos faltantes, mal ubicados o con polaridad invertida, entre otros. Debido a la dificultad que lo anterior representa, y con el objetivo de dar solución a este tipo de problemática, diversos autores han trabajado en metodologías enfocadas en reconocer, únicamente, los dispositivos presentes en estas placas.

En Lian et al. [20], mediante una modificación al modelo Mask R-CNN, se desarrolló un método para identificar, por su forma geométrica, piezas de montaje superficial. A su vez, con el objetivo de localizar la totalidad de circuitos integrados montados sobre una PCB, Gang et al. [21], implementaron y compararon los modelos: ResNet 56 y EfficientNet, para detectar y clasificar caracteres en microchips.

Por su parte, Zhang [22] mediante procesamiento de imágenes digitales, algoritmos de corrección de posicionamiento y YOLO, desarrolló un sistema capaz de distinguir y ubicar componentes presentes, ausentes o mal colocados en una placa de circuito impreso.

En Jia et al. [23], se presenta el desarrollo de un modelo de aprendizaje profundo al que llamaron “ICChaNet”, el cual es capaz de detectar los caracteres presentes en los circuitos integrados, y así determinar si éstos han sido colocados correctamente. Este modelo se basa en una red neuronal convolucional profunda, con una estructura LeNet-5 mejorada.

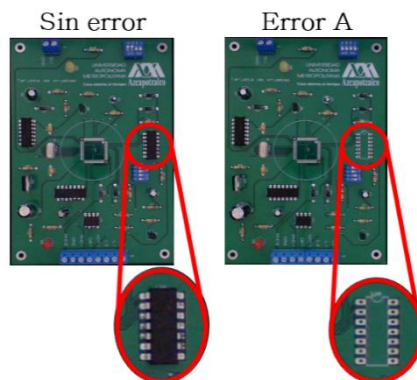


Fig. 2. Ejemplo de error en tarjeta de circuito impreso.

### 3. Base de datos

Para el entrenamiento y evaluación del modelo de aprendizaje profundo que se describe en este artículo, se creó una base de datos propia; se trata de un conjunto de imágenes digitales que corresponden a seis tipos de tarjetas de circuito impreso, ver Tabla 1. Estas placas miden 15 cm de largo por 10 de ancho, fueron fabricadas con fibra de vidrio, tienen una sola capa de cobre y no contienen componentes electrónicos de montaje superficial. La Figura 2, muestra una comparación entre la tablilla sin defecto y otra en la cual se tiene la ausencia de un circuito integrado. Cabe mencionar que, algunos errores presentes en estas tarjetas son relativamente fáciles de detectar, mientras que para otros la dificultad aumenta considerablemente. Por lo que, para aumentar la precisión en la clasificación de defectos para todas las clases, se decidió emplear una red neuronal profunda.

Inicialmente, se capturaron 36 fotografías con una dimensión de 1024 x 1280 píxeles, en formato de color RGB y una rotación de 10° entre ellas, lo anterior, para cada uno de los seis tipos de tarjetas fabricadas. Posteriormente, se generaron 10 nuevas muestras para cada una de las 36 imágenes de una clase de tablilla. Lo anterior, mediante la técnica de aumento de datos por variación de brillo, en un rango de 20% a 99%. De esta forma, se obtienen 396 imágenes por tipo de tarjeta, dando un total de 2376, las cuales, para facilitar el entrenamiento del modelo, fueron redimensionadas a 682 x 853 píxeles. Finalmente, la base de datos fue dividida aleatoriamente en: 80% para entrenar y 20% para evaluar.

### 4. Diseño del modelo de aprendizaje profundo

El diseño del modelo de aprendizaje profundo reportado en este artículo, se compone de dos partes: extracción de atributos y clasificación. La primera, consta de tres redes neuronales convolucionales, que trabajan bajo el esquema Grueso-Mediano-Fino [7]. Por otro lado, en la clasificación, se analizan los atributos extraídos, mediante una red neuronal totalmente conectada, para de esta forma obtener un resultado, ver Figura 3. Las características específicas del diseño de las dos partes que componen a la red

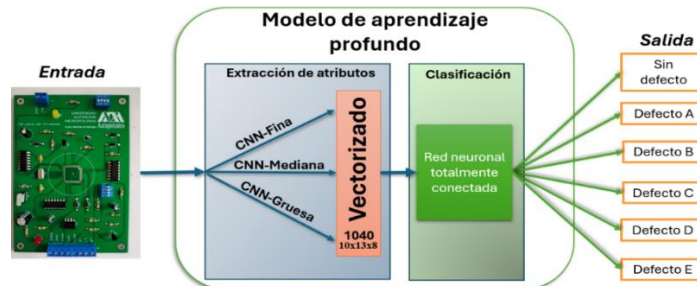


Fig. 3. Modelo de aprendizaje profundo.

neuronal profunda presentada en este artículo, son descritas en las siguientes subsecciones.

#### 4.1. Extracción de atributos

La etapa de extracción de atributos se compone, por tres redes neuronales convolucionales, bajo el esquema Grueso-Mediano-Fino [7]. Este enfoque está inspirado en la visión humana: primero se perciben estructuras gruesas, luego medianas y finalmente detalles finos. Estas tres redes se encuentran conectadas en paralelo y se les llamó: CNN-Fina, CNN-Mediana y CNN-Gruesa.

**CNN-Fina.** Esta red cuenta con seis capas convolucionales 2D para la extracción de atributos. La primera de estas, también llamada de entrada, recibe una imagen de 682 x 853 píxeles en formato RGB y está compuesta por 32 filtros. La segunda y tercera, contienen 64, mientras que para la cuarta, quinta y sexta se emplearon respectivamente 32, 16 y 8 filtros. Cabe mencionar que, para la primera capa, la dimensión de los filtros es de 7x7, mientras que para el resto es de 5x5. Adicionalmente, al final de cada bloque de extracción de atributos, se utilizó una función de activación Luky Relu, con alfa de 0.2 y un Max Pool de 2x2, ver Figura 4.

**CNN-Mediana.** Esta red cuenta con cuatro capas convolucionales 2D para la extracción de atributos. La primera de estas, recibe una imagen de 682 x 853 píxeles en formato RGB y está compuesta por 64 filtros. La segunda contiene 32, mientras que para la tercera y cuarta, se emplearon respectivamente 16 y 8 filtros. Cabe mencionar que, para la primera capa, la dimensión de los filtros es de 7x7, mientras que para el resto es de 5x5. Adicionalmente, al final de cada bloque de extracción de atributos, se utilizó una función de activación Luky Relu, con alfa de 0.2 y un Max Pool de 2x2, ver Figura 5. Por último, para que la dimensionalidad a la salida coincida con la red fina, para las capas convolucionales tres y cuatro se utilizó un paso o Stride de 2x2, es decir, los filtros utilizados para la convolución se desplazaron dos píxeles a la vez.

**CNN-Gruesa.** Esta red cuenta con tres capas convolucionales 2D para la extracción de atributos. La primera de estas, recibe una imagen de 682 x 853 píxeles en formato RGB y está compuesta por 64 filtros de 7x7. Por su parte, la segunda y tercera contienen 8 filtros de 5x5, ver Figura 6.

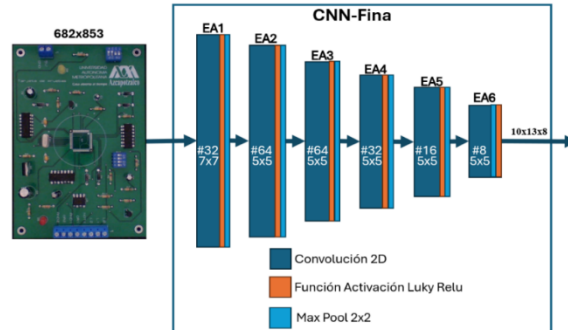


Fig. 4. Extracción de atributos: CNN-Fina.

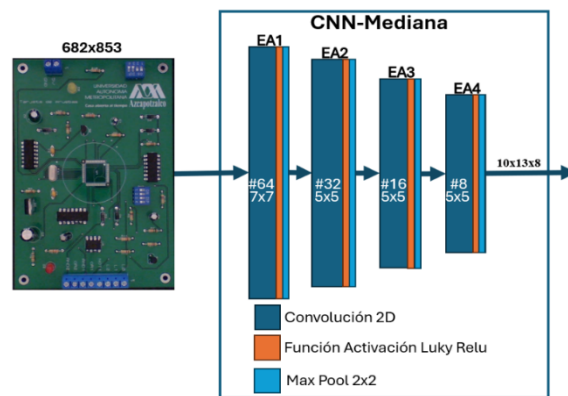


Fig. 5. Extracción de atributos: CNN-Mediana.

Adicionalmente, al final de cada bloque de extracción de atributos, se utilizó una función de activación Luky Relu, con alfa de 0.2 y un Max Pool de 2x2, Por último, para que la dimensionalidad a la salida coincida con la red fina y la mediana, en las tres capas convolucionales se empleó un Stride de 2x2.

**Total de atributos extraídos.** Las características extraídas por cada red, fueron utilizadas de acuerdo a las siguientes proporciones: 50% CNN-Fina, 30% CNN-Mediana y 20% CNN-Gruesa, ver Figura 7. De esta manera se genera un vector de 1040 elementos, que corresponde a la totalidad de los atributos extraídos.

## 4.2. Clasificación

La etapa de clasificación, mediante una red neuronal, tal y como se muestra en la Figura 8, analiza los atributos extraídos. Esta red está compuesta por tres capas totalmente conectadas con 520, 130 y 65 neuronas, respectivamente, además de una función de activación Selu. Adicionalmente, se tiene una capa de salida con seis neuronas y una función de activación SoftMax. Lo anterior, se debe a que se requiere, como resultado, una clasificación para este número de clases.

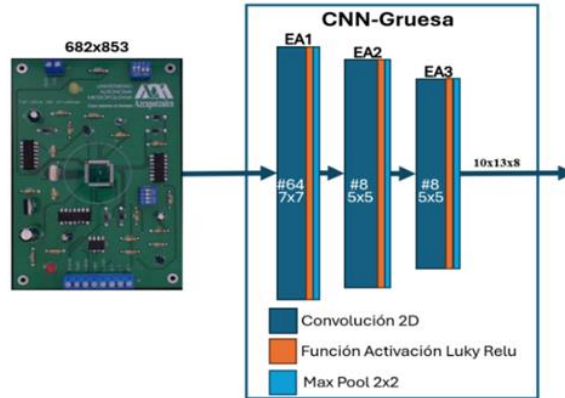


Fig. 1. Extracción de atributos: CNN-Gruesa.

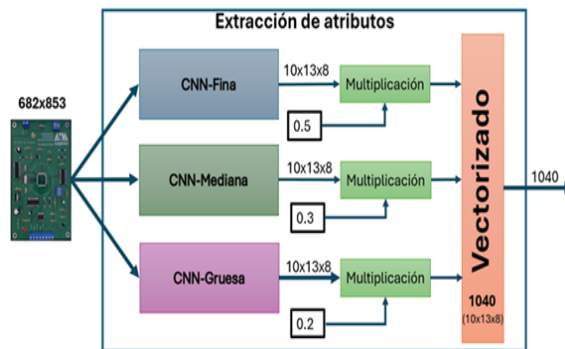


Fig. 7. Total de atributos extraídos.

## 5. Entrenamiento y evaluación

En esta sección se describe el equipo, los programas de cómputo y las métricas utilizadas para entrenar y evaluar el modelo de aprendizaje profundo, cuyo diseño ha sido descrito. Así mismo, se detalla el procedimiento y resultados obtenidos al realizar estas dos tareas.

### 5.1. Equipo y programas de cómputo utilizados

El entrenamiento y evaluación del modelo de aprendizaje profundo, fue realizado en una computadora marca Dell Alienware Aurora R13 Gaming Desktop, equipada con un procesador Intel Core i9 de doceava generación, 128 GB de memoria RAM, 2 TB SSD y 2 TB HDD. Adicionalmente, se empleó una GPU: Nvidia Geforce Rtx 3090 con 16 núcleos, a 5.2 GHz y memoria RAM de 24 GB GDDR6X. El sistema operativo utilizado fue Ubuntu 22.04 LTS de 64-bits. Como lenguaje de programación, se utilizó Python 3.9, complementado con las bibliotecas: Tensorflow y Keras versión 2.13, scikit-learn, Matplotlib y NumPy.



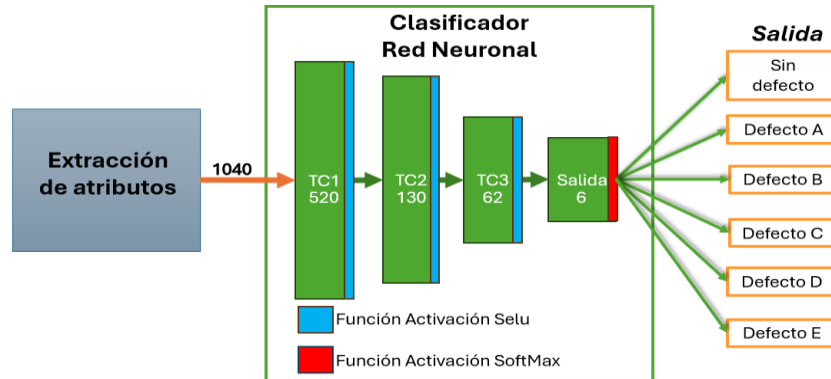


Fig. 8. Clasificación.

## 5.2. Métricas de entrenamiento y evaluación

La función de pérdida de tipo entropía cruzada categórica (categorical cross-entropy), fue utilizada durante el entrenamiento y la evaluación, para determinar el error del modelo, ecuación 1:

$$f_{p_{\text{categorical cross-entropy}}} = -\frac{1}{N} \sum_{i=0}^N \sum_{c=0}^N y * \log(\hat{y}) , \quad (1)$$

donde:

$N$  = Número de ejemplos en el conjunto de datos.

$c$  = Número de clases.

$y$  = Valor binario cuyo valor es 1 si  $i$  pertenece a la clase  $c$ , y 0 si no.

$\hat{y}$  = Probabilidad de que  $i$  pertenezca a la clase  $c$ .

Por su parte, para la evaluación fueron utilizadas las métricas: exactitud, precisión, sensibilidad y F1 score [24]. Estas, básicamente dependen de la cantidad de veces que se acertó o no en la predicción de la clasificación, de tal forma que dan información sobre el rendimiento del modelo y determinan si este es o no funcional.

**Exactitud.** Permite evaluar el rendimiento general de un modelo de clasificación. Se define como el número de predicciones correctas dividido entre el número total de intentos, ecuación 2:

$$Exactitud = \frac{\text{Verdaderos positivos} + \text{Verdaderos negativos}}{\text{Total de predicciones}}. \quad (2)$$

**Precisión.** Representa la cantidad de predicciones correctas del total de intentos, para una clase en particular, ecuación 3:

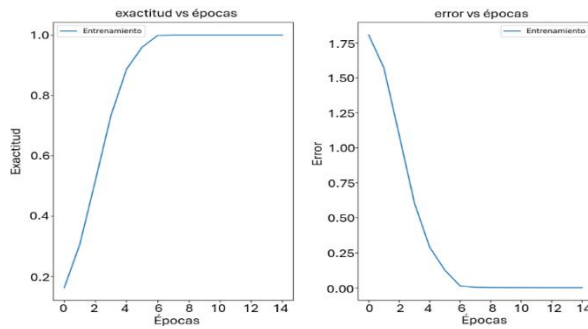


Fig. 9. Gráficas obtenidas durante el entrenamiento del modelo de aprendizaje profundo.

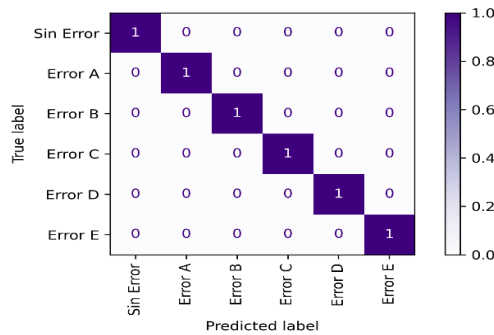


Fig. 10. Matriz de confusión correspondiente a la evaluación del modelo de aprendizaje profundo.

$$Precision = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos} \tag{3}$$

**Sensibilidad.** Se refiere a la capacidad del modelo para identificar la cantidad de casos positivos correctamente clasificados, para una clase en particular, ecuación 4:

$$Sensibilidad = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos} \tag{4}$$

**F1 score.** Se utiliza comúnmente para conocer el equilibrio entre la precisión y la sensibilidad, para una clase en particular, en un modelo computacional de clasificación. Un valor bajo de F1 score puede indicar falta de calidad en las predicciones, este se calcula mediante la ecuación 5:

$$F1_{score} = \frac{2(Precision * Recall)}{Precision + Recall} \tag{5}$$

Adicionalmente, para completar la evaluación del modelo, fue generada la matriz de confusión correspondiente. Esta herramienta, es una representación de los aciertos obtenidos en la clasificación de un conjunto de datos, para los cuales se conoce a que

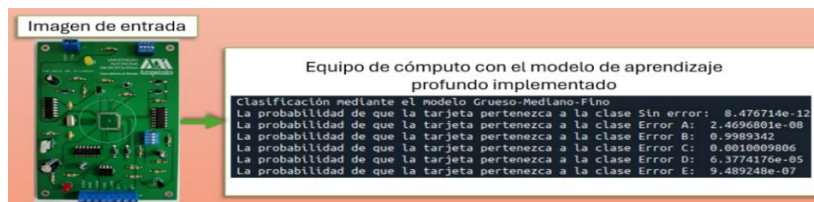


Fig. 11. Implementación del modelo de aprendizaje profundo.

Tabla 2. Resultados de las pruebas de implementación del modelo de aprendizaje profundo.

Tipo de Tarjeta	Sin Error	Error A	Error B	Error C	Error D	Error E
Sin Error	10	0	0	0	0	0
Error A	0	10	0	0	0	0
Error B	0	0	10	0	0	0
Error C	0	0	0	10	1	0
Error D	0	0	0	0	9	0
Error E	0	0	0	0	0	10

clase pertenecen. Las filas de esta matriz representan las clases reales, mientras que las columnas corresponden a las clases predichas, cada celda contiene el conteo correspondiente a la clasificación realizada por el modelo.

### 5.3. Entrenamiento

Para su entrenamiento, el modelo de aprendizaje profundo fue compilado utilizando una función de pérdida de tipo entropía cruzada categórica, un optimizador Adam, con una tasa de aprendizaje de 0.0005.

Posteriormente, el modelo se entrenó durante 15 épocas con un tamaño de lote (Batch size) de 10 muestras, utilizando el 80% de la base de datos descrita anteriormente.

De manera que, de las 2376 imágenes, 1900 fueron utilizadas para el entrenamiento. Al término de esta tarea, el error fue de cero, con una exactitud del 100%. Las gráficas de la Figura 9 muestran el comportamiento de las métricas por época.

### 5.4. Evaluación

La evaluación del modelo de aprendizaje profundo fue realizada utilizando el 20% de la base de datos descrita anteriormente. De manera que, de las 2376 imágenes, 476 fueron utilizadas para esta tarea. Al término de este proceso de valoración, los resultados corroboraron lo conseguido en el entrenamiento, ya que se obtuvo un error de 0.00034, con una exactitud del 100%. Los valores de precisión, sensibilidad y F1 score obtenidos para cada clase fueron del 100%. Por último, en la Figura 10 se muestra la matriz de confusión normalizada, obtenida a partir de la evaluación de dicho modelo.

## **6. Implementación**

Posterior al entrenamiento y evaluación del modelo de aprendizaje profundo, descrito en las secciones anteriores, se realizó su implementación en un equipo de cómputo. El modelo recibe una imagen de la tarjeta de circuito impreso, la cual previamente se reduce de 1024x1280 a 682x853 píxeles. Posteriormente, se realiza la predicción correspondiente y, en tiempo real, se indica la probabilidad de que la tablilla presenté, o no, algún tipo de defecto. La Figura 11 muestra la información que el equipo de cómputo despliega en pantalla al realizar la predicción.

En pruebas realizadas con 10 imágenes por cada tipo de tarjeta de circuito impreso, se encontró una precisión en la clasificación del 90% para la clase: Error D, mientras que para todas las demás fue del 100%. La Tabla 2 muestra los resultados obtenidos en este procedimiento, las filas representan las clases reales, mientras que las columnas corresponden a las clases predichas y cada celda contiene el conteo correspondiente a la clasificación realizada por el modelo.

## **7. Conclusiones**

El diseño, entrenamiento, evaluación e implementación de un nuevo modelo de aprendizaje profundo, basado en redes neuronales convolucionales, aplicado en la inspección de tarjetas de circuitos impresos, fue descrito en este artículo. Este modelo, el cual es exclusivo a un único diseño de PCB, está compuesto por dos partes: extracción de atributos y clasificación. El modelo es capaz de determinar si una placa presenta, o no, alguno de los 5 defectos propuestos.

Los resultados obtenidos en el entrenamiento, muestran un error de cero y una exactitud del 100%, en la clasificación de las tarjetas. Por su parte, los valores que estas métricas alcanzaron en la evaluación realizada, fueron prácticamente iguales, corroborando el buen rendimiento del modelo. Por lo que, en comparación con otros métodos, se tiene una eficacia similar y en ocasiones mayor. Por su parte, el esquema Grueso-Mediano-Fino, empleado en la extracción de atributos, funcionó correctamente en el problema planteado en este artículo.

### **7.1. Trabajo a futuro**

Como trabajo a futuro, se plantea la modificación del modelo de aprendizaje profundo presentado en esta publicación académica, para añadir una etapa de segmentación, la cual sea capaz de identificar la totalidad de componentes presentes en la tarjeta de circuito impreso y con esto mejorar la inspección. Adicionalmente, se buscará aplicar el modelo en la identificación de componentes electrónicos de tamaños y naturaleza distintos a los presentados en este artículo.

## **Referencias**

1. Pastor-López, I., Santos, I., Santamaría-Ibirika, A., Salazar, M., De-la-Peña-Sordo, J., Bringas, P.G.: Machine-learning-based surface defect detection and categorisation in high-

- precision foundry. In: 7th IEEE Conference on Industrial Electronics and Applications pp. 1359–1364 (2012). DOI: 10.1109/ICIEA.2012.6360934.
2. Neethu, N., Anoop, B.: Role of computer vision in automatic inspection systems. *International Journal of Computer Applications*, vol. 123, no. 13, pp. 28–31 (2015)
  3. Wu, C., Wang, D., Ip, A., Wang, D.W., Chan, C., Wang, H.: A particle swarm optimization approach for components placement inspection on printed circuit boards. *Journal of Intelligent Manufacturing*, vol. 20, no. 5, pp. 535–549 (2009). DOI: 10.1007/s10845-008-0140-2.
  4. Tornero-Montserrat, J., Armesto, L., Mora-Aguilar, M.C., Montés-Sánchez, N., Herraiz-Martínez, A., Asensio, J.M.: Detección de defectos en carrocerías de vehículos basado en visión artificial: Diseño e implantación. *Revista Iberoamericana de Automática e Informática Industrial*, vol. 9, no. 1, pp. 93–104 (2012). DOI: 10.1016/j.riai.2011.11.010.
  5. Kumar, R., Rani, P.: Machine Learning strategies in real-world engineering applications: A comprehensive survey. In: *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 16, pp. 131–140 (2024)
  6. Aparicio-Pico, L.E., Amaya-Marroquín, O.J., Devia-Lozano, P.A: Aplicación de Deep Learning para la identificación de defectos superficiales utilizados en control de calidad de manufactura y producción industrial: Una revisión de la literatura. *Ingeniería*, vol. 28, no. 1 (2023). DOI: 10.14483/23448393.18934.
  7. Avilés-Cruz, C., Ferreyra-Ramírez, A., Zúñiga-López, A., Cortéz, J.: Coarse-Fine convolutional Deep-Learning strategy for human activity recognition. *Sensors*, vol. 19, no. 7, pp. 1556 (2019). DOI: 10.3390/s19071556.
  8. Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., Tang, S.: Using Deep Learning to detect defects in manufacturing: A comprehensive survey and current challenges. *Materials*, vol. 13, no. 24, pp. 5755 (2020). DOI: 10.3390/ma13245755.
  9. Janiesch, C., Zschech, P., Heinrich, K.: Machine Learning and Deep Learning. *Electronic Markets*, vol. 31, no. 3, pp. 685–695 (2021). DOI: 10.1007/s12525-021-00475.
  10. Kulkarni, R., Kulkarni, S., Dabhane, S., Lele, N., Paswan, R.S.: An automated computer vision based system for bottle cap fitting inspection. In: 2019 Twelfth International Conference on Contemporary Computing, pp. 1–5 (2019). DOI: 10.1109/IC3.2019.8844942.
  11. Rajan, A.J., Jayakrishna, K., Vignesh, T., Chandradass, J., Kannan, T.T.M.: Development of computer vision for inspection of bolt using convolutional neural network. *Materials Today: Proceedings*, vol. 45, no. 7, pp. 6931–6935 (2021). DOI: 10.1016/j.matpr.2021.01.372.
  12. Tran, H.T., Tran, M.Q., Tran, Q.H., Pham, V.C.: A computer vision system for power transmission line inspection robot. In: 2021 International Conference on System Science and Engineering, pp. 289–294 (2021). DOI: 10.1109/ICSSE52999.2021.9538440.
  13. Ficzer, M., Mészáros, L.A., Kállai-Szabó, N., Kovács, A., Antal, I., Nagy, Z.K., and Galata, D.L.: Real-time coating thickness measurement and defect recognition of film coated tablets with machine vision and Deep Learning. In: *International Journal of Pharmaceutics*, vol. 623, no. 1, pp. 121957 (2022). DOI: 10.1016/j.ijpharm.2022.121957.
  14. Aggarwal, N., Deshwal, M., Samant, P.: A survey on automatic printed circuit board defect detection techniques. In: 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 853–856, India (2022). DOI: 10.1109/ICACITE53722.2022.9823872.
  15. Kim, J., Ko, J., Choi, H., Kim, H.: Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors*, vol. 21, no. 15, pp. 4968 (2021). DOI: 10.3390/s21154968.

16. Park, J., Shin, S.Y.: Printed circuit board defect detection using generative Deep Learning model. In: 13th International Conference on Information and Communication Technology Convergence, pp. 463–466 (2022). DOI: 10.1109/ICTC55196.2022.9952939.
17. Austria, Y.D., Fajardo, A.C.: Defect detection and classification in printed circuit boards using convolutional neural networks. In: 2nd International Conference on Edge Computing and Applications, pp. 1498–1504 (2023). DOI: 10.1109/ICECAA58104.2023.10212195.
18. Kim, Y.G., Park, T.H.: SMT Assembly inspection using dual-stream convolutional networks and two solder regions. *Applied Sciences*, vol. 10, no. 13, pp. 4598 (2020). DOI: 10.3390/app10134598.
19. Sezer, A., Altan, A.: Detection of solder paste defects with an optimization-based Deep Learning model using image processing techniques. *Soldering and Surface Mount Technology*, vol. 33, no. 5, pp. 291–298 (2021). DOI: 10.1108/SSMT-04-2021-0013.
20. Lian, J., Wang, L., Liu, T., Ding, X., Yu, Z.: Automatic visual inspection for printed circuit board via novel Mask R-CNN in smart city applications. *Sustainable Energy Technologies and Assessments*, vol. 44, no. 1, pp. 101032 (2021). DOI: 10.1016/j.seta. 2021.101032.
21. Gang, S., Fabrice, N., Chung, D., Lee, J.: Character recognition of components mounted on printed circuit board using Deep Learning. *Sensors*, vol. 21, no. 9, pp. 2921 (2021). DOI: 10.3390/s21092921.
22. Zhang, K.: Using deep learning to automatic inspection system of printed circuit board in manufacturing industry under the internet of things. *Computer Science and Information Systems*, vol. 20, no. 2, pp. 723–741 (2023). DOI: 10.2298/CSIS220718020.
23. Jia, X., Liu, Z.: Character identification for integrated circuit components on printed circuit boards using Deep Learning. *Journal of Electrical Engineering and Technology*, vol. 17, no. 1, pp. 601–616 (2022). DOI: /10.1007/s42835-021-00885-4.
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Duchesnay, É.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, no. 1, pp. 2825–2830 (2011)

# Traductor inglés-español por reconocimiento de texto

Miriam Calderón Reyes, Daniela Jiménez Pano,  
Norma Angélica Márquez Sulca, Luis Carlos Padierna García

Universidad de Guanajuato,  
División de Ciencias e Ingenierías León,  
México

{m.calderon.reyes, d.jimenezpano  
na.marquezsulca, lc.padierna}@ugto.mx

**Resumen.** En la búsqueda continua de soluciones pragmáticas en el ámbito de la inteligencia artificial, presentamos un proyecto concreto que fusiona la visión computarizada y el procesamiento de lenguaje natural. Nuestro enfoque se centra en el desarrollo de un traductor automático que, a través de una Red Neuronal Convolutiva (CNN), analiza imágenes para extraer texto, y posteriormente utiliza una Red Transformer para realizar traducciones precisas. El uso de CNN permite identificar y aislar regiones de texto en imágenes, convirtiendo datos visuales en información legible. Este proceso sirve como punto de partida para la siguiente etapa, donde una Red Transformer se encarga de contextualizar y traducir el texto reconocido. Esta combinación técnica no solo simplifica la experiencia del usuario al eliminar la necesidad de ingreso manual de texto, sino que también demuestra aplicaciones prácticas en entornos donde la comunicación multilingüe basada en imágenes es esencial. Aunque la precisión actual del traductor sea del 60%, se están explorando posibles mejoras para maximizar su eficacia.

**Palabras clave:** CNN, red Transformer, padding, deep learning.

## English-Spanish Text Recognition Translator

**Abstract.** In the ongoing quest for pragmatic solutions in the field of artificial intelligence, we present a concrete project that merges computer vision and natural language processing. Our approach focuses on developing an automatic translator that, utilizing a Convolutional Neural Network (CNN), analyzes images to extract text, and subsequently employs a Transformer Network to perform accurate translations. The use of CNN enables the identification and isolation of text regions in images, converting visual data into readable information. This process serves as a starting point for the subsequent stage, where a Transformer Network contextualizes and translates the recognized text. This technical combination not only simplifies user experience by eliminating the need for manual text input but also demonstrates practical applications in environments where multilingual image-based communication is essential. Although the current translator accuracy stands at 60%, we are exploring potential enhancements to maximize its effectiveness.

**Keywords:** CNN, red Transformer, padding, deep learning.

## 1. Introducción

La historia de las redes neuronales convolucionales (CNN) comienza en la década de 1980, esto sucede cuando Yann LeCun, nacido en Francia en 1960, desarrolló la arquitectura LeNet, dicha arquitectura fue pionera en el reconocimiento de patrones y visión por computadora, dado que era capaz de reconocer dígitos escritos a mano con gran precisión, como presentó en su artículo “Gradient-Based Learning Applied to Document Recognition” [1].

La arquitectura LeNet se diseñó para el reconocimiento de caracteres en imágenes, esto se logró realizando una serie de capas convolucionales y capas de pooling, continuando con capas completamente conectadas para alcanzar su objetivo. Como es habitual, la estructura LeNet se ha modificado al pasar de los años para optimizar y ampliar su funcionalidad como la segmentación de imágenes e incluso la traducción de texto a partir de imágenes, tal como se lleva a cabo en este proyecto.

En los años 90, se introdujeron capas de activación no lineales, tal como lo son la función ReLU o sigmoide, así como técnicas de regularización, para evitar el overfitting. En este último siglo, con el aumento de popularidad del deep learning, se incrementó la cantidad disponible de datasets y aparecieron arquitecturas como VGGNet o ResNet, las cuales son prueba de un avance notable en el reconocimiento de objetos.

Posteriormente, un artículo titulado *Attention is All You Need* introdujo una visión completamente nueva en el procesamiento de lenguaje natural, puesto que introdujo la arquitectura Transformer, esta es un modelo de red neuronal no recurrente independiente de capas recurrentes debido a que se basan en mecanismos de atención que permiten que cada palabra dentro de una secuencia interactúe con el resto de ellas dado que asignan pesos a cada palabra en función de su relevancia [2].

Hoy los Transformers son una base fundamental para algunos de los modelos de Inteligencia Artificial más conocidos y avanzados, como GPT y BERT.

## 2. Dataset

### 2.1 Identificación de texto a partir de imágenes

Para la parte de redes neuronales convolucionales (CNN) se utilizaron 2 conjuntos de datasets, EMNIST Y MNIST. Estos dos fueron obtenidos en Kaggle [3]. MNIST es un estándar para reconocer dígitos escritos a mano, que contiene 70,000 imágenes en total, cada una de 28x28 píxeles, representando números escritos a mano del 0 al 9. Por otra parte, el conjunto de datos EMNIST incluye más datos, siendo estas 814,000 imágenes de 28x28 píxeles para incluir las letras del alfabeto, representando tanto mayúsculas como minúsculas.

### 2.2 Traductor

El dataset utilizado para la traducción consta de dos conjuntos de datos distintos. El primero se obtuvo de Kaggle [4] y contiene un total de 118,964 palabras u oraciones en inglés y sus respectivas traducciones al español. Este dataset incluye una variedad de expresiones en inglés, algunas de las cuales se repiten para capturar las diferentes formas



de traducción al español. Además de las frases recopiladas, se agregaron traducciones básicas que abarcan listas de animales y colores.

Con el objetivo de ampliar la longitud de las traducciones y permitir que el Transformer no solo aprenda el equivalente de palabras, sino también la estructura textual en inglés se incorporó un segundo dataset [5]. Este conjunto de datos, creado por nuestro equipo para este proyecto, consta de 18,838 traducciones adicionales y se compone de la traducción verso a verso de más de 300 canciones, lo que proporciona una variedad de estructuras y estilos de texto en inglés.

### 3. Arquitectura

#### 3.1 Identificación de texto a partir de imágenes

Se empleó la arquitectura de red neuronal convolucional (CNN) LeNet para crear un traductor eficiente que analiza imágenes y extrae texto para su traducción al inglés. Esta versión de LeNet [6] consta de 7 capas en total, incluyendo 2 de convolución, 2 de pooling y 2 totalmente conectadas, seguidas por una capa de salida.

Las capas de convolución (C1 y C3) detectan características locales como bordes y texturas en las imágenes de entrada, siendo esenciales para la extracción de texto. Las capas de pooling (S2 y S4) reducen la dimensionalidad de la salida de las capas convolucionales manteniendo las características más importantes.

Las capas totalmente conectadas proporcionan una representación más abstracta de la imagen, permitiendo el reconocimiento de patrones complejos, mientras que la capa de salida clasifica las imágenes en categorías específicas como letras o palabras, crucial para la tarea de traducción.

La adaptabilidad de LeNet a conjuntos de datos específicos como EMNIST y MNIST mejora la precisión del reconocimiento de letras y, por ende, la calidad de las traducciones generadas. La capacidad de LeNet para reconocer letras en imágenes [7], se alinea perfectamente con la tarea de traducción de texto.

Además, se extendió la arquitectura clásica de LeNet incorporando capas de Batch Normalization después de cada capa convolucional y completamente conectada. Se adaptó la red para permitir un número variable de clases de salida, aumentando así su flexibilidad y capacidad de generalización.

#### 3.2 Traductor

Antes de utilizar el dataset para entrenar el modelo, se realizó un preprocesamiento. Este tratamiento incluyó la limpieza del texto, eliminando caracteres como acentos y signos de puntuación, además, se normalizó el texto convirtiéndolo completamente a minúsculas. Este preprocesamiento ayuda a garantizar que el modelo pueda aprender las relaciones entre las palabras y las estructuras de las oraciones, independientemente de las variaciones en la capitalización o los caracteres especiales. Posteriormente, se generó una bolsa de palabras con todas las palabras presentes en el dataset. A cada palabra se le asoció un número único y se incorporaron los términos clave *start*, *end* y *pad* para el padding. El vocabulario resultante fue de 15,209 palabras distintas.

**Tabla 1.** Configuración de la red transformer utilizada.

Parámetro	Valor	Parámetro	Valor
token_num	137802	head_num	4
embed_dim	32	hidden_num	128
encoder_num	2	dropout_rate	0.05
decoder_num	2	use_same_embed	False

Con el objetivo de representar cada oración en el dataset como un vector numérico, se generaron vectores compuestos por los números que representan cada palabra. Todos los vectores se ajustaron a la misma longitud, la cual corresponde a la dimensión del espacio vectorial más largo, en este caso, 43. Aquellos vectores de longitud inferior fueron rellenados con padding para alcanzar esta longitud estándar.

El modelo Transformer, importado de la librería Keras-Transformer [8], fue configurado en base a los parámetros descritos en el artículo *Attention is All You Need* [2], seleccionando valores cercanos, pero más bajos dada de la capacidad computacional del equipo utilizado y el tamaño del dataset. Estos valores se detallan en la tabla 1.

Se procedió a compilarlo utilizando el optimizador Adam y la función de pérdida `sparse_categorical_crossentropy`. Se optó por Adam debido a su capacidad para adaptar automáticamente las tasas de aprendizaje durante el entrenamiento, lo que es esencial para lidiar con la variabilidad en la longitud de las secuencias de entrada y salida en el proceso de traducción [9]. Por otro lado, la función de pérdida `sparse_categorical_crossentropy` se seleccionó debido a su capacidad para modelar la predicción de múltiples clases, palabras en el vocabulario de salida, de manera eficiente [10]. Es importante destacar que el preprocesamiento descrito en esta sección se basa en las recomendaciones de la investigación previa sobre el uso de transformers para tareas de traducción automática [11].

### 3.3 Metodología

La Figura 1 presenta el diagrama de la metodología del proyecto. Se trabaja en paralelo en dos aspectos: por un lado, el reconocimiento de texto utilizando una CNN, y por el otro, la traducción del texto mediante una red Transformer. Los parámetros seleccionados para cada proceso se especifican en la Figura 1.

La Figura 2 muestra la evaluación del modelo. En esta etapa, el texto reconocido en la imagen es generado como salida por la CNN y luego es procesado por el Transformer como entrada para la traducción.

## 4. Resultados

Para evaluar la precisión y asertividad del proyecto, utilizamos dos conjuntos de datos creados internamente. El primer conjunto cuenta con 667 imágenes frases y palabras en español, mientras que el segundo contiene 823 imágenes frases y palabras en inglés. Se evaluó la capacidad del traductor tanto para traducir del español al inglés como del inglés al español utilizando las imágenes reconocidas como entrada.

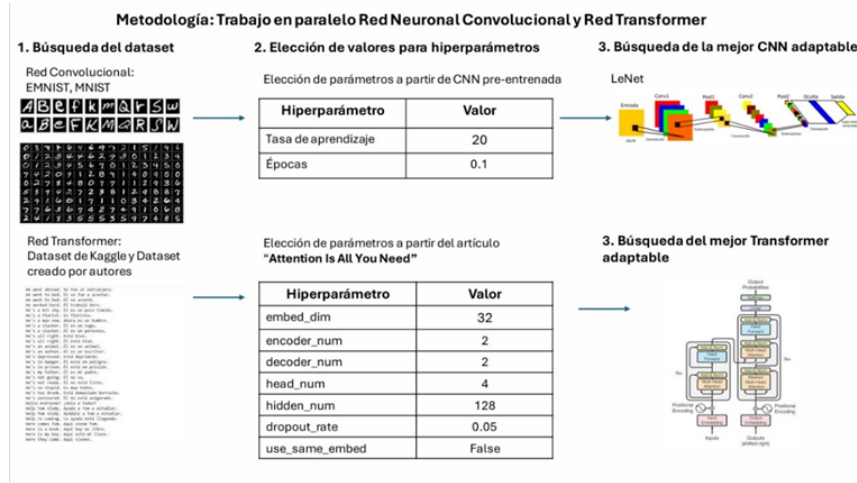


Fig. 1. Diagrama de pre-procesamiento y procesamiento de los inputs.

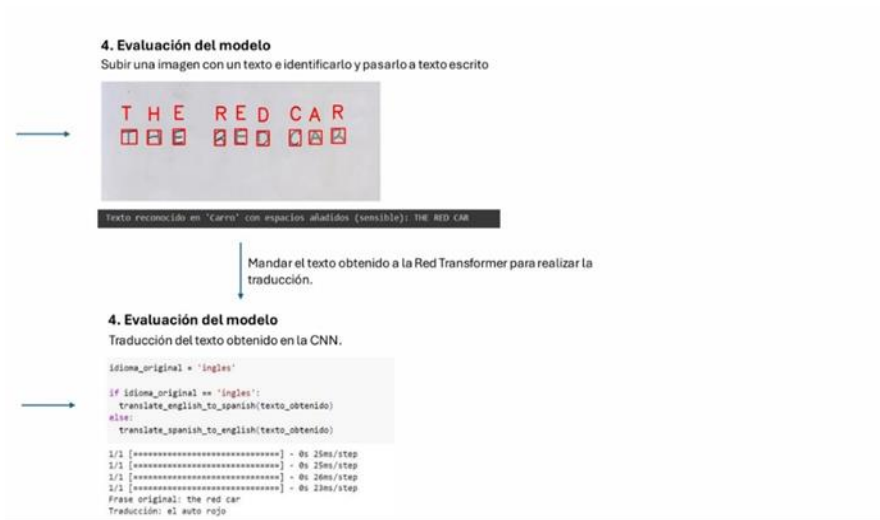


Fig. 2. Evaluación de los modelos, el producto de uno (CNN) siendo el input del otro (Red Transformer).

La evaluación de asertividad la realizó un humano especialista en inglés, donde se consideraron correctas aquellas traducciones que capturaban el significado contextual de las palabras y frases, siempre y cuando la entrada de imagen fuera correctamente reconocida y procesada por el sistema. Esto significa que el porcentaje de asertividad presentado se refiere específicamente a las traducciones realizadas con éxito en las entradas de imagen correctamente procesadas.

Durante la evaluación, se observó que las traducciones resultantes del programa requerían un proceso de postprocesamiento debido a posibles errores como caracteres

**Tabla 2.** Ejemplo de funcionamiento del traductor automático.

Imagen del texto	Texto detectado	Traducción	Revisión
	Always	Siempre	Correcto
	Barato	Cheap	Correcto
	Nacional	Vicious	Incorrecto
	Destroy	Lalalalala	Incorrecto
	Fecha	Changed i out	Incorrecto
	Healthy	Sano	Correcto
	Resfriado	Cold man	Correcto
	Red dress	Vestido rojo	Correcto

**Tabla 3.** Precisión de la traducción.

Modelo	Traducciones correctas	Porcentaje de asertividad
Inglés a español	526	63.9%
Español a inglés	376	56.4%

adicionales o repeticiones de frases. A pesar de estos detalles, se consideraron correctas aquellas traducciones que reflejaban con precisión el significado de las frases, incluso si presentaban errores en la estructura gramatical o las conjugaciones verbales.

En la Tabla 2 se presentan seis de las imágenes de entrada y las respectivas salidas del modelo que corresponden a las traducciones.

Notablemente, el modelo desarrollado no solo proporcionó traducciones palabra por palabra entre idiomas, sino que mostró un entendimiento rudimentario del contexto y significado de las frases. Por ejemplo, la palabra 'resfriado' fue traducida como "cold man" (véase Tabla 2), lo que refleja una interpretación del concepto más allá de una traducción literal.

Esto también se evidencia en la traducción de la frase en inglés "red dress" (véase Tabla 2), donde se observa que el traductor comprende la estructura de ambos idiomas. Por ejemplo, el orden del sustantivo y el adjetivo se invierte al pasar del inglés al español, lo que demuestra una capacidad para interpretar y adaptar la estructura lingüística de manera contextual.

La Tabla 3 muestra el porcentaje de asertividad obtenido con la ecuación 1:

$$Asertividad = \frac{Traducciones\ correctas}{Traducciones\ totales} \times 100. \tag{1}$$

Es importante destacar que estos resultados son producto de una evaluación interna. Para una evaluación más objetiva se requiere contar con especialistas externos que analicen el funcionamiento y rendimiento del traductor automático basado en reconocimiento de texto en imágenes.

## **5. Conclusiones**

Durante este proyecto, se han mezclado armoniosamente la noción de visión por computadora con procesamiento de lenguaje natural para desarrollar un traductor, es decir, combina una red neuronal convolucional para la identificación de texto en imágenes con una arquitectura Transformer. Este modelo, logró una precisión del 56 – 64% entre la traducción bilateral de español e inglés, a pesar de que son resultados positivos, es evidente que aún hay limitaciones en la comprensión del significado, sobre todo contextual, como en el ejemplo mencionado anteriormente con “resfriado”.

Es importante destacar que el tamaño del dataset utilizado, si bien es adecuado y suficiente para los propósitos del proyecto, no es destacablemente grande. El modelo demostró poder aprender patrones significativos y generar traducciones coherentes para diversos casos.

## **Referencias**

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324 (1998). DOI: 10.1109/5.726791.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008 (2017)
3. Kaggle: <https://www.kaggle.com/datasets/crawford/emnist> (2024)
4. Kaggle: <https://www.kaggle.com/datasets/lonnieqin/englishspanish-translation-dataset?select=data.csv> (2024)
5. Github: Miricalderonr.: Traductor\_por\_reconocimiento\_de\_texto. Repositorio de GitHub, <https://github.com/miricalderonr/Traductor-por-reconocimiento-de-texto> (2024)
6. Github: Repositorio de GitHub, [https://github.com/bmitt/emnist\\_ocr](https://github.com/bmitt/emnist_ocr) (2024)
7. Keras.io: [https://keras.io/api/losses/probabilistic\\_losses/](https://keras.io/api/losses/probabilistic_losses/) (2024)
8. Kingma, D.P., Ba, J.: A Method for Stochastic Optimization. In: *ICLR* (2015)
9. Lin, T., Wang, Y., Liu, X., and Qiu, X.: A survey of transformers. *Artificial Intelligence Open Access*. vol. 8, no. 4 (2022). DOI: 10.1016/j.aiopen.2022.10.001.
10. Al-Jawfi, R.: Handwriting Arabic character recognition LeNet using neural network. *The International Arab Journal of Information Technology*, vol. 6, no. 3, pp. 304–309 (2009)
11. Github: Codificandobits: Traductor\_con\_redes\_Transformer. Repositorio de GitHub, [https://github.com/codificandobits/Traductor\\_con\\_redes\\_Transformer](https://github.com/codificandobits/Traductor_con_redes_Transformer) (2024)



## **Análisis de ataques de denegación de servicio distribuido (DDoS) en un conjunto de datos usando aprendizaje automático**

Jesus Barranco Castillo, Arturo Zúñiga-López,  
Carlos Avilés-Cruz, Cesar Benavides-Álvarez

Universidad Autónoma Metropolitana,  
Unidad Azcapotzalco, Departamento de Electrónica,  
México

{al2163054940, azl, caviles, cesarbenavides}@azc.uam.mx

**Resumen.** La seguridad en las redes es y seguirá siendo un tema de actualidad, pues la cantidad de ataques hacia empresas, instituciones o incluso personas, es una constante hoy en día. Gracias a las herramientas que se tienen actualmente, los atacantes ya no necesitan tener grandes conocimientos sobre redes y programación para realizar ataques muy sofisticados. En este trabajo se utilizó la base de datos que contiene ataques DDoS, creada por el Instituto Canadiense de Ciberseguridad llamada "CICDDoS2019". La base de datos fue analizada, utilizando las técnicas de red neuronal artificial (RNA) y máquinas de soporte vectorial (MSV). Los datos también fueron tratados con análisis de componentes principales (PCA) para reducir la dimensionalidad, además de reducir el poder de cómputo necesario al entrenar y predecir los datos. De las técnicas utilizadas se obtuvieron buenos resultados ya que en clasificación binarias (determinar si existe ataque) se obtiene una exactitud del 99% y de clasificación múltiple (si existe ataque y de que tipo es) se obtiene una exactitud del 96%, con la técnica de RNA, que fue la técnica que mejores resultados obtuvo.

**Palabras Clave:** Ataques DDoS, redes neuronal artificiales, máquinas de soporte vectorial, análisis de componentes principales.

### **Analysis of Distributed Denial of Service (DDoS) Attacks on a Dataset Using Machine Learning**

**Abstract.** Network security is and will continue to be a topical issue, since the number of attacks against companies, institutions or even individuals is a constant nowadays. Thanks to the tools currently available, attackers no longer need to have great knowledge of networks and programming to carry out very sophisticated attacks. In this work, the database containing DDoS attacks created by the Canadian Cybersecurity Institute called "CICDDoS2019" was used. The database was analyzed using artificial neural network (ANN) and support vector

machines (SVM) techniques. The data were also treated with principal component analysis (PCA) to reduce dimensionality, as well as to reduce the computational power required for data processing and prediction. Good results were obtained from the techniques used, since in binary classification (determining if there is an attack) an accuracy of 99% was obtained and in multiple classification (if there is an attack and what type it is) an accuracy of 96% was obtained, with the ANN technique, which was the technique that obtained the best results.

**Keywords:** DDoS attacks, artificial neural network, support vector machines, principal component analysis.

## 1. Introducción

La red es uno de los principales medios de transporte de la información en la actualidad, es por esto, que es muy propensa a sufrir ataques por parte de agentes malintencionados cuya finalidad es perjudicar a otros usuarios, en la actualidad tenemos diferentes tipos de ataques, como lo son ataque de ARP (protocolo de resolución de direcciones) spoofing, Man-In-The-Middle, denegación de servicio, denegación de servicio distribuido, etc., por lo que este trabajo está enfocado en los ataques de denegación de servicio distribuido.

Un ataque de denegación de servicio (DoS – Denial of Service), es un tipo de ataque dentro de una red, el cual consiste en hacer que un recurso o servicio no se encuentre disponible y entonces perjudica tanto al que ofrece el servicio como al que lo utiliza. Este ataque se puede volver más complejo, cuando el ataque no lo realiza un equipo sino múltiples equipos a la vez, entonces esto es conocido como ataque de denegación de servicio distribuido (DDoS). Existen varios tipos de ataques DDoS, sin embargo, el presente trabajo se enfoca en dos categorías: Los ataques DDoS basados en reflexión y los ataques DDoS basados en la explotación.

Los ataques DDoS de reflexión se pueden llevar a cabo usando protocolos de capa de transporte, como lo son el protocolo de control de transmisión (TCP) con ataques MSSQL (Microsoft SQL) y SSDP (protocolo simple de detección de servicio), y el protocolo de datagramas de usuario (UDP) con ataques CharGen (protocolo generador de caracteres), NTP (protocolo de tiempo de red) y TFTP (protocolo trivial de transferencia de archivos), también hay otros tipos de ataques como son DNS (sistema de nombres de dominio), LDAP (protocolo ligero de acceso a directorios), NETBIOS (sistema de entrada salida básica de red) y SNMP (protocolo simple de gestión de redes) que utilizan los protocolos antes mencionados.

Los ataques DDoS basados en la explotación también se pueden llevar a cabo usando protocolos de capa de transporte (TCP y UDP). Los ataques de explotación basados en TCP incluyen inundación SYN (bandera que indica el inicio de conexión entre host) y los ataques basados en UDP incluyen inundación UDP y UDP-Lag.

Por otro lado, el aprendizaje automático permite desarrollar herramientas que puedan brindar apoyo a la seguridad en las redes y en los sistemas de información, ya que permite que las computadoras puedan tener un proceso de aprendizaje, identificar patrones en un amplio conjunto de datos y obtener predicciones basadas en dichos patrones. Dentro del aprendizaje automático se tiene una rama llamada aprendizaje



supervisado (AS) la cual es usada en este trabajo, en donde se conocen claramente las características (entrada de información) y etiquetas (salidas), las cuales son usadas en el proceso de entrenamiento y en la determinación de la precisión del modelo de aprendizaje.

En este trabajo se usó el conjunto de datos generado por el Instituto Canadiense de Ciberseguridad (CIC) [1] llamado “CICDDoS2019” [2], el cual contiene una taxonomía de tráfico en la red con características importantes para detectar ataques de DDoS.

## **2. Antecedentes**

En el trabajo [3] se analizan ataques y tráfico normal en el conjunto de datos “CICDDoS2019” en donde se aplican las redes convolucionales ResNet50 y EfficientNet. Dado que dichas redes funcionan mejor con imágenes, se transforman los datos del tráfico de la red a imágenes, las cuales se introducen en esas redes para una clasificación múltiple (12 clasificaciones diferentes). Se determina como mejor solución a EfficientNet con 0.903 de precisión. En [4] propone DDosNet basada en red neuronal recurrente-autoencoder para la detección de ataques DDoS, se utiliza los datos de “CICDDoS2019” y se realiza una clasificación binaria, la red propuesta obtiene 0.99 de precisión al detectar ataques y 1.0 al detectar tráfico normal.

En [5] proponen la taxonomía de CicDDoS2019 como un nuevo conjunto de datos con características importantes para la detección de ataques DDoS, se implementan cuatro modelos conocidos de aprendizaje automático los cuales son ID3, Random Forest, Naïve Bayes y regresión logística con precisiones de 0.78, 0.77 0.41 0.25 respectivamente. En otro trabajo [6] se utiliza el conjunto de datos “CICDDoS2019, para determinar ataques de DDoS por reflexión en cuatro diferentes categorías (Portmap, LDAP, NetBIOS y MSSQL) y donde se obtuvo las siguientes precisiones Portmap 0.9958, LDAP 0.9999 NetBios 0.9999 y MSSQL 0.9999.

En el trabajo [7], proponen las técnicas de red neuronal artificial Naïve Bayes y máquinas de soporte vectorial también utilizan la base de datos CICDDoS 2019 y se determinó a la red neuronal artificial como la técnica más precisa con 0.99 de precisión, Naïve Bayes y máquinas de soporte vectorial se obtuvieron 0.982 0.981 de precisión respectivamente.

En [8] Comparan la precisión entre Snort (un sistema de detección de intrusos) y los sistemas de detección basados en aprendizaje profundo para detectar ataques DDoS. Presentaron que la precisión de Snort, RNA, RNA apilado y CNN (red neuronal convolucional) es 0.4716, 0.9976, 0.9956, respectivamente. Sin embargo, a Snort le tomo poco tiempo en detectar los ataques.

Hou et al. [9] introduce un método para detectar tráfico DDoS mediante aprendizaje automático. Extrajeron características basadas en flujos a partir de datos de muestreo en tiempo real utilizando NetFlow. Los resultados muestran que la precisión promedio es superior al 99% y un falso positivo inferior al 0.5%.

Roopak et al. [10] proponen un modelo híbrido CNN+LSTM que tiene una mayor precisión que otros modelos de aprendizaje profundo y modelos de aprendizaje automático. Se utilizó el conjunto de datos de CICIDS2017 y la precisión de detección de flujos de ataque es del 97.16%.

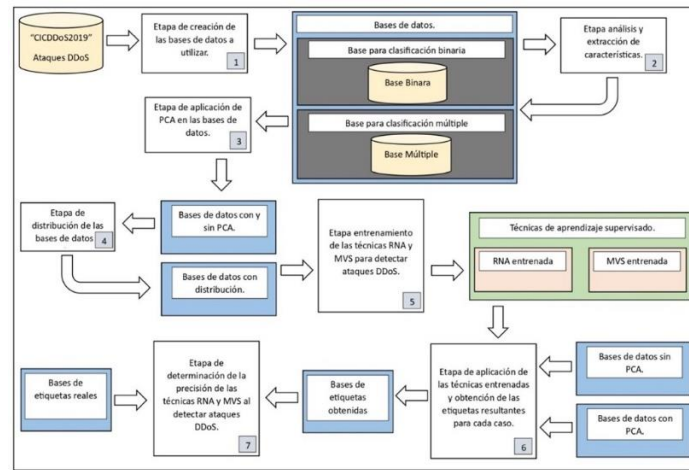


Fig. 1. Diagrama de la metodología.

En [11] proponen un sistema de defensa contra instrucciones y ataques DDoS. Utilizaron el método GRU (Gated Recurrent Units) en el sistema de detección. Se utiliza CICDDoS2019, el resultado promedio de exactitud es de 99.94% que es más alto que otros métodos, como DN, CNN, LSTM y máquinas de soporte vectorial.

### 3. Metodología

En la figura 1 se muestra las distintas etapas de la metodología, en donde los recuadros azules son una representación del conjunto de la base de datos a utilizar tanto para la clasificación binaria, es decir en esta etapa el sistema solamente determina si hay o no un ataque de DDoS, y para la clasificación múltiple, el sistema determina si no hay un ataque o si existe, qué tipo de ataque es el que hay de 8 posibles tipos. Algunas de las etapas importantes son:

La etapa de creación de la base de datos a utilizar tiene el objetivo de generar una nueva base de datos con los tipos de ataques y con los atributos que consideramos más relevantes, por ejemplo, se descartan los atributos IP (protocolo de internet) de origen y destino, puerto de origen y destino. Con la nueva información creada se generan varias bases de datos como son: la de la clasificación binaria donde únicamente hay dos posibles salidas (ataque o no ataque), la de clasificación múltiple en donde existe 9 posibles salidas que no haya ataque o un tipo de ataque en particular, por último, las bases de datos de se obtuvieron aplicando PCA a los datos.

En la etapa de análisis y extracción de características se extraen las características más relevantes con el fin de reducir el tiempo y el nivel de procesamiento de cómputo durante la etapa de entrenamiento. Por último, en la etapa de la precisión se aplica la métrica “label\_ranking\_average\_precision\_score” de Sklearn para determinar la precisión de las técnicas.

Dado que “CICDDoS2019” [2] se encuentra distribuida en bases de datos con ataques DDoS de diferente tipo y cada base cuenta con una pequeña porción de tráfico

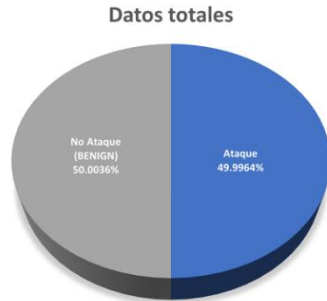


Fig. 2. Distribución de los datos para clasificación binaria.

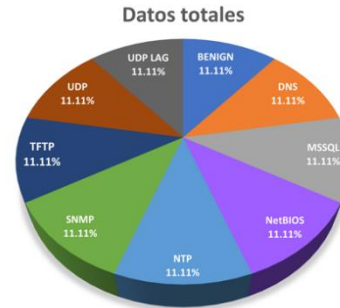


Fig. 3. Distribución de los datos para clasificación múltiple

normal, primero fueron separados los ataques del tráfico normal y posteriormente fueron unidos en una sola base de datos con una cantidad homogénea de datos para cada clasificación.

Lo anterior fue realizado para dos bases de datos, la primera base cuenta con 34,661 datos sobre tráfico normal el cual fue llamado "BENIGN", y con 34,656 datos sobre ataques DDoS (4,332 datos para cada tipo de ataque), esta base de datos fue usada para una clasificación binaria por lo que las etiquetas de los datos tipo "BENIGN" fue cambiada a 0 y las etiquetas de los datos sobre ataques fue cambiada a 1, La figura 2 muestra la distribución de la base binaria.

La segunda base de datos cuenta con 34,661 datos "BENIGN" y 34,661 datos por cada tipo de ataque (311,949 datos en total), esta base de datos fue usada para una clasificación múltiple, donde las etiquetas fueron cambiadas de la siguiente manera: BENIGN: 0, DrDoS\\_DNS: 1, DrDoS\\_MSSQL: 2, DrDoS\\_NetBIOS: 3, DrDoS\\_NTP: 4, DrDoS\\_SNMP:5, TFTP: 6, DrDoS\\_UDP: 7 y UDP-lag: 8, la Figura 3 muestra la distribución de la base múltiple.

#### 4. Trabajo propuesto

En esta sección se muestran las redes propuestas para detectar ataques DDoS en las bases de datos definidas anteriormente.

**Clasificación binaria:** Se propone una RNA para clasificación binaria, la cual es usada con datos sin y con PCA, con la diferencia de que la capa de entrada es de 69 y 21 (estos atributos se eligieron porque la suma de ellos describe la mayor cantidad de varianza, aproximadamente el 98%) neuronas respectivamente. Se trata de un modelo secuencial con una capa de entrada (las neuronas son definidas por los datos con y sin PCA), cinco capas ocultas de 64, 32, 16, 8 y 4 neuronas respectivamente y una capa de salida de dos neuronas, fue usada la función de pérdida "binary\_crossentropy" y el optimizador "adam". También fue usada la técnica Máquinas de soporte vectorial (MSV) con kernel "linear".

**Para la clasificación múltiple:** Se propone una RNA con todos los datos y con datos obtenidos de PCA donde la capa de entrada para datos sin PCA es de 69 y 21 para datos con PCA. Es un modelo secuencial con una capa de entrada (las neuronas son definidas

Valores esperados	Verdadero Positivo	Verdadero negativo	Falso Positivo	Falso Negativo
Real Predicción	Benign Benign	Ataque Ataque	Ataque Benign	Benign Ataque
Técnica Usada	Valores Obtenidos			
RNA	10,411	10,371	5	9
RNA (PCA)	10,409	10,372	4	11
MVS	10,385	10,363	13	35
MVS (PCA)	10,385	10,362	14	35

Fig. 4. Distribución de los datos para clasificación múltiple.

Precisión de las técnicas			
Técnica	Clasificación	PCA	Precisión.
RNA	Binaria	No	0.9996633968070783
		Si	0.9996393537218696
	Múltiple	No	0.9632704671332124
		Si	0.9632609689349022
MVS	Binaria	No	0.9988459319099827
		Si	0.9988218888247740
	Múltiple	No	0.8387490872822809
		Si	0.7958457255136616

Fig. 5. Precisión de las técnicas RNA y MSV.

por los datos con y sin PCA), cinco capas ocultas de 1280, 640, 320, 80 y 20 neuronas respectivamente y una capa de salida de 9 neuronas.

Además, después de las capas de 1280, 640, 320 y 80 neuronas, se hizo un Dropout de 0.35, 0.30, 0.25 y 0.20 respectivamente, con el fin de evitar el sobre entrenamiento y tener mejores resultados, fue usada la función de pérdida "binary\_crossentropy" y el optimizador "ADAM". También fue usada la técnica máquinas de soporte vectorial (MSV) con kernel "poly".

## 5. Resultados

### 5.1 Clasificación binaria

Como se puede observar en la Figura 4 ambas técnicas tienen buenos resultados en una clasificación binaria, además en la figura 5 se puede observar que la RNA propuesta tienen mayor precisión (Siendo estrictos) que la MSV propuesta.

Como podemos observar en la figura 4 y en la figura 5 la RNA propuesta tiene casi los mismos resultados cuando los datos ingresados están tratados con y sin PCA, por lo que surge la pregunta ¿es necesario usar PCA?, cuando los datos tienen PCA el tiempo en que tarda la red en realizar las predicciones baja de 1.111s a 0.8437s y la cantidad de parámetros a calcular se reduce de 7,270 a 4,198, por lo que es mejor usar datos con PCA en la red neuronal, además al tener menos parámetros de entrenamiento se utiliza menos memoria de la computadora.

### 5.2 Clasificación múltiple

Como se puede observar en la figura 6 los mejores resultados están en la técnica RNA sin aplicar PCA a los datos, mientras que los peores resultados se encuentran en

	RNA	RNA (PCA)	MVS	MVS (PCA)
Benign	99.89	99.90	99.68	99.54
DNS	86.79	86.60	79.24	62.45
MSSQL	96.51	96.99	94.32	94.69
NetBios	98.98	99.07	98.18	91.92
NTP	99.71	99.69	95.37	80.11
SNMP	88.35	88.14	73.73	73.10
TFTP	98.64	98.63	46.15	46.18
UDP	98.34	98.38	58.64	58.35
UDP-Lag	95.59	95.38	91.05	86.55

Fig. 6. Comparativa de las precisiones de las técnicas RNA y MVS.

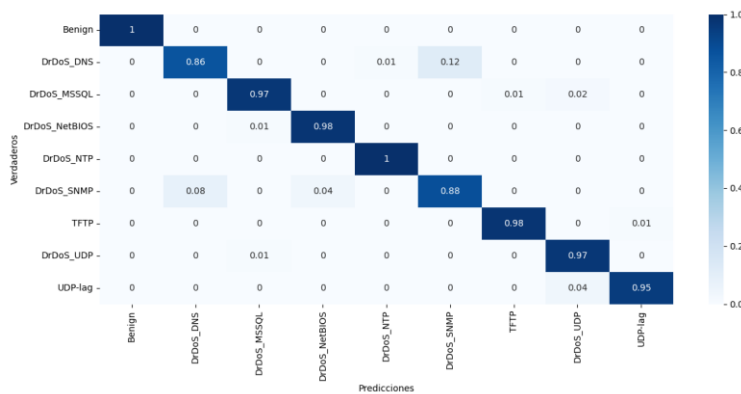


Fig. 7. Matriz de confusión para la técnica RNA.

la técnica MSV, esto se reafirma en la figura 5 donde claramente se observa como RNA es más precisa. Por lo tanto, usar la técnica RNA es mejor opción, sin embargo, vale la pena saber si es necesario usar PCA en los datos o no.

Como se puede observar en la figura 5 la precisión de RNA sin PCA es de 0.96327, mientras que con PCA es de 0.96326, además, los parámetros a calcular sin PCA son 1,142,112, mientras que con PCA son 1,081,952, por lo que hay 60,160 parámetros de diferencia, finalmente el tiempo de predicción sin PCA es de 11.2350 s, mientras que con PCA es de 10.1252 s.

Dados estos resultados, se puede determinar a la técnica RNA como mejor técnica en cuanto a precisión usando datos sin PCA, sin embargo, la diferencia de precisiones cuando se usan datos con y sin PCA es mínima, además, al usar PCA se reduce el tiempo de predicción y los parámetros a calcular, por lo que usar esta red con datos PCA es lo mejor. La figura 7 muestra la matriz de confusión normalizada en clasificación múltiple para la RNA propuesta usando los datos del análisis de PCA.

## 6. Conclusiones

El Instituto Canadiense de Ciberseguridad generó un conjunto de datos sobre ataques DDoS llamado “CICDDoS2019” [2], el cual cuenta con una correcta taxonomía de tráfico en cuanto a características se refiere pues los resultados muestran que es posible

realizar clasificaciones entre ataques y tráfico normal, además cuenta con una gran cantidad de datos para cada tipo de ataque, sin embargo, la cantidad de datos sobre tráfico benigno es muy poca, por lo que sería necesario tener más tráfico de este tipo para un mejor entrenamiento de la red neuronal. Sin embargo, estos datos fueron suficientes para el entrenamiento de las técnicas red neuronal artificial como máquinas de soporte vectorial. En la clasificación binaria demostraron ser eficientes al diferenciar entre tráfico benigno y tráfico con ataques, pues las dos obtuvieron 99% de precisión en sus predicciones. En la clasificación múltiple la RNA propuesta también demostró ser eficiente, pues obtuvo 96% de precisión al clasificar tráfico normal y los 8 tipos de ataques más, siendo estos ataques DNS, MSSQL, NetBIOS, NTP, SNMP, TFTP, UDP y UDP-LAG, por otro lado, la máquina de soporte de vectorial propuesta obtuvo 83% de precisión para las mismas clasificaciones.

En la clasificación binaria se determinó que es necesario aplicar PCA (con 21 componentes) a los datos, pues, aunque las precisiones son casi iguales, para datos sin PCA y con PCA respectivamente, al usar PCA se reducen los parámetros a calcular durante el entrenamiento y se obtiene un tiempo menor al realizar las predicciones. Por otro lado, en la clasificación múltiple se determinó que también es necesario aplicar PCA (con 21 componentes) a los datos, ya que la diferencia de precisiones es mínima, pero al usar PCA se reducen los parámetros a calcular durante el entrenamiento y se obtiene un tiempo menor al realizar las predicciones.

Finalmente, la RNA propuesta muestra ser eficiente en clasificaciones tanto binarias como múltiples, por lo que es posible aplicarla en un entorno real para detección de ataques DDoS.

## Referencias

1. Canadian Institute for Cybersecurity: Canadian Institute for Cybersecurity, Available: <https://www.unb.ca/cic/> (2022)
2. Canadian Institute for Cybersecurity: DDoS Evaluation Dataset (CIC-DDoS'19), (2019). <https://www.unb.ca/cic/datasets/ddos-2019.html> (2024)
3. Satpathy, S.P., Mohanty, S., Kumar, R.: A lightweight dos and ddos attack detection mechanism-based on deep learning. In: 5th International Conference on Computational Intelligence and Networks. pp. 1–6 (2022). DOI: 10.1109/CINE56307.2022.10037402.
4. Elsayed, M.S., Le-Khac, N.A., Dev, S., Jurcut, A.D.: Ddosnet: A deep-learning model for detecting network attacks. In: IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks, pp. 391–396 (2020). DOI: 10.1109/WoWMoM49955.2020.00072.
5. Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A.: Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: IEEE. International Carnahan Conference on Security Technology, pp. 1–8 (2019). DOI: 10.1109/CCST.2019.8888419.
6. Ahuja, V., Kotkar, M., Bhongade, R., Kshirsagar, D.: Reflection based distributed denial of service attack detection system. In: 6th International Conference on Computing, Communication, Control and Automation, pp. 1–5 (2022). DOI: 10.1109/ICCUBEA54992.2022.10011055.
7. Castillo, A., Lineses, A.B., Go, B., Labanan, R., Octaviano, M.: Trojan Malware Detection using ANN, Naïve Bayes and SVM Machine Learning Algorithms. In: IEEE 2nd

- International Conference in Information and Computing Research, pp. 72–76 (2022). DOI: 10.1109/iCORE58172.2022.00033.
8. Chockwanich, N., Visoottiviseth, V.: Intrusion detection by deep learning with tensorflow. In: 21st international conference on advanced communication technology, pp. 654–659, (2019). DOI: 10.23919/ICACT.2019.8701969.
  9. Hou, J., Fu, P., Cao, Z., Xu, A.: Machine learning based DDos detection through NetFlow analysis. In: MILCOM'18 IEEE military communications conference, pp. 1–6 (2018). DOI: 10.1109/MILCOM.2018.8599738.
  10. Roopak, M., Tian, G.Y., Chambers, J.: Deep learning models for cyber security in IoT networks. In: IEEE 9th annual computing and communication workshop and conference, pp. 0452–0457 (2019). DOI: 10.1109/CCWC.2019.8666588.
  11. Assis, M.V., Carvalho, L.F., Lloret, J., Proença, Jr, M.L.: A GRU deep learning system against attacks in software defined networks. *Journal of Network and Computer Applications*, vol. 177, pp. 102942 (2021). DOI: 10.1016/j.jnca.2020.102942.





# Sistema de entrenamiento personalizado para equipos de programación competitiva mediante aprendizaje supervisado

Juventino Aguilar-Correa, Aldonso Becerra-Sánchez,  
Roberto Solís-Robles, Gustavo Zepeda-Valles

Universidad Autónoma de Zacatecas,  
Unidad Académica de Ingeniería Eléctrica,  
México

superjuve@outlook.es, {a7donso, rsolis, gzepeda}@uaz.edu.mx

**Resumen.** Los concursos de programación competitiva abonan en mejorar las técnicas de los programadores que persiguen perfeccionar su habilidad. En búsqueda de coadyuvar los entrenamientos de participantes en competencias de este tipo, se desarrolló una propuesta de arquitectura de sistema y una aplicación web basada en Django que propicia aprendizaje personalizado a los niveles de destreza del alumno. Esta aplicación emplea algoritmos de redes neuronales y regresión logística como apoyo para predecir y obtener los tipos de problemas particulares que el participante requiere realizar, dependiendo de su nivel de soltura en diferentes áreas de programación. Para realizar esta tarea, el módulo emplea información en tiempo real de los resultados de competencias de programadores de Codeforces (intentos de resolver un problema, dificultad del problema, categorías y veredicto de resolución). Los resultados de entrenamiento de los modelos generan altas expectativa de predicciones, ya que los valores de F1-score, precisión y exactitud son robustos.

**Palabras clave:** Aprendizaje máquina, aprendizaje supervisado, programación competitiva.

## Personalized Training System for Competitive Programming Teams Through Supervised Learning

**Abstract.** Competitive programming contests help improve the techniques of programmers who seek to perfect their skills. In search of contributing to the training of participants in competitions of this type, a system architecture proposal and a web application based on Django were developed that promote personalized learning at the student's skill levels. This application uses neural network and logistic regression algorithms as support to predict and obtain the types of particular problems that the participant needs to perform, depending on their level of fluency in different programming areas. To perform this task, the module uses real-time information on the results of Codeforces programmer competitions (attempts to solve a problem, problem difficulty, categories, and resolution verdict). The training results of the models generate high prediction expectations, since the F1-score, precision and accuracy values are robust.

**Keywords:** Competitive programming, machine learning, supervised learning.

## 1. Introducción

El uso de sistemas de software se ha vuelto cada vez más relevante en la sociedad actual, ya que se utilizan en una amplia variedad de campos y aplicaciones, desde el sector empresarial y gubernamental hasta el entretenimiento y la educación. Donde formalmente el software suele ser concebido como un conjunto de programas informáticos, procedimientos, archivos de configuración, documentación y datos relacionados al funcionamiento de un sistema informático [15], [26]. La fabricación de sistemas de software es un proceso complejo que implica la planificación, diseño, construcción, pruebas y mantenimiento de los sistemas de software.

A medida que los sistemas de software han evolucionado, también lo han hecho las técnicas utilizadas para su desarrollo. Una de las disciplinas que ha tomado auge en el proceso de mejoras de desarrollo de software y programación en general es la programación competitiva, la cual implica una actividad mental en la que los participantes resuelven problemas algorítmicos bajo un límite de tiempo [25].

Esta actividad pone a prueba las habilidades de resolución de problemas, el conocimiento de los algoritmos y la capacidad de escribir código eficiente [24]. La popularidad de la programación competitiva ha crecido significativamente en los últimos años (con cientos de participaciones en 1997, hasta varios miles en la actualidad) [14], gracias al surgimiento de competencias a nivel mundial, como la International Collegiate Programming Contest (ICPC).

El ICPC es un concurso de programación algorítmica para estudiantes universitarios, dividido en etapas, que van desde clasificatorios, regionales y la final mundial. Equipos de tres, en representación de su universidad, trabajan para resolver problemas algorítmicos, promoviendo la colaboración, la creatividad, la innovación y la capacidad de desempeñarse bajo presión [13]. Al igual que el ICPC, existen y han existido otras competiciones organizadas por diferentes compañías como Google [10] y Meta [19], las cuales tienen como objetivo la búsqueda de programadores con gran potencial para reclutarlos dentro de sus equipos.

En la actualidad existen algunos sistemas que ayudan a recomendar problemas para practicar en programación competitiva, los cuales permiten a los equipos sin entrenadores experimentados o clubes de algoritmia, mejorar sus habilidades a través de diferentes técnicas. Algunos de estos sistemas permiten que el usuario defina su propia ruta de aprendizaje a modo de gamificación del entrenamiento [6], mientras que otros se enfocan en recomendar y clasificar problemas según su dificultad y temática [3]. También existen investigaciones que utilizan técnicas de machine learning y deep learning para predecir el desempeño de equipos en futuras competencias [2].

Los equipos de programación competitiva de ciertas universidades en México que no cuentan con apoyo institucional, un club de programación o un entrenador experimentado para su formación, se enfrentan al desafío de no saber cómo entrenar de manera efectiva para mejorar sus habilidades y conocimientos en los distintos temas abordados en este tipo de competencias. Esto los coloca en una situación desventajosa para contender en las competencias regionales. Universidades que podemos destacar que cuentan con este tipo de asociaciones de estudiantes son, la Universidad de Guadalajara, con el Club de Algoritmia CUCEI [5]; el Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Monterrey, con el Club de algoritmia

**Tabla 1.** Casos de uso de los requisitos funcionales del sistema.

ID	Usuario	Requerimiento	Descripción
CU01	Competidor	Realizar pruebas de diagnóstico.	Conocer el nivel actual en la categoría escogida y establecer una línea de base.
CU02	Competidor	Solicitar recomendaciones.	Recomendaciones de problemas adaptados a su habilidad en una categoría.
CU03	Competidor	Consultar progreso personal.	Progreso en el tiempo en categorías de elección.
CU04	Admin	Gestionar problemas	Agregar, modificar, consultar o eliminar problemas dentro del sistema.
CU05	Admin	Gestionar categorías	Crear, consultar, modificar y eliminar según sea necesario.
CU06	Admin	Gestionar usuarios	Consultar, agregar, eliminar y modificar usuarios del sistema.

ITESM MTY [16]; la Escuela Superior de Cómputo del Instituto Politécnico Nacional, con su Club de Algoritmia [9] y la Universidad Nacional Autónoma de México, con el Club de Programación Competitiva UNAM [29].

Actualmente existen sitios populares como Codeforces [4] u omegaUp [21], donde los interesados en programación competitiva pueden entrenar resolviendo problemas del gran repertorio con el que estas plataformas cuentan. Ambas separan sus problemas en categorías dependiendo de las estrategias que se utilizan para resolverlos. Tienen jueces automáticos que dan respuesta a las soluciones enviadas por los participantes, y de igual manera permite el acceder a concursos en tiempo real; al igual que revisar concursos anteriores para conocer los problemas expuestos en cada uno.

La falta de apoyo y entrenamiento personalizado para los equipos de programación competitiva puede limitar su desempeño y potencial, debido a que actualmente el resultado de los equipos en las competencias mundiales y regionales depende en gran parte a cómo entrenan y el tratar de resolver muchísimos problemas diferentes [18]. El mejorar la manera en que se entrena para este tipo de competencias no sólo ofrece beneficios directos para la programación competitiva, sino que también provee a sus participantes con la mejora en sus capacidades para resolver problemas, trabajo en equipo y entrega de resultados bajo presión [1].

En este sentido, el aprender a resolver problemas con distintos métodos aumentan las posibilidades de obtener entrevistas de trabajo en empresas como Google, Meta y Amazon, mejorando en las habilidades sobre teoría de números, combinatorias y algoritmos geométricos [28]. El objetivo de este trabajo es proponer una arquitectura y herramienta de software de apoyo a la falta de mecanismos personalizados de entrenamiento para los equipos de programación competitiva. Los resultados de este proyecto pueden tener un gran impacto dentro de las competencias de programación al brindar apoyo para mejorar el desempeño de los equipos y, por tanto, aumentar

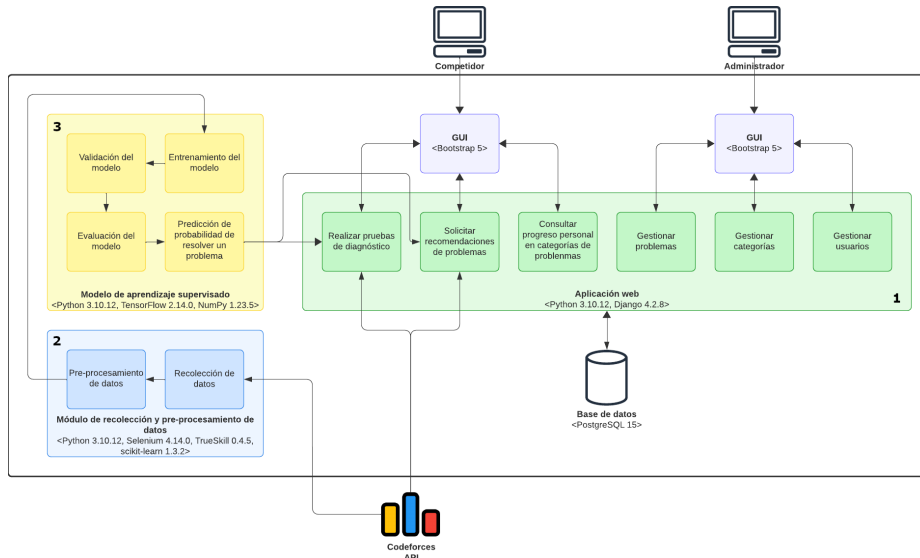


Fig. 1. Arquitectura del sistema acorde a los 3 módulos desarrollados.

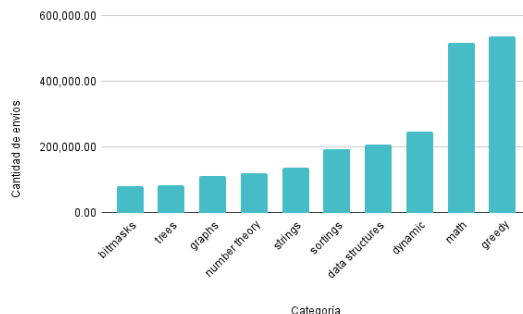
sus posibilidades de éxito en las competencias. El resto del documento se organiza de la siguiente forma, la sección 2 incluye una descripción de trabajos relacionados, mientras que la sección 3 incluye la propuesta de la aplicación desarrollada. En tanto que la sección 4 muestra los resultados obtenidos. Finalmente, la sección 5 comenta las consideraciones finales de conclusiones y trabajo futuro.

## 2. Trabajos relacionados

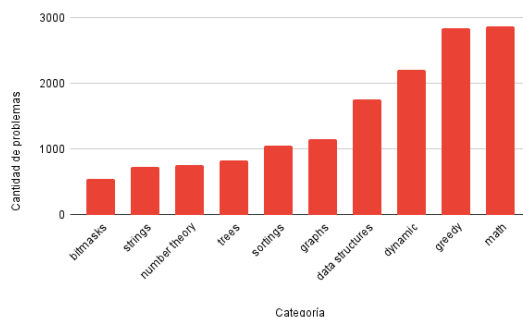
La programación competitiva es una disciplina que ha ganado cada vez más popularidad en los últimos años [14], debido a su capacidad para mejorar el pensamiento crítico y solución de problemas de los equipos de programadores [24]. En este contexto, la inteligencia artificial se puede presentar como una herramienta valiosa para el desarrollo de sistemas de entrenamiento personalizado, permitiendo mejorar el rendimiento de los equipos de programación en competencias.

Recientemente se desarrolló un sistema para generar entrenamientos en programación competitiva que permitiera crear rutas de aprendizaje para estudiantes y utilizar insignias para rastrear su progreso [6], empleando gamificación. Dentro de otro de los sistemas propuestos se generó un módulo capaz de clasificar problemas de programación competitiva a partir de su descripción y categorías, así como recomendar problemas para practicar con base en el historial de soluciones previas del usuario [3]. Este sistema utilizó técnicas de machine learning para realizar la clasificación y recomendación de problemas.

Otro ejemplo de este tipo de proyectos [22] nos presenta un sistema implementado en una universidad que regulariza a los estudiantes de ingeniería en computación utilizando problemas de programación de un juez en línea.



(a) Envíos por categoría.



(b) Problemas por categoría.

**Fig. 2.** Información general del dataset extraído: envíos y problemas.

El sistema recomienda problemas a los estudiantes en función de su puntuación y la de los problemas, utilizando el sistema de clasificación ELO [20]. En este otro caso de estudio [27], el objetivo de la investigación es utilizar una red neuronal convolucional (CNN) [11] para identificar las técnicas empleadas en las soluciones, y recomendar problemas en función de características similares a los problemas previamente resueltos por los usuarios.

Mientras que en otras investigaciones se proponen sistemas de recomendación para programadores novatos en competencias de programación en línea. En ello se utiliza filtrado colaborativo para encontrar problemas similares y recomendarlos a los usuarios [31]. De manera similar, también existe otro sistema de recomendación de problemas, utilizando el algoritmo de vecinos más cercanos (KNN) [12], para encontrar usuarios similares y realiza recomendaciones basadas nuevamente en el filtrado colaborativo y la similitud de contenido [23].

Estos trabajos han abordado la problemática de la programación competitiva y la falta de entrenamiento personalizado, ofreciendo sistemas basados en inteligencia artificial y otras técnicas que permiten recomendar problemas adecuados al nivel y temas de interés de los programadores.



(a) Inicio de sesión con credenciales.



(b) Creación de cuenta.

**Fig. 3.** Pantallas iniciales del sistema.

### 3. Esquema propuesto

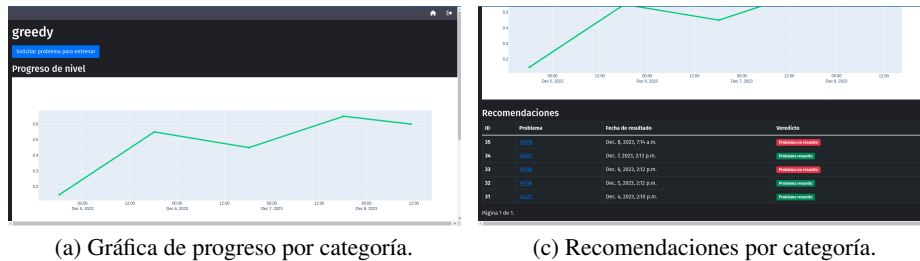
Para el desarrollo de la propuesta, se empleó la metodología de prototipo, que es un método de desarrollo de sistemas en el que se construye un prototipo, se prueba y después se retrabaja según sea necesario hasta que se obtenga un resultado aceptable del cual se puede desarrollar el sistema completo [26, 30]. La fases de esta metodología incluyen: i) definición de objetivos, ii) definición de funcionalidad, iii) diseño y desarrollo del prototipo, y iv) evaluación del prototipo.

#### 3.1. Definición de los objetivos del prototipo

Los objetivos representan las funcionalidades clave que los competidores (principales usuarios del sistema) podrán acceder y utilizar en el sistema, además del experimento necesario para medir la eficacia del prototipo. Estos incluyen: i) gestionar problemas, categorías y usuarios en el sistema; ii) recomendar problemas a los competidores que se adapten a su habilidad actual en la categoría en la que estén entrenando; iii) realizar diagnósticos iniciales a los competidores para evaluar su nivel de habilidad en las categorías de problemas en las que quieran entrenar; iv) proporcionar un seguimiento visual del progreso del competidor en sus categorías de interés a lo largo del tiempo; y v) experimento para evaluar la eficacia del sistema.

#### 3.2. Definición de la funcionalidad

El sistema se centra en ofrecer una experiencia de entrenamiento para equipos de programación competitiva.



(a) Gráfica de progreso por categoría.

(c) Recomendaciones por categoría.

**Fig. 4.** Progresos y recomendaciones de usuario acorde a su evolución.

Los miembros de estos equipos recibirán recomendaciones personalizadas de problemas para su práctica, así como realizar un seguimiento de su progreso a través de estadísticas detalladas centradas en los temas de algoritmia de su interés. La recomendación de problemas se llevará a cabo mediante un enfoque de aprendizaje supervisado, seleccionando aquellos para los cuales el modelo haya predicho una probabilidad intermedia de resolución. Esta estrategia busca proporcionar retos adecuados a los usuarios, fomentando así su mejora continua. La Tabla 1 presenta los casos de uso (CU) que forman parte de la funcionalidad del sistema.

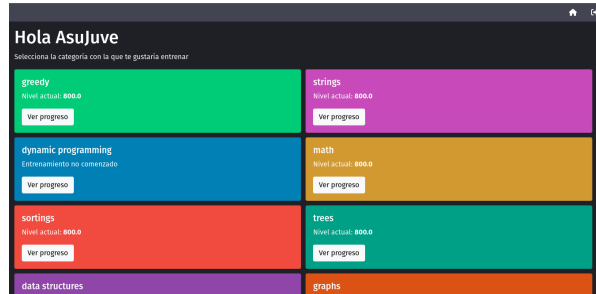
### 3.3. Desarrollo

**Diseño del prototipo.** El sistema se estructura en tres módulos esenciales (ver Fig. 1): i) una aplicación web para competidores y administradores; ii) un módulo de recolección y pre-procesamiento de datos; iii) y un modelo de aprendizaje supervisado.

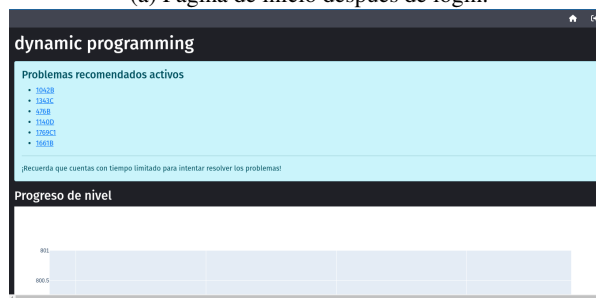
**Módulo de aplicación web.** En el desarrollo de la aplicación web se ha empleado el framework Django [7], una herramienta que facilita la creación de aplicaciones web eficientes en Python. Este framework sigue el patrón de diseño Modelo-Vista-Controlador (MVC), donde la base de datos PostgreSQL y su interacción con el controlador nos define el modelo, la GUI (Front-end con Bootstrap) nos define la Vista, y los módulos de gestión de manipulación de datos y consultas (módulos 1, 2 y 3) nos define el controlador.

En cuanto a la gestión de datos, se eligió PostgreSQL como sistema de gestión de bases de datos relacional. Para comprender la manera en la que se representa la habilidad de un competidor dentro de la tabla "Nivel", es necesario hablar de TrueSkill. TrueSkill es un sistema de ranking basado en habilidades desarrollado por Microsoft Research, diseñado originalmente para evaluar el rendimiento de jugadores en Xbox Live.

Este sistema modela la habilidad de un jugador con dos parámetros clave:  $\mu$  ( $\mu$ ), que representa la habilidad promedio, y  $\sigma$  ( $\sigma$ ), que mide la incertidumbre asociada a esa habilidad. La combinación de  $\mu$  y  $\sigma$  genera una curva de creencia sobre la habilidad del jugador. Cuando  $\sigma$  es alto, el sistema tiene menos certeza sobre la verdadera habilidad, reflejando un rango más amplio de posibles habilidades. En contraste, un  $\sigma$  bajo indica una mayor certeza de que la habilidad del jugador está cerca del promedio. TrueSkill utiliza este enfoque en partidas de juego para ajustar el nivel de habilidad de los jugadores en función de los resultados.



(a) Página de inicio después de login.



(b) Recomendaciones activas de usuario.

Fig. 5. Pantallas funcionales del sistema una vez ingresado el usuario.

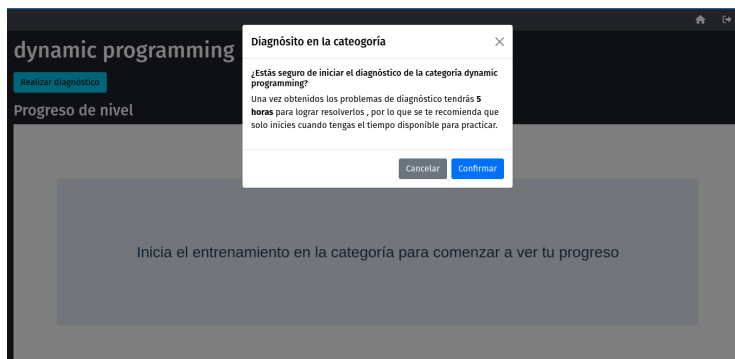
Durante la implementación y adaptación de TrueSkill dentro del sistema de entrenamiento, se decidió que los problemas actúen como "jugadores", y la resolución de problemas se equipara a participar en una "partida", permitiendo que los competidores mejoren su nivel a medida que resuelven desafíos más complejos. Estos niveles son registrados, incorporando  $\mu$  y  $\sigma$  para mantener una representación dinámica y precisa de la habilidad del competidor en categorías específicas. El valor  $\mu$  aumenta en caso de que se haya resuelto el problema y disminuye en el caso contrario. Este proceso dinámico asegura una adaptación continua del sistema TrueSkill a medida que los competidores participan en más "partidas" de resolución de problemas.

### Módulo de recolección y procesamiento de datos

**Recolección de datos.** En este módulo se llevaron a cabo varias etapas para obtener un conjunto de datos significativo y limpio que sirviera como base para el entrenamiento del modelo. Primeramente se utilizaron técnicas de web scraping con Selenium para extraer 6000 nombres de usuarios de Codeforces. Estos usuarios abarcaban diferentes valores de ranking dentro de la plataforma.

Una vez con los nombres de usuarios, se empleó la API de Codeforces para obtener todos sus envíos realizados. Cada envío representa un intento de resolver un problema y contiene información valiosa; como la dificultad del problema, sus categorías y el veredicto sobre si el problema fue resuelto o no.





(a) Confirmación de inicio de diagnóstico.



(b) Confirmación de recomendación.

Fig. 6. Modales de confirmación dependiendo de la acción realizada.

**Pre-procesamiento de datos.** Se le dio un formato correcto a los campos de los envíos para facilitar el entrenamiento del modelo. Esto implicó transformar la información de manera que fuera adecuada para su análisis y uso en el sistema, como normalizar y categorizar campos. Además del uso de los campos obtenidos mediante la API de Codeforces, se llevó a cabo la creación y actualización del campo `user_rating`.

Este campo representa el nivel del usuario y se calculó utilizando el sistema de ranqueo TrueSkill, del cuál se habló en el módulo anterior. El procedimiento se llevó a cabo tomando en cuenta que los envíos obtenidos desde la API de Codeforces están ordenados cronológicamente. Después de cada envío, se actualizó el nivel del usuario. Este enfoque permitió reflejar los cambios en la habilidad del usuario a lo largo del tiempo, considerando sus interacciones con problemas de distintas categorías y dificultades.

**Filtrado de envíos.** Para optimizar la calidad del conjunto de datos se realizaron algunas condiciones de filtrado. Por un lado, solo se consideró el último envío por cada usuario para un problema específico, evaluando si finalmente lograron resolver el problema. También se eliminaron envíos de problemas en los que el usuario no realizó, al menos, tantos intentos como el promedio de intentos para resolver ese problema. Esto se hizo para centrarse en problemas que los usuarios realmente intentaron resolver.

**Tabla 2.** Métricas obtenidas en los modelos de predicción.

Modelo	Precisión	Exactitud	Sensibilidad	F1	Matriz de confusión
Red neuronal	0.9004	0.9004	1	.95	[257, 24312] [152, 221078]
Regresión logística	0.9002	0.9002	1	.95	[152, 24417] [93, 221137]

**Resultados del proceso.** Después del proceso, se obtuvo un conjunto de datos limpio y significativo que consta de 1 228 994 envíos realizados por 6 000 usuarios en relación con 8 299 problemas. En la Fig. 2a se muestra la cantidad de envíos por categoría, y en la Fig. 2b la cantidad de problemas por categoría.

### Modelo de aprendizaje supervisado

**Objetivo de predicción.** El objetivo central del modelado de datos en este proyecto es crear un sistema de recomendación de problemas eficaz y personalizado para los competidores de programación competitiva. El sistema debe ser capaz de proporcionar recomendaciones de problemas basadas en la probabilidad de que un competidor resuelva un problema en función de su nivel de habilidad en las categorías del problema, la cual a su vez se basa en el historial de envíos y su desempeño en las mismas categorías.

**Procesamiento de datos.** Para lograr los objetivos de predicción, se utilizó el conjunto de envíos de soluciones generado en el módulo del sistema anterior. Tomando el campo de *verdicto* como la variable objetivo, siendo este la probabilidad de que un usuario, con cierto nivel de habilidad, solucione un problema con un cierto nivel de dificultad y perteneciente a categorías de algoritmia específicas. Para poder utilizar el conjunto de datos pre-procesado dentro de los modelos a entrenar, se utilizaron tecnologías como Pandas para la manipulación de datos, y scikit-learn para la división del conjunto en datos de entrenamiento y datos de prueba para evaluar el rendimiento del modelo en datos no vistos.

**Métodos de modelado.** Para lograr el objetivo de un sistema de recomendación preciso y personalizado, se consideraron dos enfoques de modelado principales: redes neuronales y regresión logística. La idea de utilizar redes neuronales en el sistema radica en su capacidad para modelar relaciones complejas y no lineales en los datos, lo que es esencial para capturar los patrones que determinan si un competidor es probable que resuelva un problema específico [8].

Para construir y entrenar la red neuronal se utilizó el modelo Sequential de la librería TensorFlow, donde se trabajó con hiperparámetros tales como el optimizador (Adam, en nuestro caso), función de pérdida (entropía cruzada, en nuestro caso), funciones de activación y tasa de aprendizaje (0.01 en este ejemplo); esto para intentar mejorar el rendimiento del modelo y obtener mayor precisión. La estructura de la red neuronal (completamente conectada secuencial) está definida por 13 neuronas de entrada en la capa 1, dos capas ocultas con 200 y 100 neuronas (relu) respectivamente, mientras que la capa de salida consta de una neurona sigmoidea.

Todo este proceso de entrenamiento realizado en 10 épocas con un tamaño de batch de 32. El total de elementos del dataset fue 2 538 392 muestras, divididas en 80 % para entrenamiento y 20 % para pruebas. Por otro lado, la regresión logística es un método eficiente y ampliamente utilizado para modelar la probabilidad de un resultado discreto, como sí o no, verdadero o falso [17]. En este caso, se está interesado en clasificar si un competidor logrará resolver un problema específico o no y en qué medida, lo que se ajusta perfectamente a la naturaleza binaria de la regresión logística. Para la implementación de la regresión logística se utilizó el modelo LogisticRegression (con solucionador *liblinear*), proporcionado por la librería *scikit-learn*.

**Experimentación.** Para evaluar la eficacia del prototipo se propone llevar a cabo un experimento que involucre dos grupos de equipos de programación competitiva. Por un lado, habría un grupo que se entrene utilizando el sistema desarrollado, mientras que el otro grupo seguiría con su entrenamiento habitual. El escenario de prueba se desarrollaría a lo largo de diversas competencias de programación competitiva tipo RPC, similares a las competencias ICPC.

Durante una competencia reciente, se recopiló información sobre el rendimiento de todos los equipos que podrían participar en el experimento. Se registraron datos como la cantidad de problemas resueltos, el puntaje obtenido y la posición en la tabla de clasificación. Entre las diversas competencias subsecuentes, los equipos continuarían su entrenamiento de acuerdo con los grupos a los que pertenecen. La subsecuentes competencias proporcionarían datos adicionales sobre el desempeño de los equipos. La evaluación de la eficacia del sistema se llevaría a cabo comparando los resultados continuamente.

Esta comparación permitirá determinar si los equipos que utilizaron el sistema mostraron una mejora significativa en comparación con el grupo de control. Es importante destacar que la comparación se realizaría a nivel individual, considerando que los equipos no tienen un nivel de habilidad inicial homogéneo. Además de los datos recopilados de las competencias, se tomará en cuenta el tiempo y la dedicación al entrenamiento por parte de cada equipo. Esta metodología proporcionará una visión integral de la eficacia del sistema en el contexto específico de la programación competitiva, permitiendo identificar el impacto del sistema en el rendimiento individual de los equipos a lo largo del tiempo.

### **3.4. Evaluación**

Durante la primera iteración del desarrollo del prototipo se enfocó en implementar la gestión de problemas, categorías y usuarios en el sistema. Esto le permite a los usuarios administradores el registro y modificación de instancias de estas entidades en el sistema. Para la segunda iteración se trabajó en la generación de predicciones, para lo que fue necesario el pre-procesamiento de los datos. Posterior al procesamiento de los datos limpios, se implementaron las funcionalidades de recomendación de problemas y realización de diagnósticos a los competidores, debido a que ambas utilizan la predicción de probabilidad de un competidor de resolver un problema. Se usaron modelos tanto de red neuronal y regresión logística, y el modelo final fue el que obtuvo el mejor rendimiento.

Durante la tercera fase se agregaron al sistema los elementos visuales para el seguimiento del progreso en categorías de interés. Esto incluyó el graficar el nivel de los competidores a través del tiempo con base a los resultados de los problemas recomendados, además de una tabla con información de cada recomendación.

Además del experimento, para obtener una comprensión completa de la experiencia de los competidores, se ha diseñado una encuesta para obtener retroalimentación de los usuarios. Esta encuesta se enfocará en la facilidad de uso de la aplicación web, la presencia de posibles problemas técnicos en el sistema y la adecuación de los problemas recomendados. Los participantes, además de contestar a las preguntas, podrán proporcionar opiniones sobre su experiencia y sugerencias para posibles mejoras.

## **4. Discusión de resultados**

### **4.1. Aplicación web del sistema**

Para la autenticación de usuarios se tiene la página de inicio de sesión, Fig. 3a, y la página de creación de cuenta, Fig. 3b. En esta última se incluye un campo para el nombre de usuario de Codeforces, el cual se confirma mediante el uso de la API de la misma plataforma. Esto es requerido para que el sistema pueda verificar las soluciones de los problemas recomendados hechas por el usuario, permitiendo actualizar el nivel de habilidad en las categorías a entrenar. Dentro de la página inicial del sitio, Fig. 3a, se encuentran listadas las categorías disponibles para practicar, ahí se muestra el nivel actual en cada una en caso de haber comenzado el entrenamiento en esa categoría.

Dentro de la página de cada categoría se encuentra una gráfica que muestra el nivel del usuario en esa categoría a través del tiempo, Fig. 4a. También se observa una tabla con la información de cada una de los problemas que se le han recomendado al usuario, Fig. 4b. En caso de que el usuario aún no haya comenzado el entrenamiento de una categoría, tendrá la opción de realizar un diagnóstico inicial para conocer su nivel en la categoría. Al seleccionar esta opción se abre un modal, Fig. 4a, para confirmar la elección. Esto último debido a que se tiene que tener en cuenta que el usuario tendrá solo 5 horas para resolver los problemas que se le recomienden en el diagnóstico. La aplicación verificará si se lograron resolver los problemas antes del tiempo límite utilizando la API de Codeforces.

Una vez que se ha completado el diagnóstico en una categoría, se pueden solicitar más problemas para entrenar. Similar al modal de confirmación de diagnóstico, se abre también un modal para confirmar esta acción, Fig. 5b, solo que ahora para una sola recomendación de problema; en este caso tendremos 2 horas para solucionar el problema. El sistema de igual manera utilizará la API de Codeforces para verificar que se haya resuelto el problema. Cuando existen recomendaciones de problemas para las que aún queda tiempo de resolverlas, estas se mostrarán en la parte superior de la página, a manera de enlaces para que el competidor pueda redirigirse a Codeforces a intentar resolverlas, Fig. 5b.

### **4.2. Modelos de aprendizaje supervisado**

Se utilizó el conjunto de datos pre-procesados para el entrenamiento de dos modelos de aprendizaje supervisado, una red neuronal y un modelo de regresión logística.

Las métricas obtenidas de cada modelo se pueden ver en la Tabla 2. Tras evaluar exhaustivamente el rendimiento de dos modelos, la red neuronal y la regresión logística, con base a diversas métricas clave, ambos demostraron un desempeño sobresaliente. Ambos modelos exhibieron una alta precisión, exactitud y sensibilidad, con un F1-score que refleja un equilibrio notable entre precisión y exactitud.

Las matrices de confusión revelan un bajo número de falsos positivos y falsos negativos para ambas arquitecturas. Sin embargo, al analizar minuciosamente los resultados, se observa que la red neuronal presenta una ligera ventaja en términos de precisión, con un valor ligeramente superior. En consecuencia, la elección provisional recae en la red neuronal, respaldada por su leve mejora en precisión, una métrica crítica para este contexto particular.

## **5. Conclusiones y trabajo futuro**

Con el fin de apoyar en el desafío de desconocer la forma efectiva de entrenarse para perfeccionar las habilidades de los concursantes de programación competitiva en los diversos temas abordados en este ámbito, se desarrolló un sistema de recomendación de problemas de programación. Esta propuesta utiliza aprendizaje supervisado para predecir la probabilidad que tiene un usuario de resolver un ejercicio, y a partir de ahí recomendar actividades con una probabilidad intermedia de resolución, proporcionando retos que ayuden a mejorar a los usuarios.

Se ofrece así la oportunidad de obtener sugerencias de problemas para entrenar a partir de las temáticas de algoritmos en las que se quiera mejorar. Para el entrenamiento del modelo de aprendizaje supervisado, se utilizó la información de envíos de soluciones de problemas realizadas por miles de usuarios en la plataforma de Codeforces. En este sentido, para la ponderación del nivel de habilidad de los usuarios se trabajó con el sistema de clasificación TrueSkill.

Como trabajo futuro, se requiere ir aplicando el modelo sugerido en diversas competencias generadas a lo largo del año, lo cual irá proporcionando elementos que brinden argumentos de su nivel de utilidad. A la vez se pueden arrojar factores que muestren si los algoritmos empleados pueden mejorarse o substituirse.

## **Referencias**

1. ACM MPSTME: Benefits of Competitive Programming | LinkedIn (2022) [https://www.linkedin.com/pulse/benefits-competitive-programming-association-for-computing-machiner/?trk=organization-update-content\\_share-article](https://www.linkedin.com/pulse/benefits-competitive-programming-association-for-computing-machiner/?trk=organization-update-content_share-article)
2. Alnahhas, A., Mourtada, N.: Predicting the performance of contestants in competitive programming using machine learning techniques. *Olympiads in Informatics*, pp. 3–20 (2020) doi: 10.15388/ioi.2020.01
3. Avkirkar, P.: FYProject - competitive coding problem classifier and problem recommendation (2023) [https://github.com/ParasAvkirkar/-Competitive-Coding-Problem-Classifer-and-Recommendier/blob/2a93e19905f3922b85a0f0674ef56334ab4f57d1/final\\_report.pdf](https://github.com/ParasAvkirkar/-Competitive-Coding-Problem-Classifer-and-Recommendier/blob/2a93e19905f3922b85a0f0674ef56334ab4f57d1/final_report.pdf)
4. Codeforces: Codeforces (2023) <https://codeforces.com/>
5. CUCEI: Club de Algoritmia CUCEI (2023) <http://algoritmia.club/#/home>

6. Di-Mascio, T., Laura, L., Temperini, M.: A framework for personalized competitive programming training. In: 17th International Conference on Information Technology Based Higher Education and Training, IEEE, pp. 1–8 (2018) doi: 10.1109/ITHET.2018.8424620
7. Django: Django project (2023) <https://www.djangoproject.com/>
8. Edgar, T. W., Manz, D. O.: Chapter 4 - Exploratory Study. In: Research Methods for Cyber Security, Syngress, pp. 95–130 (2017), doi: 10.1016/B978-0-12-805349-2.00004-2
9. ESCOM: Clubs Académicos Estudiantiles - ESCOM (2023) <https://www.escom.ipn.mx/htmls/escomunidad/clubs.php>
10. Google's Coding Competitions: Google's Coding Competitions - Code Jam, Hash Code and Kick Start (2023) <https://codingcompetitions.withgoogle.com/>
11. IBM: What are convolutional neural networks? | IBM (2023) <https://www.ibm.com/topics/convolutional-neural-networks>
12. IBM: What is the k-nearest neighbors algorithm? | IBM (2023) <https://www.ibm.com/topics/knn>
13. ICPC: ICPC (2018) <https://icpc.global/>
14. ICPC: ICPC FACT SHEET. Technical report (2021) <https://icpc.global/community/history/Factsheet2020.pdf>
15. IEEE: IEEE standard glossary of software engineering terminology (1990) doi: 10.1109/IEEESTD.1990.101064
16. ITESM MTY: Club de algoritmia ITESM MTY | Facebook (2023) <https://www.facebook.com/algorithmiaITESM/>
17. Kiang, M. Y.: Neural networks. Encyclopedia of Information Systems, pp. 303–315 (2003) doi: 10.1016/B0-12-227240-4/00121-0
18. Liu, R.: Training ICPC teams: A technical guide (2014)
19. Meta: Meta hacker cup (2022) <https://www.facebook.com/codingcompetitions/hacker-cup/>
20. Mittal, R.: What is an ELO rating? (2022) <https://medium.com/purple-theory/what-is-elo-rating-c4eb7a9061e0>
21. omegaUp: Welcome – omegaUp (2023) <https://omegaup.com/>
22. Prisco, A., dos-Santos, R., Nolibos, A., Botelho, S., Tonin, N., Bez, J.: Evaluating a programming problem recommendation model - a classroom personalization experiment. In: 2020 IEEE Frontiers in Education Conference (FIE), pp. 1–6 (2020) doi: 10.1109/FIE44824.2020.9274028
23. Ramesh, P., Subramanian, K.: Context-aware practice problem recommendation using learners' skill level navigation patterns. Intelligent Automation & Soft Computing, vol. 35, pp. 3845–3860 (2023) doi: 10.32604/iasc.2023.031329
24. Sharma, R.: Unlocking the world of competitive programming: A comprehensive guide (2023) <https://emeritus.org/blog/coding-competitive-programming/>
25. Sinza, Y., Oliva, J., Guerrero, S.: Análisis de los componentes relacionados en programación competitiva: Un mapeo sistémico de la literatura. Revista politécnica, vol. 19, no. 38 (2023)
26. Sommerville, I.: Software engineering. Addison-Wesley (2010)
27. Sudha, S., Arun Kumar, A., Muthu Nagappan, M., Suresh, R.: Classification and recommendation of competitive programming problems using CNN. In: International Conference on Intelligent Information Technologies, Springer, pp. 262–272 (2018) doi: 10.1007/978-981-10-7635-0\_20
28. Tangri, A.: What are the benefits of learning competitive programming? (2019) <https://www.quora.com/What-are-the-benefits-of-learning-competitive-programming>
29. UNAM: Club de Programación Competitiva UNAM | Mexico City | Facebook (2023) <https://www.facebook.com/clubpumasmas>
30. Weitzenfeld, A., Guardati, S.: Capítulo 12: Ingeniería de software: el proceso para el desarrollo de software, Introducción a la Computación. CENGAGE Learning (2007)

31. Yera Toledo, R., Caballero Mota, Y., Martínez, L.: A recommender system for programming online judges using fuzzy information modeling. *Informatics*, vol. 5, no. 2, pp. 17 (2018)  
doi: 10.3390/informatics5020017





# Detección de micro fraudes en transacciones de tarjetas de crédito utilizando aprendizaje automático: Generación de la base de datos y métodos de detección

Jaime Alberto Quiñones-Beltrán<sup>1</sup>, Karina M. Figueroa-Mora<sup>1</sup>,  
Juan Pablo Maldonado-López<sup>2</sup>

<sup>1</sup> Universidad Michoacana de San Nicolás de Hidalgo,  
México

<sup>2</sup> University of New York in Prague,  
Czechia

karina.figueroa@umich.mx, pmaldonado@unyp.cz

**Resumen.** Los micro fraudes en Tarjetas de Crédito (carding en inglés) han ido en aumento en los últimos años, en México se estima que el 12% de las tarjetas han sido comprometidas. Sin embargo, a pesar de ser un tipo de problema muy interesante para su estudio, existen limitadas bases de datos con este tipo de información. Lo cual es justificado pues debe ser prioritaria la seguridad de la información de los clientes para las compañías bancarias. Por lo tanto, en este artículo se propone una herramienta que permite construir bases de datos sintéticas con información de transacciones en tarjetas bancarias, donde un porcentaje muy pequeño de ellas son consideradas fraudulentas. La construcción de estas bases de datos sintéticas refleja ciertas hipótesis sobre el comportamiento de posibles clientes, lo cual es reflejado en este artículo considerando diversas distribuciones de transacciones en clientes. Con estas bases de datos sintéticas se analizaron con técnicas de aprendizaje de máquina buscando determinar si fuera posible la detección de dichas transacciones fraudulentas.

**Palabras clave:** Detección de fraudes, aprendizaje de máquina, finanzas.

## Detection of Micro Frauds in Credit Card Transactions Using Machine Learning: Database Generation and Detection Methods

**Abstract.** Micro frauds in Credit Cards (carding in English) have been increasing in recent years. In Mexico, it is estimated that 12% of cards have been compromised. However, despite being a very interesting problem for study, there are limited databases with this type of information, which is justified since the security of customer information must be a priority for banking companies. Therefore, this article proposes a tool that allows the building of synthetic databases with information on bank card transactions, a small fraction of which

are considered fraudulent. The construction of these synthetic databases reflects certain hypotheses about the behavior of potential clients, as evidenced by the consideration of various distributions of client transactions in this article. These synthetic databases were then subjected to a meticulous analysis using advanced machine learning techniques, ensuring the validity and reliability of the research findings, to determine if it was possible to detect these fraudulent transactions.

**Keywords:** Fraud detection, machine learning, finance.

## **1. Introducción**

Las tarjetas de crédito (TC) se han convertido en uno de los principales métodos de pago usados en el mundo, lo que las ha llevado a ser un objetivo común por defraudadores. Se sabe de antemano que la detección de patrones en fraudes en este tipo de transacciones es deseable pero muy difícil por diversas causas. Una de estas es el hecho de tener un enorme volumen de transacciones diarias que hace que la detección de fraudes sea casi imposible para un humano. En [1] muestran que las pérdidas globales durante el año 2022 por fraudes a tarjetas de pago ascienden a los 34 mil millones de dólares.

Existen diversos tipos de fraudes, entre ellos: el robo de la tarjeta física, la clonación, el robo de los datos para su uso en tiendas virtuales, etc. [5]. Un tipo de fraude casi indetectable por el mismo dueño de las tarjetas son los micro fraudes (carding) pues se hacen pasar por transacciones normales de cantidad moderada y realizados en sitios usuales. Por ejemplo, el costo de la entrada a una función de cine con una descripción relacionada a tiendas comunes. Al menos el 12 % de los mexicanos han sido víctimas de un ataque de este estilo [2].

Otro gran problema es que las compañías bancarias no revelan bases de datos con este tipo de transacciones por razones de seguridad. Por lo tanto, estudiar el problema es prácticamente imposible. Básicamente solo se tiene acceso a una base de datos en el sitio web de Kaggle [3], sin embargo, ésta tiene distintos tipos de fraudes y los datos son información resultante de algún PCA, además la información solo es de una institución bancaria. En este artículo se presenta una herramienta de creación de datos sintéticos con información bancaria y con registros marcados como micro fraudes. Además se muestra el análisis con técnicas de aprendizaje de máquina empleadas sobre ellos.

## **2. Antecedentes**

La única base de datos existente y que además es ampliamente usada se puede ver en [3]. Esta base de datos tiene 284,807 registros de los cuales solo el 0.172 % (492) son fraudes con transacciones en septiembre de 2013 hechas por clientes europeos. La evolución del Carding durante la pandemia del 2020 ha sido enorme. Las debilidades de esta base de datos son evidentes, en 2013 los patrones de fraudes han evolucionado fechas recientes y el número de transacciones diarias por cliente es completamente distinto a lo popularizado de hoy en día. Esta base de datos solo contiene información

---

**Algorithm 1** Generación de la base de datos

---

```

function GENERAR_DATASET( $c=n\_clientes$ ,  $w=n\_webs$ ,  $f=n\_días$ ,  $t=fecha\_inicio$ )
    tabla_perfil_clientes  $\leftarrow$  GENERAR_TABLA_PERFIL_CLIENTES( $n\_clientes$ )
    tabla_perfiles_web  $\leftarrow$  GENERAR_TABLA_PERFILES_WEB( $n\_webs$ )
    transacciones_df  $\leftarrow$  TRANSACCIONES(cliente, tabla_perfil_clientes, tabla_perfiles,
     $f=n\_días$ ) return transacciones_df
end function
carding_victimas  $\leftarrow$  RANDOM.CHOICE(clientes, tamaño=12 %( $clientes$ ))
transacciones_df['CARDING']  $\leftarrow$  0
for clientes en carding_victimas do
    4_transacciones  $\leftarrow$  RANDOM.CHOICE(transacciones_del_cliente, tamaño=4)
    para cada transacción en 4_transacciones hacer
        transacciones_df[transacción, 'cantidad']  $\leftarrow$  GENERAR_FRAUDE
        transacciones_df[transacción, 'CARDING']  $\leftarrow$  1
end for

```

---



---

**Algorithm 2** Generación de las transacciones

---

```

function TRANSACCIONES(cliente, tabla_perfil_clientes, tabla_perfiles,  $f=n\_días$ ).
    for cliente en tabla_perfil_clientes do
        GENERAR_TABLA_TRANSACCIONES(webs,  $n\_días$ )
    end for
end function

```

---

numérica la cual es resultado de una transformación PCA. Los autores argumentan razones de confidencialidad para no poner disponibles los datos originales ni el contexto de las 28 características con las que cuenta. La únicas características no procesadas son el tiempo y la cantidad. Los resultados sobre esta base de datos presentado son casi del 100 % en precisión y recall.

Finalmente, dada la construcción de la base de datos, ésta no permite concentrarse en un tipo específico de fraude. En [4] el autor describe una revisión de la aplicación de modelos de Aprendizaje de Máquina en la detección de fraudes en tarjetas de crédito a la base de datos del sitio web de Kaggle [3]. El autor concluye que a pesar de haber comparado nueve técnicas no asegura tener un rendimiento óptimo.

Los autores en [5] presentan una revisión de la literatura sobre el uso de modelos de Aprendizaje de Máquina y Aprendizaje Profundo en ciberfraudes con tarjetas de crédito. En este trabajo se examinaron 181 artículos publicados entre 2019 y 2021, donde 108 utilizaban Aprendizaje de máquina, 34 Aprendizaje Profundo y 39 una combinación de ambos.

La mayoría de los algoritmos aplicados fueron supervisados, destacando Random Forest, SVM y Regresión logística. El artículo concluye destacando la importancia de que los bancos proporcionen conjuntos de datos públicos con diversos tipos de fraudes para investigaciones futuras.

En la mayoría de los trabajos aseguran no poder compartir los datos por cuestiones de seguridad. Finalmente, en [6] se presenta una metodología para detectar fraudes en tarjetas de crédito. Éste incluye la creación de datos sintéticos y la predicción de fraudes mediante técnicas de Aprendizaje de Máquina y profundo.

---

**Algorithm 3** Generación de la tabla de perfil de clientes

---

```
function  GENERAR_TABLA_PERFIL_CLIENTES( $c=n\_clientes$ ,  $w=n\_webs$ ,  $f=n\_días$ ,  
 $t=fecha\_inicio$ )  
  for cliente_id en rango( $c$ ) do  
    monto_promedio  $\leftarrow$  DISTRIBUCIÓN_UNIFORME(5, 100)  
    desv_est_monto  $\leftarrow$  monto_promedio / 2 num_transacciones_promedio_por_día  $\leftarrow$   
    DISTRIBUCIÓN_UNIFORME(1, 5)  
    añadir [cliente_id,  
            monto_promedio, desv_est_monto,  
            num_transacciones_promedio_por_día]  
  end for  
end function
```

---

---

**Algorithm 4** Generación de la tabla de perfil web

---

```
function  GENERAR_TABLA_PERFIL_WEB( $c=n\_clientes$ ,  $w=n\_webs$ ,  $f=n\_días$ ,  $t=fecha\_inicio$ )  
  for web_id en rango( $n\_webs$ ) do  
    porcentajes  $\leftarrow$  %Porcentajes de las categorías más compradas %  
    categoría  $\leftarrow$  ELECCIÓN_ALEATORIA(porcentajes)  
    categoría_id  $\leftarrow$  CODIFICAR(categoría)  
    añadir [web_id, categoría, categoría_id]  
  end for  
end function
```

---

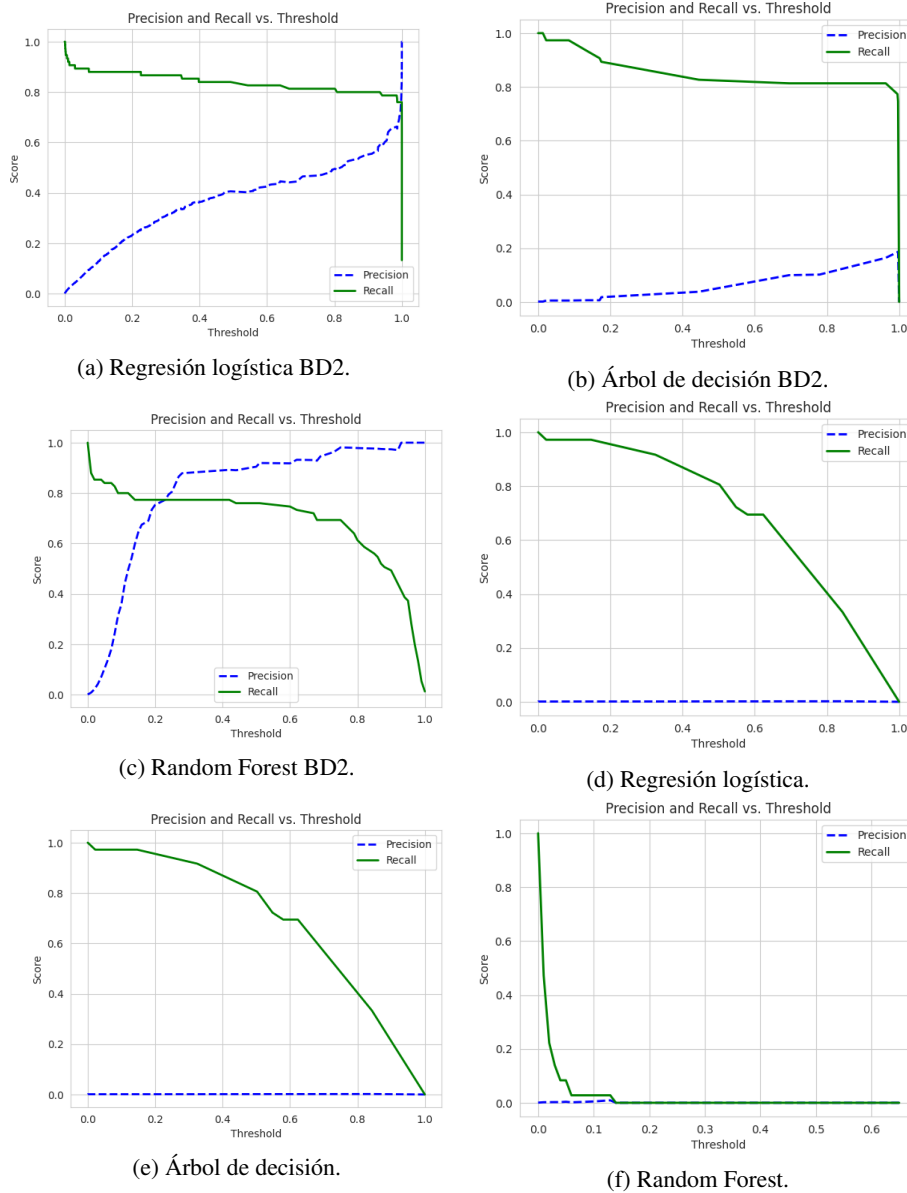
La Base de datos generada considera transacciones hechas en terminales físicas dentro de un radio establecido por usuario. Así mismo, genera tres tipos de fraudes, de los cuales en uno de ellos lo considera fraude siempre que sea mayor a 220. Esta es la principal literatura en este tipo de bases de datos sintéticas, y que además es coherente con las prácticas convencionales.

### 3. Propuesta

En este trabajo se presenta una herramienta de creación de datos sintéticos con transacciones bancarias, algunos de los cuales serán marcados como micro-fraude. El algoritmo propuesto considera tres premisas, que se describen a continuación:

- **Premisa 1:** Un consumidor escoge una distribución uniforme entre 1 y 5 el número de compras que usualmente se realizan en un día.
- **Premisa 2:** Para cada compra, el consumidor escoge una distribución uniforme entre 5 y 100 para decidir cuanto gasta en ese día.
- **Premisa 3:** el atacante escoge (estocásticamente) al 12 % de los consumidores, y los estafa, con montos dentro de una distribución normal con parámetros  $(\mu, \sigma)$ . Estos parámetros pueden ser calculados estadísticamente por una institución bancaria a partir de un fraude detectado.

En la siguiente sección se describe a detalle la creación de la base de datos.



**Fig. 1.** Curva precisión/recall en el conjunto de prueba para la base de datos de referencia (izquierda a,b,c) y en nuestra base de datos sintética (derecha d,e,f).

### 3.1. Generación de base de datos de micro fraudes

Para la creación de transacciones bancarias se consideró la creación de perfiles de clientes  $c$  y páginas web  $w$  de transacciones (el origen de la ésta). El primer paso fue crear una tabla de clientes que reflejara comportamientos de gastos únicos por

**Tabla 1.** Resultados de tres técnicas de aprendizaje de máquina en la base de datos en el conjunto de prueba.

Modelo	Precision	Recall	F1
LR	0.001	0.778	0.002
DT	0.001	0.806	0.002
RF	0	0	0

**Tabla 2.** Resultados en la base de datos de referencia sobre el conjunto de prueba.

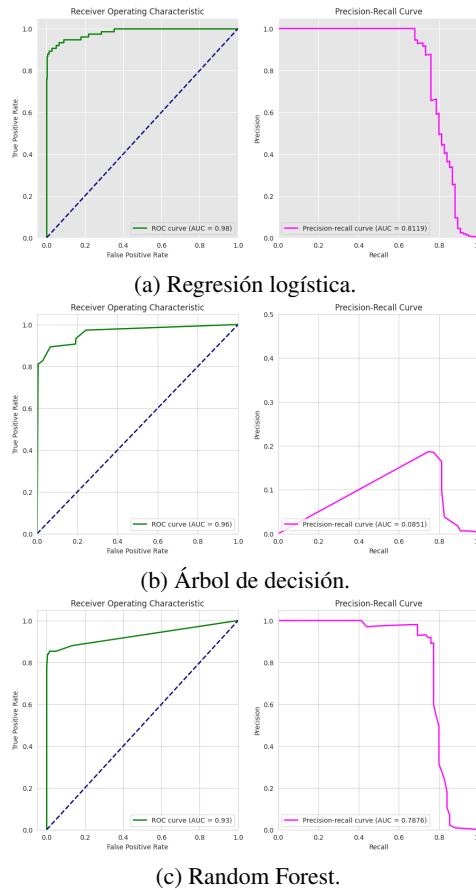
Modelo	Precision	Recall	F1
LR	0.403	0.827	0.542
DT	0.099	0.813	0.176
RF	0.891	0.76	0.82

cliente. Se generaron  $c=3,000$  clientes con características específicas, como un monto promedio de transacción distribuido uniformemente en el intervalo  $m=[5, 100)$  y un número promedio de transacciones diarias también distribuido uniformemente en el rango  $f=[1, 5)$  (*premisas 1 y 2*). Estos números, en la práctica, pueden ser adaptados a los datos propios del banco.

Posteriormente, se generó una segunda tabla que representaba  $w=100$  páginas web donde se realizaron las transacciones. La generación de fraudes se llevó a cabo tomando el  $c_a = 12\%$  de los clientes como usuarios afectados. En cada uno de estos casos, se seleccionaron estocásticamente  $f=4$  transacciones para ser reemplazadas por transacciones fraudulentas que siguieron una distribución normal (*premisa 3*). Nuevamente, el porcentaje y cantidad de transacciones es adaptable a las necesidades.

Como resultado y de acuerdo a lo esperado, se obtuvo un conjunto de datos altamente des-balanceado, donde las transacciones etiquetadas como fraudes representaban solo el  $0.06\%$  del total. Este conjunto de datos final fue de 2,605,617 registros y 9 características. Estas son: *etiqueta*, *transacción\_id*, *tiempo*, *cliente\_id*, *web\_id*, *categoría\_web*, *cantidad*, *tiempo\_segundos*, *tiempo\_días* (estos últimos dos entre transacciones). Todos los datos fueron procesados para convertir las características en números enteros o reales, lo que resultó en un conjunto de datos final listo para su uso en modelos de detección de fraude de la siguiente forma:

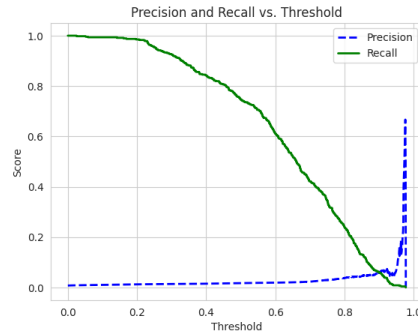
- **etiqueta**, un indicador consecutivo
- **web\_id**, un id de un sitio web donde se efectuó la operación
- **cantidad**, monto de la transacción
- **durante\_entresemana**, Tiene un 1 si la transacción fue realizada entre semana
- **durante\_noche**, tiene un 1 si la transacción fue realizada de noche
- **número\_transacciones\_por\_cliente\_1día\_ventana**, cuenta el número de transacciones realizadas anteriormente por cliente en 1 día



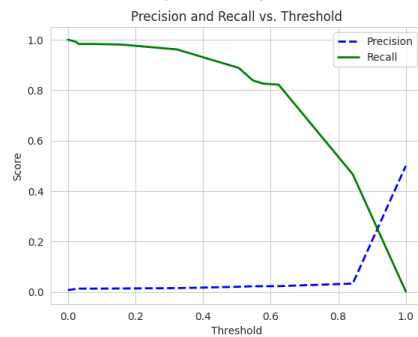
**Fig.2.** Curvas ROC y precisión-Recall en el conjunto de prueba para la base de datos de referencia.

- **promedio\_cantidad\_por\_cliente\_1día\_ventana**, calcula el promedio de las transacciones realizadas anteriormente por cliente en 1 día
- **número\_transacciones\_por\_cliente\_7días\_ventana**, cuenta el número de transacciones realizadas anteriormente por cliente en 7 días
- **promedio\_cantidad\_por\_cliente\_7días\_ventana**, calcula el promedio de las transacciones realizadas anteriormente por cliente en 7 días
- **número\_transacciones\_por\_cliente\_30días\_ventana**, cuenta el número de transacciones realizadas anteriormente por cliente en 30 días
- **promedio\_cantidad\_por\_cliente\_30días\_ventana**, promedio de las transacciones realizadas anteriormente por cliente en 30 días

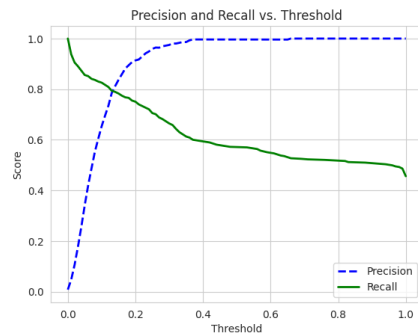
El pseudocódigo se presenta en los algoritmos 1, 2, 3, 4 mostrados a continuación.



(a) Regresión logística.



(b) Árbol de decisión.



(c) Random Forest.

**Fig. 3.** Curva precisión/recall en el conjunto de prueba de la base de datos sintética con aumento de fraudes al 0.2 %.

### 3.2. Modelos estudiados

Una vez que es posible tener los datos sintéticos se analizaron con las siguientes propuestas: regresión logística (LR) [7] , Árboles de Decisión (DT) [8] con una profundidad de 4 utilizando la impureza Gini; y por último Random Forest (RF) [7, 9] usando 100 estimadores y la impureza de Gini sin profundidad máxima.



**Métricas.** Las métricas utilizadas para medir el desempeño de los modelos fueron precision, recall y F1 [7]. La matriz de confusión es una tabla que describe el rendimiento de un modelo de clasificación en un conjunto de datos, mostrando la cantidad de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN). La precisión (precision) mide la proporción de predicciones positivas que fueron correctas, y se calcula como:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (1)$$

La sensibilidad (recall) mide la proporción de instancias positivas que fueron correctamente identificadas por el modelo, y se calcula como:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (2)$$

El F1-score es la media armónica de la precisión y el recall, lo que proporciona un equilibrio entre ambas métricas. Se calcula como:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (3)$$

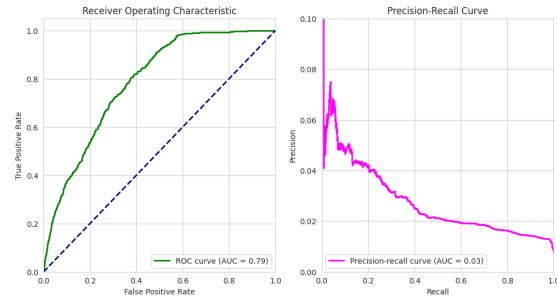
## 4. Experimentación

Para trabajar con esta base de datos evidentemente desbalanceada en las clases se hizo una validación presecuencial con folds=4 aplicados a tres modelos de Aprendizaje de Máquina: Regresión Logística, Árbol de Decisión y Bosque Aleatorio. Además de ello, se utiliza la técnica SMOTE (Synthetic Minority Oversampling Technique) para lograr el balance de las clases. El conjunto de entrenamiento tuvo un 70 % y el de prueba 30 %. Los resultados se muestran en la tabla 1. Note que se priorizó un recall alto (threshold=0.5), con excepción del bosque aleatorio, el cual no logra destacar con sus predicciones.

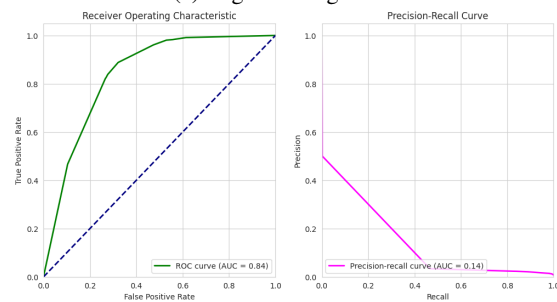
Considerando que solo existe una base de datos [3], la comparación se llevó a cabo uniformizando los criterios a ella, es decir, se usaron casi la misma cantidad de datos, es decir, 303,868. También se incrementó el número de fraudes en esa cantidad, esto es 178 fraudes (0.058 % del total). Los resultados de esta segunda base de datos (a la que llamaremos BD2) se muestran en la tabla 2. En ellos se destaca el desempeño de RF.

### 4.1. Variando el umbral

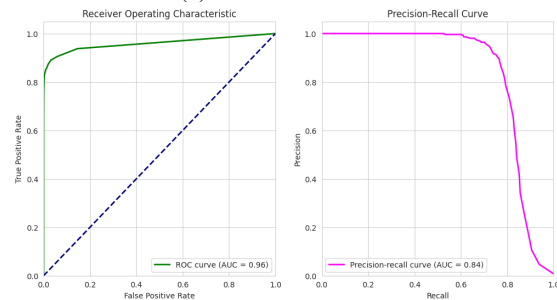
[t!] En esta sección se presenta el resultado de esta variar el umbral o threshold, es decir, cambiar la cantidad de fraudes detectados sobre la BD2. Evidentemente ampliar el umbral permite catalogar como fraudulentas las transacciones que tienen probabilidades más bajas como puede verse en la figura 1. En estas gráficas es claro observar el valor del umbral para cada algoritmo y mantener un equilibrio entre precisión y sensibilidad. Esto se puede ver reflejado también en la figura 2, donde los valores AUC resultan ser aceptables en general.



(a) Regresión logística.



(b) Árbol de decisión.



(c) Random Forest.

**Fig. 4.** Curvas ROC y precisión-recall en el conjunto de prueba de la base de datos sintética con aumento de fraudes al 0.2 %.

Si hacemos lo mismo con nuestra base de datos sintética, es posible conseguir valores más altos de recall al disminuir el umbral. Sin embargo, la precisión no logra despuntar debido a la muy poca cantidad de fraudes lo cual compromete considerablemente esta métrica. En la columna derecha de la figura 1 es posible el desempeño de los modelos sobre nuestra base de datos.

#### 4.2. Variando el porcentaje de fraudes

La base de datos sintética tiene un mayor nivel de des-balance entre clases, lo que compromete su desempeño. Al aumentar prematuramente la cantidad de fraudes (clase 1) de tal forma que llegue del 0.058 % al 0.2 % de transacciones marcadas como fraude,

se consigue tener el mismo porcentaje de la base de datos de referencia. Los resultados obtenidos se muestran en la figura 3. Note que tanto la regresión logística como el árbol de decisión se vuelven ligeramente más precisos conservando un recall alto. Por otro lado, el Random Forest supera con creces a los dos modelos anteriores, a la par que con un umbral de 0,5 se obtienen un F1 del 0,769 que supera al resultado del conjunto de datos por referencia con este mismo umbral. De la misma manera, las curvas de la figura 4 presentan valores más altos comparados con el primer experimento.

## **5. Conclusiones y trabajo a futuro**

Las transacciones a tarjetas de crédito han crecido enormemente en los últimos años, es de destacarse que el comportamiento de los clientes recientemente es muy distinto al de años previos a la pandemia. Además, es remarcable que también los fraudes han crecido considerablemente, en especial el conocido como micro fraude, el cual es de gran complejidad debido a lo desapercibidos que pueden ser incluso para el mismo cliente, básicamente se resume a transacciones con montos pequeños con nombres de establecimientos comunes (por ejemplo, un cargo en un centro de entretenimiento). Aunado a lo anterior, al no haber bases de datos públicas se vuelve aun más difícil encontrar un comportamiento de estos fraudes.

La propuesta presentada en este artículo consiste tanto en la generación de datos sintéticos de transacciones bancarias con micro fraudes controlada por diversos parámetros como el número de clientes afectados, en este caso al 12%, y sustituyendo solamente 4 de todas sus transacciones como fraude, siguiendo una distribución normal. El código para la generación de las bases de datos sintéticas se encuentra disponible en<sup>3</sup>. En el presente trabajo se pudo observar que al incrementar la cantidad los fraudes se logró conseguir mejores resultados con las técnicas de aprendizaje de máquina empleadas.

En particular se resalta la técnica de Random Forest o bosques aleatorios, que superaron con su desempeño a la regresión logística y a los árboles de decisión. Es claro que al tener números tan bajos de transacciones fraudulentas, se vuelve una tarea compleja predecir tales movimientos con una precisión perfecta. Sin embargo, se insta a conseguir niveles de sensibilidad o recall más altos pues esto significa conseguir todos los fraudes, lo que a la vez se traduce a menor número de pérdidas monetarias para la ambas partes.

En conclusión, con esta herramienta mostramos que es viable para predecir micro fraudes en tarjetas de crédito de los que se tenga información sobre su distribución. Es decir, los modelos de aprendizaje automático si podrían distinguir entre datos generados de dos distribuciones, uniforme y normal. En la vida real, un banco podría hacer un análisis de sus propios datos y reemplazar los valores mencionados (1 y 5,5 y 100, 12%) por los propios, a los que tiene acceso en su data warehouse, así como estimar ( $\mu$ ,  $\sigma$ ), que también tiene manera de estimar. Como trabajo a futuro se propone probar otros modelos que puedan ser más aptos para este problema, tal como PBC4cip para el balanceo de clases, entre otros.

<sup>3</sup>[github.com/jqbeltran/Carding](https://github.com/jqbeltran/Carding)

## **Referencias**

1. Radage, K.: Credit card fraud in 2023 (2023) <https://www.clearlypayments.com/blog/credit-card-fraud-in-2023/>
2. Rodríguez, D.: “Me hicieron un cargo al mes sin darme cuenta”: Así son las estafas virtuales a las tarjetas de los mexicanos. EL PAÍS México (2022) <https://elpais.com/mexico/2022-10-21/me-hicieron-un-cargo-al-mes-sin-darme-cuenta-asi-son-las-estafas-virtuales-a-las-tarjetas-de-los-mexicanos.html>
3. Kaggle.: Worldline and the machine learning group of université libre de bruxelles: Credit card fraud detection (2023) <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?select=creditcard.csv>
4. Faraji, Z.: A review of machine learning applications for credit card fraud detection with a case study. SEISENSE Journal of Management (2022) <https://journal.seisense.com/jom/article/view/770>.
5. Marazqah-Btoush, E. A. L., Zhou, X., Gururajan, R., Chan, K. C., Genrich, R., Sankaran, P.: A systematic review of literature on credit card cyber fraud detection using machine and deep learning. PeerJ Computer Science, vol. 9, pp. e1278 (2023) doi: 10.7717/peerj-cs.1278
6. Le Borgne, Y., Siblini, W., Lebichot, B., Bontempi, G.: Reproducible machine learning for credit card fraud detection - practical handbook. Université Libre de Bruxelles (2022) <https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook>.
7. Géron, A.: Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. 2nd edition, O'Reilly Media, Inc., Sebastopol, CA (2019)
8. Shalev-Shwartz, S. Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge University Press (2014)
9. Kelleher, J., Namee, B., D'Arcy, A.: Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies. 2nd edition MIT Press (2015)

## Técnicas de aprendizaje automático para agrupar planetas similares a la Tierra

Marcos Macías-Juárez, Edgar Moyotl-Hernández

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias Físico Matemáticas,  
México

marcos.macias@alumno.buap.mx, emoyotl@fcfm.buap.mx

**Resumen.** Este trabajo presenta un modelo de aprendizaje automático, con técnicas de agrupamiento y clasificación, que puede utilizarse para formar un grupo de exoplanetas con características similares a las de la Tierra. Para la parte de clasificación, se utilizó k-nn y para la parte del agrupamiento, se usó k-means++, DBSCAN, agglomerative clustering y divisive clustering. Aplicando el modelo a un conjunto de datos con información de más de 5000 exoplanetas, descubiertos por la NASA, se identificó que el mejor modelo de aprendizaje se consigue con agglomerative clustering o divisive clustering y k-nn, juntos logran encontrar un grupo de 66 exoplanetas con características comparables a las del planeta Tierra. Los resultados obtenidos muestran que las técnicas de aprendizaje automático pueden ser aplicadas en diversas áreas.

**Palabras clave:** Aprendizaje automático, agrupamiento, clasificación, astronomía, exoplanetas.

## Machine Learning Techniques to Group Earth-like Planets

**Abstract.** This work presents a machine learning model that uses clustering and classification techniques which can be used to form a group of exoplanets with characteristics similar to those of the Earth. For the classification part, we used k-nn and for the clustering part, we used k-means++, DBSCAN, agglomerative clustering and divisive clustering. Applying the model on a dataset with information on more than 5000 exoplanets discovered by NASA, it was identified that the best learning model is achieved with agglomerative clustering or divisive clustering and k-nn, together they achieve to find a cluster of 66 exoplanets with characteristics comparable to those of the planet Earth. The results obtained show that machine learning methods can be applied in various areas.

**Keywords:** Machine learning, clustering, classification, astronomy, exoplanets.

## 1. Introducción

Una de las líneas de estudio en astrofísica se encarga de la búsqueda de exoplanetas y del estudio de los mismos. Un exoplaneta es un planeta que se encuentra fuera del sistema solar, esto significa que orbita alrededor de una estrella central distinta al Sol. Los exoplanetas se estudian con el propósito de detectar planetas con características similares a las de la Tierra, esto es de suma importancia dado que, analizando la composición de estos planetas y teniendo en cuenta factores como la distancia entre este y su estrella anfitriona, es posible deducir teóricamente si podrían albergar vida.

La misión Kepler de la NASA revolucionó el campo de la exploración de exoplanetas y ha permitido a los investigadores recopilar datos sobre objetos extrasolares de interés. Para manejar la enorme cantidad de datos recolectados, es importante desarrollar métodos de análisis de datos que sean capaces de identificar y confirmar que un objeto dado sea un exoplaneta y más aún, métodos para clasificar exoplanetas (según sus propiedades físicas) en categorías apropiadas de habitabilidad para dar una idea de qué tan similar o diferente es un exoplaneta a la Tierra.

El aprendizaje automático constituye esa herramienta capaz de tratar con enormes cantidades de información, debido a esto las técnicas y algoritmos de aprendizaje automático se han vuelto muy importantes en la astronomía y en muchos otros campos, siendo empleados en múltiples tareas. Este trabajo se centra en la aplicación de técnicas de aprendizaje automático supervisado y no supervisado, en primer lugar, para entrenar un modelo capaz de agrupar exoplanetas con características similares entre ellos; y en segundo lugar, para clasificar a los planetas del sistema solar, en particular el planeta Tierra, en alguno de estos grupos.

Con este modelo, se busca un grupo de planetas con características parecidas a las de la Tierra y así, posiblemente, encontrar algún planeta en el cuál sea posible albergar la vida que alberga la Tierra. En este estudio, se utilizó un conjunto de datos de la NASA que contiene características de exoplanetas confirmados. Además, para la parte de aprendizaje supervisado, se utilizó el algoritmo k-nn y para la parte de aprendizaje no supervisado, se usaron los métodos k-means++, DBSCAN, agglomerative clustering y divisive clustering.

El artículo está organizado de la siguiente manera. En la sección 2 se exponen los trabajos relacionados. La sección 3 presenta la propuesta de solución donde se explica la metodología que se siguió. En la sección 4 se describen las pruebas con el modelo propuesto. En la sección 5 se analizan y discuten los resultados obtenidos de los experimentos. Finalmente, la sección 6 muestra las conclusiones y el trabajo futuro.

## 2. Trabajo relacionado

Existen diversos estudios que exploran la eficacia del aprendizaje automático para la detección de exoplanetas y para clasificar exoplanetas en clases de habitabilidad, basándose en las características físicas de los propios exoplanetas y sus estrellas madre o anfitrión. En [3] el objetivo del estudio fue comparar la eficiencia de los métodos k-nn, logistic regression y decision trees con respecto a la detección de exoplanetas.

**Tabla 1.** Características de cada exoplaneta.

Característica	Descripción
name	Nombre del planeta dado por la NASA.
distance	Distancia del planeta a la tierra en años luz.
stellar_magnitude	Brillo del planeta.
planet_type	Tipo de planeta derivado de los del sistema solar.
discovery_year	Año en que se descubrió el planeta.
mass_multiplier	Multiplicidad de masa del planeta con el planeta de 'mass_wrt'.
mass_wrt	Masa comparada a la de algún planeta del sistema solar.
radius_multiplier	Multiplicidad de radio del planeta con el planeta de 'radio_wrt'.
radius_wrt	Radio comparado al de algún planeta del sistema solar.
orbital_radius	Radio orbital de los planetas alrededor de su Sol (en AU).
orbital_period	Años que tardan en completar 1 órbita de su estrella anfitrión.
eccentricity	Indica qué tan circular es la trayectoria orbital del planeta.
detection_method	Método utilizado por la NASA para encontrar ese planeta.

Utilizaron estos algoritmos de aprendizaje supervisado para analizar un conjunto de datos con muestras tanto reales como sintéticas y entrenar un modelo capaz de predecir con precisión si un objeto astrofísico es un exoplaneta o no. Los resultados mostraron que k-nn, al ser un clasificador no lineal, fue el más eficiente en clasificar correctamente los exoplanetas, logrando una precisión del 98.22 %.

Por su parte, en [4] llevaron a cabo aprendizaje supervisado y no supervisado en dos conjuntos de datos recopilados por la NASA, el conjunto de datos Kepler y un conjunto de datos de exoplanetas confirmados, el NASA Exoplanet Archive. Por un lado, al conjunto de datos Kepler lo usaron para predecir la existencia de candidatos a exoplanetas como tarea de clasificación, utilizando los algoritmos: decision trees, random forest, naïve Bayes y redes neuronales.

Como resultado, los métodos obtuvieron precisiones del 99.06 %, 92.11 %, 88.50 % y 99.79 %, respectivamente. Por otro lado, el conjunto de datos de exoplanetas confirmados lo usaron para encontrar exoplanetas habitables, como tarea de agrupamiento. Para esto, dividieron los exoplanetas confirmados en diferentes grupos utilizando el algoritmo k-means. Antes de agrupar los exoplanetas agregaron las características de la Tierra al conjunto de datos. Luego, dividieron todos los planetas en 100 clusters y consideraron que los exoplanetas que tienen más probabilidades de ser habitables se encontraban en el grupo que contenía a la Tierra.

Esto significa que encontraron un grupo de 100 exoplanetas con características parecidas a las de la Tierra. En contraste con los trabajos citados, este trabajo, se centra en la aplicación de la clasificación y el agrupamiento para la detección de exoplanetas con características similares a las de la Tierra. Primero, agrupa exoplanetas con características similares entre ellos; y segundo, clasifica a los planetas del sistema solar, en alguno de estos grupos.

**Tabla 2.** Características de los planetas del sistema solar.

Planeta	mass_multiplier	radius_multiplier	orbital_radius	eccentricity
Mercurio	0.055	0.38	0.38	0.2000
Venus	1.000	1.00	0.99	0.0170
Tierra	0.820	0.95	0.72	0.0068
Marte	0.500	0.53	1.50	0.0930
Júpiter	318.000	11.00	5.20	0.0480
Saturno	95.180	9.40	9.50	0.0566
Urano	0.140	4.00	19.10	0.0460
Neptuno	17.000	3.80	30.06	0.0097

Los ocho planetas del sistema solar, en orden de cercanía al Sol, son: Mercurio, Venus, Tierra, Marte, Júpiter, Saturno, Urano y Neptuno. La clasificación y el agrupamiento son técnicas para encontrar patrones usadas en el aprendizaje automático (machine learning, en Inglés). La tarea de clasificación se enmarca en el aprendizaje supervisado, se enfoca en la creación de modelos a partir de un conjunto de datos etiquetados (es decir, datos para los que ya se conoce la respuesta correcta) y permite predecir a que clase pertenece un objeto nuevo.

En cambio, la técnica de agrupamiento o clustering pertenece al aprendizaje no supervisado y su objetivo es crear modelos, a partir de un conjunto de datos no etiquetados, que encuentren grupos de objetos similares de forma que los elementos del mismo grupo estén más relacionados entre ellos que aquellos elementos de diferentes grupos [7], estos grupos se conocen como clústeres o clusters.

### 3. Metodología

El modelo de aprendizaje que se propone consta de dos partes, algoritmos de aprendizaje supervisado para clasificación y algoritmos de aprendizaje no supervisado para clustering. Para la parte de clasificación, se utilizó k-nn y para la parte del agrupamiento, se uso k-means++, DBSCAN, agglomerative clustering y divisive clustering. Así mismo, las medidas de calidad para determinar el rendimiento del modelo propuesto fueron el coeficiente de silueta y el índice de Davies Bouldin.

#### 3.1. Método K-NN

El algoritmo de k vecinos más cercanos, también conocido como k-nn (del inglés k-nearest neighbours), es un clasificador de aprendizaje supervisado no paramétrico y no lineal, que utiliza la proximidad entre puntos de datos para hacer predicciones sobre la clasificación de un punto individual. Al final, el nuevo objeto pertenece a la categoría con mayoría de votos entre su k vecinos más cercanos. La “k” en k-nn representa el número de vecinos más cercanos considerados [7].



**Algorithm 1:** Modelo de aprendizaje propuesto

---

**Input:** Conjunto de exoplanetas  $X$ , conjunto de planetas del sistema solar  $P$ , número máximo de iteraciones  $MAXITE$ .

**Output:** Un grupo de planetas similares  $C_t \subset X$ .

```

1 for  $i \leftarrow 1$  to  $MAXITE$  do
2    $k \leftarrow \text{card}(P)$ ; // Cardinalidad del conjunto  $P$ 
3    $C \leftarrow \text{agrupar}(X, k)$ ;
4   for  $j \leftarrow 1$  to  $k$  do
5      $y_j \leftarrow \text{clasificar}(p_j, C)$ ;
6     if  $p_j = \text{Tierra}$  then
7        $t \leftarrow y_j$ ; // Etiqueta del grupo de la Tierra
8     end
9   end
10   $S \leftarrow P \cap C_t$ ; // Planetas en el grupo de la Tierra
11  if  $\text{card}(S) = 1$  then
12    return  $C_t$ ;
13  end
14   $X \leftarrow C_t$ ;
15   $P \leftarrow S$ ;
16 end

```

---

**3.2. Métodos k-means y k-means++**

k-means es un algoritmo de aprendizaje no supervisado que agrupa objetos en k grupos basándose en prototipos.

El agrupamiento basado en prototipos significa que cada grupo se representa por un centroide o por un medoide [7]. De acuerdo con la información presentada en [2], k-means es el algoritmo de agrupación particional más utilizado. El método k-means es muy bueno para identificar grupos con una forma esférica pero uno de sus inconvenientes es que se tiene que especificar, con antelación, la cantidad k de grupos a generar [7]. Por su parte, el algoritmo k-means++ selecciona cuidadosamente los centroides iniciales para la agrupación de k-means y luego se realiza la agrupación con k-means clásico utilizando estos centroides [1].

**3.3. Método DBSCAN**

Entre los métodos de agrupamiento, los algoritmos basados en densidad enfocan el problema de dividir el conjunto de datos en grupos teniendo en cuenta la distribución de densidad de los puntos [6]. DBSCAN (Density Based Spatial Clustering of Applications with Noise) es el primer algoritmo de agrupamiento basado en densidad [5]. Tiene dos parámetros principales que son el radio de la vecindad (eps) que es la distancia máxima entre dos puntos para poder ser considerados pertenecientes al mismo vecindario, y el número mínimo de puntos (minpts) en un vecindario para que un punto pueda ser considerado de alta densidad. DBSCAN puede resolver problemas en los cuales k-means puede fallar puesto que funciona muy bien con formas complejas que no tienen que ser esféricas, además, identifica los valores atípicos o ruido en los datos y, no necesita que se defina de antemano el número de clústeres [6].

**Tabla 3.** Cantidad de planetas por cluster (primera iteración).

Cluster	Algoritmos de agrupamiento			
	k-means++	DBSCAN	Agglomerative	Divisive
0	<b>4526</b>	1717	245	2
1	1	<b>3002</b>	44	7
2	2	7	7	<b>3909</b>
3	1	12	<b>3909</b>	498
4	5	6	498	245
5	224	9	2	59
6	5	6	59	44
7	1	6	1	1

### 3.4. Método Agglomerative clustering

Los métodos jerárquicos construyen una jerarquía de clústeres en el conjunto de datos. Esta jerarquía se representa como un árbol (o dendrograma). La raíz del árbol es el único grupo que contiene a todos los puntos de datos, siendo las hojas los grupos con un sólo dato. Hay dos tipos de algoritmos de agrupamiento jerárquico con enfoques inversos: el algoritmo aglomerativo (agglomerative clustering) y el algoritmo divisivo (divisive clustering).

El algoritmo jerárquico aglomerativo, es una estrategia ascendente que inicia considerando cada punto de datos como un grupo individual y luego va fusionando pares de grupos similares progresivamente hasta que todos los datos pertenezcan a un único grupo [7].

### 3.5. Método divisive clustering

El algoritmo jerárquico divisivo es inverso al aglomerativo, sigue un enfoque de arriba hacia abajo, comienza con un sólo grupo que contiene todos los puntos de datos y lo va dividiendo iterativamente en grupos más pequeños hasta obtener un grupo para cada dato [7]. El método aglomerativo, generalmente, resulta ser más sencillo de implementar que el método divisivo porque existe un único modo de unir dos grupos, mientras que existen muchas maneras de separar un conjunto de puntos en dos grupos, lo cual es muy costoso en términos computacionales. Por otro lado, una de las ventajas de este tipo de algoritmos sobre k-means es que no requiere que los datos se representen en espacios vectoriales, ya que solo requiere de una medida de distancia entre puntos. Tampoco requiere especificar por anticipado el número de grupos. Sin embargo, pueden producir grupos de tamaños desiguales.

### 3.6. Métricas de calidad

Las técnicas de aprendizaje no supervisado, a diferencia de las supervisadas, se entrenan con conjuntos de datos sin etiquetas, por lo que, cuantificar la calidad de los resultados proporcionados por un algoritmo de agrupamiento no es un problema fácil.

**Tabla 4.** Resultados de agglomerative clustering por iteración.

Agglomerative clustering							
Cluster	Iteración 1		Iteración 2	Iteración 3		Iteración 4	
	EA	PC	PC	EA	PC	EA	PC
0	245	0	1	763	0	85	2
1	44	0	0	587	0	86	1
2	7	0	0	879	1	79	0
3	<b>3909</b>	<b>8</b>	<b>5</b>	269	0	<b>66</b>	<b>1</b>
4	498	0	0	<b>316</b>	<b>4</b>	-	-
5	2	0	1	-	-	-	-
6	59	0	0	-	-	-	-
7	1	0	1	-	-	-	-
<i>S</i>	0.8286		0.6304	0.4967		0.5012	
<i>I<sub>DB</sub></i>	0.4257		0.5409	0.5519		0.6080	

En este sentido, el coeficiente de silueta y el índice de Davies Bouldin son medidas de calidad de los grupos resultantes de un agrupamiento y pueden aplicarse a distintos algoritmos. Las dos métricas calculan la cohesión (distancia promedio de un punto al resto de puntos dentro del mismo grupo) y la separación (distancia promedio de un punto a todos los puntos en el grupo más cercano) [7].

El coeficiente de silueta  $S$  se calcula como la diferencia entre la cohesión y la separación del grupo dividida por el valor más grande de los dos;  $S$  puede tomar valores entre -1 y 1, siendo 1 el valor ideal. En cambio, el índice de Davies Bouldin  $I_{DB}$  se calcula como la relación promedio entre la cohesión y la separación; los valores de  $I_{DB}$  más próximos a 0 son mejores.

## 4. Experimentación

El objetivo del presente estudio es comparar la eficiencia de varios algoritmos de aprendizaje automático respecto a la agrupación de exoplanetas.

### 4.1. Conjunto de datos

Para los experimentos de aprendizaje no supervisado se utilizó el NASA Exoplanet Archive, este conjunto de datos contiene información sobre los exoplanetas descubiertos por la NASA<sup>1</sup> en diversas misiones espaciales, observatorios terrestres y otras fuentes<sup>2</sup>. Este conjunto de datos de planetas (ya confirmados como exoplanetas) incluye información como el nombre del planeta, masa, radio, distancia desde su estrella

<sup>1</sup><https://www.nasa.gov>

<sup>2</sup>Los datos están accesibles en la plataforma Kaggle en el siguiente enlace: <https://www.kaggle.com/datasets/adityamishram/nasaexoplanets>

**Tabla 5.** Resultados de k-means por iteración.

k-means++											
Cluster	Iteración 1		Iteración 2		Iteración 3		Iteración 4		Iteración 5		Iteración 6
	EA	PC	EA	PC	EA	PC	EA	PC	EA	PC	PC
0	<b>4526</b>	<b>8</b>	<b>4185</b>	<b>8</b>	137	0	1000	1	<b>1447</b>	<b>4</b>	<b>4</b>
1	1	0	13	0	41	0	21	1	1	0	0
2	2	0	1	0	130	0	1	0	5	0	0
3	1	0	122	0	<b>3272</b>	<b>6</b>	144	1	621	1	0
4	5	0	2	0	154	1	<b>2079</b>	<b>5</b>	5	0	-
5	224	0	1	0	96	0	27	0	-	-	-
6	5	0	322	0	227	1	-	-	-	-	-
7	1	0	59	0	128	0	-	-	-	-	-
<i>S</i>	0.9052		0.8394		0.5848		0.5341		0.4937		0.4884
<i>I<sub>D</sub>B</i>	0.4542		0.4166		0.5698		0.5593		0.6607		0.7538

anfitriona, período orbital y otras características físicas. También incluye información sobre la estrella anfitriona, como su nombre, masa y radio. Cabe mencionar que el archivo se actualiza periódicamente a medida que se descubren nuevos exoplanetas. La última actualización del conjunto de datos se hizo en febrero del 2023, el archivo contiene en total 5250 muestras de planetas confirmados pero no etiquetados como similares o no a la Tierra. En la Tabla 1 se describen las 13 características almacenadas de cada exoplaneta conocido hasta el momento.

#### 4.2. Preprocesamiento y limpieza de datos

Este paso es fundamental en el análisis de datos puesto que implica identificar y corregir errores, inconsistencias y datos incompletos en los conjuntos de datos. Por lo tanto, es crucial para garantizar la precisión de los resultados del análisis realizado. A continuación, se explican los pasos involucrados en este proceso sobre los datos del NASA Exoplanet Archive.

**Datos nulos.** Dentro del conjunto de datos se logró identificar la existencia de datos nulos en ciertas características, la cantidad de datos nulos no era significativa por lo que no se eliminó por completo ninguna característica, en cambio se eliminaron muestras que contenían un dato nulo en alguna de sus características. Una vez realizado este proceso el conjunto de datos pasó de tener 5250 muestras a tener 4765 (el 90.76 % del conjunto original).

**Transformación de datos.** En el conjunto de datos se identificó que los valores de las características `mass_multiplier` y `radius_multiplier` poseen una escala de medición distinta, específicamente la medición de la multiplicidad de la masa y el radio se hizo con respecto a los planetas Tierra y Júpiter. Por esta razón, todas las muestras cuya multiplicidad de masa y radio fueron medidas con respecto a Júpiter se transformaron para que la escala sea con respecto a la Tierra. Mediante una investigación en [8] y en [10], la masa de Júpiter es 318 veces mayor que la masa de la Tierra, mientras que el radio de Júpiter es 11 veces mayor que el de la Tierra.

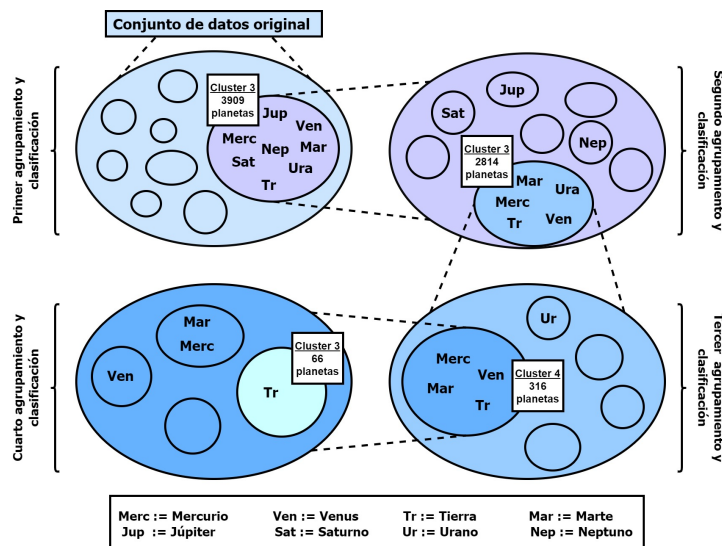


Fig. 1. Resultados obtenidos con agglomerative clustering.

Considerando esta información, todos los valores de `mass_multiplier` y `radius_multiplier` que se midieron con respecto a Júpiter, se multiplicaron por 318 y 11, respectivamente.

**Eliminación de variables.** En el conjunto de datos existen tanto datos numéricos como categóricos, pero, las técnicas de aprendizaje automático aquí utilizadas no permiten trabajar con variables alfanuméricas, por lo que, se descartan las variables `mass_wrt`, `radius_wrt` y `detection_method`; es importante señalar que dichas variables no representan una característica física de los exoplanetas, lo cual es importante para el propósito de este trabajo. Por otro lado, la característica `discovery_year` pese a ser una variable numérica, tampoco representa una característica física de los exoplanetas por lo que, también se descartó.

Además se eliminó la variable `planet_type`, ya que esta etiqueta no corresponde a la que se desea predecir. Más bien, es el tipo de exoplaneta derivado de los planetas del sistema solar y categoriza los exoplanetas en los siguientes tipos: gigante gaseoso, neptuniano, supertierra y terrestre. Otra de las tareas comunes en la limpieza de datos es la eliminación de valores atípicos (outliers), sin embargo, se ha optado por omitir esta tarea para tomar en cuenta todos los posibles valores encontrados y analizar e interpretar los resultados que se obtengan en el entrenamiento del modelo.

**Selección de variables.** La información en [9] influye al momento de seleccionar las variables más apropiadas para la experimentación. Por lo que se considero lo siguiente. La distancia de un planeta a su Sol o estrella anfitriona es un factor fundamental que afecta una amplia gama de características y procesos planetarios, desde el clima y la habitabilidad hasta la dinámica orbital y la exploración espacial; es un aspecto central en la comprensión del cosmos y en la búsqueda de vida en otros mundos. En este sentido, la variable `orbital_radius` resulta ser una característica importante a considerar para el entrenamiento del modelo.

Caso contrario es el de la variable `distance` la cual no proporciona información comparable con respecto a la Tierra ya que esta variable en si misma ya está comparada con respecto a la Tierra. Lo mismo sucede con la variable `stellar_magnitude`, la cual cuantifica el brillo de una estrella o cuerpo celeste observado desde la Tierra, por lo que, convendría no considerar esta variable para el entrenamiento.

Por todo lo anterior, las variables más relevantes para realizar la experimentación son: `mass_multiplier`, `radius_multiplier`, `orbital_radius` y `eccentricity`. La masa y el radio de los exoplanetas se miden en kg. y km., respectivamente, pero los dos primeros datos representan unidades de multiplicidad (números reales no negativos), el radio orbital es medido en AU (unidad astronómica, 1 AU = 149 597 870 700 metros) y la excentricidad es un valor entre 0 y 1.

**Conjunto de planetas.** Para la parte de aprendizaje supervisado se creó un segundo conjunto de datos recopilando información sobre los planetas del sistema solar, información correspondiente a las cuatro características seleccionadas para el agrupamiento. Los datos de estos planetas muestran la increíble diversidad del sistema solar, adicionalmente, proporcionan una variedad de tipos de planetas que posiblemente se pueda encontrar en otras partes del universo observable, por eso al realizar las tareas de aprendizaje no supervisado se pretende que la agrupación de los datos se haga de tal manera que exista un grupo por cada uno de los planetas del sistema solar. En la Tabla 2 se encuentran las características de los ocho planetas del sistema solar.

### 4.3. Pruebas

Una vez realizada la limpieza de datos y la selección de características que se utilizarán en los experimentos, se probará la metodología propuesta. El entrenamiento del modelo consta de dos partes, un algoritmo para agrupamiento y otro para clasificación. Para estas dos partes, se utilizó el conjunto de datos de exoplanetas confirmados por la NASA y el conjunto de datos de planetas del sistema solar, respectivamente. Para la parte de aprendizaje no supervisado, se utilizaron los métodos de `k-means++`, `DBSCAN`, `agglomerative clustering` y `divisive clustering`. Mientras que, para la parte del aprendizaje supervisado, se utilizó `k-nn`.

**Modelo de aprendizaje propuesto.** En el algoritmo 1 se muestra el proceso del modelo empleado para encontrar el grupo de planetas más parecidos a la Tierra, siendo  $X$  el conjunto de exoplanetas,  $C_1 \cup \dots \cup C_M = \cup C \subseteq X$  donde  $C = \{C_1, \dots, C_M\}$  son los grupos de exoplanetas y  $P = \{p_1, \dots, p_M\}$  el grupo de planetas del sistema solar. Inicialmente la cardinalidad  $M$  de los conjuntos,  $C$  y  $P$ , es ocho porque son ocho los planetas del sistema solar.

Durante las pruebas, en el paso 3, se usaron cada uno de los algoritmos de agrupamiento presentados anteriormente. Este paso devuelve los grupos que se forman con los datos de exoplanetas. Una vez creados los clústeres se usó el algoritmo `k-nn`, paso 5, para determinar a qué grupo pertenece cada uno de los planetas del sistema solar. Este mismo algoritmo de clasificación se utilizó con todas las técnicas de agrupamiento utilizadas. El valor de salida al clasificar los planetas corresponde a la etiqueta del cluster al que pertenecen los planetas del sistema solar de acuerdo con el resultado del agrupamiento.

En el paso 6 se identifica el cluster en el cual se clasificó la Tierra. Dentro del sistema solar hay planetas que no son muy diferentes a la Tierra (en cuanto a las características que se están considerando) por lo que, es de esperar que en la etapa de clasificación más de un planeta quede dentro del mismo grupo que la Tierra, lo cual se determina en el paso 9.

Finalmente, en el paso 10, se verifica si en el cluster donde se clasificó la Tierra éste fue el único planeta (de los 8 del sistema solar) en clasificarse ahí, si es así entonces se termina el proceso y dicho cluster corresponde al de los planetas más similares a la Tierra, paso 11; en caso contrario el proceso se repite sobre el grupo donde se encuentre clasificada la Tierra, para ello, se actualiza el conjunto de exoplanetas y el conjunto de planetas del sistema solar, pasos 13 y 14, respectivamente. En resumen, el proceso termina cuando la Tierra es el único planeta del sistema solar que se clasifica dentro de alguno de los clusters creados automáticamente.

Nótese que el algoritmo va filtrando la información para quedarse únicamente con los datos de los exoplanetas y los planetas del sistema solar que se encuentran en el cluster donde se clasifica la Tierra. Esto permitirá llevar a cabo una reducción de candidatos a planetas similares a la Tierra, enfocándose únicamente en este subconjunto de datos y descartando los datos de los demás clusters y planetas que no se clasificaron junto con la Tierra.

## **5. Análisis y resultados**

Con base al objetivo de este trabajo, se implementó el modelo con los diferentes algoritmos de agrupamiento y clasificación mencionados anteriormente para comparar su rendimiento. Para la implementación se usó el lenguaje de programación Python y sus bibliotecas `scikit-learn` para los algoritmos de clustering y `pandas` para el tratamiento de datos. En esta sección se presentan los resultados de las pruebas y su respectivo análisis.

En la Tabla 3 se pueden apreciar los resultados de cada algoritmo de agrupamiento en la primera iteración. Una vez agrupados los datos y después de clasificar los planetas del sistema solar se observó que el cluster en donde se clasificaba la Tierra era “representativo” de los demás en cuanto a la cantidad de exoplanetas que se agrupaban en éste, ya que prácticamente en todos los casos más del 50 % de los datos se agrupaban en dicho cluster. En la primera iteración, con `k-means++` y el parámetro  $k=8$  (porque son 8 los planetas del sistema solar), la Tierra se clasificó en el cluster etiquetado con 0, pero no sólo la Tierra, sino que todos los demás planetas del sistema solar también se clasificaron ahí.

Claramente este cluster era representativo y al analizarlo se encontró que en él se encontraban 4526 exoplanetas, correspondientes al 95 % de los datos, mientras que en el cluster 5 se agrupó el 4.7 % de los datos y el resto entre los demás clusters. Con `DBSCAN`, usando los parámetros por defecto  $eps=0.5$  y  $min\_samples=6$ , después de crear los clusters y de clasificar los planetas del sistema solar, cinco de estos se clasificaron en el cluster con etiqueta 1, incluida la Tierra. La relevancia de este cluster es que en él se agruparon 3002 exoplanetas, que corresponden al 63 % de los datos disponibles.

Con agglomerative clustering estableciendo el parámetro de `n_clusters = 8`, todos los planetas del sistema solar se clasificaron dentro del cluster etiquetado con 3, dicho cluster contenía 3909 exoplanetas, el 82 % de los datos totales. Para divisive clustering se mantuvo el mismo parámetro respecto al número de clusters y al igual que agglomerative clustering todos los planetas del sistema solar se clasificaron dentro del cluster etiquetado con 2 que también contenía 3909 exoplanetas, el 82 % de los datos. Por lo anterior, se puede concluir que, en la primera iteración todos los métodos de agrupamiento crearon grupos de tamaños muy desiguales, es decir, grupos desbalanceados (uno muy grande y otros demasiado pequeños).

Ahora se analizarán los clusters representativos. Es evidente que k-means++ fue el algoritmo que agrupó más exoplanetas en su primer cluster representativo, por lo que se compararon los datos del cluster representativo de cada algoritmo con respecto al de k-means++. En la comparación se encontró que los 3002 datos que agrupó DBSCAN también los agrupó k-means++ en un mismo grupo. De igual manera, los 3909 datos que agruparon agglomerative clustering y divisive clustering en un mismo grupo, forman parte de los 4526 datos que agrupó k-means++ en su cluster representativo. Por otra parte, agglomerative clustering y divisive clustering son algoritmos jerárquicos, y dado que en sus clusters representativos se agruparon la misma cantidad de datos, estos se compararon y se encontró que eran iguales en cuanto a los datos que agruparon en dicho cluster. Por lo tanto, en la primera iteración, todos los clusters representativos de los distintos métodos eran similares respecto a los datos que contenían.

A partir de los resultados obtenidos con los diferentes algoritmos de agrupamiento se pudo ver que el algoritmo agglomerative clustering junto con divisive clustering tuvieron el mejor desempeño. A continuación, se describe el proceso de entrenamiento del modelo sólo con agglomerative clustering (en la parte de agrupamiento del modelo propuesto). En la figura 1 se puede apreciar como a partir del conjunto de datos original, se van formando subconjuntos en los cuales se clasifican los planetas del sistema solar. Como se puede ver, se necesitó repetir un total de 4 veces el proceso de agrupar y clasificar hasta conseguir que la Tierra quedará en un cluster con 66 elementos del conjunto original sin ningún otro planeta del sistema solar.

Los resultados de cada iteración se muestran en la Tabla 4, donde cada columna se subdivide en dos columnas, la primera muestra la cantidad de exoplanetas agrupados (EA) en el cluster correspondiente, mientras que la segunda muestra el número de planetas del sistema solar clasificados (PC) en dicho cluster. Además, se incluyen los valores de las medidas de calidad del agrupamiento, el coeficiente de silueta  $S$  y el índice de Davies Bouldin  $I_{DB}$ . Esto permite concluir que, para el método aglomerativo, aunque la calidad del agrupamiento disminuía (gradualmente) los grupos eran balanceados lo que disminuyó el número de iteraciones necesarias para encontrar el grupo con exoplanetas similares exclusivamente a la Tierra.

Cabe mencionar que los mejores resultados con agglomerative clustering se obtuvieron al mantener el parámetro  $k = 10$  fijo en el método de clasificación k-nn. Mediante la experimentación se observó que al variar este parámetro según el tamaño del cluster (inicialmente 100, luego 50, 20 y 10, respecto a la iteración) el número de iteraciones realizadas aumentaba a 5 y el tamaño del grupo de planetas candidatos a ser similares a la Tierra incrementaba a 86.



Los resultados de k-means++ presentados en la Tabla 5 muestran que la calidad de los clusters disminuye con cada iteración y el tamaño de los grupos sigue desbalanceado (sus clusters representativos siguen agrupando una gran cantidad de exoplanetas), lo cual indica que se requieren de más iteraciones para obtener el resultado deseado. Además, no es posible determinar con exactitud cuantas iteraciones más se requieren para obtener resultados como los del agrupamiento jerárquico. En cuanto a DBSCAN, en la segunda iteración, el algoritmo ya solo generaba un cluster, claramente no proporcionó los resultados esperados.

## **6. Conclusiones y trabajo futuro**

Para la clasificación, el algoritmo k-nn proporcionó buenos resultados. En tanto que, para el agrupamiento los modelos jerárquicos, los métodos agglomerative clustering y divisive clustering, obtuvieron en general los mejores resultados pero, los resultados de k-means++ y DBSCAN no fueron nada esperanzadores. Es evidente que el mejor modelo de aprendizaje se consigue con agglomerative clustering o divisive clustering y k-nn, juntos logran encontrar un grupo de 66 exoplanetas con características similares a la Tierra de entre 5250 planetas.

Los resultados demuestran la importancia de seleccionar técnicas de aprendizaje automático apropiadas para el problema específico que se está abordando, ciertos algoritmos pueden ser más adecuados que otros para detectar patrones en los datos. El desempeño de los algoritmos también puede indicar qué características o variables son más relevantes para identificar exoplanetas parecidos a la Tierra. Para un trabajo futuro, se podría realizar una validación adicional de los resultados utilizando diferentes conjuntos de datos.

Adicionalmente, intentar mejorar la selección de características relevantes para encontrar exoplanetas similares a la Tierra. Finalmente, aunque se ha identificado un algoritmo de agrupamiento efectivo, aún es posible explorar otros algoritmos de agrupamiento y de clasificación para ver si alguno de ellos mejora los resultados. En definitiva, se ha demostrado que la tarea de descubrir planetas fuera del sistema solar y eventualmente encontrar un exoplaneta con condiciones de habitabilidad para los humanos es posible con la ayuda de algoritmos de aprendizaje automático.

## **Referencias**

1. Arthur, D., Vassilvitskii, S.: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics. pp. 1027–1035 (2007)
2. Aggarwal, C. & Reddy, C.: Data clustering: Algorithms and applications. Chapman and Hall/CRC (2013)
3. Herur, A., Tajmohamed, R., Ponsam, J.: Exploring Exoplanets using KNN, Logistic Regression and Decision Trees, 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), pp. 1–7 (2022) doi: 10.1109/ICSES55317.2022.9914278

4. Jin, Y., Yang, L., Chiang, C.: Identifying exoplanets with machine learning methods: A preliminary study. *International Journal on Cybernetics & Informatics (IJCI)*, pp. 31–42 (2022) doi: 10.5121/ijci.2022.110203
5. Martin, E., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, vol. 96, no. 34 pp. 226–231 (1996)
6. Pineda, C.: *Aprendizaje automático y profundo en Python*. RA-MA (2022)
7. Raschka, S., Mirjalili, V.: *Python machine learning: Aprendizaje automático y aprendizaje profundo con Python, scikit-learn y TensorFlow*. Marcombo (2019)
8. Rogers, J.: *The giant planet Jupiter*, Cambridge University Press, vol. 6 (1995)
9. Perryman, M.: *Exoplanet handbook*. Cambridge University Press (2018)
10. Rodríguez, H.: *Planeta Júpiter: El gigante gaseoso del sistema solar*. National Geographic (2023)

## Revisión de los criterios y análisis de diseño de GUI desde una perspectiva de género

Paulo César Portilla-Tirado<sup>1</sup>, Juan Villegas-Cortez<sup>2,3</sup>,  
Marco Vinicio Ferruzca-Navarro<sup>3</sup>, Román Anselmo Mora-Gutiérrez<sup>3</sup>

<sup>1</sup> Universidad Autónoma Metropolitana,  
Unidad Azcapotzalco,  
Departamento de Procesos y Técnicas de Realización,  
México

<sup>2</sup> Universidad Autónoma Metropolitana,  
Unidad Azcapotzalco,  
Departamento de Sistemas,  
México

<sup>3</sup> Universidad Autónoma Metropolitana,  
Unidad Azcapotzalco,  
Departamento de Investigación y Conocimiento del Diseño,  
México

{pcpt, juanvc, mvfn, mgra}@azc.uam.mx

**Resumen.** Hablar sobre género implica conceptualizarlo como una construcción social, cuyo desarrollo en diferentes sociedades tiene repercusiones según los roles definidos en los extremos de la masculinidad y la feminidad. Desde la década de los 80, cuando el uso de computadoras personales se popularizó, ha surgido un interés en comprender cómo se ha consolidado un sistema operativo vinculado no sólo a un tipo específico de computadora y función (ya sea personal, profesional o móvil), sino también en entender cómo su diseño contiene elementos relacionados al género, ya sea de los usuarios para los que se diseña o de quienes los han diseñado. En este sentido, presentamos la revisión de ocho Interfaces Gráficas de Usuario bajo una perspectiva de género, para proponer un instrumento de evaluación que sirva para entender el devenir de las mismas y su relación con la Inteligencia Artificial y sus técnicas.

**Palabras clave:** GUI, HCI, género.

### Review of the Criteria and Analysis of GUI Design from a Gender Perspective

**Abstract.** Talking about gender implies conceptualizing it as a social construct, whose development in different societies has repercussions according to the roles defined at the extremes of masculinity and femininity. Since the 1980s, when the use of personal computers became popular, there has been an interest in understanding how an operating system linked not only to a specific type of

computer and function (whether personal, professional, or mobile) has been consolidated, but also in understanding how its design contains elements related to gender, either of the users for whom it is designed or of those who have designed them. In this sense, we present the review of eight User Interface Designs from a gender perspective, to propose an evaluation tool that serves to understand their evolution and their relationship with Artificial Intelligence and its techniques.

**Keywords:** GUI, HCI, gender.

## **1. Introducción**

Interactuar con dispositivos como la computadora, los teléfonos inteligentes y las redes sociales, se ha convertido en un fenómeno común y entendido en las actividades personales: realizar un trámite con una dependencia gubernamental, solicitar una entrevista de empleo o expresar una idea por medio de un número limitado de palabras, son tareas que actualmente en la mayoría de los casos, se ejecutan por medio de plataformas digitales mediadas por Interfaces Gráficas de Usuario (GUI del inglés graphical user interface).

Estas herramientas además ejecutar tareas desde ambientes digitales y ofrecer un tipo de usabilidad, también “se incorporan a la vida cotidiana, en función del sentido de pertenencia e identidad de las personas” [6]. Categorías como género, raza y clase, además de otras distinciones sociales en las relaciones interpersonales, juegan un papel preponderante en el binomio humano-computadora y fundamentalmente, en la forma cómo los sujetos desarrollan y expresan su individualidad.

Ignorar estos vectores desde la interacción humano computadora IHC o HCI (HCI, del inglés Human Computer Interaction), como la disciplina que se ocupa del diseño, la evaluación y la implantación de sistemas informáticos interactivos para uso humano y del estudio de los principales fenómenos que los rodean, conduce a reforzar prejuicios y desigualdades a través de la falta de representación de grupos diversos generando una discriminación algorítmica. Y por otro lado, puede derivar en la reproducción de sesgos e ideas preconcebidas pues la tecnología es creada y desarrollada por seres humanos con sus propias perspectivas y experiencias.

Ante este escenario, surge la pregunta de cómo satisfacer las necesidades informáticas del mundo real sin perpetuar la marginación de las mujeres y de cualquier grupo en el ámbito tecnológico. Esta pregunta planteada por Bardzell [2], representa un desafío significativo en el dominio de la HCI, pues el avance tecnológico actual es continuo y acelerado a través de herramientas como el internet de las cosas (IoT) o la Inteligencia artificial (AI).

Históricamente, los estudios de género han puesto foco en la exclusión de grupos sociales en diversos frentes además del feminista. Las distinciones sociales, determinadas a partir de los mecanismos de poder, han conducido a la marginalización e invisibilización de grupos que no encuentran lugar en la lógica colectiva de la corriente principal.

Aunque en la historia reciente de la HCI, se ha observado un creciente avance conceptual y metodológico para el diseño y desarrollo de GUI, aún existe una escasez de investigación en lo que respecta al concepto de género como construcción social, tanto en los usuarios como en los diseñadores. Por ello proponer criterios de diseño bajo una perspectiva de género es fundamental para garantizar la equidad y justicia social. Este enfoque reconoce la diversidad de experiencias y desafíos que enfrentan las personas en la sociedad, abordando las intersecciones entre género, raza, clase social y otras identidades. De manera que, reconociendo estos cruces, es posible identificar y abordar sesgos en el desarrollo y la implementación de tecnologías, promoviendo la igualdad de oportunidades en el acceso y uso de las mismas.

## **2. Estado del arte**

Hoy en día, los postulados de teoría queer y los estudios interseccionales han derivado en una serie de cuestionamientos que impactan sobre la cultura general en México, modificando la manera de concebir al sujeto históricamente determinado por su condición binaria y occidental. Desde esta perspectiva ignorar las categorías de género, raza y clase puede traducirse en situaciones de exclusión del sujeto en los diferentes ámbitos en los que se desenvuelve.

Ejemplo de ello son los trámites o servicios primarios de identidad que se llevan a cabo a través de plataformas digitales gubernamentales –vía Internet–, cuyas GUI, en aras de conformar ambientes virtuales, ubicuos y universales, se configuran a partir de diseños que no reparan en las particularidades de la perspectiva de género interseccional y, por ende, las convierte en plataformas potencialmente excluyentes que no cumplen los objetivos para las que fueron generadas.

Dicha cuestión evidencia la necesidad de llevar a cabo una investigación que, considerando a los usuarios de dichas plataformas de servicios como agentes contingentes en cuanto a su identidad sexo-genérica, permita analizar su diseño y con ello, determinar las condiciones para el desarrollo de interfaces con una mayor experiencia de inclusión en cuanto a las categorías mencionadas.

Por otro lado, existen iniciativas innovadoras en el ámbito institucional como el procedimiento para actualizar los datos por cambio de género en la Universidad Autónoma Metropolitana (Figura 1), sin embargo, para tramitarlo primero es necesario hacer el levantamiento de una nueva acta de nacimiento para el reconocimiento de la identidad ante el Registro Civil en México, que es un trámite complejo y con comentarios negativos en cuanto a perspectiva de género por parte de los usuarios.

La HCI nace como disciplina en 1960 resultado del crecimiento y la evolución de las tecnologías de la información y la necesidad de comprender cómo los seres humanos y las computadoras interactúan de forma efectiva. Durante los primeros días de la informática, los operadores humanos desempeñaron un papel crucial en ayudar a las computadoras a llevar a cabo tareas que involucraban cálculos complejos. A medida que la disciplina evolucionó, el enfoque inicial en la interfaz hombre-máquina orientada a tareas se complementó con conceptos como la simbiosis hombre-máquina, propuesta por Licklider [5], y la mejora humana, introducida por Engelbart [3].



**Fig. 1.** Procedimiento para actualizar datos por cambio de género. Fuente: UAM, 2023.

El enfoque del diseño de los sistemas informáticos interactivos en este momento, resaltaba la necesidad de entender al humano a través del estudio de los fenómenos contextuales con apoyo de disciplinas como las ciencias sociales o cognitivas. En la década de los 80 y con el inicio de la masificación de la computadora personal, la HCI se ve inmersa en una encrucijada, pues por un lado se enfocaba en la mejora a través de las capacidades humanas y la aplicación de la psicología en individuos que trabajaban con dispositivos y aplicaciones; y por otro lado, la atención se dirigía a considerar a los humanos como usuarios, destacando una orientación que priorizaba la innovación, la creación de tecnología y enfoques metodológicos.

El camino por el que se condujo la disciplina fue el segundo con un enfoque principal en la innovación, invención y diseño tecnológicos y metodológicos en lugar de analizar los factores humanos. Centrando su atención en el concepto de usuario, Norman y Draper [8] destacan de manera explícita las necesidades y los intereses de los individuos en relación a un sistema informático. Emerge entonces el diseño centrado en el usuario (UCD), como una metodología de alto alcance desde la ingeniería pues sitúa al sujeto como una parte activa en la construcción, modificación y evaluación de un sistema computacional, convirtiéndose en una guía sistemática de usabilidad.

Paulatinamente, además del diseño de interacción, el diseño en general se impregna del mismo enfoque: centrar el proceso en los usuarios implica involucrarlos desde el comienzo en la dinámica creativa. No obstante, desde los estudios de género descoloniales, el término “usuario” es problemático pues esencializa a las personas en sujetos “cifrados” a través del esfuerzo para lograr una computación universal y ubicua, en lugar de matizar la individualidad como un atributo sustantivo a considerar en el diseño y desarrollo de interfaces.

Dourish y Mainwaring [4], consideran prioritario descolonizar la metodología de investigación de la informática actual. De la misma forma, el enfoque de la HCI feminista [2] resalta las implicaciones del concepto usuario en términos de exclusión: no considerar a grupos minoritarios o históricamente discriminados en el proceso de análisis, creación producción y comercialización de productos tecnológicos, contribuye a fomentar activamente dicha exclusión.

El género está relacionado de diversas formas en la investigación y la práctica de la HCI aun cuando no se aborde explícitamente en la parte del diseño y el desarrollo de productos tecnológicos. Una manera de analizar cómo las normas y valores de género se reflejan en tecnologías e interacciones específicas es mediante la exploración de los “guiones” de género [1]. Estos guiones están relacionados con diversos aspectos de los objetos tecnológicos, incluyendo su diseño físico, ubicación, interfaz, así como su estrategia de comercialización, publicidad y materiales de instrucción. Un ejemplo es la publicidad de marcas como Apple durante la década de los 90, producida para promocionar las actualizaciones en los computadores personales y los sistemas operativos, en la cual en su mayoría, los usuarios están representados por hombres, ejecutivos y de piel blanca.

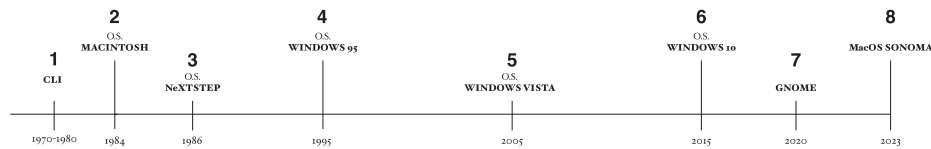
La evolución de las interfaces de usuario ha sido impulsada por avances en la tecnología de visualización, dispositivos de entrada y software de usuario. Desde los primeros sistemas de línea de comandos hasta las interfaces táctiles y gestuales de hoy en día, las interfaces de usuario han recorrido un largo camino y continúan evolucionando para adaptarse a las necesidades cambiantes de los usuarios y las demandas tecnológicas. La comprensión de la historia y el desarrollo de las interfaces de usuario es fundamental para apreciar cómo la interacción hombre-máquina ha llegado a ser lo que es hoy.

Con lo anterior las preguntas de investigación para nuestro trabajo son: ¿Qué implicaciones tienen la teoría queer y los estudios interseccionales en el desarrollo de servicios gubernamentales en línea por medio de GUI?, ¿Cuáles son los vectores de opresión, desde los estudios interseccionales, que derivan en una mayor experiencia de exclusión en México?, ¿Cómo se cruzan los vectores de opresión, para contribuir con experiencias de exclusión en México, respecto al uso de servicios gubernamentales en línea mediante GUI?, ¿Cómo se configura el concepto de heteronormatividad en el uso de servicios gubernamentales en línea mediante GUI, desde la teoría queer y los estudios interseccionales? Para lograr nuestro objetivo hemos desarrollado un sondeo de opinión visual entre la comunidad de varias universidades a las que se ha podido llegar el instrumento, en una primera fase de implementación que aquí compartimos y que se presenta en la siguiente sección.

## **2.1. Historia de la GUI**

La interfaz gráfica de usuario (GUI) es un software que permite a los usuarios interactuar con una computadora a través de símbolos, representaciones visuales y dispositivos de señalización. La GUI sustituyó a la interfaz de línea de comandos de las primeras computadoras con un sistema más intuitivo apoyado de elementos gráficos, lo que facilita no sólo aprender a usar la computadora, sino también hace la experiencia más agradable y natural.

En la actualidad, la GUI es la interfaz de computadora estándar y sus elementos se han convertido en elementos culturales icónicos. A continuación se mencionan ocho ejemplos representativos de la historia reciente del procesador personal, los cuales funcionarán como referencias para el análisis semántico propuesto más adelante.



**Fig.2.** Línea de tiempo de 8 GUI en la historia del computador personal. Fuente: elaboración propia.

## 2.2. CLI, 1970-1980

Una interfaz de línea de comandos (Command Line Interface, por sus siglas en inglés CLI) es un software que permite enviar instrucciones a programas informáticos o al sistema operativo (S.O. por sus siglas en inglés de “Operating System” ), utilizando líneas de texto simples. Estas órdenes pueden introducirse de forma interactiva o automatizada a través de scripts conocidos como archivos de procesamiento por lotes o batch.

La CLI es la primer sistema de interacción entre el usuario y la computadora sólo precedida por métodos como las tarjetas perforadas. Su disponibilidad es amplia, abarcando una variedad de programas, sistemas operativos y tipos de hardware, con funcionalidades diversas. Si bien la CLI no se considera una GUI, que posteriormente ofrecerá una experiencia más atractiva y simplificada, es contemplada para el estudio por su relevancia en la historia de la HCI y por sus características visuales.

## 2.3. S.O. Macintosh, 1984

Apple Computer introduce al mercado el primer computador personal con éxito comercial con una GUI. Esta revolucionó la forma cómo los usuarios interactuaban con las computadoras al ofrecer una experiencia visual e intuitiva. Los antecedentes del S.O. o se basan en desarrollos tecnológicos y decisiones estratégicas que precedieron su creación, como el trabajo en Xerox PARC durante la década de 1970, especialmente con el Xerox Alto, que sentó las bases con conceptos como las aplicaciones software, los dispositivos de entrada y salida o la Red de Área Local que permitía la comunicación entre varias computadoras.

Y por otro lado, la Apple Lisa también de Apple Computer, lanzada en 1983, que introdujo elementos visuales como ventanas e iconos, aunque su alto precio limitó su masificación. La GUI del Macintosh permitía a los usuarios manipular elementos en la pantalla utilizando un dispositivo de entrada llamado “ratón”. En lugar de depender de comandos escritos, los usuarios podían hacer clic en iconos, ventanas y menús para realizar tareas.

Los usuarios eran recibidos con un “escritorio” virtual adornado con imágenes que representaban archivos, carpetas y programas. Podían abrir y manipular múltiples “ventanas” que contenían aplicaciones y documentos. En la parte superior de la pantalla, una barra de menú proporcionaba acceso a opciones y comandos para la aplicación activa, como “Archivo”, “Editar” y “Ver”.



Una de las características más destacadas fue la introducción del “arrastrar y soltar”, que permitía a los usuarios mover archivos a la “impresora” para imprimirlos o a la “papelería” para eliminarlos. Así también, los usuarios podían organizar sus archivos y programas en “iconos” y “carpetas” para una mejor gestión.

#### **2.4. S.O. NeXTSTEP, 1986**

NeXTSTEP fue un sistema operativo multitarea orientado a objetos, fue creado por NeXT Computer, Inc. específicamente para uso en las computadoras NeXT. El sistema fue pionero en el uso de tecnologías como el entorno de desarrollo orientado a objetos y el lenguaje de programación Objective-C, que simplificaron el proceso de desarrollo de aplicaciones y fomentaron la creación de software robusto y escalable.

Otra característica relevante fue su arquitectura de software modular y multitarea, que proporcionaba estabilidad y rendimiento incluso en entornos exigentes. NeXTSTEP también se destacó por su capacidad para la creación de aplicaciones multimedia y de diseño gracias a herramientas como Digital Librarian e Interface Builder.

#### **2.5. S.O. Microsoft Windows, 1995**

También conocida como la versión 4, este sistema operativo desarrollado por Microsoft (MS), representó una evolución en la línea la marca comercial pues su arquitectura de 32 bits proporcionaba una estabilidad y rendimiento mejorados. Esta transición ofreció una mayor capacidad de procesamiento y una mejor gestión de la memoria, lo que resultó en un sistema más eficiente y robusto. Además, MS-Windows 95 introdujo un nuevo entorno de red integrado, lo que permitió a los usuarios conectarse y administrar redes de manera más efectiva. Esta integración simplificó las tareas de configuración y administración de redes, lo que mejoró la productividad y la facilidad de uso para los usuarios corporativos y domésticos.

Otra característica distintiva fue la incorporación del navegador nativo MS-Internet Explorer 3.0. Esta integración marcó el inicio de la navegación web integrada en el S.O. Con la inclusión de Internet Explorer 3.0, los usuarios tuvieron acceso directo a la World Wide Web (WWW) desde su sistema operativo, lo que facilitó la exploración de la creciente cantidad de recursos en línea disponibles en ese momento. Esta integración sentó las bases para futuras versiones de MS-Windows que se centrarían cada vez más en la conectividad a Internet.

#### **2.6. S.O. Microsoft Windows Vista, 2005**

Lanzado por Microsoft en enero de 2007, representó un paso significativo en la evolución de los sistemas operativos MS-Windows. Una de las características distintivas de MS-Windows Vista fue su interfaz de usuario renovada, conocida como Aero, que presentaba efectos visuales mejorados, transparencias y animaciones que proporcionaban una experiencia más estética y moderna. Igualmente, MS-Windows Vista introdujo el concepto de “Sidebar”, una barra lateral que albergaba

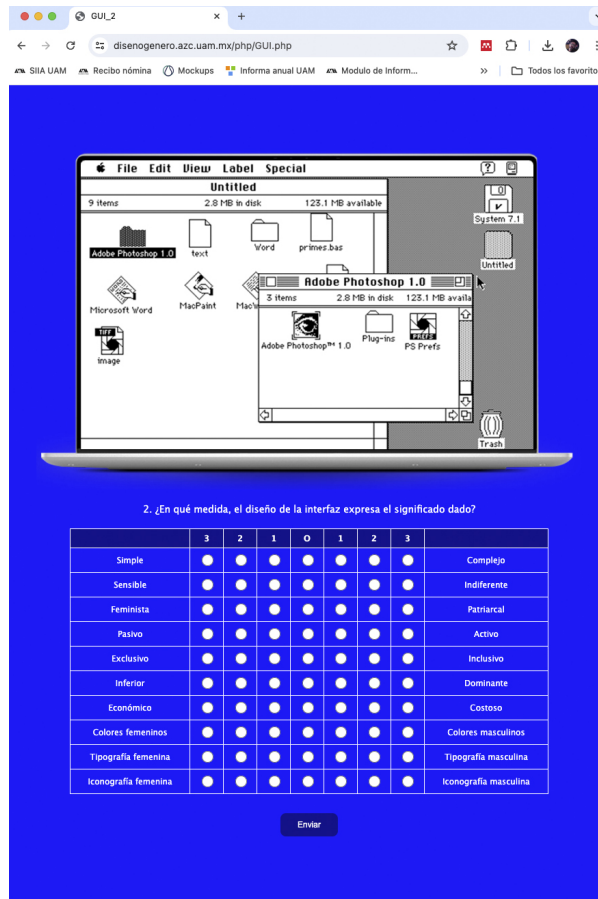


Fig. 3. Herramienta de análisis semántico en GUI. Fuente: elaboración propia.

miniaplicaciones conocidas como “Gadgets”, que ofrecían información en tiempo real como noticias, clima y calendario directamente en el escritorio. Otra característica importante de Windows Vista fue el enfoque en la seguridad. Se implementaron mejoras significativas en el sistema de seguridad, incluyendo el Control de Cuentas de Usuario (UAC), que requería confirmación de los usuarios para realizar ciertas acciones que pudieran afectar al sistema, con el objetivo de proteger contra malware y ataques cibernéticos. Además, se introdujo Windows Defender, un programa antimalware integrado para proteger contra virus, spyware y otras amenazas en línea.

MS-Windows Vista también incluyó mejoras en la búsqueda y organización de archivos con la introducción de la función “Instant Search”, que permitía a los usuarios encontrar archivos y documentos de manera rápida y eficiente directamente desde el menú de inicio. A su vez, se implementaron mejoras en el rendimiento y la fiabilidad del sistema, aunque la recepción inicial de Windows Vista estuvo marcada por críticas debido a problemas de compatibilidad de hardware y rendimiento.

## **2.7. S.O. Microsoft Windows 10, 2015**

Una de sus características más destacadas es la reintroducción del menú de inicio, que combina elementos del menú de inicio clásico de MS-Windows 7 con los mosaicos dinámicos introducidos en MS-Windows 8. Esta combinación ofrece a los usuarios una experiencia familiar y moderna al mismo tiempo. Así también, MS-Windows 10 introdujo el concepto de “Continuidad del Escritorio”, que permite a los usuarios trabajar de manera fluida en múltiples dispositivos con MS-Windows 10, sincronizando aplicaciones y configuraciones a través de la nube.

Otra característica distintiva es la integración de la asistente virtual Cortana, que proporciona funciones de búsqueda y asistencia por voz, así como recordatorios y notificaciones personalizadas. Cortana está diseñada para adaptarse al usuario y ofrecer sugerencias contextuales basadas en el uso del dispositivo. Además, MS-Windows 10 introdujo el navegador web MS-Edge, que reemplaza a MS-Internet Explorer como el navegador predeterminado del S.O. Edge ofrece características avanzadas de seguridad y rendimiento, así como integración con servicios en la nube de Microsoft.

MS-Windows 10 también se destaca por su enfoque en la seguridad y la protección de datos. Introdujo características como MS-Windows Hello, que permite a los usuarios iniciar sesión de manera segura utilizando reconocimiento facial o de huellas dactilares, y Windows Defender, una suite de seguridad integrada que ofrece protección contra virus, malware y amenazas en línea.

## **2.8. GNOME, 2020**

Entorno de escritorio de software libre y de código abierto que ofrece una experiencia de usuario intuitiva y personalizable en sistemas operativos basados en GNU Linux. Una de las características distintivas de GNOME es su enfoque en la simplicidad en el diseño de la interfaz de usuario. Con un apariencia minimalista, proporciona un entorno de trabajo eficiente y sin distracciones. Además ofrece una amplia gama de aplicaciones integradas y herramientas de productividad, incluyendo un navegador web, un gestor de archivos, reproductores multimedia y herramientas de oficina. Estas aplicaciones están diseñadas para integrarse en el entorno de escritorio, y proporcionar una experiencia de usuario coherente.

El entorno de escritorio GNOME se caracteriza por ser accesible para todos los usuarios, independientemente de sus capacidades físicas o cognitivas. Esto se logra mediante el diseño de características y herramientas de accesibilidad, como lectores de pantalla, teclado de acceso rápido y soporte para dispositivos de entrada alternativos. Además, GNOME es altamente personalizable, permitiendo a los usuarios adaptar el entorno de escritorio a sus necesidades y preferencias individuales. Los usuarios pueden cambiar el diseño, el tema y la disposición de los elementos de la interfaz de usuario para crear un entorno de trabajo personalizado.

## **2.9. Apple MacOS Sonoma, 2023**

Vigésima versión principal de MacOS, el sistema operativo de escritorio de Apple para ordenadores Mac.

**Tabla 1.** Género y edades en el instrumento en la fase primera.

<b>Género:</b>	<b>Edad (años):</b>		
Mujeres	69	Menos de 20	15
Hombres	92	20 a 30	67
No binaria	2	31 a 40	24
		41 a 50	28
		Más de 51	29

Las mejoras que destacan son la accesibilidad, la gestión de archivos, el centro de control y las notificaciones. En el ámbito de la accesibilidad, MacOS Sonoma introduce innovaciones para ofrecer una experiencia más inclusiva. Se incorporan nuevas opciones de control diseñadas para personas con discapacidades motoras, como el control por voz y el control por movimiento.

Además, se implementan mejoras en VoiceOver, permitiendo una navegación más ágil y precisa en aplicaciones y páginas web. En cuanto a la gestión de archivos, MacOS Sonoma facilita aún más la tarea de encontrar y organizar archivos. La función de búsqueda se ha mejorado, proporcionando resultados más precisos y rápidos. Asimismo, se añaden nuevas opciones de etiquetado y metadatos para simplificar la clasificación y organización de los archivos.

El centro de control es otra área que experimenta cambios con la llegada de MacOS Sonoma. Este nuevo centro brinda acceso rápido a configuraciones y controles esenciales, como el brillo de la pantalla, el control de volumen y el modo silencio, entre otros. Además, ofrece la posibilidad de personalizar accesos directos y agregar opciones favoritas para acceder de manera más eficiente a las funciones más utilizadas. En la Figura 2 presentamos un resumen de la línea de tiempo de la evolución de las interfases de escritorio, como herramienta fundamental de la computadora con un S.O. y la persona usuario.

### 3. Metodología

El género está relacionado de diversas formas en la investigación y la práctica de la HCI aún cuando no se aborde explícitamente en la parte del diseño y el desarrollo de productos tecnológicos. Una manera de analizar cómo las normas y valores de género se reflejan en tecnologías e interacciones específicas es mediante la exploración de los “guiones” de género [1]. Estos guiones están relacionados con diversos aspectos de los objetos tecnológicos, incluyendo su diseño físico, ubicación, interfaz, así como su estrategia de comercialización, publicidad y materiales de instrucción.

Con el fin de analizar cómo las GUI reproducen un tipo de discurso relacionado con el género, se presenta el prototipo de una herramienta bajo el concepto de guiones de género de Akrich. Para ello se trabaja con el método de diferencial semántico, técnica utilizada en la psicología y la investigación de mercado para medir las actitudes y percepciones de las personas hacia un determinado objeto, idea o concepto. Este modelo diferencial permite aprehender el modo de percepción de un fenómeno dado por medio

de la aproximación al valor connotativo que un individuo atribuye a un fenómeno cualquiera [7]. La técnica consiste en presentar a los participantes una batería de 10 pares de adjetivos opuestos, colocados en los extremos de una escala de siete valores.

Estos adjetivos describen las dimensiones relevantes del objeto o concepto evaluado, además de relacionarse con los tres vectores de exclusión desde los estudios interseccionales: género, raza y clase.

La lista completa de adjetivos es la siguiente: Simple  $\longleftrightarrow$  Complejo; Sensible  $\longleftrightarrow$  Indiferente; Feminista  $\longleftrightarrow$  Patriarcal; Pasivo  $\longleftrightarrow$  Activo; Exclusivo  $\longleftrightarrow$  Inclusivo; Inferior  $\longleftrightarrow$  Dominante; Económico  $\longleftrightarrow$  Costoso; Colores femeninos  $\longleftrightarrow$  Colores masculinos; Tipografía femenina  $\longleftrightarrow$  Tipografía masculina; Iconografía femenina  $\longleftrightarrow$  Iconografía masculina.

Los participantes deben ubicar un punto en la escala que represente su percepción o actitud hacia la GUI mostrada en la parte superior como lo muestra la Figura 3.

#### 4. Implementación técnica y despliegue abierto en la internet

La herramienta se alojó en un servidor de la Universidad Autónoma Metropolitana Unidad Azcapotzalco, con la siguiente URL: <https://disenogenero.azc.uam.mx>. Para su desarrollo se trabajó con las tecnologías front-end HTML, CSS y JavaScript. Y en la parte de back-end se trabajó con el lenguaje PHP, junto con una base de datos para gestionar datos de identidad y las respuestas de los reactivos del diferencial semántico. Se diseñó bajo un formato responsivo, garantizando una experiencia óptima tanto en computadoras de escritorio como en dispositivos móviles.

#### 5. Resultados

En total respondieron al instrumento 163 personas, con las siguientes características, tal como se muestra en la Tabla 1.

##### 5.1. Análisis de resultados

Para el análisis de resultados y su representación visual se trabajó con diagramadas de caja y mapas de calor como se muestra en la Figura 4 y se describe a continuación. En la Figura 4(a), las GUI 1, 6 y 8 son asociadas con los conceptos {Complejo, Indiferente, Patriarcal, Activo, Inclusivo, Dominante, Costoso, Masculino}. En contraste las 4 y 7 se asocian a los conceptos {Simple, Sensible, Feminista, Pasivo, Exclusivo, Inferior, Económico, Femenino}.

Por otro lado las GUI restantes se encuentran en un espectro neutro, en decir, se encuentran dos grupos de interfaces entre los cuales se presenta un alto grado de diferencial semántico sobre las ideas a las que asocian estos grupos son el formado por las interfaces 1, 6 y 8; por otro lado el segundo grupo lo forman las interfaces 4 y 7. En la Figura 4(b) se observa que las personas de 40 a 50 años en promedio asocian a ideas tales como {Simple, Sensible} las GUI analizadas; mientras que las personas con más de 50 años y el grupo 30 a 40 años las asocian a ideas tales como {Complejo,

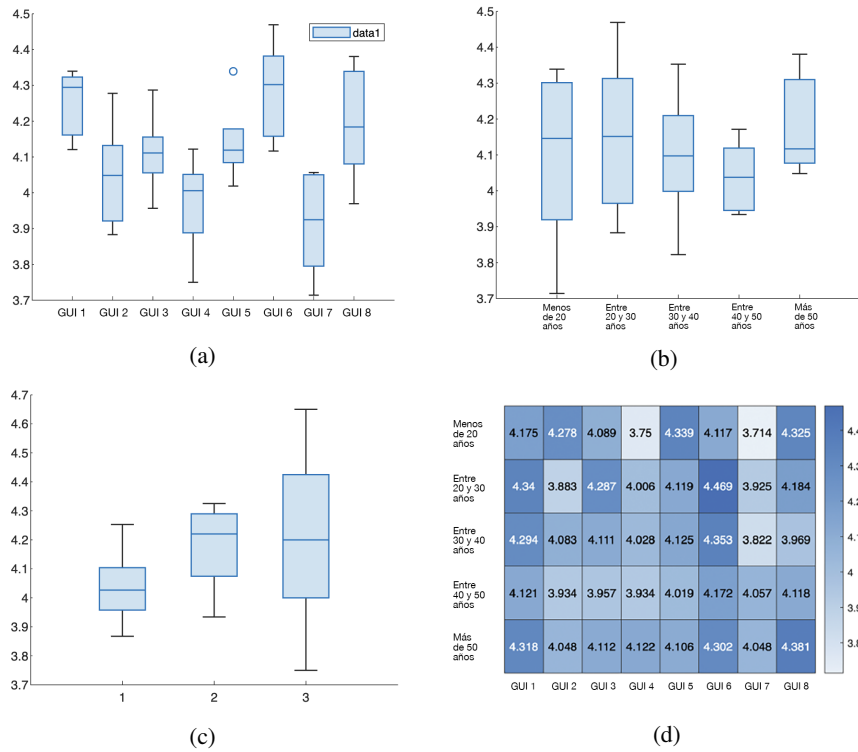


Fig. 4. Resultados de la aplicación del instrumento de sondeo en primera fase de experimentación.

Indiferente}. Estos grupos de edad presentan poca dispersión en las opiniones si se contrastan con los grupos de edades menores como de 20 años y de 20 a 30 años. En la Figura 4(c) se observa que en el género femenino existe poca dispersión en las opiniones tendiendo ligeramente a asociar las GUI a {Simple, Sensible}, etc. Mientras que el género masculino y otros géneros presentan una mayor dispersión asociando las interfaces a las ideas {Complejo, Indiferente}.

Finalmente, en la Figura 4(d) se muestra el promedio de las apreciaciones de cada interfaz y los grupos de edad. Se observa que el grupo de menos de 20 años valora a las interfaces 4 y 7 con ideas tales como femenino y a la interfaces 5 y 8 como masculinas. En contraste el grupo de edad 40–50 presenta una apreciación semántica a las GUI que tienden a la neutralidad.

## 6. Conclusiones

En este trabajo hemos presentado un primer avance en la revisión de los criterios y análisis de diseño de las GUI desde una perspectiva de género. Nos hemos apoyado en una metodología de sondeo de opinión con base a las 8 GUI más relevantes según el estado del arte para los sistemas operativos de computadoras tipo PC principalmente, o

de uso personal. Una de nuestros objetivos ha sido poder visualizar la respuesta ante la interfase por parte del usuario, en una oportunidad de verse a sí mismo como sujeto de investigación para conocer sus impresiones y reacciones ante la computadora a través de lo que percibe en una pantalla o monitor de la computadora en turno. Para esta primera fase experimental las personas participantes han sido de nacionalidad mexicana.

Se debe mencionar que los efectos sobre la apreciación y asociación de las interfaces a un conjunto de ideas se ve influenciados directamente con el grupo de edad. En contraste el género de los participantes provoca un efecto menor, sin embargo la apreciación de las personas de género femenino es menos dispersa que la masculina.

Como trabajo futuro se estudiarán los diferentes niveles de percepción con nuevas personas participantes en el estudio, buscando llegar a más personas con diferente nacionalidad y por ende una perspectiva distinta a la mexicana, y en esto les invitamos cordialmente a sumarse en responder el instrumento en la URL proporcionada; así también se planea poder establecer nuevas relaciones que asocien aspectos de identidad de género y el impacto en perspectivas de desarrollo profesional o de un uso particular.

## Referencias

1. Akrich, M.: The De-description of Technical Objects. *Shaping Technology/Building Society Studies in Sociotechnical Change*, pp. 205–224 (1992)
2. Bardzell, S.: Feminist HCI: Taking stock and outlining an agenda for design. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, vol. 2, pp. 1301–1310 (2010) doi: 10.1145/1753326.1753521
3. Chignell, M., Wang, L., Zare, A., Li, J.: The evolution of HCI and human factors: Integrating human and artificial intelligence. *ACM Transactions on Computer-Human Interaction*, vol. 30, no. 2, pp. 1–30 (2023) doi: 10.1145/3557891
4. Dourish, P., Mainwaring, S. D.: Ubicomp's colonial impulse. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 133–142 (2012) doi: 10.1145/2370216.2370238
5. Licklider, J.: Man-computer symbiosis. *Human Factors*, , no. 1, pp. 4–11 (1960) doi: 10.1109/THFE2.1960.4503259
6. Light, A.: HCI as heterodoxy: Technologies of identity and the queering of interaction with computers. *Interacting with Computers*, vol. 23, no. 5, pp. 430–438 (2011) doi: 10.1016/j.intcom.2011.02.002
7. Moles, A., Rohmer, E., Covarrubias, J.: *Las ciencias de lo impreciso* (1995)
8. Norman, D., Draper, S.: *User centered system design: new perspectives on human-computer interaction*. N.Y.C. (1986)





Electronic edition  
Available online: <http://www.rcs.cic.ipn.mx>



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



Centro de Investigación  
en Computación