EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 155 No. 7
July 2026

Research in Computing Science

Research in Computing Science, Año 25, Volumen 155, No. 7, julio de 2026, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2026-043011360400-102. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de julio de 2026.

Research in Computing Science, year 25, Volume 155, No. 7, July 2026, is published monthly by the Center for Computing Research of IPN.

Volume 155(7)

Advances in Artificial Intelligence

Roberto A. Vázquez (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2026

ISSN: in process

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

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

Table of Contents

Page

Gamificación aplicada en matemáticas para discentes de nivel básico empleando avatar emocional reactivo (por intentos)

David Gustavo Lara Velázquez, Luis Ángel Reyes Hernández, Humberto Marín Vega, Ignacio López Martínez, Giner Alor Hernández

Tecnológico Nacional de México/I.T. de Orizaba, Orizaba, División de Estudios de Posgrado e Investigación, México

{M19011379, luis.rh, humberto.mv, ignacio.lm,giner.ah}@orizaba.tecnm.mx

Resumen. En el contexto de la educación básica mexicana, existe una crisis en el aprendizaje de las matemáticas. Debido al modelo de enseñanza tradicional en el que el principal objetivo es la memorización a través de la repetición. La prueba PISA del 2022 posicionó a México como el tercer país de la OCDE con el peor rendimiento en matemáticas. Por tal motivo, es de vital importancia trabajar el enfoque de la enseñanza de matemáticas con énfasis en el desarrollo de habilidades blandas como la resiliencia y perseverancia. Con la finalidad de reconfigurar la perspectiva del error para disminuir el estrés y ansiedad matemática en discentes de nivel primaria. De tal manera que el objetivo del presente trabajo es presentar la propuesta de una herramienta pedagógica en un entorno web diseñada para discentes de educación básica que integra un avatar emocional reactivo capaz de brindar la situación emocional actual del discente según su desempeño. Además, como parte de la gamificación implementa un sistema de recompensas como incentivo clave para fomentar la motivación intrínseca y el compromiso con el aprendizaje.

Palabras clave: Gamificación, matemáticas, videojuego, avatar reactivo, educación básica.

Gamification Applied to Mathematics for Basic Education Learners Using an Emotionally Reactive Avatar (based on Attempts)

Abstract. In the context of Mexican basic education, there is a crisis in mathematics learning. This is largely due to a traditional teaching model where the primary objective is rote memorization through repetition. The 2022 PISA test ranked Mexico as the OECD country with the third-lowest performance in mathematics. Therefore, it is vital to restructure the approach to math education, emphasizing the development of soft skills such as resilience and perseverance. The aim is to reframe the perspective on making mistakes to reduce math-related

stress and anxiety among elementary school students. Consequently, this paper presents a proposal for a web-based pedagogical tool designed for basic education students. This platform integrates a reactive emotional avatar capable of providing the current emotional status based on the student's performance. Furthermore, as part of the gamification strategy. It implements a reward system as a key incentive to foster intrinsic motivation and engagement with the learning process.

Keywords: Gamification, mathematics, videogame, reactive avatar, elementary education.

1. Introducción

En el contexto de la educación básica mexicana, existe una crisis en el aprendizaje de las matemáticas. Debido al modelo de enseñanza tradicional en el que el principal objetivo es que los discentes memoricen a través de la repetición los diferentes temas abordados de acuerdo con el nivel educativo. La prueba PISA DEL 2022 dejó muy en claro que este es el principal factor por el cual México se encuentra como uno de los países con el peor rendimiento en el área de las matemáticas [1]. Una de las consecuencias más perjudiciales de este enfoque tradicional es la penalización sistemática del error. Esta práctica fomenta altos niveles de estrés y "ansiedad matemática" en los discentes. Lo que perjudica su persistencia y constituye una percepción negative hacia la disciplina.

Para hacer frente a este desafío, a continuación, se presenta la propuesta de una aplicación web gamificada para matemáticas para discentes de nivel básico. Por medio de la cual se propone fomentar la resiliencia para minimizar el estrés matemático, de tal manera que se reconfigure la perspectiva del error que el modelo de enseñanza tradicional fomentó. Debido a que es un entorno web en el que se desarrollará la aplicación, las tecnologías utilizadas son Next.js, Nest.js y Phaser, con el objetivo de mantener un código estructurado y la armonía de lenguajes de programación, en este caso Typescript.

2. Estado del arte

Para sustentar el desarrollo de la aplicación de gamificación se realizó una revisión de la literatura relacionada, mediante la cual se exploran las diferentes técnicas de gamificación con mayor relevancia para una correcta implementación en cada contexto académico.

Respecto a las herramientas web existentes, en Ojeda et al. [2] se analizaron las técnicas de gamificación con el objetivo de mejorar la motivación y compromiso del discente mediante herramientas web ya existentes como Kahoot!, Memorise, Classcraft y Makebadges. Para analizar las tendencias de investigación en la gamificación matemática, Elmawati et al. [3] proporcionaron un mapeo sistemático de diferentes bases de datos académicas de tal manera que lograron identificar vacíos en estudios interdisciplinarios, sostenibilidad a largo plazo e investigación longitudinal. En cuanto

a la efectividad diferencial de cada tipo de juego educativo a través del modelo GEAM, en Heintz et al. [4] los hallazgos fueron que los rompecabezas benefician el aprendizaje matemático mientras que los juegos de simulación demostraron mayor efectividad para ciencias, los híbridos demostraron versatilidad interdisciplinaria y efectividad moderada comparada con los tipos especializados, de tal manera que la conclusión fue la necesidad de personalización en la selección de juegos y desarrollo de algoritmos.

Desde la perspectiva docente, en Hodedatov et al. [5] identificaron las buenas prácticas y las dificultades del docente al integrar juegos en educación primaria mediante entrevistas, observaciones de aula, análisis de documentos institucionales y grupos focales con discentes, como resultado obtuvieron que las barreras frecuentes es tiempo limitado, formación insuficiente, dificultades técnicas y preocupaciones sobre el tiempo en pantalla. La exploración a profundidad de la percepción, experiencia y creencias de los docentes sobre la gamificación en Koutroumani et al. [6] remarcó la importancia del desarrollo profesional en habilidades técnicas, enfoques de implementación gradual e investigaciones que demuestren la conexión exacta entre la gamificación y las medidas tradicionales de éxito académico.

En el ámbito del desarrollo aplicado, en Li et al. [7] desarrollaron un videojuego de Realidad Aumentada colaborativo para matemáticas elementales, mediante el cual los discentes no solo mejoraron en el área de las matemáticas, sino también las habilidades blandas como resiliencia, interacción social, ayuda mutua y comunicación matemática.

La inteligencia artificial también juega un rol importante, como en Koravuna et al. [8] que desarrollaron la plataforma NOVA Labs, mediante la cual implementaron inteligencia artificial con la finalidad de mejorar la alfabetización digital en discentes universitarios, las herramientas que utilizaron principalmente fueron Python, HTML5, TensorFlow y JavaScript. Para incrementar la motivación museística en discentes de nivel primaria, en Ioannou et al. [9] desarrollaron aplicaciones gamificadas para dispositivos móviles con códigos QR mediante el cual identificaron factores críticos en la gamificación como narrativa coherente, progresión clara en la dificultad y oportunidades regulares para interacción social.

De manera similar, en Stappen et al. [10] desarrollaron MathBuilder, en donde implementaron realidad aumentada para construir soluciones en espacios compartidos, mediante el cual los discentes mejoraron la comprensión conceptual, motivación, comunicación y eficacia en geometría y fracciones.

En el contexto de enseñanza STEM, en Ranosz et al. [11] Integraron sistemas de juegos 3D inmersivos, en los cuales incorporaron sistemas de progresión interdisciplinarios, portafolios digitales de proyectos, competencias inter-escolares y mentorías virtuales con profesionales STEM, como resultado demostraron mejoras significativas en la motivación hacia carreras científicas, colaboración y pensamiento computacional.

Para fortalecer el pensamiento lógico de los discentes, en Liang et al. [12] desarrolló una plataforma web gamificada, en donde implementaron técnicas de gamificación como sistemas de logros por tipos de razonamiento, desafíos de tiempo adaptativos, pistas contextuales y celebraciones de soluciones conceptuales.

Otro elemento esencial es la cultura, en Alaa et al. [13] exploró cómo la relevancia cultural en juegos educativos impacta en los resultados de aprendizaje, motivación y la

atención del discente, mediante el cual identificaron que elementos culturalmente familiares facilitaron conexiones entre conocimientos previos y nuevos.

Las narrativas en Kaymakci et al. [14] potenciaron la retención y motivación de los discentes, de tal manera que los discentes demostraron patrones de exploración más profundos, mayor tendencia a re-visitar contenidos y desarrollo de conexiones inter-conceptuales más sofisticadas.

Para aumentar el acceso a actividades de programación computacional en educación básica Hunter et al. [15] desarrollaron y evaluaron un currículo gamificado que integró narrativas, arte digital, creación de música y activismo social para dar relevancia a la programación, concluyeron que es esencial implementar sistemas de apoyo sostenido y materiales curriculares culturalmente sensibles.

Para mejorar la motivación y competencia en inglés, en Xue et al. [16] implementaron elementos de gamificación mediante los cuales obtuvieron una tasa de adquisición de vocabulario con mejora del 67%, la precisión de la pronunciación mejoró un 43%, además de un aumento motivacional significativo para el aprendizaje del inglés.

En general, se concluye que cada trabajo relacionado implementó y analizó una técnica diferente para enriquecer la gamificación. Los trabajos relacionados se distinguen de este proyecto en que no tomaron en cuenta todos los factores importantes que no están relacionados con la gamificación, como el aspecto cultural y social. En consecuencia, la presente propuesta hará uso de las técnicas para el enriquecimiento de las actividades gamificadas más relevantes y con mayor efectividad para incrementar el interés en los discentes. De tal manera que no solo se enfoque en el aprendizaje matemático, sino también en impulsar el desarrollo de las habilidades blandas de los discentes.

3. Metodología

En esta sección se presenta la metodología empleada para delimitar el grado óptimo para la aplicación de este proyecto, los elementos de gamificación correctos de acuerdo con docentes experimentados y el temario adecuado.

3.1. Arquitectura de la aplicación web

La propuesta consiste en desarrollar un videojuego de matemáticas con universos y misiones lúdicas con un avatar emocional el cual mediante inteligencia artificial proporcione retroalimentación inmediata al discente sobre su avance. De igual manera el docente tendrá acceso a las estadísticas de las partidas de cada discente con la finalidad de analizar los resultados de cada discente. El diagrama de la arquitectura se encuentra en la Fig. 1.

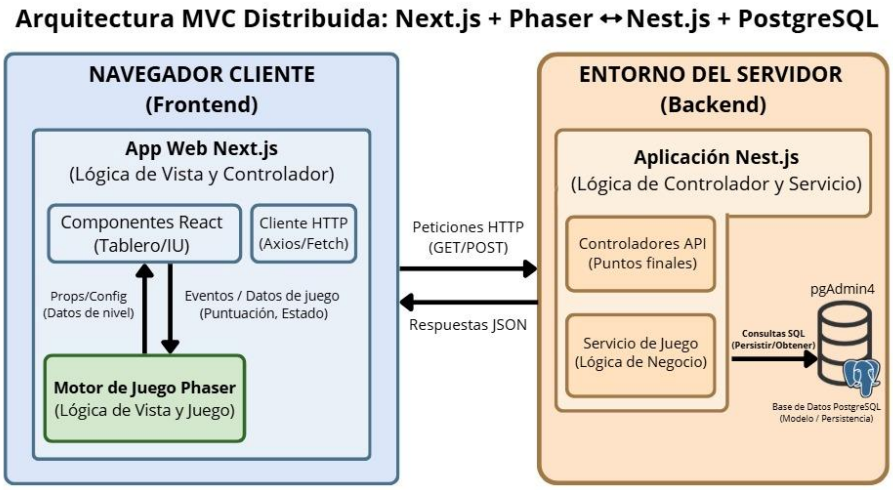

Fig. 1. Arquitectura MVC para el desarrollo de la aplicación web que contiene el videojuego.

La arquitectura del proyecto es MVC (Modelo, Vista, Controlador) como se puede observar en la ilustración. Debido al uso de microservicios que conectan y procesan la información para guardarla en la base de datos, así como mostrar la información proveniente de la base de datos en el navegador del usuario. En donde el navegador del cliente es la vista. Incrustado en una etiqueta canva, se encuentra también el videojuego desarrollado con phaser. Los usuarios interactúan con la vista, de tal manera que todos los tableros que muestran información, formularios que reciben información y el juego que también recibe y genera información emitirá una petición HTTP o recibirán una respuesta JSON.

La lógica del negocio está repartida en el videojuego en phaser, el frontend en Next.js y el backend en Nest.js. En el juego se encuentran los algoritmos para el avatar emocional; como el de acumulación por umbrales [17] y el de máquina de estados finitos [18], algoritmos de puntaje, entre otros. Debido a que el frontend también se comunicará con el videojuego para definir la dificultad del nivel también esto cumple con los requisitos para tomarse en cuenta como lógica del negocio. En el backend también se encuentran los algoritmos para tratar la información.

El controlador se encuentra en el backend, el cual se encarga de recibir información de la vista y enviarla a la base de datos y viceversa. Recibe una petición HTTP de parte de la vista y da una respuesta JSON con la información que la vista necesite de la base de datos.

3.2. Análisis de requisitos

Encuesta. Se llevó a cabo una encuesta[1] dirigida a normalistas cuya formación se especializa en educación básica, ellos actualmente ya cuentan con experiencia vasta

[1] https://forms.gle/S7Zfqr5kGjh7wtu16

Fig. 2. Personaje ganador.

frente a grupos de primaria y logran identificar las carencias y las virtudes de cada grado en específico. A continuación, se presenta el resumen de la encuesta aplicada.

La encuesta arrojó que el grado con mayor problema en matemáticas es el cuarto grado, debido a que en este nivel se introducen los temas de suma, resta, multiplicación y división y en los siguientes grados es un repaso de lo aprendido en cuarto grado. Por consiguiente, si se tienen buenas bases, en los niveles venideros los discentes tienen menos problemas y sufren menos ansiedad matemática.

La propuesta de un videojuego incrustado en una aplicación web fue aceptada ya que el entorno de uso es un ambiente controlado, en donde está prohibido para los discentes de estas edades portar con un teléfono inteligente. De tal manera que una aplicación web es óptima para ingresar desde cualquier dispositivo que el docente tenga acceso.

La encuesta indicó que la mejor dinámica de juego es la de plataformas y destreza, junto con un progreso visual del avatar para fomentar el hábito de estudio. Además de un enfoque mixto del tiempo, en donde la práctica no tenga tiempo límite pero que existan pruebas con tiempo límite. Un Sistema de vidas y reinicio de nivel es un punto clave para fomentar la atención del discente, ya que de lo contrario el discente podría perder el interés en la práctica. Un refuerzo auditivo de las instrucciones es esencial ya que no todos los discentes tienen la misma manera de aprendizaje. Las recompensas emocionales por parte del avatar junto con el rango de emociones “Muy triste”, “Triste”, “Normal”, “Feliz”, “Muy Feliz” son adecuados para estas edades ya que ayudaría al discente a entender más sus emociones.

Además, en una versión posterior se implementarán recompensas como estrellas almacenables para cada discente, otorgadas al finalizar un intento con cantidad variable dependiendo de la eficaz del discente durante el juego y fanfarrias al finalizar un intento correcto, todo esto crea un entorno adecuado para incrementar la motivación del discente.

Se presentaron seis propuestas de personajes y el personaje para el avatar más votado de acuerdo con lo popular entre estas edades fue el ajolote (ver Fig. 2) y la temática seleccionada fue el planeta tierra (ver Fig. 3), con hincapié en la inmersión del discente en el videojuego, ya que este es el ambiente en el que los discentes de estas edades conocen más.

Análisis de los temarios de primaria. Se realizó el análisis de los temarios de la Nueva Escuela Mexicana de cuarto, quinto y sexto grado de primaria, a continuación, se

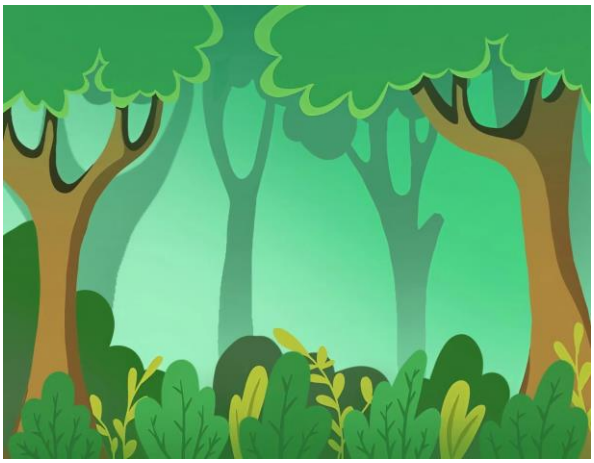

Fig. 3. Temática ganadora.

muestran los resultados del análisis del temario de cuarto grado [19] ya que este es el recomendado por los normalistas encuestados.

1. Estudio de los números:
   - Lectura, escritura y orden de números naturales (5 cifras máximo)
   - Identificación de sucesiones numéricas ascendentes y descendentes
   - Uso de fracciones para expresar medidas y repartos (medios, cuartos, octavos y dieciseisavos)
   - Fracciones equivalentes y comparación de fracciones con igual numerador o denominador
2. Suma y resta:
   - Algoritmos convencionales de suma y resta con números naturales de hasta cinco cifras
   - Calculo mental de sumas y restas con múltiplos del 100 y 1000
   - Suma y resta de fracciones con el mismo denominador
3. Multiplicación y división:
   - Multiplicación de números naturales (3 cifras X 2 cifras)
   - División de números naturales (cociente de hasta 3 cifras y residuo)
   - Uso del algoritmo de la división
4. Cuerpos geométricos y figuras:
   - Simetría: identificación de ejes de simetría en figuras y objetos
   - Características de cuerpos geométricos (número de caras, aristas y vértices)
   - Clasificación de cuadriláteros
5. Medición:
   - Cálculo de perímetros y áreas (mediante cuadrículas y formulas básicas para rectángulos)
   - Noción del tiempo (lectura del reloj y calendario)
6. Organización de datos:
   - Lectura e interpretación de tablas de doble entrada y barras

Fig. 4. Prototipo de nivel. Caso de desenlace positivo.

Fig. 5. Prototipo de nivel. Caso de desenlace positivo.

– Moda con un conjunto de datos.

Prototipo de nivel. A continuación, se muestra el primer prototipo funcional de un nivel para práctica de multiplicación de uno por dos dígitos.

Como se observa en la primera captura de pantalla (ver Fig. 4) el avatar emocional reactivo se encuentra en el centro de la pantalla en la parte inferior con un estado "Normal". Si el discente recolecta los números correctos que multiplicados dan el resultado solicitado, el avatar emocional reactivo cambia a estado "Feliz" (ver Fig. 5). Finalmente, cuando el discente recolecta los números correctos y se aproxima a la puerta de salida el avatar emocional reactivo cambia de estado a "Muy Feliz", se le recompensa al discente con una estrella y la puerta se abre (ver Fig. 6).

Fig. 6. Prototipo de nivel. Caso de desenlace positivo.

Fig. 7. Prototipo de nivel. Caso de desenlace negativo.

Para el caso de prueba con un desenlace negativo, se observa en la primera captura de pantalla que el discente recolecta un número correcto y el avatar emocional reactivo cambia a estado “Feliz” (ver Fig. 7). Sin embargo, cuando el discente recolecta un número equivocado, el avatar emocional reactivo cambia de estado a “Triste” (ver Fig. 8). Finalmente, cuando el discente se aproxima a la puerta de salida se muestra un mensaje para que lo vuelva a intentar y la puerta no se abre (ver Fig. 9).

4. Conclusiones

Se remarcó la importancia de la gamificación de matemáticas con un avatar emocional reactivo en un entorno web dirigido a educación básica, centrado en

Fig. 8. Prototipo de nivel. Caso de desenlace negativo.

Fig. 9. Prototipo de nivel. Caso de desenlace negativo.

discentes de cuarto grado con temas seleccionados, los cuales son: suma, resta, multiplicación y división. Además, la literatura revisada junto con las sesiones con los normalistas sirvió de gran ayuda para aclarar puntos clave como delimitar el alcance del proyecto, enfoque y mecánicas principales para lograr la correcta implementación de la gamificación, esta revisión sistemática justificó y validó las decisiones tomadas.

Así mismo, la implementación de mecánicas de gamificación como recompensas como incentivo demostró ser eficaz para mitigar la ansiedad matemática reportada, de tal manera que fomenta la motivación intrínseca al celebrar la perseverancia sobre la velocidad de respuesta, lo que se convierte en resiliencia.

Finalmente, la implementación en un entorno web garantizó la accesibilidad del recurso, factor crucial para brindar un acceso en una amplia variedad de dispositivos, de tal manera que se eliminen las limitaciones de hardware y los discentes tengan

acceso a la práctica sin barreras. La propuesta tiene el potencial de reconfigurar la percepción del error tradicional en las nuevas generaciones, por medio de la implementación de una experiencia de aprendizaje segura y motivadora.

Como trabajo a futuro se contempla el desarrollo de las actividades y misiones lúdicas, además de una fase de pruebas en una escuela primaria para validar la utilidad del contenido educativo, jugabilidad y usabilidad del software.

Agradecimientos. Se expresa un agradecimiento al Tecnológico Nacional de México por su respaldo, destacando al Instituto Tecnológico de Orizaba como el anfitrión en el desarrollo de este proyecto.

Referencias

1. PISA 2022: Dos de cada tres estudiantes en México no alcanzan el nivel básico de aprendizajes en Matemáticas. https://imco.org.mx/pisa-2022-dos-de-cada-tres-estudiantes-en-mexico-no-alcanzan-el-nivel-basico-de-aprendizajes-en-matematicas/, last accessed 2025/10/12 (2025)
2. Ojeda-Lara, O.G., Zaldívar-Acosta, M. del S.: Gamificación como Metodología Innovadora para Estudiantes de Educación Superior. Revista Docentes 2.0., Vol. 16, pp. 5–11 (2023). Doi:10.37843/rted.v16i1.332.
3. Elmawati, E., Martadiputra, B.A.P., Samosir, C.M.: Gamification Research Focus in Learning Mathematics. ACM International Conference Proceeding Series, Vol. 142–149 (2023). Doi:10.1145/3631991.3632012.
4. Heintz, S., Law, E.L.C.: Digital Educational Games. ACM Transactions on Computer-Human Interaction (TOCHI), Vol. 25, (2018). Doi:10.1145/3177881.
5. Hodedatov, S., Avidov-Ungar, O., Hayak, M.: The integration of digital games in elementary schools: The principals' point of view. Educ. Inf. Technol. (Dordr), Vol. 29, pp. 18003–18021 (2024). Doi:10.1007/S10639-024-12568-4.
6. Koutroumani, M., Balaskas, S., Rigou, M.: Learning through Gamification as perceived by Educators: A Survey-based Analysis of Affecting and Mediating Factors using SEM. Proceedings - 28th Pan-Hellenic Conference on Progress in Computing and Informatics with International Participation, PCI, pp. 307–314 (2025). Doi:10.1145/3716554.3716601.
7. Li, J., Van Der Spek, E.D., Yu, X., Hu, J., Feijs, L.: Exploring an augmented reality social learning game for elementary school students. Proceedings of the Interaction Design and Children Conference, IDC, pp. 508–518 (2020). Doi:10.1145/3392063.3394422.
8. Koravuna, S., Surepally, U.K.: Educational gamification and artificial intelligence for promoting digital literacy. ACM International Conference Proceeding Series (2020). Doi:10.1145/3415088.3415107.
9. Ioannou, I., Kyza, E.A.: The role of gamification in activating primary school students' intrinsic and extrinsic motivation at a museum. ACM International Conference Proceeding Series (2017). Doi:10.1145/3136907.3136925.
10. Van Der Stappen, A., Liu, Y., Xu, J., Yu, X., Li, J., Van Der Spek, E.D.: MathBuilder: A collaborative AR math game for elementary school students. CHI PLAY - Extended

Abstracts of the Annual Symposium on Computer-Human Interaction in Play. 731–738 (2019). Doi:10.1145/3341215.3356295.
11. Ranosz, J., Leszczyński, C., Kumor, S., Popiel, A., Głowaczewska, J., Garwol, P., Kaczmarek, M., Maik, M.: Advancing STEM Education in Primary Schools with an Integrated System of 3D Games. Proceedings - Web3D 2023: 28th International Conference on Web3D Technology (2023). Doi:10.1145/3611314.3615919.
12. Liang, H., Sitthiworachart, J.: Game-based learning to enhance students logical thinking abilities in primary school Mathematics. ACM International Conference Proceeding Series, pp. 86–93 (2023). Doi:10.1145/3629296.3629336.
13. Alaa, M., Hammouda, N., Abdennadher, S.: The effect of culture in educational games for school students. ACM International Conference Proceeding Series (2023). Doi:10.1145/3623462.3630609.
14. Kaymakci Ustuner, K., Law, E.L.C., Li, F.W.B.: Digital Educational Games with Storytelling for Students to Learn Algebra. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 14145 LNCS, pp. 459–463 (2023). Doi:10.1007/978-3-031-42293-5_54.
15. Hunter, H., Payton, J., Julien, C.: Expanding Elementary School Computer Science Education with an Introduction to Machine Learning Through Rhythmic Studies. Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 463–467 (2023). Doi:10.1145/3565287.3617623.
16. Xue, S., Han, Z.: Using Game Elements to Enhance Chinese Middle School Students' English Learning Motivation in the Classroom, pp. 100–104 (2024). Doi:10.1145/3729434.3729442.
17. Belokurov, V., Koshelev, V., Kagalenko, M.: The use of Characteristic Functions in the Multi-Frame Accumulation Algorithm. 2019 8th Mediterranean Conference on Embedded Computing, MECO 2019 - Proceedings (2019). Doi:10.1109/MECO.2019.8760010.
18. Zeng, Z., Goodman, R.M., Smyth, P.: Learning Finite State Machines With Self-Clustering Recurrent Networks. Neural Comput., Vol. 5, pp. 976–990 (1993). Doi:10.1162/neco.1993.5.6.976.
19. Secretaría de Educación Pública: Programa de Estudio para la Educación Primaria: Programa Sintético de la Fase 4 (2024)

Metodología para entrenamiento de redes neuronales artificiales utilizadas como controladores inteligentes en procesos industriales

Paul Enrique Arias-Lizárraga, Julio César Picos-Ponce,
Guillermo Javier Rubio-Astorga, David Enrique Castro-Palazuelos

Tecnológico Nacional de México/Instituto Tecnológico de Culiacán,
México

{M24171313, julio.pp}@culiacan.tecnm.mx

Resumen. El uso de aplicaciones de inteligencia artificial (IA) en la industria 4.0 son imprescindibles en la actualidad. Sin embargo, el hardware estándar con el que se implementan las aplicaciones de automatización y control en la industria quedan limitados para tal fin. Hoy en día, existe hardware especializado capaz de ejecutar aplicaciones complejas de IA. La diversidad de dispositivos que existen en entornos industriales y la necesidad de que estos funcionen junto a dispositivos modernos se presenta como un reto. En este trabajo se busca integrar estas dos tecnologías: el hardware estandarizado para aplicaciones industriales con todas sus ventajas inherentes como modularidad, diversidad de protocolos de comunicación, robustez, entre otros, con esta nueva arquitectura de hardware capaz de ejecutar algoritmos complejos de IA como control inteligente. Para lograr esto, se desarrolló una metodología donde se enlazan un proceso a controlar, un dispositivo de control convencional (controlador lógico programable) y un dispositivo moderno (Nvidia Jetson Orin Nano Super) comunicados vía Modbus TCP/IP. Esto con el fin de generar una arquitectura capaz de adquirir y procesar datos, además de ejecutar tareas de control con redes neuronales artificiales. Se adquirieron datos de un proceso, los cuales se utilizaron para el entrenamiento de una red neuronal artificial. Posteriormente se implementó un control inteligente con la misma infraestructura de hardware y se comparó con una implementación de control PID. Los resultados de la comparación a través de pruebas estadísticas muestran que el control con redes neuronales artificiales es posible con la metodología utilizada.

Palabras clave: Control inteligente, industria 4.0, base de datos.

Methodology for Training Artificial Neural Networks Used as Intelligent Controllers in Industrial Processes

Abstract. The use of artificial intelligence (AI) applications in Industry 4.0 is essential today. However, the standard hardware used to implement automation and control applications in industry is limited for this purpose. Today, specialized hardware exists that can run complex AI applications. The diversity of devices in industrial environments and the need for them to work alongside modern devices

presents a challenge. This work seeks to integrate these two technologies: standardized hardware for industrial applications, with all its inherent advantages such as modularity, diverse communication protocols, and robustness, among others, with this new hardware architecture capable of running complex AI algorithms for intelligent control. To achieve this, a methodology was developed that links a process to be controlled, a conventional control device (programmable logic controller), and a modern device (Nvidia Jetson Orin Nano Super) communicating via Modbus TCP/IP. This aims to generate an architecture capable of acquiring and processing data, as well as executing control tasks with artificial neural networks. Data was acquired from a process and used to train an artificial neural network. Subsequently, intelligent control was implemented using the same hardware infrastructure and compared to a PID control implementation. The results of the comparison, obtained through statistical tests, demonstrate that control with artificial neural networks is feasible using the methodology employed.

Keywords: Intelligent control, industry 4.0, database.

1. Introducción

La industria 4.0 se ha caracterizado por la integración de tecnologías emergentes como lo son el Internet de las cosas, la Inteligencia Artificial (IA), la computación en la nube, entre otras[1]. En este marco se ha visto una vasta cantidad de aplicaciones de la IA, las cuales utilizan a las redes neuronales artificiales (RNA) como uno de sus núcleos [2].

Además, ha cobrado relevancia el concepto de Edge Computing, la cual se caracteriza por la cercanía al usuario, así como por un tratamiento local de los datos. De la mano de compañías como Nvidia se han desarrollado dispositivos que caben dentro de dicho concepto, como la tarjeta de desarrollo Jetson Orin Nano Super (JONS), además, poseen la capacidad para ejecutar RNA. Por otro lado, las herramientas de software moderno han permitido facilitar el desarrollo de aplicaciones basadas en RNA.

Los enlaces de comunicación industrial modernos deben ser estables, tener la capacidad para el almacenamiento de grandes cantidades de datos y tener la capacidad de integrar herramientas modernas de hardware y software. En consecuencia, en el área de instrumentación industrial se ha identificado como un reto la integración de las nuevas tecnologías con las ya existentes. Esta problemática se origina, en gran medida, debido a la diversidad de dispositivos utilizados en una red de control industrial [3].

Se han encontrado ventajas en el uso de aplicaciones de RNA para el control de procesos, ya sea como complemento a sistemas de control más tradicionales con el fin de mejorar el comportamiento de un sistema [4] o en la generación de algoritmos de control adaptable a través de la creación de bases de datos adecuadas [5].

Con la finalidad de aumentar la cantidad de aplicaciones de RNA en la industria, debido a las ventajas que presentan, se debe buscar la integración de hardware y software adecuado para dichos entornos. Además, se deben establecer metodologías claras para la creación de bases de datos para los entrenamientos, así como para las aplicaciones de RNA.

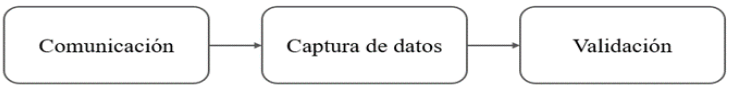

Fig. 1. Proceso para generación de base de datos.

2. Trabajos relacionados

Al ser necesaria la integración de múltiples dispositivos para la recopilación y uso de grandes cantidades de datos, surgen arquitecturas de sistemas con el fin de cubrir esa necesidad. Una de ellas es la presentada por Radlbauer, en la cual se prioriza la integración de diferentes protocolos de comunicación con el fin de utilizar tecnologías modernas como antiguas, centradas en la recolección y visualización de los datos [6].

En aplicaciones como el mantenimiento predictivo, como lo mostrado por Halenar, se han desarrollado sistemas donde se adiciona hardware con la capacidad para cubrir los requerimientos de almacenamiento y monitoreo de datos. Se muestran las capacidades del hardware moderno, se utilizan dispositivos de Edge computing, para la recopilación de datos útiles con la finalidad de un uso posterior por redes neuronales [7].

Los dispositivos de *Edge computing*, permiten adicionar o facilitar el uso de ciertas herramientas de software, por lo tanto, se vuelven muy útiles no solo para un tratamiento local de datos, sino también para funcionar como un intermediario entre sistemas locales y sistemas en la nube. Con el fin de poder escalar el almacenamiento de datos, Magnus presenta un sistema donde combina un sistema de *Edge computing* con almacenamiento en la nube [8].

Las herramientas de software desarrolladas para la integración de sistemas son muy útiles, sin embargo, aquellas que son cerradas y dependen de una compañía, como las utilizadas por Radlbauer, pueden tener limitaciones. La metodología de agregar capas a un proceso ya establecido con la finalidad de añadir funcionalidades nuevas, como la predicción del mantenimiento predictivo presentado por Halener, sin necesidad de alterar aquello que ya funciona se ve limitada cuando se quiere agregar funcionalidades directamente relacionadas con los sensores y actuadores del proceso.

Complementar sistemas locales con herramientas como la computación en la nube tiene ventajas, como la escalabilidad de almacenamiento de datos y el poder desplegarlos de maneras que un usuario pueda entenderlo con facilidad en cualquier dispositivo con acceso a la nube. Sin embargo, en situaciones donde no se tenga un acceso estable a estos sistemas, la selección de un hardware, software y métodos adecuados para optimizar el manejo de datos necesario para que herramientas como las redes neuronales funcionen adecuadamente, cobre relevancia.

Por lo anterior, en este trabajo se busca presentar una arquitectura donde se integre hardware, como controladores lógicos programables (PLC por sus siglas en inglés) y dispositivos que permitan el tratamiento cercano de los datos, así como protocolos de comunicación estandarizados, además de una metodología para recopilación de datos que sea útil para el entrenamiento de redes neuronales para control. Finalmente, se

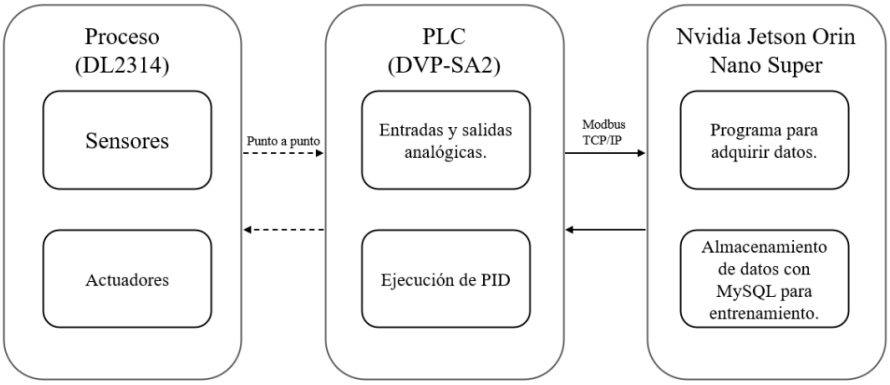

Fig. 2. Arquitectura de comunicación para toma de datos.

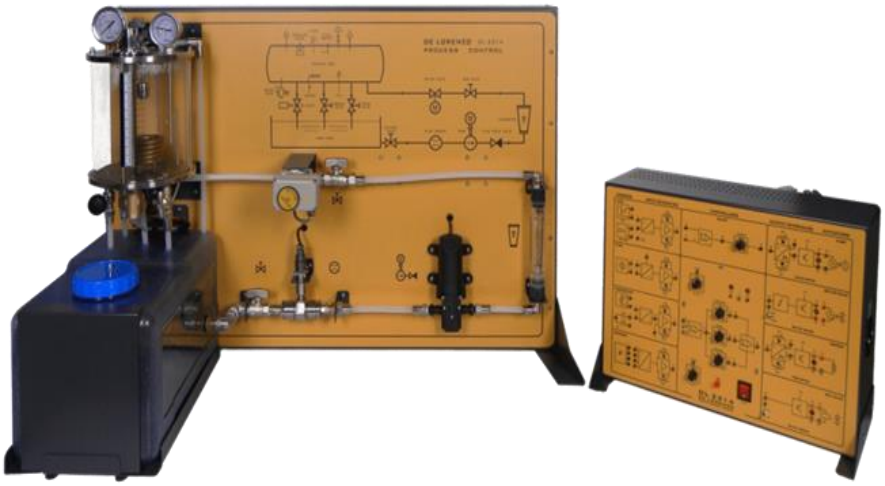

Fig. 3. Proceso utilizado, marca De Lorenzo modelo DL2314 [10].

prueba esta arquitectura con la implementación de un control inteligente de una variable en un proceso.

3. Metodología propuesta

El trabajo realizado se encuentra dividido en tres bloques, en la Fig. 1 representa las etapas que se deben realizar para la generación de una base de datos útil para el entrenamiento de una red neuronal. Las siguientes secciones se encuentran divididas de manera correspondiente a los bloques del sistema propuesto. En cada sección se detalla lo necesario para cada bloque, su funcionamiento y el resultado proporcionado.

3.1 Comunicación

En este bloque se integran los diferentes instrumentos de hardware necesarios para el almacenamiento y procesamiento de la información. Para ello se necesita un nodo con la capacidad de leer sensores y activar actuadores, además de poseer la capacidad de trasmitir los datos a través de un puerto de red, un nodo con la capacidad de procesar datos, además de una planta para utilizar sus respectivos sensores y actuadores.

La planta proporciona las señales de sensores, son capturadas por un PLC. Dicho PLC funciona como un servidor comunicado por el protocolo Modbus TCP/IP, lo cual permite el envío de datos a otro dispositivo en red que lo solicite y utilice el mismo protocolo de comunicación. Se seleccionó el protocolo al mostrar buenos resultados para el envío de datos [9].

La planta es un sistema de control de procesos, modelo DL2314 de la compañía De Lorenzo, se puede ver en la Fig. 3, el cual cuenta con un tanque equipado con sensores para medición de nivel con salida analógica de 0 a 10 vcd, además, para el llenado del tanque cuenta con una bomba, tres salidas de agua de activación manual y una salida con activación por electroválvula de dos posiciones [10]. La conexión con el hardware encargado de leer los datos proporcionados por el sensor es del tipo punto a punto.

Para la lectura de sensores y activación de actuadores se realiza con un PLC marca DELTA, modelo DVP-12SA2, adicionado con el módulo DVP-06XA-S para manipular señales analógicas y el módulo DVPEN01 para comunicación por red.

El nodo encargado del almacenamiento de los datos es una tarjeta de desarrollo de la marca Nvidia, modelo Jetson Orin Nano Super (JONS), el procesador es 6-core Arm (R) Cortex (R)-A78AE v8.2, cuenta con 8 GB RAM [11]. Para lograr la comunicación se configura como un cliente Modbus TCP/IP con el uso de Python y la librería pymodbus versión 2.1.0.

3.2 Captura de datos

Para obtener datos que una red neuronal pueda utilizar para replicar un control, se debe tener un sistema que los pueda generar. Con este fin se utilizó un control PID (proporcional-integral-derivativo) sintonizado a través del método de Ziegler-Nichols [12]. Entre los datos que se almacenaron, por cada marca de tiempo, para el posterior entrenamiento de la red neuronal se encuentran el error, la derivada del error, el valor deseado y la potencia de activación del actuador.

Para la ejecución del algoritmo de control PID y obtención de los datos, se programó el PLC con el software ISPSoft versión 3.21, en la Fig. 4 se muestra el diagrama de flujo. Una vez que se genera el dato es leído por el cliente y almacenado en una base de datos, para la gestión de esta última se utilizó MySQL.

El conjunto de datos de entrenamiento se generó a partir de la ejecución del algoritmo PID en un periodo de tiempo de 2400 segundos con un periodo de muestreo de 4 segundos. Durante este tiempo, se hicieron 30 cambios aleatorios del valor deseado (*setpoint*), con el fin de capturar la dinámica del sistema PID, dando un total de 600 conjuntos de datos almacenados en JONS. No fue necesario realizar ningún tratamiento a los datos posterior a su almacenamiento.

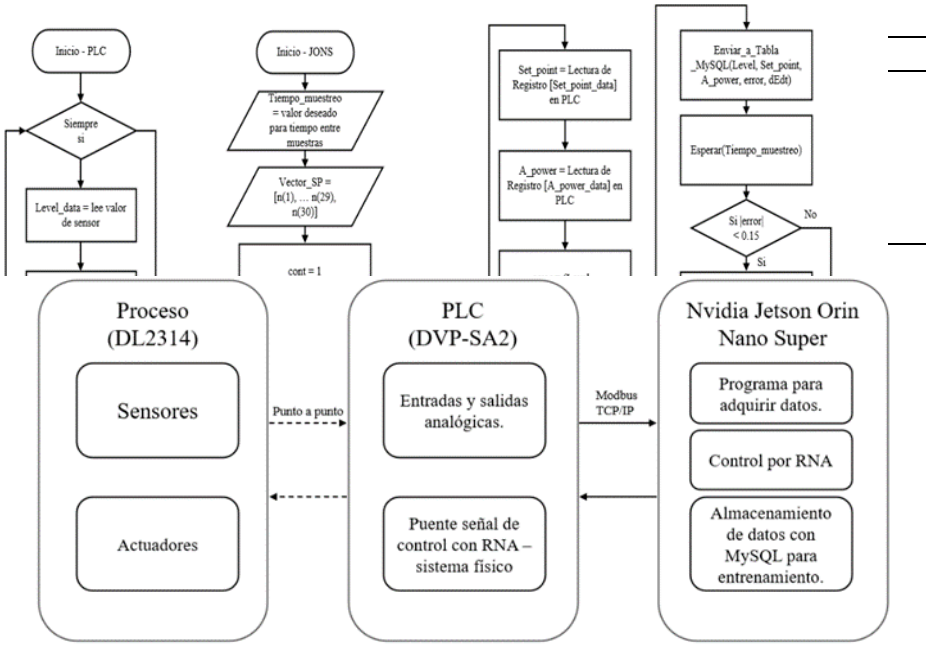

Fig. 5. Arquitectura de comunicación para ejecución de control inteligente con RNA.

Tiempo de muestreo

Con la finalidad de optimizar el almacenamiento, se debe seleccionar el tiempo de muestreo adecuado, se debe seguir un criterio que establezca el límite máximo del tiempo entre muestras. Para este caso, al ser la planta un tanque de almacenamiento de agua se considera como un sistema de primer orden [13], de tal modo que el intervalo se puede obtener a partir del ancho de banda del sistema, el tiempo de muestreo debe permitir tomar datos entre 20 y 40 veces el valor considerado del ancho de banda [14].

A partir de lo anterior, se establece el tiempo para la frecuencia del ancho de banda cuando el valor de la señal indique el 63.2% del cambio total, dada una entrada del tipo escalón con la potencia para permitir que el sistema llegue a un estado estable. Dicho valor de tiempo se obtuvo de manera experimental. De lo anterior se obtuvo un tiempo de 4 segundos entre muestras a guardadas en la base de datos. En la Tabla 1 se encuentra un resumen de la configuración utilizada para el entrenamiento de la red neuronal.

4. Validación

Para la validación de la base de datos como funcional para el entrenamiento de redes neuronales capaces de ejecutar tareas de control, se sustituyó el control PID ejecutado por el PLC con la red neuronal entrenada en ejecución sobre la tarjeta JONS, sobre la misma estructura de comunicación. se puede ver un esquema en la Fig. 5. Para la

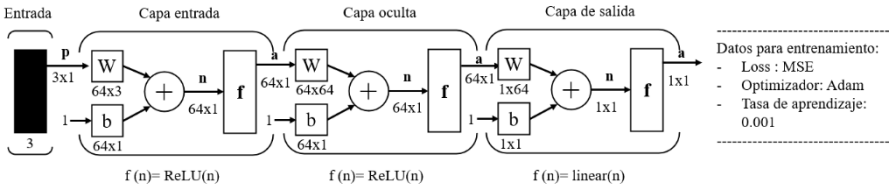

Fig. 6. Arquitectura de RNA utilizada para ejecución de control inteligente.

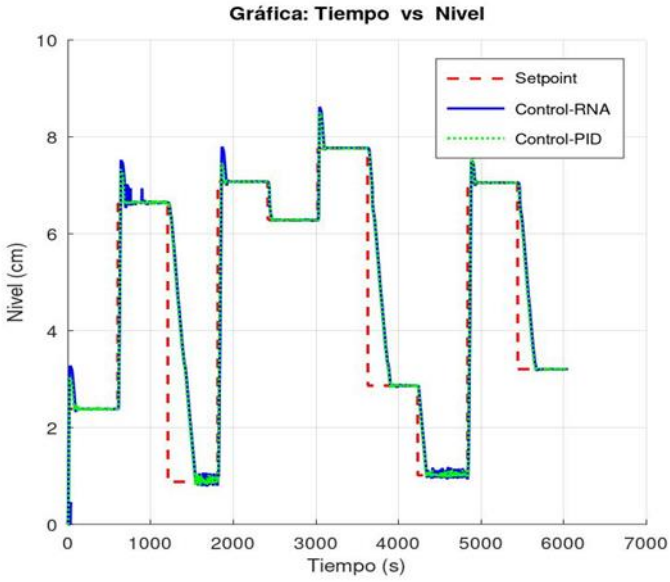

Fig. 7. Gráfica comparativa *Setpoint* /Control-RNA/Control-PID.

configuración de la red neuronal se utilizó Python y la librería Tensorflow versión 2.20. La red neuronal se entrenó un total de 809 épocas, por el mismo equipo de hardware encargado de ejecutarla. La estructura de la red neuronal utilizada se puede ver en la Fig. 6.

4.1 Resultados

Una vez entrenada la red neuronal se realizaron pruebas sobre diez valores deseados distintos, elegidos de forma aleatoria. Se comparo con el control PID, el comportamiento de ambos se puede ver en la Fig. 7. Para evaluar el comportamiento se utilizaron criterios vistos en trabajos relacionados [15], en este caso se utilizó el error cuadrático medio (MSE), el error absoluto medio (MAE) y el error estándar (SE). Además, se agregó la evaluación de resultados mediante la prueba de rangos con signos de Wilcoxon sobre el error absoluto de cada controlador respecto al *setpoint* [16].

Tabla 2. Comparativa de control con RNA y PID, con parámetro MSE.

Valor deseado (cm)	Control con RNA (cm^2)	Control con PID (cm^2)
2.380	0.00003	0.00003
6.640	0.00028	0.00042
0.875	0.00454	0.00372
7.065	0.00003	0.00003
6.280	0.00024	0.00010
7.755	0.00012	0.00003
2.860	0.00006	0.00036
1.010	0.00405	0.00209
7.045	0.00002	0.00001
3.200	0.00002	0.00003

Tabla 3. Comparativa de control con RNA y PID, con parámetro MAE.

Valor deseado (cm)	Control con RNA (cm)	Control con PID (cm)
2.380	0.0043	0.0028
6.640	0.0128	0.0178
0.875	0.0525	0.0480
7.065	0.0043	0.0035
6.280	0.0130	0.0080
7.755	0.0108	0.0043
2.860	0.0053	0.0158
1.010	0.0533	0.0335
7.045	0.0030	0.0020
3.200	0.0038	0.0043

Tabla 4. Comparativa de control con RNA y PID, con parámetro SE.

Valor deseado (cm)	Control con RNA (cm)	Control con PID (cm)
2.380	0.0011	0.0012
6.640	0.0038	0.0045
0.875	0.0147	0.0113
7.065	0.0009	0.0012
6.280	0.0032	0.0022
7.755	0.0007	0.0013
2.860	0.0017	0.0043
1.010	0.0139	0.0094
7.045	0.0009	0.0007
3.200	0.0008	0.0012

Tabla 5. Resumen de resultados de prueba de rangos con signo de Wilcoxon. Para la prueba se consideró Error_PID – Error_RNA.

Parámetro	Valores
Numero de diferencias no nulas	1229
W^+	582477
W^-	173358
W	173358
p	< 0.0001

Para los cálculos de evaluación en las tablas se consideraron las ultimas 20 muestras previas al cambio de valor deseado.

En la Tabla 2 se puede observar como la red neuronal supera al 30% y empata 20% de los casos al control PID. Por otro lado, en la Tabla 3, la red neuronal solo supera en 30% de los casos al control PID. Con el parámetro del error estándar plasmado en la Tabla 4, en 60% de los casos el control PID es superado por el control con RNA.

La prueba de rangos con signo de Wilcoxon aplicada sobre el error absoluto de ambos controladores, resumida en la Tabla 5, muestra que los valores representados por W^+ son mayores a los representados por W^-. Al representar a los errores de los controladores PID y la RNA respectivamente, se indica que el controlador PID presenta mayor error en la mayoría de los puntos evaluados.

5. Conclusión

En el presente trabajo se estableció una arquitectura de comunicación que permite, a través de un protocolo de comunicación abierto, integrar tres sistemas individuales: un proceso con sensores y actuadores, un controlador lógico programable como servidor con la capacidad de lectura de sensores y activación de actuadores y un tercer dispositivo como cliente capaz de almacenar y procesar datos de manera local, así como de ejecutar tareas de control inteligente.

Se generó una base de datos con información suficiente para el entrenamiento de una red neuronal, basándose en información generada por un control PID. La información fue almacenada en una tabla relacional gestionada por MySQL. Además, se estableció un tiempo de muestreo adecuado para optimizar el almacenamiento de los datos.

Se creó un control con redes neuronales artificiales entrenado con los datos del PID, posteriormente se implementó, evaluó y comparó con el mismo control utilizado para entrenarse. Los resultados muestran que la metodología para la creación de una base de datos es efectiva al alcanzar valores, con el control con RNA, cercanos a los valores del control utilizado para su entrenamiento, inclusive superándolo en algunos casos.

Agradecimientos. Se agradece al Centro de Investigación, Innovación y Desarrollo Tecnológico (CIIDETEC) en Culiacán por facilitar el uso de instalaciones y equipos para el desarrollo de este trabajo.

Referencias

1. Sharma, M., Tomar, A., Hazra, A.: Edge Computing for Industry 5.0: Fundamental, Applications, and Research Challenges. IEEE Internet Things J., Vol. 11, pp. 19070–19093 (2024). Doi:10.1109/JIOT.2024.3359297.
2. Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A.E., Arshad, H.: State-of-the-art in artificial neural network applications: A survey. Heliyon. 4, e00938 (2018). Doi:10.1016/j.heliyon.2018.e00938.
3. Sumarlin, T., Kusumajaya, R.A.: AI Challenges and Strategies for Business Process Optimization in Industry 4.0: Systematic Literature Review. Journal of Management and Informatics. Vol. 3, pp. 195–211 (2024). Doi:10.51903/jmi.v3i2.25.
4. Sahrir, N.H., Basri, M.A.M.: Intelligent PID Controller Based on Neural Network for AI-Driven Control Quadcopter UAV. International Journal of Robotics and Control Systems. Vol. 4, pp. 691–708 (2024). Doi:10.31763/ijrcs.v4i2.1374.
5. Yakout, A.H., Dashtdar, M., Aboras, K.M., Ghadi, Y.Y., Elzawawy, A., Yousef, A., Kotb, H.: Neural Network-Based Adaptive PID Controller Design for Over-Frequency Control in Microgrid Using Honey Badger Algorithm. IEEE Access. Vol. 12, pp. 27989–28005 (2024). Doi: 10.1109/ACCESS.2024.3367288.
6. Radlbauer, E., Moser, T., Wagner, M.: Designing a System Architecture for Dynamic Data Collection as a Foundation for Knowledge Modeling in Industry. Applied Sciences (Switzerland). Vol. 15, pp. 1–28 (2025). Doi: 10.3390/app15095081.
7. Halenar, I., Halenarova, L., Tanuska, P., Vazan, P.: Machine Condition Monitoring System Based on Edge Computing Technology. Sensors. Vol. 25, (2025). Doi:10.3390/s25010180.
8. Wahlgård, M., Krajnik, P., Stahre, J.: A Scalable Edge Computing Solution for Real-Time Process Monitoring and Optimization in Industrial IoT. Procedia CIRP., Vol. 134, Vol. 903–908 (2025). Doi:10.1016/j.procir.2025.02.219.
9. Tapia, E., Sastoque-Pinilla, L., Lopez-Novoa, U., Bediaga, I., López de Lacalle, N.: Assessing Industrial Communication Protocols to Bridge the Gap between Machine Tools and Software Monitoring. Sensors, Vol. 23, pp. 1–15 (2023). Doi:10.3390/s23125694.
10. Procesos, E.D.E.C.D.E.: Entrenador de control de procesos DL 2314 (2024)
11. Nvidia Corporation: NVIDIA Jetson Orin Nano Developer Kit (2023)
12. Ziegler, J.G., Nichols, N.B.: Optimum settings for automatic controllers. Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, Vol. 115, pp. 220–222 (1993). Doi:10.1115/1.2899060.
13. Ogata, K.: Dinamica de sistemas. Pretince Hall (2010)
14. Franklin, G., Powell, D., Emami-Naeini, A.: Feedback Control of Dynamic Systems. Pearson (1986). Doi:10.1109/MCS.1986.1105084.
15. Gerardo-Parra, C., Barreto-Salazar, L.E., Flores-López, L.M., Picos-Ponce, J.C., Castro-Palazuelos, D.E., Rubio-Astorga, G.J.: Development of a Neural Network-Based Controller for a Greenhouse Irrigation System at Laboratory Scale. Agriculture, Vol. 16, pp. 245 (2026). Doi:10.3390/agriculture16020245.
16. Montgomery, Douglas C; Runger, G.C.: Probabilidad y Estadistica aplicadas a la ingeniería (2000)

Redes neuronales unimodales para clasificar mensajes misóginos en redes sociales

Itzely Viridiana Medina Torres, Pablo Pancardo García,
Gabriel Omar Domínguez Ruiz

Universidad Juárez Autónoma de Tabasco,
División Académica de Ciencias y Tecnologías de la Información, Tabasco,
Mexico

242H21001@alumno.ujat.mx,pablo.pancardo@ujat.mx,
242h21002@alumno.ujat.mx

Resumen. La detección automática de mensajes misóginos en redes sociales representa un desafío relevante en el procesamiento del lenguaje natural para el español mexicano. Este trabajo presenta una comparación sistemática de cuatro arquitecturas de redes neuronales unimodales —LSTM, CNN, BETO y XLM-RoBERTa large— sobre un corpus fusionado de 6903 tweets extraídos de los conjuntos de datos MEX-A3T (IberEval 2018) y EXIST 2024 (IberLEF 2024). La distribución real del corpus exhibió un desequilibrio de clases de 76.1%/23.9%, identificado durante la fase experimental y abordado mediante ponderaciones proporcionales de las clases. Los resultados demuestran que BETO logró el mejor desempeño con un Macro-F1 de 85.28% y una precisión de 88.42%, superando a XLM-RoBERTa large (Macro-F1: 84.13%) a pesar de tener cinco veces menos parámetros. CNN obtuvo una puntuación Macro-F1 del 77.99 %, mientras que LSTM no logró una convergencia estable bajo desequilibrio de corpus sin preentrenamiento en español. Los resultados confirman que la especialización lingüística en español es más determinante que la capacidad paramétrica, y que los pesos de clase proporcionales superan al submuestreo en corpus de tamaño moderado.

Palabras clave: Redes neuronales unimodales, misoginia, transformers, BETO, XLM-RoBERTa, español mexicano, clasificación de texto, desequilibrio de clases.

Unimodal Neural Networks for Classifying Misogynistic Messages on Social Media

Abstract. The automatic detection of misogynistic messages in social networks represents a relevant challenge in natural language processing for Mexican Spanish. This work presents a systematic comparison of four unimodal neural network architectures, LSTM, CNN, BETO, and XLM-RoBERTa large, over a fused corpus of 6,903 tweets drawn from the MEX-A3T (IberEval 2018) and EXIST 2024 (IberLEF 2024) datasets. The actual corpus distribution exhibited a class imbalance of 76.1%/23.9%, identified during the experimental phase and

addressed through proportional class weights. Results demonstrate that BETO achieved the best performance with a Macro-F1 of 85.28% and Accuracy of 88.42%, outperforming XLM-RoBERTa large (Macro-F1: 84.13%) despite having five times fewer parameters. CNN obtained a Macro-F1 of 77.99%, while LSTM failed to achieve stable convergence under corpus imbalance without Spanish-language pretraining. The findings confirm that linguistic specialization in Spanish is more decisive than parametric capacity, and that proportional class weights outperform undersampling on moderately sized corpora.

Keywords: Unimodal neural networks, misogyny, transformers, BETO, XLM-RoBERTa, Mexican Spanish, text classification, class imbalance.

1. Introducción

La misoginia, definida como el desprecio, aversión o violencia simbólica hacia las mujeres, se manifiesta con frecuencia creciente en plataformas digitales y redes sociales [1]. De acuerdo con datos de la Organización Mundial de la Salud, una de cada tres mujeres ha experimentado alguna forma de violencia física o sexual, problemática que se ha extendido al espacio digital a través del lenguaje misógino en línea [2]. En el contexto latinoamericano, esta situación es especialmente preocupante: México reportó un incremento del 43% en casos de feminicidio en los últimos cinco años, con evidencia de correlación entre el discurso misógino en Twitter y estadísticas de violencia de género a nivel estatal [3].

La detección automática de este tipo de contenido mediante técnicas de procesamiento del lenguaje natural (PLN) presenta desafíos específicos: la misoginia es frecuentemente sutil, puede ocultarse tras palabras aparentemente neutras, humor o ironía, y presenta particularidades lingüísticas y culturales que varían entre regiones hispanohablantes [4]. Estos factores hacen que los modelos entrenados en corpus genéricos o en inglés obtengan resultados subóptimos cuando se aplican al español mexicano.

Los trabajos previos en detección de misoginia en español han explorado principalmente clasificadores de aprendizaje automático clásicos — SVM, Naïve Bayes, RNA feedforward — sobre datasets de tamaño reducido, reportando resultados de hasta 82.59% de accuracy [5]. La introducción de modelos Transformer preentrenados como BERT ha transformado el estado del arte en clasificación de texto, pero su aplicación al español mexicano bajo condiciones de desbalance de clases pronunciado no ha sido ampliamente estudiada en comparaciones controladas que incluyan también modelos multilingüe de alta capacidad.

Este trabajo aborda estas brechas realizando una comparación sistemática de cuatro arquitecturas neuronales unimodales sobre un corpus fusionado de los datasets MEX-A3T y EXIST 2024, con particular atención al tratamiento del desbalance de clases emergente y al papel de la especialización lingüística en el rendimiento de los modelos Transformer. Las contribuciones principales son:

– Evidencia empírica de que BETO (modelo monolingüe español, 110M parámetros) supera a XLM-RoBERTa large (modelo multilingüe, 560M parámetros) para la

detección de misoginia en español mexicano, confirmando la primacía de la especialización lingüística sobre la capacidad paramétrica.

- Demostración de que class weights proporcionales superan al undersampling para corpus desbalanceados de tamaño moderado, preservando la totalidad del conjunto de entrenamiento.
- Análisis del comportamiento de LSTM ante desbalance severo sin preentrenamiento, documentando el colapso hacia la clase mayoritaria independientemente de la configuración de regularización o los embeddings empleados.
- Identificación y documentación del desbalance de clases emergente al fusionar MEX-A3T y EXIST 2024, atribuible a diferencias en el esquema de etiquetado entre datasets.

2. Trabajos relacionados

2.1. Detección de misoginia en español

La detección de misoginia en español ha sido impulsada principalmente por las competiciones IberEval e IberLEF. Fersini et al. [1] organizaron la primera tarea compartida de identificación automática de misoginia (AMI) en IberEval 2018, introduciendo el dataset MEX-A3T con 3,307 tweets en español mexicano. Los mejores sistemas participantes en esta competición reportaron accuracy de entre 70% y 81% utilizando enfoques basados en SVM y representaciones TF-IDF.

Frenda y Bilal [6] exploraron la misoginia en tweets españoles e ingleses del dataset AMI, demostrando que la lematización mediante Freeling (herramienta de análisis morfológico y lematización) mejora el rendimiento de los clasificadores en español, mientras que en inglés lo reduce. Vera Lagos [5] implementó clasificadores supervisados (MNB, SVM, RNA feedforward) sobre un corpus propio de 1,100 tweets mexicanos balanceado 50/50, alcanzando 82.59% de accuracy con RNA y representación de bigramas. Su trabajo evidenció la escasez de recursos etiquetados específicamente para el español mexicano y las limitaciones de herramientas de preprocesamiento no adaptadas al vocabulario coloquial mexicano.

García-Díaz et al. [7] propusieron un enfoque basado en características lingüísticas y representaciones vectoriales de palabras para la detección de misoginia en tweets españoles, alcanzando accuracy de 85.17% sobre el corpus MisoCorpus-2020 (7,682 tweets). Aldana-Bobadilla et al. [8] desarrollaron un modelo basado en BERT con extracción multifuente de características (web crawler, letras de canciones, proverbios) para el español latinoamericano, reportando mediana de accuracy de 0.88 sobre datos de aprendizaje y 0.92 sobre datos del mundo real. Su trabajo es el más cercano al nuestro en términos del modelo base, aunque utiliza BERT como extractor de características con Regresión Logística, no ajuste fino de extremo a extremo.

Tabla 1. Estadísticas de los datasets utilizados.

Dataset	Total	Misógino	No Misógino
MEX-A3T	3,307	1,577 (47.7%)	1,730 (52.3%)
EXIST 2024	3,660	3,633 (99.3%)	27 (0.7%)*
Fusionado	6,903	5,250 (76.1%)	1,653 (23.9%)

* Tras armonización de etiquetas. La mayoría de registros EXIST no misóginos no coincidieron con el esquema binario.

2.2. Modelos transformer para español

BETO (dccuchile/bert-base-spanish-wwm-cased) [9] es una adaptación de BERT entrenada sobre 3,000 millones de palabras en español mediante whole word masking, estableciéndose como referencia para tareas de PLN en español. Rodríguez et al. [10] aplicaron preentrenamiento adaptativo al dominio sobre BETO para la detección de tweets misóginos usando el dataset AMI, demostrando mejoras significativas respecto al fine-tuning estándar.

XLM-RoBERTa [11] es un modelo multilingüe entrenado sobre 2.5 TB de texto en 100 idiomas; aunque su mayor capacidad paramétrica sugiere potencial superioridad, estudios comparativos recientes en español [12] han mostrado que modelos monolingüe como BETO y MarIA obtienen mejores resultados en tareas de clasificación de texto en español que modelos multilingüe de mayor tamaño, motivando la comparación directa realizada en este trabajo.

3. Metodología

3.1. Datasets y fusión

Se utilizaron dos datasets públicos en español. El primero, MEX-A3T, proviene del shared task AMI de IberEval 2018 [1] y contiene 3,307 tweets en español mexicano etiquetados en forma binaria (misógino/no misógino) con distribución 47.7%/52.3%. El segundo, EXIST 2024 de IberLEF 2024 [13], contiene 3,660 tweets etiquetados en múltiples dimensiones de sexismo bajo el paradigma Learning with Disagreement. Para la tarea binaria se armonizaron las etiquetas: MEX-A3T misogynous=1 → misógino; EXIST labels_task1="YES" → misógino.

Tras la fusión y eliminación de duplicados, el corpus contiene 6,903 mensajes. Un hallazgo relevante fue la distribución resultante: 76.1% misógino / 23.9% no misógino (ratio 1:3.18), significativamente más desbalanceada que los datasets individuales. Este desbalance se originó porque EXIST 2024 aportó exclusivamente registros de clase misógina bajo el esquema de etiquetado armonizado, dado que sus categorías de sexismo no misógino no correspondían con la etiqueta "no misógino" de MEX-A3T. La Tabla 1 resume las estadísticas de los datasets.

4. Preprocesamiento

El preprocesamiento aplicado a ambos datasets incluyó: conversión a minúsculas, eliminación de URLs (regex http/www), eliminación de menciones (@usuario), eliminación de caracteres especiales conservando letras, dígitos, espacios y caracteres del español (á, é, í, ó, ú, ñ, ü), y normalización de espacios. No se aplicó eliminación de stopwords ni lematización, dado que los modelos Transformer manejan el contexto semántico completo mediante mecanismos de atención, y la CNN captura n-gramas relevantes independientemente de las palabras funcionales.

Se evaluaron dos estrategias de compensación del desbalance: (a) class weights proporcionales calculados con scikit-learn balanced (Clase 0: 2.0872; Clase 1: 0.6575), aplicados a la función de pérdida, preservando los 6,903 registros de entrenamiento; (b) undersampling proporcional hasta distribución 50/50 (3,306 registros, reducción del 52.1%). Los datos se dividieron estratificadamente en 70% entrenamiento (4,834), 15% validación (1,033) y 15% prueba (1,036).

5. Arquitecturas evaluadas

5.1. LSTM

Se configuró con una capa de embedding de 128 dimensiones, SpatialDropout1D (0.1), capa LSTM de 64 unidades con dropout de 0.3 y dropout recurrente de 0.2, BatchNormalization, capa densa de 32 unidades con activación ReLU, dropout de 0.3 y salida sigmoide; optimizador Adam con lr=1e-3. Se evaluaron sistemáticamente variantes con embeddings aleatorios, embeddings FastText preentrenados (cc.es.300.vec), y múltiples configuraciones de regularización (dropout 0.2–0.5, L2 1e-4–1e-3).

En todos los casos el modelo colapsó hacia la clase mayoritaria (TN≈0, Macro-F1<50%), comportamiento atribuible a la combinación del desbalance 1:3.18 con la ausencia de representaciones preentrenadas en español: sin contexto semántico previo, 4,834 muestras de entrenamiento resultaron insuficientes para aprender representaciones discriminativas bajo este nivel de desbalance.

Este análisis se propone como trabajo futuro para explorar técnicas de data augmentation que pudieran habilitar la convergencia de LSTM.

5.2. CNN

Se empleó una capa de embedding de 128 dimensiones, SpatialDropout1D (0.2), tres ramas convolucionales paralelas con filtros de tamaños 2, 3 y 4 palabras (64 filtros cada una), GlobalMaxPooling1D, concatenación, BatchNormalization, capa densa de 64 unidades con activación ReLU, dropout de 0.4 y salida sigmoide; optimizador Adam con lr=1e-3 [14].

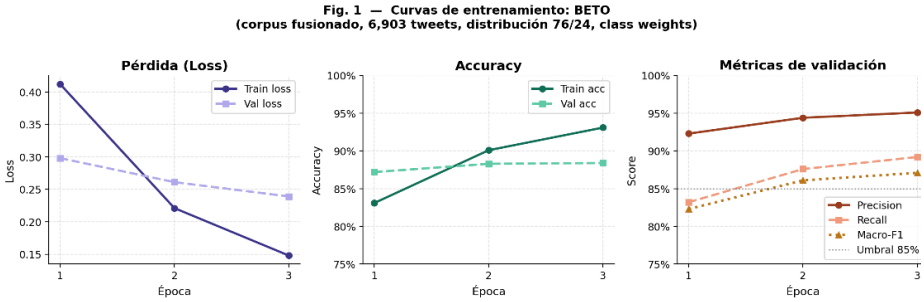

Fig. 1. Curvas de entrenamiento de BETO. Pérdida (Loss), Accuracy y métricas de validación (Precision, Recall, Macro-F1) por época sobre el corpus fusionado (6,903 tweets, 76/24, class weights). La línea punteada indica el umbral Macro-F1 ≥ 85%.

5.3. BETO

Se aplicó ajuste fino del modelo dccuchile/bert-base-spanish-wwm-cased [9] (110M parámetros), con lr=1e-5, AdamW con weight decay=0.05, dropout=0.2, warmup del 20%, tamaño de lote de 16 y máximo de 3 épocas con early stopping sobre val_loss (paciencia=2).

5.4. XLM- RoBERTa large

Se utilizó el modelo xlm-roberta-large [11] (560M parámetros, entrenado en 100 idiomas), con una configuración idéntica a la de BETO para garantizar una comparación controlada.

Todos los experimentos se realizaron en Google Colaboratory con GPU NVIDIA Tesla T4 (16 GB), Python 3.12, TensorFlow 2.19/Keras para LSTM y CNN, y PyTorch 2.x con Transformers 4.x para los modelos Transformer.

Por restricciones computacionales — especialmente el costo de XLM-RoBERTa large (~40 min/época en T4) — cada arquitectura se evaluó con su configuración óptima en una partición fija. La aplicación de validación cruzada k-fold y múltiples semillas se propone como trabajo futuro para habilitar pruebas de significancia estadística.

6. Resultados y discusión

6.1. Curvas de entrenamiento — BETO

La Fig. 1 muestra las curvas de entrenamiento de BETO sobre el corpus fusionado. La pérdida de validación decrece consistentemente durante las tres épocas sin divergencia respecto a la pérdida de entrenamiento, evidenciando ausencia de sobreajuste.

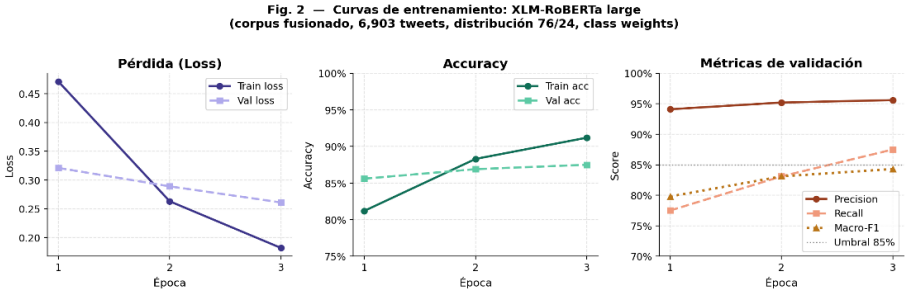

Fig. 2. Curvas de entrenamiento de XLM-RoBERTa large. Pérdida, Accuracy y métricas de validación por época sobre el corpus fusionado. La línea punteada indica el umbral Macro-F1 $\geq$ 85%.

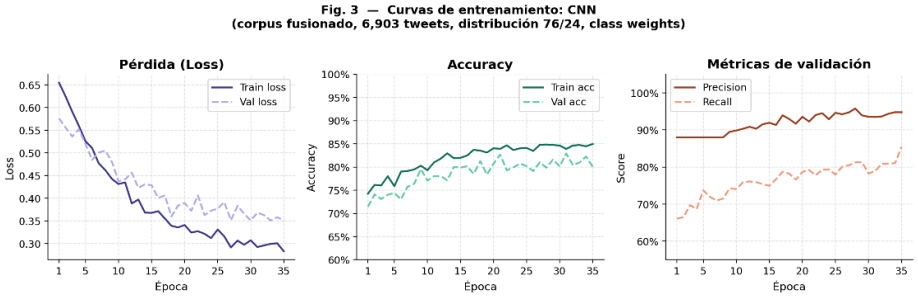

Fig. 3. Curvas de entrenamiento de la CNN. Pérdida (Loss), Accuracy, Precision y Recall por época sobre el corpus fusionado (6,903 tweets, 76/24).

El Macro-F1 en validación supera el umbral del 85% desde la época 2 (0.861), estabilizándose en 0.871 en la época 3. La convergencia estable y rápida se atribuye al preentrenamiento en 3,000 millones de palabras en español, que proporciona representaciones contextuales ricas desde el inicio del fine-tuning.

6.2. Curvas de entrenamiento — XLM-RoBERTa large

La Fig. 2 presenta las curvas de entrenamiento de XLM-RoBERTa large. Al igual que BETO, las curvas muestran convergencia estable sin sobreajuste. Sin embargo, la Precision se mantiene cercana a 0.96 durante las tres épocas mientras el Recall parte de 0.775 y escala hasta 0.875, sugiriendo que el modelo tarda más en aprender representaciones discriminativas para la clase minoritaria.

El Macro-F1 en validación llega a 0.843 en la época 3, sin alcanzar el umbral del 85%. Este comportamiento refleja las limitaciones del preentrenamiento multilingüe para capturar el matiz lingüístico específico del español mexicano con solo 3 épocas de fine-tuning.

Tabla 2. Resultados en conjunto de prueba — dataset fusionado con class weights.

Modelo	Acc. (%)	Macro-F1 (%)	Prec. (%)	Rec. (%)	FP	FN	Hipótesis
BETO	88.42	85.28	95.13	89.21	29	85	Sí (>85%)
XLM-RoBERTa large	87.45	84.13	95.57	87.56	32	98	No
CNN	81.85	77.99	93.99	81.35	41	147	No
LSTM	—	—	—	—	—	—	No converge

6.3. Curvas de entrenamiento — CNN

La Fig. 3 muestra las curvas de entrenamiento de la CNN a lo largo de 35 épocas. Las curvas de pérdida de entrenamiento y validación convergen de manera alineada desde las primeras épocas, confirmando la ausencia de sobreajuste tras la eliminación de la regularización L2.

La Precision de validación se estabiliza en torno a 0.950 desde la época 10, mientras que el Recall converge a 0.810, reflejando la tendencia de la CNN a priorizar la clase mayoritaria (misógino) al capturar patrones léxicos recurrentes vía filtros convolucionales de n-gramas.

La convergencia en ~35 épocas es notablemente más lenta que los modelos Transformer (3 épocas), aunque el costo computacional por época es significativamente menor.

7. Comparación de resultados

La Tabla 2 resume los resultados de las cuatro arquitecturas sobre el conjunto de prueba (1,036 mensajes).

Para LSTM se reportan los valores cuantitativos observados en todas las configuraciones exploradas: en todos los casos el modelo presentó TN≈0 (falsos negativos tendentes a cero para la clase positiva), Recall de clase minoritaria≈0.00 y Macro-F1<50%, equivalente a clasificación aleatoria sesgada hacia la clase mayoritaria.

BETO es el único modelo que confirma la hipótesis (Macro-F1 ≥ 85%), alcanzando 85.28% con solo 29 falsos positivos y 85 falsos negativos. XLM-RoBERTa large obtiene 84.13% de Macro-F1, inferior en 1.15 puntos a pesar de sus 560M parámetros. Este resultado es consistente con la literatura que documenta la superioridad de modelos monolingüe sobre multilingüe para tareas de PLN en lenguas específicas [12]: la especialización de BETO en el corpus español — incluyendo lenguaje coloquial y argot

de Twitter — proporciona representaciones semánticas más ajustadas que el preentrenamiento multilingüe generalista. Cabe destacar que XLM-RoBERTa obtiene la mayor Precision (95.57%, 32 FP), posicionándose como alternativa preferible cuando minimizar la censura de contenido legítimo es prioritario.

CNN alcanza Macro-F1 de 77.99% con ~1.4M parámetros, representando el 91.5% del Macro-F1 de BETO con el 1.3% de sus parámetros, confirmando su posición como mejor alternativa en términos de balance costo-rendimiento. El modelo LSTM con embeddings aleatorios (sin preentrenamiento) actúa implícitamente como línea base sin preentrenamiento, ilustrando el beneficio absoluto del preentrenamiento: frente al colapso total de LSTM, CNN —con embeddings entrenables desde cero— logra 77.99% de Macro-F1, y los modelos Transformer con preentrenamiento masivo en español alcanzan 84–85%.

Los falsos negativos de BETO (85 casos) corresponden predominantemente a tweets donde el lenguaje misógino aparece codificado en ironía, humor o argot coloquial mexicano no recurrente en el corpus de preentrenamiento. Los falsos positivos (29 casos) corresponden mayoritariamente a tweets que discuten o citan violencia de género usando vocabulario misógino sin ser ellos mismos misóginos. Por restricciones de privacidad y extensión, no se reproducen ejemplos directos de tweets, pero estos patrones de error sugieren que técnicas de augmentación de datos orientadas a ironía y lenguaje coloquial podrían mejorar los resultados futuros.

8. Conclusiones

Este trabajo presentó una comparación sistemática de cuatro arquitecturas de redes neuronales unimodales para la detección automática de mensajes misóginos en español mexicano, abordando el problema del desbalance de clases emergente al fusionar los datasets MEX-A3T y EXIST 2024. Los resultados conducen a cuatro conclusiones principales.

Primero, la especialización lingüística en español es más determinante que la capacidad paramétrica: BETO (110M parámetros) supera a XLM-RoBERTa large (560M parámetros) en Macro-F1 (85.28% vs 84.13%) y en reducción de errores (29 vs 32 FP; 85 vs 98 FN). Este hallazgo es consistente con evidencia previa en la literatura [12] y aporta sustento empírico específico para el dominio de detección de misoginia en español mexicano.

Segundo, los pesos de clase proporcionales superan al undersampling preservando el volumen de información de entrenamiento y obteniendo mejoras de hasta 1.56 puntos en Macro-F1 respecto a la reducción del corpus. Tercero, LSTM sin preentrenamiento en español no logra convergencia estable ante desbalance 1:3.18 con corpus moderado, delimitando las condiciones bajo las que las arquitecturas recurrentes resultan viables para detección de discurso de odio. Cuarto, CNN con ~1.4M parámetros representa la alternativa con mejor balance costo-rendimiento, alcanzando el 91.5% del Macro-F1 de BETO con el 1.3% de sus parámetros.

Como trabajo futuro se propone explorar: (a) modelos RoBERTa monolingüe en español como MarIA-BNE; (b) data augmentation textual para la clase minoritaria; (c)

clasificación multiclase de categorías de misoginia; y (d) evaluación en corpus de otras plataformas.

Referencias

1. Fersini, E., Rosso, P., Anzovino, M.: Overview of the Task on Automatic Misogyny Identification at IberEval 2018. IberEval@SEPLN 2150, pp. 214–228 (2018)
2. WHO: Global and Regional Estimates of Violence against Women. World Health Organization, Geneva (2013)
3. Molina-Villegas, A.: La incidencia de las voces misóginas sobre el espacio digital en México. In: Jóvenes, Plataformas Digitales y Lenguajes. Elementum, Pachuca (2021)
4. Hewitt, S., Tiropanis, T., Bokhove, C.: The problem of identifying misogynist language on Twitter. In: ACM Web Science, pp. 333–335 (2016)
5. Vera Lagos, V.: Detección de misoginia en textos cortos mediante clasificadores supervisados. Tesis de Licenciatura, BUAP, México (2021)
6. Frenda, S., Bilal, G.: Exploration of Misogyny in Spanish and English Tweets. In: IberEval 2018, Vol. 2150, pp. 260–267 (2018)
7. García-Díaz, J.A., Cánovas-García, M., Colomo-Palacios, R., Valencia-García, R.: Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings. Future Gener. Comput. Syst., Vol. 114, pp. 506–518 (2021)
8. Aldana-Bobadilla, E., Molina-Villegas, A., Montelongo-Padilla, Y., Lopez-Arevalo, I., Sordia, O.S.: A Language Model for Misogyny Detection in Latin American Spanish Driven by Multisource Feature Extraction and Transformers. Appl. Sci., Vol. 11, 10467 (2021)
9. Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., Pérez, J.: Spanish Pre-Trained BERT Model and Evaluation Data. In: PML4DC at ICLR (2020)
10. Rodríguez, D.A., Diaz-Escobar, J., Díaz-Ramírez, A. et al.: Domain-adaptive pre-training on a BERT model for the automatic detection of misogynistic tweets in Spanish. Soc. Netw. Anal. Min., Vol. 13, pp. 126 (2023)
11. Conneau, A., et al.: Unsupervised Cross-lingual Representation Learning at Scale. In: ACL 2020, pp. 8440–8451 (2020)
12. Salmerón-Ríos, S., et al.: Evaluation of transformer models for tasks in Spanish. PeerJ Comput. Sci. (2024). Doi:10.7717/peerj-cs.1992.
13. Plaza-del-Arco, F.M., et al.: Overview of EXIST 2024 — Learning with Disagreement for Sexism Identification and Characterization. In: CLEF (2024)
14. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: EMNLP, pp. 1746–1751 (2014)
15. Pereira, A.F., et al.: AI-UPV at IberLEF-2021 EXIST task: Sexism Prediction Using Monolingual and Multilingual BERT. In: IberLEF@SEPLN (2021)

Sistema inteligente de detección y asistencia para personas con discapacidad en cruces peatonales utilizando visión por computadora

Martin Chavez-Jaramillo, Cesar Benavides-Alvarez, Israel Santoyo-Luévano

Universidad Autónoma Metropolitana, Unidad Azcapotzalco,
Departamento de Electrónica, Ciudad de México,
México

{al2163036728, cesarbenavides, issl}@azc.uam.mx

Resumen. Los sistemas de semaforización convencionales carecen de mecanismos inclusivos para personas con movilidad reducida durante los intervalos de cruce peatonal. Esta ausencia representa un riesgo significativo en entornos de tráfico vehicular denso, especialmente para usuarios de silla de ruedas que requieren tiempos de cruce más prolongados. Estudios previos en movilidad urbana indican que la falta de adaptación en los cruces semaforizados incrementa la siniestralidad de este colectivo en intersecciones de alto flujo vehicular. En el presente trabajo se propone un sistema inteligente de detección en tiempo real mediante visión por computadora e inteligencia artificial. Para ello, se implementó el modelo YOLO11 optimizado al formato TensorFlow Lite sobre una placa Raspberry Pi 5, integrando un módulo de control de señalizaciones visuales compuesto por LEDs y una pantalla OLED, el cual prioriza automáticamente el tiempo de cruce al detectar al usuario. El modelo fue entrenado con un dataset de 1,100 imágenes etiquetadas en condiciones diurnas y nocturnas, empleando aumentación de datos para simular oclusión parcial y diferentes perspectivas. Los resultados obtenidos muestran una precisión de 82.9 % en la métrica mAP50-95 bajo condiciones controladas, con detección efectiva a distancias de hasta 4.5 metros y un tiempo de respuesta inferior a 2 segundos. Estos hallazgos demuestran la viabilidad y escalabilidad técnica de implementar soluciones de asistencia tecnológica de bajo costo y alta eficiencia para la movilidad inclusiva urbana. Como trabajo futuro, se plantea integrar sensores de profundidad y comunicación inalámbrica con semáforos existentes, así como extender la validación a usuarios con bastones o andadores.

Palabras clave: Visión por computadora, detección de objetos en tiempo real, movilidad inclusiva, sistemas embebidos, semaforización inteligente.

Intelligent Detection and Assistance System for People with Disabilities at Pedestrian Crossings Using Computer Vision

Abstract. Conventional traffic light systems lack inclusive mechanisms for people with reduced mobility during pedestrian crossing intervals. This absence represents a significant risk in dense vehicular traffic environments, especially for wheelchair users who require longer crossing times. Previous studies on urban mobility indicate that the lack of adaptation at signalized crosswalks increases the accident rate among this group at high-traffic intersections. This paper proposes an intelligent real-time detection system based on computer vision and artificial intelligence. To this end, the YOLO11 model, optimized to the TensorFlow Lite format, was implemented on a Raspberry Pi 5 board, integrating a control module for visual signals composed of LEDs and an OLED screen, which automatically prioritizes crossing time upon detecting the user. The model was trained on a dataset of 1,100 labeled images under daytime and nighttime conditions, using data augmentation to simulate partial occlusion and different viewing angles. The results obtained show a precision of 82.9% in the mAP50-95 metric under controlled conditions, with effective detection at distances of up to 4.5 meters and a response time of less than 2 seconds. These findings demonstrate the technical feasibility and scalability of implementing low-cost, high-efficiency technological assistance solutions for inclusive urban mobility. As future work, the integration of depth sensors and wireless communication with existing traffic lights is proposed, as well as extending validation to users with canes or walkers.

Keywords: Computer vision, real-time object detection, inclusive mobility, embedded systems, intelligent traffic light system.

1. Introducción

En las ciudades modernas, el derecho a la movilidad representa un pilar fundamental para la inclusión ciudadana. Sin embargo, la infraestructura vial actual no cuenta con un diseño adecuado para garantizar seguridad y autonomía a las personas en silla de ruedas, particularmente en los cruces peatonales. Esta situación constituye un problema social de relevancia, dado que la ausencia de señalización adaptada genera riesgos elevados en entornos de tráfico vehicular denso.

El sistema de semaforización tradicional en México únicamente contempla alarmas sonoras orientadas a personas invidentes, omitiendo las necesidades de usuarios en silla de ruedas, quienes requieren tiempos de cruce más prolongados [6]. Esta carencia vulnera los principios universales de accesibilidad vial y limita la integración autónoma de las personas con discapacidad en el entorno urbano.

Ante este escenario, la inteligencia artificial (IA) y, en particular, la visión por computadora, han demostrado ser herramientas eficaces para abordar problemáticas de accesibilidad y seguridad vial. Diversas investigaciones han explorado el uso de modelos de detección de objetos basados en redes neuronales convolucionales para asistir a poblaciones vulnerables en entornos urbanos [6][7][10]. No obstante, la mayoría de estas soluciones se desarrollan sobre infraestructura costosa o de difícil implementación en países en desarrollo.

En este contexto, el presente trabajo propone un sistema inteligente basado en visión por computadora e inteligencia artificial con el objetivo de desarrollar un prototipo a escala capaz de detectar usuarios en silla de ruedas en tiempo real, mediante una arquitectura embebida de bajo costo sustentada en una placa Raspberry Pi 5. El sistema integra el modelo YOLO11, versión más reciente de la familia *You Only Look Once*, optimizado para dispositivos de cómputo reducido, contribuyendo así a la autonomía, seguridad vial y accesibilidad urbana de personas con movilidad reducida [5][12].

2. Estado del arte

La visión por computadora es un subcampo de la inteligencia artificial que permite a los sistemas computacionales interpretar y analizar información visual del mundo real a partir de imágenes o secuencias de video. Su potencial radica en el uso del aprendizaje profundo (*Deep Learning*), subcapa del aprendizaje automático (*Machine Learning*), que entrena modelos para simular la percepción visual humana en entornos complejos. El componente fundamental de este enfoque son las redes neuronales convolucionales (RNC), estructuras compuestas por capas de filtros capaces de extraer características visuales progresivamente, desde formas geométricas simples hasta escenas de alta complejidad. Sobre esta base técnica se construyen los modelos de detección de objetos en tiempo real, siendo la familia YOLO (*You Only Look Once*) uno de los enfoques más adoptados en la literatura reciente [1][9].

Diversas investigaciones han explorado el uso de modelos YOLO para asistir a personas con discapacidad o movilidad reducida en entornos urbanos. Louis et al. [7] desarrollaron un sistema avanzado de asistencia al conductor de silla de ruedas eléctrica, basado en detección, estimación de profundidad y seguimiento de objetos mediante YOLOv3 y una cámara Intel RealSense, con el objetivo de construir mapas semánticos tridimensionales del entorno inmediato. En una línea similar, Ainandafiq et al. [2] propusieron un sistema de navegación autónoma para sillas de ruedas en interiores empleando YOLOv8 con módulos de atención geográfica, donde el mecanismo de Atención de Coordenadas (AC) alcanzó una precisión del 99.5 %, superando al Módulo de Atención de Bloques Convolucionales (MABC) con un 96.15 %. Por su parte, Shisei et al. [10] abordaron la monitorización de usuarios de silla de ruedas en habitaciones privadas mediante imágenes térmicas de baja resolución y YOLOv8 en su variante *nano*, priorizando la eficiencia en latencia y consumo de hardware.

Otros trabajos han centrado su atención en la detección de peatones en situaciones de riesgo vial. Lee et al. [6] implementaron un algoritmo YOLOv5 específicamente orientado a la identificación de peatones que utilizan dispositivos de asistencia, como sillas de ruedas y muletas, en pasos de cebra, dada su mayor vulnerabilidad ante accidentes de tráfico. Hsu y Lin [4] abordaron la problemática de las oclusiones en la detección peatonal, donde factores como la vestimenta, obstrucciones parciales o condiciones de iluminación reducida disminuyen la precisión del modelo; para ello, combinaron YOLOv4 con técnicas de segmentación y fusión adaptativa multirresolución, logrando mayor robustez en escenarios adversos. Complementando este enfoque, Tomás et al. [11] propusieron la evaluación de accesibilidad de rutas peatonales para usuarios de silla de ruedas mediante imágenes aéreas y geográficas procesadas con YOLOv8, aportando una perspectiva de planificación urbana inclusiva.

La rápida evolución de la familia YOLO ha sido documentada sistemáticamente en la literatura. Marco et al. [9] realizaron una revisión exhaustiva del algoritmo aplicado al reconocimiento de señales de tráfico, evaluando cinco versiones (YOLOv1 a YOLOv5) en términos de aplicaciones, conjuntos de datos, métricas, hardware y desafíos. En la misma dirección, Ahmed et al. [1] presentaron un *benchmark* comparativo entre YOLOv4 y YOLOv8, demostrando que es posible simplificar la arquitectura del modelo sin comprometer el rendimiento en detección en tiempo real. Khanam y Hussain [5] documentaron los avances arquitectónicos de YOLO11, la versión más reciente de la familia, destacando la introducción del bloque C3K2, los mecanismos SPPF y C2PSA, así como una reducción del 22 % en parámetros respecto a YOLOv8, manteniendo una mayor precisión media promedio (mAP) [12].

A pesar de los avances descritos, la mayoría de las soluciones revisadas se enfocan en asistencia en interiores, navegación autónoma o análisis de rutas, sin abordar de forma integral la problemática de la semaforización inclusiva en cruces peatonales urbanos. Asimismo, varias de estas implementaciones dependen de hardware especializado de alto costo, lo que limita su escalabilidad en contextos con recursos reducidos. El presente trabajo propone cubrir esta brecha mediante la implementación de YOLO11 sobre una plataforma embebida de bajo costo (Raspberry Pi 5), orientada específicamente a la detección de usuarios en silla de ruedas en tiempo real y a la activación automatizada de tiempos extendidos de cruce semafórico [5][12].

3. Metodología

El sistema inteligente propuesto se estructuró en tres módulos independientes y complementarios: el módulo de ejecución, encargado del entrenamiento e inferencia del modelo; el módulo de control, responsable de la gestión del hardware embebido; y el módulo de interfaz, que integra la visualización en tiempo real del sistema. Esta separación modular facilita el mantenimiento, la escalabilidad y la reproducibilidad del prototipo. La arquitectura general del sistema se ilustra en la Figura 1.

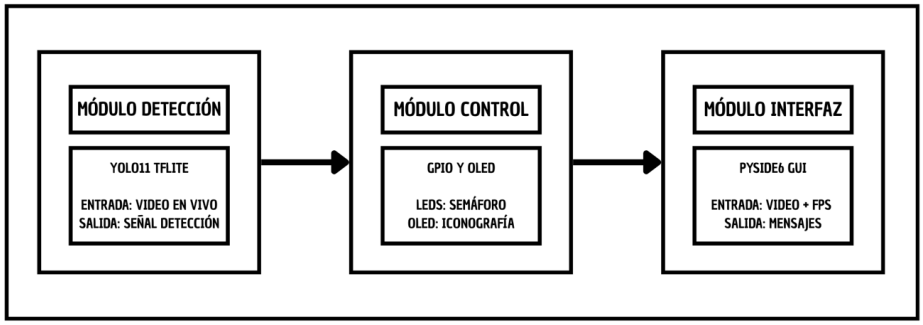

Fig. 1. Diagrama a bloques de los módulos del sistema.

3.1. Módulo de detección

Este módulo constituye el núcleo computacional del sistema, abarcando el entrenamiento del modelo de detección y su posterior optimización para su despliegue en hardware embebido. El entrenamiento se realizó en la plataforma Google Colab, aprovechando el acceso a unidades de procesamiento gráfico (GPU) de manera gratuita a través de los servidores de Google. El hardware GPU que se utilizó fue un Tesla T4 dentro del entorno de ejecución Python 3 con una VRAM de 15GB.

Conjunto de datos Se construyó un conjunto de datos compuesto por 1,100 imágenes de personas en silla de ruedas, recolectadas desde Open Images Dataset v7 de Google [3]. Las imágenes se dividieron en dos subconjuntos: 1,000 para entrenamiento y 100 para validación. Se opto por el 10% de imágenes de validación para maximizar los datos de entrenamiento dada la especificidad de la clase única, asegurando que el modelo viera la mayor cantidad de variaciones de sillas de ruedas posibles en el entorno de aprendizaje. Todas procesadas a una resolución de 320×320 píxeles para equilibrar precisión y velocidad de inferencia en la GPU virtual disponible.

Cada imagen fue etiquetada mediante la herramienta MakeSense AI [8], generando por cada muestra un archivo `.txt` compatible con el formato YOLO. Dicho archivo contiene cinco valores: el identificador de clase (*wheelchair*, asignado con valor 0), las coordenadas normalizadas del centro del *bounding box* en los ejes X e Y, y las dimensiones normalizadas del ancho y alto del *bounding box*, todas en un rango de 0 a 1. La representación visual de este formato se aprecia en las Figuras 2 y 3, donde se aprecia las dimensiones en coordenadas X y Y, alto y ancho del etiquetado de la imagen y además la clase a la que representa. Para este ejemplo se tiene a clase persona y clase corbata. En las coordenadas de formato YOLO se observa el valor que se le asigno a la respectiva clase, es decir, para persona se le asigno el valor 0 y para corbata el valor 27.

Fig. 2. Representación de *bounding box* con distintas clases [12].

zidane.txt

```
0  0.481719 0.634028 0.690625 0.713278
0  0.741094 0.524306 0.314750 0.933389
27 0.364844 0.795833 0.078125 0.400000
```

Fig. 3. Coordenadas en formato YOLO con respecto al *bounding box* [12].

Configuración del entrenamiento YOLO11 cuenta con cinco variantes de modelo según su capacidad computacional: *nano* (n), *small* (s), *medium* (m), *large* (l) y *extra-large* (x) [12]. Para este trabajo se seleccionó la variante *nano* (YOLO11n), justificada por su reducido tamaño en memoria, su alta velocidad de inferencia y su compatibilidad con los recursos limitados de la placa Raspberry Pi 5.

El entrenamiento se configuró mediante la librería Ultralytics con el siguiente comando:

```
!yolo detect train data=data.yaml model=yolo11n.pt \
epochs=150 imgsz=320 patience=25 \
mosaic=0 plots=True
```

Los hiperparámetros se seleccionaron con base en los requisitos del sistema y se justifican de la siguiente manera:

- `epochs=150`: número de ciclos completos de entrenamiento sobre el conjunto de datos.
- `imgsz=320`: resolución de entrada que equilibra precisión y velocidad de inferencia.
- `patience=25`: detención anticipada del entrenamiento si las métricas de validación no mejoran en 25 épocas consecutivas, previniendo sobreajuste.

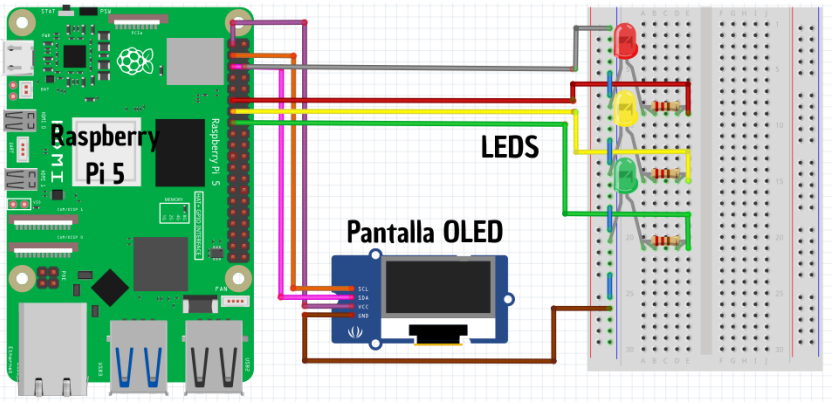

Fig. 4. Diagrama simulado de la configuración de LEDs y pantalla OLED.

- `mosaic=0`: aumentación desactivada para preservar la integridad espacial de cada imagen, evitando distorsiones artificiales que podrían degradar la precisión en la clase única entrenada.

Exportación del modelo Al concluir el entrenamiento, se obtuvo el modelo resultante en formato PyTorch (`best.pt`). Para su despliegue en la placa Raspberry Pi 5, el modelo fue exportado al formato TensorFlow Lite mediante el siguiente comando:

```
!yolo export model=best.pt format=tflite
```

Se seleccionó la variante `float32` (`best_float32.tflite`) por ser la que menor pérdida de precisión sufre durante el proceso de cuantificación, reduciendo el tamaño del modelo y optimizando la latencia sin impactar negativamente el rendimiento del sistema embebido.

3.2. Módulo de control

Este módulo comprende la arquitectura hardware del sistema embebido, diseñada para simular el comportamiento de un semáforo urbano inteligente. Se empleó una placa Raspberry Pi 5 conectada a tres LEDs (rojo, amarillo y verde) mediante una protoboard y cables jumpers, así como una pantalla OLED que muestra la señalización representativa del usuario en silla de ruedas y el tiempo exclusivo de espera asignado. El diagrama de la configuración física se ilustra en la Figura 4.

La lógica de control se implementó en Python, actuando como puente entre la salida de inferencia del modelo YOLO11 y los componentes físicos, a través de un entorno virual dentro del sistema operativo de la placa Raspberry Pi 5. Al momento en que el modelo detecta un usuario en silla de ruedas con una confianza superior al umbral configurado, el sistema activa automáticamente la señalización de cruce extendido.

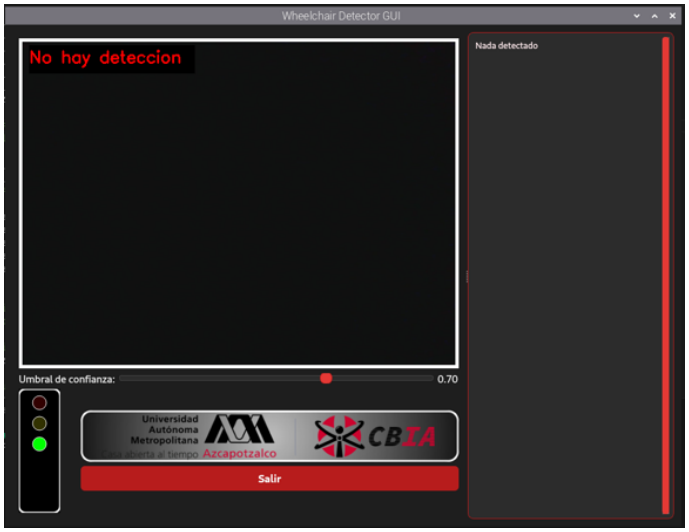

Fig. 5. Interfaz gráfica del sistema.

3.3. Módulo de interfaz

El módulo de interfaz integra todos los componentes del sistema en una aplicación de escritorio desarrollada con la librería PySide6, un framework multiplataforma de software libre para la construcción de interfaces gráficas (GUI) modernas e interactivas. La interfaz cumple las siguientes funciones:

- Visualización del video en tiempo real con indicador de fotogramas por segundo (FPS).
- Registro histórico de detecciones, con marcas de tiempo de inicio y fin de cada evento.
- Semáforo virtual sincronizado con el estado físico de los LEDs, para verificación inmediata del comportamiento del sistema.
- Barra de calibración del umbral de confianza, ajustable entre el 70 % y el 90 %, para adaptar el sistema a distintas condiciones ambientales como variaciones de iluminación, clima adverso o posibles oclusiones parciales.

La interfaz gráfica del sistema se aprecia en la Figura 5.

4. Resultados

En esta sección se presentan los resultados obtenidos en dos etapas: la evaluación cuantitativa del modelo durante el entrenamiento en Google Colab y la validación experimental del sistema en condiciones reales controladas.

Métrica / Condición Validación en Colab (YOLO11n)	
Precisión (P)	0.985
Recall (R)	1.000
mAP50	0.995
mAP50-95	0.829

Table 1. Resultados del entrenamiento YOLO11 en Google Colab

Cuadrante	Valor	Interpretación
Verdaderos positivos	452	Silla de ruedas detectada correctamente
Falsos positivos	32	Fondo detectado como silla de ruedas
Falsos negativos	1	Silla de ruedas presente no detectada
Verdaderos negativos	0	No aplicable (clase única)

Table 2. Matriz de confusión del modelo YOLO11 entrenado

4.1. Evaluación del entrenamiento

Al concluir el proceso de entrenamiento, el modelo generó automáticamente un conjunto de gráficas de desempeño que permiten analizar la evolución de las métricas a lo largo de las épocas. En la Figura 8 se presentan las curvas de precisión obtenidas como resultado final del entrenamiento, recall, mAP50 y mAP50-95 obtenidas.

Los valores finales de las métricas de evaluación se resumen en la Tabla 1.

La métrica de mayor relevancia para evaluar el desempeño global del modelo es mAP50-95, dado que considera múltiples umbrales de intersección sobre la unión (IoU), desde el 50 % hasta el 95 %, ofreciendo una medida integral de la precisión en distintos niveles de dificultad. El modelo alcanzó un valor de 82.9 % en esta métrica, lo que indica un desempeño robusto para la detección de una clase única bajo condiciones controladas [12].

Para evaluar cuantitativamente la capacidad de discriminación del modelo, se analizó la matriz de confusión generada sobre 485 instancias de predicción, cuyos resultados se ilustran en la Figura 9 y se detallan en la Tabla 2.

Al analizar los 32 falsos positivos registrados en la matriz de confusión, se observa que el modelo mantiene una alta fidelidad visual respecto a las etiquetas originales, véase la Figura 10. Los errores de precisión se atribuyen a una ligera sobre-sensibilidad del modelo en entornos con alta densidad de texturas o estructuras metálicas que comparten patrones geométricos con la estructura de las sillas de ruedas. Sin embargo, como se aprecia en la Figura 11, el modelo logra una superposición casi exacta de los bounding boxes respecto a la verdad de campo, lo que garantiza la fiabilidad del sistema en la activación del semáforo.

4.2. Validación en entorno real

Con el propósito de evaluar el desempeño del sistema fuera del entorno de laboratorio, se llevó a cabo una prueba experimental en las instalaciones de la

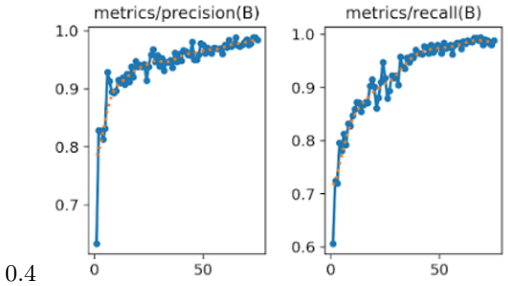

Fig. 6. Métricas de precisión y recall.

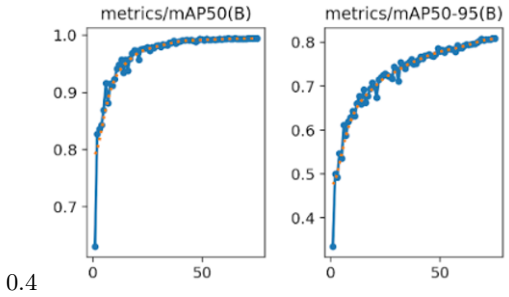

Fig. 7. mAP50 y maAP50-95.

Fig. 8. Graficas generadas al finalizar el entrenamiento del modelo YOLO.

Universidad Autónoma Metropolitana, Unidad Azcapotzalco. Se utilizó una silla de ruedas real y uno de los autores actuó como usuario simulado de movilidad reducida. la cámara se situó a una altura de 1.5 metros respecto al suelo, con un ángulo de inclinación de 10 grados, simulando la posición típica de una unidad de control en un semáforo peatonal.

Las pruebas se realizaron bajo dos configuraciones de umbral de confianza: 80 % y 90 %, con el objetivo de evaluar el alcance de detección efectiva del modelo en condiciones reales. Los resultados obtenidos se describen a continuación:

- **Umbral del 80 %**: la detección se extendió hasta una distancia de 4.5 metros, véase la Figura 12, lo que representa una distancia operativamente más relevante para entornos de semaforización real, aunque con una ligera reducción en la especificidad de las detecciones.
- **Umbral del 90 %**: el modelo logró detectar correctamente al usuario a una distancia aproximada de 2.5 metros, véase la Figura 13. Si bien esta distancia resulta limitada para una aplicación directa en cruces semafóricos urbanos, confirma que el sistema opera con alta precisión en condiciones favorables de iluminación y visibilidad.

Las distancias de 4.5 y 2.5 metros fueron seleccionas como casos de estudio, porque para el modelo estas distancias representaban su límite de detección,

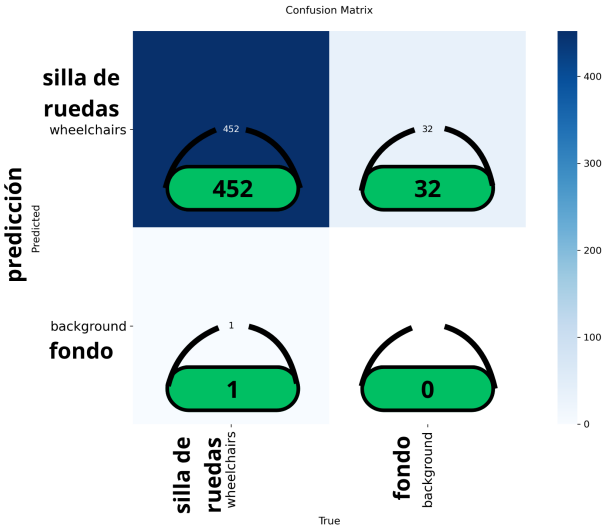

Fig. 9. Matriz de confusión del modelo YOLO11n entrenado.

pues al superarlas con sus respectivos umbrales, comenzaban los falsos positivos o falsos negativos.

Adicionalmente, el sistema demostró un tiempo de respuesta inferior a 2 segundos desde la detección del usuario hasta la activación de la señalización de cruce extendido, realizando el proceso de inmediato, superando el objetivo inicial establecido para el prototipo.

5. Conclusiones y trabajo futuro

El presente trabajo demostró la viabilidad técnica de implementar un sistema inteligente de detección en tiempo real para usuarios en silla de ruedas mediante visión por computadora y hardware embebido de bajo costo. La integración del modelo YOLO11n, optimizado al formato TensorFlow Lite sobre una placa Raspberry Pi 5, permitió alcanzar una precisión de 82.9 % en la métrica mAP50-95 bajo condiciones controladas, con detección efectiva a distancias de hasta 4.5 metros y un tiempo de respuesta inferior a 2 segundos desde la detección del usuario hasta la activación de la señalización de cruce extendido.

Si bien la detección a un umbral de confianza del 90 % se limitó a una distancia de 2.5 metros, los resultados obtenidos con un umbral del 80 % evidencian que el sistema opera de forma estable en condiciones reales, considerando las restricciones inherentes al hardware embebido, las variaciones de iluminación y la calidad de la imagen capturada. La combinación hardware-software propuesta constituye una base sólida y replicable para el desarrollo de soluciones de accesibilidad tecnológica en infraestructura vial urbana.

Fig. 10. Imágenes de validación originales.

Fig. 11. Predicción de las imágenes de validación durante el entrenamiento.

Como trabajo futuro se identifican las siguientes líneas de investigación y desarrollo:

- **Ampliación y diversificación del conjunto de datos**: incorporar imágenes capturadas en condiciones de baja iluminación, clima adverso y entornos urbanos de alta densidad, con el fin de mejorar la robustez del modelo ante escenarios no controlados.
- **Evaluación de variantes superiores de YOLO11**: explorar las versiones *small* y *medium* del modelo sobre hardware con mayor capacidad de cómputo, como la Raspberry Pi 5 con acelerador Hailo-8, para incrementar el alcance de detección sin comprometer la latencia del sistema.
- **Integración con infraestructura semafórica real**: validar el prototipo en cruces peatonales reales en coordinación con autoridades de movilidad

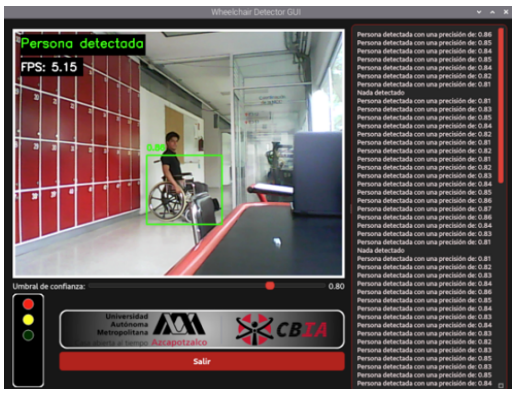

Fig. 12. Detección de silla de ruedas con umbral del 80 % a 4.5 metros de distancia.

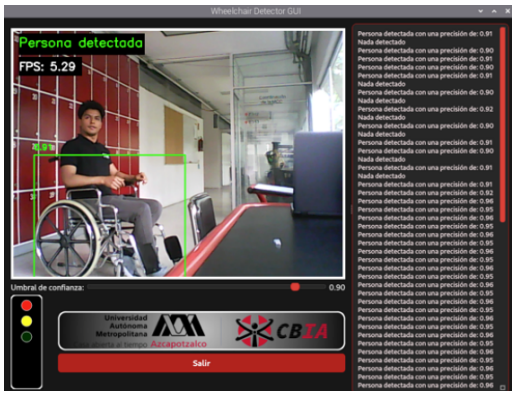

Fig. 13. Detección de silla de ruedas con umbral del 90 % a 2.5 metros de distancia.

urbana, evaluando su desempeño en condiciones de tráfico vehicular activo.
- **Detección multiclase**: extender el sistema para identificar otros perfiles de movilidad reducida, como personas con muletas, andadores o adultos mayores, ampliando el impacto social de la solución.

References

1. Ahmed, M., Israa, J., Tawfeq, P., Ahmen: Real time pedestrian and objects detection using enhanced yolo integrated with learning complexity-aware cascades. TELKOMNIKA Telecommunication Computing Electronics and Control pp. 364–369 (2024)
2. Ainandafiq, F., Muhammad, R., Corina, F.: Attention module in yolo-based object detection method for autonomous smart wheelchair room navigation system. International Conference on Robotics, Automation, and Artificial Intelligence pp. 313–316 (2024)

3. Google: Open images dataset v7 (2026), https://storage.googleapis.com/openimages/web/index.html, accedido: 25-03-2026
4. Hsu, W.Y., Lin, W.Y.: Adaptive fusion of multi-scale yolo for pedestrian detection. Open Access Journal pp. 3–10 (2021)
5. Khanam, R., Hussain, M.: An overview of the key architectural enhancements. Arxiv pp. 3–4 (2024)
6. Lee, K., Na, H., Kwak: A study on traffic vulnerable detection using object detection-based ensemble and yolov5. Journal of The Korea Society of Computer and Information pp. 62–67 (2024)
7. Louis, R., Nicolas, B., Romain, N., Ertau: Deep learning-based object detection, localisation and tracking for smart wheelchair healthcare mobility. International Journal of Environmental Research and Public Health pp. 5–13 (2021)
8. MakeSense: Makesense ai (2026), https://www.makesense.ai/, accedido: 25-03-2026
9. Marco, C., Diego, J., Bryan, B.D., Juan, D., María, J.: Traffic sign detection and recognition using yolo object detection algorithm: A systematic review. Mathematics pp. 5–24 (2024)
10. Shisei, M., Miwa, Y., Yuichi, H.: Detecting falls and slips of wheelchair users using low-resolution thermal image analysis. IEEE Systems, Man and Cybernetics Society Section pp. 136883–136891 (2025)
11. Tomás, Agustín, S., Pierpaolo: Urban pedestrian routes' accessibility assessment using geographic information system processing and deep learning-based object. Sensors pp. 6–16 (2024)
12. Ultralytics: Yolo11 overview (2024), https://docs.ultralytics.com/es/models/yolo11/#overview, accedido: 25-03-2026

Análisis comparativo de redes de Kolmogorov-Arnold y perceptrones multicapa para la representación de funciones especiales

Luis Enrique Ramon-Pedrero, José Adán Hernández-Nolasco, Pablo Pancardo

Universidad Juárez Autónoma de Tabasco, México

{242H21005, Adan.hernandez, pablo.pancardo}@alumno.ujat.mx

Resumen. Este artículo presenta un análisis comparativo entre las redes neuronales de Kolmogorov-Arnold (KAN) y el perceptrón multicapa (MLP) para tareas de representación de funciones continuas unidimensionales. El objetivo fue evaluar ambas arquitecturas bajo condiciones de entrenamiento homogéneas, considerando no solo la precisión predictiva, sino también la eficiencia computacional durante la inferencia. Para ello, se construyó un conjunto de datos sintéticos a partir de cinco funciones objetivo: *bessel*, *bessel_esferica*, *mathieu_je*, *sin_20x* y *sin_30x_exp*. Ambos modelos se entrenaron utilizando el mismo pipeline de aprendizaje supervisado y se compararon mediante las métricas MSE_{test}, R^2_{test} y *flops_forward*. Los resultados muestran que KAN logró consistentemente un menor error y un mejor ajuste que MLP en todas las funciones evaluadas. Además, KAN requirió un menor costo de inferencia, con 704 FLOPs frente a 832 FLOPs para el modelo MLP. En general, la evidencia experimental sugiere que KAN ofrece una relación precisión-costo más favorable que MLP en problemas de representación funcional. Sin embargo, la generalización de estos hallazgos debe validarse con mayor profundidad en otras categorías de funciones y configuraciones experimentales.

Palabras clave: Redes neuronales, redes de Kolmogorov-Arnold, perceptrón multicapa, aproximación de funciones, eficiencia computacional.

Comparative Analysis of Kolmogorov-Arnold Networks and Multilayer Perceptrons for Representing Special Functions

Abstract. This paper presents a comparative analysis between Kolmogorov-Arnold Networks (KAN) and Multi-Layer Perceptron (MLP) neural networks for one-dimensional continuous function representation tasks. The objective was to evaluate both architectures under homogeneous training conditions, considering not only predictive

accuracy but also computational efficiency during inference. To this end, a synthetic dataset was constructed from five target functions: *bessel*, *bessel_esferica*, *mathieu_je*, *sin_20x*, and *sin_30x_exp*. Both models were trained using the same supervised learning pipeline and compared through the metrics MSE_{test}, R^2_{test}, and *flops_forward*. The results show that KAN consistently achieved lower error and better fit than MLP across all evaluated functions. In addition, KAN required a lower inference cost, with 704 FLOPs versus 832 FLOPs for the MLP model. Overall, the experimental evidence suggests that KAN provides a more favorable accuracy-cost tradeoff than MLP in functional representation problems. However, the generalization of these findings should be further validated on additional function categories and experimental configurations.

Keywords: Neural networks, Kolmogorov-Arnold networks, multi-layer perceptron, function approximation, computational efficiency.

1. Introducción

Las redes Perceptrón Multicapa (MLP) son redes neuronales de alimentación hacia adelante compuestas por múltiples capas de neuronas [4,10]. Se utilizan ampliamente para modelar relaciones no lineales complejas mediante entrenamiento basado en descenso del gradiente y retropropagación, y su capacidad de aproximación universal ha sustentado su uso en clasificación, regresión y representación funcional [3].

Por otra parte, el teorema de representación de Kolmogorov-Arnold establece que una función continua multivariable puede expresarse como superposición finita de funciones continuas univariadas y sumas [6,2]. A partir de esta base, Liu et al. propusieron las redes de Kolmogorov-Arnold (KAN), donde las funciones univariadas asociadas a las conexiones son aprendibles, a diferencia de las MLP, que emplean activaciones fijas y ajustan principalmente pesos y sesgos [9].

Estudios recientes han mostrado resultados mixtos al comparar KAN y MLP, por lo que resulta pertinente realizar evaluaciones controladas en tareas específicas de representación funcional [9,15,11]. En este trabajo se compara el desempeño de una KAN respecto a una MLP en términos de exactitud y eficiencia computacional al representar funciones especiales y funciones elementales de alta frecuencia.

El estudio se plantea como una evaluación controlada inicial, limitada a funciones continuas unidimensionales. Esta delimitación permite aislar el efecto de la arquitectura bajo condiciones homogéneas de entrenamiento, aunque no pretende generalizar los resultados a problemas multivariables, de mayor dimensionalidad o escenarios aplicados, los cuales se consideran líneas futuras de validación.

2. Trabajo relacionado

Las redes KAN se inspiran en el teorema de representación de Kolmogorov-Arnold y difieren de las MLP en que emplean funciones aprendibles en las aristas, mientras que las MLP utilizan activaciones fijas en los nodos. Liu et al. presentan su arquitectura, fundamento matemático, leyes de escala y métodos de simplificación, mostrando que KAN puede superar a MLP en diversas tareas [9].

Diversos trabajos han extendido o evaluado KAN en distintos escenarios. En detección de intrusiones para entornos IoT, Abd Elaziz et al. reemplazan capas MLP por capas basadas en KAN, mejorando el desempeño del sistema [1]. En ecuaciones diferenciales y aprendizaje de operadores, Shukla et al. comparan KAN y MLP, reportando que algunas variantes de KAN pueden superar a MLP, aunque también pueden presentar inestabilidad en funciones polinomiales de orden elevado [11].

También se han propuesto aplicaciones de KAN en arquitecturas híbridas y datos reales. Li et al. integran una capa KAN en U-Net para segmentación y generación de imágenes médicas, obteniendo mejoras frente a variantes previas [7]. Asimismo, Vaca-Rubio et al. aplican KAN al pronóstico de tráfico satelital, reportando mayor precisión y eficiencia con menos parámetros que MLP [14].

Estos antecedentes muestran el interés reciente por KAN y motivan comparaciones controladas frente a MLP en tareas específicas de aproximación funcional, como las funciones especiales y señales de alta frecuencia consideradas en este trabajo.

3. Metodología

La metodología empleada en este trabajo se estructura en siete etapas.

3.1. Definición del problema de representación funcional

Se delimitó el problema como una tarea de representación de funciones continuas en una variable, con el propósito de comparar dos familias de modelos neuronales: KAN y MLP. La formulación metodológica se orientó a evaluar, bajo condiciones homogéneas, la capacidad de cada modelo para reproducir funciones objetivo mediante entrenamiento supervisado y validación en datos no vistos.

Desde el punto de vista matemático, una MLP aproxima una función mediante composiciones de transformaciones afines y activaciones no lineales fijas:

$$f_{\mathrm{MLP}}(x) = W_L \sigma \left(W_{L-1} \sigma \left(\cdots \sigma(W_1 x + b_1) \right) \right) + b_L. \tag{1}$$

En contraste, una KAN reemplaza los pesos escalares por funciones univariadas aprendibles en las aristas, de modo que una capa puede expresarse como:

$$x_j^{(l+1)} = \sum_i \phi_{ij}^{(l)} \left(x_i^{(l)} \right), \tag{2}$$

donde $\phi_{ij}^{(l)}$ representa una función aprendible, usualmente parametrizada mediante splines. Esta diferencia estructural permite que KAN ajuste localmente la forma de la función objetivo, mientras que MLP depende de activaciones fijas combinadas con pesos entrenables.

3.2. Construcción del conjunto de funciones y datos de evaluación

Se seleccionó un conjunto de funciones representativas para el estudio: `bessel`, `bessel_esferica`, `mathieu_je`, `sin_20x` y `sin_30x_exp`. Para cada función se generaron datos sintéticos en el dominio definido, con muestreo uniforme, partición entrenamiento/prueba y preprocesamiento mediante normalización. Esta etapa aseguró consistencia entre experimentos y comparabilidad entre modelos.

El propósito de utilizar estas funciones es evaluar la capacidad expresiva de ambas arquitecturas frente a datos de alta complejidad matemática, determinando cuál red tiene una mayor eficiencia y exactitud al modelar este tipo de datos.

3.3. Implementación y configuración de los modelos KAN y MLP

Se implementaron ambos modelos dentro de un mismo pipeline de entrenamiento y evaluación, manteniendo una configuración estable para todos los experimentos. Se fijaron las arquitecturas base y los hiperparámetros principales de optimización, con control de semilla para reproducibilidad. Esta decisión metodológica permitió que las diferencias observadas respondieran al modelo y no a variaciones del procedimiento.

3.4. Ejecución del entrenamiento y evaluación comparativa

Para cada función objetivo se entrenó una KAN y una MLP bajo el mismo esquema supervisado. Las entradas se normalizaron a $[-1, 1]$ y, cuando fue necesario, también se normalizó la salida para estabilizar el entrenamiento. Las predicciones se desnormalizaron antes de calcular las métricas en la escala original.

Durante el entrenamiento se registró la pérdida por época y se generaron gráficas de valor real contra predicción e histogramas de residuos. Estas evidencias permitieron evaluar la convergencia, la forma del error y la capacidad de cada modelo para reproducir la función objetivo.

Las mayores dificultades aparecieron en funciones de alta frecuencia, especialmente en `sin_30x_exp`, donde la combinación de oscilaciones rápidas y modulación exponencial incrementó la complejidad de aproximación.

3.5. Análisis de la relación precisión–costo computacional

La comparación final se planteó bajo un enfoque multicriterio que considera precisión predictiva, costo estimado de inferencia y tiempo de entrenamiento. Se

analizaron conjuntamente `mse_test`, `r2_test`, `flops_forward` y t_{train}, con el fin de identificar el equilibrio entre calidad de representación, costo de inferencia y costo práctico de entrenamiento.

3.6. Integración y documentación de los resultados para su interpretación

Los resultados se consolidaron en un esquema de reporte unificado (tablas y figuras) que facilita la trazabilidad de cada experimento. Esta integración permitió estructurar de forma clara la discusión y las conclusiones, además de dejar una base metodológica reproducible para futuras extensiones del estudio.

3.7. Alcance experimental y limitaciones

La evaluación se restringió a funciones continuas unidimensionales para comparar KAN y MLP bajo condiciones homogéneas de dominio, partición de datos, función de pérdida y entrenamiento. Por ello, los resultados no deben interpretarse como una validación general en problemas multivariables, de mayor dimensionalidad o escenarios aplicados con ruido, restricciones físicas o datos reales. Asimismo, aunque se reporta `flops_forward` como métrica controlada de costo de inferencia, futuras evaluaciones deberán complementar este criterio con tiempo de inferencia, consumo de memoria y estabilidad numérica en hardware específico.

4. Datos y configuración experimental

4.1. Datos y partición

Los datos se generaron sintéticamente evaluando las funciones objetivo del estudio (`bessel`, `bessel_esferica`, `mathieu_je`, `sin_20x` y `sin_30x_exp`) sobre el dominio $[0, 20]$. Para cada función se muestrearon $n\_\text{samples} = 1000$ puntos mediante muestreo uniforme. La partición se realizó de forma aleatoria con `train_split`=0.8 para entrenamiento y el resto para prueba, utilizando `seed`=42 para asegurar reproducibilidad.

Las entradas se normalizaron a $[-1, 1]$. La salida se normalizó únicamente cuando su magnitud superó el umbral definido (`threshold`=50), conservando los parámetros de normalización para desescalar las predicciones durante la evaluación y reportar métricas en la escala original.

4.2. Modelos y entrenamiento

Se compararon dos arquitecturas compactas para mantener un presupuesto computacional comparable: KAN con `width=[1,16,1]`, `grid=19`, `k=3`; y MLP con capas `[1,16,16,1]` y activación `tanh`. Ambos modelos se entrenaron minimizando `MSELoss` con `Adam`, `epochs`=1200, `learning_rate`=0.001 y `batch_size=full`.

La configuración base busca equilibrar capacidad de aproximación y costo de inferencia: MLP usa dos capas ocultas de 16 neuronas, mientras que KAN usa una capa oculta de ancho 16 con funciones por arista parametrizadas mediante splines cúbicos. Dado que no se realizó una búsqueda exhaustiva, el análisis de sensibilidad sobre hiperparámetros clave se plantea como trabajo futuro.

4.3. Evaluación y criterio de costo

La evaluación reportó métricas en la escala original: `mse_test` y `r2_test`. Además, se calculó `flops_forward` como estimación del costo computacional de una pasada hacia adelante y se midió el tiempo total de entrenamiento, denotado como t_{train}, en segundos. Esta métrica complementa el análisis de FLOPs con una medida práctica del costo observado durante el entrenamiento.

Las métricas se definieron formalmente como:

$$MSE_{test} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3}$$

$$R^2_{test} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{4}$$

$$\text{FLOPs}_{\text{forward}} = \sum_{l=1}^{L}\text{FLOPs}_l, \tag{5}$$

donde y_i es el valor real, $\hat{y}_i$ la predicción, $\bar{y}$ el promedio de los valores reales, n el número de muestras de prueba y FLOPs_l el costo estimado de la capa l. Adicionalmente, t_{train} corresponde al tiempo total, en segundos, requerido para completar las épocas de entrenamiento de cada modelo bajo el mismo entorno de ejecución.

La interpretación de los resultados combinó métricas cuantitativas y evidencia gráfica. Un menor `mse_test` indica menor error promedio, mientras que un `r2_test` cercano a 1 refleja mayor capacidad para explicar la variabilidad de la función objetivo. Valores negativos de `r2_test` indican un desempeño inferior al de una predicción basada en el promedio. Además, los histogramas de residuos permiten observar si los errores se concentran cerca de cero, lo cual evidencia un ajuste más estable.

5. Resultados

5.1. Comparación KAN vs MLP bajo presupuesto computacional

Se compararon KAN y MLP usando métricas de precisión en prueba, costo estimado de inferencia y tiempo de entrenamiento. La comparación consideró `mse_test`, `r2_test`, `flops_forward` y t_{train}. La Tabla 1 resume los resultados para `bessel`, `bessel_esferica`, `mathieu_je`, `sin_20x` y `sin_30x_exp`.

Tabla 1. Desempeño en prueba, costo estimado de inferencia y tiempo de entrenamiento para MLP y KAN.

Función	**MLP**				**KAN**			
	MSE_{test}	R^2_{test}	FLOPs	t_{train} (s)	MSE_{test}	R^2_{test}	FLOPs	t_{train} (s)
bessel	8.92e-03	-0.0405	832	10.33	4.93e-06	0.9994	704	50.27
bessel_esferica	2.26e-03	0.9700	832	12.80	3.48e-07	0.9999	704	55.61
mathieu_je	2.99e-06	0.9885	832	10.09	5.12e-09	0.9999	704	52.75
sin_20x	5.25e-01	-0.0075	832	10.31	2.82e-03	0.9945	704	50.07
sin_30x_exp	1.19e-01	-0.0611	832	10.03	6.30e-02	0.4367	704	50.48

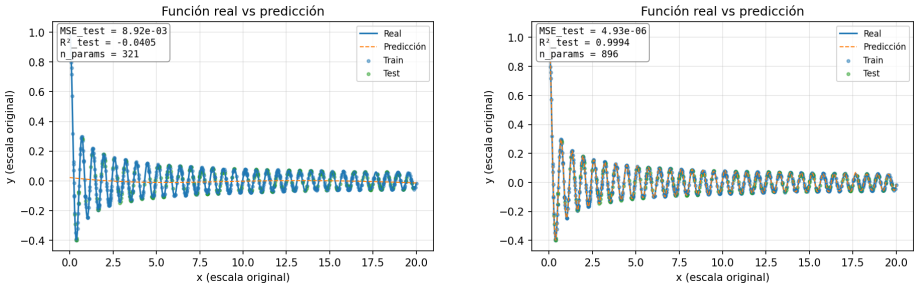

Fig. 1. Ajuste real vs. predicción para `bessel`. Izquierda: MLP. Derecha: KAN.

Los resultados muestran que KAN mejora consistentemente a MLP en `mse_test` y `r2_test`, alcanzando valores de R^2_{test} cercanos a 1 en `bessel`, `bessel_esferica`, `mathieu_je` y `sin_20x`. Además, KAN reduce el costo de inferencia estimado de 832 a 704 FLOPs. Sin embargo, su tiempo de entrenamiento es mayor: aproximadamente 50–56 s frente a 10–13 s de MLP. Por tanto, KAN presenta ventaja en precisión e inferencia, mientras que MLP resulta más eficiente durante el entrenamiento.

5.2. Resultados por función

bessel Para la función `bessel`, KAN supera claramente a MLP en generalización con menor costo estimado de inferencia. En test, MLP obtiene $MSE_{\text{test}} = 8{,}92 \times 10^{-3}$ y $R^2_{\text{test}} = -0{,}040517$ con 832 FLOPs, mientras que KAN alcanza $MSE_{\text{test}} = 4{,}93 \times 10^{-6}$ y $R^2_{\text{test}} = 0{,}999425$ con 704 FLOPs. Como se observa en la Fig. 1, KAN sigue con mayor fidelidad la relación real–predicción. Además, la Fig. 2 muestra una mayor concentración de residuos alrededor de cero para KAN.

bessel_esferica En `bessel_esferica`, KAN vuelve a presentar mejor rendimiento que MLP con menor costo de inferencia. MLP reporta $MSE_{\text{test}} = 2{,}26 \times 10^{-3}$ y $R^2_{\text{test}} = 0{,}970043$ (832 FLOPs), mientras que KAN obtiene $MSE_{\text{test}} = 3{,}48 \times 10^{-7}$ y $R^2_{\text{test}} = 0{,}999995$ (704 FLOPs). La Fig. 3 confirma

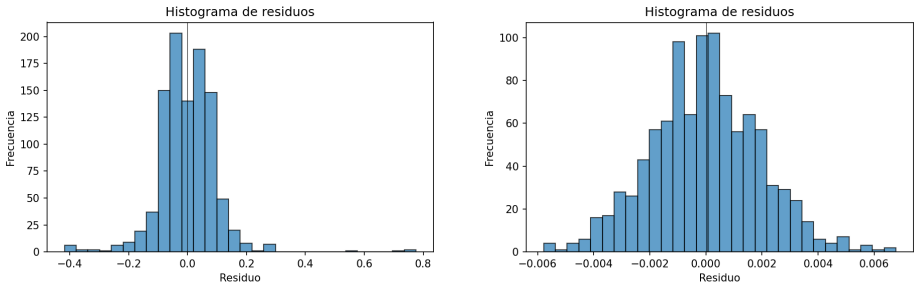

Fig. 2. Histograma de residuos para `bessel`. Izquierda: MLP. Derecha: KAN.

el mejor ajuste de KAN. De forma consistente, la Fig. 4 evidencia residuos de menor magnitud para KAN.

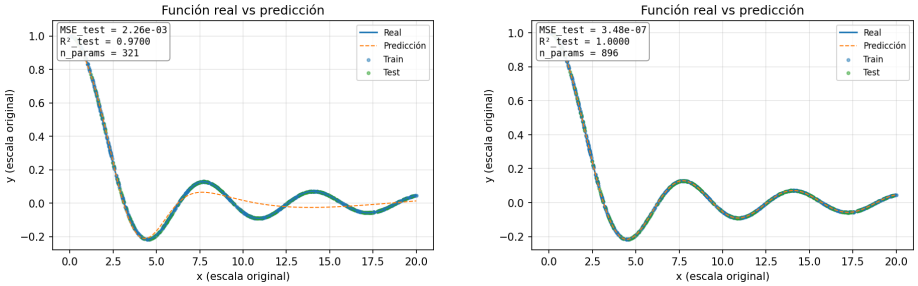

Fig. 3. Ajuste real vs. predicción para `bessel_esferica`. Izquierda: MLP. Derecha: KAN.

mathieu_je Para `mathieu_je`, KAN mantiene una ventaja clara frente a MLP en el conjunto de prueba. MLP alcanza $MSE_{\text{test}} = 2{,}99 \times 10^{-6}$ y $R^2_{\text{test}} = 0{,}988568$ (832 FLOPs), mientras que KAN logra $MSE_{\text{test}} = 5{,}12 \times 10^{-9}$ y $R^2_{\text{test}} = 0{,}999980$ (704 FLOPs). La Fig. 5 muestra la mayor fidelidad del ajuste con KAN. Asimismo, la Fig. 6 confirma una distribución de errores más concentrada cerca de cero.

sin_20x En `sin_20x`, la diferencia entre arquitecturas es especialmente marcada. MLP presenta $MSE_{\text{test}} = 5{,}25 \times 10^{-1}$ y $R^2_{\text{test}} = -0{,}007591$ con 832 FLOPs, mientras que KAN obtiene $MSE_{\text{test}} = 2{,}82 \times 10^{-3}$ y $R^2_{\text{test}} = 0{,}994583$ con 704 FLOPs. La Fig. 7 evidencia que KAN captura mejor el comportamiento global de la señal. En la Fig. 8 se observa también una reducción clara de residuos para KAN.

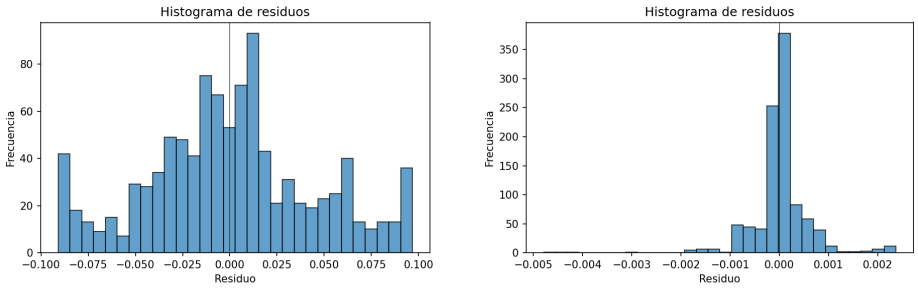

Fig. 4. Histograma de residuos para `bessel_esferica`. Izquierda: MLP. Derecha: KAN.

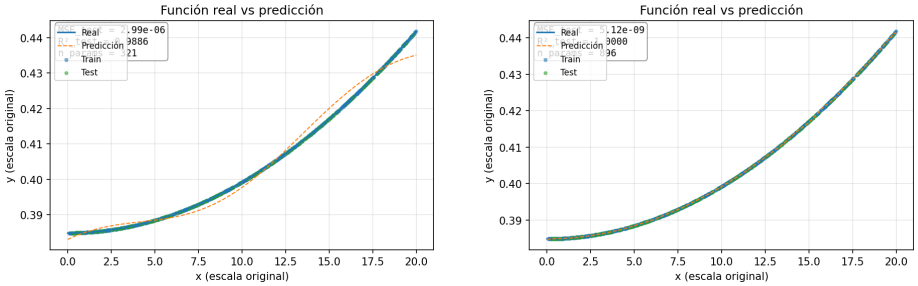

Fig. 5. Ajuste real vs. predicción para `mathieu_je`. Izquierda: MLP. Derecha: KAN.

`sin_30x_exp` En `sin_30x_exp`, KAN también mejora frente a MLP, aunque con una brecha menor que en `bessel` o `mathieu_je`. En test, MLP obtiene $MSE_{\text{test}} = 1{,}19 \times 10^{-1}$ y $R^2_{\text{test}} = -0{,}061152$ (832 FLOPs), mientras que KAN alcanza $MSE_{\text{test}} = 6{,}30 \times 10^{-2}$ y $R^2_{\text{test}} = 0{,}436762$ (704 FLOPs). La comparación de la Fig. 9 sugiere una mejora del ajuste con KAN, aunque aún con desviaciones apreciables. Esto es consistente con la Fig. 10, donde KAN reduce la dispersión residual respecto a MLP, pero sin la concentración observada en los casos de mejor desempeño.

6. Discusión

KAN mostró una mejora consistente frente a MLP, al reducir `mse_test` y aumentar `r2_test` en todas las funciones evaluadas. Las mayores ventajas se observaron en `bessel`, `bessel_esferica`, `mathieu_je` y `sin_20x`; en cambio, `sin_30x_exp` presentó una mejora más moderada, asociada a su mayor complejidad oscilatoria.

En términos computacionales, KAN obtuvo menor costo estimado de inferencia, con 704 FLOPs frente a 832 FLOPs de MLP. Sin embargo, MLP requirió

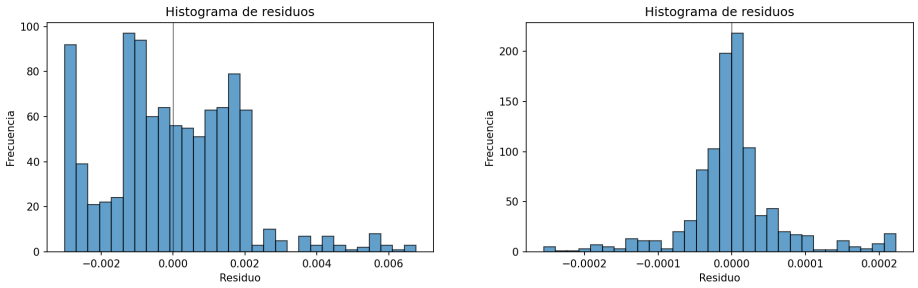

Fig. 6. Histograma de residuos para `mathieu_je`. Izquierda: MLP. Derecha: KAN.

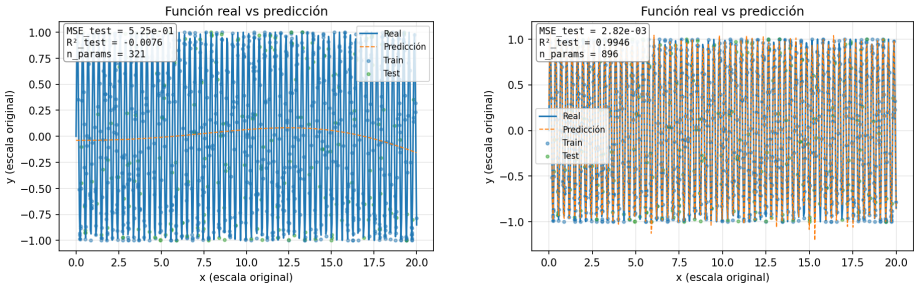

Fig. 7. Ajuste real vs. predicción para `sin_20x`. Izquierda: MLP. Derecha: KAN.

menor tiempo de entrenamiento, por lo que la ventaja de KAN se concentra principalmente en precisión e inferencia, no en costo computacional global.

Este comportamiento puede explicarse por las funciones univariadas aprendibles de KAN, que permiten ajustar variaciones locales de funciones estructuradas u oscilatorias con mayor flexibilidad que las activaciones fijas de MLP. No obstante, los resultados deben interpretarse dentro del alcance definido, ya que el estudio se limita a funciones unidimensionales, una configuración experimental fija y métricas parciales de eficiencia.

El caso `sin_30x_exp` sugiere que funciones con alta frecuencia y modulación de amplitud pueden requerir mayor capacidad, una malla más fina o estrategias adaptativas de muestreo para mejorar la aproximación.

7. Conclusiones y trabajos futuros

En este trabajo se compararon MLP y KAN en tareas de representación funcional, considerando precisión, costo estimado de inferencia mediante `flops_forward` y tiempo de entrenamiento t_{train}. Los resultados muestran que KAN obtuvo menor `mse_test`, mayor R^2_{test} y menor número de FLOPs que MLP en las funciones evaluadas.

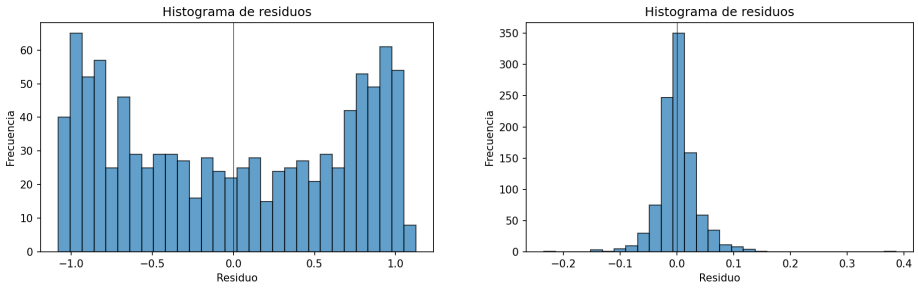

Fig. 8. Histograma de residuos para `sin_20x`. Izquierda: MLP. Derecha: KAN.

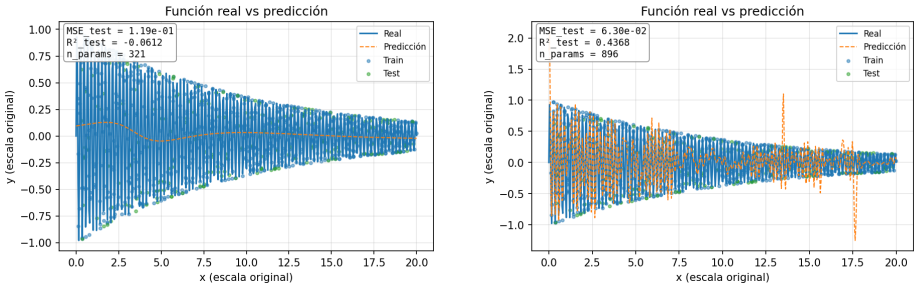

Fig. 9. Ajuste real vs. predicción para `sin_30x_exp`. Izquierda: MLP. Derecha:KAN.

Sin embargo, KAN requirió mayor tiempo de entrenamiento, por lo que su ventaja se concentra en calidad de aproximación e inferencia, no en eficiencia computacional global. Así, KAN presenta una relación precisión–inferencia favorable, mientras que MLP conserva ventaja en entrenamiento.

Como trabajo futuro, se propone validar estos hallazgos en funciones multivariables, problemas de mayor dimensionalidad y configuraciones adicionales. También se plantea ampliar el análisis de sensibilidad, incorporar tiempo de inferencia, memoria y estabilidad numérica, e incluir arquitecturas modernas de aproximación funcional, como Fourier Features [13], SIREN [12], redes residuales [5] y operadores neuronales [8], con el fin de contextualizar la ventaja observada de KAN frente a modelos especializados.

Referencias

1. Abd Elaziz, M., Ahmed Fares, I., Aseeri, A.O.: CKAN: Convolutional kolmogorov-arnold networks model for intrusion detection in IoT environment. IEEE Access 12, 134837–134851 (2024)
2. Arnold, V.I.: On functions of three variables. Doklady Akademii Nauk SSSR 114(4), 679–681 (1957)

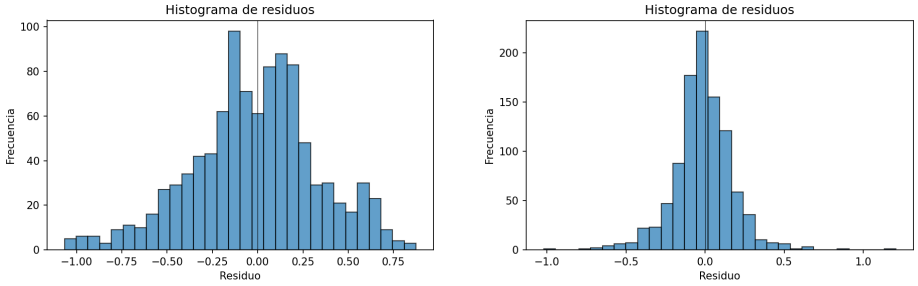

Fig. 10. Histograma de residuos para `sin_30x_exp`. Izquierda: MLP. Derecha: KAN.

3. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems 2(4), 303–314 (1989)
4. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA (2016)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
6. Kolmogorov, A.N.: On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In: Doklady Akademii Nauk. vol. 114, pp. 953–956. Russian Academy of Sciences (1957)
7. Li, C., Liu, X., Li, W., Wang, C., Liu, H., Liu, Y., Chen, Z., Yuan, Y.: U-KAN makes strong backbone for medical image segmentation and generation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 4652–4660 (2025)
8. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Fourier neural operator for parametric partial differential equations. In: International Conference on Learning Representations (2021)
9. Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., Tegmark, M.: Kan: Kolmogorov-Arnold networks. arXiv preprint arXiv:2404.19756 (2024)
10. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature 323(6088), 533–536 (1986)
11. Shukla, K., Toscano, J.D., Wang, Z., Zou, Z., Karniadakis, G.E.: A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks. Computer Methods in Applied Mechanics and Engineering 431, 117290 (2024)
12. Sitzmann, V., Martel, J.N.P., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Advances in Neural Information Processing Systems. vol. 33, pp. 7462–7473 (2020)
13. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In: Advances in Neural Information Processing Systems. vol. 33, pp. 7537–7547 (2020)
14. Vaca-Rubio, C.J., Blanco, L., Pereira, R., Caus, M.: Kolmogorov-Arnold networks (Kans) for time series analysis. arXiv preprint arXiv:2405.08790 (2024)

15. Yu, R., Yu, W., Wang, X.: KAN or MLP: A fairer comparison. arXiv preprint arXiv:2407.16674 (2024)

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

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación