



Aplicación de microalgoritmos genéticos para la optimización de hiperparámetros en métodos de clasificación

Application of micro-genetic algorithms to the hyperparameter optimization in classification methods

Antonio Guerrero Juárez^{1*}, Abel García Nájera²,
Saúl Zapotecas Martínez³ y Karen Miranda⁴

¹Posgrado en Ciencias Naturales e Ingeniería, Universidad Autónoma Metropolitana Unidad Cuajimalpa
Av. Vasco de Quiroga 4871, Col. Santa Fe Cuajimalpa, 05348, Cuajimalpa de Morelos, Ciudad de México, México

²Departamento de Matemáticas Aplicadas y Sistemas, Universidad Autónoma Metropolitana Unidad Cuajimalpa
Av. Vasco de Quiroga 4871, Col. Santa Fe Cuajimalpa, 05348, Cuajimalpa de Morelos, Ciudad de México, México

³Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Tonantzintla, 72840, San Andrés Cholula, Puebla, México

⁴Departamento de Sistemas de Información y Comunicaciones, Universidad Autónoma Metropolitana Unidad Lerma
Av. Hidalgo Poniente 46, Col. La Estación, 52006, Lerma de Villada, Estado de México, México

*antonioguerrerojuarez@gmail.com

PALABRAS CLAVE: RESUMEN

Microalgoritmos genéticos, computación evolutiva, aprendizaje automático, métodos de clasificación

Este estudio propone el uso de microalgoritmos genéticos como una técnica de optimización de hiperparámetros para mejorar la precisión y eficiencia de algunos métodos de clasificación. Se evaluaron cuatro modelos sin optimización de hiperparámetros y, posteriormente, se aplicó un microalgoritmo genético diseñado específicamente para ajustar el valor de sus hiperparámetros. El objetivo fue analizar el impacto de esta técnica en la mejora de la precisión de los métodos. Los resultados demostraron que la implementación de microalgoritmos genéticos no solo aumentó significativamente la exactitud de los métodos de clasificación, sino que también redujo el tiempo de entrenamiento, mostrando una mejora en la eficiencia. Estos hallazgos sugieren que los microalgoritmos genéticos pueden ser una herramienta efectiva para optimizar el rendimiento de los métodos de clasificación y resolver problemas de clasificación con mayor exactitud y rapidez.

KEYWORDS: ABSTRACT

Micro-genetic algorithms, evolutionary computation, machine learning, classification methods.

This study proposes the use of micro-genetic algorithms as a hyperparameter optimization technique to improve the accuracy and efficiency of certain classification methods. Four models were initially evaluated without hyperparameter optimization, and subsequently, a specifically designed micro-genetic algorithm was applied to fine-tune their hyperparameters. The objective was to analyze the impact of this technique on improving classification accuracy. The results demonstrated that the implementation of micro-genetic algorithms not only significantly increased the accuracy of the classification methods but also reduced training time, indicating improved efficiency. These findings suggest that micro-genetic algorithms can be an effective tool for optimizing the performance of classification methods and solving classification problems with greater accuracy and speed.

• Recibido: 15 de noviembre de 2024 • Aceptado: 1 de abril de 2025 • Publicado en línea: 4 de octubre de 2025

1. INTRODUCCIÓN

En la última década, el aprendizaje automático se ha consolidado como una herramienta esencial en diversas áreas como la salud, la economía, la tecnología y las ciencias sociales. Esto se debe al incremento en la capacidad de procesamiento, la disponibilidad de grandes volúmenes de datos y los avances en algoritmos de aprendizaje. Uno de los desafíos en este campo es desarrollar modelos de predicción que sean no solo precisos sino también eficientes en términos computacionales [1].

La precisión y eficiencia de los métodos de clasificación dependen en gran medida de la adecuada configuración de sus hiperparámetros. El ajuste y optimización de estos hiperparámetros es un proceso fundamental que puede influir notablemente en el rendimiento del método. Sin embargo, las técnicas tradicionales de optimización suelen enfrentar limitaciones al aplicarse a problemas complejos y de alta dimensionalidad, ya que pueden quedarse atrapados en óptimos locales y requieren un alto costo computacional [2].

Una de las estrategias comúnmente empleadas para enfrentar estos desafíos es el uso de algoritmos evolutivos. Según Coello et al. [3], los algoritmos evolutivos, como los algoritmos genéticos, han demostrado ser altamente eficaces para resolver problemas de optimización. Estos algoritmos permiten manejar de manera eficiente múltiples criterios de evaluación, lo que es especialmente útil en la optimización de hiperparámetros de los métodos de clasificación.

Por otro lado, Eiben y Smith [4] han destacado la importancia de la diversidad en las poblaciones de los algoritmos evolutivos, una categoría en la que se incluyen los algoritmos genéticos y sus variantes, como los microalgoritmos genéticos que operan con poblaciones pequeñas. La diversidad en la población es fundamental para garantizar

una exploración efectiva del espacio de soluciones. En su trabajo, Eiben y Smith, señalan que una población diversa permite evitar que los algoritmos se queden atrapados en óptimos locales, lo que mejora la calidad de las soluciones finales. En el contexto de la optimización de hiperparámetros, esta característica es clave, ya que permite a los microalgoritmos genéticos explorar de manera detallada las configuraciones posibles, logrando modelos robustos y precisos.

Además, se ha demostrado que los algoritmos evolutivos, como los microalgoritmos genéticos, pueden reducir considerablemente el tiempo de convergencia sin sacrificar la calidad de las soluciones. La reducción en el tiempo de convergencia es especialmente valiosa cuando se trabaja con grandes conjuntos de datos o en contextos donde los recursos computacionales son limitados. Esto les da a los microalgoritmos genéticos una ventaja considerable frente a otras técnicas de optimización, haciendo que su uso sea cada vez más común en aplicaciones prácticas de aprendizaje automático.

En este contexto, los microalgoritmos genéticos emergen como una alternativa prometedora para la optimización de hiperparámetros en métodos de clasificación [5]. Estos algoritmos, inspirados en los principios de la evolución natural, son capaces de explorar eficientemente el espacio de soluciones y encontrar óptimos globales con menor costo computacional. Se caracterizan por utilizar poblaciones pequeñas y ciclos de vida cortos, lo que permite una convergencia rápida sin sacrificar la calidad de la solución.

El objetivo de este estudio es evaluar el impacto de los microalgoritmos genéticos en la optimización de hiperparámetros de algoritmos de clasificación y determinar si su aplicación puede mejorar significativamente la precisión y eficiencia estos [6]. Para ello, se seleccionaron cuatro algoritmos de

clasificación ampliamente utilizados: árboles de decisión, K vecinos más cercanos, perceptrón y descenso por gradiente estocástico. Se realizaron experimentos comparando el rendimiento de estos algoritmos antes y después de aplicar la optimización de hiperparámetros con microalgoritmos genéticos.

Los resultados obtenidos demuestran que la aplicación de microalgoritmos genéticos mejora notablemente la precisión de los métodos de clasificación. Esto sugiere que los microalgoritmos genéticos tienen un alto potencial para resolver problemas de optimización en aprendizaje automático.

En la siguiente sección se presenta una revisión de trabajos previos relacionados con la optimización de hiperparámetros en métodos de clasificación. Después, se describen los algoritmos y la metodología utilizados. Posteriormente, se detallan los experimentos realizados y se analizan los resultados obtenidos. Finalmente, se ofrecen algunas conclusiones y se proponen posibles líneas de investigación futura.

2. TRABAJO PREVIO

La investigación sobre el uso de la computación evolutiva aplicada a problemas de predicción ha sido extensa y abarca diversas áreas. Por ejemplo, Smith SL y Cagnoni S [7] examinaron cómo las técnicas de computación evolutiva, como los algoritmos genéticos y la optimización de colonias de hormigas, pueden mejorar la predicción y el diagnóstico en el ámbito de la salud. Estas técnicas se utilizaron para optimizar modelos predictivos en la detección temprana de enfermedades y en la personalización de tratamientos, mostrando mejoras significativas tanto en la precisión como en la eficiencia de los modelos.

Por su parte, un estudio relevante de Jin-Young Kim y Sung-Bae Cho [8] aborda la optimización de hiperparámetros y la

evolución de arquitecturas neuronales. En este trabajo, los autores implementaron técnicas evolutivas para optimizar hiperparámetros en modelos de aprendizaje profundo, demostrando que estos métodos pueden superar a las técnicas tradicionales en problemas de clasificación y regresión. Sus hallazgos resaltan cómo los algoritmos evolutivos pueden explorar y optimizar arquitecturas de redes neuronales, mejorando significativamente su rendimiento.

Suganuma, Shirakawa & Nagao [9] abordaron la optimización de hiperparámetros en modelos de aprendizaje profundo mediante algoritmos evolutivos. En su estudio, implementaron un método de optimización evolutiva que resultó en mejoras significativas en la precisión de modelos aplicados a conjuntos de datos de imágenes, como CIFAR-10 y CIFAR-100. Los resultados demostraron que los métodos evolutivos no solo superaron a la búsqueda en cuadrícula y la búsqueda aleatoria en problemas de alta dimensionalidad, sino que también redujeron los costos computacionales. Este trabajo enfatiza el potencial de los algoritmos evolutivos para explorar el espacio de hiperparámetros de manera eficiente, logrando modelos precisos y optimizados.

En otro estudio, Shchygol [10] comparó diversas técnicas evolutivas, como los algoritmos genéticos, la estrategia evolutiva con adaptación de la matriz de covarianza (CMA-ES) y el optimizador de campo de fuerza de Monte Carlo (MCFF), para la optimización de hiperparámetros en ReaxFF. Los resultados sugieren que los algoritmos genéticos son eficaces para evitar mínimos locales, mientras que CMA-ES logra los menores errores en la mayoría de los casos.

Murthy [11] investigó las estrategias evolutivas para la optimización de hiperparámetros en modelos de aprendizaje profundo. El estudio concluyó que los algoritmos evolutivos mejoran

significativamente el rendimiento de estos modelos, al optimizar de manera eficiente los hiperparámetros en espacios de búsqueda complejos.

Venske et al. [12] demostraron que la aplicación de la evolución diferencial adaptativa en el problema de predicción de la estructura de proteínas es altamente efectiva. Este enfoque permitió una búsqueda eficiente en el espacio de conformaciones, resultando en predicciones precisas y rápidas de las estructuras proteicas, lo cual es clave para el desarrollo de nuevas terapias y medicamentos.

Por último, Sorano et al. [13] investigó la optimización multiobjetivo de hiperparámetros en modelos de aprendizaje automático aplicados a la salud, utilizando algoritmos evolutivos. Los autores mostraron que este enfoque no solo mejora la precisión de los modelos, sino que también permite equilibrar el recall, ofreciendo a los tomadores de decisiones un conjunto de soluciones óptimas de Pareto, lo cual es relevante en aplicaciones médicas donde es crucial mantener altos niveles de rendimiento en varias métricas simultáneamente.

A pesar de los avances significativos logrados mediante técnicas de computación evolutiva, todavía existe un margen considerable para la optimización y adaptación de estos métodos a problemas específicos. En particular, la aplicación de microalgoritmos genéticos (MAG) para la optimización de hiperparámetros en métodos de clasificación ofrece una vía prometedora para superar las limitaciones de los métodos tradicionales.

Esta investigación busca llenar el vacío en la literatura existente al proporcionar una evaluación y comparación del rendimiento de los MAG en comparación con otros enfoques de optimización. Además, los hallazgos de esta investigación tienen el potencial de aplicarse en una variedad de contextos, desde la salud digital hasta la agricultura e

industria, contribuyendo así al avance del conocimiento y la práctica en estas áreas.

3. ALGORITMOS DE CLASIFICACIÓN Y MICROALGORITMOS GENÉTICOS

En esta sección se describen los algoritmos de clasificación utilizados en este estudio y se presenta el enfoque de optimización de hiperparámetros mediante microalgoritmos genéticos. El objetivo es proporcionar una descripción clara de estas herramientas y cómo su combinación puede mejorar el rendimiento de los modelos de predicción.

3.1. Algoritmos de clasificación

Los algoritmos de clasificación son fundamentales en el aprendizaje automático supervisado, ya que permiten asignar etiquetas o categorías a nuevas instancias basadas en un conjunto de datos de entrenamiento. El proceso de clasificación consiste en encontrar un modelo o función que, a partir de los datos de entrada (características), pueda predecir una etiqueta o clase asociada. Estos algoritmos son ampliamente utilizados en diversas aplicaciones, como diagnóstico médico, detección de fraude, reconocimiento de imágenes y clasificación de texto, entre muchas otras.

La calidad y precisión de los algoritmos de clasificación dependen de múltiples factores, entre ellos la naturaleza de los datos, la selección de características relevantes, la capacidad del modelo para generalizar sobre nuevos datos y, en muchos casos, la optimización de sus hiperparámetros. Esta optimización es crucial para mejorar el rendimiento y reducir tanto el sesgo como la varianza del modelo, logrando un equilibrio entre ajuste excesivo y subajuste. A continuación, se describen los cuatro algoritmos de clasificación seleccionados para este estudio:

Árbol de decisión

El árbol de decisión es un modelo predictivo que utiliza un árbol como estructura para representar las decisiones y sus posibles consecuencias. Cada nodo interno representa una prueba en una característica, cada rama representa el resultado de la prueba y cada hoja representa una etiqueta de clase. Los árboles de decisión son fáciles de interpretar y pueden manejar datos categóricos y numéricos. Sin embargo, pueden ser propensos al sobreajuste si no se controlan adecuadamente sus hiperparámetros, como la profundidad máxima del árbol [14].

K vecinos más cercanos

El algoritmo K vecinos más cercanos es un método basado en instancias que clasifica un punto de datos en función de cómo se clasifican sus vecinos más cercanos en el espacio de características. Para determinar la clase de una nueva instancia, el algoritmo calcula la distancia (generalmente euclidiana) a los K puntos de datos más cercanos en el conjunto de entrenamiento y asigna la clase más frecuente entre ellos. La elección del valor de K y la métrica de distancia son hiperparámetros críticos que afectan el rendimiento del modelo [15].

Perceptrón

El perceptrón es el modelo más básico de una red neuronal artificial, utilizado para problemas de clasificación binaria. Es un clasificador lineal que toma una decisión basándose en una combinación lineal de las características de entrada. El algoritmo ajusta los pesos asignados a cada característica mediante un proceso iterativo, minimizando el error de clasificación. Aunque es sencillo y eficiente, el perceptrón solo es capaz de resolver problemas linealmente separables [16].

Descenso por gradiente estocástico

El descenso por gradiente estocástico es un método de optimización utilizado para ajustar los hiperparámetros de un modelo de aprendizaje, minimizando una función de pérdida. En el contexto de la clasificación, se utiliza para entrenar modelos lineales, como regresión logística y máquinas de vectores de soporte. También actualiza los hiperparámetros para cada instancia de entrenamiento, lo que lo hace eficiente para conjuntos de datos grandes. Los hiperparámetros clave incluyen la tasa de aprendizaje y el tipo de regularización [17].

3.2. Algoritmos genéticos

Los algoritmos genéticos (AG) son métodos de búsqueda inspirados en los procesos de selección natural y genética presentes en los seres vivos. Estos algoritmos utilizan una población de soluciones candidatas que evolucionan a lo largo de varias generaciones, aplicando operadores genéticos como la selección, el cruzamiento y la mutación. A continuación, se presenta la Figura 1, la cual ilustra el ciclo de un AG, mostrando los pasos clave que se llevan a cabo en cada generación.

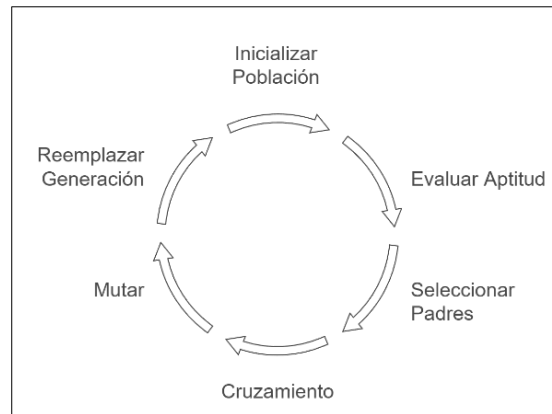


Figura 1. Ciclo de un algoritmo genético.

Los pasos generales de un AG son:

- 1) Inicialización: Se genera una población inicial de soluciones aleatorias.
- 2) Evaluación: Cada individuo es evaluado

mediante una función de aptitud que refleja qué tan buena es la solución en relación con el problema.

- 3) Selección: Se seleccionan los individuos más aptos, es decir, aquellos con mejores valores de aptitud, para reproducirse.
- 4) Cruzamiento: Los individuos seleccionados se combinan para producir descendientes, mediante un proceso que intercambia parte de su información genética.
- 5) Mutación: Se aplican cambios aleatorios a algunos descendientes para mantener la diversidad genética dentro de la población.
- 6) Reemplazo: Se forma una nueva generación combinando los descendientes o los individuos más aptos de la generación anterior.

El objetivo de este proceso es ir refinando la población para que las soluciones sean cada vez mejores con respecto a la función de aptitud. A medida que el ciclo se repite, las soluciones convergen hacia un óptimo global o cercano al óptimo, dependiendo del problema que se esté resolviendo.

3.3. Microalgoritmos genéticos

Los microalgoritmos genéticos son una variante de los algoritmos genéticos (AG) tradicionales, diseñados para operar con poblaciones pequeñas y ciclos de vida cortos. Este enfoque permite una convergencia más rápida y eficiente en comparación con los AG estándar, reduciendo el tiempo de ejecución sin sacrificar la calidad de las soluciones. Al trabajar con poblaciones reducidas, los microalgoritmos genéticos son especialmente útiles en problemas de optimización que requieren una rápida adaptación o que se enfrentan a limitaciones de recursos computacionales, como la optimización de hiperparámetros en modelos de aprendizaje automático [18].

Una de las principales ventajas de los microalgoritmos genéticos es su capacidad para escapar de mínimos locales y acelerar la búsqueda de soluciones óptimas en espacios de búsqueda complejos. Debido a su diseño

compacto y dinámico, pueden mantener una diversidad genética suficiente para evitar la convergencia prematura, que es un problema común en AG con poblaciones grandes y tiempos de evolución prolongados. Además, su implementación requiere menos memoria y menos procesamiento, lo que los hace adecuados para aplicaciones en tiempo real o en sistemas con restricciones de hardware.

Los microalgoritmos genéticos modifican el enfoque tradicional al utilizar poblaciones pequeñas (generalmente entre 5 y 10 individuos) y un número reducido de generaciones. Esto reduce el tiempo de cómputo y evita la convergencia prematura a óptimos locales. Los MAG suelen incluir mecanismos para reintroducir diversidad en la población, como reinicios periódicos o mutaciones más agresivas. En este estudio, los MAG se utilizaron para optimizar los hiperparámetros de los algoritmos de clasificación mencionados. El proceso consistió en la codificación de los hiperparámetros, donde los hiperparámetros de cada modelo se representaron como genes en un cromosoma. Por ejemplo, para el algoritmo K vecinos más cercanos, tanto el valor de K como el tipo de métrica de distancia fueron incluidos en el cromosoma. Asimismo, se definió la función de aptitud, la cual se basó en la precisión del modelo en un conjunto de validación, siendo el objetivo maximizar dicha precisión.

El modelo con el mejor conjunto de hiperparámetros encontrados al final del proceso —aquellos que lograron la mayor precisión en las evaluaciones de las generaciones anteriores— fue entrenado nuevamente y su rendimiento evaluado en un conjunto de prueba independiente, asegurando así la validez y generalización de los resultados obtenidos.

4. ESTUDIO EXPERIMENTAL

En esta sección se presenta cómo se realizó el estudio experimental, el cual se enfocó en

evaluar el rendimiento de diversos algoritmos de clasificación al optimizar sus hiperparámetros mediante un MAG. El objetivo fue determinar si la optimización de parámetros utilizando un MAG mejora significativamente la precisión y la eficiencia de estos modelos de predicción en comparación con sus versiones sin optimización. A continuación, se describen los conjuntos de datos utilizados, los recursos computacionales empleados y la configuración de los experimentos realizados.

4.1. Conjuntos de datos utilizados

Para llevar a cabo los experimentos, se seleccionaron seis conjuntos de datos de diferentes dominios, todos disponibles públicamente en el repositorio UCI Machine Learning Repository [19]. Estos conjuntos de datos fueron elegidos debido a su diversidad, abarcando distintos campos, lo que permite evaluar la capacidad de generalización y la eficacia tanto de los algoritmos de clasificación como de la optimización mediante microalgoritmos genéticos.

- 1 Diabetes (CD1): Contiene 768 instancias con 8 atributos médicos, utilizados para predecir la presencia de diabetes. Este conjunto de datos es fundamental para estudios en el área de salud y diagnóstico médico.
- 2 Ingresos del censo (Adult) (CD2): Consta de 48,842 instancias y 14 atributos demográficos y económicos. Se emplea para predecir si el ingreso anual de una persona supera los \$50,000, siendo útil en estudios socioeconómicos.
- 3 Cáncer de mama (Breast Cancer Wisconsin) (CD3): Incluye 569 instancias con 30 atributos que describen características celulares. Se utiliza para clasificar tumores como benignos o malignos, siendo clave en la detección temprana del cáncer de mama.
- 4 Calidad del vino (CD4): Contiene 4,898 instancias de vino blanco y 1,599 de vino tinto, con 11 atributos químicos. Se emplea para predecir la calidad del vino en una

escala de 0 a 10, relevante para la industria vitivinícola.

- 5 Enfermedades del corazón (CD5): Consiste en 303 instancias con 13 atributos clínicos. Se utiliza para diagnosticar enfermedades cardíacas, aportando valor en cardiología preventiva.
- 6 Calidad del aire (CD6): Comprende 9,358 instancias con 15 atributos relacionados con contaminantes atmosféricos y condiciones meteorológicas. Se emplea para predecir la calidad del aire, siendo esencial en estudios medioambientales.

La diversidad de estos conjuntos de datos asegura que los experimentos no estén sesgados hacia un solo tipo de problema, proporcionando un marco más robusto para analizar cómo la optimización de los hiperparámetros afecta el rendimiento en diferentes situaciones. Algunos de los conjuntos incluyen problemas de clasificación binaria y multiclase, con características numéricas y categóricas, lo que proporciona una rica variedad de retos para los algoritmos de clasificación. Al trabajar con datos heterogéneos, se puede determinar si los beneficios observados mediante la optimización de hiperparámetros son consistentes a través de diversos escenarios o si se limitan a dominios específicos.

4.2. Recursos computacionales

Los experimentos se realizaron utilizando el lenguaje de programación Python, aprovechando sus bibliotecas especializadas en aprendizaje automático y optimización. En la Tabla 1 se detalla el software utilizado.

Tabla 1. Software utilizado en el estudio experimental.

Software	Descripción
Scikit-learn	Para implementar y evaluar los algoritmos de clasificación.
NumPy y Pandas	Para manipulación y análisis de datos.
Matplotlib y Seaborn	Para la visualización de resultados.
SciPy	Para realizar pruebas estadísticas y análisis adicionales.

Se realizaron ejecuciones preliminares en Google Colab para analizar el comportamiento de los algoritmos y sus características. Estas ejecuciones iniciales no constituyeron pruebas formales, sino que se emplearon para ajustar y preparar los modelos para los experimentos posteriores. Los experimentos y pruebas formales se llevaron a cabo en un equipo cuyas especificaciones se muestran en la Tabla 2.

Tabla 2. Especificaciones del equipo de experimentación.

Hardware	Especificaciones
Procesador	Intel Core i7 12th Gen 2.30 GHz
Memoria RAM	64 GB
Almacenamiento	SSD de 512 GB
Sistema Operativo	Windows 11

Esta configuración permitió ejecutar los experimentos de manera eficiente y manejar los conjuntos de datos de mayor tamaño sin problemas de rendimiento.

4.3. Configuración experimental

Para evaluar el impacto de la optimización con MAG, se utilizaron los algoritmos de clasificación previamente seleccionados. Estos modelos fueron sometidos al proceso de optimización de hiperparámetros con el fin de analizar cómo esta técnica mejora su rendimiento en comparación con su configuración estándar. En la Tabla 3 se detallan los hiperparámetros de cada método de clasificación.

4.4. Configuración del microalgoritmo genético

El MAG se configuró como se muestra en la Tabla 4 para cada conjunto de datos y método de clasificación.

4.5. Procedimiento experimental

- Preprocesamiento de datos:
 - Los datos de los seis conjuntos seleccionados fueron preprocesados para garantizar su compatibilidad con los algoritmos de clasificación. Las variables numéricas fueron estandarizadas,

ajustándolas para que tuvieran media cero y desviación estándar uno, lo cual permitió eliminar problemas derivados de diferencias en escalas entre atributos.

- Las variables categóricas fueron transformadas utilizando técnicas como la codificación *one-hot* o *label encoding*, dependiendo de la naturaleza de los datos. Esto convirtió las variables no numéricas en formatos adecuados para el procesamiento por los algoritmos de aprendizaje automático.
- Se manejaron los valores faltantes aplicando técnicas de imputación (por ejemplo, utilizando la media o moda) o eliminando registros incompletos cuando fue apropiado, garantizando así la integridad de los conjuntos de datos antes de su uso en los experimentos.

Tabla 3. Hiperparámetros de los métodos de clasificación considerados en el estudio experimental.

Algoritmo	Hiperparámetros
Árbol de decisión	<ul style="list-style-type: none"> • max_depth: Profundidad máxima del árbol • min_samples_split: Número mínimo de muestras para dividir un nodo • min_samples_leaf: Número mínimo de muestras en una hoja • criterion: Función para medir la calidad de una división (gini, entropy)
K vecinos más cercanos	<ul style="list-style-type: none"> • n_neighbors: Número de vecinos considerados • weights: Función de ponderación (uniform, distance) • algorithm: Algoritmo para computar vecinos (ball_tree, kd_tree, brute) • leaf_size: Tamaño de la hoja para los árboles • p: Potencia del parámetro de Minkowski (1 para Manhattan, 2 para Euclidiana)
Perceptrón	<ul style="list-style-type: none"> • penalty: Tipo de regularización (None, l2, l1, elasticnet) • alpha: Parámetro de regularización • max_iter: Número máximo de iteraciones
Descenso por gradiente estocástico	<ul style="list-style-type: none"> • penalty: Tipo de regularización • alpha: Parámetro de regularización • max_iter: Número máximo de iteraciones • learning_rate: Estrategia de ajuste de la tasa de aprendizaje • eta0: Tasa de aprendizaje inicial

- División de conjuntos:
 - Los datos fueron divididos en un conjunto de entrenamiento (60%) y un conjunto de

prueba (40%). La división se realizó de manera estratificada para asegurar que las proporciones de las clases se mantuvieran consistentes en ambos conjuntos. Esto permitió una evaluación más precisa del rendimiento de los modelos.

Tabla 4. Configuración general del microalgoritmo genético.

Parámetro	Valor
Tamaño de la población	10 individuos
Número de generaciones	10
Número de ejecuciones independientes	30
Tasa de mutación	0.1
Operadores genéticos	Cruza: Uniforme Mutación: Aleatoria con probabilidad de 0.1
Selección Natural	Elitista

- Entrenamiento sin optimización:
 - Se entrenaron los modelos de clasificación previamente seleccionados, utilizando los valores predeterminados de sus hiperparámetros proporcionados por la biblioteca scikit-learn.
 - Una vez entrenados, se evaluó el rendimiento de estos modelos en el conjunto de prueba, registrando la precisión y otras métricas relevantes para su posterior comparación.
- Optimización con MAG:
 - Para cada algoritmo de clasificación, se generó una población inicial con combinaciones aleatorias de sus hiperparámetros dentro de rangos predefinidos específicos para cada modelo. A través de 10 generaciones, se aplicaron operadores genéticos para mutar algunos de los hiperparámetros, manteniendo la diversidad en la población.
 - La evaluación de los individuos se realizó entrenando el modelo con el conjunto de entrenamiento y evaluando su rendimiento en el conjunto de prueba. El mejor conjunto de hiperparámetros fue seleccionado basado en su rendimiento promedio, permitiendo así el entrenamiento del modelo optimizado.
- Entrenamiento con hiperparámetros optimizados:

- Utilizando los mejores valores de los hiperparámetros obtenidos a través del proceso de optimización con el MAG, se volvió a entrenar cada uno de los modelos de clasificación.
- Los modelos optimizados fueron evaluados en el conjunto de prueba y se registraron los resultados de precisión y otras métricas clave para compararlos con los modelos entrenados sin optimización.
- Análisis estadístico:
 - Una vez obtenidos los resultados de los modelos con y sin optimización, se aplicó la prueba estadística de Mann-Whitney U para comparar los desempeños en términos de precisión y tiempo de entrenamiento. Esta prueba no paramétrica permitió determinar si las diferencias observadas entre los modelos optimizados y no optimizados eran estadísticamente significativas.
 - El análisis se realizó sobre los resultados de las 30 ejecuciones de cada modelo, asegurando robustez en las conclusiones estadísticas. Además, se calcularon las desviaciones estándar de los resultados para evaluar la consistencia en el rendimiento de los modelos.
- Repetición de experimentos:
 - Todo el procedimiento se repitió 30 veces para cada algoritmo y conjunto de datos, variando las divisiones de entrenamiento y prueba mediante diferentes semillas aleatorias (random_state), con el fin de obtener promedios y desviaciones estándar confiables de los resultados. Estos resultados fueron almacenados en un archivo para su posterior análisis y comparación.

5. RESULTADOS EXPERIMENTALES

En esta sección, se presentan los resultados obtenidos de la evaluación de cuatro algoritmos de clasificación (árbol de decisión, K vecinos más cercanos, perceptrón y descenso por gradiente) en seis conjuntos de

datos diferentes. Los experimentos comparan la precisión promedio de cada modelo antes y después de aplicar la optimización de hiperparámetros mediante un microalgoritmo genético. Los resultados resumidos se muestran en la Tabla 5. Esta tabla presenta los valores de precisión promedio junto con su desviación estándar para cada modelo, tanto con optimización

(MAG) como sin optimización (No MAG). Los valores subrayados representan las mejores precisiones obtenidas para cada conjunto de datos. Si la desviación estándar está en negritas, indica una diferencia estadísticamente significativa entre las versiones con y sin optimización, la cual se obtuvo de la prueba estadística de Mann-Whitney U realizada.

Tabla 5. Resultados de precisión media y desviación estándar con y sin MAG.

Datos	Árbol de decisión		K vecinos		Perceptrón		Descenso por gradiente	
	No MAG	MAG	No MAG	MAG	No MAG	MAG	No MAG	MAG
CD1	0.9494	0.9708	0.9566	0.9569	0.9334	0.9484	0.9579	0.9591
	± 0.0009	± 0.0008	± 0.0011	± 0.0025	± 0.0133	± 0.0077	± 0.0014	± 0.0016
CD2	0.8072	0.8497	0.8173	0.8282	0.7951	0.8064	0.8339	0.8463
	± 0.0038	± 0.0040	± 0.0036	± 0.0040	± 0.0143	± 0.0195	± 0.0052	± 0.0031
CD3	0.9211	0.9395	0.9633	0.9675	0.9610	0.9696	0.9630	0.9727
	± 0.0133	± 0.0175	± 0.0119	± 0.0107	± 0.0144	± 0.0109	± 0.011	± 0.0122
CD4	0.8649	0.8814	0.8680	0.9024	0.8417	0.8691	0.8568	0.8726
	± 0.0191	± 0.0095	± 0.0079	± 0.0085	± 0.0237	± 0.0086	± 0.0201	± 0.0110
CD5	0.7230	0.7927	0.8157	0.8501	0.7658	0.8266	0.7849	0.8417
	± 0.0405	± 0.0313	± 0.0229	± 0.0192	± 0.0605	± 0.0319	± 0.0364	± 0.0291
CD6	0.9456	0.9503	0.9556	0.9612	0.9456	0.9575	0.9559	<u>0.9646</u>
	± 0.0131	± 0.0162	± 0.0106	± 0.0104	± 0.0187	± 0.0122	± 0.0124	± 0.0124

Como podemos observar en la Tabla 5, en general, la optimización mejora significativamente la precisión en todos los algoritmos, destacándose los modelos árbol de decisión y perceptrón. Estos modelos experimentan un incremento de precisión notable, especialmente en los conjuntos de datos CD4 y CD5. Asimismo, la reducción en la desviación estándar indica una mayor consistencia en los resultados tras la optimización, lo que es determinante en aplicaciones donde la fiabilidad es esencial. Un caso destacado es el árbol de decisión en el conjunto de datos CD1, donde la precisión pasó de 0.9494 ± 0.0009 a 0.9708 ± 0.0008 , evidenciando no solo una mejora en el rendimiento sino también una mayor estabilidad en los resultados.

Es importante resaltar que, aunque todos los modelos se benefician de la optimización, la magnitud de la mejora varía según el algoritmo y el conjunto de datos. Por ejemplo, el modelo K vecinos más cercanos mostró un incremento particularmente notable en el conjunto de datos CD5, donde

la precisión promedio aumentó de 0.8157 ± 0.0229 a 0.8501 ± 0.0192 , indicando una mejora significativa en la capacidad del modelo para generalizar. Por otro lado, en el caso del descenso por gradiente, se observa una mejora más consistente y uniforme en todos los conjuntos de datos, destacando su robustez frente a la optimización. Esto sugiere que el impacto del microalgoritmo genético no solo mejora los resultados globales, sino que también ayuda a equilibrar el rendimiento entre los diferentes escenarios evaluados.

A continuación, en la Figura 2, se visualiza la mejora promedio en precisión para cada modelo tras aplicar la optimización con MAG. Este gráfico de líneas destaca la variabilidad en las mejoras para los diferentes modelos y conjuntos de datos. La Figura 2 muestra la mejora en precisión obtenida tras la optimización para cada algoritmo y conjunto de datos. Se observa que el árbol de decisión y el perceptrón son los modelos que más se benefician del proceso de optimización, alcanzando mejoras de hasta 0.05 puntos en

algunos conjuntos, como CD5. Aunque K vecinos más cercanos y el descenso por gradiente también presentan mejoras, estas son más modestas en comparación con los otros modelos. En general, se observa que los conjuntos CD4 y CD5 muestran un mayor impacto de la optimización, lo que indica que el efecto de los MAG depende de las características de cada conjunto de datos.

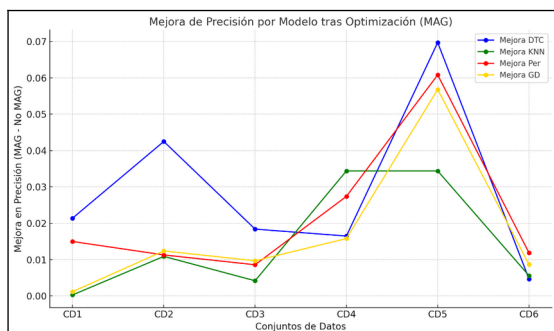


Figura 2. Mejora de precisión por modelo tras la optimización con MAG.

Posteriormente, se presentan análisis específicos sobre la relación entre la precisión y el tiempo de entrenamiento. Las Figuras 3, 4 y 5 muestran gráficos de dispersión que comparan el rendimiento de diferentes algoritmos, con y sin la optimización mediante MAG. En estas figuras, se observa cómo la optimización con MAG reduce el tiempo de entrenamiento en algunos casos, a la vez que mejora la precisión. En los gráficos, el eje vertical representa la precisión, mientras que el eje horizontal muestra el tiempo de entrenamiento.

En el caso del Perceptrón en CD1, véase la Figura 3, se puede observar que la versión optimizada con MAG logra una mayor precisión, con tiempos de entrenamiento similares o incluso inferiores. Esto sugiere que los MAG logran una configuración de hiperparámetros más eficiente, que mejora el rendimiento sin aumentar los costos computacionales.

La Figura 4 muestra la comparación del K vecinos más cercanos en CD5 y refleja un

comportamiento similar, donde la versión optimizada alcanza una precisión superior a la del modelo normal en casi todos los casos. Sin embargo, el impacto en el tiempo de entrenamiento es más variable, lo que indica que en algunos casos la optimización puede conllevar un costo computacional mayor.

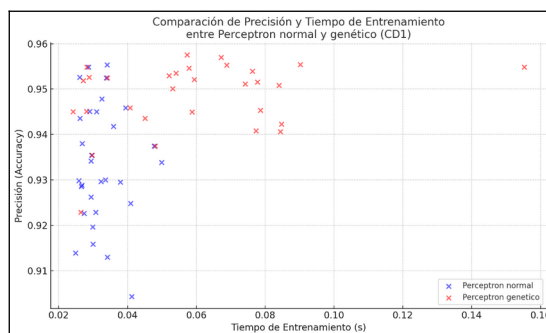


Figura 3. Comparación entre Perceptrón normal y genético en el conjunto de datos CD1.

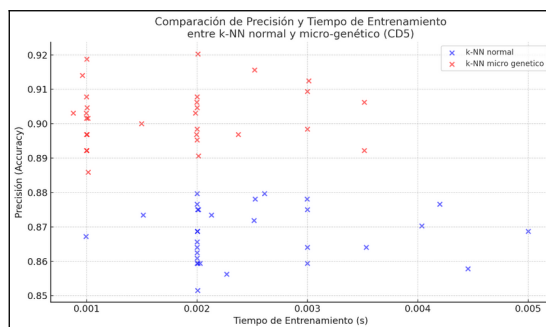


Figura 4. Comparación entre K vecinos más cercanos normal y algoritmo microgenético en el conjunto de datos CD5.

En la Figura 5 se puede observar que la versión optimizada del árbol de decisión en CD2 no solo mejora la precisión, sino que también reduce significativamente el tiempo de entrenamiento, lo que subraya la eficiencia de los MAG en este caso. Por último, en la Figura 6 se presenta un gráfico de diagrama de caja que permite comparar visualmente la dispersión de la precisión entre las versiones con y sin optimización para cada modelo en los distintos conjuntos de datos. El gráfico muestra cómo la optimización no solo mejora la precisión, sino que también puede reducir la variabilidad en los resultados, especialmente para los modelos árbol de decisión y

perceptrón.

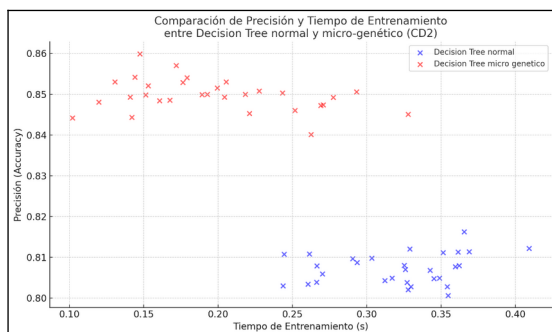


Figura 5. Comparación entre árbol de decisión normal y algoritmo microgenético en el conjunto de datos CD2.

En la Figura 6, se observa que la precisión de cada método se representa mediante diagramas de caja y bigote, donde el eje Y indica el valor de precisión, y el eje X muestra los algoritmos evaluados: árbol de decisión (DTC), K vecinos más cercanos (KNN), perceptrón (Per), y gradiente descendente (GD), cada uno con y sin optimización (MAG y No_MAG). Las cajas reflejan la variabilidad de la precisión en los experimentos, mientras que la línea central representa la mediana. En general, se observa una mejora significativa tanto en la precisión como en la reducción de la variabilidad de los resultados. Los algoritmos que más se benefician de la optimización son el árbol de decisión (DTC) y el perceptrón (Per), especialmente en los conjuntos de datos CD1, CD4 y CD5.

En el conjunto de datos CD1, el árbol de decisión optimizado presenta una notable mejora en precisión, reduciendo además la dispersión de los resultados en comparación con su versión no optimizada. Este aumento en precisión y consistencia es clave para aplicaciones que dependen de decisiones precisas y replicables. El perceptrón también muestra una mejora significativa en precisión, lo que sugiere que la optimización mediante MAG es altamente efectiva para estos algoritmos en este conjunto, brindando una ventaja competitiva en el procesamiento de datos complejos.

El conjunto de datos CD4 refuerza esta tendencia, donde ambos algoritmos, árbol de decisión y perceptrón, muestran incrementos importantes en precisión tras la optimización. Además, la variabilidad en los resultados disminuye, lo que indica una mayor consistencia en las predicciones de los modelos optimizados. Esto es particularmente relevante para aplicaciones donde se requiere estabilidad y fiabilidad en los resultados, ya que una menor variabilidad permite una toma de decisiones más confiable y precisa en contextos críticos.

En CD5, se observa una de las mejoras más pronunciadas, especialmente en el árbol de decisión. Los resultados optimizados no solo muestran una mayor precisión, sino que también la dispersión de los resultados se reduce considerablemente, lo cual refuerza la estabilidad del modelo tras la optimización. El perceptrón también experimenta una mejora notable, aunque su variabilidad sigue siendo mayor en comparación con otros conjuntos de datos, lo que podría sugerir una sensibilidad particular a las características del conjunto CD5.

6. CONCLUSIONES

Los resultados obtenidos en esta investigación confirman la eficacia de los microalgoritmos genéticos en la optimización de hiperparámetros para algoritmos de clasificación. En todos los conjuntos de datos analizados, la aplicación de los MAG resultó en una mejora significativa de la precisión, especialmente en los algoritmos árbol de decisión y perceptrón, que mostraron aumentos promedio de precisión de hasta 5 puntos porcentuales. La optimización mediante MAG no solo mejoró la precisión, sino que también logró reducir el tiempo de entrenamiento en muchos casos, como se observó en la Figura 3, 4 y 5 de comparación de tiempo y precisión.

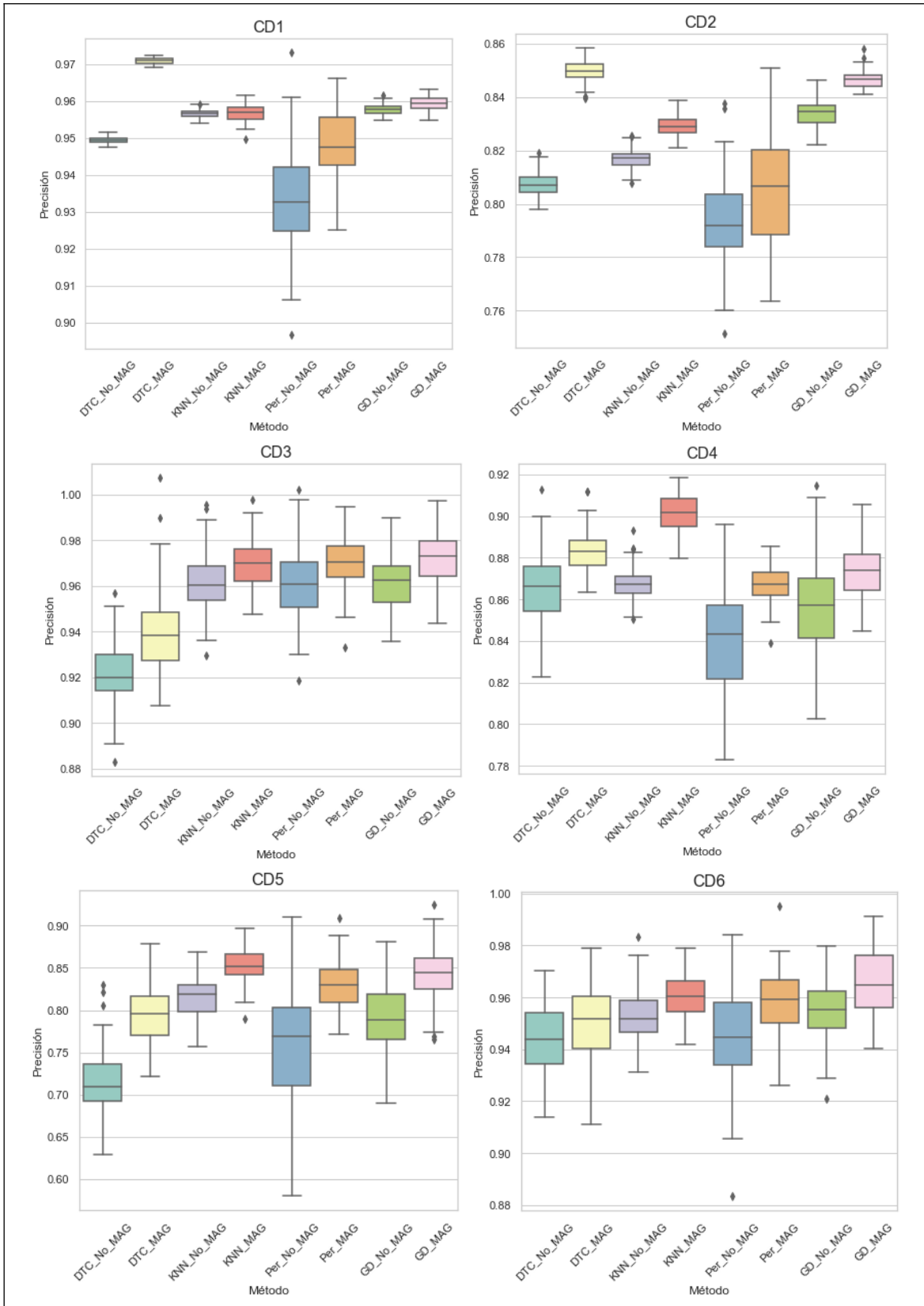


Figura 6. Comparación de precisión con y sin MAG mediante diagrama de caja y bigotes.

Por otro lado, K vecinos más cercanos, aunque también mostró mejora, no fue tan pronunciada como en el caso de los otros modelos. La diferencia en precisión entre las versiones con y sin MAG fue menos significativa en varios conjuntos de datos. Por ejemplo, en CD1, la precisión apenas varió de 0.9566 a 0.9569, lo que representa un incremento del 0.031%. Sin embargo, en otros casos como CD5, la mejora fue más notable, pasando de 0.8157 a 0.8501, lo que representa un incremento del 4.22%. Esto sugiere que el impacto del MAG en K vecinos más cercanos depende en gran medida de las características del conjunto de datos.

En términos de tiempo de entrenamiento, se observó una reducción importante en los modelos optimizados. Por ejemplo, en el conjunto de datos CD2, el árbol de decisión normal requería tiempos de entrenamiento significativamente mayores en comparación con su versión optimizada, lo que refuerza la idea de que los MAG no solo mejoran la precisión, sino que también permiten una búsqueda más eficiente en el espacio de hiperparámetros. Esta combinación de mejoras en precisión y reducción de tiempos tiene importantes implicaciones prácticas, sobre todo en aplicaciones donde la rapidez es esencial.

Además de mejorar el rendimiento global, los MAG demostraron ser eficaces en reducir la variabilidad en los resultados. Esto es particularmente visible en las Figura 6, donde se aprecia que los modelos optimizados presentan menor dispersión en la precisión. Este hallazgo es relevante en escenarios donde la consistencia del modelo es tan importante como la precisión misma, como ocurre en aplicaciones críticas donde los resultados deben ser predecibles y estables.

A pesar de las claras ventajas de los MAG, cabe destacar que su aplicación no está exenta de costos computacionales adicionales. Si bien la optimización de hiperparámetros redujo los tiempos de entrenamiento en muchos casos, la etapa

inicial de exploración mediante MAG requiere una inversión significativa de recursos, particularmente en conjuntos de datos grandes o cuando se optimizan múltiples hiperparámetros simultáneamente. No obstante, los beneficios obtenidos, tanto en precisión como en eficiencia, justifican ampliamente este costo inicial en la mayoría de los escenarios evaluados.

También es importante mencionar que el rendimiento del MAG depende de la adecuada selección de sus propios parámetros, como el tamaño de la población, la tasa de mutación y el número de generaciones. Si bien los resultados obtenidos en este estudio son positivos, investigaciones futuras podrían centrarse en la automatización de la selección de estos parámetros o en la exploración de variantes del MAG que reduzcan aún más el costo computacional sin comprometer la precisión. En conclusión, este estudio ha demostrado la capacidad de los microalgoritmos genéticos para mejorar significativamente el rendimiento de diferentes algoritmos de clasificación en diversos conjuntos de datos. Las mejoras consistentes observadas en precisión, estabilidad y tiempos de entrenamiento posicionan a los MAG como una herramienta altamente eficaz para la optimización de hiperparámetros en aprendizaje automático, con potencial para ser aplicada en una amplia gama de problemas y aplicaciones prácticas.

REFERENCIAS

- [1] Domingos P. A few useful things to know about machine learning. *Communications of the ACM*. 2012;55(10):78-87. doi: [10.1145/2347736.2347755](https://doi.org/10.1145/2347736.2347755)
- [2] Erden C, Demir HI, Kökçam AH. Enhancing Machine Learning Model Performance with Hyper Parameter Optimization: A Comparative Study. *arXiv preprint arXiv:2302.11406*. 2023. doi: [10.48550/arXiv.2302.11406](https://doi.org/10.48550/arXiv.2302.11406)
- [3] Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary*

- Computation. 2007;8(3):256-279. doi: [10.1109/TEVC.2004.826067](https://doi.org/10.1109/TEVC.2004.826067)
- [4] Eiben AE, Smith JE. Introduction to Evolutionary Computing. Springer-Verlag; 2003. doi: [10.1007/978-3-662-44874-8](https://doi.org/10.1007/978-3-662-44874-8)
- [5] Mirjalili S. Genetic algorithm. In: Evolutionary Algorithms and Neural Networks. Springer, Cham; 2019. p. 43-55. doi: [10.1007/978-3-319-93025-1](https://doi.org/10.1007/978-3-319-93025-1)
- [6] Coello CAC, Van Veldhuizen DA, Lamont GB. Evolutionary algorithms for solving multi-objective problems. Springer Science & Business Media; 2002. doi: [10.1007/978-0-387-36797-2](https://doi.org/10.1007/978-0-387-36797-2)
- [7] Smith SL. A Review of Medical Applications of Genetic and Evolutionary Computation. In: Smith SL, Cagnoni S, editors. Genetic and evolutionary computation: medical applications. Wiley; 2011. Disponible en: <https://www.wiley.com/en-us/Genetic+and+Evolutionary+Computation%253A+Medical+Applications-p-9781119956785>
- [8] Kim JY, Cho SB. Evolutionary Optimization of Hyperparameters in Deep Learning Models. In: 2019 IEEE Congress on Evolutionary Computation (CEC); 2019 Jun 10-13; Wellington, New Zealand. IEEE; 2019. p. 831-837. doi: [10.1109/CEC.2019.8790354](https://doi.org/10.1109/CEC.2019.8790354)
- [9] Suganuma M, Shirakawa S, Nagao T. A genetic programming approach to designing convolutional neural network architectures. In: Proceedings of the 2017 Genetic and Evolutionary Computation Conference. New York: ACM; 2017. p. 497-504. doi: [10.1145/3071178.3071229](https://doi.org/10.1145/3071178.3071229)
- [10] Shchygol G, Yakovlev A, Trnka T, van Duin ACT, Verstraelen T. ReaxFF Parameter Optimization with Monte Carlo and Evolutionary Algorithms: Guidelines and Insights. Journal of Chemical Theory and Computation. 2019;15(12). doi: [10.1021/acs.jctc.9b00769](https://doi.org/10.1021/acs.jctc.9b00769)
- [11] Murthy DHR, Mohan KG, Jacob A. Evolutionary Strategies for Parameter Optimization in Deep Learning Models. International Journal of Intelligent Systems and Applications in Engineering. 2023;12(2s):371-378.
- [12] Venske SM, Gonçalves RA, Benelli EM, Delgado MR. ADEMO/D: an adaptive differential evolution for protein structure prediction problem. Expert Systems with Applications. 2016;56:209-226. doi: [10.1016/j.eswa.2016.03.009](https://doi.org/10.1016/j.eswa.2016.03.009)
- [13] Sorano R, Ripon KSN, Magnusson LV. Evolutionary multi-objective optimization of hyperparameters for decision support in healthcare. In: Kulkarni AJ, Gandomi AH, editors. Handbook of formal optimization. Springer Nature Singapore; 2023. p. 1-26. doi: [10.1007/978-981-19-8851-6_28-2](https://doi.org/10.1007/978-981-19-8851-6_28-2)
- [14] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. Wadsworth & Brooks/Cole Advanced Books & Software; 1984. <https://doi.org/10.1201/9781315139470>
- [15] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician. 1992;46(3):175-185. doi: [10.1080/00031305.1992.10475879](https://doi.org/10.1080/00031305.1992.10475879)
- [16] Freund Y, Schapire RE. Large margin classification using the perceptron algorithm. Machine Learning. 1999;37(3):277-296. doi: [10.1145/279943.279985](https://doi.org/10.1145/279943.279985)
- [17] Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747; 2016. doi: [10.48550/arXiv.1609.04747](https://doi.org/10.48550/arXiv.1609.04747)
- [18] Coello CAC, Pulido GT. A niched Pareto genetic algorithm for multiobjective optimization. In Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512) (Vol. 1, pp. 83-90). IEEE; 2000. doi: [10.1109/ICEC.1994.350037](https://doi.org/10.1109/ICEC.1994.350037)
- [19] Kelly M, Longjohn R, Nottingham K. The UCI Machine Learning Repository [Internet]. Disponible en: <https://archive.ics.uci.edu>

ACERCA DE LOS AUTORES



Antonio Guerrero Juárez es Ingeniero en Computación por la Universidad Autónoma Metropolitana, con experiencia en el diseño y construcción de software a gran escala, con especialización en análisis de datos y

optimización de procesos para la toma de decisiones en entornos empresariales. Actualmente, es Líder de Inteligencia Comercial en el Tecnológico de Monterrey, donde se dedica al análisis de información y creación de indicadores que optimizan los procesos de ventas mediante técnicas de machine learning. Anteriormente, desempeñó roles en Thomson Reuters como desarrollador Full Stack, participando en la optimización de software contable, en Grupo Salinas fue Líder de TI y Especialista en Bases de Datos, diseñando arquitecturas para grandes volúmenes de datos en proyectos de crédito y cobranza. En el ámbito académico trabajó como Ayudante en la Universidad Autónoma Metropolitana, apoyando en la formación de futuros ingenieros en computación y matemáticas aplicadas.



Abel García Nájera es Profesor-Investigador en el Departamento de Matemáticas Aplicadas y Sistemas de la UAM Unidad Cuajimalpa. Es Ingeniero en Electrónica por la UAM, Maestro en Ciencias de la Computación por el CICESE y Doctor en Ciencias de la Computación por la Universidad de Birmingham, Reino Unido. Sus líneas de investigación se engloban en la intersección de la optimización combinatoria multiobjetivo y de las metaheurísticas inspiradas en la naturaleza, la cual tiene aplicaciones en el diseño de rutas vehiculares, en la ingeniería de software, en el análisis de la influencia y en la detección de comunidades en redes sociales y en la optimización de portafolios de inversión. El Dr. García Nájera es miembro regular de la Academia Mexicana de Computación y es Nivel I del Sistema Nacional de Investigadoras e Investigadores.



Saúl Zapotecas Martínez recibió el grado de Doctor en Ciencias en Computación por el Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN). Actualmente es investigador Titular de Tiempo Completo en la Coordinación de Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Es miembro del Sistema Nacional de Investigadoras e Investigadores (SNII) y de la Academia Mexicana de Computación (AMEXCOMP). Ha sido autor y coautor de más de 70 trabajos de investigación publicados en revistas, capítulos de libros y memorias de congreso con arbitraje internacional estricto. Actualmente colabora con el grupo de investigación “BONUS: Big

Optimization and Ultra- Scale Computing” en el INRIA-Lille, Francia y en el laboratorio internacional “MODO: Frontiers in Massive Optimization and Computational Intelligence” de la Universidad de Shinshu, Nagano, Japón.



Karen Miranda es Doctora en Ciencias (Computación) por la Universidad de Lille 1 en Francia, maestra en Ciencias y Tecnologías de la Información de la Universidad Autónoma Metropolitana Unidad Iztapalapa y licenciada en Informática de la UNAM. Sus intereses de investigación se centran en el diseño y evaluación de protocolos para las redes de sensores y redes de relevos móviles así como en la aplicación de técnicas de Inteligencia Computacional en problemas relacionados con redes de computadoras. Sobre estos temas ha publicado en diversas revistas y conferencias internacionales. La Dra. Miranda es Profesora Investigadora adscrita al Departamento de Sistemas de Información y Comunicaciones de la UAM Unidad Lerma, en donde además ha sido Jefa del Área de Sistemas de Información y Ciencias Computacionales y Coordinadora de Estudios de la Licenciatura en Ingeniería en Computación y Telecomunicaciones.