Comparativa de algoritmos metaheurísticos para clustering orientado a la explicación de errores de clasificación

Aya Attioui[0009−0003−1220−2179], Aurora Ramírez[0000−0002−1916−6559], and José Raúl Romero[0000−0002−4550−6385]

Universidad de Córdoba, Córdoba 14071, España
{z52atata,aramirez,jrromero}@uco.es

Resumen Comprender el comportamiento de los modelos de aprendizaje automático requiere no solo analizar las predicciones correctas, sino también estudiar por qué comete errores de clasificación. En este trabajo investigamos el uso de técnicas de clustering para identificar y analizar patrones en instancias mal clasificadas, dentro del marco de la Inteligencia Artificial Explicable. El objetivo del estudio es explorar el potencial de los métodos de clustering basados en metaheurísticas, donde se tenga mayor flexibilidad a la hora de determinar el número óptimo de clusters. El estudio experimental propuesto evalúa el comportamiento de cuatro metaheurísticas frente a un algoritmo particional tradicional, analizando además el impacto de algunos parámetros de configuración. Los resultados muestran que los métodos de clustering metaheurístico son competitivos y, en algunos casos, superan al método de referencia, particularmente en términos de calidad y robustez. Estos hallazgos destacan el potencial del clustering metaheurístico como una herramienta eficaz para apoyar la posterior explicación de errores, contribuyendo a sistemas de aprendizaje automático más transparentes.

Keywords: Inteligencia Artificial Explicable · Clustering · Errores de clasificación · Metaheurísticas

1. Introducción

Los modelos de aprendizaje automático (ML, *Machine Learning*) son ampliamente utilizados en aplicaciones del mundo real debido a su alto rendimiento predictivo. Sin embargo, incluso los modelos altamente precisos inevitablemente producen errores de clasificación. En el caso de la clasificación binaria, los categorizamos en Falsos Positivos (FP) y Falsos Negativos (FN). Comprender por qué se cometen estos errores es esencial para mejorar la fiabilidad, la transparencia y la confianza del modelo, especialmente en dominios críticos [5]. En este contexto, es interesante aplicar los principios de la Inteligencia Artificial Explicable (XAI, *Explainable Artificial Intelligence*) [7]. No obstante, la mayoría de los enfoques existentes se centran en explicar predicciones individuales o extraer la relevancia de las características, sin prestar especial atención a los errores del modelo.

En un trabajo anterior, propusimos una metodología basada en clustering y explicaciones locales para analizar la naturaleza de los errores de clasificación [11]. El objetivo era demostrar que los errores de clasificación no son puramente aleatorios, sino que pueden presentar similitudes en el espacio de características. Gracias al clustering, las técnicas locales de XAI disponían de un ejemplo representativo para dotar al usuario de una explicación sobre ese tipo de error. Para la agrupación de instancias mal clasificadas, nuestro método utilizaba el algoritmo Affinity Propagation (AP) [6]. Su elección correspondía a dos motivos principales: no necesita especificar el número de clusters y devuelve una instancia del conjunto de datos como representante de cada grupo. Ambas características eran necesarias para el tipo de análisis propuesto, pues no es posible determinar de antemano cuántos tipos de error está cometiendo el modelo y la explicación será más útil si se obtiene a partir de datos realistas.

Si bien el algoritmo obtenía un buen rendimiento en general, presentaba una limitación importante. AP puede no converger o devolver un número excesivo de clusters, lo cual compromete la fiabilidad de los resultados y limita su aplicación en algunos conjuntos de datos. Para abordar estas limitaciones, este trabajo propone analizar alternativas al método de clustering. Para ello comparamos experimentalmente varios algoritmos metaheurísticos [3], con el fin de estudiar si sus estrategias de optimización pueden mejorar la calidad de los resultados. Nuestro estudio comparativo mantiene AP como método de referencia, e incluye cuatro métodos de clustering basados en Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO) y Whale Optimization Algorithm (WOA). La principal contribución de este trabajo es, por tanto, el estudio del rendimiento de algoritmos de clustering metaheurístico para el análisis de errores de clasificación. Además, nuestro estudio experimental analiza el impacto de dos parámetros clave en la optimización, como son el número de generaciones y el tamaño de la población. La mejora de los resultados en la fase de clustering tiene como fin último facilitar una mejor comprensión de las explicaciones que generaría una técnica XAI local.

El resto del artículo se organiza de la siguiente manera. La Sección 2 revisa el trabajo relacionado. La Sección 3 describe el problema del análisis de errores de clasificación y los métodos de clustering utilizados. La Sección 4 presenta la metodología experimental, incluyendo preguntas de investigación, algoritmos, conjuntos de datos y métricas de evaluación. La Sección 5 presenta y discute los resultados. Finalmente, la Sección 6 resume las conclusiones y trabajo futuro.

2. Trabajo relacionado

El análisis de errores de clasificación se ha convertido en una importante línea de investigación en XAI, ya que comprender los fallos del modelo es esencial para mejorar la fiabilidad y la confianza en los sistemas de aprendizaje automático. Algunos trabajos recientes que abordan este problema en dominios específicos como el análisis de regiones mal clasificadas en imágenes [1] o la explicación de errores en previsión de ventas [9]. De modo más general, Rizzi et al. proponen

estudiar el comportamiento global del modelo a partir de las explicaciones locales obtenidas para todas las instancias mal clasificadas [12]. La propuesta de Kim et al. se basa en explorar ejemplos y contraejemplos, localizando estos últimos mediante un algoritmo voraz [8].

Nuestro trabajo previo, basado en clustering con el algoritmo AP, ofrecía dos ventajas respecto a otras propuestas [11]. Por un lado, la metodología es de ámbito general, aplicable a cualquier conjunto de datos tabular de clasificación binaria. Por otro lado, la identificación de un número reducido de errores representativos alivia la fase de explicabilidad, pues solo se necesita inspeccionar un subconjunto de muestras. Nuestra metodología ha sido aplicada en un problema de mantenimiento predictivo [5], lo que refuerza su utilidad práctica. Este trabajo mantiene el uso de AP como algoritmo de clustering, pero adapta la fase de explicabilidad. En concreto, se utiliza el algoritmo iSAGE para explicar instancias en un contexto de aprendizaje multi-etiqueta y *online learning*.

Por último, nuestro trabajo está relacionado con la aplicación de algoritmos metaheurísticos para clustering [3]. En este tipo de enfoques, cada solución se codifica como un vector de centroides y se evalúa mediante una función de fitness basada en métricas internas de clustering. La población de soluciones evoluciona mediante operadores específicos de cada algoritmo. En este trabajo exploramos tres tipos de metaheurísticas diferentes: GWO, que simula la jerarquía de caza de los lobos grises; PSO, que modela el movimiento colectivo de partículas; y WOA, que imita la estrategia de caza en espiral de las ballenas.

3. Enfoque propuesto

Esta sección define el problema planteado (Sección 3.1) y describe los algoritmos de clustering empleados para el estudio comparativo (Sección 3.2).

3.1. Definición del problema

Dado un conjunto de datos etiquetado, se entrena un modelo de clasificación y se evalúa sobre la partición de prueba. Las predicciones resultantes se utilizan para identificar las instancias mal clasificadas, ya sean FP o FN.

El problema abordado en este trabajo consiste en determinar si estas instancias mal clasificadas pueden organizarse en grupos significativos que reflejen características comunes de los errores del modelo. Para ello, se aplica un proceso de clustering a los subconjuntos de FP y FN combinados. La metodología es agnóstica al método de clustering, si bien es preferible un algoritmo que descubra por sí mismo el número de clusters. Una vez determinados los centroides, serían analizados por una técnica XAI local para explicar qué características han sido las más influyentes en esa predicción errónea. No obstante, este último paso queda fuera del alcance de este trabajo, donde nos centramos en mejorar el rendimiento del proceso de clustering para lograr una mejor caracterización de los errores.

3.2. Métodos de clustering

A continuación se presentan las principales características de cada método de clustering. En primer lugar, se explica el funcionamiento de AP, que servirá como método de referencia. Los métodos metaheurísticos han sido elegidos por dos razones: la posibilidad de configurar el número máximo de clusters, lo cual permite acotar el número máximo de explicaciones a generar, y la disponibilidad de sus implementaciones en librerías Python, lo que permite integrarlos fácilmente con el resto de la metodología.

Affinity Propagation (AP). Este algoritmo determina automáticamente el número de clusters mediante un intercambio iterativo de mensajes entre instancias [6]. Cada instancia puede convertirse en representante de un grupo y el algoritmo decide cuáles son los mejores representantes (llamados "exemplar") a través de dos criterios: la *responsabilidad*, que mide si una instancia es buena candidata a "exemplar", y la *disponibilidad*, que mide si una instancia debería unirse a un "exemplar" concreto. Al finalizar el proceso, si el algoritmo converge, cada cluster queda representado por una instancia real del conjunto de datos.

Metaheurísticas. Utilizamos la librería EvoCluster [10], que proporciona implementaciones nativas de diferentes paradigmas bioinspirados, así como la opción de utilizar optimizadores externos. Para la comparativa, se selecciona la implementación nativa de PSO (EvoCluster-PSO) y tres optimizadores externos, provenientes de la librería MEALPY [13]: MEALPY-PSO, MEALPY-GWO y MEALPY-WOA. En todos los casos, cada solución candidata es un vector de k centroides, y cada punto del dataset se asigna al centroide más cercano usando distancia Euclídea. La calidad de cada solución se mide minimizando el índice Davies-Bouldin. A diferencia de AP, los algoritmos basados en metaheurísticas no aseguran que los centroides óptimos sean instancias reales del dataset. Esto implica que, para generar la posterior explicación, se debería elegir una instancia real por cada cluster. No obstante, nuestro estudio pretende analizar si relajar esta restricción puede favorecer la obtención de mejores resultados en la fase de clustering. Igualmente, establecer un límite superior al máximo de cluster puede favorecer una mejor convergencia y reducir el número de explicaciones a mostrar al usuario.

4. Metodología experimental

Esta sección describe el marco experimental diseñado para comparar los métodos de clustering. En primer lugar se formulan las preguntas de investigación (Sección 4.1). A continuación, se detalla la configuración de los algoritmos (Sección 4.2), los conjuntos de datos considerados para la experimentación (Sección 4.3), y las métricas para evaluar los resultados (Sección 4.4).

4.1. Preguntas de investigación

Se definen dos preguntas de investigación para evaluar el enfoque propuesto. **RQ1** *¿Cuál es el impacto de los parámetros de configuración en el rendimiento de los métodos de clustering metaheurístico?* Se consideran dos parámetros clave, comunes a todas las metaheurísticas evaluadas, que determinan el compromiso entre convergencia y diversidad en la exploración del espacio de búsqueda: el tamaño de la población (número de soluciones candidatas) y el número de iteraciones. Para garantizar una evaluación robusta, cada configuración se ejecuta diez veces con diferentes semillas aleatorias. Los resultados se analizan tanto en términos de rendimiento (valores de las métricas de calidad del clustering) como de estabilidad (consistencia entre semillas).
RQ2 *¿Mejoran los métodos de clustering metaheurístico los resultados de Affinity Propagation para el análisis de errores de clasificación?* Para responder a esta cuestión, se seleccionan las mejores configuraciones de parámetros identificadas en RQ1 para cada método metaheurístico. Estas configuraciones se utilizan posteriormente para agrupar las instancias mal clasificadas, bajo las mismas condiciones experimentales que AP. Al igual que en RQ1, cada experimento se repite con diez semillas aleatorias. Este análisis se va a centrar en tres aspectos: las diferencias en la calidad del clustering, las variaciones en el número de clusters detectados y la estabilidad de los resultados entre semillas.

4.2. Configuración de los algoritmos

La Tabla 1 resume los parámetros utilizados para cada método de clustering. Para AP, se emplean los valores por defecto de scikit-learn, ya que nuestro estudio previo demostró que son suficientemente estables [11]. Respecto al número de clusters, los métodos metaheurísticos necesitan que se especifique un límite máximo (k). Establecemos k al valor de la moda del número de clusters devueltos por AP en sus diez ejecuciones. Si dicho valor es superior a 20, fijamos $k = 20$ para limitar el número de explicaciones que se generarían. No obstante, los métodos metaheurísticos pueden encontrar soluciones con un número inferior de clusters. El resto de los parámetros se mantienen por defecto.

Tabla 1. Parámetros de configuración de los métodos de clustering.

Algoritmo	**Parámetro**	**Valor**
Affinity Propagation	damping	0.5
	max_iter	200
	convergence_iter	15
MEALPY-GWO, MEALPY-PSO MEALPY-WOA, EvoCluster-PSO	Tamaño de población	{20, 50}
	Número de iteraciones	{30, 50}
	Nº mín. de clusters	2
	Nº máx. de clusters	min(20, moda(AP))

En cuanto a los modelos de clasificación sobre los que se analizarán los errores, este trabajo considera los mismos modelos que nuestro trabajo anterior [11]: Random Forest (RF), Support Vector Machine (SVM) y Extreme Gradient Boosting (XGBoost). En todos los casos se aplica una partición del 70 % de instancias para entrenamiento y del 30 % de instancias para test. Tras la evaluación en el conjunto de prueba, se extraen las instancias mal clasificadas para su posterior análisis mediante clustering.

4.3. Conjuntos de datos

Para evaluar el enfoque propuesto, se utilizan diez conjuntos de datos públicos obtenidos del repositorio UCI [4]. Estos conjuntos de datos cubren diferentes dominios y presentan distintos niveles de complejidad en términos de número de instancias y características. Esta diversidad genera diferentes conjuntos de FP y FN según el conjunto de datos y el clasificador utilizado, lo que permite evaluar la robustez de los métodos de clustering ante distintos escenarios de error. La Tabla 2 resume las principales características de los conjuntos de datos considerados en este estudio, incluyendo el número de instancias y características.

Tabla 2. Características de los conjuntos de datos utilizados en la experimentación.

Conjunto de datos	**Instancias**	**Características**
Diabetes	768	8
WDBC	569	30
Column	310	6
Blood Transfusion	748	4
isolet_bin	600	617
usps_bin	1424	256
Phoneme	5404	5
KC1	2109	21
ozone-level-8hr	2534	72
hill_valley	1212	101

4.4. Métricas de evaluación

Para evaluar la calidad de los resultados de clustering se consideran tres métricas de validación interna ampliamente utilizadas [2]: Silhouette Score (SC), índice de Calinski–Harabasz (CH) e índice de Davies–Bouldin (DB). Se seleccionan estas tres métricas por ser complementarias entre sí: SC evalúa la cohesión y separación a nivel de instancia, CH captura la estructura global de los clusters, y DB mide la compacidad relativa entre clusters. Estas métricas permiten evaluar diferentes aspectos de la calidad del clustering sin requerir etiquetas reales. Puesto que los algoritmos metaheurísticos utilizan DB como fitness, incluir otras métricas es recomendable para que la comparativa con AP sea justa.

- La métrica SC mide la cohesión y separación de los clusters. Evalúa qué tan similar es una instancia a su propio cluster en comparación con otros clusters. Sus valores oscilan entre -1 y 1, donde valores más altos indican clusters mejor definidos y más separados.
- El índice CH evalúa la relación entre la dispersión inter-cluster y la dispersión intra-cluster. Valores más altos indican mejor rendimiento del clustering, reflejando clusters bien separados y compactos. No obstante, este índice puede verse influenciado por el número de clusters, favoreciendo artificialmente a métodos que detectan más grupos. Por ello, el SC y el DB se consideran métricas más fiables cuando los métodos difieren en el número de clusters.
- El índice DB mide la similitud promedio entre cada cluster y el más similar a él. A diferencia de las anteriores, valores más bajos indican mejor calidad del clustering, ya que corresponden a clusters más compactos y mejor separados.

5. Resultados experimentales

Esta sección presenta los resultados obtenidos del estudio experimental. Primero se analiza el impacto de los parámetros de configuración en los métodos de clustering (RQ1). A continuación, se comparan las mejores configuraciones con AP (RQ2). Finalmente, se presenta una discusión de los principales hallazgos.

5.1. Resultados para RQ1

Se evaluaron todas las combinaciones de tamaño de población (20, 50) y número de iteraciones (30, 50) sobre los 10 conjuntos de datos y 3 clasificadores, repitiendo cada experimento con 10 semillas aleatorias. La Figura 1 muestra la distribución de los valores de SC para cada configuración de parámetros, agregando todos los conjuntos de datos, clasificadores y semillas. Como puede observarse, las distribuciones de los resultados son bastante similares para cada configuración de parámetros, con ligeras mejoras (en promedio) cuando se considera un tamaño de población igual a 50 durante 50 iteraciones. Por tanto, esta configuración será la utilizada en RQ2.

A pesar de la tendencia general observada, es interesante analizar algunos resultados particulares. La Tabla 3 muestra la estabilidad de los cuatro métodos en el conjunto de datos *phoneme* para la mejor configuración (pop=50, iter=50). Los resultados muestran la media y la desviación estándar de las métricas SC, CH y DB considerando las 10 semillas y los 3 clasificadores. Entre los métodos metaheurísticos evaluados, EvoCluster-PSO ofrece el mejor compromiso: obtiene el mejor resultado medio tanto en SC (0.432) como en DB (0.403) con la menor variabilidad (especialmente en DB, con un valor de 0.061). En cuanto al índice CH, MEALPY-GWO obtiene el valor más alto (12.745), aunque con una elevada desviación estándar (6.597), lo que refleja inestabilidad en los resultados. Por el contrario, los tres algoritmos de MEALPY presentan desviaciones estándar superiores, especialmente en DB, reflejando mayor sensibilidad a la inicialización aleatoria.

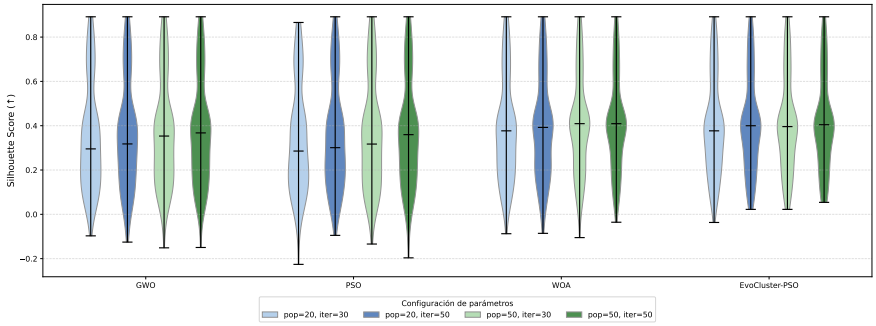


Figura 1. Distribución de SC (↑) por método y configuración de parámetros (pop∈20,50, iter∈30,50), sobre todos los datasets, clasificadores y 10 semillas.

Tabla 3. Estabilidad entre semillas (phoneme, pop=50, iter=50)

Método	**SC (media ± std)**	**CH (media ± std)**	**DB (media ± std)**
MEALPY-GWO	0.149 ± 0.167	12.745 ± 6.597	0.685 ± 0.188
MEALPY-PSO	0.067 ± 0.115	9.998 ± 6.251	0.699 ± 0.140
MEALPY-WOA	0.390 ± 0.154	5.881 ± 2.348	0.444 ± 0.118
EvoCluster-PSO	0.432 ± 0.104	5.908 ± 2.001	0.403 ± 0.061

La Tabla 4 muestra los resultados para MEALPY-PSO sobre el conjunto *phoneme* (clasificador SVM), caso representativo por su alta variabilidad. Los resultados muestran que la configuración con pop=50 e iter=50 obtiene la mejor calidad media (SC=0.124, DB=0.545) y la menor variabilidad entre semillas (std DB=0.097). Aumentar únicamente el número de iteraciones de 30 a 50 con pop=20 no mejora los resultados, de hecho el SC medio decrece de 0.106 a 0.068, lo que indica que este algoritmo converge rápidamente y que las iteraciones adicionales sin una población suficientemente grande no aportan exploración útil. En cuanto al tamaño de población, pasar de 20 a 50 sí produce una mejora consistente, reduciendo el DB de 0.717 a 0.580 con 30 iteraciones, confirmando que el tamaño de la población tiene mayor influencia que el número de iteraciones en este contexto.

Tabla 4. Impacto de los parámetros en MEALPY-PSO (phoneme, SVM, 10 semillas).

Población	**Iteraciones**	**SC (media ± std)**	**DB (media ± std)**
20	30	0.106 ± 0.044	0.717 ± 0.134
20	50	0.068 ± 0.115	0.612 ± 0.124
50	30	0.095 ± 0.191	0.580 ± 0.119
50	50	0.124 ± 0.147	0.545 ± 0.097

5.2. Resultados para RQ2

La Tabla 5 presenta la comparación de resultados entre AP y los métodos metaheurísticos utilizando la mejor configuración. Para cada conjunto de datos, se muestran los valores de SC y DB obtenidos por el algoritmo AP. Las siguientes columnas muestran el resultado de la mejor metaheurística para ese conjunto de datos, indicando el algoritmo concreto que lo consigue. Se resalta en negrita el mejor resultado para cada conjunto de datos y métrica.

Tabla 5. Comparativa entre AP y el clustering metaheurístico.

Conjunto de datos	**AP**		**Mejor metaheurística**			**p-value**
	SC	**DB**	**Algoritmo**	**SC**	**DB**	
diabetes	0.17±0.04	1.18±0.01	MEALPY-WOA	**0.39±0.04**	**0.45±0.03**	0.003*
wdbc	**0.18±0.02**	0.96±0.08	MEALPY-GWO	0.16±0.05	**0.55±0.01**	0.002*
column	0.22±0.01	0.99±0.13	MEALPY-PSO	**0.31±0.10**	**0.44±0.04**	0.182
blood-trans.	**0.41±0.00**	0.79±0.02	MEALPY-WOA	0.40±0.03	**0.37±0.02**	0.382
hill_valley	0.48±0.23	0.28±0.05	MEALPY-WOA	**0.82±0.05**	**0.12±0.03**	0.002*
isolet_bin	0.10±0.00	0.47±0.00	—	0.10±0.00	0.47±0.00	—
phoneme	0.33±0.00	0.97±0.02	EvoCluster-PSO	**0.43±0.10**	**0.40±0.06**	0.433
usps_bin	0.13±0.05	0.93±0.27	MEALPY-PSO	**0.17±0.10**	**0.59±0.05**	0.002*
kc1	0.27±0.01	0.85±0.13	MEALPY-GWO	**0.66±0.03**	**0.22±0.02**	0.002*
ozone-level-8hr	0.11±0.02	1.35±0.03	MEALPY-GWO	**0.44±0.00**	**0.41±0.00**	0.002*

* $p < 0.05$: diferencia estadísticamente significativa
N/A: todos los métodos obtienen resultados idénticos

Los resultados revelan varios hallazgos interesantes. Los métodos metaheurísticos mejoran el índice DB en 8 de los 10 conjuntos de datos. Las mayores mejoras se observan en *ozone-level-8hr* (AP: 1.346 → 0.406, −70 %), *diabetes* (AP: 1.177 → 0.449, −62 %), *column* (AP: 0.989 → 0.435, −56 %) y *kc1* (AP: 0.846 → 0.223, −74 %), indicando que los clusters generados por los enfoques metaheurísticos son más compactos y mejor separados. El test de Wilcoxon confirma que estas diferencias son estadísticamente significativas en 6 conjuntos ($p < 0.05$). Estos casos se muestran en la tabla con el símbolo $*$.

El resultado más destacado se produce para el conjunto de datos *hill_valley*. Los métodos metaheurísticos alcanzan $SC = 0{,}82$ frente a $SC = 0{,}48$ de AP (+71 %), y reducen el índice DB de 0.28 a 0.12 (−58 %). Mientras AP detecta una media de 48 clusters, los métodos metaheurísticos trabajan con un máximo de 20 clusters, obteniendo particiones más compactas y de mayor calidad. Esto también tiene implicaciones para XAI: analizar 20 grupos de errores es mucho más manejable para un usuario que analizar 48, ya que reduce el número de explicaciones a generar y facilita la comprensión global de los fallos del modelo.

Otros conjuntos de datos con resultados significativos son: *kc1*, *isolet_bin* y *wdbc* y *phoneme*. *kc1* presenta la mayor mejora en SC. MEALPY-GWO y EvoCluster-PSO alcanzan SC= 0,66, frente a SC= 0,27 de AP (+141 %), lo que indica que AP no logra identificar correctamente la estructura de errores en este conjunto. *isolet_bin* constituye un caso especial. Todos los métodos, inclui-

dos los metaheurísticos, obtienen exactamente los mismos resultados SC= 0,10, DB= 0,47, 2 clusters, independientemente de la semilla o los parámetros. Esto sugiere que la estructura de errores es extremadamente simple o que el número de instancias mal clasificadas es insuficiente para encontrar particiones diferentes. *wdbc* es el único conjunto donde AP supera a los métodos metaheurísticos en SC (0.18 frente a 0.16). Sin embargo, los métodos metaheurísticos mejoran el DB (AP: 0.96 $\rightarrow$ 0.55), sugiriendo clusters más compactos internamente. En *phoneme*, el mejor resultado lo obtiene EvoCluster-PSO con SC= 0,43 y DB= 0,40, destacando ademas por su mayor estabilidad entre semillas (Tabla 3), lo que lo posiciona como el método mas robusto en conjuntos de datos complejos.

La Tablas 6 y 7 ofrecen los resultados detallados para los conjuntos de datos *phoneme* (mayor complejidad entre los datasets comunes) y *hill_ valley* (mayor mejora observada). En *phoneme*, EvoCluster-PSO es el método más equilibrado pero produce de media solo 2.2 clusters, una partición muy gruesa que puede limitar la granularidad del análisis de errores. En *hill_ valley*, todos los métodos evolutivos convergen a resultados prácticamente idénticos (SC $\approx$ 0.816–0.818, DB $\approx$ 0.115–0.116), lo que indica que la estructura de los errores en ese conjunto está muy bien definida y es fácilmente identificable por cualquier metaheurística.

Tabla 6. Resultados detallados *phoneme* (pop=50, iter=50).

Método	**SC(↑)**	**CH(↑)**	**DB(↓)**	**Nº clusters (media)**
Affinity Propagation	0.33±0.00	65.18±23.14	0.97±0.02	20.00±3.62
MEALPY-GWO	0.15±0.17	12.75±6.60	0.68±0.19	18.33±1.27
MEALPY-PSO	0.07±0.12	10.00±6.25	0.70±0.14	18.33±1.27
MEALPY-WOA	0.39±0.15	5.88±2.35	0.44±0.12	18.33±1.27
EvoCluster-PSO	0.43±0.10	5.91±2.00	0.40±0.06	2.20±0.61

Tabla 7. Resultados detallados en *hill_ valley* (pop=50, iter=50).

Método	**SC(↑)**	**CH(↑)**	**DB(↓)**	**Nº clusters (media)**
Affinity Propagation	0.48±0.22	4952.56±3821.10	0.28±0.05	48.03±34.84
MEALPY-GWO	0.82±0.05	40.46±20.98	0.12±0.03	20.00±0.00
MEALPY-PSO	0.81±0.05	40.38±21.04	0.12±0.03	20.00±0.00
MEALPY-WOA	0.82±0.05	40.37±20.85	0.12±0.03	20.00±0.00
EvoCluster-PSO	0.82±0.05	40.54±21.11	0.12±0.03	2.40±0.50

5.3. Discusión

Respecto a la RQ1, nuestro estudio indica que el tamaño de la población tiene una mayor influencia que el número de iteraciones: pasar de $pop = 20$ a $pop = 50$ mejora consistentemente la calidad del clustering, mientras que aumentar el número de iteraciones sin aumentar la población no produce mejoras sistemáticas.

La variabilidad entre semillas es alta para MEALPY-GWO y MEALPY-PSO, lo que pone de manifiesto la necesidad de evaluar varias semillas. En este contexto, EvoCluster-PSO se presenta como el método más robusto frente a la inicialización aleatoria. El análisis de estabilidad realizado sobre el conjunto de datos *phoneme* resulta representativo del comportamiento observado en el resto de datasets: EvoCluster-PSO y MEALPY-WOA muestran la menor variabilidad entre semillas en todos los conjuntos evaluados. MEALPY-GWO y MEALPY-PSO presentan una mayor sensibilidad a la inicialización, especialmente en datasets con estructuras de error complejas como *phoneme* y *blood-transfusion*.

Respecto a la RQ2, los métodos metaheurísticos superan a AP en el índice DB en 8 de los 10 conjuntos de datos, y en SC en 7 de 10. Las mejoras son especialmente pronunciadas en conjuntos con alta dimensionalidad y un elevado número de errores, como *kc1*, *ozone-level-8hr*, *hill_valley* y *diabetes*. Sin embargo, existen casos en los que los métodos metaheurísticos no aportan mejora (por ejemplo, *isolet_bin*) o incluso obtienen un peor SC (como en *wdbc*), lo que confirma que no existe un método universalmente óptimo. La selección del algoritmo debe adaptarse a las características específicas de cada conjunto de datos.

Cabe señalar que la ventaja que ofrecen los métodos metaheurísticos en cuanto a las métricas de calidad del clustering son a costa de no garantizar que los centroides obtenidos se correspondan con instancias reales del dataset. Este aspecto sí es garantizado por AP, cuyos "exemplars" son instancias reales del conjunto de datos (es decir, instancias concretas mal clasificadas). Esto implica que, en el caso de utilizar métodos de clustering metaheurístico, sería necesario identificar la instancia real más cercana a cada centroide para poder aplicar métodos de explicabilidad local como SHAP, LIME o BreakDown de forma realista.

6. Conclusiones

Este trabajo presenta un estudio experimental sobre el uso de métodos de clustering metaheurístico para el análisis de errores de clasificación. Para ello se han evaluado y comparado cuatro enfoques basados en tres paradigmas (GWO, PSO y WOA) frente un algoritmo particional (AP). Los resultados muestran que el tamaño de población tiene mayor influencia que el número de iteraciones, siendo $pop = 50$, $iter = 50$ la configuración más equilibrada. Los métodos metaheurísticos superan a AP en el índice DB en 8 de los 10 datasets, con mejoras de más del 70 % según el conjunto de datos. Como trabajo futuro, se plantea integrar los resultados del clustering metaheurístico en la metodología XAI, obteniendo explicaciones locales con técnicas como SHAP, LIME y BreakDown. Asimismo, el estudio podría extenderse a más conjuntos de datos y clasificadores para ampliar el alcance de la experimentación y la generalización de los resultados. Finalmente, se explorará el diseño de algoritmos metaheurísticos específicos para el problema, así como estrategias adaptativas para seleccionar el mejor método según las características específicas del conjunto de datos.

Agradecimientos

Trabajo realizado en el contexto del Proyecto PID2024-155284NA-I00 financiado por MICIU/AEI/10.13039/501100011033 y por FEDER, UE.

Referencias

1. Alirezaie, M., Längkvist, M., Sioutis, M., Loutfi, A.: A symbolic approach for explaining errors in image classification tasks. In: IJCAI Workshop on Learning and Reasoning (2018)
2. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recognition **46**(1), 243–256 (2013)
3. Bagirov, A., Karmitsa, N., Taheri, S.: Metaheuristic Clustering Algorithms, pp. 179–201. Springer Nature Switzerland (2025)
4. Dua, D., Graff, C.: UCI machine learning repository (2017), https://archive.ics.uci.edu/ml
5. Esteban, A., Ramírez, A., García-Martínez, C., Zafra, A.: Explainable multi-fault predictive maintenance through analysis of wrong predictions. In: 26th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL). pp. 281–293 (2025)
6. Frey, B.J., Dueck, D.: Clustering by Passing Messages Between Data Points. Science **315**(5814), 972–976 (2007)
7. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5) (2018)
8. Kim, B., Khanna, R., Koyejo, O.: Examples are not enough, learn to criticize! Criticism for interpretability. In: 30th International Conference on Neural Information Processing Systems. pp. 2288–2296 (2016)
9. Lucic, A., Haned, H., de Rijke, M.: Why does my model fail? Contrastive local explanations for retail forecasting. In: Conference on Fairness, Accountability, and Transparency. pp. 90–98 (2020)
10. Qaddoura, R., Faris, H., Aljarah, I., Castillo, P.A.: EvoCluster: An open-source nature-inspired optimization clustering framework in Python. In: International Conference on the Applications of Evolutionary Computation (EvoStar). pp. 20–36. Springer (2020)
11. Ramírez, A., Ventura, S., Romero, J.: Método xai basado en agrupamiento para la explicación de errores en clasificación. In: XIX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA). pp. 35–40 (2021)
12. Rizzi, W., Di Francescomarino, C., Maggi, F.: Explainability in predictive process monitoring: When understanding helps improving. In: International Conference on Business Process Management. pp. 141–158 (2020)
13. Van Thieu, N., Mirjalili, S.: MEALPY: An open-source library for latest meta-heuristic algorithms in Python. Journal of Systems Architecture **139**, 102871 (2023)