EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 155 No. 6
June 2026

Research in Computing Science

Research in Computing Science, Año 25, Volumen 155, No. 6, junio 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 junio de 2026.

Research in Computing Science, year 25, Volume 155, No. 6, June 2026, is published monthly by the Center for Computing Research of IPN.

Volume 155(6)

Advances in Artificial Intelligence

Hiram Ponce (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

Predicción de series de tiempo en sistemas dinámicos caóticos mediante redes generativas adversarias

Jesús Alejandro Salazar-González[1,2], Erik Molino-Minero-Re[1,2]

[1] Posgrado en Ciencia e Ingeniería de la Computación, UNAM, Mérida, Yucatán, México

[2] Unidad Académica del IIMAS, UNAM, en Yucatán, Mérida, Yucatán, México

alejandro.salazar9812@comunidad.unam.mx

Resumen. La predicción de series de tiempo en sistemas dinámicos caóticos representa un reto debido a su alta sensibilidad a las condiciones iniciales y a las no linealidades que caracterizan su comportamiento. En este trabajo se propone evaluar el modelo *GAN-LSTM-Attention*, desarrollado para predecir índices financieros,con el objetivo de analizar su capacidad predictiva bajo condiciones altamente no lineales. La evaluación se realiza sobre un conjunto de datos sintéticos generados a partir de sistemas caóticos clásicos. Se analizan distintas configuraciones de tamaño de ventana de entrada y salida, así como esquemas de predicción autorregresivos y no autorregresivos. El enfoque no autorregresivo muestra un buen desempeño, alcanzando valores de coeficiente de determinación (R^2) y correlación de Pearson (r) superiores a 0.80 y 0.90, respectivamente. A pesar de eso, se encuentra que existe un deterioro significativo en la calidad de la predicción al incrementar la longitud de la ventana de predicción, lo cual se debe a la acumulación del error. Asimismo, se observó que un incremento del tamaño de la memoria no produce mejoras significativas en el desempeño.

Keywords: Redes neuronales, redes generativas adversarias, predicción, sistemas dinámicos, series de tiempo.

Time Series Prediction in Chaotic Dynamical Systems Using Adversarial Generative Networks

Abstract. Predicting time series in chaotic dynamical systems presents a challenge due to their high sensitivity to initial conditions and the nonlinearities that characterize their behavior. This paper proposes to evaluate the GAN-LSTM-Attention model, developed to predict financial indices, with the aim of analyzing its predictive capacity under highly

nonlinear conditions. The evaluation is performed on a set of synthetic data generated from classical chaotic systems. Different configurations of input and output window sizes are analyzed, as well as autoregressive and non-autoregressive prediction schemes. The non-autoregressive approach shows good performance, achieving coefficient of determination (R^2) and Pearson correlation (r) values greater than 0.80 and 0.90, respectively. Despite this, a significant deterioration in prediction quality is found when the prediction window length is increased, due to error accumulation. Furthermore, it was observed that increasing the memory size does not produce significant performance improvements.

Keywords: Neural networks, generative adversarial networks, prediction, dynamical systems, time series.

1. Introducción

La predicción de series de tiempo es un proceso importante para la toma de decisiones en un gran número de aplicaciones, que pueden ir desde la identificación temprana de una afección cardíaca, el pronóstico del clima e inclusive la detección de anomalías de maquinaria industrial. La manera tradicional de trabajar modelos en sistemas dinámicos consiste en estudiar el comportamiento físico del problema y en plantear ecuaciones diferenciales que lo describan. Sin embargo, en la práctica, la cantidad de variables que describan la realidad a menudo es desconocida, o ni siquiera podemos medirlas todas. También las condiciones de frontera introducen no linealidades que pueden elevar significativamente la dificultad de encontrar una solución analítica. Por ello, técnicas de aprendizaje automático (como lo son las redes neuronales) en los últimos años han tenido gran impacto en este campo debido a que no es necesario plantear un modelo analítico explícito y depende únicamente de los datos que tengamos del sistema a analizar.

Entre las variedades de técnicas del aprendizaje automático se encuentran las redes neuronales, que tienen la capacidad de reproducir características no lineales presentes en nuestro proceso de estudio. En el campo de los sistemas dinámicos, han demostrado ser eficaces para realizar predicciones de sistemas caóticos, tales como el sistema de Lorenz y el registro histórico del nivel del agua del Gran Lago Salado en Utah, Estados Unidos , ambos estudiados en [2]. En ese trabajo se utiliza una versión modificada del exponente de Lyapunov para comparar sus modelos, lo que permite estimar un horizonte de predicción en sistemas dinámicos caóticos.

En el campo del aprendizaje automático, las redes generativas están diseñadas principalmente para crear datos realistas a partir de muestras representativas. A partir de esta perspectiva, el presente artículo explora el potencial de las redes generativas adversarias para la predicción de series de tiempo mediante la evaluación de una arquitectura *GAN-LSTM-Attention* aplicada a sistemas dinámicos caóticos. Dicha arquitectura, previamente utilizada en la predicción de datos financieros, integra módulos LSTM (*Long Short-Term Memory*) y

mecanismos de atención para capturar dependencias temporales relevantes en la señal. Se analizan distintos tamaños de ventana de entrada y salida (denominados tamaño de memoria y horizonte de predicción, respectivamente), además de esquemas de predicción autorregresivos y no autorregresivos.

2. Trabajo relacionado

Los sistemas caóticos representan un gran desafío en la predicción debido a la alta sensibilidad a las condiciones iniciales. Los métodos tradicionales de predicción incluyen sistemas determinísticos, que buscan una aproximación a la solución de ecuaciones que modelan la dinámica sistema. Un ejemplo de uso de aprendizaje automático en este ámbito es GenCast, empleada para la predicción del clima, un modelo de difusión probabilística que ha logrado superar en el 97.2 % de las métricas evaluadas al sistema de predicción por conjuntos (ENS por sus siglas en inglés) del Centro Europeo de Predicciones Meteorológicas a Plazo Medio [10]. Con ello se destaca que los modelos de aprendizaje automático poseen un gran potencial para la predicción de sistemas dinámicos.

Debido a la naturaleza recurrente de los sistemas dinámicos, las redes LSTM han sido ampliamente utilizadas para modelar dependencias temporales en series de tiempo [12].Las arquitecturas con LSTM se han utilizado en conjunto de transformaciones matemáticas o métodos estadísticos para modelar series de tiempo, pero se ha observado que estos enfoques no siempre se comportan bien con dinámicas altamente sensibles a las condiciones iniciales o caóticas [9].

En lo que respecta al análisis de series de tiempo, usualmente se añaden otros módulos que ayudan a extraer características locales y globales del sistema. Fan H. [4], en su artículo, propone el uso de un reservorio de cómputo que actualice esporádicamente el sistema real; con ello evita que los sistemas sensibles diverjan. Otra manera de extraer características es empleando la técnica de transformar las señales en imágenes, por ejemplo, estas pueden ser generadas por mapas de recurrencias, las cuales en conjunto de alguna red convolucional suelen ser útiles para la predicción [8].

En este contexto, también se han empleado auto-decodificadores variacionales, que transforman los datos a un espacio latente de menor dimensión, usualmente coherente con una distribución gaussiana. Esto permite extraer características relevantes e incluso logra la reducción del ruido. TimeVAE es un ejemplo de este tipo de red [3].

Por otro lado, las redes generativas adversarias (GAN) han cobrado relevancia en los últimos años. Estas emplean dos redes neuronales: la primera es un generador especializado en generar datos sintéticos a partir de ruido aleatorio, y la segunda es un discriminador que identifica si los datos de entrada son reales o sintéticos [1]. Al entrenar ambas redes en conjunto, se obtiene un modelo capaz de generar datos que parecen reales a partir de ruido aleatorio. Se han utilizado en la detección de anomalías en datos de telemetría espacial [11], en donde la idea principal es dejar que las redes aprendan la representación latente de datos normales; después de ello, se utilizará el discriminador para

identificar entre datos anómalos y normales. Asimismo, las GAN se han aplicado en modelos de hornos rotatorios; en estos se combinan varias herramientas que incluyen la reconstrucción de espacios de fases, la generación de mapas globales de recurrencia y el uso de LSTM para la obtención de características [5].

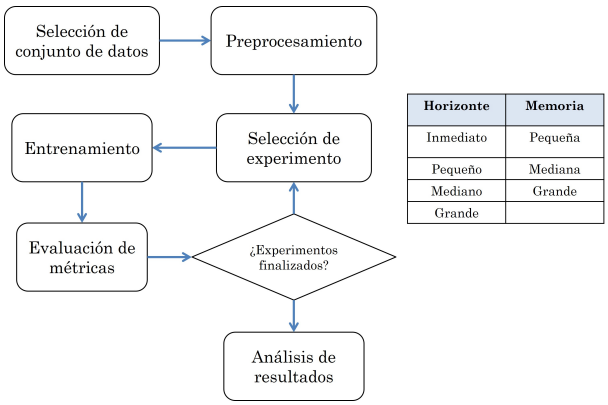
En [7] se discute un modelo denominado GAN-LSTM-Attention, aplicado a la predicción de precios de acciones consideradas representativas del mercado (S&P 500, Apple, AMD y Google). La predicción bursátil se caracteriza por su fuerte dinamismo, no lineal y volátil, lo que la convierte en una tarea compleja. El estudio demuestra que la combinación de mecanismos generativos, redes recurrentes y módulos de atención puede ofrecer mejoras en la precisión y la robustez de la predicción de datos nuevos.

En dicho modelo, el generador emplea una capa LSTM para capturar dependencias a largo plazo, un mecanismo de atención para asignar mayor peso a la información temporal más relevante, y finalmente capas densas para producir la salida. El discriminador utiliza capas convolucionales unidimensionales para extraer patrones locales en la secuencia, seguidas de capas densas que permiten clasificar entre datos reales y generados. Un aspecto importante de este enfoque es que únicamente utiliza datos temporales como entrada; no requiere representaciones visuales como mapas de recurrencia. Por lo tanto, la cantidad de datos necesaria para entrenar esta arquitectura es considerablemente menor que la del modelo GRP-lstmGAN, lo que representa una ventaja.

3. Metodología

Las fases principales de este trabajo se representan en el diagrama esquemático de la Figura 1. La primera fase consiste en seleccionar la base de datos y procesarla. Posteriormente, con el propósito de evaluar distintos horizontes de predicción y tamaños de memoria, se realizan siete experimentos.

El entrenamiento de la red generativa adversaria se basa en la arquitectura propuesta por [7], utilizada para la predicción de los precios de las acciones de la bolsa. Para la evaluación del desempeño del modelo, se emplea un conjunto de 4 métricas, entre las cuales se encuentran el coeficiente de determinación (R^2), el coeficiente de correlación de Pearson (r), el error cuadrático medio ($RMSE$) y la Deformación Temporal Dinámica (dynamic time warping, DTW). Finalmente, se analizan los resultados de los experimentos.



Horizonte	Memoria
Inmediato	Pequeña
Pequeño	Mediana
Mediano	Grande
Grande	

Fig. 1. Flujo de trabajo.

Selección de la base de datos y preprocesamiento. Un sistema dinámico se caracteriza por la evolución de su estado a lo largo del tiempo, de acuerdo con ciertas reglas. Se conoce una gran variedad de sistemas de este tipo, por ejemplo, la oscilación de un péndulo, el perfil de velocidades de un fluido e incluso el famoso sistema de Lorenz, por lo que la selección de un conjunto de datos que sea útil para el entrenamiento de nuestra red y pueda adaptarse fácilmente a sistemas no vistos es de suma importancia.

En el artículo [6] se emplea el modelo *PANDA*, entrenado con una base de datos sintética generada mediante algoritmos evolutivos. El proceso inicia con una población base de sistemas caóticos clásicos (como los de Lorenz, Rössler, Chua, entre otros), cuyos parámetros y condiciones iniciales han sido ajustados para garantizar comportamientos caóticos estables. A partir de esta población, se aplican operadores evolutivos, entre los cuales destacan la mutación, mediante la adición de ruido gaussiano a los parámetros, y la recombinación, en la que dos sistemas se acoplan mediante un producto sesgado aditivo (*skew-product*), produciendo un nuevo sistema híbrido.

Para el entrenamiento, se seleccionan aleatoriamente 100 series de tiempo tridimensionales de 4086 pasos, provenientes del conjunto de datos original de [6]. Dependiendo del experimento, se aplica una ventana deslizante para generar un conjunto de datos con un tamaño de entrada (memoria) y un horizonte de predicción definidos. A cada serie se le aplica un proceso de estandarización con $\mu = 0$ y $\sigma = 1$. Asimismo, se construye el conjunto de datos de prueba a partir de 20 series de tiempo no vistas por los modelos.

Experimentos. Para evaluar los resultados, se realizan experimentos sobre el horizonte de predicción y el tamaño de memoria. En el primero de estos se selecciona una ventana de entrada de 100 pasos y se varía el tamaño de salida entre los descritos en la tabla 1. En el caso del tamaño de memoria, se utilizan

las características descritas en la tabla 2, en la que el horizonte de predicción se fija en 10 muestras a futuro.

Tabla 1. Características de experimentos para predicción. Se fija un tamaño de memoria de 100 pasos y se varía el número de pasos para predecir.

Abreviatura	Horizonte de predicción	Pasos a futuro	Porcentaje de memoria
HI	Inmediato	1	1 %
HS	Pequeño	10	10 %
HM	Mediano	25	25 %
HL	Largo	50	50 %

Tabla 2. Características de los experimentos de tamaño de memoria. Se fija un horizonte de predicción a 10 pasos futuros y se varía el tamaño de memoria.

Abreviatura	Tamaño de memoria	Pasos en el tiempo	Porcentaje de serie
MS	Pequeña	100	2.5 %
MM	Mediana	200	5 %
MG	Grande	400	10 %

Entrenamiento. El entrenamiento se realiza tomando como base el modelo propuesto en [7]. Este cuenta con dos fases alternadas: en la primera se entrena al discriminador y se congela al generador, con el fin de que el discriminador sea capaz de diferenciar datos reales y sintéticos. En la segunda fase, se entrena al generador mientras el discriminador se congela, para que el generador pueda aproximarse cada vez más a la distribución de datos reales. Las funciones de pérdida incluyen la entropía cruzada binaria para el discriminador, mientras que el generador incorpora una pérdida proveniente del error cuadrático medio, con la adición de una pérdida adversarial. La arquitectura del generador está conformada por:

1. **Una capa de entrada lineal**: que toma el tamaño de la memoria seleccionada.
2. **Dos capas LSTM**: la primera con 128 unidades y la segunda con 64, ambas con retorno de secuencias, encargadas de capturar dependencias temporales de distintas profundidades en la serie de entrada.
3. **Mecanismo de atención**: se calcula un puntaje mediante una capa densa y se normaliza con una función softmax para obtener los pesos de atención. Estos pesos se aplican a las salidas de las LSTM y se obtiene un vector de contexto mediante una suma ponderada.

4. **Capas densas**: una capa con 128 neuronas y activación ReLU para transformar el contexto y extraer representaciones no lineales.
5. **Capa de salida**: una capa densa que realiza la predicción; su tamaño depende del horizonte de predicción seleccionado.

Mientras que la arquitectura del discriminador está conformada por:

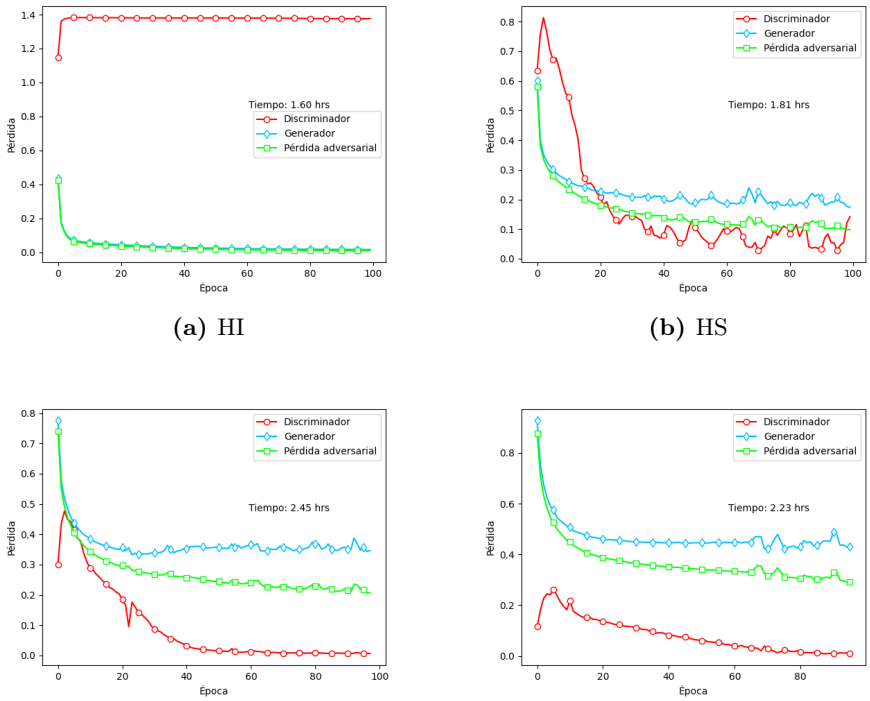
1. **Una capa de entrada lineal**: que toma el tamaño de la memoria seleccionada.
2. **Tres capas convolucionales 1D**: con 64, 128 y 256 filtros respectivamente, todas con activación ReLU, encargadas de extraer patrones locales en las secuencias predichas.
3. **Capa de aplanado**: convierte las características extraídas en un vector unidimensional.
4. **Capas densas**: una con 128 neuronas y otra con 64, ambas con activación ReLU, para reducir la dimensionalidad y aprender representaciones más abstractas.
5. **Capa de salida sigmoide**: produce una probabilidad que indica si la secuencia de entrada corresponde a datos reales o generados.

Evaluación de métricas. Una vez finalizado el entrenamiento, se realizan las predicciones bajo dos esquemas, no autorregresivo y autorregresivo. En el primero de ellos, el modelo recibe únicamente datos reales de entrada, mientras que el segundo se alimenta de sus propias predicciones. Para predecir las series de tiempo se emplea una ventana deslizante, debido a que en la mayoría de los experimentos el horizonte de predicción produce múltiples pasos hacia adelante; algunas predicciones se solapan. El valor de predicción será el promedio de las predicciones solapadas. Una vez generada la serie de predicción, se determinan las métricas R^2, r, $RMSE$ y DTW.

4. Resultados y discusión

Funciones de pérdida. Se grafican los resultados obtenidos durante el entrenamiento de los 7 modelos propuestos. En las figuras 2 y 3 se muestran los resultados de los experimentos de horizonte de predicción y de memoria, respectivamente. En el caso del horizonte de predicción, podemos notar en la figura 2 (b) que, en un principio, el discriminador presenta una pérdida demasiado alta, pero al continuar con el entrenamiento esta decrece, lo que indica que el modelo está aprendiendo a reconocer muestras reales de las falsas. En torno a la época 40 se observa que esta pérdida oscila, lo cual puede interpretarse como que el generador ha llegado a un punto en el que es capaz de engañar al discriminador con mayor facilidad.

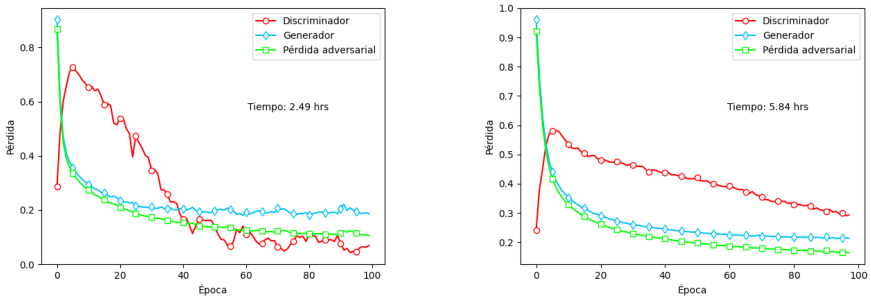
Por otro lado, en las figuras 2 (a) se observa que el discriminador no es capaz de distinguir muestras reales de las falsas, y en 2 (c) y (d) este obtiene una dominancia muy fuerte, lo que provoca que el generador no sea capaz de mejorar a partir de cierto punto.



(a) HI **(b)** HS

(c) HM **(d)** HL

Fig. 2. Curvas de pérdida de horizonte de predicción.



(a) MM **(b)** ML

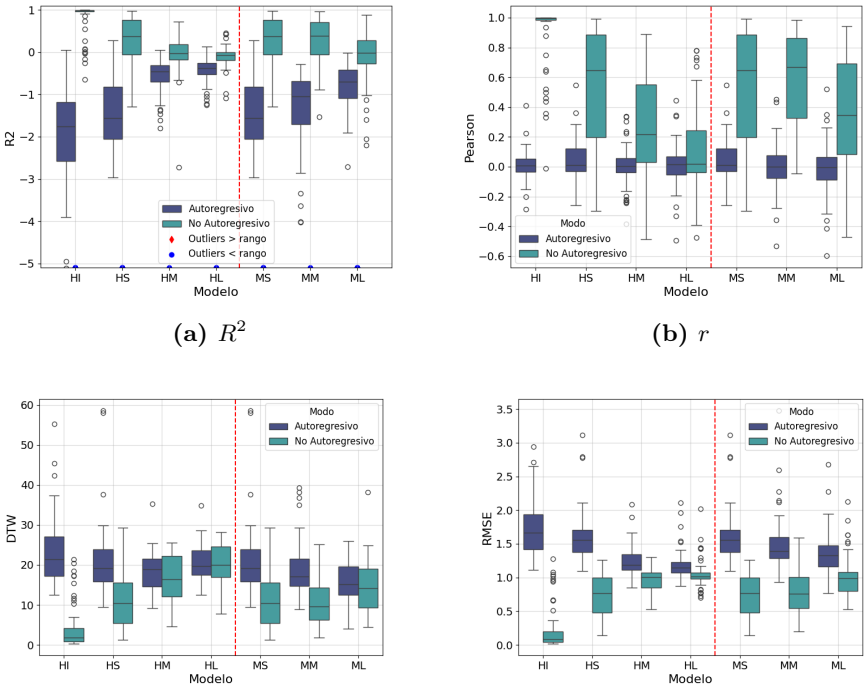
Fig. 3. Curvas de pérdida de tamaño de memoria.

Predicciones. En la figura 4 se encuentran los resultados obtenidos para los esquemas autorregresivos y no autorregresivos respectivamente. En el caso no autorregresivo, R^2 obtiene valores positivos, alcanzando un máximo de 0.816 en el modelo HI, lo que indica que el modelo sí logra capturar la dinámica del sistema. En este mismo caso se observa un valor alto del coeficiente de correlación de Pearson, alcanzando un 0.904, indicando una buena correspondencia entre datos predichos y reales.

En el caso autorregresivo, los valores de R^2 son negativos en la mayoría de los casos, lo que indica que el modelo no logra captar la dinámica del sistema, lo cual es de esperarse en sistemas caóticos, donde cambios pequeños en las condiciones acumulan un error rápidamente, provocando que las predicciones se vayan deteriorando. Asimismo, se observa que el coeficiente de Pearson es cercano a 0, lo que indica una baja correlación entre la predicción y los datos reales. Algo que destaca es que la métrica DTW parece encontrar un mínimo en el modelo HM, lo que sugiere que un horizonte de predicción intermedio logra capturar mejor la dinámica temporal de las series.

En lo que respecta al tamaño de la memoria, en el esquema autorregresivo el tamaño no mejora significativamente los resultados en todas las métricas; por lo tanto, en este caso, la mejor opción es escoger un tamaño de memoria pequeño, pues el tiempo de entrenamiento es menor (1.81 vs. 5.84 hrs).

Los resultados anteriores muestran que el esquema no autorregresivo es claramente superior, especialmente cuando trabajamos con horizontes cortos. El esquema autorregresivo, aunque es interesante para simular a largo plazo, sufre de una acumulación de errores que no logra capturar la dinámica de los sistemas caóticos. Una forma de mejorar estos resultados podría ser emplear una estrategia en la que el sistema reciba una actualización esporádica de los datos, similar a la mencionada por [4] .



(a) R^2 **(b)** r

(c) *Dynamic Time Warping* **(d)** *RMSE*

Fig. 4. Diagramas de caja de las métricas obtenidas en el conjunto de prueba para los modelos entrenados. Se muestran ambos esquemas de predicción. La línea roja punteada separa las pruebas de horizonte de predicción y de memoria. Para fines de visualización se señalan los valores atípicos que están fuera del rango de la imagen.

En la figura 5 se muestra cómo se superponen las gráficas de predicción con la serie real de 3 series provenientes del conjunto de pruebas (se muestra una serie distinta por fila). En el lado izquierdo de la figura se muestra un esquema no autorregresivo y en el derecho el autorregresivo, donde se aprecia con claridad la diferencia en la calidad de predicción.

5. Conclusiones

Este trabajo aborda la predicción de series de tiempo en sistemas dinámicos caóticos mediante el uso de redes generativas adversarias, en particular a través de un modelo basado en *GAN-LSTM-Attention*. El objetivo es evaluar el potencial de las redes generativas para la predicción en sistemas altamente no lineales y sensibles a las condiciones iniciales. Dicha arquitectura ha sido previamente

empleada en la predicción de precios de acciones, caracterizados por su comportamiento volátil. En este trabajo, se explora su uso en sistemas caóticos para estudiar su capacidad de modelar dinámicas complejas en escenarios distintos al financiero.

Para evaluar la arquitectura se recurrió a un conjunto de datos empleado para entrenar un algoritmo denominado *PANDA*, que contiene datos sintéticos generados mediante algoritmos evolutivos, cuya población inicial estuvo compuesta por sistemas caóticos clásicos. Se analizan diversas configuraciones de horizonte y memoria bajo dos esquemas de predicción distintos: no autorregresivo y autorregresivo.

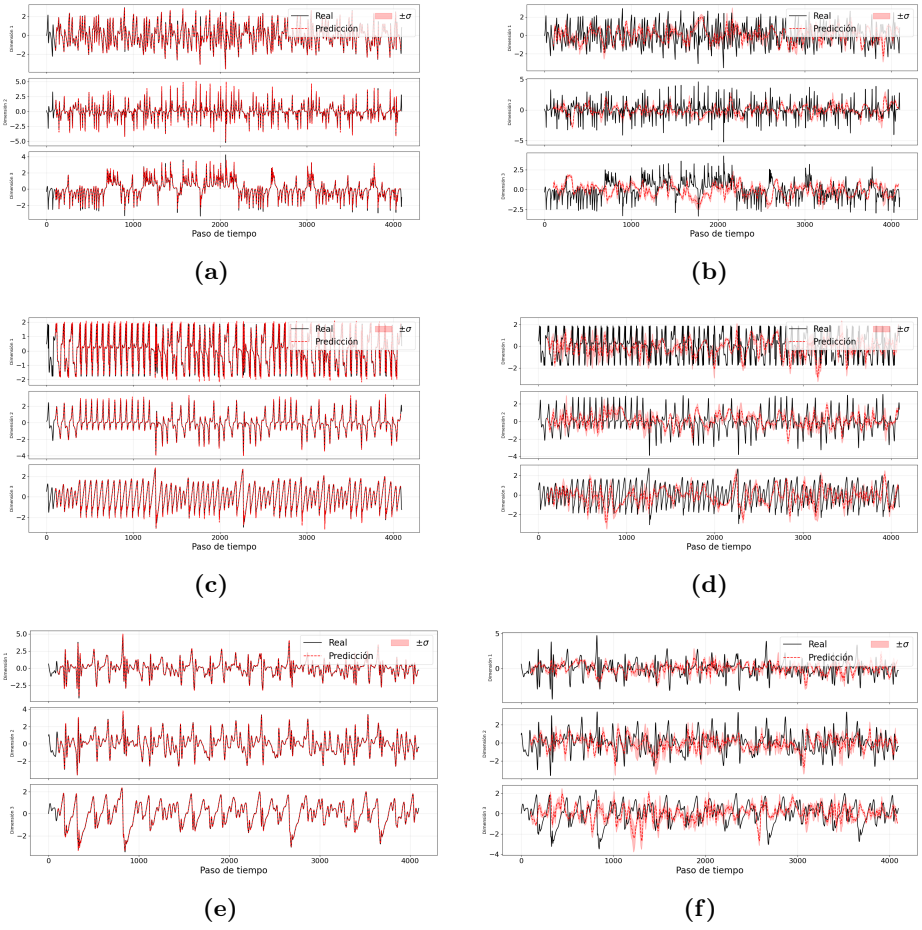


Fig. 5. Comparación entre los ambos esquemas de predicción, paneles izquierdos muestran esquema no autoregresivo mientras que los derechos el autoregresivo. (a), (c) y (e) muestran el modelo HI mientras que (b), (d) y (f) emplean el modelo HM.

Los resultados muestran que, bajo el esquema no autorregresivo, los valores R^2 y r se encuentran por encima de 0.80 y 0.90 respectivamente, indicando buen desempeño predictivo. A pesar de esto, al incrementar el horizonte de predicción, la calidad de los resultados disminuye rápidamente debido a la acumulación de errores, lo que constituye un punto crítico en un sistema sensible a las condiciones iniciales. En lo que respecta al incremento de la memoria, no se observa un aumento significativo en la calidad de la predicción, por lo que se sugiere utilizar configuraciones más pequeñas para reducir el costo computacional. Como trabajo a futuro, se propone explorar enfoques basados en reservorios de cómputo que permitan agregar información de las observaciones en tiempo real, con el objetivo de minimizar la propagación del error y mejorar la precisión de las predicciones a largo plazo.

Agradecimientos. Este proyecto fue apoyado por el programa UNAM-PAEP IG101725. JASG agradece el apoyo económico de SECIHTI.

Referencias

1. Abdel-Basset, M., Moustafa, N., Hawash, H.: Generative adversarial networks (gans). In: Deep Learning Approaches for Security Threats in IoT Environments, pp. 271–285. Wiley (2022)
2. Alfaro, M., Fuertes, G., Vargas, M., Sepúlveda, J., Veloso-Poblete, M.: Forecast of chaotic series in a horizon superior to the inverse of the maximum lyapunov exponent. Complexity 2018, 1452683 (2018)
3. Desai, A., Freeman, C., Wang, Z., Beaver, I.: Timevae: A variational auto-encoder for multivariate time series generation (2021)
4. Fan, H., Jiang, J., Zhang, C., Wang, X., Lai, Y.C.: Long-term prediction of chaotic systems with machine learning. Physical Review Research 2(1), 012080 (2020)
5. Gao, Y., Sosnowski, M., Grabowska, K., Skrobek, D., Moeen Uddin, G., Kulakowska, A., Zylka, A., El Fil, B., Hu, W., Mao, Z.: Forecasting for chaotic time series based on grp-lstmgan model: Application to temperature series of rotary kiln. Entropy 25(1), 52 (2023)
6. Lai, J., Bao, A., Gilpin, W.: Panda: A pretrained forecast model for chaotic dynamics (2025)
7. Li, P., Wei, Y., Yin, L.: Research on stock price prediction method based on the gan-lstm-attention model. Computers, Materials and Continua 82(1), 609–625 (2025)
8. Li, X., Kang, Y., Li, F.: Forecasting with time series imaging. Expert Systems with Applications 160, 113680 (2020)
9. Mahmoudi, A.: Investigating LSTM-based time series prediction using dynamic systems measures. Evolving Systems (2025), https://link.springer.com/article/10.1007/s12530-025-09703-y
10. Price, I., Sanchez-Gonzalez, A., Alet, F., Andersson, T.R., El-Kadi, A., Masters, D., Ewalds, T., Stott, J., Mohamed, S., Battaglia, P., Lam, R., Willson, M.: Probabilistic weather forecasting with machine learning. Nature 637(8044), 84–90 (2024)
11. Song, Y., Yu, J., Tang, D., Han, D., Wang, S.: Telemetry data-based spacecraft anomaly detection using generative adversarial networks. In: ICSMD 2020. pp. 297–301 (2020)

12. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. Neural Computation 31(7), 1235–1270 (2019)

Sistema de moderación automática de contenido basado en PLN para entornos que implementan chats en tiempo real

Jonathan Fernández Córdova

TecNM/Instituto Tecnológico de Veracruz,
Departamento de Sistemas y Computación, Veracruz, Ver.,
México

L25020480@veracruz.tecnm.mx

Resumen. Este trabajo presenta el diseño e implementación de un sistema de moderación automática de contenido basado en técnicas de Procesamiento de Lenguaje Natural (PLN) para entornos de chat en tiempo real. El sistema propuesto tiene como objetivo detectar y clasificar mensajes generados por los usuarios en dos categorías: discurso de odio (hate) y contenido no dañino (no hate), con el fin de mejorar la calidad y seguridad de las interacciones digitales. El modelo fue entrenado utilizando múltiples datasets públicos en español e inglés, lo que permite un enfoque multilingüe y una mayor capacidad de generalización. Se emplean técnicas de aprendizaje automático basadas en modelos de lenguaje preentrenados para la clasificación de texto. Además, el sistema se integra con un bot de moderación capaz de analizar mensajes en tiempo real y aplicar acciones automáticas según la clasificación obtenida. Los resultados experimentales muestran un desempeño sólido en la identificación de contenido dañino, reduciendo la necesidad de moderación manual y permitiendo soluciones escalables para plataformas digitales.

Palabras clave: Procesamiento del lenguaje natural, clasificación de texto, redes neuronales, detección de discurso de odio, moderación de chats.

NLP-based Automated Content Moderation System for Environments Implementing Real-time Chats

Abstract. This work presents the design and implementation of an automatic content moderation system based on Natural Language Processing (NLP) techniques for real-time chat environments. The proposed system aims to detect and classify messages generated by users into two categories: hate speech (hate) and non-harmful content (no hate), in order to improve the quality and security of digital interactions. The

model was trained using multiple public datasets in Spanish and English, allowing for a multilingual approach and greater generalizability. Machine learning techniques based on pre-trained language models are used for text classification. In addition, the system is integrated with a moderation bot capable of analyzing messages in real time and applying automatic actions according to the classification obtained. Experimental results show strong performance in identifying harmful content, reducing the need for manual moderation and enabling scalable solutions for digital platforms.

Keywords: Natural language processing, text classification, neural networks, hate speech detection, chat moderation.

1. Introducción

En los últimos años, especialmente tras la pandemia de COVID-19, las plataformas de transmisión en vivo han experimentado un crecimiento exponencial, consolidándose como uno de los principales medios de interacción digital en tiempo real [1]. Servicios de streaming y redes sociales han permitido a millones de usuarios participar activamente mediante sistemas de chat en vivo, donde se intercambian opiniones, comentarios e ideas de manera instantánea. Este fenómeno ha transformado la forma en que las comunidades digitales se comunican, generando espacios altamente dinámicos y participativos.

Sin embargo, este incremento en la interacción también ha traído consigo importantes desafíos relacionados con la moderación del contenido. La aparición frecuente de mensajes ofensivos o de odio representa un problema significativo, ya que puede afectar la experiencia de los usuarios y deteriorar la calidad de las comunidades en línea. La moderación manual resulta insuficiente ante el volumen y la velocidad de los mensajes generados en chats de alta concurrencia. Se estima que transmisiones con miles de espectadores requieren múltiples moderadores para mantener el control del flujo de mensajes, lo que incrementa los costos operativos y limita la escalabilidad de este enfoque.

En este contexto, el uso de técnicas de Procesamiento de Lenguaje Natural (PLN) ha emergido como una solución prometedora para la detección automática de contenido dañino. Los avances en aprendizaje automático y redes neuronales han permitido desarrollar modelos capaces de comprender y clasificar texto con un alto grado de precisión [2], facilitando la automatización de tareas que anteriormente dependían exclusivamente de la intervención humana.

A pesar de estos avances, la integración de sistemas de moderación automática en entornos de chat en tiempo real continúa representando un desafío, debido a la necesidad de procesar grandes volúmenes de datos de forma eficiente y con baja latencia. Asimismo, es fundamental que estos sistemas no solo detecten contenido inapropiado, sino que también sean capaces de ejecutar acciones correctivas de manera inmediata dentro de la plataforma.

En este trabajo se propone un sistema de moderación automática basado en PLN, diseñado para analizar y clasificar mensajes en tiempo real en una

tarea binaria: discurso de odio (*hate*) y contenido no dañino (*no hate*). Esta formulación reduce la ambigüedad semántica presente en enfoques multiclase y mejora la consistencia del modelo en la detección de contenido dañino. El sistema considera un enfoque bilingüe, permitiendo el análisis de mensajes en español e inglés, lo cual amplía su aplicabilidad en entornos digitales globales. El modelo de clasificación se integra con un bot que permite aplicar acciones automatizadas dentro del chat, como advertencias o sanciones.

Como contribución principal, esta investigación presenta una solución escalable que combina modelos de PLN con mecanismos de acción en tiempo real, permitiendo mejorar la calidad de la interacción en plataformas digitales. El objetivo es reducir la dependencia de la moderación manual y contribuir a la creación de entornos digitales más seguros, eficientes y sostenibles.

El resto de este documento se organiza de la siguiente manera: la Sección 2 presenta el trabajo relacionado con la temática abordada; la Sección 3 describe los materiales y métodos utilizados en el desarrollo del sistema, incluyendo el dataset, el preprocesamiento y el modelo empleado; la Sección 4 expone los resultados obtenidos y su respectivo análisis; finalmente, la Sección 5 presenta las conclusiones del trabajo y posibles líneas de investigación futura.

2. Trabajo relacionado

En los últimos años, la detección automática de contenido tóxico y discurso de odio ha sido ampliamente estudiada dentro del campo del Procesamiento de Lenguaje Natural (PLN), impulsada por el crecimiento de las plataformas digitales y la necesidad de moderar grandes volúmenes de texto en tiempo real. Diversos enfoques han sido propuestos, desde métodos tradicionales de aprendizaje automático hasta modelos basados en arquitecturas profundas.

Uno de los avances más relevantes en esta área es el uso de modelos de lenguaje preentrenados como BERT [2], los cuales han demostrado un alto desempeño en tareas de clasificación de texto al capturar el contexto semántico de las palabras. Posteriormente, modelos más robustos como RoBERTa [10] han optimizado el proceso de preentrenamiento, logrando mejoras significativas en múltiples tareas de PLN, incluyendo la detección de contenido ofensivo.

Asimismo, variantes multilingües como Multilingual BERT han permitido extender estas capacidades a múltiples idiomas, facilitando la detección de contenido dañino en contextos globales.

En cuanto a los datos utilizados para el entrenamiento, iniciativas como el Toxic Comment Classification Challenge de Jigsaw [3] han impulsado el desarrollo de modelos robustos mediante el uso de datasets a gran escala.

Por otro lado, estudios como el de Davidson et al. [11] evidencian la dificultad de distinguir entre lenguaje ofensivo y discurso de odio, señalando que ambos conceptos no son equivalentes y pueden generar ambigüedades en los modelos de clasificación. Esta problemática resalta la importancia de una adecuada definición de las etiquetas en sistemas de moderación automática.

En esta misma línea, Kolhatkar et al. [12] proponen la clasificación de comentarios constructivos, destacando la relevancia de diferenciar entre críticas útiles y contenido dañino. Este enfoque permite enriquecer los sistemas de moderación al considerar no solo la toxicidad, sino también la intención comunicativa del mensaje.

Más recientemente, Mathew et al. [13] introducen HateXplain, un dataset que incorpora anotaciones humanas sobre las razones detrás de una clasificación, promoviendo el desarrollo de modelos explicables. Este tipo de enfoques resulta especialmente relevante en aplicaciones reales, donde la transparencia y la interpretabilidad son fundamentales.

Adicionalmente, modelos basados en redes neuronales recurrentes, como Long Short-Term Memory (LSTM) [4], han sido utilizados para la clasificación de texto, logrando resultados competitivos al capturar dependencias secuenciales en el lenguaje. Sin embargo, estos enfoques presentan limitaciones en términos de eficiencia y escalabilidad frente a modelos basados en transformers.

Finalmente, diversos estudios de revisión [6] han analizado el estado del arte en la detección de discurso de odio, destacando que este problema no depende únicamente de palabras explícitas, sino también del contexto, la intención y factores culturales, lo cual representa un desafío significativo para los sistemas automáticos.

A pesar de estos avances, la mayoría de los trabajos existentes se centran únicamente en la detección de contenido dañino, sin considerar la integración de sistemas capaces de ejecutar acciones automáticas en tiempo real dentro de entornos de chat. En este sentido, el presente trabajo propone un enfoque integral que combina un modelo de clasificación bilingüe basado en PLN con un sistema de moderación automatizado, permitiendo no solo identificar contenido inapropiado, sino también actuar de manera inmediata dentro de la plataforma.

3. Materiales y métodos

En esta sección se describen los materiales y métodos utilizados para el desarrollo del sistema de moderación automática basado en técnicas de Procesamiento de Lenguaje Natural (PLN).

3.1. Dataset

El conjunto de datos fue construido mediante la integración de múltiples datasets públicos ampliamente utilizados en tareas de detección de discurso de odio y análisis de sentimiento, con el objetivo de garantizar diversidad lingüística, robustez y representatividad del problema.

Para el idioma español, se emplearon los datasets Spanish Hate Speech Superset y TASS 2019 [15,16], los cuales contienen ejemplos de lenguaje ofensivo y análisis de sentimiento. En el caso de TASS, las etiquetas originales fueron adaptadas a una formulación binaria (hate / no hate) para mantener consistencia con el enfoque del modelo.

Asimismo, se incorporó el dataset HatEval [17], enfocado en la detección de discurso de odio en redes sociales, particularmente contra grupos vulnerables.

Para el idioma inglés, se utilizó el dataset Hate Speech and Offensive Language [11], así como el dataset Stanford Sentiment Treebank (SST-2) [18], el cual contribuye a mejorar la comprensión contextual del modelo en textos no ofensivos.

Tabla 1. Datasets utilizados

Dataset	Idioma	Tipo
Spanish Hate Speech Superset	Español	Hate Speech
TASS 2019	Español	Sentimiento adaptado
HatEval	Español/Inglés	Hate Speech
Hate Speech Offensive Language	Inglés	Hate / Offensive
SST-2	Inglés	Sentimiento

El dataset final está compuesto por aproximadamente 56,000 instancias textuales, integrando datos en español e inglés. Esta combinación permite entrenar un modelo bilingüe capaz de generalizar en distintos contextos lingüísticos.

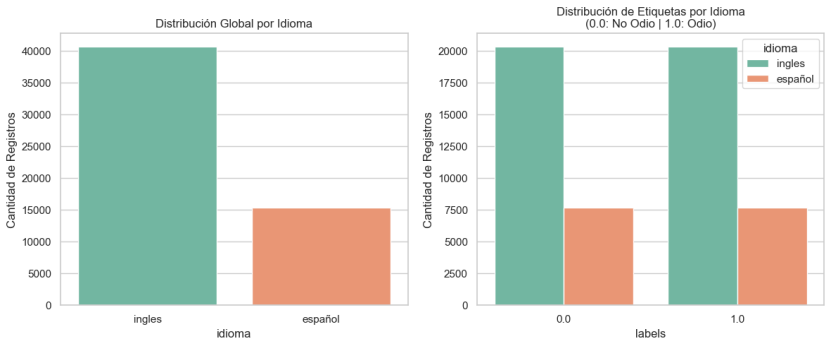


Fig. 1. Distribución del dataset por idioma y etiquetas (0: no hate, 1: hate).

Como se muestra en la Figura 1, se observa una mayor proporción de datos en inglés en comparación con el español. No obstante, la distribución de etiquetas se mantiene balanceada entre las clases de discurso de odio (hate) y contenido no dañino (no hate), lo cual favorece el entrenamiento del modelo.

Asimismo, el análisis de la longitud de los textos revela diferencias entre idiomas. En general, los textos en español presentan una mayor cantidad de palabras y caracteres en comparación con los textos en inglés, lo cual introduce un desafío adicional en términos de generalización del modelo.

El conjunto de datos fue dividido en tres subconjuntos: entrenamiento (70 %), validación (15 %) y prueba (15 %). Esta partición permite entrenar el modelo,

ajustar hiperparámetros y evaluar su desempeño en datos no vistos. La división se realizó de manera estratificada para preservar la distribución de clases en cada subconjunto.

Tabla 2. Distribución por idioma

Idioma	Porcentaje
Inglés	72 %
Español	28 %

3.2. Preprocesamiento

Los datos fueron sometidos a un proceso de limpieza que incluyó la eliminación de caracteres especiales, normalización del texto y tokenización. Posteriormente, los textos fueron convertidos a identificadores numéricos (input IDs) y se generaron máscaras de atención (attention masks).

Como se muestra en la Tabla 3, el proceso de tokenización transforma el texto original en una representación estructurada que puede ser interpretada por el modelo.

Tabla 3. Ejemplo del proceso de tokenización

Etapa	Resultado
Texto original	Ï hate this game"
Tokenización	[Ï", "hate", "this", "game"]
Input IDs	[101, 5223, 2023, 2208, 102]
Attention Mask	[1, 1, 1, 1, 1]

3.3. Modelo

Se utilizó el modelo *bert-base-multilingual*, basado en la arquitectura de transformers [2]. Para su implementación se empleó la biblioteca PyTorch [7], junto con la librería Transformers de HuggingFace [8], la cual facilita la carga y ajuste fino de modelos preentrenados.

Este modelo cuenta con un vocabulario de aproximadamente 119,547 tokens y más de 90 millones de parámetros, lo que le permite procesar múltiples idiomas y capturar relaciones contextuales complejas en el texto.

Durante el entrenamiento, se incorporó una capa de regularización mediante *Dropout*, con el objetivo de reducir el sobreajuste y mejorar la capacidad de generalización del modelo. El modelo fue entrenado utilizando el optimizador AdamW durante 10 épocas, determinando este valor de manera empírica mediante el monitoreo del conjunto de validación.

Desde el punto de vista matemático, BERT recibe como entrada una secuencia de tokens:

$$X = (x_1, x_2, ..., x_n).$$

Cada token es representado mediante la suma de embeddings de token, posición y segmento:

$$E_i = E_i^{token} + E_i^{position} + E_i^{segment}.$$

Como se muestra en la Figura 2, la representación de entrada integra estas tres componentes, permitiendo al modelo capturar información contextual tanto a nivel léxico como estructural [2].

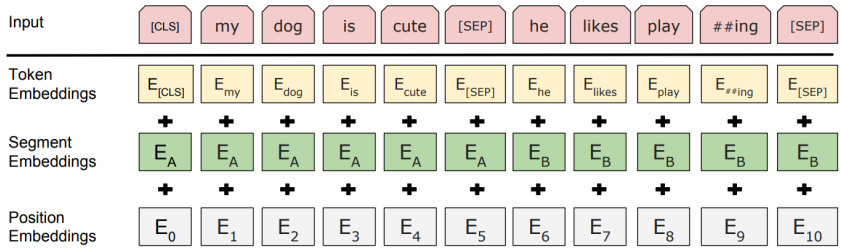


Fig. 2. Representación de entrada en BERT. Los embeddings de entrada se obtienen como la suma de los embeddings de token, posición y segmento. Adaptado de [2].

Estas representaciones se procesan a través de capas de autoatención, definidas como [14]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$

donde Q, K y V representan las matrices de consultas, claves y valores, respectivamente.

El modelo utiliza atención multi-cabeza para capturar diferentes relaciones semánticas [14]:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W^O.$$

donde cada cabeza se define como:

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V).$$

Para la tarea de clasificación binaria (*hate / no hate*), se utiliza la representación del token especial $[CLS]$, siguiendo el esquema de clasificación propuesto en BERT [2]:

$$\hat{y} = \text{softmax}(Wh_{[CLS]} + b),$$

donde $h_{[CLS]}$ corresponde a la representación final del token $[CLS]$, y W, b son los parámetros entrenables de la capa de salida.

3.4. Configuración de entrenamiento

El modelo utilizado fue *bert-base-multilingual-cased*, basado en la arquitectura Transformer encoder y compuesto por 12 capas, 12 cabezas de atención y una dimensión oculta de 768 características. Se realizó un proceso de *fine-tuning* completo, sin congelamiento de capas, permitiendo la actualización de todos los parámetros del modelo durante el entrenamiento.

La representación contextual obtenida del token especial $[CLS]$ fue utilizada como entrada de una capa lineal de clasificación binaria. Antes de la capa de salida, se aplicó regularización mediante *Dropout* con probabilidad $p = 0{,}4$, con el objetivo de reducir el sobreajuste durante el ajuste fino.

La tokenización se realizó mediante *BertTokenizer*, utilizando truncamiento y *padding* hasta una longitud máxima de secuencia de 150 tokens. Cada entrada fue transformada en *input IDs* y *attention masks*, preservando el formato requerido por el modelo BERT.

El conjunto de datos fue dividido de manera estratificada en entrenamiento (70 %), validación (15 %) y prueba (15 %), manteniendo la distribución original de las clases en cada subconjunto.

El entrenamiento se ejecutó utilizando un tamaño de lote (*batch size*) de 32 muestras durante un máximo de 7 épocas. La función de pérdida utilizada fue *CrossEntropyLoss*, adecuada para tareas de clasificación multiclase y binaria basadas en logits.

Para la optimización de parámetros se utilizó el algoritmo AdamW [2], configurado con una tasa de aprendizaje inicial de 2×10^{-5} y un término de regularización L_2 (*weight decay*) de 0.01. El scheduler empleado fue *linear learning rate decay* sin fase de warmup, calculado en función del número total de pasos de entrenamiento:

$$\text{total\_steps} = |\mathcal{D}_{train}| \times \text{epochs},$$

donde $|\mathcal{D}_{train}|$ representa el número de lotes del conjunto de entrenamiento.

Durante la retropropagación se aplicó *gradient clipping* con norma máxima igual a 1.0:

$$\|\nabla\theta\|_2 \leq 1{,}0$$

con el objetivo de estabilizar el entrenamiento y prevenir explosión de gradientes.

Asimismo, se implementó entrenamiento en precisión mixta (*Automatic Mixed Precision, AMP*) mediante `torch.amp.autocast` y `GradScaler`, reduciendo el consumo de memoria GPU y acelerando el tiempo de entrenamiento.

El proceso de entrenamiento incorporó una estrategia de *Early Stopping* basada en la pérdida de validación. El entrenamiento finalizaba automáticamente cuando no se observaban mejoras durante 2 épocas consecutivas (*patience* = 2). Además, se almacenó el estado correspondiente al menor valor de pérdida de validación para restaurar posteriormente el mejor modelo obtenido.

Finalmente, el desempeño del modelo fue evaluado mediante las métricas *accuracy* y *macro F1-score*, calculadas sobre el conjunto de prueba, complementadas con matriz de confusión y reporte de clasificación para analizar el comportamiento del modelo en ambas clases.

3.5. Sistema de Implementación

El sistema fue desarrollado en Python 3.10.20, utilizando un enfoque modular orientado a la integración de modelos de Procesamiento de Lenguaje Natural (PLN) en entornos de moderación automática en tiempo real.

La inferencia del modelo se implementó mediante FastAPI, proporcionando una arquitectura basada en servicios REST para el procesamiento de mensajes en tiempo real. La API recibe mensajes de texto, ejecuta el proceso de tokenización utilizando *BertTokenizer* y posteriormente realiza la inferencia utilizando el modelo *bert-base-multilingual-cased* ajustado mediante fine-tuning.

La integración con plataformas de streaming se realizó mediante TwitchIO, una biblioteca asincrónica para la interacción con el protocolo IRC de Twitch. El bot desarrollado monitorea continuamente los mensajes publicados en el chat y envía cada instancia textual al sistema de inferencia para su clasificación automática.

El flujo operacional del sistema se compone de las etapas presentadas en la Fig. 3.

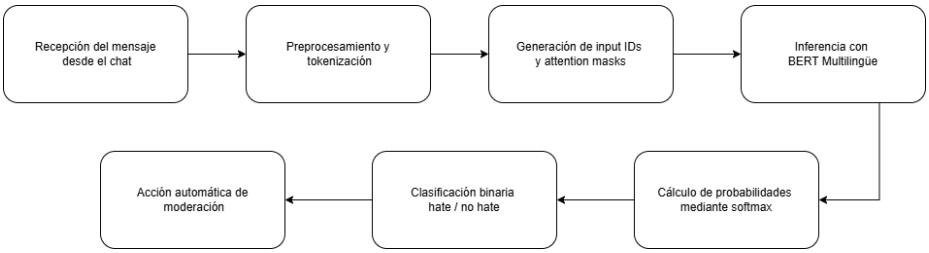


Fig. 3. Flujo operacional del sistema de moderación automática basado en BERT multilingüe.

Cuando un mensaje es clasificado como contenido dañino, el sistema ejecuta automáticamente acciones de moderación sobre el chat, incluyendo la eliminación del mensaje y la generación de advertencias dirigidas al usuario involucrado. En caso contrario, el mensaje permanece visible sin intervención adicional.

La arquitectura implementada permite desacoplar el módulo de inferencia del cliente de moderación, facilitando la escalabilidad del sistema y la posible integración con otras plataformas de comunicación en tiempo real.

Adicionalmente, durante el entrenamiento e inferencia se incorporaron técnicas de optimización computacional como *Automatic Mixed Precision* (AMP), procesamiento por lotes y pre-tokenización de datos, reduciendo el tiempo de ejecución y el consumo de memoria GPU.

4. Resultados y discusión

En esta sección se presentan los resultados experimentales obtenidos durante la evaluación del modelo propuesto para detección automática de discurso de odio en escenarios bilingües. La evaluación se realizó utilizando el conjunto de prueba estratificado, aplicando métricas estándar de clasificación implementadas mediante la biblioteca Scikit-learn [9].

Las métricas utilizadas fueron *accuracy*, *precision*, *recall* y *F1-score*. La métrica *accuracy* mide la proporción total de predicciones correctas, mientras que *precision* evalúa la proporción de predicciones positivas correctas respecto al total de predicciones positivas realizadas. Por otro lado, *recall* cuantifica la capacidad del modelo para identificar correctamente instancias positivas reales, y el *F1-score* representa la media armónica entre precisión y exhaustividad.

La Figura 4 presenta la matriz de confusión obtenida sobre el conjunto de prueba. Los resultados evidencian una adecuada capacidad discriminativa entre las clases *No Odio* y *Odio*, obteniendo altos valores en la diagonal principal y una cantidad reducida de errores de clasificación.

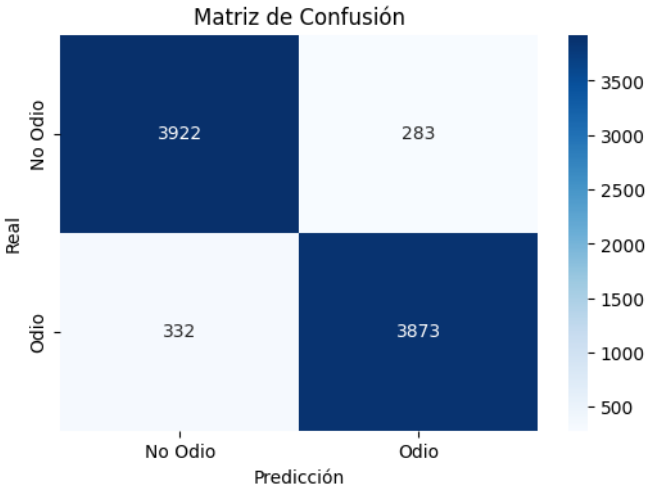


Fig. 4. Matriz de confusión obtenida sobre el conjunto de prueba para la clasificación binaria de discurso de odio.

A partir de la matriz de confusión se obtuvieron 3922 verdaderos negativos y 3873 verdaderos positivos, mientras que los falsos positivos y falsos negativos correspondieron a 283 y 332 instancias, respectivamente. Estos resultados indican que el modelo presenta un equilibrio adecuado entre sensibilidad y especificidad, minimizando errores críticos asociados a la moderación automática de contenido.

Con el objetivo de realizar una evaluación comparativa, el modelo basado en transformers fue contrastado frente a algoritmos tradicionales de clasificación de

texto ampliamente utilizados en tareas de procesamiento de lenguaje natural. La Tabla 4 muestra los resultados obtenidos.

Tabla 4. Comparación de modelos de clasificación

Modelo	**Accuracy**	**F1-score**
Naive Bayes	0.8514	0.8514
Logistic Regression	0.9063	0.9063
Random Forest	0.8923	0.8923
SVM	0.9128	0.9128
BERT	**0.9287**	**0.9287**

Los resultados muestran que el modelo *bert-base-multilingual-cased* obtuvo el mejor desempeño global, superando consistentemente a los modelos clásicos de aprendizaje automático. Este comportamiento se atribuye a la capacidad de los modelos transformer para capturar dependencias semánticas complejas y relaciones contextuales de largo alcance mediante mecanismos de autoatención multi-cabeza.

Asimismo, se realizaron múltiples ejecuciones independientes con diferentes inicializaciones aleatorias para analizar la estabilidad estadística del modelo. Los resultados se presentan en la Tabla 5.

Tabla 5. Resultados del modelo en múltiples ejecuciones

Ejecución	**Accuracy**	**Recall**	**F1-score**
1	0.93	0.93	0.93
2	0.92	0.92	0.92
3	0.92	0.92	0.92
4	0.93	0.93	0.93
5	0.93	0.93	0.93
Promedio	**0.93**	**0.93**	**0.93**

La baja variabilidad observada entre ejecuciones indica estabilidad durante el proceso de optimización y una adecuada capacidad de generalización sobre datos no vistos.

Adicionalmente, se analizó el comportamiento dinámico del entrenamiento mediante el monitoreo de las curvas de pérdida y exactitud. La Figura 5 muestra la evolución de las métricas durante el proceso de ajuste fino del modelo.

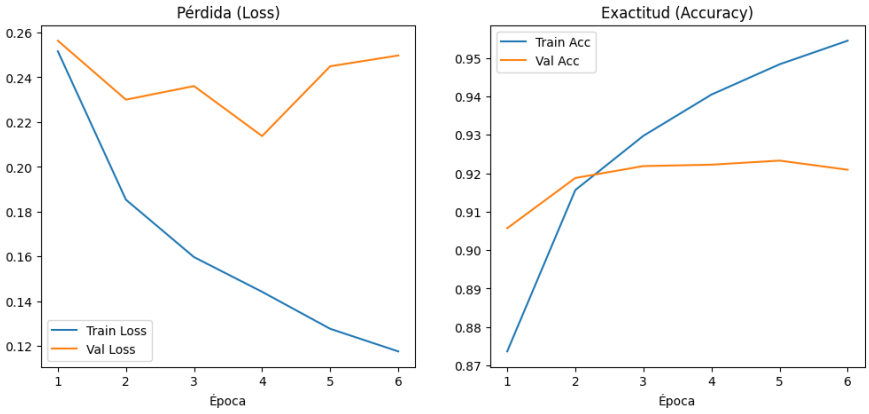


Fig. 5. Evolución de las métricas de entrenamiento y validación durante el proceso de fine-tuning del modelo BERT multilingüe.

La curva de pérdida de entrenamiento presenta una disminución progresiva a lo largo de las épocas, evidenciando convergencia durante la optimización. Por otro lado, la pérdida de validación muestra estabilidad relativa después de las primeras épocas, indicando una reducción controlada del error de generalización.

En términos de exactitud, el modelo alcanzó valores superiores al 95 % sobre entrenamiento y aproximadamente 92 % sobre validación, manteniendo una diferencia moderada entre ambas curvas. Este comportamiento sugiere una adecuada capacidad de aprendizaje sin presencia severa de sobreajuste.

La estabilización de las métricas de validación a partir de las últimas épocas justificó la utilización de mecanismos de regularización como *dropout*, *weight decay*, *gradient clipping* y *early stopping*, los cuales contribuyeron a mantener estabilidad numérica durante el entrenamiento.

4.1. Análisis de casos de falla

A pesar del desempeño obtenido por el modelo, se identificaron casos de error asociados principalmente a sarcasmo, ironía, lenguaje coloquial y expresiones semánticamente ambiguas. En algunos casos, mensajes ofensivos implícitos fueron clasificados como contenido no dañino debido a la ausencia de términos explícitamente tóxicos.

Asimismo, se observaron dificultades en mensajes con mezcla de idiomas (*code-switching*), abreviaciones y modificaciones intencionales de palabras ofensivas mediante símbolos o caracteres especiales, afectando el proceso de tokenización y representación semántica.

Otro factor relevante corresponde a la dependencia contextual de ciertos mensajes. Debido a que el modelo realiza clasificación a nivel de mensaje individual, no siempre es posible capturar referencias implícitas o información proveniente de conversaciones previas.

La Figura 4 muestra que los errores se concentran principalmente cerca del límite de decisión entre las clases *Odio* y *No Odio*, evidenciando la complejidad contextual inherente a la detección automática de discurso de odio.

Como trabajo futuro, se propone incorporar contexto conversacional y modelos especializados por idioma para reducir este tipo de errores.

5. Conclusiones

En este trabajo se presentó el diseño, implementación y evaluación de un sistema de moderación automática de contenido basado en técnicas de Procesamiento de Lenguaje Natural (PLN) y modelos transformer multilingües. El sistema fue desarrollado para la detección automática de discurso de odio en entornos de interacción en tiempo real, integrando un modelo basado en *bert-base-multilingual-cased* junto con mecanismos automáticos de moderación y respuesta.

El modelo propuesto fue entrenado mediante un proceso de *fine-tuning* completo utilizando un dataset bilingüe compuesto por textos en español e inglés provenientes de múltiples fuentes públicas. Los resultados experimentales demostraron que el modelo alcanza un desempeño competitivo, obteniendo un *accuracy* y *F1-score* promedio de 0.9287 sobre el conjunto de prueba, superando modelos tradicionales como Naive Bayes, Random Forest, Regresión Logística y SVM.

El análisis de la matriz de confusión mostró una adecuada capacidad discriminativa entre las clases *Odio* y *No Odio*, presentando una baja cantidad de falsos positivos y falsos negativos. Asimismo, las múltiples ejecuciones realizadas evidenciaron estabilidad estadística durante el entrenamiento, indicando una adecuada capacidad de generalización del modelo frente a variaciones en la inicialización aleatoria.

Desde el punto de vista de optimización, la incorporación de técnicas como *dropout*, *weight decay*, *gradient clipping*, *mixed precision* y *early stopping* permitió estabilizar el entrenamiento y reducir el riesgo de sobreajuste. Adicionalmente, el análisis de las curvas de entrenamiento mostró una convergencia progresiva y consistente de la función de pérdida y la exactitud durante el proceso de ajuste fino.

En términos de implementación, el sistema demostró capacidad de operación en tiempo real mediante la integración de FastAPI y TwitchIO, permitiendo procesar mensajes de chat, ejecutar inferencia automática y aplicar acciones de moderación con baja latencia. Esto confirma la viabilidad del sistema como una solución escalable para plataformas digitales y entornos de streaming.

Como principal contribución, este trabajo propone no únicamente un modelo de clasificación de discurso de odio, sino una arquitectura funcional de moderación automática capaz de integrarse en sistemas reales de comunicación en línea.

Como trabajo futuro, se plantea ampliar el tamaño y diversidad del dataset, incorporar evaluación prolongada en escenarios reales, realizar comparativas

frente a APIs comerciales de moderación y explorar arquitecturas transformer más recientes y especializadas para tareas multilingües de detección de contenido dañino.

Finalmente, los resultados obtenidos permiten concluir que los modelos transformer multilingües representan una alternativa efectiva para la moderación automática de contenido, ofreciendo un equilibrio adecuado entre precisión, robustez y capacidad de despliegue en tiempo real.

Referencias

1. Nabity-Grover, T., Cheung, C.M.K., Thatcher, J.B.: Inside out and outside in: How the COVID-19 pandemic affects self-disclosure on social media. International Journal of Information Management, 55 (2020)
2. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT (2019)
3. Jigsaw: Toxic Comment Classification Challenge. Kaggle (2018)
4. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation, 9(8), 1735–1780 (1997)
5. Devlin, J., et al.: Multilingual BERT Model. Google Research (2019)
6. Schmidt, A., Wiegand, M.: A Survey on Hate Speech Detection using Natural Language Processing. In: Proceedings of SocialNLP (2017)
7. Paszke, A., et al.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. NeurIPS (2019)
8. Wolf, T., et al.: Transformers: State-of-the-Art Natural Language Processing. EMNLP (2020)
9. Pedregosa, F., et al.: Scikit-learn: Machine Learning in Python. JMLR (2011)
10. Liu, Y., Ott, M., Goyal, N., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)
11. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated Hate Speech Detection and the Problem of Offensive Language. Proceedings of ICWSM (2017)
12. Kolhatkar, V., Thain, N., Sorensen, J., Dixon, L., Taboada, M.: Classifying Constructive Comments. arXiv preprint arXiv:2004.09666 (2020)
13. Mathew, B., Saha, P., Yimam, S.M., et al.: HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. Proceedings of AAAI (2021)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS) (2017)
15. Tonneau, M.: Spanish Hate Speech Superset. HuggingFace (2024)
16. Martínez-Cámara, E., et al.: Overview of TASS 2019: One More Further for the Global Spanish Sentiment Analysis. SEPLN (2019)
17. Basile, V., Bosco, C., et al.: SemEval-2019 Task 5: Multilingual Detection of Hate Speech. SemEval (2019)
18. Socher, R., et al.: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. EMNLP (2013)

Optimización mediante Improved Harmony Search (ImHS) para la planificación de trayectorias de UAV: Un enfoque geométrico-elástico en entornos de alta amenaza

Alvaro Sánchez Márquez, Josefina Hernández Tapia,
Alberto Hernández Lazcano, José Antonio Sánchez Zarate,
Hugo Suarez Ramírez

Universidad Autónoma de Tlaxcala,
Unidad Académica Multidisciplinaria Campus Calpulalpan,
México

{asanchez, jhernandezt, ahlazcano, jasanchezz, hugo.suarez.r }@uatx.mx

Resumen. La planificación de trayectorias para Vehículos Aéreos No Tripulados (UAV) en escenarios de alta peligrosidad requiere un equilibrio crítico entre la supervivencia y la eficiencia operativa. Este artículo aborda dicho problema mediante la implementación y comparación de dos enfoques metaheurísticos: el algoritmo convencional de Búsqueda Armoníca (Harmony Search, HS) y una variante optimizada denominada Improved Harmony Search (ImHS), ajustada específicamente para la navegación en un plano cartesiano 2D. La novedad de esta investigación reside en la formulación de una función objetivo tripartita que integra: a) Un modelo de decaimiento cuadrático para la exposición a radares, b) una penalización cuadrática para la restricción estricta de longitud de ruta (L_{max}), y c) Un componente de atracción geodésica, denominado *enfoque geométrico-elástico*, que minimiza la desviación lateral respecto a la ruta más corta teórica, mediante simulaciones numéricas en MATLAB para entornos con múltiples fuentes de amenaza. Los resultados indican que la inclusión del parámetro de atracción geodésica no solo mejora la linealidad de la ruta en zonas de baja amenaza, sino que optimiza el uso del presupuesto de combustible al evitar rodeos excesivos inducidos por mínimos locales de riesgo. El enfoque propuesto ofrece una solución computacionalmente eficiente para sistemas de navegación autónoma, permitiendo a los UAV operar en entornos hostiles con una configuración de compromiso configurable entre sigilo y rapidez de tránsito.

Palabras clave: Harmony search, Improved harmony search, planificación de trayectorias de UAV, atracción geodésica.

Alvaro Sánchez Márquez, Josefina Hernández Tapia, et al.

Optimization Using Improved Harmony Search (ImHS) for UAV Trajectory Planning: A Geometric-elastic Approach in High-threat Environments

Abstract. Trajectory planning for Unmanned Aerial Vehicles (UAVs) in high-risk scenarios requires a critical balance between survivability and operational efficiency. This article addresses this problem by implementing and comparing two metaheuristic approaches: the conventional Harmony Search (HS) algorithm and an optimized variant called Improved Harmony Search (ImHS), specifically tailored for navigation in a 2D Cartesian plane. The novelty of this research lies in the formulation of a tripartite objective function that integrates: a) a quadratic decay model for radar exposure, b) a quadratic penalty for the strict path length constraint (L_{max}), and c) a geodesic attraction component, called the geometric-elastic approach, which minimizes lateral deviation from the theoretical shortest path, using numerical simulations in MATLAB for environments with multiple threat sources. The results indicate that including the geodesic attraction parameter not only improves route linearity in low-threat zones but also optimizes fuel budget use by avoiding excessive detours induced by local risk minima. The proposed approach offers a computationally efficient solution for autonomous navigation systems, enabling UAVs to operate in hostile environments with a configurable trade-off between stealth and transit speed.

Keywords: Harmony search, Improved harmony aearch, planificación de trayectorias de UAV, atracción geodésica.

1. Introducción

En la última década, los Vehículos Aéreos No Tripulados (UAV) han pasado de ser herramientas exclusivas del ámbito militar a componentes esenciales en misiones de búsqueda y rescate, monitoreo ambiental entre otros. Sin embargo, la autonomía plena de estos sistemas sigue enfrentando desafíos críticos cuando operan en entornos de alta amenaza, donde la presencia de sistemas de detección de radar y restricciones estrictas de combustible limitan las opciones de navegación [1]. La planificación de trayectorias (Path Planning) es un problema de optimización NP-duro que busca encontrar una ruta desde un punto inicial a uno final minimizando una función de costo específica. En escenarios hostiles, el objetivo principal es la minimización de la exposición, definida comúnmente mediante modelos de decaimiento de señal [2]. No obstante, la literatura clásica a menudo ignora un conflicto operativo fundamental, la tendencia de los algoritmos de evasión al generar rutas excesivamente sinuosas que, aunque seguras, degradan la eficiencia de la misión al alejarse demasiado de la trayectoria ideal o geodésica.

Para abordar esta complejidad, Pajares [3] alude que el uso de la IA desde sus inicios, ha sido eficiente para resolver problemas, que al principio parecían

juegos, pero en sí, las búsquedas fueron la base para el desarrollo de algoritmos cada vez más precisos, las metaheurísticas bio-inspiradas han demostrado una superioridad notable frente a los métodos deterministas [4] . Entre ellas, la Búsqueda Armónica (Harmony Search HS) destaca por su equilibrio entre la exploración global y la explotación local [5]. En este proyecto se usa HS así como una variante mejorada de HS (especificamente el algoritmo Improved Harmony Search ImHS [6]). A pesar de los avances para la navegación de robots, persiste una brecha en la integración de restricciones elásticas de proximidad. La mayoría de los enfoques se centran únicamente en evitar obstáculos, lo que puede derivar en rutas que consumen el presupuesto de combustible (L_{max}) de manera ineficiente.

Este artículo propone un enfoque innovador denominado atracción geodésica [7,8], donde se introduce una fuerza virtual que vincula la trayectoria optimizada con la línea base de la misión (ruta más corta del inicio a fin de la misión). El presente trabajo contribuye a la disciplina mediante la formulación de un modelo de costo tripartito que penaliza la exposición, el exceso de distancia y la desviación lineal. La validación numérica demuestra cómo la *tensión* hacia la ruta más corta mejora la convergencia y la viabilidad operativa de los UAV en entornos con amenazas.

2. Formulación del problema

Consideramos un vehículo aéreo no tripulado (UAV) operando en un plano cartesiano bidimensional $\mathbb{R}^2$. El objetivo es determinar una trayectoria óptima $\mathbf{P}$ que conecte un punto de origen $\mathbf{P}_{inicio} = (x_0, y_0)$ con un punto de destino $\mathbf{P}_{fin} = (x_n, y_n)$, sujeta a amenazas externas y restricciones cinemáticas. La trayectoria se discretiza en una secuencia de n puntos de control $\mathbf{P} = \{P_0, P_1, \ldots, P_n\}$, donde cada punto se define por sus coordenadas (x_i, y_i). Para reducir la dimensionalidad del espacio de búsqueda, las coordenadas x se distribuyen uniformemente a lo largo del intervalo $[x_0, x_n]$, dejando las coordenadas y como las variables de decisión a optimizar por el algoritmo de Improved Harmony Search.

Función Objetivo La función de aptitud global $f(\mathbf{P})$ se define como un problema de minimización multiobjetivo [9] compuesto por tres términos fundamentales como se presenta en la ecuación (1):

$$f(\mathbf{P}) = \mathcal{E}(\mathbf{P}) + \mathcal{P}(\mathbf{P}) + \mathcal{G}(\mathbf{P}), \tag{1}$$

donde $\mathcal{E}$ es el índice de exposición al radar, el entorno contiene m radares ubicados en posiciones $\mathbf{C}_j$ con intensidades de detección K_j.

La exposición total se modela mediante un decaimiento cuadrático inverso, acumulado a lo largo de todos los puntos de la trayectoria:

$$\mathcal{E}(\mathbf{P}) = \sum_{i=0}^{n} \sum_{j=1}^{m} \frac{K_j}{\|\mathbf{P}_i - \mathbf{C}_j\|^2 + \epsilon}. \tag{2}$$

donde $\|\cdot\|$ denota la norma euclidiana [10] y ϵ es una constante de suavizado para evitar singularidades matemáticas.

$\mathcal{P}$ Representa una penalización por restricción de combustible, para garantizar que la trayectoria sea físicamente viable dado un límite de combustible L_{max}, se aplica una técnica de penalización exterior cuadrática sobre la longitud total de la ruta L_{total}:

$$\mathcal{P}(\mathbf{P}) = \begin{cases} \lambda(L_{total} - L_{max})^2 & \text{si } L_{total} > L_{max} \\ 0 & \text{en otro caso} \end{cases} \tag{3}$$

siendo $L_{total} = \sum_{i=1}^{n} \|\mathbf{P}_i - \mathbf{P}_{i-1}\|$ y λ un coeficiente de penalización de gran magnitud.

Por último $\mathcal{G}$ es la función de atracción geodésica, esta es la contribución principal de este trabajo, denominada *atracción geométrica-elástica*. Se define como la suma de las distancias perpendiculares de cada punto $\mathbf{P}_i$ hacia la línea de referencia $\mathbf{L}_{ref}$ que une el inicio con el fin:

$$\mathcal{G}(\mathbf{P}) = \sum_{i=1}^{n-1} \text{dist}(\mathbf{P}_i, \mathbf{L}_{ref}). \tag{4}$$

Este término actúa como un regularizador que evita que el algoritmo explore regiones del espacio de búsqueda que, aunque seguras, son ineficientes desde una perspectiva de navegación directa.

3. Harmony Search y sus modificaciones

Cuando escuchas una bonita pieza de música clásica, ¿quién puede hacer la conexión entre la interpretación musical y hallar una solución óptima para un problema de diseño difícil u otros problemas de ingeniería?, pues el algoritmo Harmony Search (HS), desarrollado por Greem en 2001 [11], fue inspirado al observar un músico interpretar armoniosamente una melodía, donde el esfuerzo para hallar una armonía es análogo al de encontrar el óptimo en un proceso de optimización. Estas similitudes son las que dan origen a este algoritmo tomando la belleza y armonía musical en una solución para problemas de optimización. Un músico puede realizar una improvisación musical de tres formas y se formaliza en el algoritmo como se muestra en la Tabla 1.

Un nuevo vector armónico se genera considerando dos probabilidades, $r_{accept} \in [0, 1]$ y $r_{pa} \in [0, 1]$, que permiten la selección aleatoria de un valor almacenado en la memoria armónica y un ajuste de tono, respectivamente. Cada tono o variable de diseño en el vector armónico se ajusta dentro de su rango factible, mediante el uso de (5):

$$x_{new} = x_{old} + rand \times bw, \tag{5}$$

donde x_{old} es el tono original, x_{new} es el nuevo tono después de ser ajustado dentro de un ancho de banda bw y $rand \in [-1, 1]$ es un número aleatorio con distribución uniforme.

Tabla 1. Relación entre la improvisación de un músico y el algoritmo de búsqueda armónica [12].

Músico	**Algoritmo Búsqueda Armónica**
1. Utiliza lo aprendido por experiencia, es decir, utiliza su memoria.	1. Se usa la memoria armónica (Harmony Memory HM).
2. Toca algo similar a lo aprendido (ajusta el tono ligeramente).	2. Ajuste de tono.
3. Interpreta algo nuevo (aleatoriedad).	3. Aleatorización.

Por otro lado si la nueva armonía X_{new} produce un mejor desempeño de la función objetivo, entonces X_{old} es remplazada y la memoria armónica es actualizada. El proceso se repite hasta que el número máximo de iteraciones o ciclos se alcanza.

4. Algoritmo Improved Harmony Search (ImHS)

Con el ánimo de tener una mejor diversificación e intensificación también conocidas como exploración y explotación del algoritmo Búsqueda Armónica (Harmony Search HS) de [11], se realizó una modificación en dos aspectos en [6] que consisten en:

1. **Mod_1 :**Reducir el tono de ajuste bw en función al número de iteraciones, es decir, mientras crece el número de iteraciones se reduce en proporción el rango de ajuste de tono de la forma $bw = (U_i - L_i)/g^a$ donde $L_i \leq x_i \leq U_i$, el número de iteraciones está representada por g y a es una constante positiva que define la velocidad con la que bw se acerca a cero, este proceso se implementa en el paso 7 del algoritmo ImHS de la Figura 1 (b).
2. **Mod_2 :** Consiste en reemplazar una componente x_i de la nueva armonía con respecto a una taza de selección asignada como ajuste inteligente r_{ia} esto es $x_i^{new} = x_i^{best}$ este proceso se implementa en el paso 12 del algoritmo ImHS de la Figura 1 (b). Por lo que la probabilidad de realizar la acción de dicho ajuste es $prob_{AI} = r_{accept} \times (1 - r_{pa}) \times r_{ia}$ donde r_{accept} y r_{pa} son la taza de selección de la memoria armónica y taza de ajuste de tono respectivamente del algoritmo en estudio.

Sus autores presentan resultados donde se observa un mejor desempeño en algoritmo ImHS comparado con el algoritmo original HS.

5. Resultados y discusión

Con el fin de validar el enfoque geométrico-elástico propuesto, se diseñó un protocolo experimental en MATLAB consistente en un entorno de alta peligrosidad de 100×100 unidades, con dos fuentes de interferencia de radar.

```
1  Define objective function f(x), x = (x_1, x_2, ..., x_N);
2  Define harmony memory accepting rate r_accept;
3  Define pitch adjusting rate r_pa;
4  Define pitch bandwidth bw;
5  Generate Harmony Memory (HM) with random
   harmonies;
6  while g < max number of iterations do
7      while i <= N do
8          if rand < r_accept then
9              index = rand(1, k) ;
10             if rand < r_pa then
11                 newH(i) =
                   MA(index, i) + bw * rand(−1, 1);
12             else
13                 newH(i) = MA(index, i)
14         else
15             newH(i) = rand(L_i, U_i)
16     Accept the new harmony (solution) if better ;
```

(a). Harmony Search (HS) original.

```
1  Define objective function f(x), x = (x_1, x_2, ..., x_N);
2  Define harmony memory accepting rate r_accept;
3  Define pitch adjusting rate r_pa;
4  Generate Harmony Memory (HM) with random
   harmonies;
5  while g <max number of iterations do
6      while i ≤ N do
7          bw = (U_i − L_i) / g^a ;                    (Mod_1)
           if rand < r_accept then
8              index = rand(1, k) ;
9              if rand < r_pa then
10                 newH(i) =
                   MA(index, i) + bw * rand(−1, 1);
11             else
                   if rand < facIntel then
                       newH(i) = MA(mejor, i);
12                 else
                       newH(i) = MA(index, i);
                                                       (Mod_2)
13         else
14             newH(i) = rand(L_i, U_i)
15     Evaluate Deb conditions;
16     Accept the new harmony (solution) if better ;
```

(b). Improved Harmony Search (ImHS).

Fig. 1. Seudocódigo de los algoritmos HS y ImHS

Para mitigar el sesgo derivado de la naturaleza estocástica de las metaheurísticas y garantizar la validez estadística de los hallazgos, se realizaron 30 ejecuciones independientes para cada algoritmo (HS e ImHS, ver Tablas 2 y 3 respectivamente).

Este tamaño de muestra ($n = 30$) se fundamenta en el Teorema del Límite Central y en las convenciones de la literatura especializada [13,14], asegurando una aproximación a la distribución normal de los costos medios. Dicho procedimiento permite contrastar con rigor la precisión y robustez de la variante ImHS frente al HS original, utilizando la desviación estándar como métrica clave para determinar la significancia de la mejora propuesta [15].

Los parámetros del escenario, y de los algoritmos Harmony Search (HS) y Improved Harmony Search (ImHS) se detallan en la Tabla 4.

Se seleccionó una discretización de $n = 19$ puntos de control más los puntos fijos de origen y final de la trayectoria, para proporcionarle suficiente flexibilidad rodeando las amenazas, manteniendo al mismo tiempo un espacio de búsqueda computable de 21 dimensiones.

El límite de combustible (L_{max}) se estableció en 160 unidades, lo que representa un margen de maniobra de apenas el 13 % sobre la distancia geodésica lineal ($\approx$ 141.4 unidades), es notable la selección de un ancho de banda extremadamente bajo (bw =0.01), esto indica que los algoritmo confían en la exploración inicial para encontrar zonas seguras y utiliza el pitch adjustment para un refinamiento local milimétrico, crucial para ajustar la ruta cerca de los gradientes de riesgo sin violar la restricción estricta de L_{max}.

Tabla 2. 30 soluciones para la trayectoria optimizada con HS

No.	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}
1	0	4.803917	10.779538	15.665011	22.960449	24.999895	30.297821	30.306931	40.469219	39.944101	37.472489	61.516250
2	0	3.862414	10.897701	15.331350	22.987605	28.598747	32.275160	35.016181	39.985916	37.427282	35.584898	60.479504
3	0	5.072814	11.319520	15.953787	20.229224	24.997429	28.56962	35.346694	37.803984	35.335918	38.957168	62.095208
4	0	4.269772	9.402446	15.002718	18.869296	24.979085	30.000595	34.998313	39.620438	37.468736	63.500501	61.894283
5	0	5.333226	9.998470	15.232363	22.709739	25.186068	29.960643	35.008113	38.481335	37.656333	36.762920	61.980604
6	0	4.958411	9.975977	15.017287	19.958082	25.017215	29.940873	35.064535	39.946995	38.366248	36.404777	61.992833
7	0	5.006621	9.169273	14.995251	19.787251	25.309776	24.057462	27.193728	30.057629	30.348392	62.958333	61.260210
8	0	4.771498	12.034389	19.704968	20.747169	28.837229	30.008825	32.639376	35.155662	39.058685	38.15374	62.466781
9	0	4.098688	9.811826	13.504526	23.153063	25.092915	31.050728	35.003910	41.656027	40.592431	39.912562	61.688614
10	0	6.440524	19.480460	24.386727	27.978932	38.398124	40.326705	35.820932	35.668389	35.126291	39.073398	43.150372
11	0	4.998912	10.003278	14.995919	19.126328	24.683802	30.006894	36.286004	38.794129	35.355718	37.320447	61.878891
12	0	5.000492	10.002340	15.155691	20.000525	24.774028	28.785777	32.968962	39.997809	35.730631	36.129086	61.013244
13	0	5.432129	9.676430	14.424022	20.140797	24.289536	30.000824	35.285418	39.262897	38.290572	63.068845	60.001892
14	0	6.434813	9.021576	15.662969	20.078398	23.017655	30.597684	34.718758	38.5407	35.626635	36.701549	61.561209
15	0	5.017290	9.965827	14.911406	20.009019	24.992100	29.995074	34.958666	39.797566	37.824021	36.630326	61.358302
16	0	5.231097	10.237511	16.586865	17.047610	24.137351	30.818467	33.558736	38.681514	35.792166	38.835330	61.325714
17	0	3.226191	13.206076	10.825101	18.477070	25.010073	30.008690	35.606088	39.238232	38.432819	36.826849	39.048526
18	0	4.992868	10.003414	14.997010	20.559799	24.999179	29.995555	35.000075	39.278550	39.061972	36.529353	66.212937
19	0	5.138725	9.738372	16.171124	13.661979	22.935979	30.168610	34.993707	39.969819	38.660118	37.997612	62.417579
20	0	3.996377	11.2145	14.394758	19.351604	26.088208	29.519569	37.525095	31.424016	36.528727	37.293060	56.431069
21	0	5.383600	9.667102	14.418237	20.201354	23.948232	31.615805	35.533968	40.355576	39.288558	36.154187	61.276882
22	0	5.931017	10.050395	15.013072	27.435774	29.532407	33.877486	39.397339	39.807240	38.455460	38.560910	60.577372
23	0	6.510269	9.222477	11.389041	19.801828	25.913984	29.411535	34.659487	38.652886	38.219872	34.547370	62.135288
24	0	4.783021	9.997266	15.889518	19.277404	26.407363	29.023630	35.655652	39.650095	38.438641	36.286513	61.467917
25	0	5.235514	10.328804	14.562744	19.285372	23.512542	30.005141	35.457750	39.315348	37.716911	63.677295	61.869821
26	0	5.400476	9.799484	15.110367	19.999077	25.095931	29.936198	35.729320	39.837597	36.007415	37.408411	60.226347
27	0	4.998483	9.998811	15.005434	20.003884	25.000159	29.995031	35.002817	39.997671	38.581492	36.862304	61.752030
28	0	4.947536	10.027942	14.986598	19.983137	25.091660	29.993735	34.952896	39.935884	38.073469	36.527049	62.427350
29	0	5.004013	9.990530	14.993444	20.009329	24.997146	30.001968	34.996086	39.999353	38.131888	36.107851	61.717412
30	0	4.949759	10.052904	15.009041	19.928926	24.974071	29.957381	34.976694	40.024453	37.475119	62.530805	61.476207

No.	y_{13}	y_{14}	y_{15}	y_{16}	y_{17}	y_{18}	y_{19}	y_{20}	y_{21}	*FO*	*DT*
1	60.019928	64.992008	70.003257	74.579597	79.999113	85.002921	91.139724	94.990210	100	63.743057	156.61358
2	58.792430	65.91990	70.000646	74.997338	79.977037	84.340832	88.178381	93.027327	100	69.091375	156.63381
3	64.540434	65.003554	69.994110	80.555985	80.007532	85.006094	89.99338	95.003474	100	65.776227	153.63524
4	60.010975	65.001468	69.995869	73.032575	80.683823	84.285519	91.043792	95.007888	100	61.117046	156.110665
5	58.524150	64.781014	70.012013	74.659743	79.999099	85.000071	89.998745	95.010720	100	60.592713	155.673882
6	59.962626	64.947108	70.077026	74.992479	79.994231	84.993611	90.023891	94.942084	100	55.866515	155.212243
7	62.831444	66.367492	69.138539	75.914939	81.823462	85.255894	89.756055	95.006271	100	77.381189	156.084801
8	61.963701	65.000731	68.798340	72.756951	80.955834	83.729542	89.631370	93.219784	100	73.684961	153.298262
9	60.463037	64.279965	72.915706	72.944872	78.381749	86.279385	90.942272	94.999948	100	70.879751	154.975822
10	49.036718	53.637160	52.924433	63.922543	64.915097	72.636131	78.160794	91.701645	100	179.905766	159.300602
11	60.004447	64.836052	69.319854	74.433398	79.624540	84.999276	90.270573	95.428832	100	60.011415	154.530359
12	60.470279	65.015005	69.997789	75.422101	79.991747	84.746206	89.992044	93.563423	100	59.728120	155.162666
13	61.170516	65.007260	69.964626	76.848939	81.767967	84.223980	89.995786	95.262987	100	61.916908	153.777944
14	62.622426	61.441674	70.137211	75.451367	77.528333	83.569995	88.976618	97.367582	100	69.989325	156.289678
15	60.098751	65.149578	70.019441	75.075241	79.963669	84.999045	89.981534	94.995300	100	55.864709	153.827496
16	60.007649	65.091581	66.809401	77.388317	81.364119	85.003819	91.498346	93.573655	100	69.254542	155.576916
17	60.751213	64.207377	70.457005	74.154362	78.836510	84.996800	90.377145	95.000226	100	73.404120	155.142484
18	62.083583	65.014392	70.001037	74.998659	79.996177	85.004206	90.089287	94.994279	100	58.035880	158.5801
19	63.344182	65.028151	72.457645	75.003812	81.017290	87.224003	89.933388	96.326031	100	68.781078	156.901177
20	61.253230	64.608569	70.354212	75.281842	76.396316	81.380378	86.210342	94.833088	100	77.549575	155.808866
21	59.977121	64.808923	69.554643	74.514699	80.585395	84.996734	88.763394	95.842324	100	62.200173	155.820833
22	61.118045	64.990575	65.840133	73.690874	79.650148	85.389448	89.285010	97.072934	100	79.218615	154.450474
23	60.733793	66.272671	69.149850	75.123307	79.731950	85.418462	90.259076	96.141845	100	64.904789	157.619213
24	61.842927	67.145288	71.373976	75.701333	78.872435	86.013604	87.343573	95.511676	100	66.173885	154.800188
25	60.564816	64.162174	69.198178	74.979672	79.301206	85.091279	88.968245	95.006620	100	61.713350	154.824657
26	59.903071	65.243277	69.485780	71.064527	82.108370	86.709095	88.830734	94.261002	100	64.724561	154.848389
27	60.000627	64.998484	70.000218	75.000070	80.001048	85.000664	90.002441	95.007330	100	55.467927	154.303840
28	60.135769	64.978380	69.976449	75.047358	79.898119	84.976241	89.953050	94.987761	100	55.874419	155.584586
29	60.001187	64.998774	69.988279	74.986335	79.997748	84.998033	89.995197	95.001457	100	55.517712	155.246762
30	60.113462	65.039562	70.002476	74.944258	80.031772	85.113330	89.999538	95.011394	100	56.525885	154.530917

La Figura 2 presenta las trayectorias optimizadas correspondientes a las cuatro mejores soluciones de los algoritmos HS e ImHS (detalladas en las Tablas 2

Tabla 3. 30 soluciones para la trayectoria optimizada con ImHS

No.	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}
1	0	4.981593	10.004178	15.125319	19.990391	25.036338	29.960428	35.039645	40.018656	37.984570	63.109698	61.306452
2	0	4.949807	9.975970	15.025410	20.059376	25.005699	30.051991	34.974361	39.879506	37.859604	36.893325	61.467348
3	0	4.987130	9.920280	15.050797	19.962489	24.967265	30.067089	34.998058	39.738099	38.013063	62.752208	62.106315
4	0	4.966026	10.071546	15.001345	19.966897	25.035016	29.979944	34.881358	39.991588	38.233816	36.509583	62.268422
5	0	5.015317	10.066186	15.016500	19.995322	25.043424	29.965592	34.974217	39.821281	38.127277	63.130726	61.922721
6	0	5.049047	10.034986	15.040058	19.997863	24.917437	30.017469	34.950588	40.003023	37.361421	36.391582	61.922755
7	0	4.915213	10.077493	15.114784	19.961087	24.913932	29.991367	35.004943	39.825789	38.137460	37.298363	61.147337
8	0	5.006889	10.000326	15.000640	19.996054	24.999121	29.998649	34.991398	40.001715	38.283496	36.499239	61.787260
9	0	4.993368	9.993286	15.026165	19.950556	25.022257	29.904032	35.032655	40.013872	38.871226	63.774169	61.424831
10	0	4.974111	9.992372	15.002598	20.024693	25.006040	30.054723	35.045196	39.946911	38.181707	62.682306	61.977397
11	0	4.963019	9.992202	14.912448	19.962671	24.931360	30.005524	34.972891	40.061655	38.180832	63.023401	61.126197
12	0	4.996879	9.960927	14.995761	19.973680	25.027402	29.990103	34.971806	39.787200	37.996978	37.015180	61.753180
13	0	5.000643	10.002272	14.999615	19.999500	24.994698	30.004368	35.000209	39.993637	38.068250	36.853049	61.702381
14	0	5.054433	9.979169	15.039051	20.072937	25.088317	29.955795	34.985545	39.965846	38.782182	63.003289	61.401251
15	0	5.005450	9.923494	14.987549	20.083060	24.954576	29.911692	34.940413	39.983348	37.548736	37.480717	61.449757
16	0	4.962577	10.026278	15.000211	19.999573	24.916328	29.961502	34.967389	39.946334	37.987032	63.262916	62.052075
17	0	5.034761	9.948097	14.867806	20.065493	25.016754	29.962903	34.984828	39.727336	37.834429	37.388064	61.323741
18	0	5.006572	10.001297	14.992100	19.996413	24.996114	29.996305	34.999334	40.001932	38.314319	36.888237	61.546341
19	0	4.983240	10.000044	14.974065	19.985303	25.011458	30.068582	34.952120	39.896017	37.836020	36.636251	61.519863
20	0	5.098961	10.099204	15.052139	19.988956	25.020279	30.036401	34.936136	40.005241	38.437417	36.051936	61.961751
21	0	4.974301	10.059174	15.011606	19.989879	24.949777	29.994254	34.900848	39.849199	38.447051	36.051424	61.219553
22	0	4.978430	9.973822	15.041939	19.971160	25.047573	29.942898	35.092274	40.006742	38.347561	36.599441	61.631556
23	0	5.078435	10.034785	14.847861	20.004476	25.037251	29.985947	34.897591	39.940696	37.986298	37.825854	61.436415
24	0	5.001657	9.926846	15.038234	19.992836	25.070217	30.038349	34.973101	39.763196	37.934161	37.153638	61.067073
25	0	4.990173	10.035654	15.010792	19.981267	25.120474	30.061635	34.964033	40.007297	37.505896	38.009140	61.831990
26	0	5.019792	9.991075	15.009022	19.890642	25.055496	29.989640	35.004615	39.886558	38.090064	37.131814	60.964866
27	0	4.981156	10.053642	14.933267	20.033059	24.997237	29.977908	34.988468	39.938889	38.112922	36.828938	62.052333
28	0	4.979087	9.930784	14.960994	19.984786	24.966461	29.985270	34.948468	39.964586	37.864095	63.272354	61.844438
29	0	4.930802	9.955223	15.036078	19.968953	24.925667	30.017619	34.974153	39.990463	37.604348	63.419747	61.613770
30	0	5.009260	10.004204	15.000178	20.002843	25.004783	29.997574	35.001170	39.997079	38.087457	36.954099	61.679138

No.	y_{13}	y_{14}	y_{15}	y_{16}	y_{17}	y_{18}	y_{19}	y_{20}	y_{21}	*FO*	*DT*
1	60.290574	64.982210	70.037334	75.041686	80.030220	85.043918	89.985517	95.010000	100	56.514896	154.383368
2	59.989352	65.039891	69.997534	75.005051	79.978796	85.077318	89.966878	95.024338	100	55.793379	153.837966
3	60.025620	65.016003	69.916480	74.954806	79.955849	84.983577	89.986806	95.001870	100	56.517868	153.924001
4	59.995917	64.964344	70.007426	75.015088	79.987249	85.035188	89.941646	95.051791	100	55.829506	155.457508
5	60.078563	64.991359	70.072117	75.008115	80.075470	85.020394	90.003603	95.020619	100	56.422632	154.208211
6	59.973111	65.033937	69.975175	75.078989	79.986751	84.956018	89.981031	95.020988	100	55.867463	155.291226
7	60.005530	64.966076	69.933228	75.046841	80.005068	84.994302	90.062320	94.942716	100	56.014138	152.858367
8	60.003403	64.997289	70.001400	74.994387	80.001439	85.003016	90.000383	95.003357	100	55.467544	154.817992
9	60.047483	65.034287	69.913439	74.980991	80.004762	85.046903	89.904930	95.092787	100	56.616339	154.357368
10	60.065140	65.032741	70.018683	74.974720	80.016194	85.037899	89.988556	94.957928	100	56.356902	153.765033
11	59.999011	64.944741	69.974274	75.021782	79.955212	84.955783	89.968271	94.944472	100	56.535401	154.338935
12	60.094425	65.153085	70.044793	75.008064	79.958237	85.007933	89.940072	95.102313	100	55.866496	153.835906
13	60.004258	64.996064	69.996293	75.000233	79.996714	84.995507	90.004375	95.005771	100	55.460184	154.260909
14	60.013990	65.017222	69.998764	75.048844	80.037357	84.977553	89.905983	94.998733	100	56.493564	153.414171
15	59.996431	65.051207	69.974311	74.958904	80.002773	85.013539	90.037436	95.142371	100	55.892386	153.384754
16	60.247160	65.073227	70.019147	74.984492	79.978041	85.019606	89.979271	94.938917	100	56.467865	154.522653
17	59.951973	65.023434	69.994305	74.987606	79.948849	84.973661	90.033302	95.057553	100	55.995103	152.986593
18	60.005005	65.002598	70.000081	75.002584	80.002734	85.002259	90.004273	94.996094	100	55.466711	154.003553
19	60.029728	64.994462	69.976661	75.038400	80.012001	84.961601	90.011279	95.058118	100	55.737496	154.191880
20	60.032068	64.981031	69.957795	75.011964	79.973241	85.022483	89.997026	94.994007	100	55.861655	155.646871
21	60.084216	64.964723	70.108380	75.025059	80.007929	85.026670	89.972708	94.899607	100	55.962262	154.498377
22	60.017861	65.006146	70.007500	74.933048	79.972963	85.100576	89.991193	95.014427	100	55.844322	154.477922
23	60.003507	64.996180	70.045406	75.063645	80.023597	84.926432	89.982048	94.997859	100	55.988548	152.802572
24	60.169870	65.026918	69.992870	75.000468	79.983441	84.946062	89.985372	95.046118	100	55.886296	152.748328
25	59.977305	65.060889	70.029758	75.022903	79.999388	85.021862	89.946949	94.999190	100	55.957393	153.451104
26	59.997529	64.880439	69.993579	74.976117	79.970270	84.996142	90.011067	94.990153	100	55.822832	152.910322
27	59.987577	65.046295	70.076484	75.024121	79.933600	84.990329	89.995344	95.164409	100	55.870071	154.703242
28	59.985735	65.017382	70.001088	75.067490	79.962827	85.042016	89.952398	94.963193	100	56.429611	154.976555
29	59.992234	64.986168	69.954782	75.094076	79.932172	85.058913	90.049208	95.029304	100	56.560272	155.546966
30	60.005123	65.005749	70.000930	75.005953	79.997898	85.000251	89.999039	94.996225	100	55.462628	154.108832

y 3), superpuestas sobre el mapa de riesgo. Al analizar el desempeño, destaca que la variante ImHS supera consistentemente al algoritmo original; de hecho, incluso

Tabla 4. Parámetros del escenario, HS e ImHS

Parámetro	**Valor**	**Parámetro**	**Valor**
$\mathbf{P}_{inicio}$	$(0, 0)$	Iteración	5000
$\mathbf{P}_{fin}$	$(100, 100)$	Tamaño de HM	10
L_{max}	160	r_{accept}/r_{pa}	0.9 / 0.3
Radar 1 $\mathbf{C}_1$ (K=800)	$(50, 50)$	bw	0.01
Radar 2 $\mathbf{C}_2$ (K=400)	$(30, 70)$	a	1
Puntos de control n	19	r_{ia}	0.7

el peor valor obtenido por ImHS ($f_{min} = 55{,}467544$) es superior al mejor desempeño alcanzado por HS ($f_{min} = 55{,}467927$), logrando una optimización máxima de $f_{min} = 55{,}460185$.Visualmente, la convergencia de ImHS es notablemente más acelerada en comparación con HS. Aunque a simple vista las trayectorias en la Figura 2 puedan parecer idénticas, esto es consecuencia de la escala espacial del escenario, la cual no permite apreciar las variaciones numéricas de alta precisión que distinguen a cada ruta.

Es importante señalar que, si bien ImHS domina en la minimización del riesgo, el algoritmo HS logró una distancia total ligeramente menor ($DT = 153{,}827496$) frente a la mejor de ImHS ($DT = 154{,}003553$). No obstante, todas las rutas demuestran la efectividad de la función objetivo tripartita al desviarse estratégicamente del núcleo crítico del radar central ($K = 800$), priorizando los pasillos de menor exposición.

Es notable que la 8 simulaciones (Figura 2) mantienen una estructura notablemente lineal en las zonas de baja amenaza (cerca del inicio y fin), lo que confirma la influencia reguladora del término de atracción geodésica $\mathcal{G}(\mathbf{P})$. Ambos algoritmos lograron encontrar un equilibrio donde el exceso de distancia ($L_{total} - 141{,}4$) es mínimo, manteniéndose por debajo del límite $L_{max} = 160$.

5.1. Enfoque geométrico-elástico

La inclusión del parámetro de atracción a la ruta más corta ha demostrado ser fundamental para la estabilidad del algoritmo. Sin este término, las heurísticas basadas en la repulsión de obstáculos tienden a generar trayectorias con oscilaciones laterales innecesarias en espacios abiertos, lo que desperdicia combustible y complica el seguimiento de la trayectoria por parte de los actuadores del UAV.

En este estudio, el enfoque geométrico-elástico actúa como un *amortiguador* virtual, ya que permite que la ruta se curve para garantizar la supervivencia cerca de los radares, pero restaura la linealidad de la misión de inmediato una vez superado el peligro. Este comportamiento multiobjetivo es el que permite que la solución final se ajuste tan estrechamente al límite de $L_{max} = 160$, optimizando el uso de los recursos disponibles.

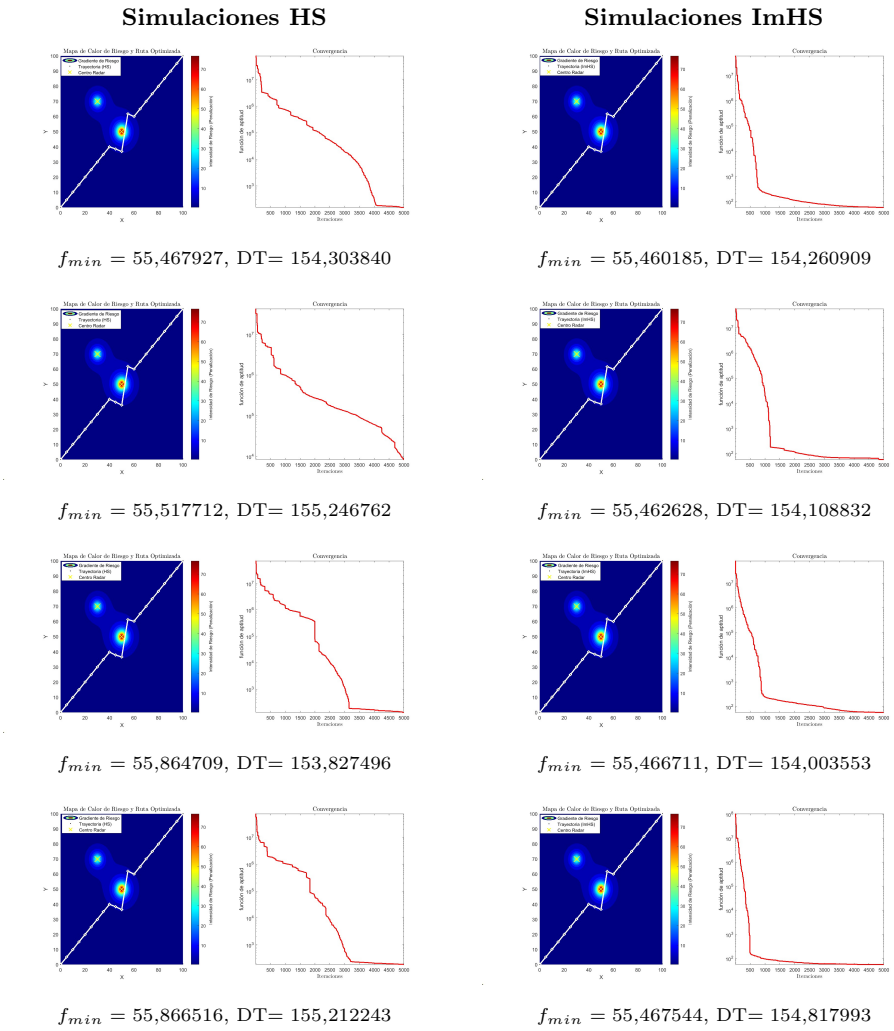


Fig. 2. Comparativa de resultados entre el algoritmo HS (columna izquierda) e ImHS (columna derecha).

6. Conclusión

En este trabajo se ha presentado un estudio comparativo y un enfoque de optimización restringida mediante las metaheurísticas de Búsqueda Armónica (HS) y su variante mejorada, Improved Harmony Search (ImHS), aplicadas a la planificación de trayectorias de UAV en entornos de alta amenaza.

A diferencia de los métodos de evasión convencionales, nuestra propuesta introduce una función de aptitud tripartita que equilibra la seguridad del agente, el consumo de combustible y la eficiencia geométrica.La integración del término

de atracción geodésica ($\mathcal{G}$) permitió generar rutas que no solo evitan los núcleos de riesgo de los radares ($K = 800$ y $K = 400$), sino que restauran la linealidad de la misión de forma inmediata tras superar los obstáculos. Los resultados demuestran que, si bien ambos algoritmos son capaces de resolver el problema, el ImHS exhibe una superioridad estadística y una convergencia más acelerada, logrando minimizar la función objetivo de manera más robusta que el HS estándar.

Esta eficiencia se traduce en una reducción significativa de las oscilaciones laterales innecesarias que suelen presentar los algoritmos basados puramente en repulsión. Ambos algoritmos demostraron precisión al ajustarse al límite de combustible ($L_{max} = 160{,}00$), pero el uso de un ancho de banda fino ($bw = 0{,}01$) en la fase de refinamiento del ImHS fue clave para permitir que la trayectoria se deslice tangencialmente a las zonas de riesgo sin exceder el presupuesto de distancia.En conclusión, el modelo propuesto ofrece una base sólida para sistemas de guiado autónomo donde el sigilo y la rapidez de tránsito son objetivos en conflicto.

La tensión ajustable hacia la ruta más corta, optimizada con mayor eficacia por el algoritmo ImHS, proporciona a los diseñadores de misiones un parámetro intuitivo para controlar la agresividad de la evasión de forma confiable.

6.1. Trabajos futuros

Se sugiere extender este modelo a entornos dinámicos donde la posición de los radares varíe en el tiempo, así como incorporar escenarios de estrés paramétrico para evaluar exhaustivamente las capacidades de respuesta de los algoritmos HS e ImHS, esto incluiría reducir drásticamente el límite de combustible (L_{max}). También se puede contemplar la extensión del algoritmo para la planificación de m vehículos, en este escenario, la función de aptitud debería incluir una condición para evitar colisiones entre ellos, permitiendo que un enjambre de UAVs coordine sus trayectorias para cruzar una zona de n radares minimizando la firma de radar colectiva. La estructura de *armonía* pasaría de representar una sola ruta a representar una partitura orquestal de múltiples trayectorias sincronizadas.

Referencias

1. Geem, Z. W.: Music-Inspired Harmony Search Algorithm: Theory and Applications. Studies in Computational Intelligence, 191, Springer-Verlag, Library of Congress Control Number: 2008944108, Meryland, USA (2009)
2. Yang, X. S.: Nature-Inspired Optimization Algorithms. 1st edn. Elsevier Insights, UK (2014)
3. Pajares, M. G., Santos, P.M.: Inteligencia artificial e ingeniería del conocimiento. 1ª edn. RA-MA S.A. Editorial y Publicaciones, España (2005)
4. Proy, S.P.: Algoritmos heurísticos. Trabajo fin de grado en matemáticas, Universidad de Valladolid, Facultad de ciencias, España (2024)
5. Feng, Q., Mohd, Z. and Kai-Qing, Z.: Algoritmo de búsqueda de armonía y variantes relacionadas: una revisión sistemática. Computación de enjambre y evolutiva, 74(101126) (2022)

6. Portilla-Flores, E. S.-P.: Enhancing the Harmony Search Algorithm Performance on Constrained Numerical Optimization. IEEE Access, 5, 25759-25780 (2017)
7. Khatib, O.: Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research, 5(1), 90–98 (1986)
8. Moore,J. and Garcia, R.: Geodesic Path Deviation in Constrained Environments for Autonomous Vehicles. Journal of Guidance, Control, and Dynamics, 35(4), 1101–1115 (2012)
9. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, New York, USA (2001)
10. Strand, G.: Introduction to Linear Algebra. 5th edn. Wellesley-Cambridge Press, USA (2016)
11. Geem, Z. W., Kim, J. H., Loganathan, G. V.: A New Heuristic Optimization Algorithm: Harmony Search. Simulation, 76(2), 60–68 (2001)
12. Sánchez-Márquez, A., Sánchez-Márquez, S., et al: Síntesis dimensional óptima de un mecanismo para seguimiento de trayectoria por medio de búsqueda armónica y evolución diferencial. Research in Computing Science, 153(7), 273-286 (2024)
13. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary algorithms in quantum computing. Swarm and Evolutionary Computation. Elsevier, 1(1), 13–18 (2011)
14. Montgomery, D. C., Runger, G. C.: Applied Statistics and Probability for Engineers. 7th edn. Wiley, Hoboken (2018)
15. Talbi,E-G.: Metaheuristics: From Design to Implementation. Wiley, Hoboken (2009)

Evaluación híbrida asistida por modelo sustituto para un algoritmo genético aplicado al problema de transporte bajo demanda

Ricardo Pérez Cabrera, Rodolfo Eleazar Pérez Loaiza, Perfecto Malaquías Quintero Flores, Edmundo Bonilla Huerta, Paulina Galindo Garrido, Omar Atriano Venta

TecNM/Instituto Tecnológico de Apizaco, Apizaco, Tlax., México

m17370613@apizaco.tecnm.mx, rodolfo.pl@apizaco.tecnm.mx, data.sci.phd@gmail.com, edmundo.bh@apizaco.tecnm.mx, m19370503@apizaco.tecnm.mx, m25370001@apizaco.tecnm.mx

Resumen. El problema del transporte a demanda (DARP, por sus siglas en inglés) modela los servicios de transporte bajo demanda, donde una flota debe satisfacer las solicitudes de recogida y entrega sujetas a restricciones operativas. En aplicaciones reales, el coste computacional de un algoritmo genético (AG) depende en gran medida de la evaluación de cada individuo, especialmente cuando esta evaluación implica la reconstrucción temporal de rutas, tiempos de viaje basados en la red y tiempos de servicio heterogéneos. Este trabajo propone una estrategia de evaluación híbrida para un AG aplicado al DARP, en la que se utiliza un sustituto estructural basado en una matriz origen-destino (OD) para preevaluar la población y reducir la frecuencia de llamadas al evaluador real. La propuesta se implementa mediante una política basada en individuos en dos fases: una fase principal orientada al ahorro computacional y una fase de refinamiento con una mayor proporción de evaluaciones reales. La evaluación experimental se llevó a cabo en tres instancias de prueba, comparando el AG de referencia y la variante asistida por el sustituto mediante diez ejecuciones independientes por condición. Los resultados demuestran que la estrategia propuesta preserva la calidad de la solución, sin observarse diferencias estadísticamente significativas en la aptitud final, a la vez que reduce significativamente tanto el número de evaluaciones reales como el tiempo de ejecución. Estos hallazgos respaldan el uso de sustitutos estructurales simples e interpretables como un mecanismo eficaz para mejorar la eficiencia computacional de los algoritmos evolutivos en problemas de transporte con respuesta a la demanda y evaluaciones costosas.

Keywords: Dial-a-Ride problem, genetic algorithm, surrogate, origin-destination matrix, hybrid evaluation, expensive combinatorial optimization.

Hybrid Evaluation Assisted by a Surrogate Model for a Genetic Algorithm Applied to the On-demand Transport Problem

Abstract. The Dial-a-Ride Problem (DARP) models demand-responsive transport services in which a fleet must satisfy pickup and delivery requests under operational constraints. In realistic applications, the computational cost of a Genetic Algorithm (GA) depends heavily on the evaluation of each individual, especially when this evaluation involves temporal route reconstruction, network-based travel times, and heterogeneous service times. This work proposes a hybrid evaluation strategy for a GA applied to the DARP, in which a structural surrogate based on an origin-destination (OD) matrix is used to pre-evaluate the population and reduce the frequency of calls to the real evaluator. The proposal is implemented through an individual-based policy in two phases: a main phase oriented toward computational savings and a refinement phase with a higher proportion of real evaluations. The experimental assessment was conducted on three test instances, comparing the baseline GA and the surrogate-assisted variant through ten independent runs per condition. The results show that the proposed strategy preserves solution quality, with no statistically significant differences observed in final fitness, while significantly reducing both the number of real evaluations and execution time. These findings support the use of simple and interpretable structural surrogates as an effective mechanism for improving the computational efficiency of evolutionary algorithms in demand-responsive transport problems with expensive evaluation.

Keywords: Dial-a-Ride problem, genetic algorithm, surrogate, origin-destination matrix, hybrid evaluation, expensive combinatorial optimization.

1. Introducción

El problema de transporte bajo demanda (DARP, *Dial-a-Ride Problem*) constituye un marco de referencia para modelar servicios en los que una flota de vehículos debe atender solicitudes de recogida y entrega bajo restricciones de capacidad, precedencia, ventanas de tiempo y duración de ruta [1]. Este problema es especialmente relevante en contextos de transporte para personas con movilidad limitada, donde no solo importa la eficiencia operativa, sino también la calidad del servicio y el cumplimiento de condiciones temporales y logísticas asociadas a cada usuario [1,2].

Los algoritmos genéticos (GA, *Genetic Algorithm*) representan una alternativa atractiva para explorar el espacio de soluciones del DARP, debido a su capacidad para manejar estructuras combinatorias complejas y restricciones múltiples [3]. Sin embargo, en aplicaciones realistas su desempeño depende en

gran medida del costo computacional del procedimiento de evaluación. Cuando la función de aptitud (*fitness*) de cada individuo exige reconstruir rutas, verificar factibilidad temporal y calcular tiempos de servicio detallados, el proceso evolutivo puede volverse costoso incluso con tamaños de población y horizontes generacionales moderados.

Esta situación adquiere mayor relevancia en el sistema base considerado en este trabajo, donde la evaluación real integra tiempos de traslado sobre la red vial y tiempos de servicio estimados mediante un sistema de inferencia difusa (FIS, *Fuzzy Inference System*) para modelar el abordaje y descenso de pasajeros con movilidad limitada [4]. Aunque esta formulación incrementa el realismo operacional del modelo, también introduce un cuello de botella computacional que limita la eficiencia de la búsqueda evolutiva.

Para enfrentar este problema, el presente trabajo propone una estrategia de evaluación híbrida para un GA aplicado al DARP, en la que un modelo sustituto o aproximado (*surrogate*) estructural basado en una matriz origen-destino (OD, *Origin-Destination*) se utiliza para preevaluar la población y reducir la frecuencia de llamadas al evaluador real. La propuesta se implementa mediante una política basada en individuos (*individual-based*) en dos fases: una fase principal orientada al ahorro computacional y una fase de refinamiento con mayor proporción de evaluaciones reales [5,6,7].

La contribución principal del artículo consiste en mostrar que esta estrategia permite reducir de manera importante el número de evaluaciones reales y el tiempo de ejecución, sin deteriorar de forma significativa la calidad de las soluciones obtenidas. Para ello, se compara el sistema base con una variante asistida por modelo sustituto sobre varias instancias de prueba, considerando métricas de calidad, esfuerzo computacional y convergencia.

2. Antecedentes

La literatura ha abordado el DARP como un problema de transporte bajo demanda con restricciones múltiples. Jaw et al. [2] propusieron una heurística para el caso multi-vehículo con solicitudes anticipadas y ventanas de tiempo, incorporando criterios de calidad del servicio. Posteriormente, Cordeau y Laporte [1] consolidaron una formulación de referencia para el DARP estático multi-vehículo y mostraron la utilidad de enfoques heurísticos y metaheurísticos para tratar su complejidad computacional. Estos trabajos establecen la base estructural del problema sobre la que se construyen enfoques posteriores.

En el ámbito de las metaheurísticas, Jorgensen et al. [3] aplicaron un GA al DARP mediante un esquema de agrupar-primero, rutear-después (*cluster-first, route-second*), evidenciando que los métodos evolutivos pueden producir soluciones competitivas cuando se combinan con mecanismos adecuados de construcción y validación de rutas. Su aportación muestra que, en el DARP, el desempeño del GA depende no solo de los operadores evolutivos, sino también del procedimiento empleado para evaluar la calidad y factibilidad de los individuos.

Cuando la evaluación de la aptitud es costosa, la literatura ha propuesto el uso de *surrogates* o modelos aproximados como mecanismos para estimar respuestas de alto costo computacional mediante evaluaciones más baratas [8]. En el contexto del cómputo evolutivo, Jin [9] clasificó los principales niveles de aproximación y advirtió que sustituir de forma indiscriminada la función real puede inducir sesgos de búsqueda u óptimos falsos. Más adelante, Jin [5] formalizó el marco de la *surrogate-assisted evolutionary computation*, distinguiendo estrategias de gestión del modelo (*model management*) basadas en individuo, en generación y en población.

En el caso específico de la optimización combinatoria costosa, Liu et al. [7] señalaron que esta línea ha recibido menos atención que la optimización continua, a pesar de su importancia en problemas de ruteo, secuenciación y asignación. Por su parte, Hanawa et al. [6] analizaron el efecto de la precisión del *surrogate* sobre distintas estrategias de *model management* y mostraron que los enfoques basados en individuo son especialmente útiles cuando el modelo aproximado no es lo suficientemente preciso como para justificar una dependencia más agresiva.

A diferencia de enfoques basados en metamodelos de regresión o aprendizaje estadístico, el presente trabajo utiliza un modelo sustituto estructural basado en una matriz OD y una verificación barata de factibilidad. Esta elección responde a tres consideraciones del dominio: el espacio de búsqueda combinatorio del DARP no es un espacio vectorial continuo donde los metamodelos clásicos se desempeñen idealmente; la matriz OD aprovecha información estructural exacta dentro de su alcance, sin requerir entrenamiento estadístico adicional; y el enfoque resulta interpretable y libre de hiperparámetros, lo cual es relevante en sistemas operativos donde la trazabilidad del costo aproximado es importante. Esta decisión se relaciona además con el sistema base de Pérez Cabrera et al. [4], en el que un FIS incrementa el realismo operacional del evaluador pero también eleva su costo computacional, lo que justifica la necesidad de la estrategia híbrida propuesta.

3. Sistema base

El método propuesto se construye sobre un sistema base para el DARP en el que cada solución representa un conjunto de rutas asignadas a una flota de vehículos. El objetivo es atender solicitudes de recogida y entrega bajo restricciones de capacidad, precedencia y ventanas de tiempo, minimizando el costo total de operación [1,3].

El sistema base emplea un GA con operadores de selección, cruce, mutación y elitismo para explorar el espacio de soluciones. La calidad del proceso evolutivo depende no solo de los operadores aplicados, sino también del mecanismo utilizado para evaluar la factibilidad y el costo de cada individuo, aspecto crítico en aplicaciones del GA al DARP [3].

La evaluación real de una solución S se realiza mediante un procedimiento que reconstruye temporalmente cada ruta y verifica el cumplimiento de las

restricciones operativas. La aptitud real puede expresarse como

$$f(S) = C_{\text{op}}(S) + P(S), \tag{1}$$

donde $C_{\text{op}}(S)$ representa el costo operacional asociado a los tiempos de desplazamiento y servicio, y $P(S)$ agrupa las penalizaciones por violaciones de capacidad, precedencia y ventanas de tiempo.

Un componente distintivo del sistema base es la incorporación de un FIS para estimar tiempos de abordaje y descenso de pasajeros con movilidad limitada [4]. Esto implica que la evaluación real no solo considera tiempos de traslado sobre la red vial, sino también tiempos de servicio dependientes del perfil funcional del usuario. Por ello, la evaluación exacta resulta operativamente más realista pero también computacionalmente más costosa, lo que motiva la incorporación de la estrategia de evaluación híbrida descrita en la sección siguiente.

El FIS empleado por el sistema base se compone de cuatro subsistemas independientes, uno por cada tipo de herramienta de apoyo del usuario (sin herramienta, bastón, andadera y silla de ruedas). Cada subsistema toma como entrada una única variable lingüística, denominada Movilidad, definida en una escala de 1 a 10 que el observador asigna al usuario, y produce como salida el tiempo de servicio en parada (*dwell time*) correspondiente al abordaje y descenso [4]. Los cuatro subsistemas siguen una estructura Mamdani–Singleton con cinco términos lingüísticos por universo, funciones de membresía gaussianas y una base de cinco reglas. El modelo fue calibrado a partir de 250 observaciones de campo recolectadas en hospitales de Apizaco, Tlaxcala, alcanzando coeficientes de determinación $R^2 \geq 0{,}86$ en validación. La incorporación del FIS al evaluador real permite representar la heterogeneidad de los tiempos de servicio entre usuarios con distinto perfil funcional, en lugar de asumir un tiempo de servicio constante. Su desventaja operativa es que la incorporación de tiempos de servicio heterogéneos durante la reconstrucción temporal de cada ruta incrementa el costo de la evaluación real, lo que motiva la incorporación del modelo sustituto descrita en la sección siguiente. Conviene aclarar que el FIS en sí solo se ejecuta una vez por cliente al inicializar el evaluador, no en cada generación del GA; la Sección 4.1 detalla esta arquitectura.

4. Metodología propuesta

4.1. Modelo sustituto basado en matriz origen-destino

El cuello de botella computacional del sistema base reside en la evaluación real de cada individuo, ya que esta requiere reconstruir temporalmente las rutas, consultar tiempos de traslado sobre la red vial y considerar tiempos de servicio del FIS. Para reducir este costo, se propone un modelo sustituto estructural basado en una matriz OD que almacena los tiempos de traslado entre los nodos relevantes de la instancia (depósito, puntos de recogida y destinos de entrega). La matriz se construye previamente y se reutiliza en evaluaciones posteriores, evitando repetir consultas costosas al modelo de red.

Sea S una solución compuesta por un conjunto de rutas $R = \{r_1, \ldots, r_m\}$. La evaluación aproximada se define como

$$\hat{f}(S) = \sum_{r \in R} \sum_{(i,j) \in A(r)} t_{ij}^{\mathrm{OD}} + \hat{P}(S), \tag{2}$$

donde $A(r)$ es el conjunto de arcos consecutivos de la ruta r, t_{ij}^{OD} es el tiempo precalculado entre los nodos i y j, y $\hat{P}(S)$ es una penalización de factibilidad de bajo costo. La evaluación aproximada es determinista para una misma solución: asigna siempre el mismo valor aproximado mientras no cambien los tiempos precalculados ni la penalización estructural. En consecuencia, el modelo sustituto no busca reemplazar el evaluador real, sino proporcionar una estimación suficientemente informativa para ordenar candidatos prometedores con un costo muy inferior.

Es importante notar que la matriz OD almacena exclusivamente tiempos de traslado entre nodos. Los tiempos de servicio del FIS no forman parte de la suma $\sum t_{ij}^{\mathrm{OD}}$ de la aptitud aproximada, ya que dependen del usuario asignado a cada nodo y no de la secuencia de la ruta. Sin embargo, durante la inicialización del evaluador, el FIS se ejecuta una única vez por cliente para obtener su tiempo de servicio en recogida y entrega, valor que queda asociado al nodo correspondiente. La penalización barata $\hat{P}(S)$ utiliza estos tiempos precalculados únicamente para estimar de forma aproximada el tiempo de llegada a cada entrega y verificar la ventana correspondiente, sin volver a invocar al FIS. Esta arquitectura preserva el carácter económico del modelo sustituto: el FIS se ejecuta una sola vez al inicio y no durante el ciclo evolutivo, y a la vez la búsqueda no queda ciega al efecto de los tiempos de servicio sobre la factibilidad temporal.

4.2. Penalización barata de factibilidad

La evaluación aproximada incorpora un término $\hat{P}(S)$ que penaliza violaciones estructurales detectables sin ejecutar la simulación completa de la ruta. Se consideran: (i) nodos faltantes o duplicados, (ii) violaciones de precedencia, cuando un nodo de entrega aparece antes de su recogida correspondiente, (iii) exceso de capacidad, e (iv) incumplimientos aproximados de ventanas de tiempo estimados a partir del tiempo acumulado con la matriz OD. Esta penalización desincentiva individuos claramente inviables antes de asignarles una evaluación real costosa. La validez definitiva de una solución se establece exclusivamente mediante el evaluador real.

4.3. Estrategia híbrida de evaluación

Se adopta una estrategia de gestión del modelo sustituto de tipo *individual-based* [5], en la cual todos los individuos reciben primero una evaluación aproximada y solo un subconjunto es reevaluado mediante el evaluador real. La política de selección opera mediante dos componentes:

$$S_{\mathrm{elite}} = \mathrm{Top}_{\alpha}(P, \hat{f}), \tag{3}$$

formado por el $\alpha\,\%$ de individuos con mejor aptitud aproximada, y

$$S_{\text{aleat}} \subset P \setminus S_{\text{elite}}, \qquad |S_{\text{aleat}}| = \beta|P|, \tag{4}$$

seleccionado de manera uniforme entre el resto de la población. El conjunto que recibe evaluación real en cada generación es entonces

$$S_{\text{real}} = S_{\text{elite}} \cup S_{\text{aleat}}. \tag{5}$$

Esta combinación equilibra intensificación y exploración: el componente élite dirige el esfuerzo hacia soluciones prometedoras, mientras que el componente aleatorio reduce el riesgo de sesgos sistemáticos del modelo sustituto y preserva diversidad [6].

4.4. Evaluación en dos fases

La gestión del modelo sustituto se organiza en dos fases. En la **fase principal**, correspondiente a la mayor parte del proceso evolutivo, se emplean $\alpha = 0{,}15$ y $\beta = 0{,}15$, de modo que aproximadamente el 30 % de la población recibe evaluación real en cada generación. Posteriormente, en el último 25 % de las generaciones, se activa una **fase de refinamiento** con $\alpha = 0{,}35$ y $\beta = 0{,}15$, por lo que alrededor del 50 % de la población recibe evaluación real. El proceso evolutivo favorece además a los individuos ya validados realmente durante selección y elitismo, lo que disminuye la probabilidad de que soluciones sobrevaloradas por el modelo sustituto dominen la evolución.

4.5. Pseudocódigo del método propuesto

La lógica general del método se resume en el Algoritmo 1 y se ilustra esquemáticamente en la Fig. 1. Ambos elementos son complementarios: el Algoritmo 1 formaliza el flujo de evaluación aproximada, selección del subconjunto S_{real} y transición de fases, mientras que la Fig. 1 resume gráficamente la diferencia operativa entre el GA base y la variante con modelo sustituto. La línea 1 garantiza que la población inicial sea siempre validada realmente; las líneas 5–7 realizan la preevaluación de toda la descendencia con la matriz OD; las líneas 8–12 controlan la transición a la fase de refinamiento en función del progreso generacional; y las líneas 13–19 efectúan la evaluación real selectiva sobre el subconjunto $S_{\text{real}} = S_{\text{elite}} \cup S_{\text{aleat}}$. Finalmente, la línea 20 asegura que el mejor histórico S^* siempre proviene de una evaluación real, evitando que soluciones sobrevaloradas por el modelo sustituto se conviertan en óptimo del proceso.

4.6. Análisis de complejidad

Sea $|S|$ el número total de nodos visitados por una solución. La evaluación aproximada requiere únicamente recorrer las rutas y sumar tiempos OD entre nodos consecutivos:

$$T_{\text{sust}}(S) = O(|S|). \tag{6}$$

Algoritmo 1 Esquema general del GA asistido por modelo sustituto para el DARP

Entrada: Población inicial P_0, número de generaciones G, proporciones α_1, β_1, α_2, β_2
Salida: Mejor solución validada realmente S^*

Evaluar realmente todos los individuos de P_0
$S^* \leftarrow$ mejor individuo de P_0
para $g \leftarrow 1$ **hasta** G **hacer**
    Generar descendencia Q_g mediante selección, cruce y mutación
    para todo $s \in Q_g$ sin evaluación real **hacer**
        Calcular $\hat{f}(s)$ usando la matriz OD y la penalización barata
    fin para
    si $g \leq 0{,}75\,G$ **entonces**
        $\alpha \leftarrow \alpha_1$; $\beta \leftarrow \beta_1$ ▷ Fase principal
    si no
        $\alpha \leftarrow \alpha_2$; $\beta \leftarrow \beta_2$ ▷ Fase de refinamiento
    fin si
    $S_{\text{elite}} \leftarrow \text{Top}_{\alpha}(Q_g, \hat{f})$
    $S_{\text{aleat}} \leftarrow$ muestra aleatoria uniforme de $Q_g \setminus S_{\text{elite}}$ con $|S_{\text{aleat}}| = \beta|Q_g|$
    $S_{\text{real}} \leftarrow S_{\text{elite}} \cup S_{\text{aleat}}$
    para todo $s \in S_{\text{real}}$ **hacer**
        Calcular $f(s)$ con el evaluador real
    fin para
    Actualizar población mediante reemplazo y elitismo, priorizando individuos con evaluación real
    si existe $s \in S_{\text{real}}$ con $f(s) < f(S^*)$ **entonces**
        $S^* \leftarrow s$
    fin si
fin para
devolver S^*

En contraste, la evaluación real implica reconstruir la secuencia temporal de cada ruta, verificar precedencia, capacidad y ventanas de tiempo, e incorporar tiempos de traslado y servicio mediante la red vial y el FIS:

$$T_{\text{real}}(S) = O(|S| + Q_{\text{red}}(S) + Q_{\text{serv}}(S)) \gg T_{\text{sust}}(S), \tag{7}$$

donde $Q_{\text{red}}(S)$ y $Q_{\text{serv}}(S)$ son los costos de cálculo de tiempos sobre la red y de validación temporal con tiempos de servicio heterogéneos, respectivamente. Si la población tiene tamaño N y el algoritmo evoluciona durante G generaciones, el número total de evaluaciones reales se aproxima por

$$E_{\text{real}} \approx \sum_{g=1}^{G} \rho_g\, N, \tag{8}$$

donde $\rho_g \approx 0{,}30$ en la fase principal y $\rho_g \approx 0{,}50$ en la fase de refinamiento. La propuesta reduce sustancialmente el número de evaluaciones costosas sin eliminar la retroalimentación de la aptitud real.

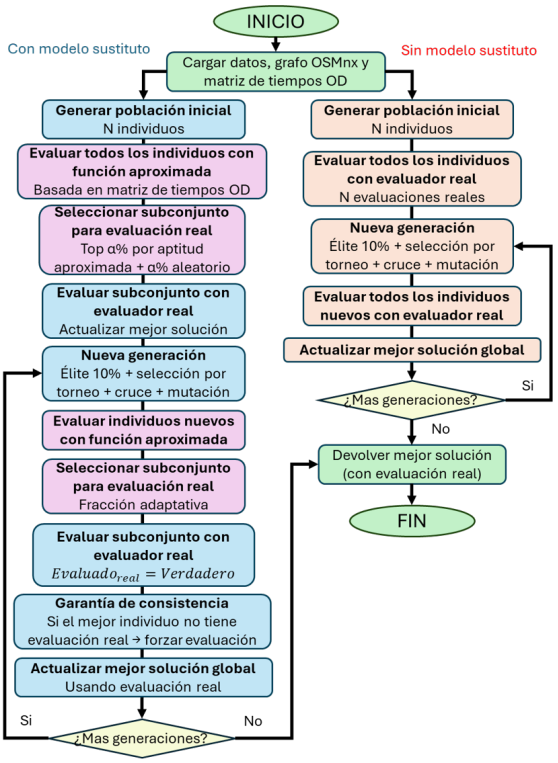


Fig. 1. Comparación esquemática entre el GA base y la variante asistida por modelo sustituto: preevaluación con matriz OD, selección parcial para evaluación real y fase de refinamiento.

5. Diseño experimental

La evaluación experimental se planteó como una comparación controlada entre dos condiciones: GA sin modelo sustituto y GA con modelo sustituto. Se utilizaron tres instancias de prueba: I1 con 30 solicitudes y 3 destinos, I2 con 40 solicitudes y 4 destinos, e I3 con 50 solicitudes y 5 destinos. Para cada instancia se realizaron 10 corridas independientes por condición, generando un total de 60 ejecuciones.

Ambas variantes mantuvieron la misma configuración general del GA: población de 200 individuos, 2 000 generaciones, tasa de cruce de 0.6, tasa de mutación de 0.1, torneo de tamaño 4 y elitismo del 10 %. La única diferencia entre condiciones fue la estrategia de evaluación. En el GA sin modelo sustituto, toda la población recibió evaluación real durante todo el proceso evolutivo. En la variante con modelo sustituto, durante la fase principal se evaluó realmente aproximadamente el 30 % de la población ($\alpha = 0{,}15$, $\beta = 0{,}15$), mientras que en el último 25 % de las generaciones se activó la fase de refinamiento, en la que la fracción evaluada realmente aumentó a alrededor del 50 % ($\alpha = 0{,}35$, $\beta = 0{,}15$).

Tabla 1. Promedios de desempeño por instancia y método.

Instancia	Método	Aptitud promedio	Eval. reales promedio	Tiempo promedio (s)
I1	Sin mod. sust.	571.13	376 981.8	493.27
I1	Con mod. sust.	582.66	140 340.0	257.70
I2	Sin mod. sust.	824.60	311 180.1	529.46
I2	Con mod. sust.	809.45	140 112.9	376.93
I3	Sin mod. sust.	1 322.61	302 532.4	627.95
I3	Con mod. sust.	1 298.65	140 079.5	473.19

Tabla 2. Resultados de la prueba de Wilcoxon por instancia.

Instancia	p-valor aptitud	p-valor eval. reales	p-valor tiempo
I1	0.492	0.002	0.002
I2	0.492	0.002	0.002
I3	0.375	0.002	0.002

Las métricas analizadas fueron aptitud final, número de evaluaciones reales, número de evaluaciones aproximadas y tiempo total de ejecución. Además, se analizó la convergencia a partir de la mejor aptitud acumulada y de la aptitud promedio generacional. Para el contraste estadístico se empleó la prueba de rangos con signo de Wilcoxon, adecuada para comparar dos condiciones con tamaño de muestra pequeño sin asumir normalidad.

6. Resultados y discusión

La comparación mostró un patrón consistente en las tres instancias: la estrategia propuesta preservó la calidad de solución y redujo de manera importante tanto el número de evaluaciones reales como el tiempo de ejecución. La Tabla 1 resume los valores promedio para ambas configuraciones y la Tabla 2 presenta los resultados de la prueba de Wilcoxon.

Antes de discutir las pruebas estadísticas conviene interpretar el patrón observado en la Tabla 1. El número de evaluaciones reales del GA con modelo sustituto se estabiliza cerca de 140 000 en las tres instancias, comportamiento consistente con la política de evaluación propuesta: con población de 200, 2 000 generaciones y fracciones de evaluación real del 30 % en fase principal y 50 % en refinamiento, el total esperado de evaluaciones reales depende de los parámetros α y β y del número de generaciones, no del tamaño de la instancia. Esta independencia es una propiedad estructural de la estrategia y permite anticipar el costo de evaluación a priori.

En contraste, en el GA sin modelo sustituto el número de evaluaciones reales se mantiene también relativamente estable (entre 300 000 y 380 000), pero el costo unitario de cada evaluación crece con $|S|$, ya que la reconstrucción temporal de cada ruta debe procesar más nodos, más arcos y más tiempos de servicio. Esto

explica que el tiempo total aumente con el tamaño de la instancia y, sobre todo, que la reducción relativa de tiempo lograda por el modelo sustituto disminuya de 47.8 % en I1 a 28.8 % en I2 y 24.6 % en I3: en instancias pequeñas, el sobrecosto fijo del evaluador real domina y eliminarlo en parte de la población produce un ahorro porcentualmente mayor; en instancias grandes, el costo se distribuye entre más nodos y la ventaja relativa, aunque sigue siendo significativa, se atenúa. Este comportamiento es coherente con el hecho de que el sobrecosto fijo del evaluador real se distribuye entre más nodos en instancias grandes; la generalización a instancias de mayor escala requiere validación adicional con un banco de pruebas más amplio.

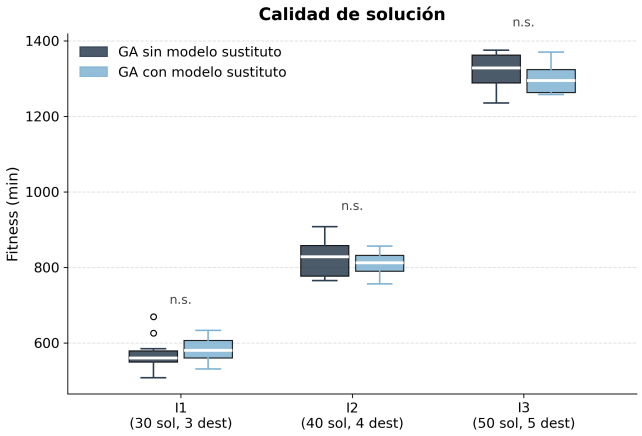


Fig. 2. Distribución de la aptitud final para el GA sin modelo sustituto y la variante con modelo sustituto en las tres instancias analizadas.

En términos de calidad de solución, ambas configuraciones produjeron resultados comparables. Los promedios de aptitud difieren ligeramente entre condiciones, pero la Tabla 2 indica que dichas diferencias no fueron estadísticamente significativas en ninguna instancia. La Figura 2 confirma que las distribuciones de aptitud permanecen próximas entre ambas variantes.

La ausencia de diferencia significativa en aptitud, junto con la reducción altamente significativa en evaluaciones reales y tiempo, sugiere que el modelo sustituto proporciona una preselección suficientemente útil para conservar soluciones competitivas, aunque su valor aproximado no replica la aptitud real exacta. La caracterización formal de la correlación de rango entre $\hat{f}$ y f queda como trabajo futuro. Esto es coherente con la lógica de la estrategia híbrida: para seleccionar candidatos prometedores no se requiere precisión absoluta del modelo aproximado, sino que su ordenamiento aproximado mantenga utilidad práctica en la región de soluciones de alta calidad. La fase de refinamiento corrige cualquier sesgo residual al final del proceso evolutivo, lo que se refleja en que el mejor histórico S^* siempre proviene de una evaluación real.

Un hallazgo que merece atención es que en I2 e I3 el GA con modelo sustituto alcanzó en promedio una aptitud ligeramente mejor que el GA base (809,45

frente a 824,60 y 1 298,65 frente a 1 322,61, respectivamente). Aunque la prueba de Wilcoxon indica que esta diferencia no es estadísticamente significativa, el patrón sugiere que el componente aleatorio S_{aleat} del 15 % no actúa únicamente como salvaguarda contra sesgos del modelo sustituto: también puede contribuir a preservar diversidad en regiones del espacio de búsqueda que el GA base explora con menor intensidad, ya que su selección concentra esfuerzo en los mismos individuos durante muchas generaciones. La evidencia disponible no permite afirmar de manera concluyente este mecanismo, pero el patrón observado es consistente con resultados reportados en la literatura sobre el papel de la diversidad en algoritmos evolutivos asistidos por modelos sustitutos [6].

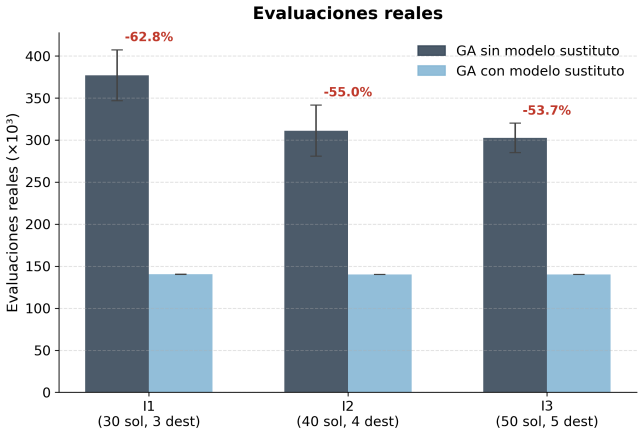


Fig. 3. Número promedio de evaluaciones reales realizadas por el GA sin modelo sustituto y la variante con modelo sustituto. Los porcentajes indican la reducción relativa obtenida con la propuesta.

El efecto más claro de la propuesta se observa en el número de evaluaciones reales. El modelo sustituto redujo este indicador de forma sustancial: de 376 981.8 a 140 340.0 en I1, de 311 180.1 a 140 112.9 en I2 y de 302 532.4 a 140 079.5 en I3, con diferencias altamente significativas ($p = 0{,}002$). La Figura 3 confirma que esta reducción es estable y constituye el resultado más robusto del estudio. Dado que el evaluador real integra reconstrucción temporal, verificación de restricciones, tiempos sobre red y tiempos de servicio del FIS, la disminución de evaluaciones reales implica una reducción directa del componente más costoso del proceso evolutivo.

En cuanto al tiempo de ejecución, la reducción también fue consistente: de 493.27 s a 257.70 s en I1, de 529.46 s a 376.93 s en I2 y de 627.95 s a 473.19 s en I3, con diferencias significativas en las tres instancias ($p = 0{,}002$). La Figura 4 muestra que la reducción del esfuerzo de evaluación sí se tradujo en una disminución observable del tiempo total de ejecución.

Las Figuras 5, 6 y 7 muestran la dinámica de convergencia de ambas configuraciones para cada instancia. En general, ambas variantes siguen trayectorias semejantes: mejora rápida en las etapas iniciales y evolución más gradual con-

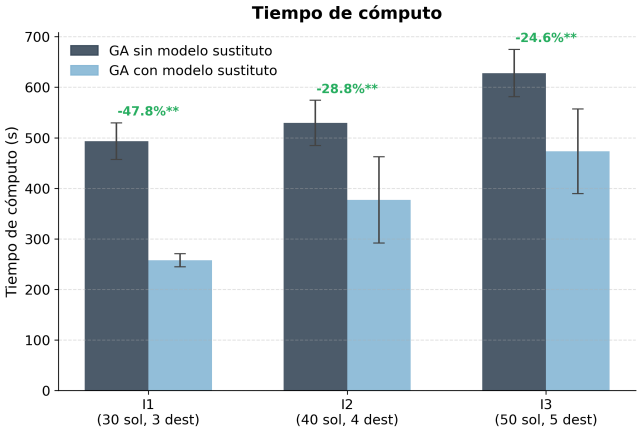


Fig. 4. Tiempo de cómputo promedio del GA sin modelo sustituto y la variante con modelo sustituto. Los porcentajes indican la reducción relativa obtenida con la propuesta.

forme la búsqueda se concentra en regiones de mayor calidad. La transición a la fase de refinamiento en la generación 1 500 no introduce inestabilidades visibles, sino una intensificación progresiva de la evaluación real, lo que indica que la estrategia híbrida no altera de forma drástica la lógica evolutiva del sistema base.

En conjunto, los resultados permiten sostener tres afirmaciones principales. Primero, la estrategia propuesta preserva la calidad de solución, ya que no se detectaron diferencias significativas en aptitud. Segundo, reduce de manera consistente y significativa el número de evaluaciones reales. Tercero, reduce también de manera significativa el tiempo de ejecución en las tres instancias consideradas.

7. Conclusiones

En este trabajo se presentó una estrategia de evaluación híbrida para un GA aplicado al DARP, en la que un modelo sustituto estructural basado en una matriz OD se integra con el evaluador real mediante una política basada en individuos en dos fases. La propuesta no sustituye el sistema base ni la aptitud real, sino que reduce el costo computacional asociado a la evaluación de individuos manteniendo la confiabilidad del proceso evolutivo.

Los resultados mostraron que la estrategia redujo de manera sustancial y consistente el número de evaluaciones reales en las tres instancias analizadas, sin deteriorar de forma estadísticamente significativa la calidad final de las soluciones. Además, esta reducción se tradujo en una disminución significativa del tiempo de ejecución en todas las instancias. El análisis de convergencia confirmó que la variante asistida por modelo sustituto conserva trayectorias evolutivas comparables a las del sistema base. La combinación entre fase principal y fase

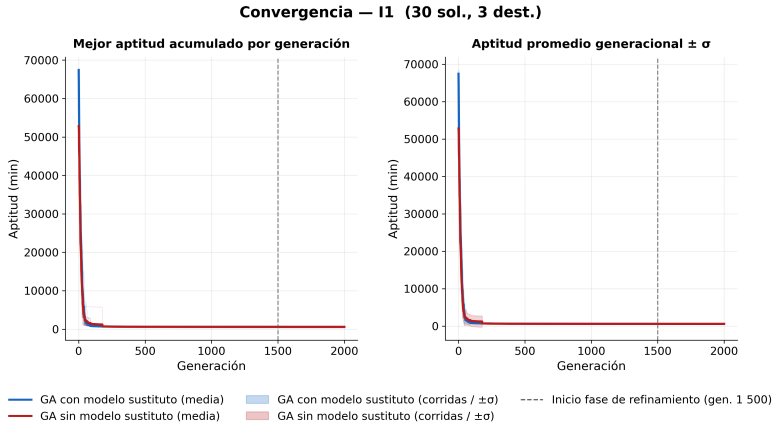


Fig. 5. Convergencia para I1: mejor aptitud acumulada (izq.) y aptitud promedio $\pm\sigma$ (der.) por generación. La línea punteada indica el inicio de la fase de refinamiento.

de refinamiento resultó adecuada para equilibrar exploración, intensificación y control del error de aproximación.

Como limitaciones del trabajo deben señalarse cuatro aspectos. Primero, los resultados deben interpretarse como una validación inicial sobre instancias representativas del sistema operativo bajo estudio (30, 40 y 50 solicitudes con 3 a 5 destinos), no como una demostración general de escalabilidad para instancias de mayor tamaño. Segundo, la ventaja observada depende de la capacidad de la matriz OD y de la penalización estructural para representar de manera razonable el costo real de la solución; en escenarios con mayor dinamismo o fuentes de variabilidad no reflejadas en dicha estructura, la precisión del modelo sustituto podría deteriorarse. Tercero, el estudio no incluye una caracterización formal de la correlación de rango entre $\hat{f}$ y f, métrica que sería deseable en un análisis más profundo del comportamiento del modelo sustituto. Cuarto, la transición entre fase principal y fase de refinamiento se realiza de forma prefijada al 75 % de las generaciones, sin un mecanismo adaptativo que ajuste dinámicamente las proporciones α y β.

En consecuencia, como trabajo futuro se proponen tres líneas: (i) caracterizar la correlación de rango entre el modelo sustituto y el evaluador real mediante coeficientes de Spearman o Kendall sobre parejas $(\hat{f}, f)$ obtenidas en corridas dedicadas; (ii) explorar variantes adaptativas de la política de evaluación híbrida que ajusten α y β en función del progreso evolutivo o de la calidad estimada del modelo sustituto; y (iii) extender la validación a instancias más grandes y a escenarios dinámicos que permitan evaluar la escalabilidad y robustez del enfoque.

En síntesis, los resultados respaldan la hipótesis central del artículo: es posible reducir de forma importante el esfuerzo de evaluación y el tiempo de ejecución en un GA para el DARP mediante un modelo sustituto OD gestionado de forma híbrida, sin afectar de manera significativa la calidad de las soluciones.

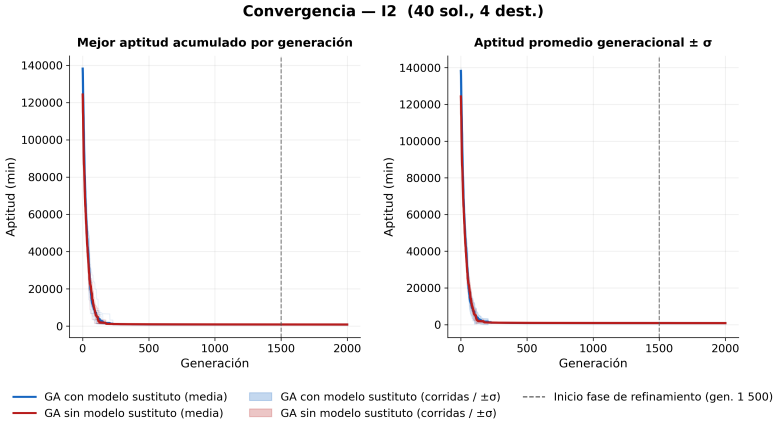


Fig. 6. Convergencia para I2: mejor aptitud acumulada (izq.) y aptitud promedio $\pm\sigma$ (der.) por generación. La línea punteada indica el inicio de la fase de refinamiento.

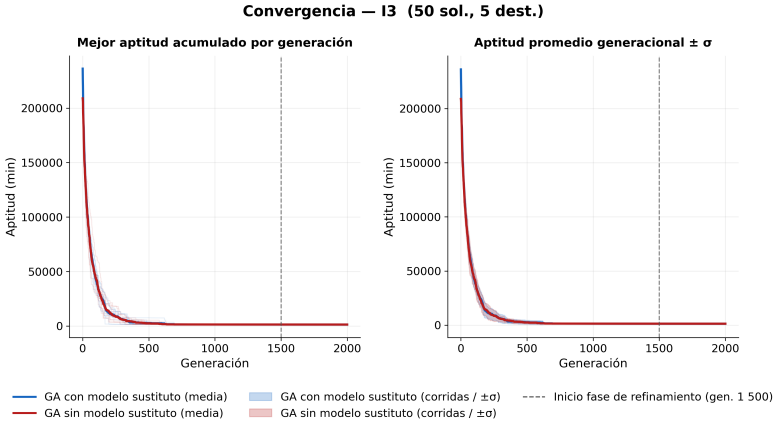


Fig. 7. Convergencia para I3: mejor aptitud acumulada (izq.) y aptitud promedio $\pm\sigma$ (der.) por generación. La línea punteada indica el inicio de la fase de refinamiento.

Referencias

1. Cordeau, J.-F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological 37(6), 579–594 (2003)
2. Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., Wilson, N. H. M.: A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transportation Research Part B: Methodological 20(3), 243–257 (1986)
3. Jørgensen, R. M., Larsen, J., Bergvinsdottir, K. B.: Solving the Dial-a-Ride Problem using Genetic Algorithms. Journal of the Operational Research Society 58(10), 1321–1331 (2007)
4. Pérez Cabrera, R., Pérez Loaiza, R. E., Quintero Flores, P. M.: Fuzzy Inference Modeling for Estimation of Boarding and Alighting Times in Passengers with Limited Mobility. Tecnología y Ciencia Aplicada 8(2), 143–150 (2025)
5. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation 1(2), 61–70 (2011)
6. Hanawa, Y., Harada, T., Miura, Y.: Impact of surrogate model accuracy on performance and model management strategy in surrogate-assisted evolutionary algorithms. Array 27, 100461 (2025)
7. Liu, S., Wang, H., Peng, W., Yao, W.: Surrogate-assisted evolutionary algorithms for expensive combinatorial optimization: A survey. Complex & Intelligent Systems 10, 5933–5949 (2024)
8. Forrester, A. I. J., Sóbester, A., Keane, A. J.: Engineering Design via Surrogate Modelling: A Practical Guide. John Wiley & Sons (2008)
9. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing 9(1), 3–12 (2005)

Medición del balance multimodal en modelos CLIP médicos usando MM-SHAP

Andrés Alberto Góngora-Ramos, Pablo Pancardo García,
Luis Enrique Ramon-Pedrero

Universidad Juárez Autónoma de Tabasco,
México

alberthg.ramos@gmail.com., pablo.pancardo@ujat.mx,
242H21005@alumno.ujat.mx

Resumen. Los modelos médicos de visión–lenguaje basados en CLIP han mostrado buen desempeño en tareas que combinan imágenes y texto; sin embargo, las métricas globales no permiten determinar si ambas modalidades contribuyen de forma equilibrada. En este trabajo se propone un marco de análisis para medir el balance multimodal mediante la aplicación experimental de MM-SHAP. Cada muestra se representa como tokens textuales y parches visuales, cuyas contribuciones se agregan mediante *TScore* e *IScore*. El análisis se evaluó en *Image–Sentence Alignment* (ISA) sobre ROCO y en *Visual Question Answering* (VQA) sobre VQA-Med 2019, considerando cuatro modelos en ISA y dos en VQA. Los resultados muestran que PubMedCLIP presenta el comportamiento más equilibrado, mientras que BioMedCLIP, RCLIP y WhyXRayCLIP exhiben distintos grados de sesgo hacia la modalidad visual. Estos hallazgos evidencian que métricas como similitud imagen–texto o exactitud no reflejan por sí solas la integración multimodal, y resaltan la utilidad de incorporar métricas de explicabilidad en la evaluación de modelos médicos.

Palabras clave: Explicabilidad multimodal, MM-SHAP, modelos de visión–lenguaje, CLIP médico, balance multimodal.

Multimodal Balance Measurement in Medical CLIP Models Using MM-SHAP

Abstract. CLIP-based medical vision-language models have shown good performance in tasks combining images and text; however, global metrics do not allow us to determine whether both modalities contribute in a balanced way. This paper proposes an analytical framework to measure multimodal balance through the experimental application of MM-SHAP. Each sample is represented as textual tokens and visual patches, whose contributions are aggregated using TScore and IScore. The analysis was evaluated in Image-Sentence Alignment (ISA) on ROCO and in Visual

Question Answering (VQA) on VQA-Med 2019, considering four models in ISA and two in VQA. The results show that PubMedCLIP exhibits the most balanced behavior, while BioMedCLIP, RCLIP, and WhyXRay-CLIP show varying degrees of bias toward the visual modality. These findings show that metrics such as image-text similarity or accuracy do not by themselves reflect multimodal integration, and highlight the usefulness of incorporating explainability metrics in the evaluation of medical models.

Keywords: Multimodal explainability, MM-SHAP, vision-language models, medical CLIP, multimodal balance.

1. Introducción

Una red neuronal multimodal, del inglés *Multimodal Neural Network* (MNN), procesa simultáneamente distintos tipos de datos, como texto e imágenes, integrando información proveniente de diferentes modalidades [15,1]. Dentro de este tipo de modelos, *Contrastive Language–Image Pre-training* (CLIP) [13] ha cobrado relevancia en el análisis de imágenes y reportes médicos, debido a su capacidad para relacionar información visual con descripciones textuales [19]. Sus versiones preentrenadas se han utilizado en aplicaciones clínicas como diagnóstico de enfermedades torácicas y segmentación de órganos [14,12,9].

En medicina, la transparencia de los modelos de inteligencia artificial (IA) es fundamental, ya que la falta de explicaciones puede limitar su adopción en escenarios clínicos [8]. En este contexto, la inteligencia artificial explicable, del inglés *Explainable Artificial Intelligence* (XAI), permite analizar por qué un modelo toma determinadas decisiones. Para sistemas multimodales, MM-SHAP extiende la técnica SHAP (*SHapley Additive exPlanations*) con el fin de estimar la contribución relativa de cada modalidad en una predicción [10].

El balance multimodal se refiere al grado en que distintas modalidades contribuyen de manera equilibrada al resultado final. Cuando un modelo depende excesivamente del texto o de la imagen, puede presentar un sesgo modal que no siempre se refleja en métricas tradicionales de desempeño. Dado lo anterior, en este trabajo se propone un marco de análisis para medir el balance multimodal en modelos médicos basados en CLIP mediante la aplicación experimental de MM-SHAP.

El resto del artículo se organiza de la siguiente manera. La Sección 2 presenta los trabajos relacionados; la Sección 3 describe la metodología; la Sección 4 detalla los datos y el diseño experimental; la Sección 5 presenta los resultados y la discusión; finalmente, la Sección 6 expone las conclusiones.

2. Trabajo relacionado

MM-SHAP extiende SHAP a modelos multimodales al cuantificar la contribución de cada modalidad, como texto e imagen, a nivel de muestra y de conjunto de datos [10]. Su aplicación ha permitido identificar dependencias modales

en modelos de visión–lenguaje, como una mayor dependencia textual en BLIP y un comportamiento más equilibrado en BLIP2 y FLAVA [3].

En medicina, SHAP se ha usado para explicar modelos multimodales, por ejemplo, en la predicción de comorbilidades en epilepsia [7]. Además, SHAP-CAT utiliza valores de Shapley para integrar modalidades histopatológicas y mejorar la clasificación de cáncer [16]. En VQA, se ha mostrado que los modelos pueden depender excesivamente del lenguaje e ignorar la imagen, lo que motiva analizar explícitamente el balance multimodal [5].

3. Metodología

En este trabajo se propone un marco de análisis para evaluar el balance multimodal en modelos médicos de visión–lenguaje mediante la aplicación experimental de MM-SHAP. La formulación matemática original de MM-SHAP no se modifica; en su lugar, se adapta su uso a modelos tipo CLIP entrenados o ajustados al dominio médico. Esta adaptación consiste en: i) representar cada entrada como una combinación de unidades textuales y visuales explicables, ii) definir una salida escalar compatible con SHAP para cada tarea evaluada, iii) implementar un esquema de enmascaramiento para tokens y parches visuales, y iv) agregar las atribuciones por modalidad para calcular métricas de contribución y balance multimodal.

3.1. Descripción general del marco de análisis propuesto

El marco propuesto estima la contribución relativa de texto e imagen en la salida de un modelo multimodal a partir de una entrada (x^{txt}, x^{img}). Para ello, cada muestra se transforma en una representación conjunta compuesta por unidades explicables: tokens en la modalidad textual y parches o regiones en la modalidad visual. Sobre esta representación se aplica MM-SHAP para obtener atribuciones a nivel de característica. Posteriormente, los valores SHAP se agregan por modalidad con el fin de cuantificar la contribución total de texto e imagen en cada predicción.

3.2. Adaptación experimental de MM-SHAP a modelos médicos de visión–lenguaje

MM-SHAP fue propuesto como un marco de explicabilidad para modelos multimodales basado en valores de Shapley. En este trabajo no se modifica su formulación matemática original; la adaptación realizada corresponde a su aplicación experimental en modelos médicos de visión–lenguaje tipo CLIP. Para ello, cada entrada se representa como $\mathbf{z} = [z_1, z_2, \ldots, z_M]$, donde cada z_i es una unidad explicable: un token textual o un parche visual. Las imágenes y los textos se preprocesan con el procesador de cada modelo, respetando su resolución de entrada, normalización visual, tokenización y longitud máxima de secuencia.

La función de predicción $f(\mathbf{z})$ se define como una salida escalar asociada a la decisión analizada. En *Image–Sentence Alignment* (ISA), corresponde a la puntuación de similitud entre la imagen médica y su descripción textual. En *Visual Question Answering* (VQA), corresponde a la puntuación asignada a una respuesta candidata, condicionada por la imagen y la pregunta. En este último caso, la entrada textual combina la pregunta y la respuesta candidata.

Bajo esta formulación, MM-SHAP estima la contribución marginal de cada token textual y parche visual respecto a $f(\mathbf{z})$ mediante coaliciones de características presentes y ausentes. Las atribuciones obtenidas se agregan por modalidad para calcular las métricas de contribución textual y visual. Por tanto, la adaptación propuesta no introduce nuevas ecuaciones, sino que especifica cómo representar las entradas médicas y cómo transformar las salidas de los modelos en una puntuación escalar comparable entre tareas y arquitecturas.

3.3. Enmascarador personalizado y envoltura de predicción

Para aplicar MM-SHAP a los modelos evaluados, se implementaron un enmascarador personalizado y una envoltura de predicción. El enmascarador genera muestras parcialmente observadas a partir de una coalición binaria, preservando la estructura requerida por los codificadores textual y visual. En texto, las características ausentes se sustituyen por un token de relleno compatible con el tokenizador, manteniendo los tokens especiales. En imagen, los parches ausentes se reemplazan por una representación base sobre el tensor de entrada, sin alterar la geometría esperada por el codificador visual.

El enmascaramiento se utiliza solo como una perturbación controlada para estimar contribuciones, no como parte de la predicción final. En imágenes médicas, se aplica a nivel de parches y no de estructuras anatómicas completas; por ello, las explicaciones deben interpretarse como atribuciones visuales aproximadas, no como segmentaciones clínicas ni localizaciones diagnósticas.

La envoltura de predicción transforma la salida del modelo en una función escalar adecuada para SHAP. Para una entrada enmascarada $\tilde{\mathbf{z}}$, el predictor devuelve una puntuación $f(\tilde{\mathbf{z}})$ asociada a la decisión analizada, como similitud imagen–texto en ISA o afinidad con una respuesta candidata en VQA.

3.4. Métricas de contribución y balance multimodal

Las métricas de contribución se calcularon siguiendo la formulación original de MM-SHAP, sin modificar sus ecuaciones. Una vez obtenidos los valores SHAP, las atribuciones individuales se agregan por modalidad:

$$S_{txt} = \sum_{i \in \mathcal{T}} |\phi_i|, \qquad S_{img} = \sum_{i \in \mathcal{I}} |\phi_i|,$$

donde $\mathcal{T}$ y $\mathcal{I}$ representan los conjuntos de características textuales y visuales, respectivamente, y ϕ_i es la atribución SHAP de la característica i.

A partir de estas cantidades se calculan puntajes modales normalizados:

$$TScore = \frac{S_{txt}}{S_{txt} + S_{img}}, \qquad IScore = \frac{S_{img}}{S_{txt} + S_{img}}.$$

Estas métricas permiten medir la contribución relativa de cada modalidad en una muestra. Valores cercanos entre *TScore* e *IScore* indican mayor balance multimodal, mientras que diferencias grandes sugieren sesgo hacia una modalidad dominante. A nivel global, estas puntuaciones se agregan sobre el conjunto de datos para identificar tendencias de dependencia modal y comparar el comportamiento de distintos modelos y tareas.

4. Datos y diseño experimental

4.1. Conjuntos de datos

Se utilizaron dos conjuntos de datos médicos de visión–lenguaje. Para *Image–Sentence Alignment* (ISA) se empleó **ROCO**[1] (*Radiology Objects in Context*) [11], compuesto por pares de imágenes radiológicas y descripciones textuales. Para *Visual Question Answering* (VQA) se utilizó **VQA-Med 2019**[2] [2], compuesto por imágenes médicas, preguntas clínicas y respuestas de referencia. Estos conjuntos permiten evaluar dos escenarios complementarios: alineación imagen–texto y razonamiento visual condicionado por lenguaje.

4.2. Tareas y formulación experimental

Se consideraron dos tareas. La primera fue ISA, formulada como un problema de correspondencia entre una imagen médica y un texto, donde el modelo debe asignar mayor afinidad a pares semánticamente consistentes. La segunda fue VQA, formulada como una tarea de *answer selection*, en la que el modelo puntúa un conjunto de respuestas candidatas condicionado por la imagen y la pregunta, seleccionando aquella con mayor afinidad. Esta formulación permitió analizar el balance multimodal bajo dos regímenes distintos: uno centrado en la alineación entre modalidades y otro en la integración visual–textual para la selección de respuestas.

4.3. Modelos evaluados

Se evaluaron cuatro modelos médicos de visión–lenguaje tipo CLIP: **PubMedCLIP** [4], **BioMedCLIP** [18], **RCLIP** [6] y **WhyXRayCLIP** [17]. Estos modelos fueron seleccionados por su disponibilidad pública, su especialización biomédica o radiológica y su capacidad para estimar afinidad imagen–texto, lo que permite comparar distintos patrones de dependencia textual y visual.

[1] `https://www.kaggle.com/datasets/virajbagal/roco-dataset`
[2] `https://github.com/abachaa/VQA-Med-2019`

Todos los modelos fueron integrados bajo el mismo *pipeline* explicable. En ISA se evaluaron los cuatro modelos. En VQA solo se consideraron **PubMedCLIP** y **BioMedCLIP**, ya que son compatibles con la formulación de *similarity-based scoring*. **RCLIP** y **WhyXRayCLIP**, orientados principalmente a alineamiento imagen–texto, no se aplicaron directamente a esta tarea.

4.4. Implementación y configuración

Los experimentos se realizaron en *Google Colab Pro* utilizando una GPU NVIDIA T4 de 15 GB. El entorno experimental se implementó en Python 3.10 con `PyTorch`, `Transformers`, `OpenCLIP` y `SHAP` como librerías principales. Para el análisis y visualización de resultados se utilizaron `NumPy`, `Pandas` y `Matplotlib`. Adicionalmente, se empleó una infraestructura modular desarrollada para este estudio, orientada a la carga de datos, ejecución de inferencia y cálculo de explicaciones multimodales.

La configuración experimental mantuvo parámetros consistentes entre modelos para asegurar comparabilidad. Las explicaciones se calcularon con un presupuesto de entre 20 y 50 evaluaciones SHAP por instancia, dependiendo de la complejidad del modelo. La representación visual se definió en términos de 49, 196 o 256 parches, de acuerdo con el codificador visual correspondiente, mientras que la entrada textual se limitó a un máximo de 77 tokens siguiendo la configuración estándar de CLIP.

4.5. Protocolo de evaluación

El análisis se realizó a nivel de muestra y a nivel de conjunto de datos. Para cada instancia se obtuvieron valores SHAP por característica, que posteriormente se agregaron por modalidad para calcular las métricas de contribución textual y visual descritas en la sección anterior. Estas métricas se resumieron globalmente para comparar el balance multimodal entre modelos y tareas.

En ROCO, el *split* de validación se utilizó para análisis preliminar, mientras que el *split test* se reservó para el reporte final en ISA. En VQA-Med 2019, el *split* de entrenamiento se empleó para análisis preliminar y el *split test* para la evaluación final. Ambos conjuntos de datos se procesaron mediante cargadores dedicados que filtran muestras inválidas y estandarizan el formato multimodal de entrada.

5. Resultados y discusión

5.1. Resultados en ISA

La Tabla 1 resume los resultados de balance multimodal para la tarea de *Image–Sentence Alignment* (ISA) en el *split test* de ROCO.

En general, **PubMedCLIP** es el único modelo con comportamiento cercano al equilibrio, con $TScore = 0.544$, $IScore = 0.456$ y $\Delta = 0.089$, además del mayor porcentaje de muestras balanceadas (35.2 %).

En contraste, **BioMedCLIP**, **RCLIP** y **WhyXRayCLIP** presentan sesgo hacia la modalidad visual, reflejado en valores negativos de Δ y menores porcentajes de muestras balanceadas.

Este sesgo es moderado en **BioMedCLIP** ($\Delta = -0.219$) y **RCLIP** ($\Delta = -0.264$), y más pronunciado en **WhyXRayCLIP** ($\Delta = -0.384$), que además registra solo 0.4 % de muestras balanceadas.

En términos de similitud imagen–texto, **RCLIP** obtiene el mayor valor medio de ℓ_{norm} (0.849), mientras que **WhyXRayCLIP** presenta el menor (0.357). Estos resultados indican que un mayor alineamiento imagen–texto no implica necesariamente un mejor balance multimodal, ya que el modelo con mayor similitud media no es el más equilibrado.

Table 1. Resultados finales de balance multimodal y similitud imagen–texto en ISA (*split test*). Los mejores resultados se resaltan en negritas. Para Δ, se resalta el valor más cercano a cero.

Modelo	**TScore** ($\mu \pm \sigma$)	**IScore** ($\mu \pm \sigma$)	Δ (μ)	**% bal.** ($\lvert\Delta\rvert < 0.1$)	ℓ_{norm} ($\mu \pm \sigma$)
PubMedCLIP	$\mathbf{0.544 \pm 0.090}$	$\mathbf{0.456 \pm 0.090}$	**0.089**	**35.2**	0.515 ± 0.131
BioMedCLIP	0.390 ± 0.084	0.610 ± 0.084	-0.219	21.8	0.503 ± 0.141
RCLIP	0.368 ± 0.101	0.632 ± 0.101	-0.264	17.6	$\mathbf{0.849 \pm 0.131}$
WhyXRayCLIP	0.308 ± 0.061	0.692 ± 0.061	-0.384	0.4	0.357 ± 0.146

La Fig. 1 resume el comportamiento modal de los modelos evaluados en la tarea de ISA sobre el *split test*. **PubMedCLIP** es el único modelo con contribuciones relativamente equilibradas entre texto e imagen, mientras que **BioMedCLIP**, **RCLIP** y **WhyXRayCLIP** muestran un predominio de la modalidad visual. Este sesgo es más pronunciado en **WhyXRayCLIP**, que presenta la mayor diferencia entre *IScore* y *TScore*. En conjunto, la figura confirma que el balance multimodal varía significativamente entre arquitecturas, aun dentro de modelos especializados en el dominio biomédico.

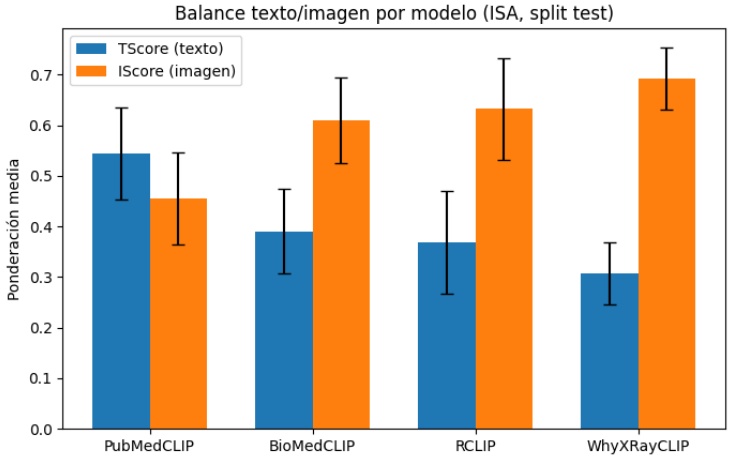


Fig. 1. Comparación de TScore e IScore por modelo en ISA (*split test*).

La Fig. 2 presenta un ejemplo cualitativo de ISA en ROCO. **PubMedCLIP** muestra la distribución más equilibrada entre tokens y parches visuales (*TScore* = 46.0%, *IScore* = 53.9%), mientras que **BioMedCLIP**, **RCLIP** y **WhyXRayCLIP** asignan mayor peso a la modalidad visual, con *IScore* superiores al 60%. Esto sugiere que, para esta muestra, la afinidad imagen–texto se apoya principalmente en regiones visuales en estos modelos, lo cual coincide con las tendencias cuantitativas del conjunto de prueba.

5.2. Resultados en VQA

La Tabla 2 resume los resultados de balance multimodal en la tarea de *Visual Question Answering* (VQA) sobre el *split test* de VQA-Med 2019. Ambos modelos presentan sesgo hacia la modalidad visual, aunque con distinta intensidad. **PubMedCLIP** muestra un sesgo moderado ($\Delta = -0.133$), con 29.2 % de muestras balanceadas, mientras que **BioMedCLIP** exhibe un sesgo visual mucho más pronunciado ($\Delta = -0.517$) y ausencia total de muestras balanceadas.

A pesar de esta diferencia en balance multimodal, ambos modelos alcanzan una exactitud similar (24.2 % en **PubMedCLIP** y 24.0 % en **BioMedCLIP**), lo que indica que el desempeño final no refleja por sí solo el grado de dependencia modal. En términos de similitud normalizada, ambos mantienen valores comparables, con ligera ventaja para **PubMedCLIP** (0.588 frente a 0.554).

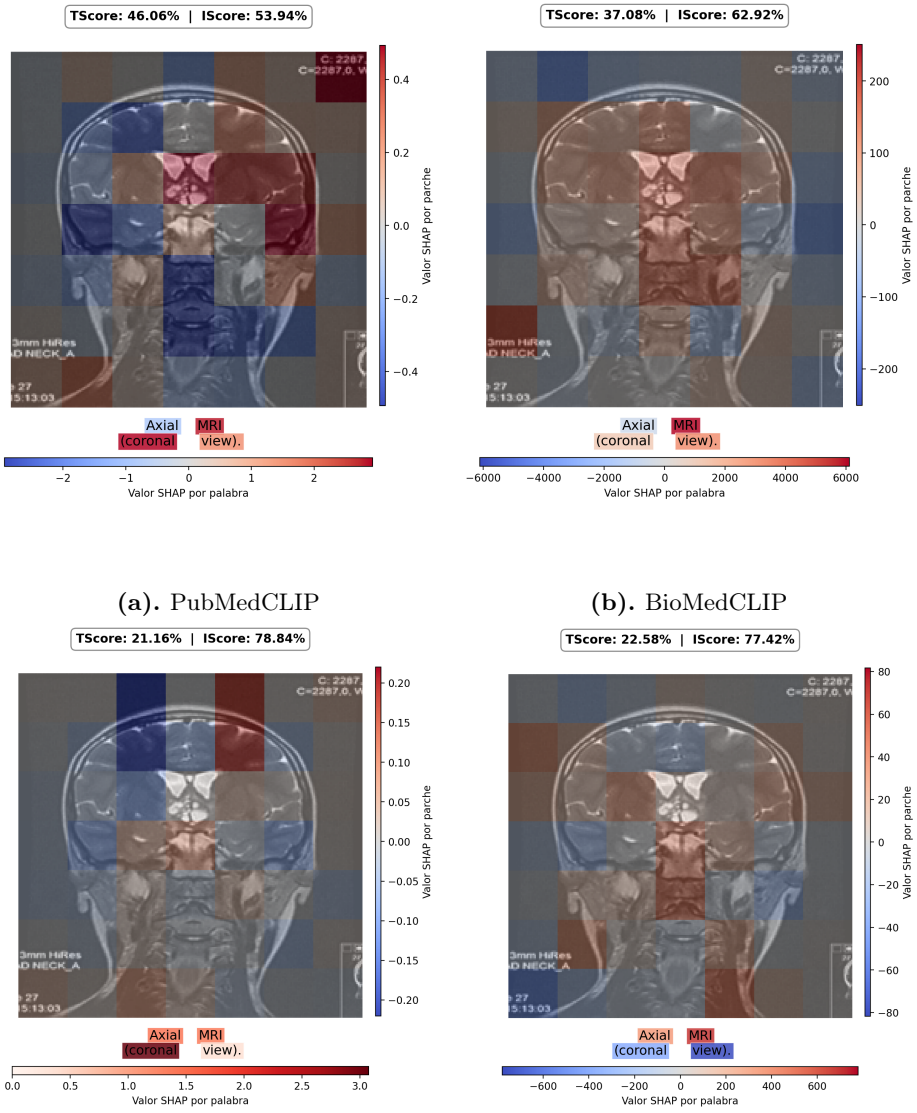
Table 2. Resultados finales de balance multimodal en VQA (*split test*). Los mejores resultados se resaltan en negritas. Para Δ, se resalta el valor más cercano a cero.

Modelo	**TScore** ($\mu \pm \sigma$)	**IScore** ($\mu \pm \sigma$)	Δ (μ)	**% bal.** ($\lvert\Delta\rvert < 0.1$)	**Accuracy (%)**	ℓ_{norm} ($\mu \pm \sigma$)
PubMedCLIP	$\mathbf{0.434 \pm 0.100}$	$\mathbf{0.566 \pm 0.100}$	$\mathbf{-0.133}$	**29.2**	**24.2**	$\mathbf{0.588 \pm 0.196}$
BioMedCLIP	0.242 ± 0.057	0.758 ± 0.057	-0.517	0.0	24.0	0.554 ± 0.172

La Fig. 3 resume el balance multimodal de los modelos evaluados en la tarea de VQA sobre el *split test*. En ambos casos se observa un predominio de la modalidad visual, aunque con distinta intensidad. **PubMedCLIP** presenta un sesgo moderado hacia la imagen, con valores de *TScore* e *IScore* relativamente cercanos, mientras que **BioMedCLIP** muestra una separación mucho mayor entre ambas métricas, evidenciando una dependencia visual más pronunciada. En conjunto, la figura confirma que dos modelos con exactitud similar pueden diferir sustancialmente en su balance multimodal. La Fig. 4 muestra un ejemplo cualitativo de MM-SHAP en VQA. En ambos modelos predomina la modalidad visual, aunque **BioMedCLIP** conserva una contribución textual relativamente mayor que **PubMedCLIP** en esta muestra. Este patrón coincide con la tendencia global hacia la modalidad visual observada en VQA.

5.3. Discusión

Los resultados muestran que el balance multimodal varía entre modelos y tareas. En ISA, **PubMedCLIP** fue el modelo más equilibrado, mientras que



(a). PubMedCLIP **(b).** BioMedCLIP

(c). RCLIP **(d).** WhyXRayCLIP

Fig. 2. Heatmaps ISA — Muestra 0.

BioMedCLIP, **RCLIP** y **WhyXRayCLIP** presentaron sesgo hacia la modalidad visual. En VQA, ambos modelos evaluados también mostraron predominio de la imagen, aunque este fue más marcado en **BioMedCLIP**.

Estas diferencias pueden explicarse por la interacción entre el preentrenamiento de cada modelo y la formulación de los conjuntos de datos. Mientras ISA evalúa

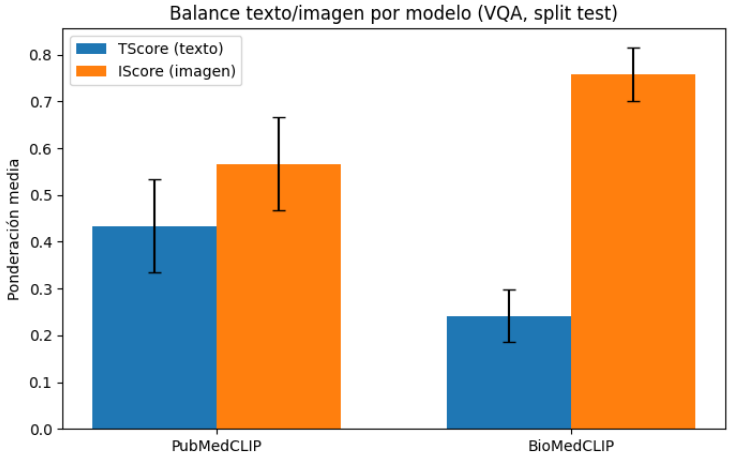


Fig. 3. Comparación de TScore e IScore por modelo en VQA (*split test*).

la correspondencia global entre imagen y texto, VQA requiere integrar imagen, pregunta y respuesta candidata. Por ello, el sesgo modal observado no debe atribuirse solo a la arquitectura, sino también a los datos de entrenamiento y a la tarea evaluada.

En conjunto, los resultados indican que métricas tradicionales como similitud imagen–texto o exactitud no reflejan por sí solas el grado de integración multimodal. Así, MM-SHAP y las métricas *TScore*, *IScore* y Δ aportan una caracterización complementaria del comportamiento de los modelos médicos de visión–lenguaje.

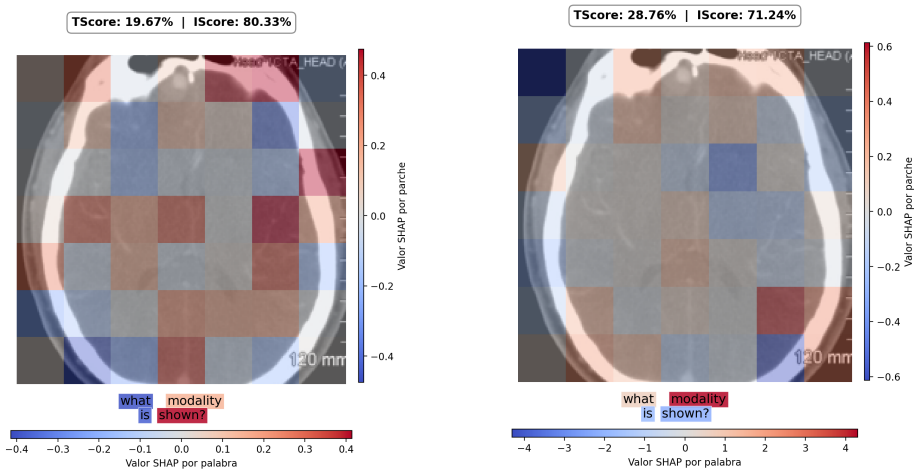
Desde una perspectiva práctica, este tipo de análisis puede apoyar la selección y auditoría de modelos médicos multimodales, al identificar casos en los que un desempeño alto oculta una dependencia excesiva de una sola modalidad.

6. Conclusiones

En este trabajo se presentó un marco de análisis para estudiar el balance multimodal en modelos médicos de visión–lenguaje mediante la aplicación experimental de MM-SHAP. El enfoque permitió cuantificar la contribución relativa de texto e imagen a nivel de muestra y de conjunto de datos, proporcionando una caracterización complementaria a las métricas tradicionales de desempeño.

Los resultados en ISA y VQA mostraron diferencias importantes entre modelos. **PubMedCLIP** presentó el balance más estable, mientras que **BioMedCLIP**, **RCLIP** y **WhyXRayCLIP** exhibieron distintos grados de sesgo hacia la modalidad visual. Asimismo, se observó que métricas como la similitud imagen–texto o la exactitud no reflejan por sí solas el grado de integración entre modalidades.

Como limitación, el análisis se restringe a dos tareas y a un conjunto acotado de modelos médicos tipo CLIP. Por ello, se plantea extender la evaluación a más



(a). PubMedCLIP **(b).** BioMedCLIP

Fig. 4. Heatmaps VQA — Muestra 0.

arquitecturas, conjuntos de datos y métodos de explicabilidad multimodal, así como estudiar la estabilidad de las explicaciones bajo distintas configuraciones.

Estos hallazgos evidencian la utilidad de incorporar métricas de explicabilidad multimodal en la evaluación de modelos médicos, especialmente en escenarios donde la interpretabilidad y la confianza son factores críticos. Además, este tipo de análisis puede apoyar la selección y auditoría de modelos al identificar dependencias excesivas hacia una sola modalidad.

References

1. Baltrusaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: A survey and taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(2), 423–443 (Feb 2019)
2. Ben Abacha, A., Hasan, S.A., Datla, V.V., Liu, J., Demner-Fushman, D., Müller, H.: VQA-Med: Overview of the Medical Visual Question Answering Task at ImageCLEF 2019. In: Working Notes of CLEF 2019. CEUR Workshop Proceedings, vol. 2380 (2019), `https://ceur-ws.org/Vol-2380/paper_272.pdf`
3. Beňová, I., Gregor, M., Gatt, A.: Cv-probes: Studying the interplay of lexical and world knowledge in visually grounded verb understanding. arXiv preprint arXiv:2409.01389 (2024)
4. Eslami, S., de Melo, G., Meinel, C.: PubMedCLIP: How Much Does CLIP Benefit Visual Question Answering in the Medical Domain? In: Findings of the Association for Computational Linguistics: EACL 2023. pp. 1181–1193. Association for Computational Linguistics (2023), `https://aclanthology.org/2023.findings-eacl.88`
5. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6904–6913 (2017)
6. Kaveh: RCLIP: CLIP Model Fine-Tuned on Radiology Images and Captions. https://huggingface.co/kaveh/rclip (2024), hugging Face model card. Consultado: 2026-05-07
7. Linden, T., De Jong, J., Lu, C., Kiri, V., Haeffs, K., Fröhlich, H.: An explainable multimodal neural network architecture for predicting epilepsy comorbidities based on administrative claims data. Frontiers in Artificial Intelligence 4, 610197 (2021), https://www.frontiersin.org/journals/artificial-intelligence
8. Liu, C., Jin, Y., Guan, Z., et al.: Visual–language foundation models in medicine. The Visual Computer (2024)
9. Liu, J., Zhang, Y., Chen, J., Xiao, J., Lu, Y., Landman, B., Yuan, Y., Yuille, A., Tang, Y., Zhou, Z.: Clip-driven universal model for organ segmentation and tumor detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21152–21164 (2023)
10. Parcalabescu, L., Frank, A.: Mm-shap: A performance-agnostic metric for measuring multimodal contributions in vision and language models and tasks. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics (2023)
11. Pelka, O., Koitka, S., Rückert, J., Nensa, F., Friedrich, C.M.: Radiology Objects in COntext (ROCO): A Multimodal Image Dataset. In: Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis. Lecture Notes in Computer Science, vol. 11043, pp. 180–189. Springer (2018)
12. Pellegrini, C., Keicher, M., Özsoy, E., Jiraskova, P., Braren, R., Navab, N.: Xplainer: From x-ray observations to explainable zero-shot diagnosis. arXiv preprint arXiv:2303.13391 (2023)
13. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8748–8763. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/radford21a.html
14. Tiu, E., Talius, E., Patel, P., Langlotz, C.P., Ng, A.Y., Rajpurkar, P.: Expert-level detection of pathologies from unannotated chest x-ray images via self-supervised learning. Nature Biomedical Engineering 6, 1399–1406 (2022)
15. Truhn, D., Eckardt, J.N., Ferber, D., Kather, J.N.: Large language models and multimodal foundation models for precision oncology. NPJ Precision Oncology 8(1), 72 (2024)
16. Wang, J., Mao, Y., Guan, N., Xue, C.J.: Shap-cat: A interpretable multi-modal framework enhancing wsi classification via virtual staining and shapley-value-based multimodal fusion. arXiv preprint arXiv:2410.01408 (2024)
17. YYUPenn: WhyXrayCLIP. https://huggingface.co/yyupenn/whyxrayclip (2024), hugging Face model card. Accessed: 2026-05-07
18. Zhang, S., Xu, Y., Usuyama, N., Xu, H., Bagga, J., Tinn, R., Preston, S., Rao, R., Wei, M., Valluri, N., Wong, C.C., Tupini, A., Wang, Y., Mazzola, M., Shukla, S., Liden, L., Gao, J., Crabtree, A., Piening, B., Bifulco, C., Lungren, M.P., Naumann, T., Poon, H.: Biomedclip: A multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. arXiv preprint arXiv:2303.00915 (2023)

19. Zhao, Z., Liu, Y., Wu, H., Li, Y., Wang, S., Teng, L., Liu, D., Cui, Z., Wang, Q., Shen, D.: Clip in medical imaging: A comprehensive survey. arXiv preprint arXiv:2312.07353 (2023)

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

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación