EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 154 No. 8
August 2025

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:

Alejandra Ramos Porras

***RESEARCH IN COMPUTING SCIENCE**, Año 25, Volumen 154, No. 8, Agosto de 2025, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, otorgado por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 08 de Agosto de 2025.*

***RESEARCH IN COMPUTING SCIENCE**, Year 25, Volume 154, No. 8, August, 2025, is a monthly publication edited by the National Polytechnic Institute through the Center for Computing Research. Av. Juan de Dios Bátiz S/N, Esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, C.P. 07738, Mexico City, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor in charge: Dr. Grigori Sidorov. Reservation of Exclusive Use Rights of Title No. 04-2019-082310242100-203. ISSN: pending, granted by the National Copyright Institute. Responsible for the latest update of this issue: the Computer Research Center, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Last modified on August 8, 2025.*

Advances in Artificial Intelligence

Obdulia Pichardo-Lagunas (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2025

ISSN: in process

Copyright © Instituto Politécnico Nacional 2025
Formerly ISSNs: 1870-4069, 1665-9899

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

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

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

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

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Un estudio exploratorio para clasificación de electrocorticogramas en estado basal y crisis epilépticas a través de redes neuronales

Ángel Luis Yoval[1], Martha Lorena Avendaño-Garrido[1], Porfirio Toledo[1], María-Leonor López-Meraz[2]

[1] Universidad Veracruzana,
Facultad de Matemáticas,
México

[2] Universidad Veracruzana,
Instituto de Investigaciones Cerebrales,
México

zs20016134@estudiantes.uv.mx, maravendano@uv.mx, ptoledo@uv.mx
leonorlopez@uv.mx

Resumen. Las señales electrocorticográficas (ECoG) registran la actividad eléctrica del cerebro a lo largo del tiempo, por lo que son ampliamente utilizadas para el estudio y análisis de crisis epilépticas. En este trabajo se utiliza un modelo de aprendizaje estadístico basado en redes neuronales artificiales para la clasificación de muestras de estas señales en estados basal y de crisis epiléptica. A través de un enfoque de aprendizaje automático supervisado, se ajusta un modelo de perceptrón multicapa entrenado con muestras de una señal ECoG obtenida de una rata. El modelo ajustado logra identificar patrones característicos de las muestras, permitiendo discriminarlas en estado basal de la actividad eléctrica cerebral y la presencia de crisis epiléptica.

Palabras clave: electrocorticograma, redes neuronales artificiales, crisis epiléptica, clasificación, aprendizaje supervisado.

Application of Neural Networks to EEG Signal Classification

Abstract. This study presents a statistical learning model using artificial neural networks aimed at the classification of EEG signals. EEG signals are a time-based recording of the brain's electrical activity; these signals are of utmost importance in the detection, diagnosis, and monitoring of epileptic seizures. Through a machine learning framework based on supervised learning techniques, a model is developed that balances computational time and accuracy, capable of learning and recognizing the characteristics of the basal, intermediate, and seizure phases that compose an EEG signal.

Keywords: EEG signals, neural networks, epilepsy, classification, supervised learning.

1. Introducción

La Organización Mundial de la Salud reconoce que la epilepsia es una enfermedad cerebral crónica no transmisible que afecta a unos 50 millones de personas en todo el mundo. Se caracteriza por crisis recurrentes, algunas de las cuales son episodios breves de movimiento involuntario que pueden involucrar una parte del cuerpo (parcial) o todo el cuerpo (generalizado), lo que tradicionalmente se conoce como convulsiones; en ocasiones se acompañan de pérdida de conciencia y control de la función intestinal o vesical [26]. Según estimaciones del gobierno federal, en México, de 10 a 20 personas de cada 1000 sufren epilepsia [11].

Hay diferentes signos y síntomas asociados a esta enfermedad, el más reconocible es la aparición de crisis; estos pueden variar y dependen de la zona del cerebro en la que inicia la alteración. La importancia de investigar esta enfermedad radica en poder ofrecer una mejor calidad de vida a quienes la padecen.

Algunos estudios y análisis de la epilepsia se realizan con datos provenientes de modelos animales, los cuales son clave en la investigación biomédica porque se pueden replicar procesos biológicos o enfermedades. Se busca evitar la realización de estos experimentos y tratamientos en pacientes humanos hasta contar con resultados suficientes que garanticen su seguridad. En modelos animales, las señales electrocorticográficas (ECoG) permiten registrar la actividad eléctrica por medio de electrodos colocados sobre la corteza cerebral. Estos registros son útiles si se desea analizar los patrones característicos de las crisis y localizar las zonas afectadas del cerebro [15].

Los patrones en estas señales son indicadores importantes que ayudan en el diagnóstico y clasificación de los tipos de epilepsia, por lo cual es importante reconocerlos. De acuerdo con Niedermeyer y Lopes da Silva [16] algunos de los patrones característicos son:

- *Punta-onda generalizada.* Este patrón es característico de epilepsias generalizadas. Las descargas de punta-onda suelen aparecer de manera sincrónica en ambos hemisferios cerebrales, indicando una afectación global en la actividad cerebral.
- *Descargas focales.* Comunes en epilepsias parciales, estas descargas se limitan a una región específica del cerebro, como el lóbulo temporal. Este patrón resulta especialmente útil para localizar el foco epileptogénico, ayudando a los médicos a identificar la zona cerebral donde se originan las crisis.
- *Picos repetitivos.* Este patrón se observa con frecuencia en algunos tipos de epilepsia del lóbulo frontal, siendo un indicador de actividad eléctrica irregular que puede acompañar a movimientos o espasmos.

El análisis y clasificación de estas señales juega un papel fundamental en la identificación de los distintos estados asociados a las crisis epilépticas. Una clasificación adecuada permitiría desarrollar mejores diagnósticos y estrategias de atención efectivas.

En este contexto, los avances en las herramientas computacionales permiten la implementación de técnicas de aprendizaje en el reconocimiento automático de patrones en estas señales. El uso de redes neuronales artificiales ha ganado relevancia por su capacidad de extraer características de forma automática, sin requerir gran manipulación de los datos [20], brindando aportes significativos a la clasificación de estas señales en comparación con enfoques tradicionales. Estos modelos han sido aplicados en la predicción de la edad cerebral [6], el diagnóstico de la esquizofrenia [19], el reconocimiento de emociones [10], la clasificación de estados de sueño y la interpretación de potenciales relacionados con eventos [7], así como en la detección de crisis epilépticas [25].

La identificación automática de patrones asociados a crisis epilépticas puede apoyar a los profesionales de la medicina a manejar grandes volúmenes de información, para realizar con mayor rapidez y precisión la detección de anomalías en las señales.

A lo largo de este estudio, se utilizarán muestras de señales de las áreas frontal y parietal del cerebro de una rata, como fuente de datos para el entrenamiento y validación del modelo. La corteza frontal está implicada en el control y aprendizaje motor, así como en la atención, mientras que la corteza parietal es un área sensorial multimodal, involucrada principalmente en la navegación en el espacio y en la memoria espacial (ver [3,14,18,23]). Los experimentos clásicos del grupo de Turski y Cavalheiro (ver [4,24]), mostraron que la pilocarpina induce *status epilepticus* (SE) de crisis motoras en ratas, asociadas a actividad electrográfica inicialmente en el hipocampo, misma que se propaga rápidamente hacia las cortezas occipital y sensoriomotora. En este contexto se ha observado que la región fronto-parietal de la corteza cerebral de la rata se activa durante el SE generado por pilocarpina [22]. Se sabe que la corteza frontal es altamente sensible a los cambios bioquímicos generados por el SE y que pueden derivar en daño neuronal (ver [5,9]). No obstante, la corteza parietal también presenta afectación celular debida al SE [24]. Lo anterior es relevante considerando que ambas presentan hiperactividad neuronal sostenida, lo que las hace más sensibles a procesos neurodegenerativos. Sin embargo, existen más detalles sobre la respuesta de la corteza frontal durante las crisis que la de la corteza parietal. De hecho, cuando se presenta el SE, existe una interacción aumentada y aberrante entre el hipocampo y la corteza frontal, siendo dicha interacción crítica para la generación y progresión de las descargas epilépticas [8].

En este trabajo se presenta una estrategia de clasificación de muestras de señales ECoG de una rata con el objetivo de contribuir al desarrollo de herramientas eficientes en el estudio de la epilepsia, lo que potencialmente podría mejorar la toma de decisiones clínicas.

La organización del presente documento se establece de la siguiente manera: en la sección 2, se detalla la metodología aplicada, incluyendo el procedimiento

para la obtención de las muestras de las señales tanto en estado basal como en crisis epiléptica. Se aborda también las características principales de las redes neuronales artificiales empleadas en el desarrollo del experimento computacional. En la sección 3, se presentan los experimentos computacionales realizados y sus resultados. Finalmente, en la sección 4 se presenta una discusión sobre los hallazgos obtenidos.

2. Metodología

En esta sección se describen los datos experimentales utilizados, el procedimiento para su adquisición y procesamiento, así como las características del modelo computacional empleado para su análisis.

2.1. Datos

Los datos estudiados fueron obtenidos en el Instituto de Investigaciones Cerebrales de la Universidad Veracruzana. Estos provienen de una rata Wistar, como sujeto experimental, alojada junto con otros especímenes en condiciones ambientales controladas de temperatura y humedad (20–28°C y 40–70 % HR, respectivamente), con ciclos de luz-oscuridad de 12/12 horas (07:00–19:00) y acceso ilimitado a agua y alimento. Para la adquisición de datos, se implantaron en el cráneo dos electrodos de acero inoxidable con forma de clavo de manera estereotáxica en la corteza frontal (anteroposterior = 1.5 mm, lateral a la línea media = ± 3 mm, respecto al bregma) y corteza parietal (anteroposterior = −3.5 mm, lateral a la línea media = ± 3 mm, respecto al bregma). El procedimiento quirúrgico se llevó a cabo bajo condiciones completamente asépticas, utilizando isoflurano como anestesia (1.5–2 %, marca Vedco, Inc., USA). La temperatura corporal se mantuvo con un sistema de regulación térmica (marca FHC, modelo 41-90-D8). Tras la cirugía, el espécimen fue rehidratado con solución salina glucosada (equivalente al 5 % del peso corporal, vía subcutánea). Durante el periodo posoperatorio, se administraron un analgésico (meglumina, 2.5 mg/kg, s.c., por 2 días) y un antimicrobiano (enrofloxacina, 5 mg/kg, s.c., por 5 días). La rata fue sometida al protocolo experimental 5 días después de la cirugía.

Respecto a la elección de estas señales cerebreles, estudios previos que emplean el modelo experimental SE usado en este trabajo, han mostrado que la corteza cerebral presenta actividad eléctrica paroxística (indicativa de un estado de hiperexcitabilidad e hipersincronía neuronal) asociada con las manifestaciones motoras propias de las crisis epilépticas (ver [4,5,8,24]). Este patrón es similar a la actividad eléctrica de las neuronas corticales que se detecta con el electroencefalograma (EEG) en humanos (ver [17,21]).

Para inducir las crisis epilépticas en el sujeto experimental, se utilizó la pilocarpina, el cual es un fármaco parasimpaticomimético (ver [24,27]), que estimula los receptores, denominados M1, en el sistema nervioso central provocando actividad epiléptica sostenida similar a los pacientes con esta enfermedad. Para nuestro caso, se administró a la rata cloruro de litio (3 mEq/kg

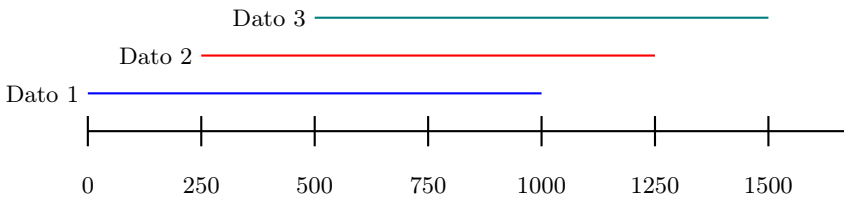

Fig. 1. Muestras con traslape y con etiqueta.

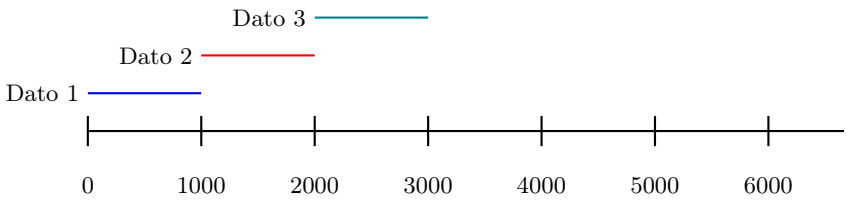

Fig. 2. Muestra sin traslape y sin etiqueta.

i.p., Sigma) y, 24 horas después, se inyectó clorhidrato de pilocarpina (30 mg/kg s.c., Sigma). El cloruro de litio potencia los efectos convulsivos de la pilocarpina facilitando el SE. Este modelo experimental se utiliza ampliamente en la investigación de la epilepsia, pues permite replicar aspectos de la enfermedad, en particular los episodios de crisis recurrentes no autolimitadas o SE.

Bajo el procedimiento descrito se obtuvieron dos señales de 40 minutos, correspondientes a la corteza frontal y la parietal, que iniciaban en un estado basal (independiente de la administración de pilocarpina y en libre movimiento) y terminaban en episodios de crisis epilépticas. En total se obtuvieron 781,000 mediciones aproximadamente, de cada una de las dos señales.

Para el ajuste y validación del modelo de aprendizaje supervizado, se tomaron muestras de la señal correspondientes a los 10 minutos iniciales (clase/etiqueta: *estado basal*) y muestras de los 10 minutos finales (clase/etiqueta: *estado en crisis*) de las señales. Las muestras corresponden a una partición de estos segmentos de 10 minutos en secciones de dimensión 1000, con un traslape de 250 mediciones, como se ilustra en la Figura 1. Con lo anterior se obtuvo un total de 1200 muestras etiquetadas por señal (frontal y parietal).

A cada muestra se le asignó una etiqueta, dependiendo del fragmento de la señal al que pertenecía; 0 si provenía de los primeros 10 minutos (clase: *estado basal*) o 1 si era de los últimos 10 (clase: *estado en crisis*).

Posterior a la validación del modelo, para analizar las señales completas, estas fueron divididas en muestras de tamaño 1000 siguiendo un proceso similar al anterior, con la particularidad de que no se utilizó traslape entre ellas, como se ilustra en la Figura 2. De esta forma, se obtuvo un total de 781 datos sin etiqueta.

2.2. Redes neuronales

En el ámbito de redes neuronales existen diferentes tipos de modelos como las redes monocapa, redes convencionales o recurrentes. En este trabajo se decidió utilizar el modelo de perceptrón multicapa, ya que es capaz de resolver problemas de clasificación no linealmente separables (ver [1] y [2]). Esta característica lo convierte en una opción adecuada para alcanzar los objetivos planteados.

Se emplearon diferentes funciones de activación para la red, todas comparten la característica de que su rango sea el intervalo $[0, 1]$, lo cual nos permite interpretar los resultados en términos probabilísticos. Entre las funciones utilizadas se encuentran la *sigmoide*, *ReLu* y *tanh* [12]. Se utilizó el algoritmo de optimización *Adaptive Moment Estimation* (Adam) [13].

2.3. Desarrollo

Para la búsqueda y análisis de las mejores arquitecturas se realizaron 100 ajustes independientes. Para el entrenamiento de la red se utilizó una proporción del 70 % de los datos etiquetados y para la evaluación se tomó el 30 % restante, obteniendo el promedio de tasa de error.

Una vez seleccionada la arquitectura con menor tasa de error, se volvió a ajustar la red con la muestra completa de datos etiquetados y, finalmente, se analizó la señal completa sin etiquetas. Se graficó el número de secuencia de la muestra de la serie de tiempo (consecutivamente) versus la probabilidad de que se corresponda con la clase *estado en crisis*.

Cabe señalar que se muestra la probabilidad de algunos datos que fueron utilizados en el entrenamiento del modelo, correspondientes al estado basal y al estado en crisis. No obstante, el interés principal radica en la sección intermedia de la señal, con el objetivo de visualizar el comportamiento de las probabilidades a lo largo del tiempo.

3. Experimentos

Los códigos para la experimentación fueron creados en el lenguaje de programación *Python* haciendo uso de las librerías *tensorflow*, *sklearn*, *pandas* y *numpy*. Se probaron múltiples configuraciones de redes neuronales, como se detalla en las Tablas 1 y 2. Los experimentos incluyeron redes con diferentes cantidades de capas, neuronas y funciones de activación. En las tablas se muestran las más destacadas. Cabe observar que, en el caso de la señal parietal, la única función sobresaliente fue la *sigmoide*.

Como se observa, las arquitecturas más eficaces para los datos considerados fueron las integradas por la función de activación *sigmoide*. En el caso de la señal frontal, la mejor arquitectura evaluada fue la compuesta por tres capas ocultas, con 30 neuronas por capa, la cual tiene un error promedio de 0.331472 y una varianza de 0.000180. Por su parte, para la señal parietal, la arquitectura de tres capas y 25 neuronas por capa alcanzó un error promedio de 0.205194 y una varianza de 0.000098.

Tabla 1. Resultados de la red neuronal para la señal frontal.

Función	#Capas	#Neuronas por capa	Error promedio	Varianza
Sigmoide	2	30	0.338278	0.000184
Sigmoide	3	15	0.364944	0.000645
Sigmoide	3	20	0.351056	0.000437
Sigmoide	3	25	0.338556	0.000216
Sigmoide	3	30	**0.331472**	**0.000180**
Tanh	3	15	0.382444	0.001915
Tanh	3	20	**0.375250**	**0.001813**
Tanh	3	25	0.378778	0.002199
Tanh	3	30	0.381472	0.002658
ReLu	3	10	0.368444	0.002972
ReLu	3	15	**0.361222**	**0.002335**
ReLu	3	20	0.378889	0.002402
ReLu	3	25	0.392833	0.002141
ReLu	3	30	0.398417	0.001714

Tabla 2. Resultados de la red neuronal para la señal parietal.

Función	#Capas	#Neuronas por capa	Error promedio	Varianza
Sigmoide	3	10	0.230278	0.000501
Sigmoide	3	15	0.217000	0.000240
Sigmoide	3	20	0.207250	0.000163
Sigmoide	3	25	**0.205194**	**0.000098**
Sigmoide	3	30	0.206583	0.000106
Tanh	3	10	0.439389	0.015284
Tanh	3	15	0.446194	0.013166
Tanh	3	20	**0.430361**	**0.013939**
Tanh	3	25	0.449194	0.016075
Tanh	3	30	0.465000	0.015932
ReLu	3	10	**0.447722**	**0.003660**
ReLu	3	15	0.464000	0.003229
ReLu	3	20	0.480222	0.002040
ReLu	3	25	0.489917	0.002786
ReLu	3	30	0.506444	0.001634

3.1. Análisis de las señales completas

Para el análisis de la señal completa, la red fue entrenada con la totalidad de los datos etiquetados con el fin de aprovechar toda la información disponible.

Para analizar la señal frontal se ajustó una red con tres capas ocultas con 30 neuronas por capa y todas con función de activación *sigmoide*. Los resultados obtenidos se muestran en la Figura 3, en donde se integra la señal para visualizar la ubicación temporal de los datos del ECoG y su probabilidad de pertenecer a la clase *estado en crisis*. Los puntos en rojo presentan probabilidades superiores a 0.5, mientras que los puntos en azul corresponden a probabilidades menores a dicho umbral, lo anterior de muestras sin etiqueta. Además los puntos en negro corresponden a las probabilidades de algunos datos etiquetados utilizados en el ajuste del modelo.

En la primera sección de la gráfica que corresponde a los primeros datos etiquetados (puntos negros), se observa cómo las probabilidades se agrupan cerca

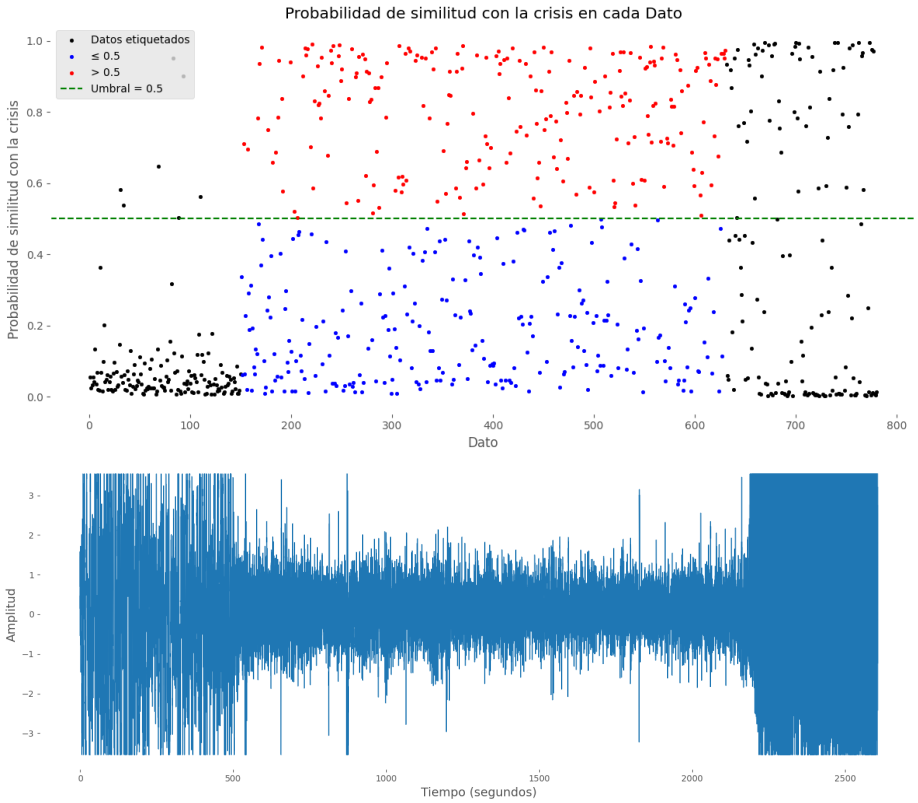

Fig. 3. Señal frontal.

de 0, lo cual indica que la red clasifica correctamente el *estado basal*. En la sección intermedia, correspondiente a los datos 181 al 601, las probabilidades muestran una mayor dispersión, lo cual refleja la transición entre los estados. Finalmente, los datos etiquetados (puntos negros) posteriores a 601, las probabilidades se distribuyen entre 0 y 1, con una predominancia de valores cercanos a 1 sobre los que están cerca de 0; es decir, los datos se clasifican en el *estado en crisis*.

Los resultados obtenidos para la señal parietal se presentan en la Figura 4. En este caso se ajustó una red con tres capas ocultas con 25 neuronas por capa y todas con función de activación *sigmoide*. En los datos del 1 al 180 (puntos negros, que se corresponden con datos etiquetados), las probabilidades se mantienen cercanas a 0, lo que indica una clasificación de los datos en *estado basal*, como se esperaba.

También se observa que la probabilidad de los datos entre 181 y 600 tiene mayor variabilidad; sin embargo, hay más datos con poca probabilidad del *estado en crisis*. Finalmente, a partir del dato 601 (puntos negros correspondientes a datos etiquetados), se nota un incremento en las probabilidades hasta que se

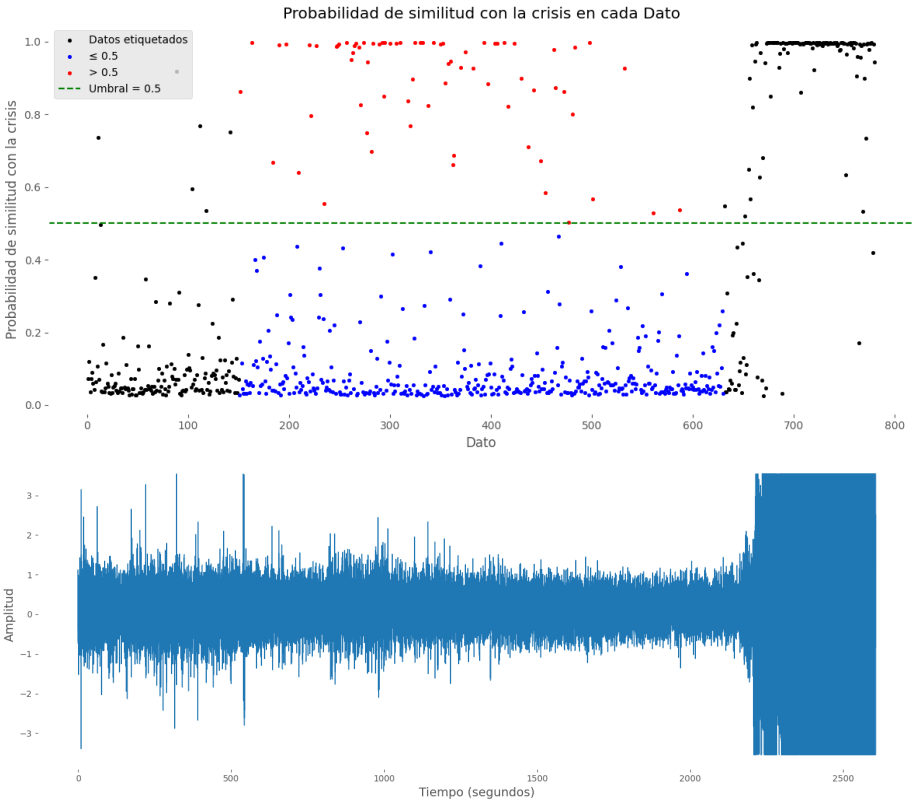

Fig. 4. Señal parietal.

agrupan en torno a 1, correspondiéndose a los datos con *estado en crisis*.

4. Discusión

Las técnicas de aprendizaje estadístico del tipo supervisado permiten ajustar modelos que logran un equilibrio adecuado entre precisión, varianza y flexibilidad. Un aspecto clave en este tipo de modelos es el uso de métodos de optimización, como el algoritmo Adam, que acelera y estabiliza la convergencia hacia un mínimo de la función de pérdida. Como se observa en los resultados obtenidos, las redes neuronales artificiales son eficaces para reconocer los patrones presentes en los ECoG.

Se utilizaron señales de las áreas frontal y parietal del cerebro de una rata, previamente descritas en la sección de metodología. Se sabe que estas señales presentaron diferencias significativas, por lo que tuvieron que implementarse arquitecturas diferentes para su análisis.

En cuanto a los resultados obtenidos, se observó que la señal proveniente de la corteza frontal presentó una mayor dispersión en las probabilidades asignadas por la red neuronal al *estado basal* y *estado en crisis*. Esto podría estar relacionado con la propia actividad motora de las ratas que se encuentran en libre movimiento, o bien con la hiperactividad e hiperconectividad cortical propias de esta región durante las crisis. Por otro lado, estos fenómenos podrían manifestarse de forma más discreta en la corteza parietal, lo cual se reflejó en una señal con menor variabilidad y una mayor proporción de datos clasificados con alta probabilidad de pertenecer al *estado en crisis*. No obstante, es necesario analizar un mayor número de datos provenientes de otros sujetos experimentales para corroborar este hallazgo.

El método computacional propuesto analiza las señales para identificar diferencias entre los dos estados, con el objetivo de distinguir la transición del *estado basal* al *estado en crisis* epiléptica. Su aplicación en un entorno clínico podría contribuir al monitoreo de la estimulación cerebral profunda. Sin embargo, se identificaron algunos desafíos, como la necesidad de recopilar una mayor cantidad de datos para mejorar el entrenamiento de la red y aumentar su fiabilidad, así como la creación de arquitecturas más precisas para disminuir la tasa de error de clasificación. Asimismo, transitar de un modelo animal a uno humano permitiría tener una aplicación directa en la práctica médica.

En la actualidad, el EEG sigue siendo el estándar de oro para el diagnóstico de epilepsia. La posibilidad de contar con diferentes configuraciones, desde la forma no invasiva hasta aquella que, siendo invasiva como el ECoG, permite registrar la actividad eléctrica de regiones subcorticales, lo ha convertido en una herramienta diagnóstica y de seguimiento al tratamiento de gran utilidad. Sin embargo, desde el punto de vista técnico, se requiere de equipamiento especializado, así como de personal capacitado para la realización de los registros y su interpretación. Además, es relevante considerar que existen múltiples grafoelementos en el registro, dependiendo del tipo de epilepsia o síndrome epiléptico en cuestión. Es un método que tampoco está exento de artefactos propios del movimiento, por ejemplo. A lo anterior se suma el hecho de que el manejo de estrategias matemáticas, como la presentada en este trabajo, para el análisis de datos derivados del EEG aún no es de dominio general. Por su parte, la presente propuesta requiere alimentarse de un mayor número de datos, pues solo se empleó la señal de un sujeto y un único modelo experimental, lo que implica seguir incorporando diferentes señales epileptiformes que mejoren su capacidad de discriminación. Así, aún existen barreras técnicas que limitan el uso de redes neuronales en la práctica clínica, aunque esta aproximación permite visualizar su futura implementación.

Este estudio muestra el potencial de la inteligencia artificial en la investigación de la epilepsia, ofreciendo una herramienta útil a médicos e investigadores que facilita el análisis de las señales ECoG. Lo anterior podría contribuir a la identificación temprana de crisis epilépticas, mejorando el diagnóstico, tratamiento y calidad de vida de los pacientes.

Referencias

1. Andrade Tepán, E. C. (2013). *Estudio de los principales tipos de redes neuronales y las herramientas para su aplicación.* Universidad Politécnica Salesiana, Ecuador. Recuperado de http://dspace.ups.edu.ec/handle/123456789/4098
2. Arenas, F., Pérez, R., & Vivas, H. (2016). Un modelo de redes neuronales para complementariedad no lineal. *Revista Integración*, *34*(2), 169–185. https://doi.org/10.18273/revint.v34n2-2016005
3. Bailey, K. R., & Mair, R. G. (2007). Effects of frontal cortex lesions on action sequence learning in the rat. *European Journal of Neuroscience*, *25*(9), 2905–2915. https://doi.org/10.1111/j.1460-9568.2007.05492.x
4. Cavalheiro, E. A., Silva, D. F., Turski, W. A., Calderazzo-Filho, L. S., Bortolotto, Z. A., & Turski, L. (1987). The susceptibility of rats to pilocarpine-induced seizures is age-dependent. *Developmental Brain Research*, *37*(1–2), 43–58. https://doi.org/10.1016/0165-3806(87)90227-6
5. Clifford, D. B., Olney, J. W., Maniotis, A., Collins, R. C., & Zorumski, C. F. (1987). The functional anatomy and pathology of lithium-pilocarpine and high-dose pilocarpine seizures. *Neuroscience*, *23*(3), 953–968. https://doi.org/10.1016/0306-4522(87)90171-0
6. Cook, Z., Zhao, C., Murray, L., Kesan, J., Belacel, N., Doesburg, S. M., Medvedev, G., Vakorin, V. A., & Xi, P. (2024). Decoding Brain Age: A Self-Supervised Graph Neural Network Framework for EEG Analysis. *2024 IEEE SENSORS*, 1–4. https://doi.org/10.1109/SENSORS60989.2024.10784646
7. Craik, A., He, Y., & Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: A review. *Journal of Neural Engineering*, *16*(3), 031001. https://doi.org/10.1088/1741-2552/ab0ab5
8. Cui, Y., Liu, J., Luo, Y., He, S., Xia, Y., Zhang, Y., Yao, D., & Guo, D. (2020). Aberrant connectivity during pilocarpine-induced status epilepticus. *International Journal of Neural Systems*, *30*(05), 1950029. https://doi.org/10.1142/S0129065719500291
9. Eraković, V., Zupan, G., Varljen, J., Laginja, J., & Simonić, A. (2000). Lithium plus pilocarpine induced status epilepticus — biochemical changes. *Neuroscience Research*, *36*(2), 157–166. https://doi.org/10.1016/S0168-0102(99)00120-0
10. Henni, K., Mezghani, N., Mitiche, A., Abou-Abbas, L., & Benazza-Ben Yahia, A. (2025). An effective deep neural network architecture for EEG-based recognition of emotions. *IEEE Access*, *13*(January), 4487–4498. https://doi.org/10.1109/ACCESS.2025.3525996
11. Instituto Nacional de Neurología y Neurocirugía (INNN). (2024). *Día Internacional de la Epilepsia.* Recuperado de https://www.gob.mx/innn/articulos/dia-internacional-de-la-epilepsia-294613
12. Izaurieta, F., & Saavedra, C. (2000). *Redes neuronales artificiales.* Departamento de Física, Universidad de Concepción, Chile.
13. Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* https://arxiv.org/pdf/1412.6980
14. Kolb, B. (1984). Functions of the frontal cortex of the rat: A comparative review. *Brain Research Reviews*, *8*(1), 65–98. https://doi.org/10.1016/0165-0173(84)90018-3
15. López-Meraz, M. L., Rocha, L., Miquel, M., Hernández, M. E., Toledo Cárdenas, R., Coria-Ávila, G. A., García, L. I., Pérez Estudillo, C. A., Aranda Abreu, G. E., & Manzo, J. (2009). Conceptos básicos de la epilepsia. *Revista Médica de La Universidad Veracruzana*, *9*(2), 31–37.

16. Niedermeyer, E., & Lopes da Silva, F. H. (2005). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields* (5ª ed.). Lippincott Williams & Wilkins.
17. Noachtar, S., & Rémi, J. (2009). The role of EEG in epilepsy: A critical review. *Epilepsy & Behavior*, *15*(1), 22–33. https://doi.org/10.1016/j.yebeh.2009.02.035
18. Pal, D., Dean, J. G., Liu, T., Li, D., Watson, C. J., Hudetz, A. G., & Mashour, G. A. (2018). Differential role of prefrontal and parietal cortices in controlling level of consciousness. *Current Biology*, *28*(13), 2145–2152.e5. https://doi.org/10.1016/j.cub.2018.05.025
19. Poddar, K., Sharma, B., Gupta, S., & Angra, S. (2024). Optimizing Convolutional Neural Networks for EEG Classification in Schizophrenia Diagnosis. *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, 1286–1290. https://doi.org/10.1109/ICSES63445.2024.10763226
20. Rajwal, S., & Aggarwal, S. (2023). Convolutional Neural Network-Based EEG Signal Analysis: A Systematic Review. *Archives of Computational Methods in Engineering*, *30*(6), 3585–3615. https://doi.org/10.1007/s11831-023-09920-1
21. Ríos, P. L., & Álvarez, D. C. (2013). Aporte de los distintos métodos electroencefalográficos (eeg) al diagnóstico de las epilepsias. *Revista Médica Clínica Las Condes*, *24*(6), 953–957. https://doi.org/10.1016/S0716-8640(13)70249-9
22. Scorza, F. A., Arida, R. M., Priel, M. R., Calderazzo, L., & Cavalheiro, E. A. (2002). Glucose utilisation during status epilepticus in an epilepsy model induced by pilocarpine: A qualitative study. *Arquivos de Neuro-Psiquiatria*, *60*(2A), 198–203. https://doi.org/10.1590/S0004-282X2002000200003
23. Torrealba, F., & Valdés, J. L. (2008). The parietal association cortex of the rat. *Biological Research*, *41*(4). https://doi.org/10.4067/S0716-97602008000400002
24. Turski, W. A., Cavalheiro, E. A., Schwarz, M., Czuczwar, S. J., Kleinrok, Z., & Turski, L. (1983). Limbic seizures produced by pilocarpine in rats: Behavioural, electroencephalographic and neuropathological study. *Behavioural Brain Research*, *9*(3), 315–335. https://doi.org/10.1016/0166-4328(83)90136-5
25. Wani, F. M., & Karki, M. V. (2024). EEG signal analysis for epilepsy detection using deep neural networks. *2024 5th International Conference on Circuits, Control, Communication and Computing (I4C)*, 213–218. https://doi.org/10.1109/I4C62240.2024.10748439
26. World Health Organization. (2024). *Epilepsy.* Recuperado de https://www.who.int/es/news-room/fact-sheets/detail/epilepsy
27. Zavala-Tecuapetla, C., & López-Meraz, M. L. (2011). Modelos experimentales de epilepsia en ratas en desarrollo. *Revista ENeurobiología*, *2*(4), 190811.

ISSN 1870-4069

Determinación automática del grado de dominio de una lengua extranjera usando modelos de lenguaje

Carlos Andrés Martínez González, David Pinto, Darnes Vilariño Ayala, Beatriz Beltrán Martínez, Yolanda Moyao Martínez

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

mg223470496@alm.buap.mx,{david.pinto,
darnes.vilarino,beatriz.beltran,
yolanda.moyao}@correo.buap.mx

Resumen. El artículo propone un sistema automatizado para evaluar el dominio del inglés como lengua extranjera utilizando modelos de lenguaje avanzados y técnicas de Procesamiento de Lenguaje Natural (PLN). El sistema integra el modelo GPT-J para la aplicación de pruebas y combina herramientas como Whisper [11] y SpeechRecognition [28] para evaluar respuestas orales, utilizando corpus como LibriSpeech (audio) [20] y el Open American National Corpus (texto) [10]. Las métricas empleadas incluyen F1-score [15], Perplexity [12], ROUGE [16] , BLEU [21] , WER [26] y PER [9] para medir aspectos como comprensión, gramática, vocabulario y pronunciación, alineándose con el Marco Común Europeo de Referencia para las Lenguas (MCER) [6]. Los resultados muestran que GPT-J tiene un rendimiento óptimo en comparación con otros modelos como GPT-Neo y GPT-3, aunque se identifican áreas de mejora en la generación de texto y la precisión fonética. El sistema demuestra ser capaz de evaluar de manera autónoma, reduciendo la necesidad de intervención humana. Como trabajo futuro, se sugiere incorporar interfaces 3D para una interacción más natural.

Palabras Clave: Language models, automated english proficiency assessment, GPT-J, natural language processing, automated evaluation.

Automatic Determination of the Degree of Mastery of a Foreign Language Using Language Models

Abstract. This article proposes an automated system for assessing English as a foreign language (EL) proficiency using advanced language models and Natural Language Processing (NLP) techniques. The system integrates the GPT-J model for testing and combines tools such as Whisper [11] and SpeechRecognition [28] to evaluate oral responses, using corpora such as LibriSpeech (audio) [20] and the Open American National Corpus (text) [10]. Metrics used include F1-score [15], Perplexity [12], ROUGE [16], BLEU [21], WER [26], and PER [9] to measure comprehension, grammar, vocabulary, and pronunciation, aligning with the Common European Framework of Reference for Languages (CEFR) [6]. The results show that GPT-J performs optimally compared to other models such as GPT-Neo and GPT-3, although areas for improvement are identified in text

generation and phonetic accuracy. The system proves capable of evaluating autonomously, reducing the need for human intervention. As future work, we suggest incorporating 3D interfaces for more natural interaction.

Keywords: Language models, automated English proficiency assessment, GPT- J, natural language processing, automated evaluation.

1. Introducción

Con el avance diario de la tecnología aplicada al campo computacional, se ha revolucionado la calidad de vida del ser humano, especialmente en áreas como la educación y la comunicación. En este contexto, los modelos de lenguaje han demostrado ser herramientas fundamentales para evaluar y mejorar la competencia lingüística en diversos idiomas. Este trabajo de investigación se centra en generar un modelo de aprendizaje automático que permita determinar de manera precisa el grado de dominio lingüístico en el idioma inglés como lengua extranjera. Se propone utilizar modelos de lenguaje basados en Procesamiento de Lenguaje Natural (PLN), para evaluar aspectos clave del dominio del idioma inglés, tomando en cuenta que el hablante no es nativo. El sistema evalúa competencias en comprensión, expresión oral y escrita, gramática, vocabulario y otros factores relevantes que determinan el nivel de competencia en inglés. Para ello, se utiliza un modelo GPT-J encargado de aplicar las pruebas, en conjunto con un submodelo interno que determina el nivel de dominio del idioma conforme a los parámetros establecidos por el Marco Común Europeo de Referencia para las Lenguas (MCER) [6].

Por lo tanto, el sistema se implementa en una plataforma web, permitiendo la interacción fluida entre el usuario y el sistema de evaluación del grado de dominio del idioma inglés. Esta interfaz utiliza tanto texto como audio, permitiendo medir las competencias lingüísticas de los usuarios, lo cual es clave para el aprendizaje efectivo del idioma, obteniendo datos reales para refinar el algoritmo basado en los resultados obtenidos. Estas pruebas permiten ajustar el modelo de acuerdo con las necesidades de los usuarios y asegurar una tasa de precisión adecuada para su implementación a mayor escala.

A fin de garantizar una evaluación precisa y confiable, se emplean métricas como F1-score [15], Perplexity [12], ROUGE [16], BLEU [21], WER [26] y PER [9]; además de un corpus representativo de inglés estadounidense proveniente del Open National American Corpus [10] y LibriSpeech [20], con el propósito de seleccionar los datos más adecuados para el entrenamiento y validación del modelo propuesto.

Una de las principales aportaciones de este trabajo es la propuesta de utilizar modelos de lenguaje de gran escala, como GPT-J, no únicamente como generadores de texto, sino como aplicadores automatizados de evaluaciones del idioma inglés. Este enfoque permite explorar nuevas aplicaciones educativas de los modelos de lenguaje, diferenciándose de los usos convencionales centrados en la generación libre de contenidos.

Además, este estudio no solo busca desarrollar un sistema de evaluación automatizada, sino también contribuir al campo del procesamiento del lenguaje natural (PLN) mediante una metodología que permita evaluar de manera efectiva la adquisición

de una segunda lengua mediante modelos avanzados de Inteligencia Artificial (IA). A continuación, se presentan los trabajos relacionados con el tema de investigación.

2. Trabajos relacionados

Dentro del estudio de la IA, diversas investigaciones han hecho hincapié que su aplicación no únicamente se restrinja al desarrollo del dominio de una lengua extranjera; sino también en entornos virtuales donde los modelos de lenguaje puede ser una herramienta adicional al progreso humano, creando así diversas experiencias inmersivas donde la realidad da un paso más allá de la convivencia humana - computacional.

Con relación a la influencia de los modelos de lenguaje para determinar el grado de dominio de un idioma, existe una investigación que destaca cómo los modelos de lenguaje están revolucionando el aprendizaje del inglés al facilitar la creación de chatbots conversacionales que interactúan con los estudiantes mediante respuestas coherentes, mejorando así la práctica del idioma. Además, estos modelos permiten adaptar el contenido educativo a las necesidades individuales de cada estudiante, ofreciendo una experiencia de aprendizaje personalizada [3].

En un artículo de la Universidad de Florida [4] sobre aplicaciones de la IA, se exploran diversas formas en que la IA se integra en modelos virtuales para evaluar y mejorar la enseñanza hacia los estudiantes. Se concluye que estos modelos son herramientas de apoyo docente efectivas, basadas en más de 20 horas de pruebas de rendimiento y funcionalidad.

Este mismo tema fue abordado por Villaroel [7], quien señala cómo el aula tradicional ha evolucionado con la inclusión de tecnologías como la IA y técnicas de análisis de información como Big Data. Estas herramientas permiten personalizar las rutas de aprendizaje de los estudiantes y optimizar la gestión educativa. Además, la importancia de las plataformas colaborativas como GitHub agilizan tareas rutinarias y ofrecen propuestas predictivas, así como la integración de la gamificación con IA, la cual mejora las habilidades cognitivas y fomenta un aprendizaje significativo y actualizado con relación a las necesidades de la actualidad.

Por otra parte, en un artículo de la Universidad del Sur de Florida [24], se explora la combinación de la Realidad Aumentada (AR) con Voicebots y modelos de lenguaje como ChatGPT para la enseñanza de lenguas extranjeras a niños pequeños a través de un entorno agradable e interactivo. Además, el uso de la AR facilita una experiencia visual interactiva que atrae la atención de los niños, mientras que los Voicebots y modelos de lenguaje como ChatGPT permiten conversaciones personalizadas que refuerzan la inmersión lingüística. Esta combinación, ayuda a que el aprendizaje de una lengua extranjera sea más efectivo y entretenido para los niños, permitiendo una mayor retención de conocimiento a través de la interacción constante.

Otra forma de evaluar el dominio de un idioma se aborda en la investigación de Muñoz y Fuertes [19], que explora cómo la IA puede ser aplicada en el ámbito educativo; en particular, destacan que la interacción entre humanos y máquinas puede ser una herramienta valiosa en el aprendizaje de una segunda lengua. Esto incluye aspectos como el aprendizaje informal, la autonomía del estudiante y la auto-evaluación.

De acuerdo con el trabajo realizado en conjunto por Guano y otros investigadores [8] sobre el modelo de aprendizaje del idioma inglés utilizando algoritmos de machine learning, se analiza cómo estos algoritmos abarcan diversas áreas, como las ciencias psicológicas, sociales, lingüísticas y pedagógicas. Aunque estos algoritmos pueden ser herramientas valiosas para personalizar el aprendizaje, evaluar habilidades y optimizar recursos educativos, también presentan riesgos, especialmente en la evaluación del dominio de una lengua extranjera. Entre estos riesgos se incluyen problemas de precisión en la evaluación, sesgo en los modelos y dependencia excesiva de la tecnología, lo que puede afectar tanto la calidad del aprendizaje como la equidad en la evaluación.

Dentro de los parámetros de evaluación sobre modelos de lenguaje, se encuentra un artículo que examina en profundidad los aspectos clave de la evaluación, centrándose en las preguntas: ¿qué evaluar?, ¿cómo evaluar? y ¿dónde evaluar? [1]. El artículo utiliza diversos estudios de rendimiento, protocolos y tareas para analizar no solo los avances realizados en los modelos de lenguaje, sino también para identificar áreas clave de mejora, estos se centran en métricas clave como Perplexity y BLEU [21], señalando áreas específicas para mejorar la precisión y efectividad de estos modelos. Además, destaca la importancia de la evaluación en términos de aplicabilidad, ética y equidad, subrayando los desafíos que enfrentan estos modelos en cuanto a sesgo, generalización y manejo de tareas complejas.

En el trabajo desarrollado por Kasneci [13], los autores exploran cómo los modelos de lenguaje grandes pueden ser utilizados de manera positiva en el ámbito educativo; destacan que, además de determinar el grado de dominio de una lengua extranjera, estos modelos pueden apoyar diversos aspectos de la educación en distintas materias. Sin embargo, también abordan desafíos asociados con su inclusión, como la necesidad de proporcionar retroalimentación instantánea y facilitar el acceso a recursos educativos, así como problemas relacionados con la precisión de las respuestas, por lo cual el uso de IA en educación plantea preocupaciones sobre la dependencia excesiva y la privacidad.

En un proyecto desarrollado sobre la corrección de errores en sistemas de reconocimiento del habla (ASR por sus siglas en inglés) [17], se puede mejorar significativamente los resultados del reconocimiento de voz, especialmente cuando se utilizan modelos generativos como ChatGPT en configuraciones zero-shot o few-shot. Este enfoque, basado en las salidas N-best, ha mostrado ser eficaz incluso con arquitecturas avanzadas como transducers y modelos con atención. Modelos como T5 (Text-to-Text Transfer Transformer), que convierten cualquier tarea de procesamiento de lenguaje en un problema de generación de texto, se destacan por su versatilidad, haciendo que sean efectivos no solo para la corrección de errores ASR, sino también en diversas aplicaciones de PLN.

En el artículo de Millière [18] se exploran las implicaciones de los modelos de lenguaje modernos, como los de la familia GPT, para la lingüística teórica. A pesar de que estos modelos están diseñados con objetivos de ingeniería, su capacidad para adquirir conocimiento lingüístico complejo a partir de grandes cantidades de datos plantea la necesidad de reconsiderar su relevancia en el ámbito de la teoría lingüística. Esta información presenta evidencia empírica que sugiere que los modelos de lenguaje pueden aprender estructuras sintácticas jerárquicas y responder a fenómenos lingüísticos, aportando una colaboración más cercana entre lingüistas y científicos

computacionales podría ofrecer valiosas perspectivas, especialmente en debates sobre el nativismo lingüístico.

En otra investigación donde se propone determinar si los modelos de lenguaje extensos (LLMs por sus siglas en inglés) pueden reemplazar evaluaciones humanas [2], realizaron experimentos en tareas de clasificación de texto, calificación automática de ensayos y análisis de contenido. Como resultado, los LLMs demostraron ser consistentes y eficientes, sin embargo, aún necesitan supervisión humana para tareas complejas; lo que puede ser empleado en evaluaciones iniciales o como complemento en entornos educativos.

En un proyecto propuesto en el área de estudio [25], se desarrolló un marco de evaluación para modelos grandes de lenguaje aplicados al audio (LLM-A por sus siglas en inglés), a través de proponer un conjunto de tareas de evaluación en diferentes modalidades: clasificación, generación, transcripción y síntesis. Incluye datasets preexistentes y adaptados para estas tareas. Este proyecto identificó las fortalezas y limitaciones de varios LLM-A existentes, destacando la necesidad de mejorar la coherencia en la generación y la calidad de la transcripción. Lo que puede favorecer el entender cómo integrar modelos de lenguaje para dominios específicos como audio, ofreciendo ideas reveladoras sobre posibles extensiones para el modelo de lenguaje aplicado al inglés.

Recientemente, se creó el proyecto AudioPaLM [22], en el cual se desarrolló un modelo de lenguaje multimodal que entiende y genera texto y audio con alta calidad. Este modelo de lenguaje combina métodos de preentrenamiento textual y de audio en un único modelo, utilizando datasets de alta calidad en ambas modalidades, también incluye aprendizaje de transferencia para tareas específicas como síntesis y traducción multimodal. Sus resultados mostraron rendimiento competitivo en tareas de síntesis y traducción, que igualan o superan a modelos especializados en audio.

Otro proyecto relevante en el ámbito de la evaluación auditiva por parte de modelos de lenguaje es SpeechVerse [5], el cual acorde a los autores, su objetivo es diseñar un modelo escalable para tareas de lenguaje y audio con énfasis en generalización. Para ello, el modelo fue entrenado en un corpus masivo, que incluye múltiples lenguajes y modalidades, también se usaron técnicas como Fine-Tuning progresivo y enmascaramiento predictivo. Este modelo de lenguaje especializado en audio destacó en tareas de generación y comprensión en diversos lenguajes, superando modelos previos en métricas como BLEU y F1, lo que indica que su enfoque puede ser relevante para evaluar cómo los modelos de lenguaje podrían manejar idiomas y sus variantes, especialmente entre dialectos o regiones.

En la investigación desarrollada por Yang [27], se presentó AIR-Bench, una plataforma de evaluación diseñada específicamente para medir la comprensión generativa en modelos de audio, este sistema introduce tareas como la comprensión narrativa del audio y la generación de respuestas, utilizando datasets anotados manualmente. Los resultados evidenciaron que los modelos actuales son efectivos en tareas de comprensión básica, pero enfrentan desafíos en escenarios más complejos o especializados. Además, en futuras aplicaciones, AIR-Bench podría ofrecer métricas y tareas adaptables para evaluar componentes específicos de modelos de lenguaje aplicados al inglés, especialmente en contextos de evaluación avanzada. A continuación, se presenta la metodología propuesta en la presente investigación.

3. Diseño de la investigación

Para llevar a cabo la determinación del grado de dominio de una lengua extranjera usando modelos de lenguaje, se utilizó una metodología propuesta para ejecutar con éxito su desarrollo e implementación. Tal metodología se concentra en los puntos siguientes:

- Realizar un estudio de las metodologías propuestas hasta el momento, revisando que se ha hecho, cómo se ha hecho y qué resultados se tienen.
- Comparar los modelos de lenguaje para determinar cuál se utilizará en este proyecto de investigación.
- Investigar la tecnología necesaria para llevar a cabo el desarrollo del proyecto.
- Obtener un corpus para obtener el modelo de lenguaje a usar en el proceso de determinación del grado de dominio.
- Entrenar el modelo de lenguaje usando GPUs.
- Realizar las pruebas para determinar la eficiencia del modelo del lenguaje utilizando métricas adecuadas, por ejemplo, la medida F1 para la precisión del modelo o BLEU entre otros para la precisión de traducción de texto.
- Realizar los ajustes necesarios para mejora del modelo.
- Obtención del modelo ajustado.
- Evaluación del modelo generado.

3.1. Técnicas de PLN: Medidas empleadas

Divergencia de Kullback-Leibler

La divergencia de Kullback-Leibler [14] mide la diferencia entre dos distribuciones de probabilidad: distribución real o esperada y distribución aproximada o predicción.

Es fundamental en tareas que comparan distribuciones, como el análisis de texto; dentro del flujo del programa, esta divergencia evalúa la similitud del vocabulario del estudiante en relación con el de los hablantes nativos.

Similitud de coseno

Ésta medida se basa en la comparación de la dirección de los términos en un espacio matemático, sin importar su tamaño o frecuencia absoluta; en concreto, se calcula la similitud entre dos vectores basándose en el ángulo entre ellos, lo cual es ampliamente usado para comparar documentos o frases en términos de similitud semántica [23].

Distribución de Zipf

En la distribución de Zipf [29], se empleó la medida para conocer el flujo de palabras y su aparición en el texto comparado debido a que es un principio estadístico que describe cómo ciertas palabras en un idioma aparecen con mayor frecuencia que otras,

siguiendo un patrón predecible; lo que facilita comprender la distribución léxica en un corpus. Con relación a las técnicas especializadas para el desarrollo del proyecto, se comprende que existen métricas de evaluación del rendimiento del modelo de lenguaje, a continuación, se describen las métricas que fueron empleadas.

Métrica F1 Score

Esta métrica es adecuada para el proyecto [15], ya que representa el promedio armónico entre la precisión (proporción de predicciones correctas sobre todas las predicciones positivas) y la exhaustividad (proporción de aciertos sobre los datos positivos reales). Su uso es particularmente útil en escenarios donde existe un desbalance entre las clases, permitiendo evaluar el rendimiento del modelo de manera equilibrada.

Métrica Perplexity

Esta medida [12], es utilizada para cuantificar la incertidumbre asociada con una distribución de probabilidad, en los campos de la estadística, análisis de los datos y la ciencia de datos; la entropía cruzada calcula la incertidumbre promedio en las predicciones del modelo. Valores bajos de perplexity indican un mejor ajuste del modelo al texto, lo que permite medir la calidad de generación del modelo y la habilidad de construcción de frases en inglés.

Métrica ROUGE

Esta métrica permite evaluar textos no necesariamente grandes [16], comparado con la métrica BLEU, lo que permite una mayor flexibilidad en la calidad de evaluación si se visualiza desde el ámbito de la conversación y el análisis de frases cortas. En particular, se compara n-gramas generados por el modelo con los del texto de referencia, evaluando la similitud; lo cual es ampliamente usado en tareas de resumen y generación de texto, midiendo precisión, recall y F1 a través de diferentes niveles de análisis.

Métrica BLEU

La métrica BLEU (Bilingual Evaluation Understudy) [21] es un método cuantitativo ampliamente utilizado para evaluar la calidad de traducciones automáticas y sistemas de generación de lenguaje natural. Esta métrica combina dos componentes principales: la precisión modificada de n-gramas (que evita premiar repeticiones excesivas) y un factor de penalización por brevedad (para castigar traducciones demasiado cortas). El resultado, es un puntaje normalizado entre 0 y 1, donde valores cercanos a 1 indican mayor concordancia con las referencias humanas.

Métrica WER

El Word Error Rate (WER), o Tasa de Error de Palabras [26], es una métrica ampliamente utilizada en la evaluación de sistemas de reconocimiento de voz, esta

medida cuantifica la precisión de un sistema al comparar la transcripción generada automáticamente con una transcripción de referencia considerada como correcta.

Un WER más bajo indica un mayor nivel de precisión en el reconocimiento de palabras, siendo esta métrica fundamental para evaluar y comparar el rendimiento de sistemas de reconocimiento de voz.

Métrica PER

El Phoneme Error Rate (PER), o Tasa de Error de Fonemas, es una métrica que evalúa la precisión en el reconocimiento de unidades fonéticas [9]. A diferencia del WER, el PER se centra en la comparación entre la transcripción fonética generada por un sistema y una transcripción fonética de referencia. Esta métrica es particularmente útil en tareas relacionadas con el análisis de la pronunciación y la evaluación de sistemas de reconocimiento del habla a nivel fonético; por lo tanto, la métrica PER es una herramienta valiosa para analizar la capacidad de un sistema de reconocer y transcribir correctamente los sonidos del habla, lo que resulta esencial en aplicaciones como la corrección de la pronunciación o la síntesis de voz.

Toda esta información fue de utilidad para la construcción del modelo de lenguaje final. En el siguiente apartado se describen los resultados obtenidos utilizando todos los componentes antes descritos, así como los conjuntos de datos que fueron empleados para las pruebas de eficiencia.

4. Evaluación de resultados

Se utilizaron volúmenes de datos provenientes del Open National American Corpus y de LibriSpeech para hacer una correcta comparación de datos de referencia en contraste con la información obtenida del usuario en la prueba y verificar aspectos como la ortografía, la pronunciación, entre otros factores, con el fin de catalogarlos correctamente; a continuación, se describe a detalle cada corpus utilizado.

4.1. Conjunto de datos

El primer corpus de información proveniente de LibriSpeech se enfoca en los fonemas obtenidos al convertir audio a texto; para ello, se utilizaron herramientas avanzadas de captura de audio, las cuales permitieron comparar y verificar si el usuario pronunció correctamente en inglés, éste corpus se compuso de 100 horas de audio libre de ruido en temas de uso cotidiano.

El segundo corpus, por su parte, se centró en la comprensión de textos, así como en la ortografía, la gramática y el vocabulario. Para este análisis, se tomaron 7 GB de ejemplos del *Open National American Corpus* sobre diversos temas desde el aspecto técnico hasta de tipo cotidiano, con el objetivo de evaluar las habilidades de escritura en inglés. Como resultado, se pudo proporcionar retroalimentación al usuario sobre su nivel de dominio del idioma. A continuación, se describe en detalle la información obtenida a partir de los corpus disponibles y utilizados en este proceso, así como las tecnologías empleadas y las métricas aplicadas.

Corpus de audio

El desarrollo de modelos de lenguaje aplicados a la evaluación del dominio del inglés requiere fuentes de datos adecuadas tanto en texto como en audio. Los corpus de texto permiten evaluar aspectos como gramática, vocabulario y comprensión lectora, mientras que los corpus de audio son esenciales para el análisis de pronunciación. Además, el uso de bibliotecas especializadas en conversión de audio a texto facilita la integración de estos recursos en modelos de evaluación automatizados. Dentro de la investigación sobre la evaluación de la pronunciación del idioma inglés estadounidense, se destacan los siguientes corpus disponibles.

LibriSpeech

Este corpus se encuentra basado en audiolibros del Proyecto Gutenberg [20], el cual se caracteriza por los siguientes aspectos.

- Alta calidad de audio: Donde se encuentran grabaciones bien pronunciadas y sin ruido de fondo.
- Transcripciones alineadas: Lo que permite comparar la pronunciación real con la esperada.
- Segmentación en conjuntos "clean" y "other": El cual permite realizar pruebas en distintos niveles de claridad y dificultad.
- Este recurso es particularmente útil para evaluar pronunciación en escenarios de lectura estructurada, complementando otras bases de datos que incluyen habla más espontánea. En otros aspectos, para integrar los corpus de audio en el modelo de evaluación, es necesario emplear bibliotecas que conviertan la entrada de voz en texto. Algunas de las herramientas más relevantes en este proceso podrían ser las siguientes:

Whisper (OpenAI)

En este modelo de reconocimiento de voz de código abierto [11] ofrece una alta precisión en la transcripción de audio a texto, asimismo la capacidad de reconocer múltiples acentos y dialectos del inglés; y, por último, procesamiento de ruido ambiental y habla espontánea. La aplicación de esta tecnología podría ayudar en la evaluación del idioma al obtener transcripciones detalladas para comparar la pronunciación del hablante con la referencia estándar.

SpeechRecognition (Python Library)

Esta biblioteca es una interfaz para diversas APIs de reconocimiento de voz [28] y se destaca por soportar múltiples motores como Google Speech-to-Text y Sphinx. Del mismo modo es muy accesible para integrar con Python al procesar archivos de audio en tiempo real. En este caso se puede emplear para realizar pruebas iniciales de reconocimiento de voz y comparación con modelos más avanzados como Whisper.

Fig. 1. Funcionamiento algorítmico de prueba.

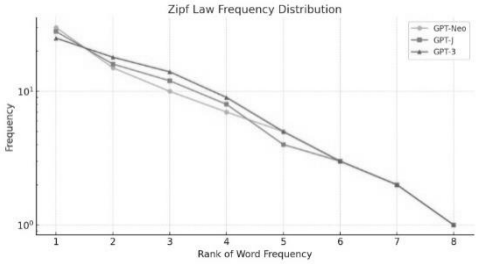

Fig. 2. Gráfica de comparación de modelos GPT (Ley Zipf).

Corpus de texto

En el ámbito de la evaluación del dominio del inglés, los corpus textuales constituyen una herramienta fundamental para analizar aspectos clave como la precisión gramatical, la riqueza léxica y la capacidad de la comprensión lectora. Estos recursos permiten examinar el uso del lenguaje en contextos reales, lo que resulta esencial para diseñar metodologías de evaluación efectivas. A continuación, se describe la fuente utilizada en este estudio.

Open National American Corpus

El Open American National Corpus (OANC) [10] es un corpus de acceso abierto diseñado para representar el inglés estadounidense en su diversidad de registros y contextos. Desarrollado como parte de un proyecto colaborativo, este recurso se destaca por su enfoque en la representatividad lingüística y su marcación detallada, del cual sus principales aportes se encuentran una amplia variedad de textos que abarcan desde documentos científicos hasta conversaciones informales, lo que permite evaluar el lenguaje en distintos niveles de formalidad. Del mismo modo también existe una marcación gramatical y semántica, facilitando el análisis estructural y funcional del lenguaje. Este corpus es especialmente valioso para evaluar la competencia gramatical en contextos diversos, así como para analizar la adaptabilidad del usuario a diferentes registros lingüísticos. Estas fuentes complementan los corpus estructurados al ofrecer material actualizado y representativo de la lengua en uso, lo que enriquece la evaluación de la competencia lectora, auditiva y la capacidad de interactuar con textos auténticos.

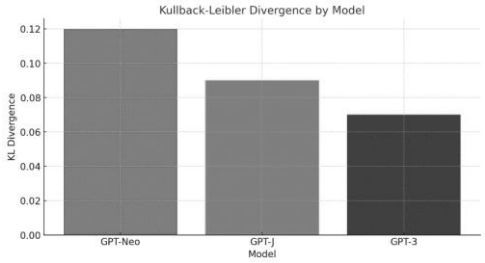

Fig. 3. Gráfica de comparación de modelos GPT (Kullback-Leibler).

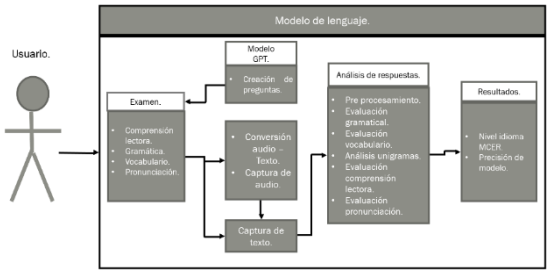

Fig. 4. Diagrama de casos de uso modelo de lenguaje.

4.2. Resultados obtenidos

Para integrar estas técnicas y determinar cuál modelo puede ser funcional para el desarrollo del proyecto, se diseñó el siguiente diagrama de flujo con el objetivo de proceder a evaluar textos de prueba y poder comparar a detalle el rendimiento de cada modelo, como se ve en la Figura 1.

Por consiguiente, se realizó la evaluación de los modelos GPT-Neo, GPT-3 y GT-J como modelos seleccionados, utilizando las técnicas de análisis PLN antes mencionadas, por lo cual se demostró una eficiencia óptima en el modelo GPT-J y GPT-3 en comparación con el modelo GPT-Neo, en la Figura 2, se visualizan los resultados obtenidos con la distribución de la ley de Zipf.

Del mismo modo, el la Figura 3, se puede comprender el rendimiento de los modelos utilizando la divergencia de Kullback-Leibler, donde acorde a las especificaciones de su uso, el rendimiento de GPT-J se encuentra en un rendimiento mayor frente a los resultados arrojados por el modelo GPT-Neo, y, por lo tanto, muy cercano al rendimiento óptimo del modelo GPT-3.

En resumen, los experimentos realizados permitieron comparar el rendimiento del modelo basado en GPT-J con otros modelos de lenguaje, incluyendo GPT-Neo y versiones pre entrenadas de GPT-3. En términos de Perplejidad, GPT-J mostró una mejora notable frente a GPT-Neo, indicando una mayor capacidad de generación de texto coherente. Sin embargo, en la comparación con GPT-3, el modelo presentó un rendimiento inferior en tareas de generación de texto específicas.

Comprendiendo la información obtenida sobre la comparación de modelos de lenguaje; GPT-J fue seleccionado para utilizarse en conjunto con el modelo de lenguaje

Tabla 1. Resultados obtenidos de las pruebas de texto.

#	Corrección Gramatical	ROUGE	Comprensión Lectora	BLEU	F1 Score	Perplexity
1	my mother was born on 1935	0.0042	0.0	**0.0**	0.0174	6.0
2	how are you?	0.0001	0.0	**0.0**	0.0005	4.0
3	my favorite animas is cats, dogs and scorpions	0.0055	0.0	2.05e-282	0.0128	9.0
4	my name is andy and i am an english teacher and computer systems engineer	0.0084	0.0	9.79e-273	0.0079	14.0
5	oh , really? think you do not	0.0068	0.0	3.19e-273	0.0190	7.0
6	the sun rises in the east	0.0032	0.0	1.45e-200	0.0152	5.0
7	what time is it now?	0.0015	0.0	**0.0**	0.0038	3.5
8	i love programming with python	0.0091	0.0	4.75e-180	0.0215	11.0
9	artificial intelligence is the future	0.0078	0.0	5.89e-150	0.0107	8.0
10	today is a beautiful day	0.0027	0.0	7.12e-170	0.0143	6.5

que evaluará el grado de dominio del idioma haciendo énfasis en el idioma inglés, por otra parte se puede hacer uso de las técnicas del análisis PLN como la divergencia de Kullback-Leibler, la similitud de coseno y distribución de Zipf para el análisis de respuestas y complementarlo con las herramientas de NLTK para la evaluación gramatical, vocabulario y comprensión lectora. Por otro lado, las herramientas como Whisper y SpeechRecognition pueden emplearse para la evaluación de la pronunciación.

Si a todo esto se le hace adición de métricas de evaluación de idioma como Perplexity, ROUGE y BLEU para el aspecto del texto. WER Y PER para medir la calidad de pronunciación comprendida en apoyo de los corpus como LibriSpeech y el Open National American Corpus; se puede determinar un flujo de trabajo del funcionamiento de todo el sistema, como se observa en la Figura 4.

Se puede apreciar que el modelo GPT genera las pruebas de evaluación del idioma, mientras que el usuario responde según el área de pregunta que presenta, el modelo de lenguaje creado se encarga de analizar y evaluar cada respuesta a través de diversos módulos especificados en la sección “Análisis de respuestas” utilizando todas las herramientas anteriormente mencionadas. Como consecuencia, los resultados se pueden interpretar en términos de nivel de idioma acorde a los parámetros establecidos por el Marco Común Europeo de Referencia para las Lenguas, asimismo, la información arrojada por las métricas computacionales, demuestran que el modelo de lenguaje es capaz de evaluar el grado de dominio del idioma y al mismo tiempo se mide la eficiencia del modelo de lenguaje construido, en la Tabla 1 se puede comprender a profundidad los resultados de texto obtenidos de las pruebas de nivel del idioma.

La evaluación mediante ROUGE y otras métricas reveló que el modelo basado en GPT obtiene una puntuación comparable a modelos comerciales en términos de

Tabla 2. Resultados obtenidos de las pruebas de pronunciación.

#	Texto Transcrito	WER	PER	Evaluación Pronunciación
1	The cat is on the table	0.12	0.08	Buena pronunciación
2	She want go to school	0.22	0.15	Errores en conjugación
3	I am happy today	0.10	0.05	Pronunciación clara
4	We going to the park	0.18	0.12	Problema con la gramática
5	This is a apple	0.25	0.18	Error en artículo
6	My brother play soccer	0.20	0.14	Error en tiempo verbal
7	They are studying English	0.08	0.04	Excelente pronunciación
8	The book on the table	0.30	0.22	Omisión de verbo
9	He is doctor	0.15	0.10	Falta de artículo
10	We have a meeting tomorrow	0.05	0.03	Pronunciación fluida

similitud textual, validando su uso en la evaluación del dominio del idioma inglés. En contraste, la divergencia de Kullback-Leibler indica que el modelo genera respuestas con una distribución de probabilidad distinta a la de hablantes nativos, sugiriendo la necesidad de ajustes adicionales en el entrenamiento del corpus.

Por otro lado, los análisis de similitud del coseno reflejaron una correlación moderada entre las respuestas generadas y los textos de referencia, lo que indica que el modelo capta estructuras lingüísticas clave, aunque con ciertas limitaciones en la comprensión semántica. Finalmente, la evaluación basada en la Ley de Zipf permitió identificar desviaciones en la frecuencia de uso de términos clave, lo que sugiere la necesidad de integrar corpus más representativos del inglés contemporáneo para mejorar la naturalidad del lenguaje generado. En cuanto a las pruebas de pronunciación, los resultados describen diversos valores en las métricas en cuanto a fonética comparado con las transcripciones del corpus antes mencionado, en la Tabla 2 se describen los resultados de pronunciación obtenidos.

En las pruebas de pronunciación, la evaluación basada en la métrica WER (Word Error Rate) y PER (Phoneme Error Rate) mostró que la transcripción automática de los audios presenta variaciones en la precisión, dependiendo de la claridad del hablante y la complejidad de las frases evaluadas. Se observó que las frases con estructuras gramaticales más simples obtuvieron menores tasas de error, mientras que aquellas con mayor longitud o con pronunciaciones poco convencionales generaron un mayor margen de desviación. Además, la comparación con el corpus de referencia reveló que el modelo mantiene una correlación aceptable en la identificación de fonemas comunes en inglés, aunque con errores recurrentes en palabras de pronunciación ambigua. Esto sugiere que el desempeño del sistema podría beneficiarse de ajustes en el preprocesamiento de audio y la incorporación de técnicas de alineación fonética más robustas.

5. Conclusiones y recomendaciones

En esta investigación se desarrolló un modelo de evaluación del dominio del idioma inglés basado en modelos de lenguaje, integrando diferentes técnicas de PLN y métricas de evaluación; para ello, se compararon distintos modelos GPT, incluyendo GPT-Neo,

GPT-J y GPT-3, determinando que GPT-J es el más adecuado para aplicar el examen, mientras el modelo MAENI (Modelo Automático de Evaluación del Nivel de Inglés) se encarga de evaluar las respuestas del usuario.

Para la evaluación de la pronunciación, se emplearon herramientas avanzadas como Whisper y SpeechRecognition en adición con el corpus de LibriSpeech, lo que permitió comparar las transcripciones generadas con referencias fonéticas estándar. Por otra parte, en el análisis de textos, se implementó un preprocesamiento con NLTK, complementado con técnicas de corrección ortográfica y evaluación del vocabulario. La comprensión lectora se midió a través de la información proporcionada por el Open American National Corpus, asegurando un análisis detallado de la semántica y la estructura del lenguaje escrito.

En cuanto al desempeño del sistema, se implementaron métricas clave como F1-score, Perplexity, ROUGE y BLEU para la comprensión textual, mientras que en el análisis de pronunciación se aplicaron WER y PER. Como consecuencia, los resultados indicaron que el modelo es capaz de operar de manera autónoma, minimizando la necesidad de intervención humana en el proceso de evaluación, además de añadir la posibilidad de adaptarse a otros idiomas.

Como trabajo futuro, aunque el modelo MAENI presenta una primera versión funcional capaz de aplicar evaluaciones del dominio del idioma inglés mediante GPT-J y métricas cuantificables, se identifican oportunidades de mejora para su evolución futura:

- **Ampliación de corpus de entrenamiento:** Se contempla integrar corpus más representativos de inglés conversacional y cotidiano, como Common Voice de Mozilla, a fin de evitar la sobre corrección de expresiones legítimas no estándar.
- **Reconocimiento de variantes dialectales:** Se prevé la adaptación del sistema a diferentes variantes del inglés (británico, australiano, etc.) mediante la inclusión de corpus específicos y ajustes de evaluación semántica.
- **Implementación de verificación semántica:** Se proyecta el desarrollo de un módulo de verificación semántica basado en coincidencia de palabras y un índice de sinónimos para flexibilizar la evaluación de respuestas correctas alternativas.
- **Módulo de auto entrenamiento supervisado:** Se planea utilizar datos del propio corpus para mejorar el rendimiento del modelo de forma incremental, optimizando la gestión de recursos textuales y reduciendo redundancia.
- **Privacidad y consideraciones éticas:** Futuras fases incluirán mecanismos para la protección de datos de voz y texto de los usuarios, cumpliendo normativas de privacidad como GDPR o LOPD.
- **Infraestructura tecnológica:** La implementación a mayor escala requeriría servidores especializados capaces de gestionar reconocimiento de voz, almacenamiento de corpus, y procesos de evaluación simultáneos.
- **Entorno inmersivo basado en avatares 3D:** Se prevé el desarrollo de una interfaz de interacción mediante avatares 3D, utilizando motores gráficos como Unity, para favorecer una simulación de conversación más natural a través de WebRequests conectados al sistema de evaluación.

Estas implementaciones abrirían nuevas posibilidades en la evaluación automatizada del dominio del inglés, combinando PLN con interfaces gráficas avanzadas para un sistema más intuitivo y eficiente.

Referencias

1. Chang, Y.: A Survey on Evaluation of Large Language Models. In: ACM Transactions on Intelligent Systems and Technology, 15(3), pp. 1–45 doi: 10.1145/3641289.
2. Chiang, C.H., Lee, H.Y.: Can Large Language Models Be an Alternative to Human Evaluations? doi: 10.48550/arXiv.2305.01937.
3. Chicaiza, R.M., Camacho, L.A., Ghose, G.: Aplicaciones de Chat GPT como inteligencia artificial para el aprendizaje de idioma inglés: avances, desafíos y perspectivas futuras. LATAM: Revista Latinoamericana de Ciencias Sociales y Humanidades, 4(2), pp. 2610–2628 (2023) doi: 10.56712/latam.v4i2.781.
4. Dai, C.P.: NSF Public Access Repository. NSF Public Access Repository: https://par.nsf.gov/servlets/purl/10343548.
5. Das, N.: SpeechVerse: A Large-scale Generalizable Audio Language Model (2024) doi: 10.48550/arXiv.2405.08295.
6. Council of Europe.: Common European Framework of Reference for Languages: Learning, Teaching, Assessment. Companion Volume. Strasbourg: Council of Europe Publishing (2020)
7. García Villarroel, J.J.: Implicancia de la inteligencia artificial en las aulas virtuales para la educación superior. Orbis Tertius UPAL, 5(10), pp. 31–52 (2021) doi: 10.59748/ot.v5i10.98.
8. Guano Merino, D.F., Herrera Andrade, Z.V., Vallejo Barreno, C.F.: Modelo de aprendizaje del idioma inglés utilizando algoritmos de machine learning. Explorador Digital, 7(1), pp. 29–43 (2023) doi: 10.33262/exploradordigital.v7i1.2451.
9. He, B., Radfar, M.: The Performance Evaluation of Attention-Based Neural ASR under Mixed Speech Input (2021) doi: 10.48550/arXiv.2108.01245.
10. Ide, N., Suderman, K., Pustejovsky, J.: The Open American National Corpus (OANC). Linguistic Data Consortium. https://www.anc.org/.
11. OpenAI.: Introducing Whisper (2022) https://openai.com/research/whisper.
12. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 2nd. Pearson Prentice Hall (2009)
13. Kasneci, E.: ChatGPT for good? On Opportunities and Challenges of Large Language Models for Education. Learning and Individual Differences, 103 (2023) doi: 10.1016/j.lindif.2023.102274.
14. Han, J., Yang, L.: Sentence Embedding Generation Framework Based on Kullback–Leibler Divergence Optimization and RoBERTa Knowledge Distillation. Mathematics, 12(24), pp. 3990 (2024) doi: 10.3390/math12243990.
15. V7 Labs. F1 Score in Machine Learning: Intro & Calculation (2022) https://www.v7labs.com/blog/f1-score-guide.
16. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004). Association for Computational Linguistics, pp. 74–81 (2004)
17. Ma, R.: Can Generative Large Language Models Perform ASR Error Correction? (2023) doi: 10.48550/arXiv.2307.04172.
18. Millière, R.: Language Models as Models of Language (2024) doi: 10.48550/arXiv.2408.07144.

19. Muñoz-Basols, J., Fuertes Gutiérrez, M.: Oportunidades de la inteligencia. La enseñanza del español mediada por tecnología. pp. 344–360 (2024) doi: 10.4324/9781003146391-18.
20. Panayotov, V.: Librispeech: An ASR Corpus Based on Public Domain Audio Books. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210 (2015) doi: 10.48550/arXiv.1506.03088.
21. Sellam, T., Das, D., Parikh, A.: BLEURT: Learning Robust Metrics for Text Generation. Proceedings of ACL (2020) doi: 10.48550/arXiv.2004.04696.
22. Rubenstein, P.K.: AudioPaLM: A Large Language Model That Can Speak and Listen. (2023) doi: 10.48550/arXiv.2306.12925.
23. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill (1983)
24. Topsakalm, O., Topsakal, E.: Framework for A Foreign Language Teaching Software for Children Utilizing AR, Voicebots and ChatGPT (Large Language Models). The Journal of Cognitive Systems, 7(2) (2022) doi: 10.52876/jcs.1227392.
25. Wang, B.: AudioBench: A Universal Benchmark for Audio Large Language Models (2024) doi: 10.48550/arXiv.2406.16020.
26. Wikipedia: Word Error Rate. https://es.wikipedia. org/w/index.php?title= Word_Error_ Rate&oldid=125822424.
27. Yang, Q.: AIR-Bench: Benchmarking Large Audio-Language Models via Generative Comprehension (2024) doi: 10.48550/arXiv.2402.07729.
28. Zhang. A. GitHub (2017) https://github.com/Uberi/speech_recognition.
29. Piantadosi, S.T.: Zipf's Word Frequency Law in Natural Language: A Critical Review and Future Directions. Psychonomic Bulletin & Review, 21(5), pp. 1112–1130 (2014) doi: 10.3758/s13423-014-0585-6.

ISSN 1870-4069

Online Sexism Detection Models

Karen Esmeralda Delgado Pérez, Néstor David García Franco,
Darnes Vilariño Ayala, Beatriz Martínez Beltrán,
David Eduardo Pinto Avendaño

Benemérita Universidad Autónoma de Puebla,
Mexico

{dp224470254, gf224470255@alm.buap.mx},
{darnes.vilarino, beatriz.beltran, david.pinto}@correo.buap.mx

Abstract. The automatic detection of online sexism represents a key challenge for content analysis in social networks, as current binary classification models are not always able to address all that sexist content may present. The present study focuses on Task 10: Task A of SemEval 2023, which consists of classifying comments as sexist or non-sexist. Two machine learning models based on natural language processing (NLP) techniques are presented. Unlike previous work focused on sentiment analysis, this approach is explicitly approached with the definition of sexism adopted by SemEval. The evaluation, performed on a dataset of Reddit and Gab comments, demonstrates that unigram-based models outperform bigram and trigram-based models in classification accuracy. This work seeks to advance the accurate and explainable detection of sexism in digital environments.

Keywords: Sexism, automatic learning, NLP.

1 Introduction

In a world where technology is constantly advancing, sexism has evolved, finding new forms of expression, such as online harassment and content that promotes gender-based hate. Despite efforts to achieve gender equality, there are still stereotypes and barriers that limit access to equal opportunities. From the wage gap to unequal representation in leadership positions, sexism affects both women and men, although it impacts differently and, in many cases, mainly women.

Sexism is understood as the discrimination of a person based on gender, promoting the idea that one sex is superior to the other and based on pre-established beliefs for one gender, negatively impacting the people affected. It is expressed in the use of language, with prejudicial attitudes about gender roles and practices that hinder access to equal rights and opportunities. Online sexism is a form of harassment, primarily against women, that aggressively focuses on minimizing their progress, making them feel assaulted and humiliated.

Automated tools help detect sexism on a large scale. However, binary detection ignores the diversity of sexist content that exists and does not provide

a clear explanation of why something is sexist, generating distrust and reducing its effectiveness. To address this issue, SemEval Task 10: SemEval 2023 Task Explainable Detection of Online Sexism (EDOS) is presented [1].

Task 10 is derived from shared tasks that dealt with the detection of abuse and hate. This task proposes and applies a taxonomy with 3 tasks: Task A is in charge of detecting whether the content is sexist or not; Task B classifies to which category of sexism it belongs and Task C subcategorizes the comments [1].

This paper addresses Task A of SemEval Task 10: detecting whether or not content is sexist with 2 different models, which determine whether comments are positive, negative or neutral, by using natural language processing techniques such as tokenization and lemmatization. The results are also compared with RoBERTa, a pre-trained model for content evaluation.

This article is organized to contextualize the results of the models that participated in SemEval 2023(Section 2). Section 3 presents the methodology of the implemented models, including the integration of LSA and LDA, meanwhile Section4 evaluates the effectiveness of models. Section 5 ends with the main conclusions of the study and future work.

2 Related Work

Among the top-performing systems presented at SemEval 2023, multiple models or ensemble-based approaches were used. Many of them applied additional training to their models and multitask learning. The teams that stood out the most were *stce* and *PASSTeam*, both of which used multitask learning and additional pre-training, obtaining better results on two or more tasks. [1].

The team *stce* used RoBERTa-large [2] and ELECTRA [3]. while the team *PASSTeam* used a multitask learning strategy [4] with tuned versions of RoBERTa y HateBERT [5].

For Task A, *PingAnLifeInsurance* team used a multitask neural network framework [6], in addition to performing additional pre-training with DeBERTa-v3 [7] and TwHIN-BERT [8] using unlabeled data and an additional Kaggle dataset. On the other hand, *FiRC-NLP* occupied a set of DeBERTa models fitted exclusively with the labeled task data.

In the case of Task B, *JUAGE* stood out as one of the few systems that used (prompt-based learning), achieving first place using the PaLM model tuned with [9] instructions for parameter optimization and majority voting over six iterations.

Moreover, the *PALI* team performed additional pre-training of DeBERTa-v3 with unlabeled data and included a second loss term based on the scaled cross-entropy, to address Task C.

Overall, Task B and Task C scores showed below average results and greater variability compared to Task A. All participating systems outperformed the simpler baseline, which was to predict the most frequent class, and most also outperformed a more complex baseline base on DeBERTa-v3 with continuous pre-training.

3 Proposed Methodology

The methodology proposed for the development of this research, which can be seen in Algorithm 1, is shown below 1.

Algorithm 1 Sentiment Analysis Process

Require: Complete Corpus C with n documents
Ensure: Classified polarity and generated vocabulary
1: $C_{limpio} \Leftarrow$ Formating(C) ▷ Cleans and saves each document
2: $Training$, $Test \Leftarrow$ RandomSampling(C_{limpio})
3: **for all** document d en $Training$ **do**
4: $polarity_d \Leftarrow$ AnalyzeFeelings(d) ▷ Classifies: positive, negative o neutral
5: $Vocabulary \Leftarrow Vocabulary \cup$ ExtractWords(d)
6: **end for**
7: **return** Polarity, Vocabulary

3.1 Format

It is in charge of cleaning the corpus data. It starts by reading the corpus, then processes the data of each document, filtering its columns depending on whether they contain information or not, removes common or repetitive words(stopwords) from the comments, and creates a folder that stores text files generated by each processed row and saves each of the documents in a new file organized by row.

3.2 Sampling

Subsequently, the set of files is randomly organized into two groups: one for training and one for test. Eighty percent of the documents are assigned for training and the remaining 20% for testing. To do this, we go through the subfolders generated in the previous point, selecting the files to be processed. It calculates how many files should be in the training set and randomly select the documents.In addition, two new subfolders are created, one for training data and one for testing. Finally, the selected files for the training and test sets are copied to the corresponding training and test folders.

3.3 Sentiment Analyzer

The VADER sentiment analyzer, which assigns a polarity score(positive, negative, neutral) to the content, and the Google Translate API were used to classify the sentiment of the comments.

The training set files are used to verify that the text is valid and that the file is not empty. Then, to classify the opinion, the text found in each of the files is translated into the following text.If the translation fails because the comment

does not exist or the file is empty, the empty, the sentiment is considered neutral. Otherwise, advance to the VADER analyzer to obtain the polarity score of the translated text. If the score is greater than 0, it is considered a positive comment; if the score is equal to 0, it is taken as neutral, and it is negative when the score is less than 0.

3.4 Vocabulary

Each text file is processed to extract the words and generate a vocabulary. Each word contained is related to the files where it appears.

To do this, we went through each of the subfolders in the indicated directory, read the text in each of the previously generated files, modified the text to lowercase, divided it by words, and returned to a file that includes all the words collected.

3.5 LSA

Latent Semantic Analysis(LSA) is a technique for creating vector representations of texts that aim to capture tptheir semantic content. The main function of LSA is to calculate the similarity of pairs of texts by comparing their vector representations. [11].

Before working with LSA a preprocessing of the content was performed, which includes cleaning up text (removing URLS, numbers, non-alphabetic characters and reducing spaces), separating the text into tokens, removing common words(in this particular case we included additional stopwords with a total of 1298 words [12], plus stopwords included by Spacy [23]) and lemmatize. The corpus cleans its save.

Subsequently, TF-IDF matrices were generated, used to calculate the relevance of a word within a document, for different n-grams (unigram, bigram, trigram). An n-gram is understood as a set of n consecutive elements for a text document [15].

3.6 LDA

Latent Dirichlet allocation (LDA) is a generative probabilistic model for collections of discrete data, such as text corpus. LDA is a three-level hierarchical Bayesian model in which each element of a collection is modeled as a finite mixture over an underlying set of topics. In turn, each topic is modeled as an infinite mixture over an underlying set of topic probabilities [13].

4 Results Analysis

In this study, different algorithms designed in Python were applied for Task A: detecting whether the content is sexist or not, classifying it as positive, negative, or neutral. Two proposals are presented: the first one consists of the Formatter,

Sampler, Sentiment Analyzer and Vocabulary. The second proposal consists of the application of the LDA and LSA algorithms.

As mentioned above, the methodology considers two proposals. The first one, consists in the consideration of n-grams, applied both to the training set and to the test set.

The second proposal applies LSA using TruncatedSVD from sklearn [10], to reduce the dimensionality of the feature space to a number of principal components and continued with a pretrained RoBERTa sentiment analyzer, the results are categorized as positive, negative, and neutral, and saved along with their polarity. This is done for both training and set.

A Hugging Face [24] feature was also added, which creates a specialized sentiment analysis pipeline. A pipeline is a series of preconfigured steps that allows specific tasks (such as sentiment analysis, translation, etc.) to be performed without having to configure each part of the process manually. In this case, the pipeline is configured to evaluate whether a text is positive, negative, or neutral. Finally, a graph of the sentiment distribution is generated.

After applying LSA, the text data were loaded and preprocessed. The TfidfVectorizer library is used to transform the texts into numerical feature vectors; converting the text into a numerical representation in which each word is represented by its TFIDF, which allows to measure the relative importance of each word in the corpus.

Class balancing techniques (SMOTE [20]) were applied, which helps to improve the performance of the models. By applying PCA (Principal Component Analysis) they were reduced to 2,000 principal components.

In addition, it was evaluated with two classification models (LDA and Random Forest) to predict the sentiment of the texts. Finally, confusion matrices were obtained, showing how many instances were correctly and incorrectly classified, and classification reports for both models (precision, recall, F1-score), which allowed us to evaluate the performance of the models.

4.1 Dimension of the Dataset

These models were trained from a corpus with one million unlabeled Reddit comments provided by SemEval (2023). Included within the dataset are uppercase, numeric, emoji, urls, and ASCII characters.

4.2 Evaluation of the Models

Training is used to build the model and it allows the classification of the test set. The dataset of 962,122 for the training set consists of the 769,696 documents, and the test set consists of 192,425 documents. To make the LSA and LDA tests more accurate, we used different n-grams (unigrams, bigrams and trigrams).

Where Accuracy shows the accuracy of the model; Precision is defined as how close we are to the data to be obtained; Recall is described as the ratio of true positive to false positive plus false negative; F1-Score is a harmonic measure

Table 1. Results obtained for unigrams with LDA.

$$\begin{bmatrix} 57531 & 49603 & 4111 \\ 9242 & 94521 & 7431 \\ 2166 & 38825 & 70248 \end{bmatrix}$$

	Unigram			
N-grams	Precision	Recall	F1-Score	Support
Negative	0.83	0.52	0.64	111245
Neutral	0.52	0.85	0.64	111194
Positive	0.86	0.63	0.73	111239
Accuracy			0.67	333678
Macro avg	0.74	0.67	0.67	333678
Weighted avg	0.74	0.67	0.67	333678

Table 2. Results obtained for unigrams with Random Forest.

$$\begin{bmatrix} 84694 & 22600 & 3951 \\ 13571 & 92338 & 5285 \\ 1359 & 6426 & 103454 \end{bmatrix}$$

	Unigram			
N-grams	Precision	Recall	F1-Score	Support
Negative	0.85	0.76	0.80	111245
Neutral	0.76	0.83	0.79	111194
Positive	0.92	0.93	0.92	111239
Accuracy			0.67	333678
Macro avg	0.84	0.84	0.84	333678
Weighted avg	0.84	0.84	0.84	333678

between precision and recall; Support is to increase the number of rare cases instead of simply duplicating the existing cases.

When evaluating the model, it is observed that the use of unigrams results in better accuracy compared to the results obtained using bigrams and trigrams.

The precision of the algorithm with unigrams with LDA is given for negative 0.83; neutral 0.52 and positive 0.86. See Table 1.

In the case of unigrams with Random Forest, negative 0.85; neutral 0.76 and positive 0.92 are obtained. See Table 2.

The classification precision for LDA bigrams was 0.72 for negative; 0.36 for neutral and 0.68 for positive. See Table 3.

In the case of bigrams with Random Forest it was 0.76 for negative; 0.38 neutral and 0.77 positive. See Table 4.

The results obtained for trigrams with LDA were negative 0.64; neutral 0.34 and positive 0.57. See Table 5.

Table 3. Results obtained for bigrams with LDA

$$\begin{bmatrix} 14009 & 94363 & 2873 \\ 3129 & 104819 & 3246 \\ 2235 & 95766 & 13238 \end{bmatrix}$$

	Bigram			
N-grams	Precision	Recall	F1-Score	Support
Negative	0.72	0.13	0.21	111245
Neutral	0.36	0.94	0.52	111194
Positive	0.68	0.12	0.20	111239
Accuracy			0.40	333678
Macro avg	0.59	0.40	0.31	333678
Weighted avg	0.59	0.40	0.31	333678

Table 4. Results obtained for bigrams with Random Forest

$$\begin{bmatrix} 25874 & 82152 & 3219 \\ 6367 & 100749 & 4078 \\ 1835 & 85335 & 24069 \end{bmatrix}$$

	Bigram			
N-grams	Precision	Recall	F1-Score	Support
Negative	0.76	0.23	0.36	111245
Neutral	0.38	0.91	0.53	111194
Positive	0.77	0.22	0.34	111239
Accuracy			0.45	333678
Macro avg	0.63	0.45	0.41	333678
Weighted avg	0.63	0.45	0.41	333678

While the precision in trigrams with Random Forest was negative 0.78; neutral 0.34 and positive 0.84. See Table 6.

To complement and contrast the results obtained using the LSA and LDA methodology, we incorporated the RoBERTa model, which has demonstrated outstanding results in text classification task, and is used for large volumes of unstructured data. In order to use RoBERTa, we applied a binary labeling scheme (sexism/no-sexism) which allowed us to evaluate its effectiveness against n-gram-based methods. This model improves the analysis of the entire data set (962,122). Show the results in the table 7.

The results obtained in the sexism class are based on the semantic and contextual complexity of the language. Unlike explicit non-sexist comments, the expressions are often composed in a subtle, ironic, or implicit manner, which makes them difficult to detect.

Table 5. Results obtained for trigrams with LDA.

$$\begin{bmatrix} 1634 & 108939 & 672 \\ 530 & 110174 & 490 \\ 370 & 109309 & 1560 \end{bmatrix}$$

	Trigram			
N-grams	Precision	Recall	F1-Score	Support
Negative	0.64	0.01	0.03	111245
Neutral	0.34	0.99	0.50	111194
Positive	0.57	0.01	0.03	111239
Accuracy			0.34	333678
Macro avg	0.52	0.34	0.19	333678
Weighted avg	0.52	0.34	0.19	333678

Table 6. Results obtained for trigrams with Random Forest.

$$\begin{bmatrix} 2904 & 108081 & 260 \\ 676 & 110310 & 208 \\ 150 & 108653 & 2436 \end{bmatrix}$$

	Trigram			
N-grams	Precision	Recall	F1-Score	Support
Negative	0.78	0.03	0.05	111245
Neutral	0.34	0.99	0.50	111194
Positive	0.84	0.02	0.04	111239
Accuracy			0.35	333678
Macro avg	0.65	0.35	0.20	333678
Weighted avg	0.65	0.35	0.20	333678

5 Conclusions and Future work

Online sexism has represented an important challenge, where anonymity and accessibility to platforms have facilitated the propagation of hate speech and gender discrimination. Throughout this study, various strategies for automatic detection of sexism have been analyzed, highlighting the effectiveness of models based on transformers and deep learning approaches. Previous results reflect that models such as RoBERTa, DeBERTa-v3 and HateBERT can identify patterns of sexism with high accuracy, although challenges remain in classifying ambiguous cases and explaining the detected biases.

The analysis of n-grams and techniques such as Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) have provided a better understanding of the linguistic structure of sexist content, providing a more detailed approach to detection. However, there is a need for further development

Table 7. Results with RoBERTa.

	RoBERTa			
Etiquetas	Precision	Recall	F1-Score	Support
Sexism	0.0357	0.0357	0.0357	36498
No-sexism	0.9620	0.9620	0.9620	925622
Accuracy			0.92689	962120
Macro avg	0.4989	0.4989	0.4989	962120
Weighted avg	0.9268	0.9268	0.9268	962120

of models that not only classify content but also provide clear explanations as to why a given text is considered sexist.

However, it is necessary to continue to develop models that not only classify content but also provide clear explanations as to why a given text is considered sexist. As future work, we plan to improve the models presented, extend the study to tasks B and C of SemEval, in order to classify the different categories and subcategories of the sexism; to analyze common errors in the classifications in order to refine both preprocessing and representation of the text, and to incorporate refine both the preprocessing and the representation of the text, and incorporate other pretrained other pretrained models, such as DeBERTa and HateBERT for sexism detection.

References

1. Kirk, Hannah, Wenjie Yin, Bertie Vidgen, and Paul Röttger. "SemEval-2023 Task 10: Explainable Detection of Online Sexism." In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, edited by Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori, 2193–2210. Toronto, Canada: Association for Computational Linguistics, 2023. https://aclanthology.org/2023.semeval-1.305/.doi:10.18653/v1/2023.semeval-1.305.
2. Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." ArXiv preprint arXiv:1907.11692, 2019. https://arxiv.org/abs/1907.11692.
3. Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators." In *International Conference on Learning Representations (ICLR)*, 2020. https://openreview.net/forum?id=r1xMH1BtvB.
4. Yu, Tianhe, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. "Gradient Surgery for Multi-Task Learning." In *Advances in Neural Information Processing Systems*, volume 33, 5824–5836. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper/2020/hash/3fe230348e9a12c13120749e3f9fa4cd-Abstract.html.
5. Caselli, Tommaso, Valerio Basile, Jelena Mitrović, and Michael Granitzer. "HateBERT: Retraining BERT for Abusive Language Detection in English."

In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, 17–25. Online: Association for Computational Linguistics, 2021. https://aclanthology.org/2021.woah-1.3/.

6. Liu, Xiaodong, Yu Wang, Jianshu Ji, Hao Cheng, Xueyun Zhu, Emmanuel Awa, Pengcheng He, Weizhu Chen, Hoifung Poon, Guihong Cao, and Jianfeng Gao. "The Microsoft Toolkit of Multi-Task Deep Neural Networks for Natural Language Understanding." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 118–126. Online: Association for Computational Linguistics, 2020. https://aclanthology.org/2020.acl-demos.14/.
7. He, Pengcheng, Jianfeng Gao, and Weizhu Chen. "DeBERTaV3: Improving DeBERTa Using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing." ArXiv preprint arXiv:2111.09543, 2021. https://arxiv.org/abs/2111.09543.
8. Zhang, Xinyang, Yury Malkov, Omar Florez, Serim Park, Brian McWilliams, Jiawei Han, and Ahmed El-Kishky. "TwHIN-BERT: A Socially-Enriched Pre-trained Language Model for Multilingual Tweet Representations." ArXiv preprint arXiv:2209.07562, 2022. https://arxiv.org/abs/2209.07562.
9. Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. "PaLM: Scaling Language Modeling with Pathways." ArXiv preprint arXiv:2204.02311, 2022. https://arxiv.org/abs/2204.02311.
10. Scikit-learn. (2025). *Installation Guide*. Recuperado de https://scikit-learn.org/stable/install.html
11. P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. (2004, November). *Latent semantic analysis*. En *Proceedings of the 16th international joint conference on Artificial intelligence*, pp. 1-14.
12. Stopwords ISO. (n.d.). Recuperado de https://github.com/stopwords-iso/stopwords-en/blob/master/stopwords-en.txt
13. D. M. Blei, A. Y. Ng, and M. I. Jordan. (2003). *Latent dirichlet allocation*. Journal of Machine Learning Research, 3(Jan), 993-1022.
14. Zahraa Berjawi. (2022). *Benevolent Sexism Detection in Text: A Data-Centric Approach*. PhD Thesis.
15. MathWorks. (n.d.). *N-grams*. Recuperado de https://la.mathworks.com/discovery/ngram.html.
16. G. T. Rahutami and F. Z. Ruskanda. (2023). *Sexism Detection and Classification Using RoBERTa and Data Augmentation*. En *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, Lombok,

Indonesia, pp. 1-6. doi: 10.1109/ICAICTA59291.2023.10390414. keywords: Training;Social networking (online);Data augmentation;Transformers;Data models;Informatics;sexism;text classification;RoBERTa;data augmentation

17. H. Mohammadi, A. Giachanou, and A. Bagheri. (2024). *A transparent pipeline for identifying sexism in social media: Combining explainability with model prediction.* Applied Sciences, 14(19), 8620.
18. E. Martinez, J. Cuadrado, J. C. Martinez-Santos, and E. Puertas. (2023). *Detection of Online Sexism Using Lexical Features and Transformer.* En *2023 IEEE Colombian Caribbean Conference (C3)*, Barranquilla, Colombia, pp. 1-5. doi: 10.1109/C358072.2023.10436298. keywords: Social networking (online);Merging;Linguistics;Transformers;Feature extraction;Natural language processing;Global communication;Transformers;Lexical Features;Social Media;Misogyny;Sexism
19. A. Chaudhary and R. Kumar. (2023). *Sexism Identification In Social Networks.* En *CLEF (Working Notes)*, pp. 891-900.
20. Imbalanced-learn. (2025). *SMOTE.* Recuperado de `https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html`.
21. Scikit-learn. (2025). *sklearn metrics — Metrics and scoring.* Recuperado de `https://scikit-learn.org/stable/api/sklearn.metrics.html`.
22. Bird, Steven, Ewan Klein, and Edward Loper. (2009). *Natural Language Processing with Python.* O'Reilly Media. Recuperado de `https://books.google.com/books/about/Natural_Language_Processing_with_Python.html?hl=es&id=Au-_DwAAQBAJ`.
23. spaCy. (2025). *spaCy 101: Everything you need to know.* Recuperado de `https://spacy.io/usage/spacy-101`.
24. Hugging Face. (2025). *Models – Hugging Face.* Recuperado de `https://huggingface.co/models`.

ISSN 1870-4069

Classification of Offensive Comments on the Web Using SVM

Claudio Eduardo Gómez Cabrera[1], Abdiel Reyes Vera[2]

[1,2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Mexico

{claedgomcab, abdielreyes81}@gmail.com

Abstract. This paper presents a web application for the classification of offensive comments using Support Vector Machines (SVM). A corpus generated by web scraping of video games and movies review platforms was used. Natural Language Processing (NLP) techniques such as tokenization and TF-IDF were implemented for data representation. The results show that the proposed model improves the identification of offensive content with high accuracy, providing an efficient solution to improve moderation on digital platforms.

Keywords: Text classification, NLP, SVM, web scraping.

1 Introduction

The growth of digital platforms has facilitated global communication, but it has also generated an increase in offensive comments, hate speech, and online harassment. Moderating this content is a challenge, as it must balance freedom of expression with the creation of safe spaces for users. Although manual moderation is still used, it is a slow and costly process, requiring automated solutions based on artificial intelligence.

This paper proposes the use of Support Vector Machines (SVM) for the classification of offensive comments, using Natural Language Processing (NLP) techniques. For this purpose, data were collected from platforms such as Steam and Metacritic through web scraping, which were preprocessed and manually labeled. The performance of SVM was compared with other classification models, demonstrating its effectiveness in detecting offensive content with high accuracy. Despite the rise of deep learning architectures such as Transformers, traditional models like SVM remain highly competitive, especially when computational resources or data availability are limited. Their lower complexity and faster training times make them ideal for integration into lightweight applications such as browser extensions.

In addition, a web extension was developed to implement real-time moderation within browsers. This tool operates independently of platform infrastructure, offering a scalable and accessible solution for content moderation. This article explores the theoretical foundations of text classification, the method- ology employed, the results

obtained, and the implications of this solution for improving moderation in digital environments.

2 Theoretical Framework

In order to develop an artificial intelligence-based offensive comment moderation system, it is essential to understand the concepts underlying this solution. This section addresses the principles of automatic learning, the use of Support Vector Machines (SVM) in text classification, and the Natural Language Processing (NLP) techniques applied in data preprocessing.

2.1 Machine Learning

Machine Learning allows models to identify patterns in data without being explicitly programmed. It is divided into several categories, with supervised learning being the most relevant in this work, as the SVM model is trained on previously labeled data.

Supervised Learning This approach is based on learning a function that relates the input data to their respective output labels. For this purpose, the data are divided into two sets: training, used to fit the model, and testing, used to evaluate its performance. (El Naqa & Murphy, 2015)

2.2 Natural Language Processing

NLP is a branch of artificial intelligence focused on the interaction between computers and human language.

In this work, techniques such as tokenization, lemmatization and TF-IDF vector representation were employed to transform comments into a format processable by classification models.

2.3 Term Frequency - Inverse Document Frequency (TF-IDF)

To represent the comments numerically, we used TF-IDF (Term Frequency - Inverse Document Frequency), a technique that measures the importance of a word in a set of documents.

This methodology made it possible to highlight terms characteristic of offensive comments and to minimize the impact of common words:

- **Term Frequency (TF)**: Represents how many times a word appears in a comment.
- **Inverse Document Frequency (IDF)**: Penalizes words that are too common throughout the corpus, reducing their weight in the classification.

The use of TF-IDF helped the SVM model to detect patterns in offensive language without the need for a predefined list of forbidden words, since insults tend to be repeated and acquire greater weight within the corpus.

2.4 Support Vector Machines (SVM)

Support Vector Machines (SVM) are supervised learning models used for classification and regression. Their goal is to find a hyperplane that maximizes the separation between different classes, optimizing the accuracy of the model.

Maximum Margin Concept: The margin is the distance between the separator hyperplane and the nearest points of each class (support vectors). Maximizing this margin improves the model's ability to generalize and minimize classification errors in unseen data. (Jakkula, 2006).

2.5 Web Scraping and Legal Considerations

Web scraping is an automated technique used to extract information from web- sites. While it facilitates the collection of data for training NLP models, its use must adhere to international legal and ethical standards. Organizations such as the United Nations (UN) and the United Nations Educational, Scientific and Cultural Organization (UNESCO) emphasize the importance of protecting per- sonal data and respecting digital rights.

To minimize legal and ethical risks, it is recommended to verify whether a platform provides an official API and to consult its terms of service before applying scraping techniques (Kinsta, 2025).

What data can be scraped: It is possible to extract any type of data from different web pages. Many websites allow scrapers to access their data, but not all do so for free. The way in which different types of data can be obtained depends on the website. If there is an API (Application Programming Interface) to obtain the necessary data, it is recommended to use them.

However, this practice is not always well-received, and some sites may im- pose limitations. There are some tricks you can implement to circumvent these limitations, taking care not to violate the website's rules or breach copyright or personal data protection laws (Kinsta, 2025).

Privacy and Data Protection Risks: A common misconception is that scraping public websites is always legally permissible. However, data protection laws apply regardless of whether the information is publicly accessible. For instance, under the General Data Protection Regulation (GDPR), personal data must be processed lawfully, fairly, and transparently. Even public data, if linked to identifiable individuals, falls under the scope of protection.

Personal data includes names, email addresses, IP addresses, device identifiers, and other information that can be associated with an individual: Improper collection or use of such data without consent may lead to significant legal consequences. Therefore, it is crucial to ensure that any scraping activity respects data privacy frameworks and ethical guidelines to avoid breaching legal obligations. (EDJ-XTECH-LAW-SCHOOL, 2023)

3 State of the Art

Automatic moderation of offensive comments has been extensively studied within Natural Language Processing (NLP) and artificial intelligence. There are multiple approaches,

ranging from traditional methods such as Support Vector Ma- chines (SVM) to advanced models such as Transformers and Convolutional Neural Networks (CNN).

3.1 Hate Speech and Offensive Language Detection

One of the main challenges in classifying offensive comments is to identify the context and intent of the language. In (Rodríguez & Pérez, 2023), we experimented with the Pysentimiento model based on Transformers, obtaining good results in the classification of emotions and polarity in Spanish. It was shown that neural models can better capture language semantics compared to traditional techniques such as TF-IDF + SVM.

On the other hand, (Jiménez & Rojas, 2022) evaluated the performance of SVM and CNNs in detecting hate on Twitter. The results showed that SVM, combined with TF-IDF, can achieve good results, although CNNs outperformed them due to their ability to capture more complex semantic relationships.

3.2 Models Based on Transformers

Transformers, such as BERT, have significantly improved the detection of offensive language in social networks. In (Basile & et al., 2019), BERT was compared with SVM, noting that BERT achieves higher accuracy and F1-score due to its ability to interpret context. However, these models require more computational power and large volumes of data for training.

A hybrid approach proposed in (Park, Shin, & Lee, 2020) combined BERT and CNN to improve the classification of offensive language, capturing both textual features of the text and local patterns of offensive keywords. Although this technique achieved better results, its high computational cost represents a limitation.

3.3 Comparison of Methods and SVM Justification

Although advanced models have demonstrated superior performance, traditional methods such as SVM remain a viable alternative in resource-limited scenarios. In this work, SVM with TF-IDF was chosen due to:

- **Size of the data set:** The corpus used consists of 4049 comments, which is insufficient for models such as BERT, which require large volumes of data.
- **Computational efficiency:** Transformers and CNNs require more processing power, making them difficult to implement in real time. SVM is a lighter and more efficient alternative.
- **Good performance with TF-IDF:** SVM achieved an accuracy of 84.09% in this study, demonstrating its feasibility in detecting offending comments with small datasets.

Given these factors, SVM represents a balance between accuracy, efficiency and ease of implementation, being suitable for automated content moderation in a web extension.

4 Methodology

To evaluate the effectiveness of Support Vector Machines (SVM) in the classification of offensive comments, a structured process was followed that included model collection, preprocessing, training and evaluation. Data were obtained by web scraping from platforms such as Steam and Metacritic, applying Natural Language Processing (NLP) techniques for cleaning and conversion to numerical representations with TF-IDF.

The SVM model was trained with a previously labeled corpus and compared with other classification algorithms, such as KNN and Random Forest. Finally, a web extension was implemented to apply the model in real time within the browsers. The main steps of the process are detailed below.

4.1 Data Collection

Web scraping was used to extract comments from review platforms, saving them in a CSV file for further processing with the Pandas library in Python. Comments were classified into two categories: 1 (Non-offensive) and 0 (Offensive). A comment was considered not offensive if it did not contain:

- Offensive words.
- Sexual Apology.
- Ambiguous content.
- Excessively short comments (two words or less).

These criteria were established after analyzing the recurrent patterns in offensive comments. In total, 4049 comments were collected, of which 3240 were used for model training and the rest for the testing phase.

4.2 Data Labeling Process

To label the dataset, three domain experts manually annotated each comment as offensive or non-offensive. The labeling criteria were based on the presence of profanity, hate speech, discrimination, or personal attacks. In cases of dis- agreement, a consensus was reached through discussion. This manual approach ensured the reliability of the labeled data used for training and testing the classification models.

If some data do not meet any of the criteria established by the consensus, the data in question are discarded to improve the quality of the overall data and not contribute useless data to the model, thus improving optimal classification in the training of the model.

4.3 Data Preprocessing

Before training the models, the text data was preprocessed through several steps: conversion to lowercase, removal of punctuation and special characters, elimination of stopwords, and tokenization:

Table 1. Accuracy of the evaluated models.

Model	Accuracy	Description
KNN	66%	Calculate the distance between the point to be classified and its nearest neighbors.
RF	72%	Create multiple decision trees and take the most common classification among them.
SVM	84%	Find an optimal hyperplane that maxi- mizes the margin between categories.

- **Tokenization:** Tokenization was used at the word level to divide the comments into individual units, eliminating punctuation marks and separating each term. This allowed structuring the text so that it could be analyzed by the classification model.
- **Elimination of Stopwords:** Stopwords are frequent words such as "the", "of", "and", which do not contribute value in the classification. They were eliminated using predefined lists in Spanish and English, reducing the dimensionality of the data and improving the efficiency of the model.
- **Text Normalization:** To ensure a uniform treatment of words, the following techniques were applied:
- **Lowercase conversion:** Prevents identical words from being treated as distinct terms (e.g., "Video game" and "video game").
- **Elimination of special characters and emojis:** Symbols and emojis were discarded as they did not provide useful information for classification.
- **Filtering Ambiguous Content:** Extremely short comments (one or two words) were eliminated because they did not provide significant information for classification. Also discarded were those composed only of special characters or graphic patterns used to generate figures with inappropriate content.

4.4 Model Training

The SVM model was trained with a manually labeled corpus, using TF-IDF for text representation. Its performance was compared with KNN and Random Forest, obtaining better results.

As shown in Table 1, SVM obtained the best performance. The regularization parameter "C", which balances margin maximization and classification error minimization, was optimized. To determine its optimal value, tests were performed with different values, observing their impact on accuracy and recall.

It was found that low values of "C" favored a wide margin with more errors, while high values reduced error tolerance, generating overfitting. After several iterations, the value offering the best balance was selected.

Table 2. Comparison of Metrics.

Model	Class	Accuracy	Recall	F1-Score
KNN (66.50%)	0 (Offensive)	0.84	0.70	0.76
	1 (Not Offensive)	0.61	0.62	0.61
Random Forest (72.00%)	0 (Offensive)	0.75	0.80	0.77
	1 (Not Offensive)	0.70	0.70	0.70
SVM (84.09%)	0 (Offensive)	0.94	0.80	0.87
	1 (Not Offensive)	0.70	0.90	0.79

4.5 Hyperparameter Tuning

Each classifier was trained using a specific set of hyperparameters. The Sup- port Vector Machine (SVM) and Random Forest (RF) classifiers were manually configured based on preliminary experimentation, while K-Nearest Neighbors (KNN) was used with default parameters due to its low observed performance.

- **Support Vector Machine (SVM):** A radial basis function (RBF) kernel was used, with a regularization parameter $C = 1$ and kernel coefficient $\gamma = 0.2$. This configuration was chosen to handle non-linear relationships within the TF-IDF-transformed feature space.
- **Random Forest (RF):** The number of decision trees was increased to 300 to enhance model stability and reduce variance. Other parameters, such as maximum depth and minimum samples per leaf, were kept at their default values.
- **K-Nearest Neighbors (KNN):** The classifier was applied with default settings, including $k = 5$ and Euclidean distance. No tuning was performed, as initial tests showed low classification accuracy, making further optimization less relevant.

4.6 Justification for the Choice of SVM

For the classification of offensive comments, different models were evaluated with metrics such as accuracy, precision, recall, and F1-score. SVM proved to be the best option.

Comparison with Other Models Evaluated KNN and Random Forest were evaluated, with inferior results:

- **KNN**: Although it had good accuracy in the offensive class (0.84), its performance in the non-offensive class was low (0.61), with an overall accuracy of 66.50%.
- **Random Forest**: Although more balanced, its accuracy was only 72.00%, with an F1-score of 0.70 in the non-offensive class.

SVM outperformed both models in accuracy and classification of non-offensive comments, being the best choice for this task.

Reasons for SVM Selection SVM was chosen for the following reasons:

- **High accuracy:** Achieved 84.09%, surpassing other models.
- **Accuracy in non-offensive comments:** Obtained an accuracy of 0.94, minimizing

false positives.

- **Efficiency in detecting offensive comments:** A recall of 0.90 indicates a low number of false negatives.
- **Effective generalization:** Thanks to the maximum margin principle, it avoids overfitting and improves classification on new data.
- **High dimensionality handling:** SVM works efficiently with TF-IDF, find- ing optimal hyperplanes in multidimensional spaces.
- **Computational efficiency:** It requires fewer resources compared to neural networks, facilitating its implementation in a web extension.

4.7 Hyperparameter Tuning

The K-Nearest Neighbors (KNN) classifier was tested with different values of *k* ranging from 3 to 10, and the best results were achieved with $k = 5$. For the Random Forest (RF) classifier, we used 100 trees with a maximum depth of 10. No grid search or cross-validation was performed for hyperparameter tuning in this initial version, but it is considered as a future improvement.

4.8 Use of TF-IDF

To represent the comments numerically, TF-IDF was used. This technique al- lowed us to evaluate the importance of each word within the corpus, assigning greater weight to terms that were distinctive of offensive comments.
The use of TF-IDF improved the classification due to:

- **Repetition of offensive words**: Insults and aggressive expressions are often recurrent in offensive comments. TF-IDF assigns greater weight to these terms, facilitating their identification.
- **Differentiation of key terms**: Offensive expressions have a wide distribution. The model is able to detect them without predefined lists of prohibited words.
- **Reduction of the impact of common terms:** TF-IDF balances the weight of frequent words, preventing the model from being biased towards overused terms in both types of comments.

Unlike a simple Bag of Words, TF-IDF allowed not only to count terms, but also to weight them according to their relevance in the classification. Thanks to this representation, SVM identified more accurately offensive linguistic patterns, improving the performance of the system.

4.9 Web Extension Development

To implement automated moderation of offensive comments, a web extension was developed that interacts with a Python API. The extension does not directly extract the comments, but sends the URL of the page to the API, which performs web scraping, classifies the comments with SVM and returns only the non offensive ones for display.

Interaction with Web Pages and Content Modification**.** The web extension

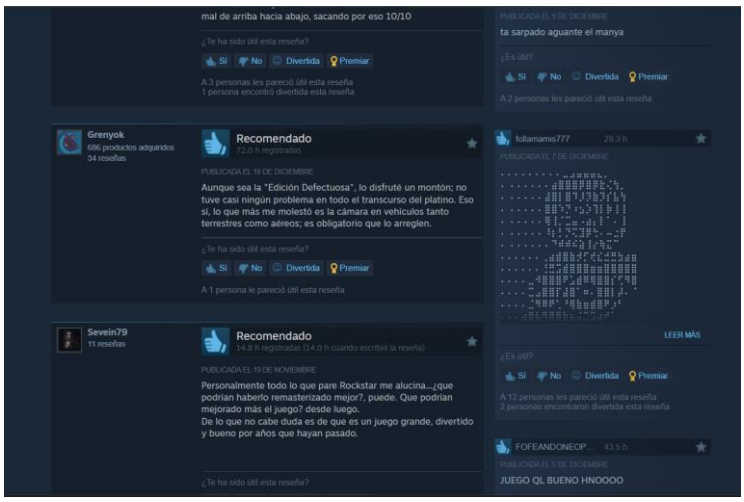

Fig. 1. Inappropriate comments in the Steam comments section.

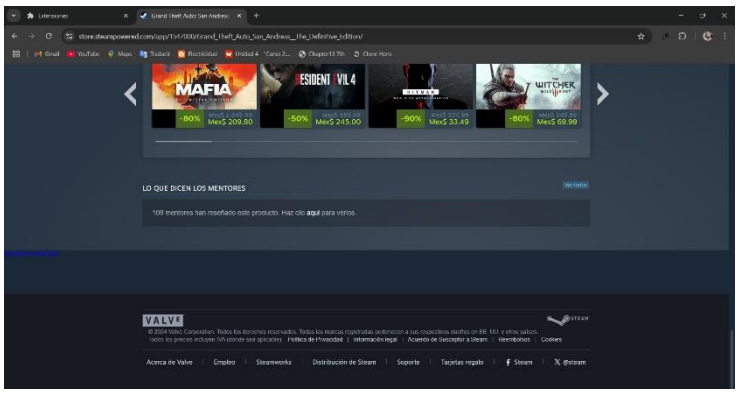

Fig. 2. Hidden inappropriate comments.

dynamically modifies the structure of the visited pages to ensure a safe environment and a clean presentation. It works as follows:

- **Sending the URL to the API**: The extension captures the URL of the page with comments and sends it to the API for analysis.
- **Extraction and classification**: The API applies web scraping to obtain the comments and uses SVM to classify them as offensive or non-offensive.
- **Display of filtered comments**: The API returns only non-offensive comments, and the extension modifies the page in real time to display them, hiding the offensive ones without affecting the visual structure.
- **Dynamic modification of the content**: The extension replaces the section original feedback with the filtered ones, ensuring a seamless experience.

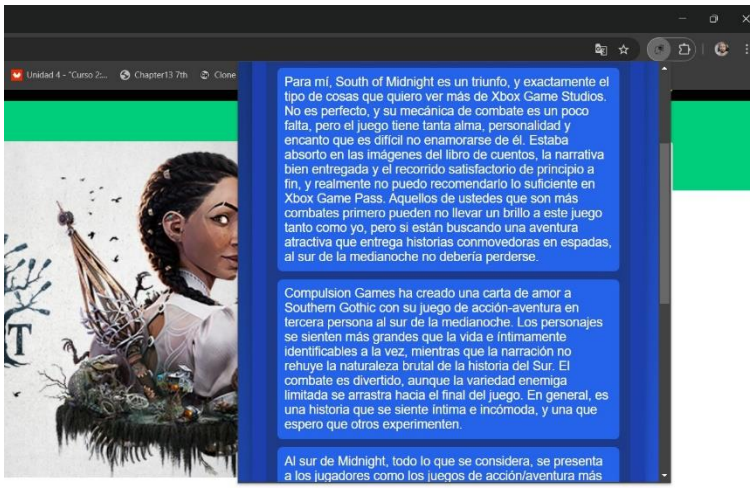

Fig. 3. Classified comments in the extension.

In Fig. 1, inappropriate content is displayed in the comments section. There- fore, the dynamic modification provided by the web extension was used to hide the comments section on the various pages to which the rating applies and pro- vide a space free of inappropriate content.

Once the web extension has modified the web page's source code, the comments section is hidden as shown in Fig.2 to prevent the user from encountering such content and leaving them only the option of viewing the appropriate comments with the web extension.

Communication between the Extension and the API The communication process between the web extension and the API follows these steps:

- **Capture and send URL**: The extension detects the visited page and sends the URL to the API via HTTP request.
- **Extraction and classification**: The API identifies the structure of the page, extracts the comments and processes them with the SVM model.
- **Return of filtered comments**: The API responds with the comments classified as appropriate Fig.3.

Impact on Security and User Experience. The extension ensures a secure environment by displaying only API-filtered comments, preventing exposure to inappropriate content without manual intervention. In addition, it visually restructures the page to avoid empty spaces or clutter in the original layout, providing a smooth user experience without alterations to the functionality of the visited website.

Beyond content moderation, the implementation of the web extension rep- resents a scalable and adaptable solution for different platforms. Its integration with an API makes it possible to continuously update and improve the classification criteria without the need to modify the extension's code. This facilitates the implementation of future improvements, such as the incorporation of new machine learning models or the adaptation to changes in the structure of the analyzed web pages.

As can be seen in Fig.3, the filtered comments are displayed in the web extension once the previously mentioned communication is completed and the classified comments are returned within the API.

5 Results

The results show that the SVM model achieved an accuracy of 84.09% in classifying offensive comments, outperforming KNN (66.50%) and Random Forest (72%). This confirms that Support Vector Machines are suitable for this task by balancing accuracy and generalization.

5.1 Model Performance

In the Table 2, shows that the SVM model achieved an F1-score of 0.87 for non offensive comments, indicating a high predictive ability in this category. For offensive comments, the model achieved a recall of 0.90, which means that it was able to identify most of the inappropriate comments with a low margin of error. In contrast, KNN showed difficulties in classifying non-offensive comments, with an F1-score of 0.61, while Random Forest obtained a moderate improvement at 0.70. This suggests that SVM not only outperforms these models in overall accuracy, but also reduces false positives and false negatives, crucial aspects for automated moderation.

5.2 Efficiency in Web Extension

During testing, the web extension ran on multiple platforms, allowing offensive comments to be filtered without affecting the structure of the pages. Integration with the API enabled seamless real-time moderation, ensuring that only comments classified as non-offensive were displayed.

Despite the good results, challenges were identified in classifying very short comments, as in some cases words with high TF-IDF weights led to incorrect classifications. To reduce these errors, a bag of words was created containing problematic terms identified in the analysis. These words were assigned a lower weight within the model, allowing to adjust its impact on the classification and improving accuracy in cases where the context was limited.

However, the system showed high effectiveness in detecting offensive language, providing a viable solution for online content moderation.

6 Conclusions

The effectiveness of the Support Vector Machines (SVM) based model for comment moderation in digital platforms was demonstrated. Through the use of Natural Language Processing (NLP) and text representation using TF-IDF, we were able to train a model capable of identifying offensive comments with high accuracy, providing an effective tool to improve security in digital environments. The use of web scraping was fundamental in data collection, allowing us to obtain a representative corpus of real

comments from platforms such as Steam and Metacritic. This technique facilitated the construction of a labeled dataset, essential for the training and evaluation of the model. Despite the challenges presented by web scraping, such as the restrictions imposed by some platforms, its implementation proved to be a viable alternative for obtaining data in text classification projects.

In addition, the integration of the model into a web extension allowed for real time implementation within browsers. This represents a practical and accessible solution for moderating comments on various platforms, without the need to modify the infrastructure of the websites. The ability of web extensions to interact with HTML code and communicate with an external API proved to be a key factor in the implementation of the offensive comment filtering system.

Despite the rise of models based on Transformers and profound neural net- works, the choice of SVM was appropriate for several reasons:

- **Lower computational cost**: SVM does not require the high computational power demanded by more complex models, which facilitates its implementation in resource-constrained environments.
- **Increased interpretability**: Compared to deep networks, SVM models allow a better understanding of the model's decisions, which is beneficial in content moderation applications.
- **Efficiency on small datasets**: With a corpus of 4049 comments, SVM achieved competitive accuracy without requiring large volumes of data for training.

In conclusion, this project presents an effective and scalable solution for the detection and moderation of offensive comments on the web. Its implementation in browsers through web extensions makes it an accessible and easy to integrate tool, with the potential to improve the user experience in digital environments.

References

1. Basile, V.: Um-iu@ling at Semeval-2019 Task 6: Identifying Offensive Tweets Using Bert and SVMs. ArXiv doi: 10.48550/arXiv.1904.03450.
2. Google Developers: Chrome extensions overview (2023) https:// developer.chrome.com/docs/extensions.
3. EDJ-XTECH-LAW-SCHOOL (2023). Los riesgos legales del web scraping: Privacidad, protección de datos y malos usos. Retrieved from https:// www.edjxtechlawschool.com/post/los-riesgos-legales-del-web-scraping-privacidad-protecci%C3%B3n-de-datos-y-malos-usos
4. Electronic Frontier Foundation: Automated Content Moderation: Challenges and Recommendations (2020) https://www.eff.org/issues/automated-content-moderation
5. El Naqa, I., Murphy, M.J. What is Machine Learning? In: Machine Learning in Radiation Oncology: Theory and Applications, pp. 3–11 (2015) doi: 10.1007/978-3-319-18305-3_1.
6. Microsoft Foundation: Browser Extensions – Introduction (2023) https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/ WebExtensions.

7. Hardeniya, N., Perkins, J., Chopra, D.: Natural Language Processing: Python and NLTK. Packt Publishing Ltd (2016)
8. Jakkula, V.: Tutorial on Support Vector Machine (SVM) (2006) https://www.public.asu.edu/~jakkula/tutorialsvm.pdf.
9. Jiménez, M., Rojas, C.: Comparison of Models for Automatic Hate Speech Detection on Twitter. https://www.kerwa.ucr.ac.cr/items/340cc182-c780-47a0-ad7b-ec4495f2dbd0
10. Kinsta: ¿Qué es el web scraping? Cómo extraer legalmente el contenido de la web (2025) https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping.
11. Park, J., Shin, J., Lee, S.G.: KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. arXiv (2020) doi: 10.48550/arXiv.2007.13184.
12. ProxyElite.: Aspectos legales del web scraping: Lo que necesita saber para evitar infringir la ley (2022) https://proxyelite.info/es/legal-aspects-of-web-scraping-what-you-need-to-know-to-avoid-breaking-the-law/.
13. Rodríguez, A., Pérez, L.: Aggression and Hate in Spanish Text Messages: Identification Using Transformers (2023) https://laccei.org/LACCEI2023-BuenosAires/papers/Contribution_1077_a.pdf
14. Spärck Jones, K.: A statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation, 28(1), pp. 11−21 (1972)
15. UNESCO.: La gobernanza de internet y la protección de derechos humanos (2021) https://unesdoc.unesco.org/ark:/48223/ pf0000377231.

ISSN 1870-4069

Modelado de un sistema difuso para el ahorro de agua en los centros de lavado de autos con modelos Mamdani, Sugeno y Tsukamoto

Bartolome Tellez Chavez, Perfecto Malaquías Quintero Flores, Rodolfo Eleazar Pérez Loaiza

Instituto Tecnológico de Apizaco, México

m23370010@apizaco.tecnm.mx
parfait.phd@gmail.com
rodolfo.pl@apizaco.tecnm.mx

Resumen. El proceso de lavado de autos representa un consumo considerable de recursos como el agua, el detergente automotriz y el tiempo de lavado, con el objetivo de optimizar este proceso, en este artículo presenta el modelado de sistema difuso para el monitoreo y control del proceso de lavado de autos buscando ahorrar el tiempo de lavado, agua y detergente automotriz. El modelado está desarrollado en la caja de herramientas de Matlab (Fuzzy Logic Toolbox) en inglés, donde se emplean tres variables de entrada: *tamaño del auto*, *programa de lavado* y la *suciedad*, cada una con sus respecivas funciones de membresía, y tres variables de salida: *cantidad de agua* (litros), cantidad de *detergente automotriz* (mililitros) y el *tiempo de lavado*(minutos). Posteriormente, se establecen reglas de inferencia difusa utilizando el método de Mamdani, Sugeno y Tsukamoto para lograr los resultados esperados, es decir, una cantidad cercana a la óptima de agua(*cantidad de agua*), *detergente automotriz* y el *tiempo de lavado* requerido para el lavado del auto.

Palabras clave: Aguas residuales, sistema difuso, variables, función de pertenencia, lógica difusa, Mamdani, Sugeno, Tsukamoto, inferencia difusa.

Modeling a Fuzzy System for Water Saving in Car Wash Centers Using Mamdani, Sugeno, and Tsukamoto Models

Abstract. The car wash process involves a considerable consumption of resources such as water, automotive detergent, and washing time. With the aim of optimizing this process, this article presents the modeling of a fuzzy system for monitoring and controlling the car wash process, seeking to reduce washing time, water usage, and automotive detergent. The model is developed using the Matlab toolbox (Fuzzy Logic Toolbox),

where three input variables are employed: *car size*, *wash program*, and *dirt level*, each with their respective membership functions. There are also three output variables: *amount of water* (liters), *automotive detergent* (milliliters), and *washing time* (minutes). Subsequently, fuzzy inference rules are established using the Mamdani, Sugeno, and Tsukamoto methods to achieve the expected results, that is, a near-optimal amount of *water*, *automotive detergent*, and the required *washing time* for cleaning the vehicle.

Keywords: Wastewater, fuzzy system, variables, membership function, fuzzy logic, Mamdani, Sugeno, Tsukamoto, fuzzy inference.

1. Introducción

Las aguas residuales, generadas por actividades domésticas, industriales y agrícolas, representan un importante desafío ambiental. Estas aguas a menudo contienen una variedad de contaminantes, como productos químicos tóxicos, exceso de nutrientes y patógenos, los cuales pueden tener efectos adversarios en los ecosistemas acuáticos y representa riesgos para la salud si no se tratan adecuadamente. El rápido aumento de la población produce enormes cantidades de aguas residuales, contaminando los arroyos, estanques y embalses, el agua dulce se convierte como un recurso limitado y valioso. Los cambios climáticos, el agotamiento de los ecosistemas, el uso inapropiado del agua y las tensiones ecológicas están inseparablemente conectados con la reducción de los caudales debido a las bajas tasas de aguas subterráneas. Además, deteriora los embalses, lo que tiene un efecto perjudicial en el suministro y la disponibilidad de agua [5]. El declive del ecosistema acuático se ha convertido en un tema crucial que limita el desarrollo urbano, acelera la escasez de agua y afecta al bienestar de las personas [8]. El consumo excesivo y el desperdicio de agua en los lavados de autos es un problema significativo, ya que cada servicio genera una gran cantidad de agua residual mezclada con detergentes y otros productos químicos. Esta agua contaminada regresa al suelo. perjudicando los ecosistemas locales y daños medio ambientales graves. Para abordar esta problemática, se propone la implementación de tecnologías avanzadas como el sistema sifuso. Los sistemas difusos, basados en la inteligencia computacional, mejora los procesos de ahorro de agua, reduciendo el desperdicio y mejorando la eficiencia.

El propósito de este artículo es tomar una decisión aproximada a la óptima sobre la cantidad de agua, la cantidad de detergente automotriz y el tiempo necesarios para los centros de lavado. Para ello, se proponen tres modelos basados en lógica difusa (modelo Mamdani, modelo Sugeno y modelo Tsukamoto) para realizar los cálculos, comparar los resultados y elegir cuál modelo es el más óptimo para resolver el problema.

En los siguientes artículos se mencionan algunos trabajos que utilizan los modelos de Mamdani, Sugeno y Tsukamoto para tomar decisiones óptimas en la resolución de problemas.

En este trabajo de investigación presenta una gestión energética óptima para un sistema híbrido de bombeo de agua impulsado por un generador fotovoltaico (GVP) y una turbina eólica. Estas dos energías renovables se utilizan como fuentes de generación de energía, mientras que una batería se utiliza como sistema de almacenamiento para controlar el flujo de energía y proporcionar un suministro de carga constante. El sistema de gestión propuesto garantiza la autonomía del sistema de bombeo en una región rural sin acceso a la red eléctrica. Como resultado, se crea un controlador de seguimiento del punto de máxima potencia (MPPT) basado en el modelo difuso Takagi-Sugeno (TS), que garantiza la máxima transferencia de potencia a la motobomba a pesar de los cambios en la velocidad del viento y la insolación. La síntesis de la ley de control MPPT implica modelos de referencia difusos TS que generan las trayectorias deseadas para el seguimiento. Se ha desarrollado un supervisor para la gestión energética cuyo principal objetivo es utilizar eficazmente la batería para satisfacer los requisitos de carga de energía, manteniendo el estado de carga (SOC) para prolongar su vida útil. Finalmente, se han realizado resultados de simulación basados en Matlab/Simulink con el objetivo de validar la eficiencia del supervisor de gestión energética propuesto [2].

Este artículo presenta el primer paso para desarrollar un sistema difuso que represente el conocimiento adquirido de las partes interesadas y los responsables de la toma de decisiones, y luego lo transforme en reglas condicionales. Este sistema de inferencia de lógica difusa busca simular escenarios reales de toma de decisiones. En este artículo, se elige como ejemplo la evaluación del rendimiento del PCS en una ciudad. Se aplica un sistema de inferencia difusa Mamdani; las variables y reglas difusas son hipotéticas y se basan en encuestas realizadas en Qian'an, una de las ciudades piloto del PCS [7].

Este artículo propone y desarrolla la aplicación del método de inferencia difusa de Mamdani, Tsukamoto y Sugeno. El autor compara los resultados finales de cada método en un mismo caso. Este método se probará en un modelo de diseño para determinar o predecir el precio de exportación de un producto básico y obtener una ganancia óptima de sus ventas. Este modelo incluye la predicción del precio de reventa de los bienes exportados al exterior. El proceso de determinación del precio de venta se calcula utilizando el método de inferencia difusa de Mamdani, Tsukamoto y Sugeno.

Las variables a observar son la demanda de bienes, la disponibilidad de los mismos en los agricultores y el precio de mercado en el extranjero. El resultado es un precio de reventa estimado y la ganancia de cada producto de exportación. La comparación de los precios de venta utilizando Mamdani, Tsukamoto y Sugeno es del 2,1 %, mientras que Mamdani y Sugeno son solo del 1 %. Tras aplicar la negación lógica a las reglas básicas, esta se aplica al 1,6 %. Con esta comparación, el autor evaluará la sección de base de reglas de Tsukamoto. Los resultados obtenidos en este estudio incluyen el modelo de predicción del precio de reventa de bienes exportados y la comparación del rendimiento de la interferencia difusa mediante los métodos lógicos de Mamdani y Tsukamoto en la optimización para obtener un precio más óptimo [6].

2. Propuesta de solución al problema

Para abordar el problema del uso indiscriminado del agua, el detergente automotriz y el tiempo de lavado de los autos, se emplea la inteligencia computacional mediante un sistema difuso, utilizando un modelo en MatLab para los cálculos y la toma de decisiones en la aproximación óptima del uso de los recursos en los centros de lavado de autos. En esta propuesta se identifican las variables de entrada: *tamaño de auto, programa de lavado y suciedad*, las cuales se usarán para la fuzzificación a través de grados de pertenencia y las variables de salida serán los *cantidad de agua, la cantidad de detergente automotriz y el tiempo de lavado* necesarios para el lavado del auto, de acuerdo con los datos de entrada.

3. Objetivo

Proponer una solución a través de un modelado de un sistema difuso para solucionar el problema del desperdicio indiscriminado del agua, el detergente automotriz y el tiempo de lavado en los centros de lavados de autos utilizando. Para ello, se emplea la caja de herramientas de lógica difusa de Matlab.

4. Desarrollo de modelado de un sistema difuso en Matlab

4.1. Identificación de variables linguisticas de entrada y salida

Para crear un modelado de sistema de control difuso, primero se deben identificar las variables de entrada y salida. Para este proyecto, en los centros de lavado de autos se identificaron varias variables, como se muestra en la Tabla 1 y 2. En la Tabla 1 se muestran las variables lingüísticas de entrada para el uso eficiente del agua, con los términos lingüísticos y universo de discurso. En la Tabla 2 se muestran las variables lingüísticas de salida , que son necesarias para la defuzzificación de los datos.

Tabla 1. Identificación de las variables lingüísticas de entrada.

Variables lingüísticas	Conjuntos lingüísticos	Universo de discurso
Tamaño de auto	*Pequeño, Mediano, Grande*	20 - 50 m^2
Programa de lavado	*Básico, Estándar, Premium*	20 - 120 minutos
Suciedad	*Ligera, Media, Intensa*	0 - 100 %

4.2. Definición del universo de discurso del sistema difuso

La lógica difusa es una forma de modelar el razonamiento lógico donde la veracidad de una afirmación no es binaria, es decir, no es simplemente verdadera o

Tabla 2. Identificación de las variables lingüísticas de salida.

Variables lingüísticas	Conjuntos lingüísticos	Universo de discurso
Detergente automotriz	*Poca, Media, Intensa*	10 - 180 ml
Cantidad de agua	*Poca, Moderada, Mucha*	10 - 50 litros
Tiempo de lavado	*Lento, Normal, Rápido*	20 - 100 minutos

falsa como ocurre con la lógica clásica. En su lugar, es un grado de veracidad que va desde 0, que es absolutamente falso, hasta 1, que es absolutamente verdadero. La lógica difusa nos permite diseñar un sistema de inferencia difusa, que es una función que mapea un conjunto de entradas a salidas usando reglas interpretables por humanos en lugar de matemáticas más abstractas. El concepto de lógica difusa es muy común, está asociado con la manera en que las personas perciben el medio(entorno), [3].

El universo de discurso representa el rango completo de valores considerados para una variable en un sistema difuso. En la modelación de un sistema difuso para centros de lavado de autos, es necesario definir el universo de discurso para las siguientes variables, *Tamaño del auto, Programa de lavado y Cantidad de Suciedad*, el sistema debe ser capaz de medir y controlar el lavado de autos, como se muestra en la Tabla 1 y 2 en la columna ***Universo de discurso*** para las variables lingüísticas salida.

La recolección de datos se realizó mediante encuestas en 10 centros de lavado, en las cuales se preguntó la marca de los autos que reciben, el tiempo que tardan en lavarlos y el programa de lavado que ofrecen. Con base en la información recolectada, se definió el universo de discurso de cada variable de entrada, al igual que las variables de salida.

4.3. Funciones de pertenencia en la lógica difusa

En esta sección se muestran las funciones de pertenencia que son utilizadas para el modelado del sistema difuso, (1) la función triangular es una de las funciones de membresía para representar conjuntos difusos que se define con sus tres parámetros, a, b y c, (2) trapezoidal la función añade un tramo en el que la función mantiene un valor máximo. Los parámetros de la función de pertenencia, especificados como el vector a, b, c y d, (3) gaussiana parámetros de la función de pertenencia, especificados como el vector μ y c, μ donde es la desviación estándar y c es la media (el punto donde la función alcanza su valor máximo de 1), y (4) bell generalizada los parámetros de la función de pertenencia, especificados como el vector a, b y c:

$$f(x; a, b, c) = \text{máx}\left(\text{mín}\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \tag{1}$$

$$f(x; a, b, c) = \text{máx}\left(\text{mín}\left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b}\right), 0\right), \tag{2}$$

$$gaussmf(x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \tag{3}$$

$$f(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}. \tag{4}$$

4.4. Particionamiento de los universos de discurso de las variables de entrada

En el modelado se aplicaron las funciones de membresía triangular, trapezoidal, gaussiana y gbell para caracterizar los términos lingüísticos de las variables lingüísticas de entrada. El particionamiento se realizó sobre todos los valores del universo de discurso en el eje x del plano cartesiano, en la Figura 2 se muestran las variables de entrada *(a) tamaño de auto* con sus tres términos lingüísticos: *pequeño, mediano y grande*, donde se aplicó la función de membresía triangular mediante la ecuación (1) para cada uno de los términos. La siguiente variable particionada es el *(b) programa de lavado*, donde el universo de discurso se define en la Tabla 1, se utilizó la función de membresía *trapezoidal* para cada uno de los términos lingüísticos *pBásico, pEstándar y pPremium*, cubriendo todos los valores dentro del universo de discurso como se muestra en la Figura 2(b), se usó la ecuación (2) en los tres términos lingüísticos y la tercera variable lingüística es la cantidad de *(c)suciedad* en el auto, con los términos lingüísticos *ligera, media e intensa*, aquí se toma el porcentaje (%) en el *universo de discurso* para medir la suciedad presente en el auto, Tabla 1, en esta variable lingüística se usaron las funciones de membresía triangular (ligera) ecuación (1), trapezoidal (media) ecuación (2) y gaussiana (intensa) ecuación (3).

4.5. Grados de pertenencia de las variables de entrada

Zadeh propuso extender la noción de membresía binaria a membresía difusa en donde se pueden tener varios grados de membresía en el intervalo continuo real [0, 1], a diferencia de los conjuntos clásicos, en donde cada valor es representado en un conjunto como verdadero o falso, 0, 1, pertenece o no pertenece.

En la Tabla 3 se muestra el grado de pertenencia de cada valor para la variable *tamaño de auto*.

Cada valor tiene un grado de pertenencia en los términos lingüísticos *pequeño, mediano y grande*, dentro de un rango de 0 a 1. Para calcular el área total del auto se usó la siguiente ecuación:

$$\text{area total} = 2(L \times A + L \times W + A \times W). \tag{5}$$

Usando la ecuación (5) y como ejemplo la muestra n°1, con un *Tamaño de auto* 25 m^2, se realizaron los cálculos con la ecuación (5).

Las medidas de los autos se obtuvieron del sitio medidas de coches [1], es donde se encuentran registradas las dimensiones de los autos (L= longitud, A= altura, W= anchura(ancho)) de distintas marcas y modelos, aplicando la función de membresía *triangular* ecuación (1) se obtuvo un grado de pertenencia de 0 para los términos *pequeño y grande*, y en *mediano* se obtuvo un grado de

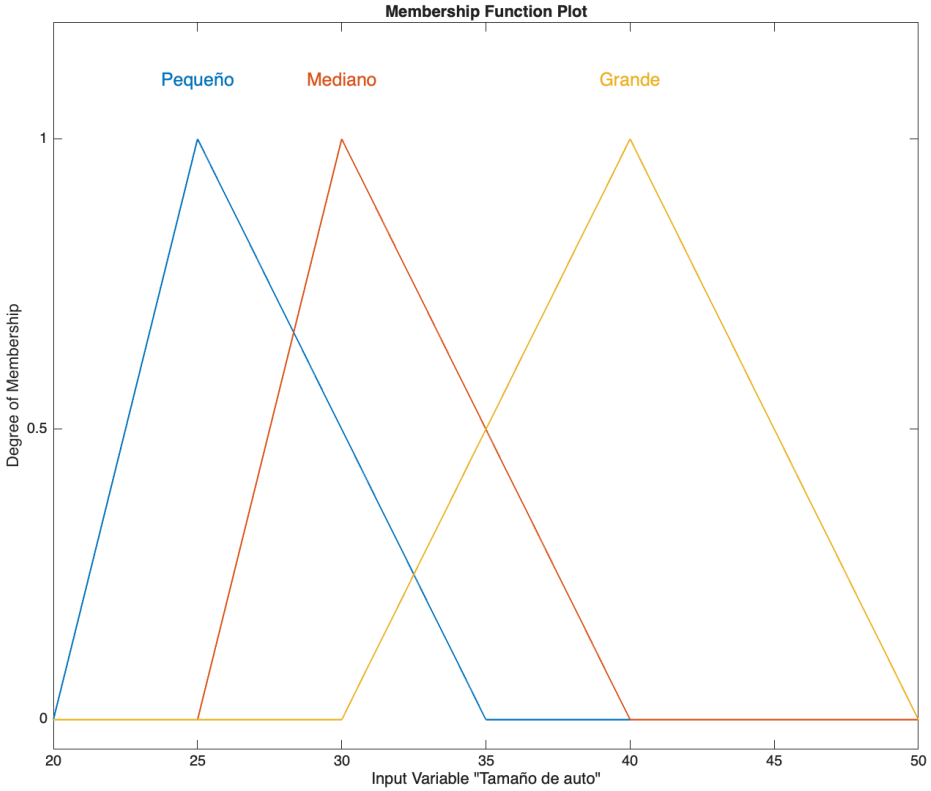

Fig. 1. Funciones de membresía que caracterizan los terminos linguisticos de las variables linguisticas: Tamaño de auto.

pertenencia de 1, esto indica que el auto pertenece completamente al término lingüístico *mediano*.

Se utilizaron un total de 50 muestras de autos de diferentes tamaños, de las cuales se seleccionaron algunas para ilustrar la metodología empleada para el desarrollo del modelo difuso aplicado a centros de lavado de autos.

Tabla 3. Grados de pertenencia de los elementos de tamaño de auto en m^2.

Muestra	Tamaño de auto m^2	Pequeño	Mediano	Grande
1	25	0	1	0
2	16	0.72	0.28	0
3	17	0.64	0.36	0
4	23	0.16	0.84	0
5	38	0	0	0.96

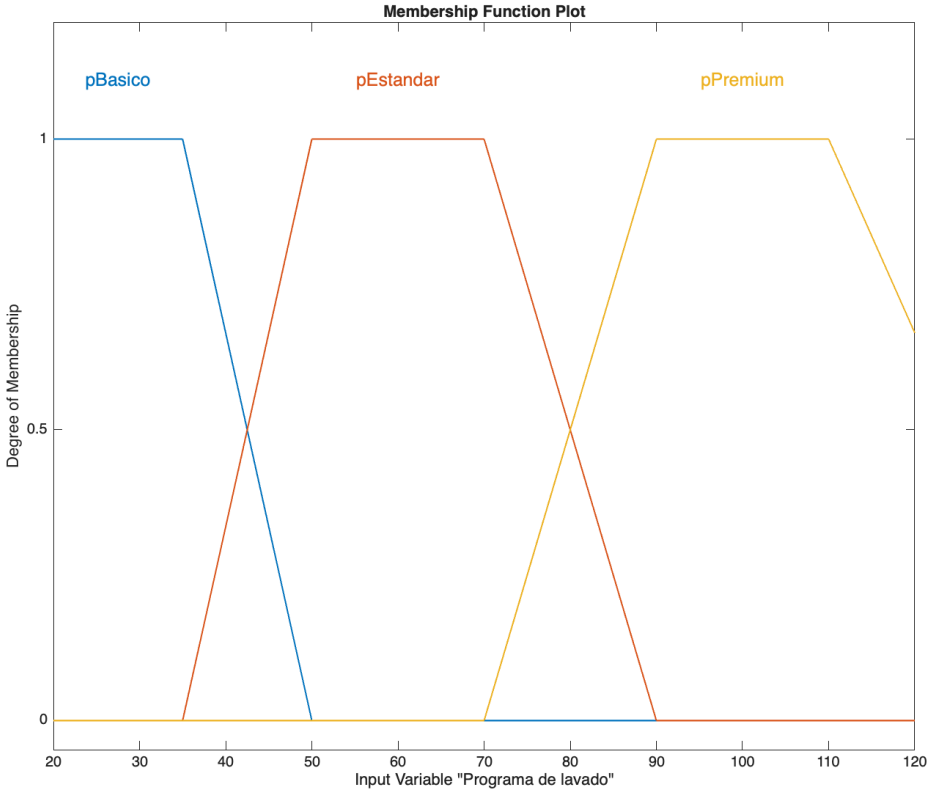

Fig. 2. Funciones de membresía que caracterizan los terminos linguisticos de las variables linguisticas: Programa de lavado.

4.6. Particionamiento de los universos de discurso de las variables de salida

En un sistema difuso en Matlab, la función de las salidas es generar respuestas o decisiones basadas en las variables de entrada, tales como el *tamaño de auto, el programa de lavado y la suciedad*. Estas variables de entrada se procesan mediante reglas difusas, y el sistema aplica dichas reglas para obtener un valor preciso o "crisp" como resultado, en un proceso conocido como defuzzificación. Este paso es esencial para generar valores de salida concreta, como la *(a)cantidad de agua*, cantidad de *(b)Detergente automotriz*, y el *(c)Tiempo de lavado* para los autos, como se muestra en la Figura 2.

4.7. Reglas de inferencia difusa

La inferencia difusa es una forma de inteligencia artificial, que permite a las computadoras imitar la forma en que lo humanos piensan y abordan la solución del problemas. En *tamaño de auto*, *pequeño* y *mediano*, son dos conceptos vagos, es difícil que una computadora determine el concepto de *pequeño* sin la lógica

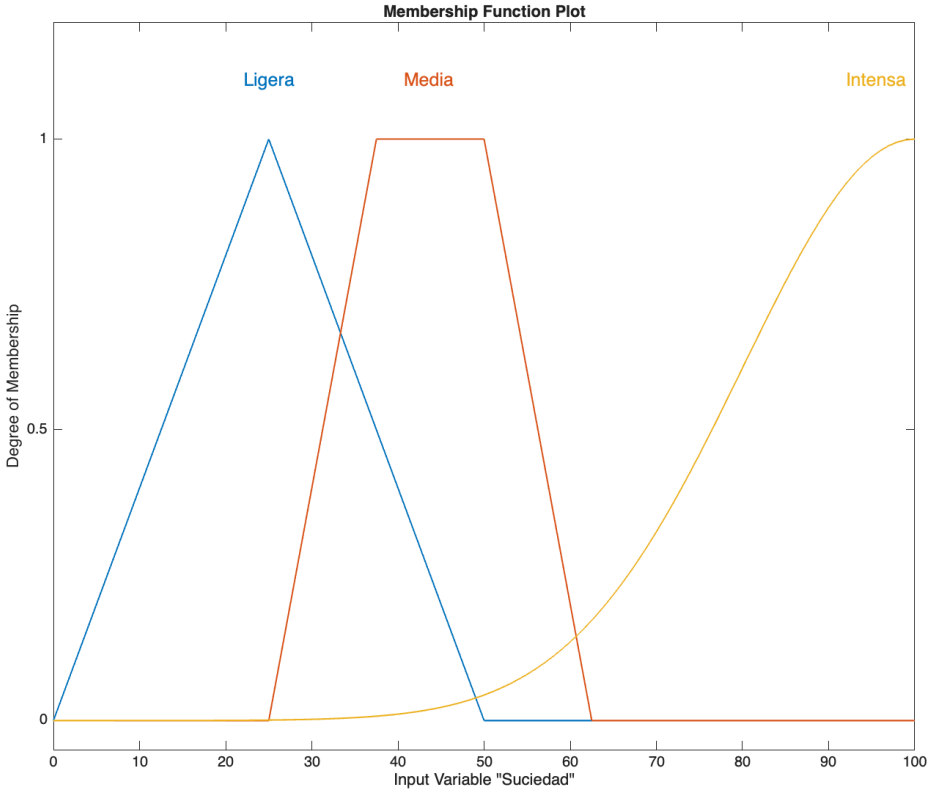

Fig. 3. Funciones de membresía que caracterizan los terminos linguisticos de las variables linguisticas: Suciedad.

difusa, que nos brinda una forma de codificar el conocimiento basado en la experiencia de manera que las computadoras puedan entenderlo en forma de reglas lógicas.

En Matlab, las reglas de inferencia difusa son el conjunto de reglas que define cómo las variables de entrada se traducen en salidas mediante el sistema difuso.

Estas reglas siguen una estructura de IF-THEN y se basan en las condiciones de las entradas y sus respectivos grados de pertenencia para generar una respuesta o una acción, como la cantidad aproximada al óptimo de agua y la cantidad de detergente automotriz para el lavado de autos.

En la Tabla 4 se muestran algunas de las reglas que se utilizan para la toma de decisiones en el modelado del sistema difuso para el ahorro de agua y detergente automotriz en los centros de lavado de autos, en este proyecto se crearon 42 reglas o combinaciones de las variables de entrada y las variables de salida.

El sistema Mamdani se utiliza en este modelado con el objetivo de obtener resultados cercanos al óptimo.

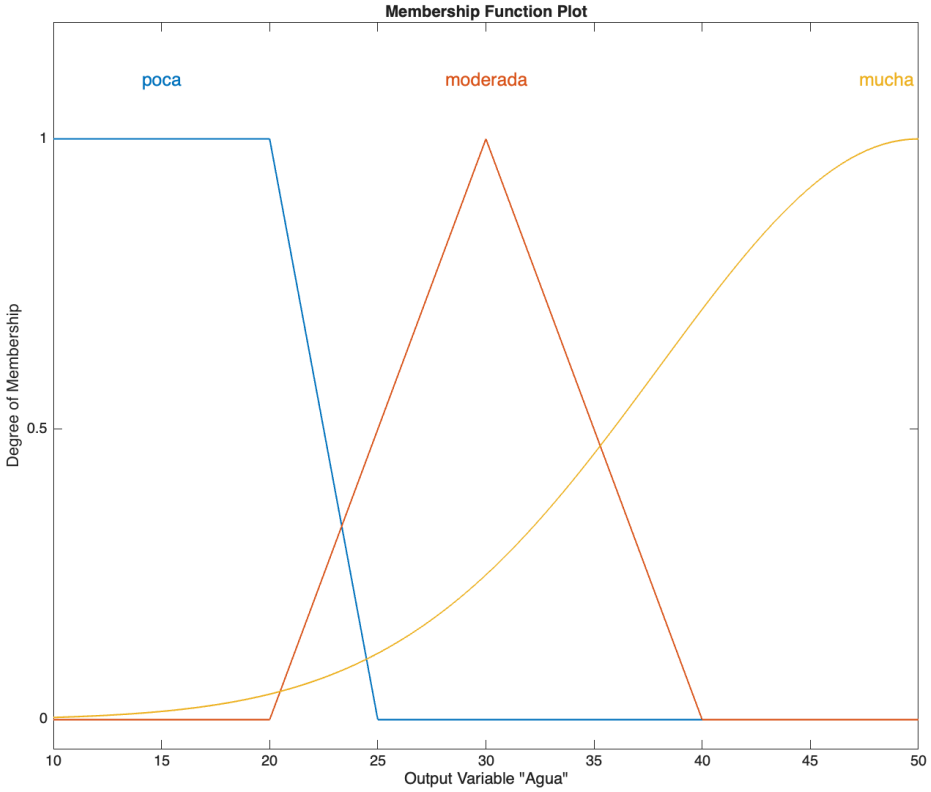

Fig. 4. Funciones de membresía que caracterizan los terminos linguisticos de las variables linguisticas: Cantidad de agua.

Este tipo de sistema es especialmente útil para resolver problemas de control y toma de decisiones en sistemas complejos que requieren un comportamiento similar al humano [4].

Tabla 4. Reglas de inferencia difusa.

N°	Regla
1	Si Tamaño de auto es Pequeño y Programa de lavado es pEstandar and Suciedad es Media, entonces Detergente automotriz es Media, Agua es moderada y Tiempo de lavado Normal
2	Si Tamaño de auto es Mediano y Programa de lavado es pEstandar y Suciedad es Media entonces Detergente automotriz es Media, Agua es moderada , Tiempo de lavado es Normal
3	Si Tamaño de auto es Pequeño y Programa de lavado es pEstandar y Suciedad es Ligera entonces Detergente automotriz es Poca, Agua es Poca , Tiempo de lavado es Rapido

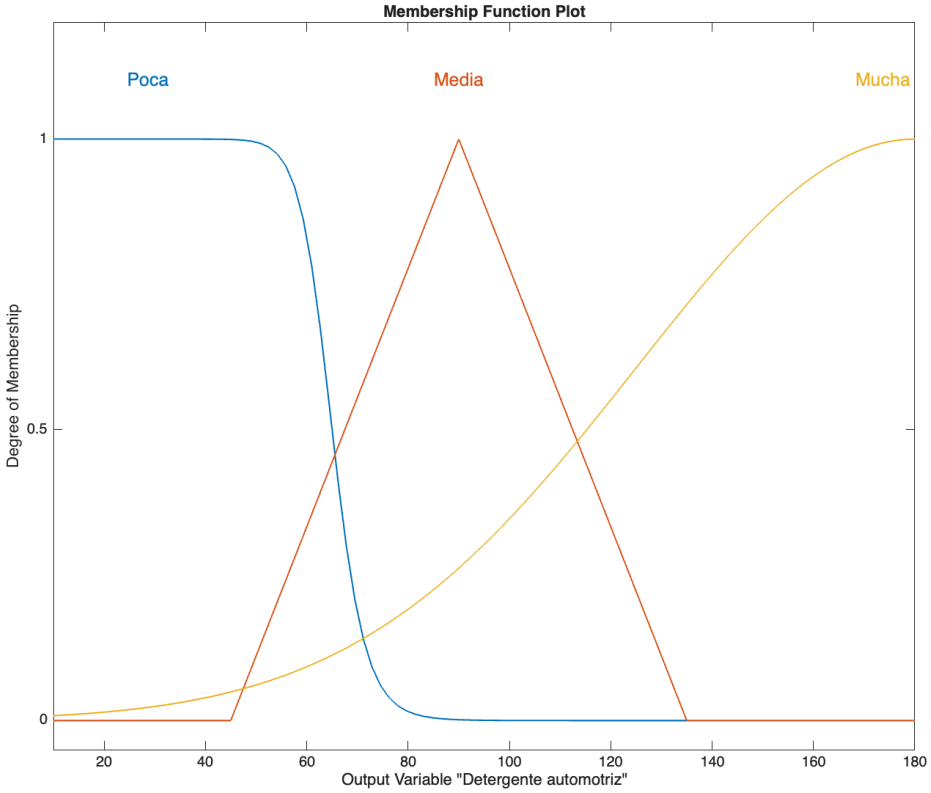

Fig. 5. Funciones de membresía que caracterizan los terminos linguisticos de las variables linguisticas: Detergente automotriz.

4.8. Defuzzificación

La difuzzificación nos dará un valor exacto de la inferencia difusa. Se refiere a la forma en que se extrae un valor nítido de un conjunto difuso como valor representativo. En este modelo se utiliza la ecuación (6) para calcular el centroide de las reglas activas, en el algoritmo 1 se explica a detallle la secuencia del modelo Mamdani, con centroide z_{COA}:

$$z_{\mathrm{COA}} = \frac{\int_Z \mu_A(z) z \, dz}{\int_Z \mu_A(z) \, dz}. \tag{6}$$

4.9. Modelo Mamdani

El modelo Mamdani ofrece un número mínimo de características, un número reducido de reglas difusas y una mayor precisión [9].

En el Algoritmo 1, correspondiente al modelo difuso de Mamdani, se explica en detalle el funcionamiento del modelado para obtener el valor crisp o concreto, es decir, el valor que necesitamos para las variables de salida del sistema difuso.

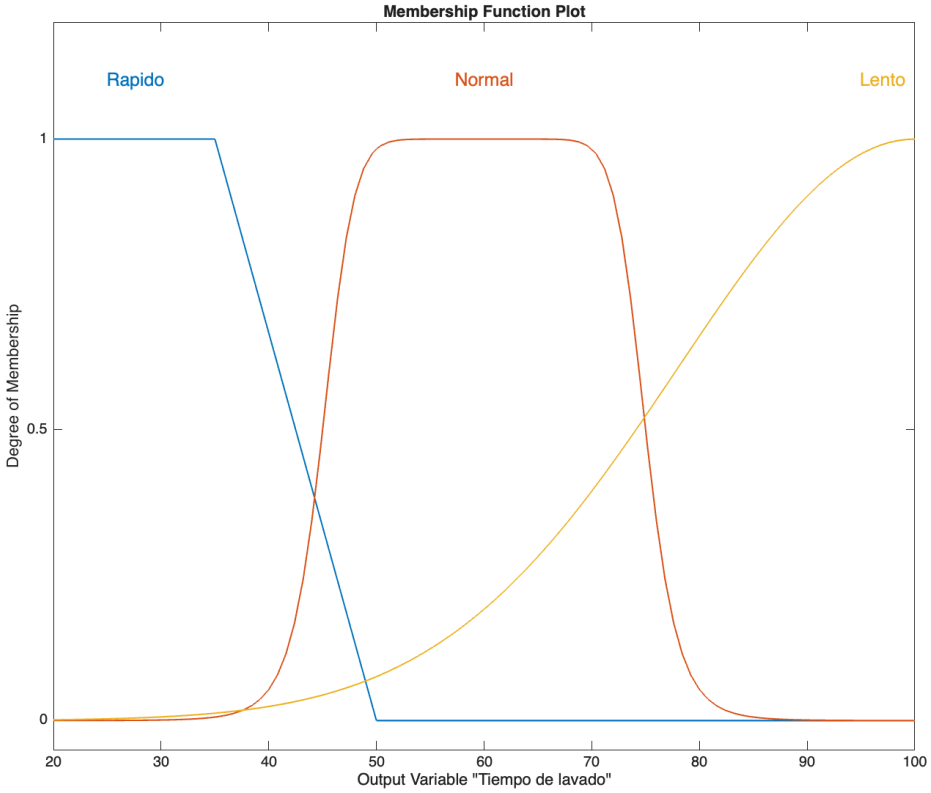

Fig. 6. Funciones de membresía que caracterizan los terminos linguisticos de las variables linguisticas: Tiempo de lavado.

Las variable son la cantidad de agua, cantidad de detergente automotriz y el tiempo necesario para lavar los autos.

Algoritmo 1 Modelo difuso Mamdani

Entradas: tamaño_auto, nivel_suciedad, tipo_programa
Salidas: cantidad_agua, cantidad_detergente, tiempo_lavado
1. Fuzzificación: transformar entradas reales a valores difusos (pequeño/mediano/grande; ligera/media/intensa; básico/estándar/premium).
2. Inferencia Difusa:
for all reglas **do**
    Calcular activación: $w_i = \text{mín}(\mu_{\text{tamaño}}, \mu_{\text{suciedad}}, \mu_{\text{programa}})$
    Obtener función salida: $\mu_{\text{salida}_i}(z) = \text{mín}(w_i, \mu_{\text{salida}_i}(z))$
end for
Agregar con OR: $\mu_{\text{agregada}}(z) = \text{máx}_i(\mu_{\text{salida}_i}(z))$
3. Desfuzzificación:
Salida nítida: Salida $= \frac{\int z \cdot \mu_{\text{agregada}}(z)\, dz}{\int \mu_{\text{agregada}}(z)\, dz}$
return cantidad_agua, **cantidad_detergente**, **tiempo_lavado**

La ZCOA asocia el centro del área formada por el número difuso, donde μA(z) es la funcion de pertenencia del conjunto de salida, cuya variable es z es el dominio o rango de integracion.

En la Figura 3 se muestran los valores de entrada para *Tamaño de auto*, *Programa de lavado* y *Suciedad* en **Input values** (40 40 60), estos valores tras pasar por los procesos de *Fuzzificación*, *Inferencia* y *Difuzzificación*, se llegó a la conclusión de que para aproximar a la optimización del agua , *detergente automotriz* y el *tiempo de lavado* se necesitan en *detergente automotriz* 90 ml, *Cantidad de agua* 25 litros de agua y *Tiempo de lavado* 50 minutos para lavar un auto de 40 m^2. En este modelado se emplea la ecuación (5) para hacer la defuzzificación de los datos utilizando el grado de pertenencia de las reglas que se activan.

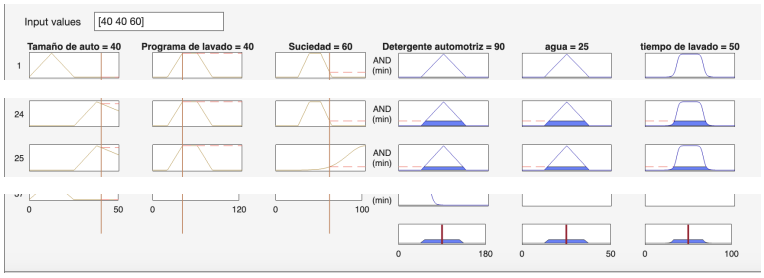

Fig. 7. Presentación de la Reglas expresadas con las funciones de pertenencia y del conjunto difuso a defusificar.

4.10. Modelo Sugeno

El modelo difuso de Sugeno (conocido como modelo difuso TSK), tiene el objetivo de desarrollar un enfoque sistemático para generar reglas difusas a partir de un conjunto de datos de entrada y salida dado. Una regla difusa Sugeno tiene la forma:

si x es A y y es B entonces $z = f(x,y)$, donde A y B son conjuntos difusos en el antecedente. mientras que $z = f(x,y)$ es una función precisa en el consecuente, $z = f(x,y)$ es un polinomio en las variables de entrada x y y, Cuando f es una constante, se obtiene un modelo difuso de Sugeno de orden cero y cuando $f(x,y)$ es un polinomio de primer orden, el sistema de inferencia difusa resultante se denomina modelo difuso de Sugeno de primer orden, en el Algoritmo 2 se explica a detalle su funcionamiento. En este proyecto también se utilizó el modelo Sugeno para comparar los valores de salida con las mismas variables de entrada utilizadas con el modelo Mamdani, como *tamaño de auto*, *programa de lavado* y la *suciedad* de los autos y las variables de salida, *detergente automotriz*, cantidad de *agua* y el *tiempo de lavado*.

Algoritmo 2 Modelo difuso Sugeno

Entradas: tamaño_auto, nivel_suciedad, tipo_programa
Salidas: cantidad_agua, cantidad_detergente, tiempo_lavado
1. Fuzzificación: convertir entradas reales a valores difusos (pequeño/mediano/grande; ligera/media/intensa; básico/estándar/premium).
2. Inferencia Difusa:
for all reglas **do**
    Grado de activación: $w_i = \text{mín}(\mu_{\text{tamaño}}, \mu_{\text{suciedad}}, \mu_{\text{programa}})$
    Salida crisp de regla: $z_i = a_i x + b_i y + c_i z + d_i$
end for
3. Defuzzificación: salida nítida por promedio ponderado:

$$\text{Salida} = \frac{\sum w_i z_i}{\sum w_i}$$

return cantidad_agua, **cantidad_detergente**, **tiempo_lavado**

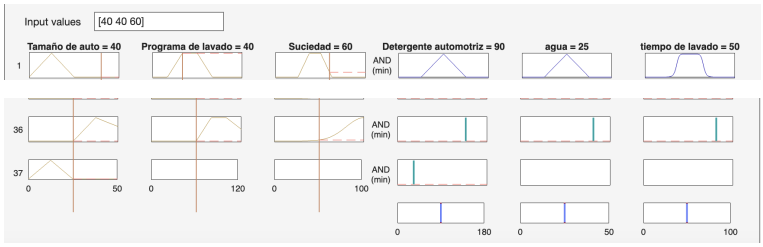

Fig. 8. Valores de entrada de las variables linguisticas, de la tabla 1, tabla 2 y tabla 3, usando modelo Sugeno.

En el algoritmo 2 muestra el funcionamiento del modelo Sugeno de primer orden utilizando las variables *Tamaño de auto, Programa de lavado y Suciedad* del auto. Para obtener un valor concreto, se consideran únicamente las reglas activadas según los valores de entrada (Figura 4). Además, para cada variable lingüística, se toman en cuenta los grados de pertenencia de los conjuntos difusos, los cuales se emplean en las operaciones del modelo Sugeno.

4.11. Modelo Tsukamoto

Otro de los modelos es el Tsukamoto, en los modelos difusos de Tsukamoto, la consecuencia de cada regla difusa de tipo "si-entonces" se representa mediante un conjunto difuso con una función de membresía monótona, como se muestra en el algoritmo 3. Como resultado, la salida inferida de cada regla se define como un valor nítido inducido por la salida de la regla. Dado que cada regla genera una salida nítida, el modelo difuso de Tsukamoto agrega las salidas de las reglas mediante el método de promedio ponderado, evitando así el proceso de defuzzificación, que suele ser más costoso en términos computacionales, [4].

Algoritmo 3 Modelo Difuso Tsukamoto

Entradas: tamaño_auto, nivel_suciedad, tipo_programa
Salidas: cantidad_agua, cantidad_detergente, tiempo_lavado
1. Fuzzificación: convertir entradas reales en valores difusos: (pequeño/mediano/grande; ligera/media/intensa; básico/estándar/premium).
2. Inferencia Difusa:
for all reglas **do**
    Calcular el grado de activación: $w_i = \text{mín}(\mu_{\text{tamaño}}, \mu_{\text{suciedad}}, \mu_{\text{programa}})$
    Obtener salida nítida z_i resolviendo la inversa de $\mu_{\text{salida}}(z)$ tal que $\mu_{\text{salida}}(z_i) = w_i$ con μ monótona.
end for
3. Defuzzificación: calcular salida crisp con promedio ponderado:

$$\text{Salida} = \frac{\sum w_i z_i}{\sum w_i}$$

return cantidad_agua, **cantidad_detergente**, **tiempo_lavado**

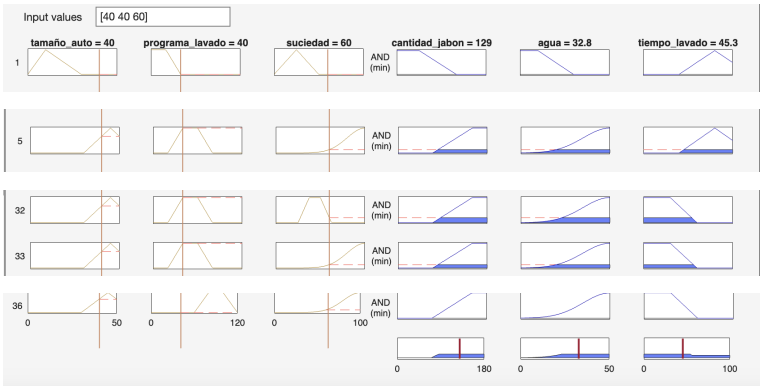

Fig. 9. Presentación de las reglas expresadas con las funciones de pertenencia y del conjunto difuso a defuzificar.

Aplicando el modelo Tsukamoto, se utilizaron las mismas variables *Tamaño de auto, Programa de lavado y Suciedad* del auto para obtener los valores esperados.

5. Experimentación e interpretación del resultados

Los resultados muestran cómo cada modelo difuso(Mamdani, Sugeno y Tsukamoto) procesa las mismas entradas para generar salidas correspondientes en variables como la *cantidad de agua*(c. de agua), *detergente automotriz*(d. automotriz) y *tiempo de lavado*(t. de lavado), como se muestra en la Tabla 5, 6 y 7. El modelo Mamdani muestra resultados coherentes con las entradas. Por

ejemplo, cuando la *Suciedad* es alta y el *Programa de lavado* es elevada(como la fila 2 de la Tabla 4), el consumo de agua y el tiempo de lavado son mayores.

Tabla 5. Experimentos con valores de las variables linguisticas con el modelo Mamdani.

T. de auto	p. de lavado	suciedad	d. automotriz	c. de agua	t. de lavado
30	85	80	118	31.4	45.6
45	100	100	134	40	79.9
45	100	30	90	25	50
25	30	100	90	25	50
40	90	100	91.7	28	61.4
20	80	50	69.8	25	47.9
50	100	100	131	39.1	78.6

En el modelo Sugeno, los resultados son más precisos, ya que las salidas se generan mediante funciones lineales o constantes. Se observa que el *cantidad de agua*(c. de agua) y el *tiempo de lavado*(t. de lavado) tienden a ser más alto en comparación con Mamdani refleja una respuesta más optimizada y matemática. Tomando en cuenta la misma posición (fila 2 de la tabla 5) *tamaño de auto* 45(t. de auto), *Programa de lavado* 100 (P.de lavado) y *Suciedad* del auto 100, nos dá como resultado *detergente automotriz* (d. automotriz) 137 mililitros y el *tiempo de lavado* también es más elevado(81.2 minutos).

Tabla 6. Experimentos con valores de las variables linguisticas con el modelo Sugeno.

T. de auto	p. de lavado	suciedad	d. automotriz	c. de agua	t. de lavado
30	85	80	128	34.7	63.7
45	100	100	137	40.6	81.2
45	100	30	90.4	25.1	50.3
25	30	100	90	25	50
40	90	100	99	28	56
20	80	50	79.1	19.8	38.9
50	100	100	137	40.6	81.2

El modelo Tsukamoto presenta una mayor variabilidad en los resultados, especialmente en situaciones donde los valores de entrada son bajos, en el caso de *tamaño de auto*(t. de auto) 25 y *programa de lavado*(p. de lavado) 30. La cantidad de *detergente automotriz*(d. automotriz) es significativamente menor(50.8), lo que refleja cómo las funciones de membresía monótonas afectan la salida y se observa que el *tiempo de lavado*(t. de lavado) es más bajo en algunos casos, lo que puede indicar una mayor sensibilidad del modelo a las condiciones de entrada, como se muestra en la Tabla 7.

Tabla 7. Experimentos con valores de las variables linguisticas con el modelo Tsukamoto.

T. de auto	p. de lavado	suciedad	d. automotriz	c. de agua	t. de lavado
30	85	80	136	36.6	25.5
45	100	100	142	38.2	46.2
45	100	30	90.4	25.1	50.3
25	30	100	50.8	12.5	49.9
40	90	100	137	36.9	49.3
20	80	50	55.8	14.4	26.2
50	100	100	137	36.9	49.3

Como parte de la experimentación, se utilizaron 50 muestras de autos de diferentes tamaños. Se aplicaron los modelos Mamdani, Sugeno y Tsukamoto para comparar los valores de las variables de salida. Con base en los resultados, se determina cuál modelo es el más adecuado para cumplir con el objetivo del artículo. En las Tablas 5, 6 y 7 se presentan únicamente 7 de las 50 muestras, con el fin de no exceder el número de páginas y facilitar la comprensión de los resultados. Las muestras seleccionadas son representativas de los diferentes niveles de entrada, permitiendo observar el comportamiento de cada modelo en distintos escenarios.

En las Tablas 8, 9 y 10 se muestra el comportamiento de los valores de salida para una muestra de 50 autos. Utilizando estadística descriptiva, se observa que el tamaño promedio de los autos es de 35.5, es decir, los autos que más se lavan tienen un tamaño aproximado de $35.38m^2$, con una desviación estándar de 9.160. Aplicando la fórmula de dispersión: 35.38 ± 9.160, se obtiene un rango de 26.22 a $44.54m^2$, lo que representa los tamaños de autos más comunes dentro del conjunto difuso de autos medianos.

En cuanto a la cantidad de agua utilizada para lavar un auto, se observa un promedio de 31.448 litros. Aplicando la desviación estándar, se determina que el rango de consumo oscila entre 22.271 y 40.625 litros. En conclusión, para lavar autos con tamaños entre 26.22 y 44.54 m^2, se requieren entre 22.271 y 40.625 litros de agua, aplicando el modelo Mamdani.

En el modelo Sugeno (Tabla 9), no hay diferencia en el tamaño del auto ya que se utilizaron las mismas muestras en cada modelo. Sin embargo, en cuanto a la cantidad de agua, se presenta una diferencia de 0.484 litros respecto al modelo Mamdani. Por su parte, el modelo Tsukamoto (Tabla 10) muestra una diferencia más significativa, de 5.398 litros.

Al comparar estos valores con los resultados de las encuestas realizadas en centros de lavado de autos, donde el consumo promedio oscila entre 20 y 80 litros por auto, se concluye que los modelos difusos permiten una estimación más eficiente del consumo de agua, lo cual contribuye al ahorro de este recurso.

Tabla 8. Estadística descriptiva de variables entrada/salida de modelo Mamdani.

	T. de auto	p. de lavado	suciedad	d. automotriz	c. de agua	t. de lavado
Promedio	35.38	79.96	57.08	94.92	31.448	57.524
Mediana	34	86.5	54	91.65	30	60
Moda	30	70	100	90	30	60
Rango	30	95	100	93.8	64.6	78.5
Desviación σ	9.160	28.369	30.087	23.819	9.177	13.191

Tabla 9. Estadística descriptiva de variables entrada/salida de modelo Sugeno.

	T. de auto	p.de lavado	suciedad	d. automotriz	c. de agua	t. de lavado
Promedio	35.38	79.95	57.68	98.744	30.964	61.138
Mediano	34	86.5	54	97	30.3	60.3
Moda	30	70	100	137	30	60
Rango	30	95	100	90.5	24.4	51.2
Desviación σ	9.160	28.369	30.087	25.559	6.277566472	12.8094335

Tabla 10. Estadística descriptiva de variables entrada/salida de modelo Tsukamoto.

	T. de auto	p. de lavado	suciedad	d. automotriz	c. de agua	t. de lavado
Promedio	35.38	79.96	57.08	93.968	26.05	40.494
Mediana	34	80.5	54	90.05	25.65	46.75
Moda	30	70	100	13.5	25	50
Rango	30	95	100	94.8	24.3	39.3
Desviación σ	9.160	28.369	30.087	36.444	9.486	10.305

6. Conclusión y trabajo a futuro

La lógica difusa puede aplicarse a diversos procesos de toma de decisiones computacionales, con el objetivo de aproximar de forma óptima el uso de recursos naturales o químicos en distintas aplicaciones. Considerando los valores de entrada presentados en las Tablas 5, 6 y 7 ([45, 100, 100]), cada modelo difuso tiene su propia forma de difusificar los conjuntos. Como se observa en los resultados de las variables de salida, los modelos de Mamdani y Sugeno presentan variaciones mínimas entre sí, mientras que el modelo de Tsukamoto muestra diferencias significativas, especialmente en el tiempo estimado de lavado.

En este artículo, se modela una parte del sistema difuso con el propósito de optimizar el consumo de agua, detergente automotriz y tiempo de lavado. Como trabajo futuro, se contempla el desarrollo de un nuevo modelo enfocado en la recuperación de agua residual, con el fin de permitir su reutilización y reducir el desperdicio generado en cada proceso de lavado. Todo este modelado estará orientado a formar parte del componente de Inteligencia dentro del enfoque del Internet de las Cosas (IoT).

Este proyecto se encuentra en desarrollo. Se recomienda realizar más experimentos con una mayor cantidad de datos y ampliar el número de reglas difusas. En este artículo se utilizaron 42 reglas; sin embargo, se sugiere agregar más con el fin de obtener resultados más cercanos al valor óptimo, lo cual contribuiría a una mejor toma de decisiones en las variables de salida utilizando los modelos Mamdani, Sugeno y Tsukamoto.

Referencias

1. Medidas de coches (2025)
2. Ben Safia, Z., Allouche, M., Chaabane, M.: Renewable energy management of an hybrid water pumping system (photovoltaic/wind/battery) based on takagi–sugeno fuzzy model. Optimal Control Applications and Methods, vol. 44, no. 2, pp. 373–390 (2023) doi: https://doi.org/10.1002/oca.2884 https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2884
3. Douglas, B.: Matlab & simulink/fuzzy logic (2021)
4. Jang, J.-S. R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Upper Saddle River, NJ (1997)
5. Mortazavi-Naeini, M., Bussi, G., Elliott, J. A., Hall, J. W., Whitehead, P. G.: Assessment of risks to public water supply from low flows and harmful water quality in a changing climate. Water Resources Research, vol. 55, no. 11, pp. 8898–8915 (2019) doi: 10.1029/2019WR025123. First published: 24 October 2019
6. Napitupulu, S., Nababan, E. B., Sihombing, P.: Comparative analysis of fuzzy inference tsukamoto mamdani and sugeno in the horticulture export selling price. In: 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), pp. 183–187 (2020) doi: 10.1109/MECnIT48290.2020.9166587
7. Wang, C., Li, Y., Du, J., Corzo, G.: Mamdani fuzzy inference system for rating the performance of sponge city programme. In: 2022 8th International Conference on Systems and Informatics (ICSAI), pp. 1–6 (2022) doi: 10.1109/ICSAI57119.2022.10005410
8. Wei, F., Zhang, X., Xu, J., Bing, J., Pan, G.: Simulation of water resource allocation for sustainable urban development: An integrated optimization approach (2020)
9. Weihong, Z., Shunqing, X., Ting, M.: A fuzzy classifier based on mamdani fuzzy logic system and genetic algorithm. In: 2010 IEEE Youth Conference on Information, Computing and Telecommunications, pp. 198–201 (2010) doi: 10.1109/YCICT.2010.5713079

ISSN 1870-4069

Differential Evolution for Feature Selection: A Systematic Literature Review

Francisco Javier Hernández-Somohano[1], Luz Ivana Correa-Hernández[2], Jesús Arnulfo Barradas-Palmeros[2], Hector Gabriel Acosta-Mesa[2], Efrén Mezura-Montes[2]

[1] Universidad Veracruzana, School of Statistics and Informatics, Mexico

[2] Universidad Veracruzana, School of administration and accounting, Mexico

[3] Universidad Veracruzana, Artificial Intelligence Research Institute, Mexico

zS21013274@estudiantes.uv.mx, zS23000652@estudiantes.uv.mx, heacosta@uv.mx, emezura@uv.mx, zs24010958@estudiantes.uv.mx

Abstract. Feature selection (FS) is an important task in data processing and analysis, aiming to reduce dimensionality and improve the performance of machine learning algorithms such as classification algorithms. Differential evolution (DE) has been successfully used for this purpose. However, a comprehensive assessment of their comparative strengths and weaknesses remains absent. In this systematic review of the literature that analyzes the state-of-the-art of DE for FS, 25 studies were selected for the review. Among the three evaluation criteria approaches (in this study, wrapper, filter, and hybrid approaches), most studies used a wrapper approach, with the k-nearest neighbors (KNN) algorithm being the most implemented. Considering how individuals are encoded, three representations were identified: real-number vectors, binary vectors, and integer-number vectors, with real-number vectors being the most used in DE for feature selection. It was found that most of the works follow a single-objective optimization process, and only a minority uses a multi-objective approach. Finally, for the main field of application, most studies focus on classification tasks using repository datasets from UCI Machine Learning Repository. This research aims to provide new insights into the state-of-the-art DE for FS.

Keywords: Feature selection, differential evolution.

1 Introduction

In various fields, feature selection (FS) plays a crucial role in reducing the dimensionality of datasets. The goal is to select the smallest and most relevant

subset of features. The latter improves the interpretability of the data, accelerates model learning, simplifies them, and improves their performance in tasks such as classification [29]. FS becomes a complex problem due to its ample search space where the number of solutions is 2^n for a dataset with n features, as mentioned in [7].

Evolutionary computing (EC) techniques are well known for their ability to perform global optimization, including genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), Artificial Bee Colony (ABC), Forest Optimization Algorithm (FOA), etc. Differential evolution (DE), proposed by Storn and Price in 1997 [23], is a recent and powerful metaheuristic approach that converges quickly and accurately. DE requires few control parameters, is robust, and is easier to use than other global optimization methods. Due to these advantages, differential evolution has been adopted by the FS community and has been successfully implemented in various studies, for example, [5,6], and [22]; however, works on DE for FS are much less than others EC techniques, such as GA and PSO.

Some studies, such as [2], propose a method for DE focused on FS, which consists mainly of four steps: initialization, mutation, crossover, and selection. Possible solutions are generated during initialization, where each solution, called a target vector, is encoded to represent a potential feature subset. After that, the evolutionary process starts, where each iteration is called a generation. In the mutation, a mutant vector is generated for each target vector. Then, the mutant vector is combined with the target vector in the crossover step, generating the trial vector. The target and trial vectors are compared in the selection step, and the one with the highest fitness is maintained in the population. The process is repeated until a stop criterion is met and the solution with the highest fitness in the population (single-objective) or a set of solutions (multi-objective) is returned.

According to [29], FS algorithms are generally classified into two categories: wrapper and filter. However, some studies have combined these two approaches, so a third category called "hybrid" was introduced. These criteria are applied in the selection step to evaluate the potential feature subsets. Wrapper approaches employ a machine learning algorithm, such as a classifier, to assess how well the subset performs within the algorithm. This approach is usually the most computationally expensive but usually gets better performance. Filter approaches evaluate subsets using statistical or theoretical measures to assess feature relevance. While computationally less expensive, this method is less accurate than a wrapper because it does not use a machine learning algorithm in the search process. Hybrid approaches integrate filter and wrapper measures for evaluation.

During the search process step, the goal is to find the optimal subset or subsets that achieve the best performance. However, there are different approaches to address this process. As discussed in [29], some studies combine classification performance and the number of selected features into an aggregate objective function, following a single-objective optimization approach. On the

other hand, some studies propose a multi-objective optimization process, where two or more conflicting objectives are optimized simultaneously. Multi-objective approaches often maximize classification performance and minimize the number of selected features.

Diverse works can be found in the literature reviewing Evolutionary Computation and Bio-Inspired methods for FS, such as [1] and [14]. Nonetheless, none of them are focused only on DE. This study aims to analyze the state-of-the-art differential evolution for feature selection, focusing on various characteristics, their popularity, and applications to guide future research.

The rest of this paper is organized into five sections. Section 2 describes the method used, including the research questions, search strategy, and the inclusion and exclusion criteria process. Section 3 presents the results of the selected studies where the proposed method was applied. Section 4 discusses the findings. Finally, Section 5 concludes the study, summarizing the main findings and potential future research.

2 Method

The method used to conduct the systematic literature review is the one proposed by Kitchenham [15]. This method is carried out in three phases: 1) Planning, 2) Conducting the review. 3) Documentation of the review.

This section describes the research questions, the search strategy, and the study selection process.

2.1 Research Questions (RQ)

The research questions formulated to guide the review are:

- **RQ1.** What subset/individual evaluation approaches are used in DE-based algorithms for FS?
- **RQ2.** What representations of solutions are used in DE for FS algorithms?
- **RQ3.** What type of optimization (single-objective or multi-objective) is implemented in DE algorithms for FS?
- **RQ4.** In which applications or domains are DE-based algorithms used for FS?

2.2 Search Strategy

A preliminary search of articles on the topic was conducted to understand it better and formulate an appropriate search string.

Search String A search string consists of keywords related to the study topic. After testing different strings based on the number of found studies, the selected search string was:

```
("Differential Evolution") AND ("feature selection")
```

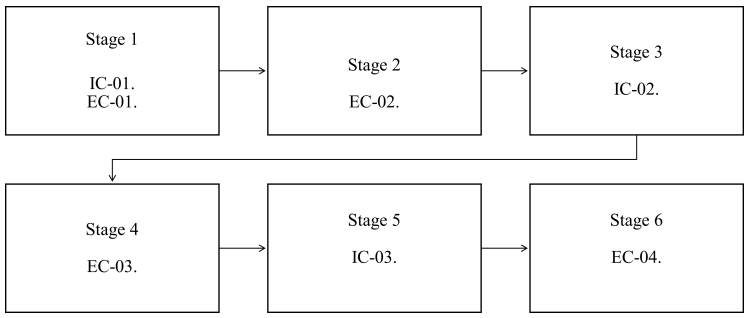

Fig. 1. Search Diagram: Inclusion and Exclusion Criteria by Phases.

Source Selection Initially, five sources were proposed (IEEE, ACM, SpringerLink, ScienceDirect, and Willey). However, Willey was discarded because no results were found during the period specified later. Finally, the search string was applied to the four remaining sources.

2.3 Study Selection

The study selection process was carried out in six phases. In the first phase, initial inclusion criteria (IC) and exclusion criteria (EC) were applied, while in the following phases, inclusion and exclusion criteria were alternated, as shown in Figure 1.

Inclusion Criteria

- **IC-01.** Studies must have been published between 2019 and 2024.
- **IC-02.** Titles must contain the terms *"Differential Evolution"* and *"Feature Selection"* to ensure topic relevance.
- **IC-03.** Studies must address at least two of the research questions in their abstract.

Exclusion Criteria

- **EC-01.** Studies written in a language other than English are excluded.
- **EC-02.** Studies that do not belong to the following categories are excluded: surveys, research articles, review articles, journals, or conference papers.
- **EC-03.** Duplicate studies found in the search are removed.
- **EC-04.** Studies that do not exclusively use differential evolution-based algorithms for feature selection are excluded.

2.4 Threats to Validity

It is important to acknowledge certain limitations that may affect the validity of this systematic review. First, our temporal scope (2019-2024) captures

only recent developments in DE for FS, potentially missing foundational work published before 2019 as well as emerging research published after our cutoff date in 2024. Second, by restricting our search to four academic databases (ScienceDirect, SpringerLink, IEEE, and ACM), we may have overlooked relevant studies published in other repositories or specialized venues. Finally, our methodology lacks quantitative measures to evaluate the quality of selected studies or assess the comprehensiveness of our search string, which limits our ability to objectively evaluate the completeness of this review. Despite these limitations, we believe our findings provide valuable insights into current trends and characteristics of DE methods for FS, while acknowledging that a more comprehensive analysis could be performed in future research.

3 Results

This section describes the selected studies, their characteristics, and the answers to the research questions.

3.1 Study Selection

After applying the method mentioned in the selected sources, 25 studies were obtained.

DE has recently been implemented in FS, which should be considered when evaluating the number of studies on this topic. Despite the reduction in the number of studies during the first two phases, a sufficient number of studies were collected.

3.2 Study Characteristics

Publication Sources Of the 25 selected studies from the four consulted sources, a similar number of studies were found in all four sources: 7 in ACM, 9 in IEEE, 8 in ScienceDirect, and 7 in SpringerLink, indicating that the topic is present similarly in the selected sources.

Additionally, 52% (13 studies) were published in journals, while the remaining 48% (12 studies) were presented at conferences. This balanced distribution indicates that the topic has been explored both in journal articles and conference communications, reflecting sustained research interest.

Publication Years Within the study period, Figure 2 shows that the years with the highest number of published studies were 2020 (4 studies), 2023 (9 studies), and 2024 (5 studies). Contrarily, the years with the fewest published studies were 2019 and 2022, with 2 studies. Although studies on the topic were published throughout the period, the number of publications increased in only three years (from 2020 to 2023), suggesting a growing interest in applying DE to FS.

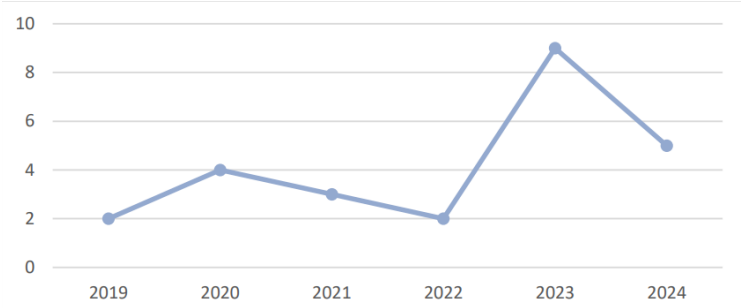

Fig. 2. Number of studies published per year.

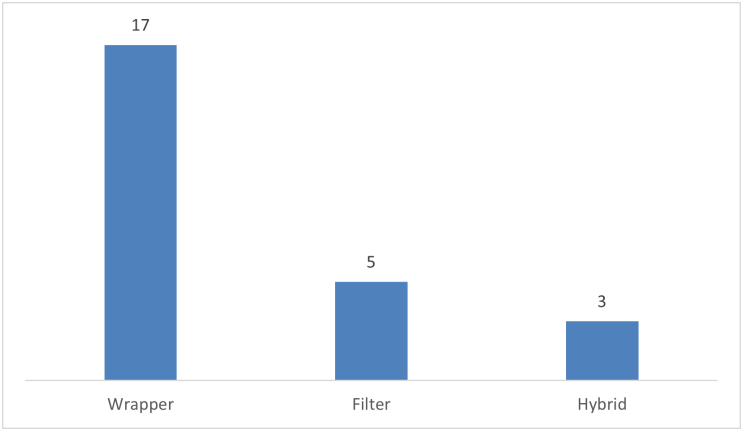

Fig. 3. Number of studies by evaluation approach.

3.3 Answers to Research Questions

Since the research questions are central to this study, all selected studies address at least one of them.

RQ1: What subset/individual evaluation approaches are used in DE-based FS algorithms? Three evaluation approaches were identified among the 25 selected studies. The most popular approach was the wrapper method, implemented in 17 studies. In contrast, filter and hybrid approaches were less common, with five and three studies, respectively, as shown in Figure 3.

As shown in Figure 4, seven classification algorithms were identified within the most popular approach (wrapper). The most commonly used were k-Nearest Neighbors (KNN) and Support Vector Machine (SVM). Three different techniques and measures were found for filter-based approaches, with correlation measures being the most frequently used. In hybrid approaches, two

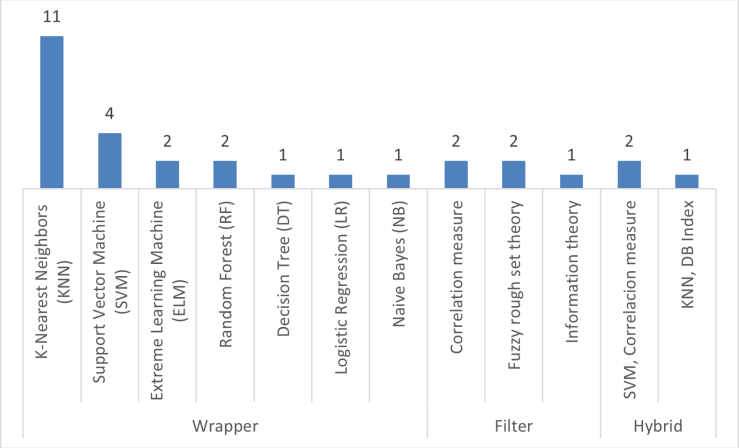

Fig. 4. Number of techniques by evaluation approach.

studies implemented correlation measures and an SVM model, while one used KNN and the Davies–Bouldin (DB) index.

RQ2: What representations of solutions are used in DE for FS algorithms? Three types of solution representations were identified. Among the 25 selected studies, the most commonly used representation was real-valued vectors, followed by binary-valued vectors. In contrast, integer-valued representations were the least used, as shown in Figure 5.

RQ3: What type of optimization (single-objective or multi-objective) is implemented in DE algorithms for FS? Among the two main types of optimization, most selected studies employed single-objective optimization. However, the number of studies using multi-objective optimization was only four fewer than those employing single-objective optimization, see Figure 6.

RQ4. In which applications or domains are DE-based algorithms used for FS? Based on the analysis of the selected studies, the application or domain where DE is implemented for FS was categorized into four main categories: classification, health and bioinformatics, security and informatics systems, and images and sensors. Table 1 specifies which study falls into each category and the specific case in which it is applied.

According to this categorization, most studies use DE for FS in classification with datasets from repositories such as the UCI Machine Learning Repository. These datasets come from various contexts and are used to evaluate the performance of the proposed algorithm across different datasets, comparing it with variations of the same algorithm or other metaheuristics.

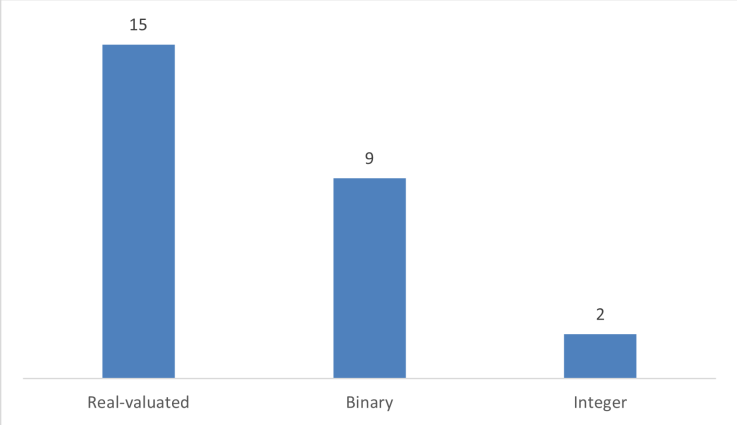

Fig. 5. Number of studies by solution representation.

Table 1. Applications and domains of DE for FS with corresponding studies.

Domain / Application	Case	Study
Classification	General classification tasks using standard benchmark datasets (UCI Machine Learning Repository and similar repositories)	[5],[6],[22],[30],[18],[12], [13],[16],[11],[27],[8],[25], [26],[4]
Health and Bioinformatics	Microarray data analysis for disease diagnosis, stroke prediction, tuberculous pleural effusion diagnosis, high-dimensional medical dataset classification	[28],[24],[20],[31],[17],[10]
Security Systems and Informatics	Network intrusion detection systems (IDS), software fault prediction	[2],[19],[9]
Images and Remote Sensors	Spectral feature selection of hyperspectral remote sensing images, Hand gesture classification using sEMG and motion sensor data	[3],[21]

However, there are studies with specific application contexts where the algorithm is implemented in datasets from a particular domain. In this aspect, most studies focus on health and bioinformatics, with six studies. Additionally, there are three studies in security and informatics systems and only two in the context of images and sensors. See Figure 7.

4 Discussion

Between 2019 and 2024, a regular ascent has been observed in the number of studies that employ DE for FS, with proportional information in the sources consulted, including survey articles and conference papers, demonstrating a growing interest in this field. This increase suggests that the implementation of this algorithm has started to be treated with more seriousness, supported by the quality and prestige of the sources where the studies are published.

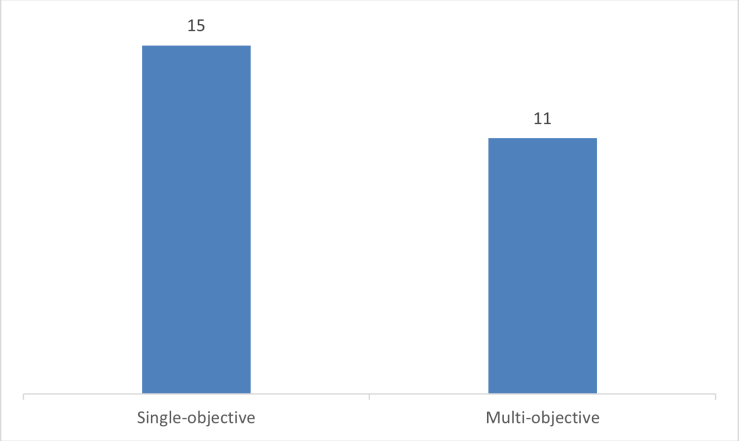

Fig. 6. Number of studies by optimization type.

Most of the selected studies use real-valued representations to encode feature subsets. The latter can be explained by the fact that DE was initially designed for continuous spaces, and working in this space allows the original structure of DE to be maintained without significant modifications. However, some studies, such as [2], have adopted a binary representation, which is appropriate for FS because the goal is to determine whether a feature is selected, transforming the search space into a binary space. This representation needs modifications to the algorithm's structure due to the alteration of the search space. On the other hand, a minority of studies (only 3) use an integer vector representation, where the selected feature index is directly utilized in a vector, such as [4], where, through a permutation strategy, use a permutational-based apace. Although this last strategy is less explored, it represents an opportunity for future work.

Regarding subset/individual evaluation approaches, most studies decide on wrapper approaches despite their higher computational cost. This popularity may be because most studies apply DE for FS in classification problems. Using a classifier as the evaluation criterion is suitable in this context, as it allows the performance of subsets to be measured in terms of their ability to predict adequately. Among the classifiers identified in the selected studies, the most common is KNN, which is less computationally expensive than more complex and robust algorithms like SVM. Despite their significantly smaller presence, Extreme Learning Machine (ELM), Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), and Naive Bayes (NB) are employed in the studies. The preceding points to a possible opportunity to explore using a more complex and robust algorithm that could improve performance while also addressing the challenge of the high computational cost. On the other hand, filter-based approaches, which use measures like correlation, fuzzy set theory, or information theory, are less frequent but offer the advantage of being more computationally efficient.

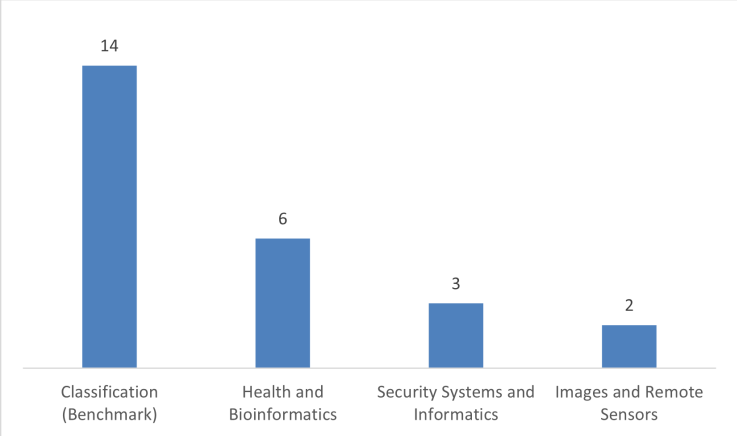

Fig. 7. Number of studies by applications or domains.

A small number of studies use a hybrid approach. In these studies, such as [22], where use a filter approach and wrapper approach in distinct stages, or such as[28] where each subset implements a redundancy measure (correlation) and the classification precision of the subset, looking to get solutions minimally redundant and predictive. Since only three studies tackle them, this field has great potential for future work. In terms of objective optimization, most studies focus on optimizing a single objective function. However, the difference in the number of studies addressing multi-objective optimization is not significant (only four fewer studies in comparison). Although multi-objective optimization involves a more complex process and generates non-dominated solutions, it can lead to better results in terms of solution diversity. The aforementioned provides a series of subsets for the user to choose the most convenient one.

Finally, in terms of applications, most studies implement DE in classification tasks using datasets from standard repositories such as UCI Machine Learning Repository, aiming to assess the algorithm's performance in various contexts and compare it with the performance of other algorithms applied to the same datasets. However, specialized applications were also identified, mainly in areas like health and bioinformatics, as well as (though less frequent) cybersecurity, and images and remote sensors. This focus on more specific applications demonstrates how DE for FS is maturing and offers valuable solutions to concrete relevant problems across various disciplines.

5 Conclusions

All research questions were successfully addressed through a systematic literature review. Trends and characteristics of DE applied to FS were identified, such as the representations used, evaluation methods, techniques employed within the approaches, the number of studies implementing multi-objective or

single-objective optimization, and the various application domains involved. Due to the nature of the research questions, all selected studies provided relevant information to address the objectives posed.

Future research could focus on exploring new representations, distinct approaches, and strategies of DE for FS, as well as on how the algorithm's performance is evaluated and compared with other algorithms. Investigating new DE adaptations with integer-based representations and novel strategies for permutation-based search spaces appears to be a particularly promising area. Furthermore, identifying additional optimization objectives, developing more complex and robust algorithms within wrapper frameworks or hybrid strategies, and applying DE-based FS methods to datasets from different disciplines should also be considered. Additionally, given the popularity and high computational demands of wrapper methods, future research should explore strategies to reduce their computational cost.

Moreover, future studies could investigate more specific aspects of DE algorithms, such as the design and integration of novel operators (e.g., mutation, crossover, and selection mechanisms) specifically tailored for FS tasks. Additionally, the development of new DE-based strategies for FS represents a promising research direction. Finally, further studies could provide a more detailed examination of the factors that influence DE's performance in FS, offering deeper insights into its underlying mechanisms.

References

1. Agrawal, P., Abutarboush, H. F., Ganesh, T., Mohamed, A. W.: Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019). IEEE Access 9, 26766–26791 (2021) https://doi.org/10.1109/ACCESS.2021.3056407
2. Almasoudy, F.H., Al-Yaseen, W.L., Idrees, A.K.: Differential Evolution Wrapper Feature Selection for Intrusion Detection System. Procedia Comput. Sci. 167, 1230–1239 (2020) https://doi.org/10.1016/j.procs.2020.03.438
3. Anamika, Gupta, R., Singh, G.: Binary Differential Evolution-Based Feature Selection for Hand Gesture Classification. In: Agrawal, R., Kishore Singh, C., Goyal, A. (eds.) Advances in Smart Communication and Imaging Systems, LNCE, vol. 721, pp. 221–232. Springer, Singapore (2021) https://doi.org/10.1007/978-981-15-9938-5_22
4. Barradas-Palmeros, J.-A., Mezura-Montes, E., Rivera-López, R., Acosta-Mesa, H.-G.: Computational cost reduction in wrapper approaches for feature selection: A case study using permutational-based differential evolution. In: IEEE Congress on Evolutionary Computation (CEC), pp. 124–131, IEEE (2024)
5. Bidgoli, A.A., Ebrahimpour-Komleh, H., Rahnamayan, S.: A novel multi-objective binary differential evolution algorithm for multi-label feature selection. In: IEEE Conference on Computational Intelligence, pp. 1–6. IEEE, (2019)
6. Bidgoli, A.A., Rahnamayan, S., Ebrahimpour-Komleh, H.: Opposition-based multi-objective binary differential evolution for multi-label feature selection. In: Springer Conference on Computational Intelligence, pp. 15–29. Springer, (2019)
7. Dash, M., Liu, H.: Feature selection for classification. Intell. Data Anal. 1(1–4), 131–156 (1997)

8. Dominico, G., Bernardes, J. S., Dorneles, L. L., Dorn, M.: Multi-Objective Wrapper Differential Evolution with Guided Initial Population for Feature Selection. In: 2023 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2023)
9. Faris, M., Mahmud, M.N., Salleh, M.F.M., Alsharaa, B.: A differential evolution-based algorithm with maturity extension for feature selection in intrusion detection system. Future Generation Computer Systems 137, 201–216 (2023)
10. Gudadhe, S., Thakare, A.: Differential evolution wrapper-based feature selection method for stroke prediction. In: Springer Advances in Data Science and Engineering, pp. 389–402. Springer, (2024)
11. Hancer, E., Xue, B., Zhang, M.: Fuzzy filter cost-sensitive feature selection with differential evolution. Knowledge-Based Systems 241, 108259 (2022) https://doi.org/https://doi.org/10.1016/j.knosys.2022.108259
12. Hancer, E.: New filter approaches for feature selection using differential evolution and fuzzy rough set theory. In: Proceedings of Springer Conference on Computational Intelligence, pp. 220–234. Springer, (2020)
13. Hu, X.-M., Guo, Z.-W.: Multimodal Bare-Bone Niching Differential Evolution in Feature Selection. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1553–1558. IEEE, Melbourne, Australia (2021) https://doi.org/10.1109/SMC52423.2021.9658633
14. Jiao, R., Nguyen, B. H., Xue, B., Zhang, M.: A Survey on Evolutionary Multiobjective Feature Selection in Classification: Approaches, Applications, and Challenges. IEEE Transactions on Evolutionary Computation 28(4), 1156–1176 (2024) https://doi.org/10.1109/TEVC.2023.3292527
15. Kitchenham, B.A., Budgen, D., Brereton, P.: Evidence-Based Software Engineering and Systematic Reviews. Chapman & Hall/CRC, (2015)
16. Li, H., He, F., Chen, Y. et al.: MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. Memetic Comput. 13, 1–18 (2021) https://doi.org/10.1007/s12293-021-00328-7
17. Mostafa, R.R., Khedr, A.M., Al Aghbari, Z., Afyouni, I., Kamel, I., Ahmed, N.: An adaptive hybrid mutated differential evolution feature selection method for low and high-dimensional medical datasets. Knowledge-Based Systems 283, 111218 (2024)
18. Nayak, S. K., Rout, P. K., Jagadev, A. K., Swarnkar, T.: Elitism based multi-objective differential evolution for feature selection: A filter approach with an efficient redundancy measure. *Journal of King Saud University - Computer and Information Sciences* 32(2), 174–187 (2020) https://doi.org/10.1016/j.jksuci.2017.08.001
19. Pethe, Y.S., Das, H.: Software fault prediction using a differential evolution-based wrapper approach for feature selection. Neural Computing and Applications 35, 457–473 (2023)
20. Prajapati, S., Das, H., Gourisaria, M.K.: Feature selection using differential evolution for microarray data classification. In: Springer Advances in Computational Intelligence, pp. 215–227. Springer, (2023)
21. Qian, Y.: Hyperspectral Feature Selection Algorithm Based on Differential Evolution and Multivariate Mutual Information. In: 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), pp. 1821–1826 (2023) https://doi.org/10.1109/ICETCI57876.2023.10176658
22. Qiu, C.: A hybrid two-stage feature selection method based on differential evolution. J. Intell. Fuzzy Syst. 39(1), 871–884 (2020) https://doi.org/10.3233/JIFS-191765
23. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 11(4), 341–359 (1997)

24. Sudhakar, P., Satapathy, S.C.: Feature selection with binary differential evolution for microarray datasets. In: Springer Advances in Intelligent Systems and Computing, pp. 341–354. Springer, (2023)
25. Wang, P., Xue, B., Liang, J., Zhang, M.: Differential Evolution With Duplication Analysis for Feature Selection in Classification. IEEE Transactions on Cybernetics 53(10), 6676–6689 (2023) https://doi.org/10.1109/TCYB.2022.3213236
26. Wang, P., Xue, B., Liang, J., Zhang, M.: Feature selection using diversity-based multi-objective binary differential evolution. Applied Intelligence 53, 1543–1562 (2023)
27. Wang, P., Xue, B., Liang, J., Zhang, M.: Multiobjective Differential Evolution for Feature Selection in Classification. IEEE Transactions on Cybernetics 53(7), 4579–4593 (2023) https://doi.org/10.1109/TCYB.2021.3128540
28. Xie, W., Li, W., Fang, Y., Chi, Y., Yu, K.: A Hybrid Feature Selection Method Based on Binary Differential Evolution and Feature Subset Correlation for Microarray Data. In: 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Las Vegas, NV, USA, pp. 3193–3200 (2022) https://doi.org/10.1109/BIBM55620.2022.9995617
29. Xue, B., Zhang, M., Browne, W.N., Yao, X.: A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. 20(4), 606–626 (2016). https://doi.org/10.1109/TEVC.2015.2504420
30. Zhang, Y., Gong, D., Gao, X., Tian, T., Sun, X.: Binary differential evolution with self-learning for multi-objective feature selection. Inf. Sci. 507, 67–85 (2020) https://doi.org/10.1016/j.ins.2019.08.040
31. Zhou, X., Chen, Y., Gui, W., Heidari, A.A., Cai, Z., Wang, M., Chen, H., Li, C.: Enhanced differential evolution algorithm for feature selection in tuberculous pleural effusion clinical characteristics analysis. Computers in Biology and Medicine 170, 106914 (2024)

ISSN 1870-4069

Bayesian Mechanics of Economic Choice: Computational Foundations of Economic Behavior

Samuel Montañez[1], Luis Alberto Quezada-Telléz[2], Ernesto Moya-Albor[3]

[1,3] Universidad Panamericana, Facultad de Ingeniería, México

[2] Universidad Autónoma de Hidalgo, México

{smontanez, emoya}@up.edu.mx, alquezada@ciencias.unam.mx

Abstract. This paper presents a theoretical unification of neuroeconomics with the Free Energy Principle (FEP) framework. We demonstrate that economic decision-making can be formulated as a variational inference problem where agents minimize expected free energy, balancing risk (aligning predictions with preferences) and ambiguity (reducing uncertainty). Our formal analysis establishes mathematical equivalence between divisive normalization in neuroeconomic models and precision-weighted prediction error minimization in active inference. We show how Expected Subjective Value Theory (ESVT) from neuroeconomics naturally emerges from the FEP under Gaussian assumptions, explaining context-dependent valuation, reference-dependence, and risk attitudes through a common computational mechanism and generative model. This unification has significant implications for artificial intelligence, providing computational principles for developing more human-like decision-making agents that balance exploration and exploitation in an information-theoretic way. By bridging Bayesian mechanics with divisive normalization, we provide a neurobiologically plausible foundation for economic behavior that encompasses both classical utility maximization and information-theoretic approaches to decision-making under uncertainty. By integrating thermodynamic principles of information processing, we demonstrate how economic decision-making operates under physical constraints, offering a theoretical foundation for AI systems that must optimize computational resources while managing uncertainty.

Keywords: Expected subjective value, variational free energy, active inference, neuroeconomics, Bayesian mechanics.

1 The Free Energy Principle as a Decision-Making Framework

Active inference has expanded beyond neuroscience into diverse domains [1, 2, 3, 4, 5, 6, 7], establishing itself as a unified framework for modeling decision-making under uncertainty. Its foundational principles—variational free energy minimization and belief updating through precision-weighted prediction errors [8]—provide mechanistic explanations for complex behaviors across multiple temporal scales.

This method which can be read as a physics of sentience is known as the new and growing field of Bayesian mechanics [8]. What distinguishes this approach is its ability to simultaneously address perception, learning, and action within a single coherent theoretical structure, accommodating both optimal and seemingly suboptimal behaviors that challenge traditional modeling approaches [9]. This integrative capacity makes active inference particularly valuable for artificial intelligence, where systems must similarly balance perception, learning, and action within unified computational architectures [1]. While reinforcement learning approaches separate perception from action and require external reward signals, active inference offers AI a more cohesive computational framework where perception, learning, and action emerge from the single imperative of free energy minimization [1].

The free energy principle (FEP) offers a neurobiologically plausible foundation for economic behavior by unifying Bayesian decision theory with statistical physics and information-theoretic approaches to uncertainty [10]. This framework shares mathematical equivalence with predictive coding [11] and relates conceptually to the Helmholtz free energy in statistical physics [12, 13]. Its computational architecture—minimizing the long-term average of sensory surprisal through an internal generative model—provides a principled basis for understanding both information processing and decision-making [14, 15, 16, 17].

Recent empirical evidence supports the role of dopamine in encoding precision of beliefs about optimal policies, demonstrating that humans employ hierarchical Bayesian inference to simultaneously determine both what they should do and how confident they should be a process that aligns more closely with active inference than with classical utility maximization [18]. This framework extends Barlow's efficient coding hypothesis [19], providing a formal basis for understanding economic decision-making as optimized information processing under biological constraints.

The application of FEP to economics has been limited despite its significant potential. While some research has applied these principles to temporal discounting [20] and bounded rationality [21], a formal connection between current neuroeconomic models and the FEP remains to be established. In this paper, we demonstrate that divisive normalization—a canonical neural computation central to Expected Subjective Value Theory (ESVT) [22] in neuroeconomics—emerges naturally from perceptual inference under the FEP. By proving this formal equivalence, we provide a theoretical foundation for understanding economic behaviors as manifestations of precision-weighted prediction error minimization in the brain, laying a bioinspired roadmap for developing AI agents capable of human-like economic decision-making and planning under the same principled information-theoretic framework [16].

2 Thermodynamic Foundations of Decision-Making

The minimization of complexity in free energy optimization has fundamental thermodynamic implications through Landauer's principle [23], which establishes a lower bound of $kT\ln(2)$ energy expenditure per bit erased. This physical constraint bridges information theory and thermodynamics, suggesting that cognitive efficiency has metabolic consequences. When economic agents employ parsimonious

representations of market dynamics, they reduce both complexity and associated energetic costs [24].

Under the FEP, neural information processing optimizes the trade-off between accuracy and complexity, manifesting biologically as the calibration of synaptic weights within hierarchical neural architectures [14]. This optimization process explains the prevalence of simplified heuristics and dimensionality reduction in successful economic strategies: such approaches preserve essential predictive power while minimizing metabolic expenditure [23] and stochasticity in choice [24].

The active inference framework formalizes this efficiency through precision-weighted prediction errors, where Bayesian belief updating is dynamically modulated by confidence estimates [8, 14]. This precision-weighting mechanism creates effective information compression, enabling adaptive decision-making even under severe computational constraints [21]. Economic agents implementing this mechanism naturally balance exploration (uncertainty reduction) with exploitation (preference satisfaction) without requiring exhaustive computation [16].

This thermodynamic perspective clarifies why seemingly "irrational" economic behaviors may represent optimal solutions under biological constraints [25]. Rather than implementing general-purpose rationality, economic cognition leverages domain-specific adaptations that exploit statistical regularities in environmental structure [26, 27]. These adaptations can be understood as instantiating a form of "Neural Darwinism," where neural architectures implement anti-entropic mechanisms that maintain functional organization against thermodynamic dissipation [14].

The resulting agent-environment system forms a Markov blanket structure that maintains integrity amid environmental fluctuations [8]. This statistical separation between internal and external states provides a formal basis for bounded rationality in economics, where access to market information is necessarily limited and costly. By minimizing expected free energy, economic agents effectively push "thermodynamically uphill" against disorder, implementing computationally efficient solutions [16]. This formulation provides a principled foundation for understanding economic decision-making as optimal inference under physical and informational constraints.

3 The Free Energy Principle and Active Inference

3.1 The Free Energy Principle Explained

The FEP posits that biological systems, including the human brain, strive to minimize free energy, which corresponds to reducing surprise or uncertainty by forming predictions about their environment [8, 10, 16]. This principle serves as a bridge between cognitive science and physics, suggesting that life, cognition, and evolution can be understood through a unified framework that aligns with variational principles such as Hamilton's principle of least action in classical mechanics [2].

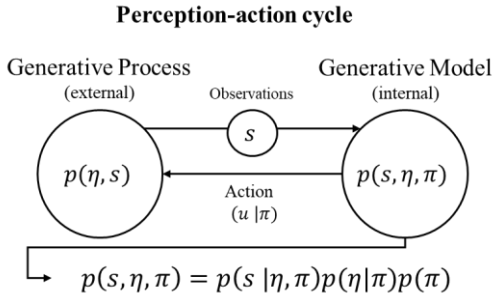

Fig. 1. Perception-action cycle in active inference. The figure illustrates the circular causal relationship between a generative process (external) and generative model (internal) connected through observations and actions. The generative process $p(\eta, s)$ represents the true causal structure governing the external states η and observations s, while the generative model $p(s, \eta, \pi)$ embodies the agent's beliefs about how observations and hidden states are generated, parameterized by policy π. Perception corresponds to inferring external states from observations, while action $(u \mid \pi)$ influences the generative process according to policies inferred from the generative model. The joint distribution of the generative model factorizes as $p(s, \eta, \pi) = p(s \mid \eta, \pi)p(\eta|\pi)p(\pi)$, where $p(s \mid \eta, \pi)$ encodes the likelihood mapping, $p(\eta|\pi)$ represents conditional beliefs about external states given policies, and $p(\pi)$ encodes prior beliefs over policies. This formulation instantiates active inference, where agents select policies that minimize expected free energy, thereby reducing the divergence between the generative process and generative model through perception and action [9].

In active inference, the expected free energy (G) quantifies the probabilistic divergence between anticipated trajectories and preferred outcomes, while accounting for uncertainty reduction. Formally, $G(\alpha[\tau])$ represents the expected free energy of a policy or autonomous path $\alpha[\tau]$, equivalent to an action functional $\mathcal{A}(\alpha[\tau])$. The FEP states that agents select paths that minimize G formalized as [9]:

$$\boldsymbol{\alpha}[\tau] = \arg min_{\alpha[\tau]} G(\alpha[\tau]), \tag{1}$$

Perceptual inference minimizes free energy $F(s, \boldsymbol{\alpha})$ by updating internal states α:

$$\dot{\boldsymbol{\alpha}}(\tau) = (Q_{\alpha\alpha} - \Gamma_{\alpha})\nabla_{\alpha} F(s, \boldsymbol{\alpha}). \tag{2}$$

This cyclical process creates a dynamic equilibrium between perception and action, formalized through the coupled equations [9]:

$$\dot{\boldsymbol{\alpha}}(\tau) = (Q_{\alpha\alpha} - \Gamma_{\alpha})\nabla_{\alpha} F(s, \boldsymbol{\alpha}), \tag{3}$$

$$\boldsymbol{\alpha}[\tau] = \arg min_{\alpha[\tau]} G(\alpha[\tau]). \tag{4}$$

The closed perception-action loop implements an approximate Bayesian filtering scheme analogous to the Hamilton-Jacobi-Bellman (HJB) equation in continuous control, but uniquely optimizes both information gain and expected utility, explaining exploration-exploitation dynamics in economic decision-making [16, 18]. Active inference generalizes standard reinforcement learning by driving agents to minimize expected surprise rather than simply maximize rewards [28, 29]. This fundamental difference provides a more comprehensive framework for modeling decision-making

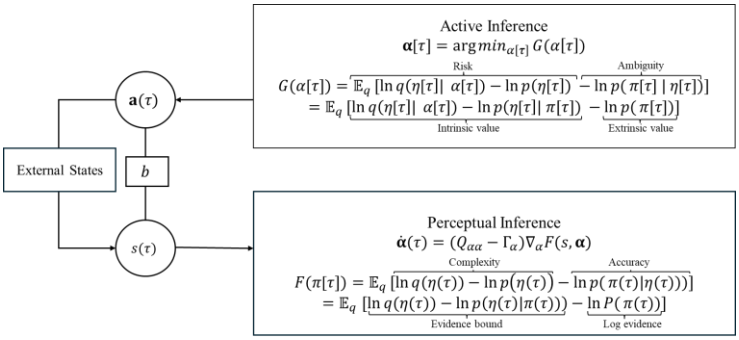

Fig. 2. Active and Perceptual Inference. This diagram illustrates the dual optimization processes in biological and sentient systems. The left panel depicts the relationship between external states $\eta \subset x$, sensory states ($s(\tau)$), and autonomous states $\boldsymbol{\alpha}(\tau)$, linked by a Markov blanket b in current time τ. The autonomous states $\boldsymbol{\alpha} = (a, \mu)$ comprise active states $a \subset x$,and internal states $\mu \subset x$, while the blanket states $b = (s, a)$ consist of sensory states $s(\tau)$ and active states $\boldsymbol{\alpha}(\tau)$, collectively creating a partition of states that separates internal from external states $\eta(\tau)$. The top right panel shows Active Inference, where agents select policies $\pi(\tau)$ that minimize expected free energy $G(\alpha[\tau])$, balancing risk (aligning with preferences) against ambiguity (reducing uncertainty). The expected free energy represents the difference between posterior and prior beliefs about external states, which can be decomposed into intrinsic value (information gain) and extrinsic value (preference satisfaction). The bottom right panel represents Perceptual Inference, where agents update beliefs through gradient flows that minimize variational free energy $F(s, \boldsymbol{\alpha})$, optimizing the trade-off between accuracy and complexity. Here $q(\eta(\tau)$ denotes the recognition density or approximate posterior distribution over external states at time τ. The term $F(s, \boldsymbol{\alpha})$ is this free energy functional, and $\nabla_\alpha F(s, \boldsymbol{\alpha})$ is its gradient with respect to internal states, with precision weighting $Q_{\alpha\alpha}$ and friction Γ_α modulating the update speed. Together, these complementary processes enable adaptive self-organization through continual prediction and uncertainty minimization [8,9].

under uncertainty, where agents update beliefs and select actions to minimize surprise rather than merely accumulate rewards.

3.2 Variational Free Energy

When considering perceptual inference, we can express the variational free energy through the Kullback-Leibler (KL) divergence between an approximate posterior distribution $q(\eta)$ and the true posterior $p(\eta|s)$ [30]:

$$KL\big(q(\eta), p(\eta|s)\big) = \int q(\eta) \left(\ln \frac{q(\eta)}{p(\eta|s)}\right) d\eta, \tag{5}$$

By substituting the definition of conditional probability and taking logarithms [30]:

$$KL\big(q(\eta), p(\eta|s)\big) = -F + \ln p(s), \tag{6}$$

where F is the negative free energy. Assuming that $q(\eta)$ is a delta distribution, the negative free energy simplifies to [30]:

$$F = \int q(\eta) \ln\left(\frac{p(s|\eta)}{q(\eta)}\right) d\eta = \int q(\eta) \ln p(s|\eta) d\eta - \int q(\eta) \ln q(\eta) d\eta \,. \quad (7)$$

This assumption implies that $q(\eta) = \delta(\eta - \hat{\eta})$, where δ is the Dirac delta function and $\hat{\eta}$ represents a point estimate. This effectively transforms the variational problem into a maximum a posteriori (MAP) estimation, eliminating the entropy term $-\int q(\eta) lnq(\eta) d\eta$ and simplifying the free energy formulation [8].

The expectation operator $\mathbb{E}_q$ represents the weighted average with respect to the distribution q:

$$F = \mathbb{E}_{q(\eta)}\left[\ln\left(\frac{p(s|\eta)}{q(\eta)}\right)\right] = \int q(\eta) \ln\left(\frac{p(s|\eta)}{q(\eta)}\right) d\eta. \quad (8)$$

If we define a generative model as $m = p(s, \eta) = p(s|\eta)\, p(\eta)$, then [9]:

$$F = \mathbb{E}_{q(\eta)}\left[\ln\left(\frac{q(\eta)}{p(\eta|s)}\right)\right] - \ln p(s), \quad (9)$$

where $q(\eta) = p(s|\eta)$, free energy reduces to surprise $-lnp(s)$.

Introducing policies the framework extends to active inference [9]:

$$F = \mathbb{E}_{q(\eta|\pi)}\left[\ln\left(\frac{q(\eta|\pi)}{p(\eta|s,\pi)}\right)\right] - \ln p(s|\pi). \quad (10)$$

With time dependencies (τ), the expectation operator becomes [9]:

$$\mathbb{E}_{q(\eta(\tau)|\pi[\tau])}[f(\eta(\tau), \pi[\tau])] = \int q(\eta(\tau)|\pi[\tau]) f(\eta(\tau), \pi[\tau]) d\eta(\tau). \quad (11)$$

The expectation notation encapsulates the integration over all possible states [9]:

$$F(\pi[\tau]) = \mathbb{E}_q\left[\ln q\big(\eta(\tau)\big) - \ln p\big(\eta(\tau)\big) - \ln p\big(\pi(\tau)|\eta(\tau)\big)\big)\right]. \quad (12)$$

This extension is often called "active inference" where the system not only infers hidden states but also selects policies that minimize expected free energy in the future. In active inference, the agent selects policies that are expected to minimize surprise (or maximize model evidence) in the future. This allows the framework to address not just perceptual inference but also decision-making and planning to fulfill their preferences or prior beliefs about desired states [16,31].

4 From Free Energy to Economics

4.1 Optimal Encoding Strategy

Steverson et al. [24] proposed that economic decision-making can be formalized as an optimization problem balancing expected utility maximization against information-processing costs. Making precise (non-stochastic) choices requires cognitive resources, which can be quantified using information theory. The optimization problem they propose takes the form:

$$\rho(x, A) \in \arg\max \textstyle\sum_{x \in A} p(x) v(x) - C_A(\Delta H(p)), \quad (13)$$

where $\rho(x, A)$ is the probability of choosing option x from choice set A, $v(x)$ is the value of option x, and $C_A(\Delta H(p)$ represents the cognitive cost of reducing choice entropy. The entropy reduction term $\Delta H(p)$ measures deviation from random choice [24]:

$$\Delta H(p) \coloneqq \ln|A| - H(p), \tag{14}$$

where $H(p)$ is the Shannon entropy of the choice distribution:

$$H(p) \coloneqq -\sum_{x \in A} \left[\frac{p(x)}{p(A)}\right] \ln \left[\frac{p(x)}{p(A)}\right], \tag{15}$$

Solving this optimization problem, the resulting choice probabilities follow the form:

$$\rho(x, A) = \frac{\exp(\frac{\gamma v(x)}{\sigma + v(A)})}{\sum_{y \in A} \exp(\frac{\gamma v(y)}{\sigma + v(A)})}. \tag{16}$$

This expression is structurally identical to a softmax over divisively normalized values [32], providing a formal derivation of context-dependent valuation. This derivation shows that divisive normalization emerges naturally when agents optimize a trade-off between maximizing value and minimizing cognitive costs. The divisive term in the denominator $\sigma + v(A)$ effectively normalizes option values based on the overall value of the choice set, just as neurons in the brain normalize their firing rates based on surrounding activity [24].

This optimization approach aligns with principles from active inference, where agents must minimize entropy to maintain a stable identity within fluctuating environments [2]. This can be formalized using the entropy of the probability distribution $p(s^*)$ of finding the agent in a given state $s^*$of its state space S^* [1]:

$$H(S^*) = \int_{s* \in S^*}^{S^*} \big(-\ln p(s^*)\big) p(s^*) ds^*, \tag{17}$$

where s^*can be replaced by observation space S. This entropy minimization corresponds precisely to the complexity term in the variational free energy functional, establishing a direct mathematical link between economic choice and the FEP framework. This connection has deep roots in statistical physics, where the Boltzmann distribution emerges as the probability distribution that minimizes Helmholtz free energy while maintaining a fixed average energy [1, 24]. The formal equivalence between thermodynamic systems and decision-making agents is not merely analogical—both involve systems that dynamically settle into probability distributions that optimize a free energy functional, subject to constraints [8, 13]. Just as physical systems minimize thermodynamic free energy to reach equilibrium, cognitive systems appear to minimize information-theoretic free energy to optimize behavior under constraints. Moreover, the entropy component parallels recent advances in reinforcement learning, where incorporating entropy regularization into reward objectives enhances algorithmic performance, stabilizes policy optimization, and improves generalization capabilities [2, 16].

4.2 From Free Energy to Expected Utility Theory

There is a clear path from the FEP to expected utility theory in economics. The FEP conceptualizes the world as a random dynamical system, with active inference explaining how self-organization emerges [8]. Within this framework, economic agents can be treated as adaptive systems whose behaviors are amenable to analysis through Bayesian mechanics [33]. This formulation offers a principled foundation for addressing unresolved challenges in economic decision-making, as it integrates effectively with recent advances in neuroeconomics through its accompanying neural process theory, enabling testable empirical predictions about neural responses [9].

When the epistemic value component is removed from the active inference framework, what remains is essentially the expected log probability of preferred outcomes. This is equivalent to maximizing expected utility ($\mathbb{E}[U_A]$) in economic theory [8,34]:

$$\mathbb{E} \ln p(\pi[\tau])] \cong \mathbb{E}[U_A] = \sum_{o \epsilon O} P(o|A)\, U(o), \tag{18}$$

where $P(o|A)$ represents the conditional probability of obtaining outcome o given action A, and $U(o)$ denotes the utility or value associated with each possible outcome o in the set O. Expected utility maximization emerges as a special case of active inference when uncertainty reduction is not prioritized [8].

Rational choice theory posits that decision-makers employ rational calculations to optimize outcomes aligned with their preferences. Within this framework, preferences over independent outcomes remain consistent regardless of irrelevant factors—a fundamental principle known as independence [35]. Von Neumann and Morgenstern [34] interpreted conditional probabilities as objective chances within a perfectly rational framework rather than as beliefs about states. Furthermore, rational choice theory assumes that options possess absolute values independent of the value or existence of alternative options [36]. Active inference provides a more general formulation by reinterpreting utility functions as prior preference distributions, suggesting that observed behavior can be understood as Bayes optimal under some prior beliefs [18].

4.3 Expected Subjective Value Theory: A Neuroeconomic Model

Divisive normalization has emerged as a critical computation employed by the brain to facilitate decision-making. It functions as a canonical neural computation, contributing to efficient processing within neural circuits [32]. The work of Reynolds and Heeger [37] indicates that rectification can approximate a power law, resulting in contrast-response functions that align more closely with electrophysiological data than previous assumptions [22, 36]. Research has posited that the brain utilizes divisive normalization in a utility-like calculation during choice-making, which entails balancing the expected value of options against the entropic cost of reducing stochasticity [38-39].

Glimcher and Tymula's biophysical implementation in Expected Subjective Value Theory (ESVT) [22]—a neuroeconomics-based model of expected utility—demonstrate that normalization emerges from interacting excitatory and inhibitory neurons, described by coupled differential equations [22, 40]:

$$\tau \frac{dR}{dt} = -R + \frac{x}{1+G}, \tag{19}$$

$$\tau \frac{dG}{dt} = -G + R, \tag{20}$$

where R represents excitatory activity, G inhibitory activity, and x the objective value of the payoff (utility). This system converges to a unique equilibrium state:

$$\tau \frac{dG}{dt} = -G + R. \tag{21}$$

Two properties emerge: (1) this equilibrium state corresponds to standard divisive normalization, and (2) normalization arises from temporal integration of value inputs. In dynamic contexts, the action potential rate evolves as [22]:

$$R_t \propto \frac{x_t}{x_t + \sum_{k=0}^{t-1} D(k) x_k}, \tag{22}$$

where the denominator represents a weighting function $D(k)$ and a time-discounted average $(\sum_{k=0}^{t-1} D(k)x_k)$ of previously encountered payoffs (x_k)—effectively implementing reference-dependent valuation through neural computation [22]. This neurobiological implementation provides an understanding of how the brain might encode subjective values. As we will demonstrate in subsequent sections, this same normalization structure emerges from free energy minimization under specific assumptions.

The core of ESVT is a subjective value function mapping objective payoffs to neural representations [22]:

$$S_t(x) = \frac{x^\alpha}{x^\alpha + M_t^\alpha}, \tag{23}$$

where $S_t(x)$ represents the subjective value of payoff $x \in \mathbb{R}$ at time t, which corresponds to the neural firing rate encoding the value representation. The term M_t denotes the payoff expectation based on previously experienced outcomes, implementing a form of reference-dependence that emerges from neurobiological architecture. The predisposition parameter α controls the curvature of the value function, capturing individual differences in risk attitudes and value sensitivity, with lower values producing concave functions (risk aversion) and higher values yielding the characteristic sigmoid functions observed in prospect theory [22].

The payoff expectation is recursively computed as a time-weighted average of previous outcomes [22]:

$$S_t(x) = \frac{x^\alpha}{x^\alpha + M_t^\alpha}, \tag{24}$$

where $\gamma \in (0,1)$ represents the forgetting rate, capturing recency effects in expectation formation.

This formulation produces a subjective value function bounded between 0 and 1, consistent with neurobiological constraints on value encoding. Critically, unlike traditional utility formulations, ESVT provides a cardinal measure of subjective value that corresponds directly to neural firing rates observed in valuation regions of the

brain. ESVT captures not just ordering but also the intensity or magnitude of preferences, allowing for more precise predictions about behavior [22].

In ESVT neuronal firing rates represent excitatory input modulated by surrounding activity parallels the precision-weighting mechanisms in perceptual inference [22]. Research indicates that dynamic divisive normalization operates not only spatially but also temporally, influencing how perceptual evidence is weighted over time during decision-making tasks [41]. The context-dependence of divisive normalization has been linked to behavioral features previously unnoticed by economists, suggesting that it can predict how individuals adapt their preferences based on reward contexts during reinforcement learning [36,42].

4.4 From Free Energy to Divisive Normalization

In this section, we formally demonstrate that divisive normalization—the key computational mechanism in ESVT—emerges naturally from perceptual inference under the FEP framework, establishing a formal equivalence between these approaches.

The FEP and ESVT both characterize how neural systems optimize information processing under biological constraints, though they emerged from different disciplinary traditions. Both frameworks describe neural systems that maximize information transmission while minimizing metabolic costs. While divisive normalization describes how neurons encode information about the world [32], this precisely corresponds to perceptual inference within the FEP framework [8].

We demonstrate that divisive normalization emerges from perceptual inference under the FEP by considering a hierarchical generative model with Gaussian priors and likelihoods. Let's consider the free energy under Gaussian assumptions [9, 30]:

$$F = D_{KL}[q(\eta|\alpha)\|p(\eta|s)]\text{-}\ln p(s). \tag{25}$$

At steady state ($\dot{\alpha} \doteq 0$), internal states satisfy:

$$\nabla_{\alpha} F(s, \boldsymbol{\alpha}) = 0. \tag{26}$$

Under Gaussian assumptions for recognition and generative densities is [8]:

$$q((\eta[\tau]|\alpha[\tau]) = \mathcal{N}(\mu_{\eta}, \Sigma_{\eta}), \tag{27}$$

$$p(\eta[\tau]|\alpha[\tau]) = \mathcal{N}\left(g(\pi[\tau]), \Pi_{\eta}^{-1}\right), \tag{28}$$

where Π_{η}^{-1} is the precision of prediction errors and Σ_{η} is the covariance matrix (inverse of the precision matrix Π_{η}). The gradient of free energy with respect to the conditional mean becomes:

$$\nabla_{\mu_{\eta}} F = \Pi_{\eta}\left(\mu_{\eta} - g(\pi[\tau])\right) - \nabla_{\mu_{\eta}} h(\eta)^{T} \Pi_{s}(s - h\left(g(\pi[\tau])\right)). \tag{29}$$

Lemma 1. (Divisive Normalization Equivalence): Under a hierarchical generative model with Gaussian priors and likelihoods, perceptual inference through free energy minimization converges to a representation that is equivalent to divisive normalization as formulated in Expected Subjective Value Theory.

Let a generative model be defined with Gaussian recognition density $\mathrm{q}((\eta[\tau]|\alpha[\tau]) = \mathcal{N}(\mu_{\eta}, \Sigma_{\eta})$ and generative density $\mathrm{p}(\eta[\tau]|\alpha[\tau]) = \mathcal{N}(\mathrm{g}(\pi[\tau]), \Pi_{\eta}^{-1})$.

At the steady state where $\nabla_\alpha F(s, \boldsymbol{\alpha}) = 0$, the optimal internal representation μ_η takes the form:

$$\mu_\eta = \frac{g(\pi[\tau]) + \sum_\eta \nabla_{\mu_\eta} h(\eta)^T \Pi_s (s - h(g(\pi[\tau])))}{1 + \sum_\eta \nabla_{\mu_\eta} h(\eta)^T \Pi_s \nabla_{\mu_\eta} h(\eta)}, \tag{30}$$

where the numerator represents direct input (predicted value), and the denominator represents a baseline term plus contextual modulation that implements precision-weighted prediction error minimization. This representation is structurally equivalent to the divisive normalization formulation in ESVT [22].

Proof: Consider the variational free energy under Gaussian assumptions [9]:

$$F = D_{KL}[q(\eta|\alpha)||p(\eta|s)] - \ln p(s), \tag{31}$$

At steady state, internal states satisfy $\nabla_\alpha F(s, \boldsymbol{\alpha}) = 0$. The gradient of free energy with respect to the conditional mean is:

$$\nabla_{\mu_\eta} F = \Pi_\eta \left(\mu_\eta - g(\pi[\tau])\right) - \nabla_{\mu_\eta} h(\eta)^T \Pi_s (s - h\big(g(\pi[\tau])\big)), \tag{32}$$

where Π_η is the precision of the prior (inverse covariance), μ_η is the conditional mean (posterior expectation), $h(\eta)^T$ is the mapping from hidden states to observations, Π_s is the precision of sensory prediction errors, and $(s - h\big(g(\pi[\tau])\big))$ represents the sensory prediction error. This can be directly mapped to the ESVT's divisive normalization equation [22]:

$$S_t(x) = \frac{x^\alpha}{x^\alpha + M_t^\alpha}, \tag{33}$$

with the following correspondences:

$S_t(x) \leftrightarrow \mu_\eta$ (neural activity encodes expected causes),

$x^\alpha \leftrightarrow g(\pi[\tau])$ (direct input maps to prior prediction),

$x^\alpha \leftrightarrow 1$ (baseline term maps to constant normalization factor),

$M_t^\alpha \leftrightarrow \sum_\eta \nabla_{\mu_\eta} h(\eta)^T \Pi_s \nabla_{\mu_\eta} h(\eta)$ (contextual modulation maps to precision-weighted prediction error terms).

Therefore, divisive normalization in neural systems can be understood as implementing precision-weighted prediction error minimization as prescribed by the FEP.

This equivalence explains both why divisive normalization appears throughout the nervous system and provides a theoretical foundation for understanding seemingly irrational economic behaviors as optimal inference under constraints. The metabolic efficiency principle aligns with our economic agent model, suggesting that context-dependent valuation and reference-dependence emerge naturally from free energy minimization. By demonstrating this mathematical correspondence, we provide a mechanistic explanation for how Bayesian mechanics can describe economic choice.

5 Conclusion

Our formal demonstration of mathematical equivalence between divisive normalization and free energy minimization under Gaussian assumptions establishes a fundamental principle of neural computation. Through Lemma 1, we proved that the steady-state solution to variational inference under the FEP yields precisely the divisive normalization model central to modern neuroeconomic models. This equivalence explains the pervasiveness of normalization across neural systems and reveals it not merely as an implementation detail, but as a necessary property of self-organizing systems maintaining integrity against environmental entropy. By establishing this bridge between neural implementation and economic theory, our analysis unifies ESVT [22] with active inference [9]. This unification has profound implications for artificial intelligence development, particularly for systems that require adaptive decision-making and planning under uncertainty.

The computational mechanisms we identify could lead to more efficient AI architectures that naturally balance exploration and exploitation, mirroring the specialized resource-constrained optimization that biological systems have evolved to implement. This integration aligns with Barlow's efficient coding hypothesis [19], demonstrating that the neural substrates of economic decision-making reflect computational imperatives constrained by biological reality. The precision-weighting mechanism explains how context-dependent valuation and reference-dependence emerge naturally from free energy minimization. Our framework explains the exploration-exploitation trade-off fundamental to economic decisions as the natural balance between ambiguity minimization and risk management within the active inference formalism. For AI, these principles provide a computational neuroscience foundation for designing systems that can adapt to context-dependent valuations and make decisions that appear irrational under classical utility theory yet are optimal when accounting for informational and computational constraints. AI agents implementing these principles could better model human economic behavior while also achieving greater computational efficiency through principled dimensionality reduction similar to what Barlow called “economy of though” [19].

This synthesis offers behavioral scientists a comprehensive framework for understanding decision-making that spans from computational principles to neural mechanisms. By adopting this perspective, researchers can develop more nuanced models of economic behavior that account for the cognitive processes underlying decision-making, particularly how agents navigate uncertainty through predictive modeling.

Future AI research could leverage these insights to develop agents that not only maximize reward but actively maintain their internal model integrity through Bayesian mechanics, potentially leading to more robust and adaptive systems in complex, changing environments. Furthermore, future research should examine whether the correspondence we identified extends beyond Gaussian assumptions. The application of the FEP to economics enables a deeper understanding of economic behavior through the solid foundations of information theory and statistical mechanics, offering a roadmap for physics-informed and bioinspired AI that is simultaneously more human-like in its decision processes and more principled in its computational implementation.

Acknowledgments. Samuel Montañez and Ernesto Moya-Albor extend their gratitude to the Faculty of Engineering at Universidad Panamericana for their support throughout this research endeavor. Some of the graphical representations and explanations were taken or inspired by papers created by K. Friston and colleagues.

References

1. Ueltzhöffer, K.: Deep active inference. Biological Cybernetics 112(6), pp. 547–573 (2018) doi: 10.1007/s00422-018-0785-7.
2. Ramstead, M.J., Constant, A, Badcock, P.B., Friston, K.J.: Variational ecology and the physics of sentient systems. Phys Life Rev 31, pp. 188–205 (2019) doi: 10.1016/j.plrev. 2018.12.002.
3. Kawahara, D., Ozeki, S., Mizuuchi, I.: A curiosity algorithm for robots based on the free energy principle. In: IEEE/SICE International Symposium on System Integration (SII), pp. 53–59 (2022) doi: 10.1109/SII52469.2022.9708814
4. Van de Maele, T., Verbelen, T., Çatal, O., De Boom, C., Dhoedt, B.: Active vision for robot manipulators using the free energy principle. Frontiers in Neurorobotics, 15, pp. 642780 (2021) doi: 10.3389/fnbot.2021.642780.
5. Wirkuttis, N., Ohata, W., Tani, J.: Turn-taking mechanisms in imitative interaction: Robotic social interaction based on the free energy principle. Entropy 25(2), pp. 263 (2023) doi: 10.3390/e25020263.
6. Smith, R., Ramstead, M.J.D., Kiefer, A.: Active inference models do not contradict folk psychology. Synthese, 200(2), pp. 81 (2022) doi: 10.1007/s11229-022-03624-y.
7. Hesp, C., Smith, R., Parr, T., Allen, M., Friston, K.J., Ramstead, M.J.: Deeply felt affect: The emergence of valence in deep active inference. Neural Computation, 33(2), pp. 398–446 (2021) doi: 10.1162/neco_a_01341.
8. Friston, K., Da Costa, L., Sajid, N., Heins, C., Ueltzhöffer, K., Pavliotis, G.A., Parr, T.: The free energy principle made simpler but not too simple. Physics Reports 1024, pp. 1–29 (2023)
9. Smith, R., Friston, K.J., Whyte, C.J.: A step-by-step tutorial on active inference and its application to empirical data. J Math Psychol 107: pp. 102632 (2021) doi: 10.1016/ j.jmp.2021.102632.
10. Parr, T., Friston, K.J.: Active inference, Bayesian optimal design, and expected utility. Int J Forecast 35(2), pp. 219–233 (2019)
11. Rao, R.P.N., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience, 2(1), pp. 79–87 (1999) doi: 10.1038/4580.
12. Dayan, P., Hinton, G.E., Neal, R.M., Zemel, R.S.: The helmholtz machine. Neural Computation, 7(5), pp. 889–904 (1995)
13. Friston, K. J., Daunizeau, J., Kilner, J., Kiebel, S. J.: Action and behavior: a free-energy formulation. Biological Cybernetics (2010) doi: 10.1007/s00422-010-0364-z.
14. Friston, K.: The free-energy principle: a unified brain theory? Nat Rev Neurosci 11, pp. 127–138 (2010) doi: 10.1038/nrn2787.
15. Friston, K.: I am ther before I think. In: The Unconscious. Routledge, London, pp 127–151 (2016)
16. Da Costa, L., Sajid, N., Parr, T., Friston, K., Smith, R.: Reward Maximization through Discrete Active Inference. Neural Computation 35(5), pp. 807–852 (2023)
17. Friston, K.J., Salvatori, T., Isomura, T., Tschantz, A., Kiefer, A., Verbelen, T., Sajid, N., Adams, R.A., Parr, T., Ramstead, M.J.: Active Inference and Intentional Behavior. Neural Computation 37(4), pp. 666–700 (2025)

18. Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G.: Active inference and learning. Neuroscience & Biobehavioral Reviews 68: pp. 862–879 (2016).
19. Barlow, H.B.: Possible principles underlying the transformations of sensory messages. In: Rosenblith WA (ed) Sensory communication. MIT Press, Cambridge, pp 217–234 (1961)
20. Henriksen, M.: Variational free energy and economics optimizing with biases and bounded rationality. Front Psychol 11, pp. 549187 (2020) doi: 10.3389/fpsyg.2020.549187.
21. Genewein, T., Leibfried, F., Grau-Moya, J., Braun, D.: Bounded Rationality, Abstraction, and Hierarchical Decision-Making: An Information-Theoretic Optimality Principle. Frontiers in Robotics and AI, 2, pp. 27 (2015) doi: 10.3389/frobt.2015.00027.
22. Tymula, A., Glimcher, P.: Expected subjective value theory (ESVT): A representation of decision under risk and certainty. Available at SSRN 2783638 (2022)
23. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J Res Dev 5, pp. 183–191 (1961) doi: 10.1147/rd.53.0183.
24. Steverson, K., Brandenburger, A., Glimcher, P.: Choice-theoretic foundations of the divisive normalization model. Journal of Economic Behavior & Organization, 164, pp. 148–165 (2019) doi: 10.1016/j.jebo.2019.05.026.
25. Tooby, J., Cosmides, L., Barrett, H.C.: The second law of thermodynamics is the first law of psychology: evolutionary developmental psychology and the theory of tandem, coordinated inheritances: comment on Lickliter and Honeycutt. Psychol Bull 129, pp. 858–865 (2003) doi: 10.1037/0033-2909.129.6.858.
26. Cosmides, L., Tooby, J.: Better than rational: Evolutionary psychology and the invisible hand. Am Econ Rev 84, pp. 327–332 (1994)
27. Cosmides, L., Barrett, H.C., Tooby, J.: Adaptive specializations, social exchange, and the evolution of human intelligence. Proc Natl Acad Sci USA 107, pp. 9007–9014 (2010) doi: 10.1073/pnas.0914623107.
28. Friston, K.J., Lin, M., Frith, C.D., Pezzulo, G., Hobson, J.A., Ondobaka, S.: Active inference, curiosity and insight. Neural Computation 29, pp. 2633–2683 (2017) doi: 10.1162/neco_a_00999.
29. Millidge, B.: Deep active inference as variational policy gradients. Journal of Mathematical Psychology 96, pp. 102348 (2020) doi: 10.1016/j.jmp.2020.102348.
30. Bogacz, R.: A tutorial on the free-energy framework for modelling perception and learning. J Math Psychol 76, pp. 198–211 (2017) doi: 10.1016/j.jmp.2015.11.003.
31. Kaplan, R., Friston, K.J.: Planning and navigation as active inference. Biol Cybernet 112, pp. 323–343 (2018) doi: 10.1007/s00422-018-0753-2.
32. Carandini, M., Heeger, D.J.: Normalization as a canonical neural computation. Nat Rev Neurosci 13, pp. 51–62 (2012) doi: 10.1038/nrn3136.
33. Da Costa, L., Friston, K., Heins, C., Pavliotis, G.A.: Bayesian mechanics for stationary processes. Proceedings of the Royal Society A 477(2256), pp. 20210518 (2021)
34. Von Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press, Princeton (1944)
35. Hemmatian, B., Varshney, L.R., Pi, F., Barbey, A.K.: The utilitarian brain: Moving beyond the Free Energy Principle. Cortex, 170, pp. 69–79 (2024) doi: 10.1016/ j.cortex.2023.11.013.
36. Louie, K., Grattan, L.E., Glimcher, P.W.: Reward value-based gain control: divisive normalization in parietal cortex. The Journal of Neuroscience, 31(29), pp. 10627–10639 (2011) doi: 10.1523/JNEUROSCI.1237-11.2011.
37. Reynolds, J.H., Heeger, D.J.: The normalization model of attention. Neuron 61(2), pp. 168–185 (2009)
38. Webb, R., Glimcher, P.W., Louie, K.: The normalization of consumer valuations: Context-dependent preferences from neurobiological constraints. Management Science 67(1), 93–125 (2021) doi: 10.1287/mnsc.2019.3557.

39. Kurtz-David, V., Sinha, S., Alladi, V., Bucher, S., Brandenburger, A., Louie, K., Glimcher, P., Tymula, A.: A Tale of Two Environments: Divisive Normalization and the Flexibility of Choice. bioRxiv (2024) doi: 10.1101/2024.08.25.609561.
40. LoFaro, T., Louie, K., Webb, R., Glimcher, P.W.: The Temporal Dynamics of Cortical Normalization Models of Decision-making. Letters in Biomathematics, 1(2), pp. 209–220 (2014). http://www.lettersinbiomath.org
41. Keung, W., Hagen, T. A., & Wilson, R. C.: A divisive model of evidence accumulation explains uneven weighting of evidence over time. Nature Communications, 11(1), pp. 2160 (2020) doi: 10.1038/s41467-020-15630-0.
42. Bossaerts, P., Murawski, C.: From behavioural economics to neuroeconomics to decision neuroscience: the ascent of biology in research on human decision making. Current Opinion in Behavioral Sciences, 5, pp. 37–42 (2015) doi: 10.1016/j.cobeha.2015.07.001.

ISSN 1870-4069

Machine Learning for Biomarker Identification in Ischemic Stroke Patients

Rodolfo Betanzos Cerqueda[1,3], Noé Macías Segura[2], Dulce Martinez-Peon[1,3], Rodrigo Sánchez Zavala[1], Fernando Góngora-Rivera[2], Christian Quintus Scheckhuber[4]

[1] Tecnológico Nacional de México, División de Estudios de Posgrado e Investigación, Mexico

[2] Universidad Autónoma de Nuevo León, Monterrey, Facultad de Medicina, Mexico

[3] Tecnológico Nacional de México, Departamento de Ingeniería Eléctrica y Electrónica, Mexico

[4] Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey, Mexico

{dd244881482, dulce.mp} @ nuevoleon.tecnm.mx

Abstract. Stroke is a medical condition that increasingly affects younger people around the world. Biomarkers are a helpful tool for diagnosing medical conditions based on genetic data. Typically, bioinformatics is used to identify which genes are candidates for a biomarker; however, this tool depends on linearly based algorithms. Machine Learning algorithms have been demonstrated to be useful for the detection of clue genes for certain diseases and medical conditions. In this work, we used a database from GEO containing data on stroke patients. We apply three algorithms—Support Vector Machines, Extreme Gradient Boosting, and Random Forest—to identify genes whose expression has a meaningful difference between the control group and stroke patients. The obtained results reveal sixteen genes: SVIL, C5AR1, MAX, KIF1B, ACOX1, PLXDC2, TNFRSF17, DOCK8, PHTF1, TRIB1, CREBBP, NPEPPS, RGS2, FAM108A3, ST8SIA4, and CD163.

Keywords: Gene expression, molecular pathways, biomarkers.

1 Introduction

Stroke is one of the leading causes of incapacity and death around the world. Each year, 12 million patients present this medical condition, and 6.5 million patients dies [1]. In Mexico, there are 118 stroke patients per 100,000 inhabitants, and 170,000 cases are reported annually, with around 36,000 deaths [2]. Among the potential risks for stroke

are diabetes, hypertension, obesity, and smoking [3]. Early detection is crucial to improving the prognosis. In this sense, recent tools like biomarkers are used to complement existing tools such as neuroimaging [4].

Gene expression through microarray systems has characterized neurological diseases and immune disorders [5]. It contains information about the RNA that is obtained from blood. Bioinformatic tools typically process this information. However, the data type provided by the microarrays has been tackled recently, with Machine Learning algorithms showing prominent results [6].

In this work, we used a public data set of Ischemic stroke patients and applied three ML algorithms, Support Vector Machines (SVM), Extreme Gradient Boosting (XGBoost), and Random Forest (RF), to identify genes with the highest expression in patients against a control group.

Recent advances in transcriptomics and machine learning have demonstrated the ability of ML algorithms to uncover complex biomarker signatures in ischemic stroke. For example, O'Connell et al. (2017) used ML to identify a peripheral blood gene-expression signature that diagnoses ischemic stroke with over 90 % sensitivity and specificity. More recently, Liu et al. (2024) applied Random Forest, Support Vector Machine (SVM), and XGBoost to coagulation-related gene expression, highlighting ACTN1, F5, and JMJD1C as robust diagnostic markers. These studies illustrate that ensemble methods and gradient-boosting frameworks excel at modeling nonlinear gene interactions in high-dimensional data. In a complementary proteomic approach, Dargazanli et al. (2020) leveraged SVM to differentiate cardioembolic from atherothrombotic thrombi, reinforcing the versatility of ML in stroke biomarker discovery. Unsupervised techniques—such as autoencoders and clustering—have also revealed hidden molecular subtypes among stroke patients (Liu et al., 2022; Burrello et al., 2022), paving the way for precision-medicine strategies. Collectively, this body of work justifies our selection of Random Forest, XGBoost, and SVM: these algorithms bring complementary strengths in handling noise, correcting misclassifications iteratively, and maximizing class separation, respectively, and each has demonstrated proven success in prior stroke-related omics investigations.

2 Materials and Methods

A dataset obtained from the Gene Expression Omnibus (GEO) database was utilized to reanalize. Methods employed are described and detailed in this section including the ML algorithms, to identify biomarkers and examine differential gene expression in patients diagnosed with ischemic stroke disease, and the statistical modelling approaches used to evaluate, see Fig. 1.

2.1 Data Acquisition and Preprocessing

We used the GSE16561 series (Accession: GSE16561) from NCBI's Gene Expression Omnibus, which comprises 63 peripheral whole-blood RNA samples—39 acute ischemic stroke patients (MRI-confirmed, >18 years, collected in PAXgene Blood RNA tubes) and 24 neurologically healthy controls—profiled on the Illumina HumanRef-8 v3.0 expression beadchip (GPL6883). The dataset comprises 39 patients

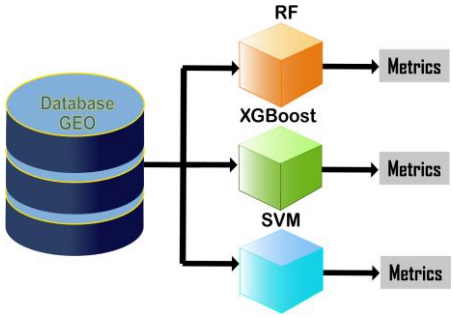

Fig. 1. Block diagram of the processing for gene data.

with ischemic stroke (IS) and 24 control individuals [5,7,8]. The process involved the removal of missing values and the transformation of the data into a numerical structure.

The analysis did not involve filtering out low-variance genes; therefore, all genes were included. This approach is justified by the aim of the study, which was to directly assess the ability of supervised models to identify the most relevant features. Applying an initial variance-based filtering step could potentially remove genes that, despite exhibiting low overall variability, may still be discriminative for classification purposes.

In this study, the dataset was already normalized in its original format; therefore, no additional normalization procedures were implemented during preprocessing. The assessment of gene expression is constrained by the high dimensionality of the data, as a single experiment may encompass tens of thousands of genes as predictive variables. To address this challenge, feature selection methods based on supervised machine learning models were employed, enabling the reduction of the gene set to those contributing the most relevant information for the classification of patients with stroke.

2.2 Classification Models

Three representative classification approaches were implemented.

Random Forest (RF): This ensemble model utilizes multiple decision trees to generate predictions. Gene importance is assessed by measuring the contribution of each gene to the overall improvement in classification accuracy.

XGBoost (Extreme Gradient Boosting): This advanced tree-based model builds predictive estimators through iterative boosting. Gene importance is evaluated based on how significantly each gene contributes to the quality of the model's decision-making during classification.

Support Vector Machine (SVM) with Linear Kernel: This algorithm identifies the optimal hyperplane that separates patients with cerebral infarction from control subjects. The relevance of each gene is determined by analyzing the magnitude of its contribution to this separation.

These models were selected due to their distinct strategies for capturing complex relationships within the data. RF is effective when genes interact in intricate, non-linear ways that do not conform to simple patterns. It is a robust model and performs well in noise or data imperfections. XGBoost is an advanced model that incrementally improves accuracy by correcting errors at each stage of learning. This iterative

Table 1. Comparative strengths and gene importance criteria of the classification models used in this study.

Model	Main Strengths	Gene importance criteria
Random Forest	-Handles nonlinear interactions between genes. -Robust to noise and overfitting -Works well with high-dimensional data	Mean decrease in impurity (e.g., Gini): measures how much each gene reduces classification error across trees.
XGBoost	-Highly accurate through iterative boosting -Efficient and scalable -Corrects misclassifications at each stage of training	Gain: evaluates the improvement in classification accuracy when a gene is used to split decision tree nodes
Linear SVM	-Finds optimal linear separation between classes -Performs well with high-dimensional, sparse data -Easy to interpret in linear form	Magnitude of model coefficients: genes with larger weights contribute more to the separation hyperplane

enhancement makes it highly efficient and capable of achieving high classification accuracy. Linear SVM identifies the optimal combination of genes that clearly distinguishes between patients with cerebral infarction and healthy individuals.

Specifically, Random Forest was selected because it handles high-dimensional data well and can model complex, nonlinear interactions among genes without overfitting, making it robust to noisy microarray measurements. XGBoost was included for its state-of-the-art gradient-boosting framework, which iteratively corrects classification errors and incorporates regularization to prevent overfitting, delivering both high accuracy and scalability on large feature sets. Finally, a linear Support Vector Machine was employed due to its proven effectiveness in high-dimensional, sparse settings like gene expression, where it finds the optimal separating hyperplane and yields easily interpretable feature weights. Together, these methods represent complementary approaches—ensemble averaging, boosting, and maximum-margin classification—that ensure a comprehensive evaluation of biomarker relevance across different modeling paradigms.

Table 1 provides a summary of the key strengths of each classification model used in this study.

2.3 Feature Importance Estimation

Each model evaluates gene importance according to its underlying criteria. RF determines the relevance of each gene by assessing how much it contributes to improving classification accuracy at each stage. If the removal of a gene leads to a significant drop in accuracy, the gene is considered essential. XGBoost measures gene

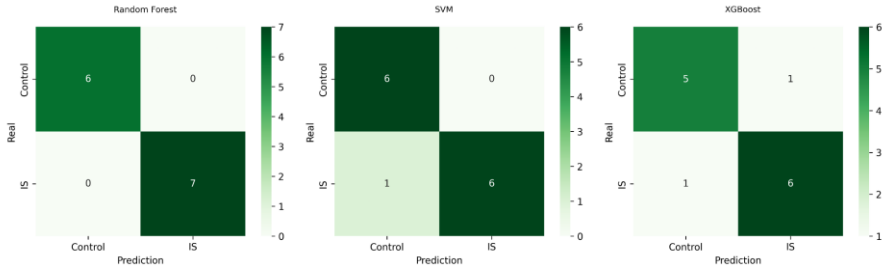

Fig. 2. Confusion matrix resume.

importance by analyzing how frequently a gene is used at critical decision points in the model. A gene repeatedly selected and enhancing prediction performance is deemed highly informative. Linear SVM identifies the optimal boundary for separating patients from controls and assigns a weight to each gene. Genes with higher absolute weights exert a more significant influence on the classification outcome.

To ensure the validity of the findings, potential biomarkers were defined as those genes that appeared among the top 100 ranked features in at least two or more models. This criterion suggests that their importance is not tied to a single method but demonstrates consistency across multiple analytical approaches.

2.4 Metrics

To evaluate the classification performance of each machine learning model, confusion matrices were generated for the top 100 most important genes selected by each method. These matrices illustrate the number of true positives, true negatives, false positives, and false negatives in distinguishing ischemic stroke (IS) patients from control group.

3 Results

Fig. 2 to Fig. 6 show the results obtained.

The RF model achieved perfect classification performance, correctly identifying all control and IS samples (6 true negatives, 7 true positives), resulting in 100% accuracy. The SVM with a linear kernel also showed high performance, correctly classifying all control samples and misclassifying only one IS case as control, achieving an overall accuracy of 92.9%. The XGBoost model correctly identified 5 out of 6 control subjects and 6 out of 7 IS patients, with one false positive and one false negative, yielding an overall accuracy of 85.7%.

These results demonstrate the high discriminative power of the selected gene subsets across different classifiers, with Random Forest showing the most robust performance in this experimental setup.

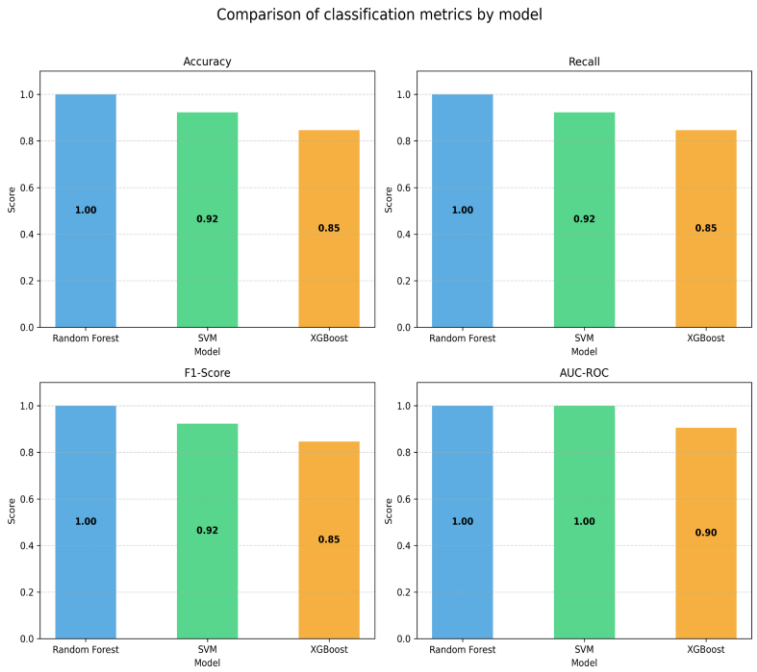

Fig. 3. Comparison of classification metrics by model.

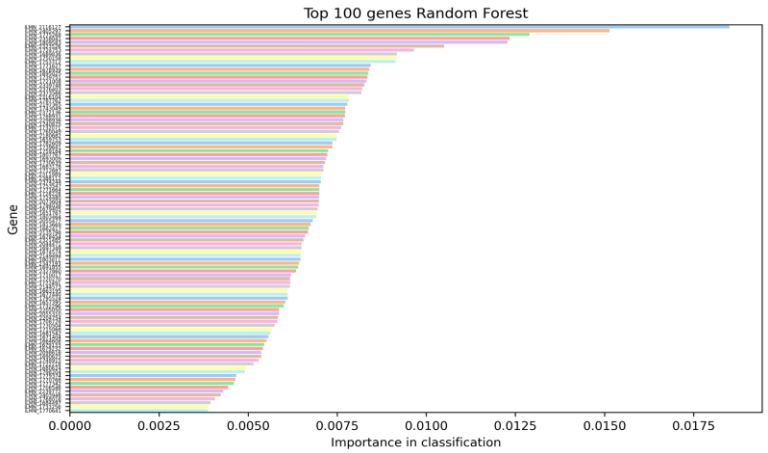

Fig. 4. Top 100 most important genes identified by the Random Forest model.

To complement the confusion matrix analysis, additional classification metrics were calculated for each model, including precision, recall, F1-score, and area under the ROC curve (AUC-ROC). As shown in the table below, Random Forest achieved perfect scores across all metrics, while SVM and XGBoost also demonstrated strong performance with slightly lower values in class-specific precision and recall. These

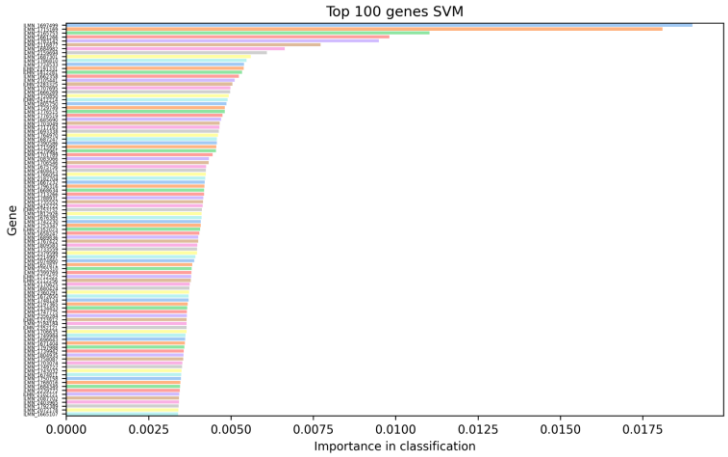

Fig. 5. Top 100 most important genes identified by the SVM model.

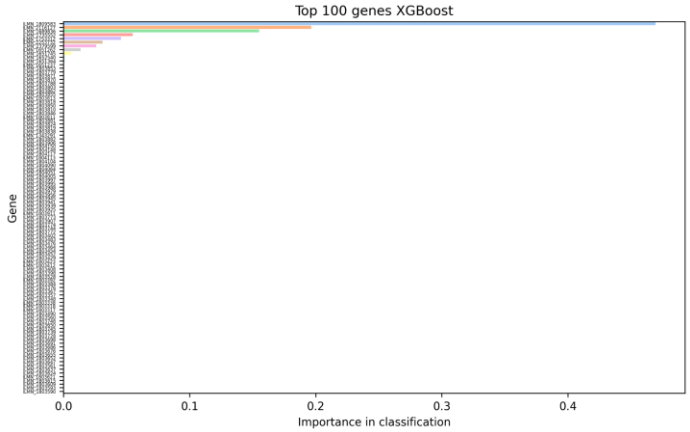

Fig. 6. Top 100 most important genes identified by the XGBoost model.

metrics reinforce the conclusions drawn from the confusion matrices, confirming that Random Forest provided the most robust classification among the three methods evaluated.

3.1. Gene Importance Comparison

The relative importance of genes in classification was analyzed across the three machine learning models: Random
Forest, Support Vector Machine (SVM), and XGBoost. Each model ranked the top 100 genes according to its internal criteria for feature relevance. The Random Forest model distributed importance more evenly across many genes, suggesting a broader contribution of features to classification decisions.

In contrast, the SVM model exhibited a steep decline in importance values, with only a small subset of genes carrying significantly higher weights, indicating a more selective dependence on a limited number of features.

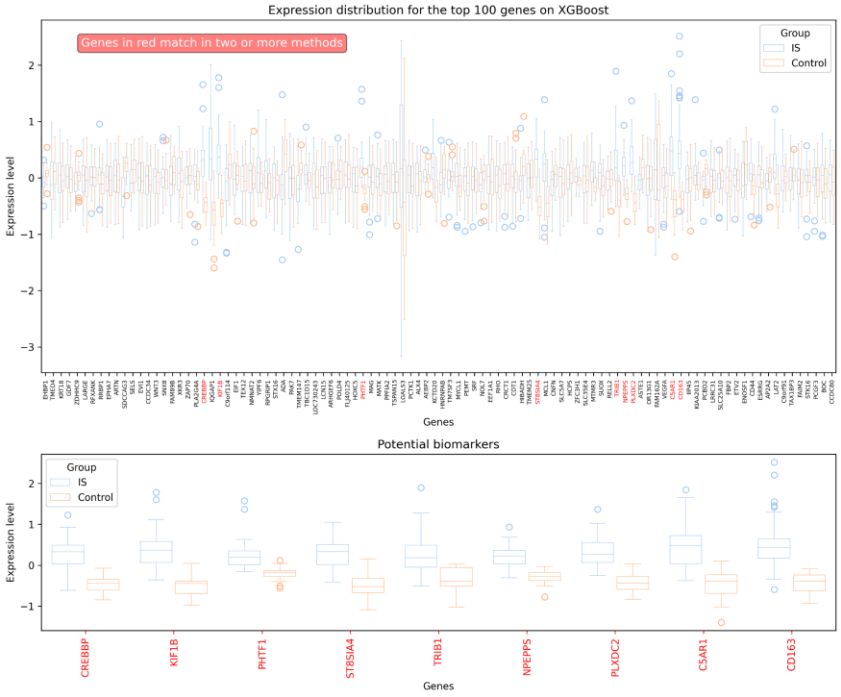

Fig. 7. Gene expression distribution for top 100 genes identified by XGBoost. Genes highlighted in red are shared by two or more models and considered potential biomarkers.

XGBoost demonstrated a highly concentrated importance profile, where only a few genes dominated the classification decision-making process. The top-ranked gene in the XGBoost model contributed disproportionately more to the overall model accuracy compared to the rest.

These patterns reflect the intrinsic characteristics of each algorithm. Random Forest benefits from ensemble averaging and tends to distribute importance broadly. SVM, being a linear classifier, identifies the most discriminative directions in the feature space.

XGBoost, as a boosting-based method, favors feature that provide the highest gain at each step of model construction. A comparison of gene rankings among models also revealed overlapping genes, which reinforces the robustness of the identified biomarkers and supports their biological relevance in the context of ischemic stroke classification.

3.2 Selection of Genes Repeated on at least Two Methods

To enhance the reliability of biomarker discovery, genes that appeared among the top 100 features in at least two out of the three models (Random Forest, SVM, and XGBoost) were considered potential biomarkers. This criterion ensures that gene

Fig. 8. Gene expression distribution for top 100 genes identified by SVM. Potential biomarkers shared across models are labeled in red.

importance is consistent across multiple learning paradigms, minimizing model-specific biases.

3.2.1 XGBoost

In the XGBoost model, several genes marked in red were identified as recurring across multiple methods. These genes showed consistent expression differences between ischemic stroke (IS) patients and control subjects. The bottom plot highlights the expression distributions for those overlapping genes, revealing clear group separation in many cases.

3.2.2 Support Vector Machine (SVM)

The SVM model revealed a broader distribution of gene expression values, and many of the recurring genes also showed distinctive expression profiles. Notably, genes such as CD163, CREBBP, and C5AR1 demonstrated clear upregulation or downregulation patterns in the IS group.

3.2.3. Random Forest

Random Forest provided a more balanced view, with overlapping genes such as PLXDC2, RGS2, TRIB1, and SVIL showing distinguishable expression levels between

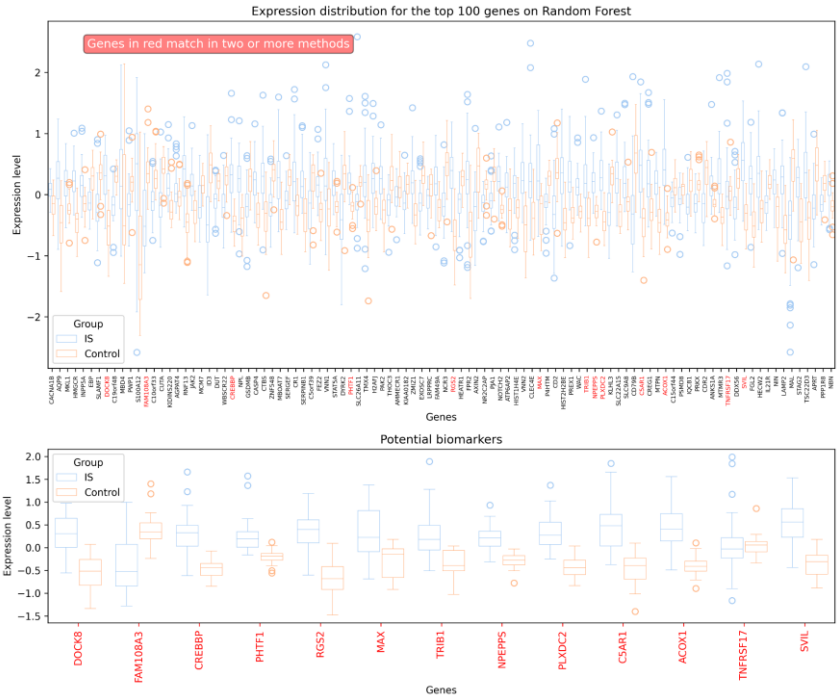

Fig. 9. Gene expression distribution for top 100 genes identified by Random Forest. Highlighted genes in red appeared in at least two models.

IS and control groups. These expression profiles further support their candidacy as robust biomarkers.

Genes that appeared in two or more methods were identified and considered potential biomarkers in this study.

The selected genes are: SVIL, C5AR1, MAX, KIF1B, ACOX1, PLXDC2, TNFRSF17, DOCK8, PHTF1, TRIB1, CREBBP and NPEPPS.

3.3. Molecular Pathway Analysis

To gain deeper insight into the biological functions and interactions of the identified genes, a network-based molecular pathway analysis was performed using the GeneMANIA platform. This tool integrates data from multiple sources to predict gene-gene interactions based on co-expression, co-localization, physical interactions, and genetic interactions.

The resulting network, shown in Fig. 10, reveals a densely interconnected structure among the selected genes. Notably, CD163, TRIB1, CREBBP, and C5AR1 emerge as central nodes, suggesting that they play pivotal regulatory roles in ischemic stroke pathology. CD163, a scavenger receptor expressed in monocytes and macrophages, contributes to anti-inflammatory responses following tissue injury. TRIB1 participates in lipid metabolism and macrophage polarization, processes closely linked to vascular

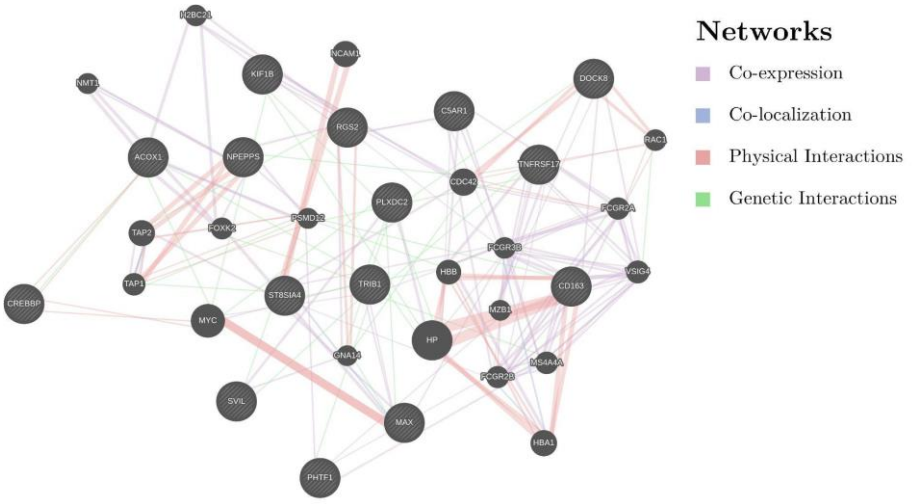

Fig. 10. Gene interaction network generated using GeneMANIA, illustrating functional associations among the selected candidate biomarkers for ischemic stroke. Edge colors represent different types of interactions: co-expression (purple), co-localization (blue), physical interactions (red), and genetic interactions (green).

inflammation. CREBBP, known as a transcriptional coactivator, modulates the expression of genes involved in cell survival and immune regulation. Meanwhile, C5AR1 acts as a receptor for complement component C5a and is strongly implicated in neuroinflammation and ischemia-reperfusion injury.

Edge colors in the network indicate different types of interactions: co-expression (purple edges) reflects genes expressed simultaneously under similar conditions, suggesting shared regulatory mechanisms; co-localization (blue edges) indicates that genes are located within the same cellular compartments, implying potential cooperation in localized biological processes; physical interactions (red edges) represent direct binding between protein products; and genetic interactions (green edges) highlight functional interdependencies inferred from genetic perturbation studies.

4 Discussion and Conclusions

This study enabled the identification of potential biomarkers with diagnostic and prognostic relevance for ischemic stroke. Machine learning models facilitated the evaluation of gene expression data, and their integration with molecular pathway analysis provided a more comprehensive perspective on the underlying biological mechanisms.

The findings suggest that the implementation of machine learning methodologies not only enhances the accuracy of biomarker detection but also simplifies the biological interpretation of genes involved in the pathology. Future research will aim to expand

the analysis by incorporating a larger patient cohort and exploring model interpretation techniques to optimize the understanding of the relationship between the identified genes and the disease.

In addition, the integration of other omics data types—such as proteomics or metabolomics—is envisioned to achieve a more holistic understanding of ischemic stroke biology and to support the development of more accurate diagnostic tools.

The presence of multiple interaction types among these key genes reinforces their biological relevance and underscores a cooperative molecular framework underlying ischemic stroke. This network-based systems biology approach points to CD163, TRIB1, CREBBP, and C5AR1 as promising biomarkers and potential targets for therapeutic intervention.

References

1. GBD Stroke Collaborators: Global, regional, and national burden of stroke and its risk factors, 1990–2019: A systematic analysis for the Global Burden of Disease Study 2019. Lancet Neurology, 20(10), pp. 795–820 (2021) doi: 10.1016/S1474-4422(21)00252-0.
2. Jiménez Muñiz, V.E.: Accidente cerebrovascular, cuarta causa de muerte en México en mayores de 45 años. Universidad de Guadalajara. Accidente cerebrovascular, cuarta causa de muerte en México en mayores de 45 años, Universidad de Guadalajara (2024).
3. Mitchell, A.B., Cole, J.W., McArdle, P.F., Cheng, Y.C., Ryan, K.A., Sparks, M.J., Kittner, S.J.: Obesity increases risk of ischemic stroke in young adults. Stroke, 46(6), pp. 1690–1692 (2015)
4. Mendioroz Iriarte, M., Cuadrado Godia, E., & Montaner Villalonga, J.: Biomarcadores plasmáticos en la enfermedad vascular cerebral isquémica [Plasma biomarkers in ischemic cerebral vascular disease]. Hipertensión, 26(6), pp. 266–274 (2009) doi: 10.1016/j.hipert.2008.07.001.
5. Barr, T.L., Conley, Y., Ding, J., Dillman, A., Warach, S., Singleton, A., Matarin, M.: Genomic biomarkers and cellular pathways of ischemic stroke by RNA gene expression profiling. Neurology, 75(11), pp. 1009–1014 (2010)
6. Tabl, A.A., Alkhateeb, A., ElMaraghy, W., Rueda, L., Ngom, A.: A machine learning approach for identifying gene biomarkers guiding the treatment of breast cancer. Frontiers in genetics, 10, pp. 256 (2019)
7. O'Connell G.C, Treadway M.B, Petrone A.B, Tennant C.S, et al.: Peripheral blood AKAP7 expression as an early marker for lymphocyte-mediated post-stroke blood brain barrier disruption. Sci Rep 7(1), pp. 1172, PMID: 28446746 (2017)
8. O'Connell G.C, Petrone A.B, Treadway M.B, Tennant C.S et al.: Machine-learning approach identifies a pattern of gene expression in peripheral blood that can accurately detect ischaemic stroke. NPJ Genom Med 2016(1), pp. 16038, PMID: 29263821 (2016)
9. O'Connell, G.C., Chantler, P.D., Barr, T.L.: Stroke-associated pattern of gene expression previously identified by machine-learning is diagnostically robust in an independent patient population. Genomics Data, 14, pp. 47–52 (2017) doi: 10.1016/j.gdata.2017.08.006.
10. Liu, J., Si, Z., Liu, J., Zhang, X., Xie, C., Zhao, W., Wang, A., Xia, Z.: Machine learning identifies novel coagulation genes as diagnostic and immunological biomarkers in ischemic stroke. (2024) doi: 10.18632/aging.205706.
11. Dargazanli, C., Zub, E., Deverdun, J., Decourcelle, M., De Bock, F., Labreuche, J., Lefèvre, P.H., Gascou, G., Derraz, I., Riquelme Bareiro, C., Cagnazzo, F., Bonafé, A., Marin, P., Costalat, V., Marchi, N.: Machine Learning Analysis of the Cerebrovascular Thrombi Proteome in Human Ischemic Stroke: An Exploratory Study. Frontiers in Neurology, 11, pp. 575376 (2020) doi: 10.3389/fneur.2020.575376.

12. Liu, J., Chou, E.L., Lau, K.K., Woo, P.Y.M., Li, J., Chan, K.H.K.: Machine Learning Algorithms Identify Demographics, Dietary Features, and Blood Biomarkers Associated with Stroke Records. Journal of the Neurological Sciences, 440, pp. 120335 (2022) doi: 1016/j.jns.2022.120335.
13. Burrello, J., Burrello, A., Vacchi, E., Bianco, G., Caporali, E., Amongero, M., Airale, L., Bolis, S., Vassalli, G., Cereda, C. W., Mulatero, P., Bussolati, B., Camici, G. G., Melli, G., Monticone, S., Barile, L.: Supervised and unsupervised learning to define the cardiovascular risk of patients according to an extracellular vesicle molecular signature. Translational Research, 244, pp. 114-125 (2022) doi: 10.1016/j.trsl.2022.02.005.

ISSN 1870-4069

Análisis de patrones comerciales mediante técnicas avanzadas: Redes neuronales y análisis estadístico aplicado a datos públicos

Manuel Torres-Vásquez[1,2], Estefanía de-la-Cruz-Bautista[1], Cesar Alejandro Lara-Ramirez[1], Susana Chávez-Cruz[1]

[1] Universidad Mundo Maya, campus Villahermosa, Jefatura Académica de Arquitectura, Ingenierías y Computación, México

[2] Tecnológico Nacional de México campus Centla, División Sistemas Computacionales, México

{vlicd2231002@universidadmundomaya, vlicd2231001@universidadmundomaya}.edu.mx, susanachavez@umma.com.mx, manuel.torres@centla.tecnm.mx

Resumen. Este estudio analiza los patrones comerciales del Brazilian E-Commerce Public Dataset, un conjunto de datos público proporcionado por la empresa Olist. Para la predicción de ventas diarias, se aplicaron los modelos estadísticos ARIMA y Prophet, así como redes neuronales recurrentes con neuronas LSTM. Se llevó a cabo un preprocesamiento del conjunto de datos original, seguido del ajuste de los modelos, la evaluación de su desempeño y el análisis de errores. Los resultados indican que los modelos estadísticos presentan limitaciones, reflejadas en errores elevados en sus predicciones. En contraste, el uso de redes neuronales recurrentes mejoró significativamente la precisión de las predicciones. Esta diferencia se atribuye a la alta variabilidad de los datos, la posible presencia de patrones no estacionarios y la falta de componentes estructurales adecuados en los modelos estadísticos empleados.

Palabras clave: Series temporales, redes neuronales, predicción, comercio electrónico.

Analysis of Business Patterns Using Advanced Techniques: Neural Networks and Statistical Analysis Applied to Public Data

Abstract. This study analyzes the commercial patterns of the Brazilian E-Commerce Public Dataset, a public dataset provided by the company Olist. For the prediction of daily sales, ARIMA and Prophet statistical models were applied, as well as recurrent neural networks with LSTM neurons. Preprocessing of the original data set was carried out, followed by model fitting, performance evaluation and error analysis. The results indicate that the statistical models have

limitations, reflected in high errors in their predictions. In contrast, the use of recurrent neural networks significantly improved prediction accuracy. This difference is attributed to the high variability of the data, the possible presence of non-stationary patterns and the lack of adequate structural components in the statistical models used.

Keywords: Time series, neural networks, prediction, e-commerce.

1 Introducción

Anticipar la demanda de un producto puede marcar la diferencia entre el éxito y la ineficiencia operativa. En un mundo cada vez más impulsado por datos, el uso de Machine Learning para predecir ventas permite a las empresas optimizar la planificación de inventarios y mejorar la gestión de sus operaciones. A través de modelos avanzados de pronóstico, es posible analizar patrones de consumo y ajustar la disponibilidad de productos para satisfacer mejor las necesidades de los clientes. En este estudio, se analizan los patrones de distintos modelos aplicados a un conjunto de datos de ventas, evaluando sus limitaciones y determinando los factores que influyen en su rendimiento. Para ello, se emplean, un conjunto de datos públicos proporcionado por la empresa Olist del Brazilian E-Commerce Public Dataset [1]. Las series temporales son secuencias de datos organizadas de manera cronológica y equidistante en el tiempo, utilizadas en diversas áreas como el análisis del turismo en una región, el impacto del ruido urbano en la salud, el estudio de imágenes satelitales y la predicción de ventas para optimizar la logística de productos. Uno de los modelos más utilizados en el análisis de series temporales es ARIMA (Autoregressive Integrated Moving Average), el cual combina tres componentes principales: Autorregresivo (AR), Integrado (I) y Media Móvil (MA). Sus parámetros incluyen: p, que representa los retardos autorregresivos; q, que indica el componente de media móvil; y d, que define el orden de diferenciación necesario para estabilizar la serie [2].

Por otro lado, Prophet es un algoritmo de regresión aditiva desarrollado por Facebook, diseñado para modelar tendencias de crecimiento lineal o logístico por partes. Su enfoque se basa en la descomposición de los datos en componentes de tendencia, estacionalidad y efectos de días festivos. Prophet incluye un componente estacional anual, modelado mediante series de Fourier, y un componente estacional semanal, representado con variables ficticias. Sin embargo, su eficacia depende de la presencia de patrones bien definidos en los datos, lo que puede afectar su rendimiento en conjuntos de datos con alta variabilidad o ruido.

Finalmente, las redes neuronales artificiales (RNA) se inspiran en la estructura de las redes neuronales biológicas del cerebro humano y están compuestas por capas de nodos interconectados de manera jerárquica. Son especialmente útiles en problemas donde la formulación explícita de restricciones lógicas es compleja, como el reconocimiento de patrones y el análisis predictivo. Dentro de este campo, las Long Short-Term Memory (LSTM) representan una variante avanzada de las redes neuronales recurrentes, diseñada específicamente para mitigar el problema de la dependencia a largo plazo. Gracias a su arquitectura especializada, las LSTM pueden retener información durante períodos extensos, lo que las hace particularmente eficaces

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at
0	5f79b5b0931d63f1a42989eb65b9da6e	00012a2ce6f8dcda20d059ce98491703	delivered	2017-11-14 16:08:26	2017-11-14 16:35:32
1	a44895d095d7e0702b6a162fa2dbeced	000161a058600d5901f007fab4c27140	delivered	2017-07-16 09:40:32	2017-07-16 09:55:12
2	316a104623542e4d75189bb372bc5f8d	0001fd6190edaaf884bcaf3d49edf079	delivered	2017-02-28 11:06:43	2017-02-28 11:15:20
3	5825ce2e88d5346438686b0bba99e5ee	0002414f95344307404f0ace7a26f1d5	delivered	2017-08-16 13:09:20	2017-08-17 03:10:27
4	0ab7fb08086d4af9141453c91878ed7a	000379cdec625522490c315e70c7a9fb	delivered	2018-04-02 13:42:17	2018-04-04 03:10:19

order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date	order_item_id	product_id
2017-11-17 15:32:08	2017-11-28 15:41:30	2017-12-04 00:00:00	1.0	64315bd8c0c47303179dd2e25b579d00
2017-07-19 19:09:37	2017-07-25 18:57:33	2017-08-04 00:00:00	1.0	84183944dc7cddca87a5d384452c1d3c
2017-03-01 15:24:20	2017-03-06 08:57:49	2017-03-22 00:00:00	1.0	9df2b21ec85378d71df4404712e17478
2017-08-19 11:34:29	2017-09-13 20:06:02	2017-09-14 00:00:00	1.0	af3ec22cce878225aae6d9eb6c7a78eb
2018-04-04 18:11:09	2018-04-13 20:21:08	2018-04-18 00:00:00	1.0	868b3136c5b206f91b8208fbfdf2cb7c

Fig. 1. Fragmento del conjunto de datos unificado de los datos proporcionados por la empresa Olist, uniendo las tablas originales usando los campos clave.

en el procesamiento de datos secuenciales, como series temporales y conjuntos de datos discretos.

2 Trabajos relacionados

La literatura especializada confirma que los LSTM superan a los enfoques clásicos de pronóstico al capturar patrones estacionales, promociones y efectos de calendario en conjuntos de ventas reales [3]. Ellos demostraron que una red LSTM con funciones de costo personalizadas y optimización genética redujo los costos de inventario de una maderería mexicana.

En [4] validó el modelo en un contexto de comercio electrónico ruso, mejorando el MAPE frente a un ARIMA base en más del 20 %. Estos trabajos coinciden en que, al combinar normalización adecuada, dropout y búsqueda sistemática de hiperparámetros, las arquitecturas LSTM generalmente de una a tres capas con 32 a 128 unidades ofrecen un equilibrio sólido entre precisión y costo computacional para la planificación de ventas en distintos sectores como la manufactura, retail y el comercio electrónico [5].

3 Materiales y métodos

El presente estudio se basa en el conjunto de datos públicos Brazilian E-Commerce proporcionado por la empresa Olist. El dataset contiene información de 100,000 pedidos del 2016 a 2018 realizados en varios mercados de Brasil. Los datos incluyen múltiples tablas interrelacionadas con detalles de pedidos, Items vendidos, Pagos, Reseñas de clientes, Datos de clientes y Datos geoespaciales.

Para la red neuronal

- **Agregación temporal:** Se transformaron los datos a una representación de serie de tiempo diaria por categoría de producto. Específicamente, a partir de la fecha y hora de compra de cada pedido (order_purchase_timestamp), se extrajo la fecha y se contabilizó el número de productos vendidos por categoría en cada día. Agrupamos las ventas por fecha y por nombre de categoría de producto, y se calculó el total de ventas diarias por categoría (campo ventas_totales). De este modo, para cada categoría de producto obtuvimos una secuencia temporal que refleja cuántos productos de esa categoría se vendieron cada día. Cabe destacar que, si una categoría no tuvo ventas en cierto día, dicha fecha no aparecía en la agrupación inicial; para asegurar continuidad temporal, se podrían imputar valores cero en días sin ventas, garantizando series cronológicas completas. Cada serie resultante quedó identificada por su categoría de producto[6].

Características de temporalidad (estacionalidad): Con el fin de ayudar al modelo a captar patrones estacionales o repetitivos en las series (ciclos semanales, tendencias anuales), se agregaron variables sintéticas periódicas derivadas de la fecha. Para cada registro diario se calculó:

- una representación cíclica del año en curso mediante $\sin(2\pi t/\text{año})$ y $\cos(2\pi t/\text{año})$,
- una representación del mes dentro del año (ciclo anual subdividido en 12) mediante $\sin(2\pi t/\text{mes})$ y $\cos(2\pi t/\text{mes})$, y
- una representación de la semana del año mediante $\sin(2\pi t/\text{semana})$ y $\cos(2\pi t/\text{semana})$. Aquí t representa la fecha en formato numérico (p. ej., timestamp Unix) y los denominadores corresponden a la duración de un año, mes o semana en la misma unidad temporal. Codifican de manera continua la posición del día dentro de cada ciclo temporal, permitiendo al modelo discernir, por ejemplo, qué tan avanzado está el año o si un dato corresponde a inicios o fines de semana. Se eliminó la columna original de order_date del conjunto de datos, ya que en su forma de entero no aporta significado directo y sus efectos de temporalidad ahora están capturados por las nuevas variables.

Codificación de la categoría: A fin de que el modelo de redes neuronales distinga cuál es la categoría que está procesando, se añadió una codificación one-hot de la categoría de producto. En la representación tabular agregada, esto implicó crear columnas binarias para cada categoría única, inicializándolas en 0. Para cada serie de categoría, se marcó con valor 1 la columna correspondiente a esa categoría y 0 las de las demás categorías. Por ejemplo, la serie de la categoría toys tendrá un campo toys=1 en todas sus filas, mientras que health_beauty=0, baby=0, etc., y la serie de health_beauty tendrá health_beauty=1 y las demás 0, y así sucesivamente. Esta codificación sirve como entrada al modelo para indicarle explícitamente de qué categoría son los datos de esa secuencia.

('toys',

	order_date	ventas_totales	toys	health_beauty	baby	cool_stuff	\
0	1475452800	0.000000	1	0	0	0	
1	1475539200	0.067568	1	0	0	0	
2	1475625600	0.081081	1	0	0	0	
3	1475712000	0.027027	1	0	0	0	
4	1475798400	0.054054	1	0	0	0	

	bed_bath_table	sports_leisure	fashion_bags_accessories	pet_shop	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	garden_tools	furniture_decor	telephony	housewares	consoles_games	'
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

Fig. 2. Fragmento del conjunto de datos de entrenamiento previo a la inclusión de las características de temporalidad y su segmentación en ventanas temporales

Normalización de valores: Dada las variaciones significativas en el volumen de ventas entre categorías, donde algunas presentan valores máximos diarios sustancialmente mayores, se aplicó una normalización min-max a cada serie de ventas de forma independiente. Utilizando la función MinMaxScaler de la biblioteca scikit-learn, el campo ventas_totales de cada serie fue escalado al rango [0,1]. Este procedimiento permite expresar cada serie temporal en términos de la proporción respecto a su propio valor máximo, evitando que las categorías con mayores volúmenes absolutos sesguen el proceso de aprendizaje del modelo. Finalmente, en la tabla procesada, el campo ventas_totales normalizado reemplazó el conteo original de ventas diarias, asegurando una representación homogénea entre categorías.

Segmentación en ventanas temporales: Los datos fueron preparados en formato de series de tiempo, con ventas diarias normalizadas, indicadores de categoría y atributos estacionales, completando así el conjunto de datos. Se procedió a construir los ejemplos de entrenamiento para la red neuronal en forma de secuencias. Se definió una ventana de entrada de 90 días y una ventana de predicción (salida) de 30 días, valores escogidos con base en la granularidad del negocio (un horizonte de un mes para previsión) y experimentación preliminar. A partir de cada serie diaria por categoría, se extrajeron pares de secuencia de entrada. Cada secuencia de entrada es una matriz de 90 filas (días) por N columnas de características. En este caso, las columnas incluyen:

- la venta diaria normalizada (ventas_totales escalada) de ese día,
- el indicador one-hot de cada categoría (71 columnas, de las cuales solo una tendrá valor 1 y las demás 0, determinado por la categoría de la serie), y
- las 6 columnas de seno/coseno anuales, mensuales y semanales para ese día. En total, la dimensión de características N es 1 (venta) + 71 (categorías) + 6 (estacionalidad) = 78 características por día. La etiqueta o secuencia de salida correspondiente es un vector de 30 valores que representa las ventas normalizadas de los 30 días siguientes (misma serie y categoría). Este procedimiento de ventaneo deslizante se repitió para todas las series de todas las categorías, generando un amplio conjunto de pares entrada-salida. Como resultado final del preprocesamiento, se obtuvo un dataset de aprendizaje compuesto por x

(secuencias de 90×78) y y (vectores de 30×1) que reúne información de todas las categorías de producto.

Para la construcción y entrenamiento del modelo de red neuronal, se utilizó Python 3 como lenguaje de programación. La manipulación de datos, la combinación de tablas y agregaciones por fecha se realizó con Pandas, mientras que NumPy se aplicó para las operaciones numéricas de bajo nivel. El preprocesamiento incluyó la normalización Min-Max y la división del conjunto de datos en entrenamiento y prueba, utilizando la librería sk-learn.

La implementación de la red neuronal recurrente se llevó a cabo con TensorFlow v2 y su API de alto nivel Keras, utilizando capas LSTM y rutinas de entrenamiento optimizadas para GPU. El entrenamiento se ejecutó en un entorno Kaggle con aceleración mediante una GPU NVIDIA Tesla T4, aplicando la estrategia de distribución tf.distribute.MirroredStrategy para la paralelización del cálculo. Este enforque permitió entrenar eficientemente el modelo a pesar del gran volumen de secuencias generadas[7].

Se empleó una red neuronal recurrente LSTM de una sola capa. La arquitectura del modelo consta de una capa LSTM con 32 unidades de memoria tal como Deng en [3] señala que "cuando el número de neuronas ocultas de un LSTM es pequeño, existen ventajas como una alta eficiencia computacional y la prevención del sobreajuste". Estas capas reciben secuencias de entrada de 90 pasos temporales con 78 características por paso. Esta capa procesa la secuencia completa y genera un vector de estado interno de tamaño 32 al final (return_sequences=False), extrayendo únicamente la salida del último paso.

La representación interna obtenida se transfiere a una capa Dense totalmente conectada con 30 neuronas de salida lineales, donde cada neurona predice las ventas normalizadas para un día en la ventana de salida de 30 días. Los pesos iniciales de esta capa se establecieron en cero para generar predicciones iniciales neutras, cercanas al promedio de la escala. No se aplicó función de activación en la capa de salida, ya que el modelo realiza una regresión sobre valores continuos en el rango [0,1].

El modelo se compiló utilizando el Error Cuadrático Medio (MSE) como función de perdida, adecuada para problemas de regresión continua, y el Error Absoluto Medio (MAE) como métrica de evaluación. Se empleo el optimizador Adam, un algoritmo de descenso de gradiente estocástico adaptativo ampliamente utilizado por su eficiente convergencia.

Para prevenir el sobreajuste, se implementó la técnica de early stopping, configurando un monitoreo de la pérdida en el conjunto de validación. El conjunto de datos se dividió aleatoriamente en 90% para entrenamiento y 10% para prueba mediante train_test_split (test_size=0.1). Este mismo 10% se usó como conjunto de prueba final y como validación durante el entrenamiento.

El modelo se entrenó por un máximo de 20 épocas, con un tamaño de lote predeterminado por TensorFlow. En cada época, el modelo procesó todas las secuencias de entrenamiento y ajustó sus pesos para minimizar la pérdida MSE. Aunque early stopping podía interrumpir el entrenamiento antes de las 20 épocas si la pérdida en validación dejaba de mejorar, en la práctica, el modelo completó todas las iteraciones, ya que la pérdida siguió disminuyendo gradualmente. Finalmente, el modelo ajustado se guardó para su posterior evaluación en el conjunto de prueba.

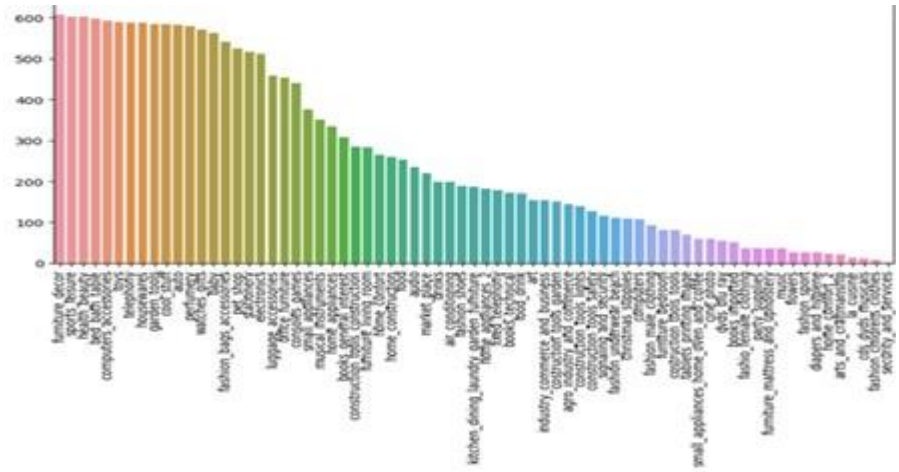

Fig. 3. Visualización por categorías la cantidad de ventas totales, ordenado por la categoría más vendida, a lo largo del periodo de tiempo que presenta el conjunto de datos.

El modelo ARIMA fue construido y ajustado utilizando Python 3, la manipulación de datos, la conversión de fechas de formato Unix a datatime y la agrupación de registros diarios se empleó Pandas. Las operaciones numéricas de bajo nivel aplicamos NumPy. La implementación del modelo se realizó con la biblioteca statsmodels, que facilita la definición, ajuste y evaluación de modelos de series temporales[2].

Inicialmente, se transformó la columna de fechas y se agregaron las ventas diarias para conformar la serie temporal. La estacionariedad de la serie fue evaluada mediante la prueba Dickey-Fuller Aumentada (ADF), cuyo p-valor inferior a 0.05 confirmó que la seria era estacionaria y adecuada para el ajuste directo del modelo. La selección de los parámetros óptimos (p, d, q) se realizó con base en los criterios de información AIC y BIC. Todos los coeficientes resultaron estadísticamente significativos ($p < 0.001$), como en los términos AR (1) = +0.8095 y AR(2) = –0.7992, que indican oscilaciones significativas en la serie, acompañadas por otros componentes MA. Una vez ajustado el modelo con los datos históricos, se generaron predicciones para un horizonte temporal definido[8].

La varianza estimada del error ($\sigma^2 \approx 3877$) fue alta, y los diagnósticos de residuos revelaron problemas importantes: el test de Jarque-Bera reportó un valor extremo (JB ≈ 947600, $p \approx 0.00$), evidenciando que los errores no siguen una distribución normal. Además, se detectó heterocedasticidad (H de White ≈ 3.22, $p \approx 0.00$), lo que implica que la varianza de los residuos cambia en el tiempo.

Los resultados muestran que el desempeño del modelo fue moderadamente bueno, arrojando los siguientes resultados:

- MAE ≈ 30.8,
- MSE ≈ 3715.1,
- RMSE ≈ 60.95,
- MAPE ≈ 31.7%.

El MAPE obtenido lo ubica en la categoría "razonable" (25–50%) según la

clasificación en [9]. Si bien este valor no alcanza el nivel “bueno” (<25%), se alinea con reportes previos sobre modelos de ventas con MAPE del 17–20% [8]. Concluimos que el modelo captura la tendencia general de la serie, aunque no predice con alta precisión los extremos.

Para el modelo Prophet utilizamos Python 3, ocupamos las bibliotecas Pandas y NumPy para la manipulación y transformación de datos. La modelización se llevó a cabo con la biblioteca Prophet, diseñada para ajustar modelos de regresión aditiva que descomponen las series temporales en componentes de tendencia, estacionalidad y efectos de días festivos o eventos especiales[10]. Para la preparación de la serie temporal, las columnas fueron renombradas a ds (fecha) y y (ventas), conforme a los requisitos del modelo. Tras el ajuste del modelo con los datos históricos, se generó un conjunto de datos con fechas futuras para obtener predicciones a corto y mediano plazo. La capacidad del modelo para capturar patrones estacionales se evaluó mediante la visualización de sus componentes, incluyendo la tendencia general y la estacionalidad semanal y anual. Esto facilitó la interpretación de la dinámica de ventas y la cuantificación de la incertidumbre en las predicciones.

Sin embargo, el modelo mostró un desempeño muy deficiente. Como puede observarse en la figura, la línea azul de predicción no sigue adecuadamente los puntos reales, especialmente en los picos y valles. Prophet tiende a suavizar excesivamente la serie, lo que reduce su capacidad para capturar la alta variabilidad. Al igual que en el modelo ARIMA, se aplicaron las métricas; MAE, MSE, RMSE y MAPE para evaluar la precisión del modelo. Las métricas in-sample reflejan esta falta de precisión:

- MAE ≈ 39.98,
- MSE ≈ 5495.69,
- RMSE ≈ 74.13,
- MAPE ≈ 93.8%.

Este MAPE supera ampliamente el 50%, umbral como indicador de un modelo "poco confiable". Las posibles causas de este bajo rendimiento incluyen:

- Ausencia de regresores adicionales que expliquen mejor la variabilidad.
- Falta de ajuste de hiperparámetros.
- Comportamientos en la serie que no se explican bien por las estacionalidades estándar que Prophet incorpora.

Preprocesamiento de datos para los Modelos ARIMA y Prophet:

- Conversión de Fechas y Agrupación: La columna order_date, originalmente en formato Unix timestamp, fue convertida a datetime para facilitar su interpretación. Posteriormente, las ventas diarias fueron agregadas a nivel global y por categoría, generando series temporales estructuradas.
- Verificación de Estacionariedad: En el modelo ARIMA, la estacionariedad de la serie temporal fue evaluada mediante la prueba de Dickey-Fuller Aumentada (ADF), obteniendo un p-valor de 0.035. Este resultado indica que la serie es estacionaria y no requiere diferenciación adicional para su modelado [2].

4 Procedimiento experimental

El análisis de datos y la predicción de ventas se realizaron mediante dos enfoques principales: un modelo de redes neuronales LSTM y un enfoque estadístico utilizando los modelos ARIMA y Prophet. Pasos implementados para cada caso:

1. **Obtención de los datos:** El conjunto de datos utilizado corresponde al dataset público Olist. Se importaron al entorno de trabajo las tablas relevantes, incluyendo información sobre pedidos, ítems de pedidos, pagos, reseñas de clientes, productos, traducción de nombre de categoría y clientes.

2. **Integración y preparación de datos:** Para la construcción del conjunto de datos, se integraron múltiples tablas mediante operaciones de merge, consolidando la información en una única tabla unificada. Utilizando esta tabla, se realizaron diversas transformaciones para estructurar series temporales por categoría de producto:

- Se generó una columna de fecha diaria a partir del timestamp de compra, eliminando la información horaria.
- Los datos fueron agrupados por order_date y product_category_name_english, calculando el número total de ventas por día y categoría. Esto dio lugar a una serie temporal donde cada fila representa el total de ventas de una categoría en un día específico.
- La tabla agregada fue reestructurada para incluir una columna binaria por cada categoría única. En cada fila, solo la columna correspondiente a la categoría del registro se marcó con 1. Las demás se mantuvieron en 0.
- Se añadieron características estacionales mediante funciones seno y coseno para capturar ciclos anuales, mensuales y semanales. La columna de fecha original fue eliminada.
- Para la preparación de datos de entrada a la red neuronal, se generaron secuencias temporales de longitud fija. Se aplicó una ventana deslizante de 90 días sobre cada serie temporal normalizada, utilizando estos 90 días como entrada (X) y los 30 días siguientes como salida esperada (y).
- Finalmente, todas las secuencias de todas las categorías fueron recopiladas en dos arreglos denominados x_unificado e y_unificado, obteniendo aproximadamente 11,670 secuencias. Tras la división en conjuntos de entrenamiento y prueba (90/10), se dispuso de alrededor de 1,503 secuencias para entrenamiento y 1,167 para prueba.

3. **Modelo de Redes Neuronales:** Se diseño e implementó una red neuronal recurrente LSTM. La implementación se realizó con Keras, utilizando tf.keras.Sequential para definir el modelo, al cual se incorporaron dos capas: una LSTM con 32 unidades y una capa Dense con 30 neuronas. El entrenamiento se llevó a cabo con un tamaño de lote predeterminado y se aplicó la estrategia de distribución MirroredStrategy para optimizar el uso de la GPU. El modelo fue compilado con el optimizador Adam, la función de pérdida MSE y la métrica MAE para el monitoreo del desempeño.

El modelo LSTM fue entrenado utilizando la función model.fit, con un máximo de 20 épocas, procesando el conjunto de entrenamiento en cada iteración completa. Se

```
DATOS DE ENTRENAMIENTO
Valor de pérdida: 0.01188591867685318, Valor de métrica (MAE): 0.07283851504325867
DATOS DE PRUEBA
Valor de pérdida: 0.012535602785646915, Valor de métrica (MAE): 0.07460156828165054
37/37 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step
```

Fig. 4. Resultados de la evaluación del modelo con los datos de prueba y entrenamiento a través del método model.evaluate().

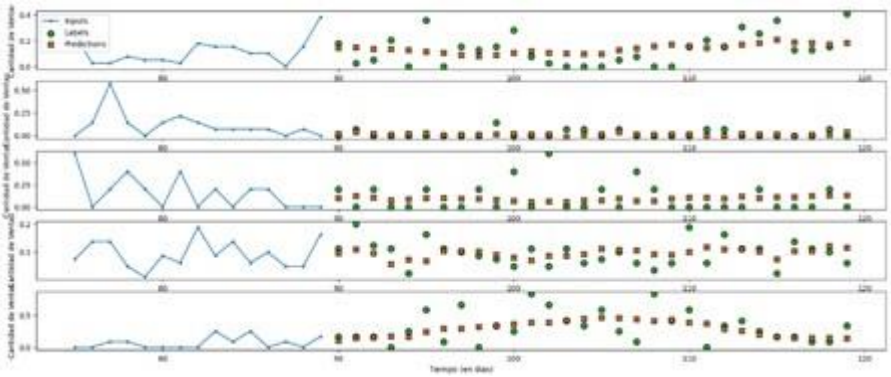

Fig. 5. Predicciones generadas con los datos de prueba. La línea azul punteada representa los últimos 20 datos de entrada, los puntos verdes representan las series temporales reales consecutivas, y las cruces naranjas son las predicciones generadas por el modelo.

empleo el subconjunto de prueba como validation_data para monitorear el desempeño en datos no utilizados en el ajuste. Para prevenir el sobreajuste, se implementó la estrategia de early stopping. Durante el proceso, los pesos del modelo fueron ajustados mediante backpropagation through time (BPTT) después de cada lote de secuencias. El entrenamiento se extendió hasta completar las 20 épocas, ya que la condición de parada anticipada no se activó, observándose una reducción progresiva de la pérdida de validación (val_loss), que alcanzó un valor aproximado de ~0.0125 al finalizar.

Una vez realizado el entrenamiento, se evaluó el desempeño del modelo utilizando tanto el conjunto de prueba como el de entrenamiento. Para ello, se suministraron al modelo las secuencias de entrada (x_pru) de 90 días, previamente no utilizadas en el ajuste, y se compararon las predicciones generadas para los 30 días siguientes (y_pred) con los valores reales correspondientes (y_pru). La evaluación cuantitativa se realizó mediante la función model.evaluate().

También, se generaron predicciones específicas utilizando la función model.predict sobre las secuencias de prueba, con el objetivo de visualizar y analizar los resultados. A partir de estas predicciones, se elaboraron gráficos comparativos entre las series de valores reales y los valores estimados para un subconjunto de categorías seleccionadas.

Para facilitar este análisis, se implementó la función auxiliar graficar_datos, la cual permite visualizar, para n casos de prueba, los últimos días de la ventana de entrada junto con la evaluación real futura y la estimada por el modelo. Este análisis visual complementa las métricas numéricas, proporcionando una mejor comprensión del desempeño del modelo en distintos escenarios.

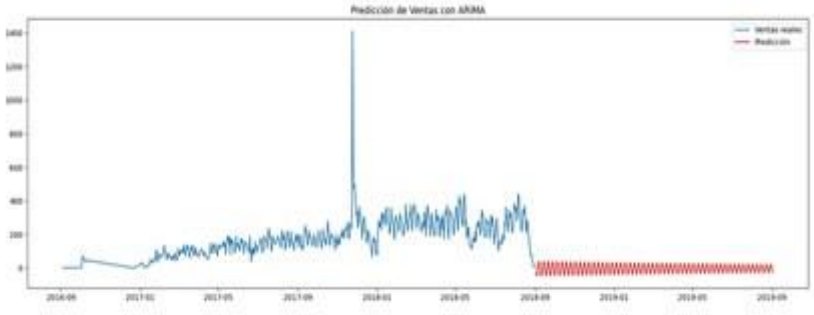

Fig. 6. Predicción de las ventas esperadas por Olist con el modelo estadístico ARIMA: La línea azul representa los valores reales de ventas, mientras que la línea roja muestra las predicciones generadas para un horizonte extendido.

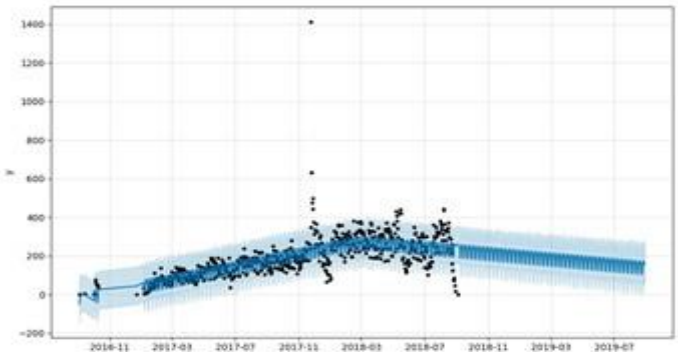

Fig. 7. Predicción de ventas diarias utilizando el modelo Prophet. Los puntos negros representan las observaciones reales, mientras que la línea azul indica la tendencia estimada y las bandas sombreadas corresponden a los intervalos de confianza.

4. **Modelos Estadísticos:** De forma paralela, se abordó la predicción de ventas utilizando dos métodos estadísticos:

ARIMA:

- Se transformó la columna order_date a formato datetime y se agrupó la serie de ventas diarias.
- Se aplicó la prueba de Dickey-Fuller Aumentada (ADF), la cual arrojó un p-valor inferior a 0.05, indicando estacionariedad y permitiendo el ajuste directo del modelo.
- Con criterios de información (AIC y BIC), se seleccionaron parámetros óptimos y se ajustó el modelo con la biblioteca statsmodels.
- Se generaron predicciones para un horizonte definido por 365 días y se evaluó el desempeño utilizando métricas: MAE, MSE, RMSE y MAPE, complementando el análisis mediante gráficos comparativos de las series reales y las predichas.

Prophet:

- Se preparó la serie temporal renombrando las columnas a 'ds' (fecha) y 'y' (ventas_totales) según los requerimientos del modelo.
- El modelo Prophet se ajustó a la serie histórica y se generó un conjunto de datos de fechas futuras para obtener predicciones a corto y mediano plazo (hasta 360 días).
- Se analizaron los componentes del modelo, lo que permitió interpretar los patrones subyacentes en las ventas y cuantificar la incertidumbre en las predicciones.
- Se calcularon métricas de error MAE, MSE, RMSE y MAPE para evaluar la precisión del modelo.

5 Resultados y discusión

Una vez realizado el entrenamiento del modelo de redes neuronales, se obtuvo en el conjunto de prueba, en la escala normalizada, un MAE de aproximadamente 0.0746, un MSE de aproximadamente 0.0125, un RMSE de 0.1125 y un SMAPE de 67.4727%. Estos resultados sugieren que, a pesar de la alta variabilidad en la serie temporal, el modelo logró capturar la dinámica de las ventas de manera efectiva. Además, la similitud entre los errores obtenidos en los conjuntos de entrenamiento y prueba indica una adecuada capacidad de generalización, sin evidencias de sobreajuste.

La figura 5 presenta ejemplos de predicción de ventas diarias para cinco casos del conjunto de prueba. En estos gráficos, la línea azul punteada representa los últimos 20 días de la ventana de entrada, los puntos verdes corresponden a las ventas reales de los 30 días posteriores y las cruces naranjas indican las predicciones generadas por el modelo. Los resultados muestran que la red neuronal logra capturar de manera efectiva la dirección de las tendencias y los patrones estacionales de las series temporales. No obstante, se identifican leves desviaciones en algunos casos, como la subestimación de picos de demanda o un ligero adelantamiento en la detección de caídas abruptas en las ventas.

En contraste, los métodos estadísticos presentaron mayores dificultades para modelar la compleja dinámica de la serie temporal. En particular, el modelo ARIMA, mostró un desempeño inferior en comparación con la red neuronal. Se obtuvieron errores estadísticamente significativos más altos, con un MAE de 30.792, un MSE DE 3714.466, un RMSE de 60.946 y un MAPE de 31.71%, de igual forma la varianza estimada del error ($\sigma^2 \approx 3877$) fue alta, y los diagnósticos de residuos revelaron problemas importantes: la prueba de Jarque-Bera reportó un valor extremo (JB ≈ 947600, $p \approx 0.00$), evidenciando que los errores no siguen una distribución normal. Además, se detectó heterocedasticidad (H de White ≈ 3.22, $p \approx 0.00$), lo que implica que la varianza de los residuos cambia en el tiempo.

La figura 6 muestra la predicción de ventas realizada con ARIMA, donde la línea roja, que representa las predicciones, presenta desviaciones notables respecto a la línea azul, que indica los valores reales. Estas diferencias reflejan las limitaciones del modelo para capturar cambios abruptos y adaptarse a la alta volatibilidad de la serie temporal.

El modelo Prophet también mostró un desempeño inferior, registrando un MAE de 39.984, un MSE de 5495.686, un RMSE de 74.133 y un MAPE de 93.80%. Como se observa en la figura 7, aunque el modelo capta la tendencia general (línea azul) y

muestra los intervalos de confianza, las predicciones presentan errores notables en la estimación de picos y valles, lo que indica dificultades para ajustarse a la alta variabilidad de las ventas diarias.

Comparando ambos enfoques, se observa que el modelo ARIMA entregó un desempeño significativamente mejor que Prophet. ARIMA alcanzó un MAPE ≈ 31.7%, lo cual, aunque no ideal, se considera razonablemente confiable. Por el contrario, Prophet obtuvo un MAPE cercano al 94%, lo que lo clasifica como un modelo altamente impreciso para este conjunto de datos.

Estos resultados reflejan que ARIMA logró capturar la tendencia general y comportamientos autorregresivos importantes de la serie, mientras que Prophet, con su configuración por defecto, fracasó en representar adecuadamente la variabilidad. De acuerdo con lo reportado por Krishna en [10] los modelos bien ajustados deberían presentar MAPEs inferiores al 20%, lo cual se aproxima ARIMA pero no Prophet.

En contraste, el enfoque basado en redes neuronales recurrentes, al entrenarse conjuntamente en todas la categorías, permite aprovechar patrones comunes y modelar relaciones no lineales, lo que se traduce en una mayor precisión en la predicción.

Sin embargo, se ha identificado oportunidades de mejora en el modelo de redes neuronales. La incorporación de variables externas (como promociones, feriados o tendencias de búsqueda) y la exploración de arquitecturas más profundas (por ejemplo, modelos secuencia-a-secuencia o con mecanismos de atención) podrían aumentar la capacidad predictiva y mejorar la robustez frente a cambios en la demanda.

Podemos mencionar que los hallazgos indican que la red neuronal presenta un desempeño sobresaliente en la predicción de ventas diarias, superando a los modelo ARIMA y Prophet en términos de precisión y adaptación a la alta variabilidad de la serie. Aunque los métodos estadísticos permiten una interpretación clara de los componentes de la serie temporal, su capacidad predictiva es limitada en escenarios con fluctuaciones abruptas. Estos hallazgos abren la posibilidad de futuras investigaciones que integren enfoques híbridos o incorporen información adicional para optimizar la toma de decisiones en el comercio electrónico. Los modelo estadísticos y la red neuronal presentada fueron publicados en el siguiente repositorio de GitHub para el estudio público: https://github.com/CesarLara08/COMIA_PAPER_155_2025.

6 Conclusiones

En este estudio se desarrolló un modelo predictivo de ventas basado en una red neuronal recurrente con arquitectura LSTM y se comparó con métodos estadísticos tradicionales (ARIMA y Prophet) usando el conjunto de datos Olist. La red neuronal mostró alta precisión en la predicción de ventas diarias, con errores mínimos en datos no vistos, lo que evidencia su capacidad para capturar la compleja dinámica y estacionalidad de las distintas categorías de productos. En contraste, aunque los modelos ARIMA y Prophet permitieron descomponer la seria en componentes de tendencia y estacionalidad, sus errores fueron significativamente mayores, subrayando las limitaciones de los enfoques lineales y aditivos ante la volatilidad y los picos irregulares del comercio electrónico.

La efectividad de la red neuronal sugiere que un enfoque basado en aprendizaje profundo puede apoyar la toma de decisiones empresariales en el sector minorista,

facilitando la planificación de inventarios, la gestión de la cadena de suministro y el diseño de estrategias de marketing proactivas. Además, entrenar un modelo global que abarque todas las categorías simplifica el mantenimiento y la escalabilidad, en comparación con la necesidad de ajustar modelos individuales con métodos estadísticos.

Como líneas de trabajo futuro, se propone incorporar variables exógenas para enriquecer el modelo y explotar arquitecturas de redes más avanzadas para capturar dependencias a largo plazo. También se sugiere comparar enfoques híbridos que combinen las fortalezas de los métodos estadísticos y el aprendizaje profundo, permitiendo un análisis más robusto y adaptable a las condiciones del mercado.

En conclusión, el estudio demuestra la viabilidad y efectividad del enfoque basado en LSTM para la predicción de ventas en comercio electrónico, superando las restricciones de los métodos estadísticos tradicionales y abriendo nuevas perspectivas para optimizar estrategias comerciales en la era del Big data.

Referencias

1. Olist, A. S.: Brazilian E-Commerce Public Dataset by Olist. https://www.kaggle.com/dsv/195341 (2024)
2. Ayala-Aldana, N., Monleon-Getino, A., Canela-Soler, J., Retamal-Contreras, E., Predicción con modelo ARIMA en series temporales de Salmonella spp en Chile entre 2014-2022. Ciencia Latina Revista Científica Multidisciplinar, 7(1), pp. 1337–1351 (2023) doi: 10.37811/cl_rcm.v7i1.4484.
3. Deng, M.: The Analysis of Hidden Units in LSTM Model for Accurate Stock Price Prediction. INSTICC, pp. 419–424 (2024) doi: 10.5220/0012799100003885.
4. Bajoudah, A., Alsaidi, M., Alhindi, A.: Time Series Forecasting Model for E-commerce Store Sales Using FB-Prophet. In: 14th International Conference on Information and Communication Systems (ICICS), pp. 1–6 (2023) doi: 10.1109/ICICS60529.2023.10330530.
5. Weytjens, H., Lohmann, E., Kleinsteuber, M.: Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. Electronic Commerce Research, 21(2), pp. 371–391 (2021). doi: 10.1007/s10660-019-09362-7.
6. Yu, Y., Si, X., Hu, C., Zhang, J.: A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Comput, 31(7), pp. 1235–1270 (2019) doi: 10.1162/neco_a_01199.
7. Zhang, X., Guo, F., Chen, T., Pan, L., Beliakov, G., Wu, J.: A Brief Survey of Machine Learning and Deep Learning Techniques for E-Commerce Research. Multidisciplinary Digital Publishing Institute (MDPI) (2023) doi: 10.3390/jtaer18040110.
8. Hendra Saputra, B.: Evaluation of ARIMA model performance in projecting future sales: Case study on electronic products. Jurnal Mantik, 16(5) (2024. doi: 10.35335/cit.Vol16.2024.993.pp329-337.
9. Khatibi, A., Da Silva, A.P.C., Almeida, J.M., Gonçalves, M.A.: A quantitative analysis of the impact of explicit incorporation of recency, seasonality and model specialization into fine-grained tourism demand prediction models. PLoS One, 17(12) (2022) doi: 10.1371/journal.pone.0278112.

10. Krishna, K., Samal, R., Sathya, K., Santosh, B., Das, K., Acharaya, A.: Time Series based Air Pollution Forecasting using SARIMA and Prophet Model (2019) doi: 10.1145/3355402.3355417.

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación