

Balanceo de clases mediante evolución diferencial

Rafael Muñoz-Cervantes, Efrén Mezura-Montes,
Héctor-Gabriel Acosta-Mesa

Universidad Veracruzana,
Instituto de Investigaciones en Inteligencia Artificial,
México

rafamucerv@gmail.com, {emezura, heacosta}@uv.mx

Resumen. El desbalanceo de clases en bases de datos suele ser muy común cuando nos enfrentamos a problemas de la vida real, por lo que se han desarrollado diversas técnicas para dar solución. En este trabajo se aborda el problema de balanceo de clases combinando técnicas de muestreo mediante Evolución Diferencial. La propuesta se compara contra técnicas clásicas de balanceo de clases y también se comparan variantes de Evolución Diferencial. Los resultados muestran que nuestra propuesta logra encontrar el método con sus hiperparámetros que mejor se desempeña para las bases de datos probadas.

Palabras clave: Datos desbalanceados, evolución diferencial, balanceo de clases.

Class Balancing Using Differential Evolution

Abstract. Class imbalance in databases is very common when we face real-life problems, then several techniques have been developed to provide a solution. In this work, we address the balancing class problem by combining sampling techniques by means of Differential Evolution. We compare the proposal against classical balancing techniques and Differential Evolution variants are also compared. The results show that our proposal manages to find the method with its hyperparameters that performs best for the tested databases.

Keywords: Imbalance data, differential evolution, class balancing.

1. Introducción

El aprendizaje automático ha logrado ocupar un lugar muy importante en áreas científicas e industriales debido a la flexibilidad y capacidad de contender con problemas relacionados a la vida real gracias a la existencia de distintas

técnicas capaces de hacer frente a la diversidad de situaciones con las que nos podemos encontrar. Uno de los problemas que aparece frecuentemente cuando se trabaja con bases de datos creadas a partir de problemas reales, es el desbalance.

Una base de datos se dice que está desbalanceada cuando el número de instancias pertenecientes a una o más clases es menor (*clase minoritaria*) en comparación con el resto de las clases disponibles (*clase mayoritaria*).

Este tipo de bases de datos se pueden encontrar en distintos campos de la computación como la Visión Computacional [4,5,9] y la Seguridad Informática [13], así como en tareas relacionadas con la detección de fraudes bancarios [20], clasificación de textos [15], predicción de mantenimiento [14] o detección de cáncer [21].

Las complicaciones que conlleva el aprendizaje a partir de una base de datos desbalanceada se ven reflejadas en el desempeño de los clasificadores, pues se crea un sesgo hacia la clase mayoritaria, por lo que muchos de los clasificadores obtienen un mal desempeño respecto a la clase minoritaria, la cual, suele ser la clase de interés en gran parte de las aplicaciones de la vida real. Para hacer frente a este problema, se han realizado contribuciones que logran contrarrestar los efectos de una base de datos desbalanceada al realizar una tarea de clasificación.

Estas aproximaciones se pueden dividir en 4 grupos: métodos de preprocesamiento, métodos de aprendizaje sensibles al costo, métodos centrados en algoritmos y métodos híbridos [8]. Dentro de los métodos de preprocesamiento se encuentran los métodos de selección de atributos y los métodos basados en muestreo, siendo estos últimos de nuestro interés debido a su popularidad [6], así como su efectividad para abordar el problema de desbalanceo de datos al realizar tareas de aprendizaje supervisado [3].

Los métodos basados en muestreo se dividen en tres clases: *hybridsampling*, *oversampling* y *undersampling*. Estas aproximaciones buscan aumentar el número de instancias de la clase minoritaria (*oversampling*), disminuir el número de instancias de la clase mayoritaria (*undersampling*) o realizar ambos procedimientos (*hybridsampling*).

Dentro de la literatura revisada se han encontrado más de 50 ejemplos de métodos basados en muestreo de las distintas aproximaciones mencionadas [6,17,19], sin embargo, de toda la literatura revisada, no logramos detectar alguna propuesta sobre la forma de elegir un método de muestreo para una base de datos dada, lo que encontramos fue que usualmente se seleccionan los métodos más populares y se fijan sus parámetros sin un criterio específico.

Como resultado del intento por mejorar los métodos de balanceo basados en muestreo, así como también mejorar la selección de estos mismos para su aplicación, se ha realizado una combinación entre este campo del aprendizaje automático con técnicas alternativas de optimización como la computación evolutiva. La combinación de estas áreas ha demostrado tener un desempeño significativamente mejor, en comparación con las técnicas de aprendizaje por sí solas [18]. Un ejemplo de esto se puede ver en [10], donde se combina la optimización por enjambre de partículas para realizar tareas de optimización

de parámetros usados en técnicas de oversampling (SMOTE) y selección de instancias para realizar undersampling a una base de datos.

Otro ejemplo es el que se muestra en [22] donde se hace uso del algoritmo *Cross generational elitist selection Heterogeneous recombination Cataclysmic mutation algorithm* (CHC) para realizar una selección de instancias de la clase mayoritaria de tal forma que se realice un proceso de undersampling y las instancias seleccionadas sean lo más representativas posibles.

Tabla 1. Representación del individuo.

Método Oversampling	$\% \rho_O$	k_O	μ	Método Undersampling	$\% \rho_U$	k_U	Orden
{1, 2, 3}	[0, 1]	{1, ..., 30}	{1, ..., 20}	{1, 2, 3}	[0, 1]	{1, ..., 30}	{1, 2, 3, 4}

También se ha usado el algoritmo de Evolución Diferencial para inspirar la creación de nuevas instancias en procesos de oversampling de la misma forma en la que ED explora el espacio de soluciones [7,?]. Por último, en [12] se propone el uso de la tercera versión del algoritmo *Non-Dominated Sorting Genetic Algorithm* (NSGA-III) para seleccionar un método de balanceo con sus respectivos hiperparámetros dada una base de datos desbalanceada, mediante una aproximación de mucho objetivos, donde se hace uso de cinco métricas relacionadas con distintas formas para medir el desempeño de los métodos.

Dentro de todas estas propuestas y aproximaciones se puede notar que a excepción de la última mencionada, no se realiza una selección entre distintos métodos de balanceo, simplemente se toma un método y se optimizan sus hiperparámetros o se usa el algoritmo bio-inspirado para seleccionar un número de instancias de forma que se realice undersampling al eliminar las instancias no seleccionadas. En la última propuesta mencionada ([12]) se realiza una selección del método de balanceo, sin embargo no se puede seleccionar la aproximación de hybridsampling, puesto que sólo se puede seleccionar un método de todas las opciones que se muestran.

Dado lo anterior, proponemos la selección de un método de balanceo con el cual se pueda introducir una base de datos desbalanceada y por medio de Evolución Diferencial, se realice la selección de una de las tres distintas aproximaciones basadas en muestreo (hybridsampling, oversampling y undersampling), así como la elección de sus respectivos hiperparámetros.

Para evaluar el método propuesto se seleccionan diez bases de datos desbalanceadas con un distinto nivel de desbalanceo medido por el radio de desbalanceo (IR) y se compara con todos los métodos usados de forma individual con los parámetros por defecto, de igual forma se comparan con dos algoritmos de balanceo que usan la aproximación hybridsampling con sus parámetros por defecto.

El resto de este trabajo se organiza como sigue. La Sección 2 presenta la propuesta de forma detallada tomando en cuenta la representación de soluciones, los métodos de balanceo seleccionados y la función de evaluación que se usa. La

Sección 3 muestra los experimentos y resultados, donde se habla de las bases de datos seleccionadas, los parámetros utilizados, así como la forma en que se desarrolló la experimentación. En la Sección 4 se realiza una discusión de los resultados obtenidos. Por último, en la Sección 5 se presentan las conclusiones de los resultados obtenidos y se habla del trabajo futuro.

2. Propuesta

El método propuesto para seleccionar la técnica de balanceo basada en muestreo para una base de datos dada, hace uso de Evolución Diferencial para explorar el espacio de soluciones, dicho espacio está compuesto por la combinación de las técnicas disponibles con sus respectivos parámetros. En esta sección se muestra un resumen del algoritmo evolutivo usado, así como dos puntos importantes dentro de esta propuesta que son la representación y la función de aptitud o desempeño.

2.1. Evolución diferencial

La evolución diferencial es un algoritmo de optimización propuesto por Storn y Price en 1995 [16]. Este algoritmo de optimización pertenece a la clase de algoritmos basados en poblaciones ya que se explora el espacio de búsqueda mediante la creación aleatoria de un conjunto de soluciones llamado población y mediante la aplicación de operadores de variación se generan nuevas soluciones para muestrear el espacio de búsqueda. Las operaciones que se aplican a los individuos de la población son: mutación, cruza y selección.

Las primeras dos operaciones manejan a los individuos representados mediante vectores y la última operación requiere de la evaluación de los individuos en una función que determinará el valor a optimizar, esta función se llama función de aptitud o desempeño. En el Algoritmo 1 se detalla el procedimiento utilizado por Evolución Diferencial. Dadas las necesidades de estas operaciones, se requiere una forma de representar a los individuos mediante un vector y establecer una función de desempeño para evaluar las soluciones, estos dos puntos se abordarán a continuación.

2.2. Representación

En el vector, las posiciones 1 a la 4 seleccionan el método de oversampling y sus respectivos parámetros, de la 5 a la 7 seleccionan el método de undersampling con sus parámetros, y por último, la posición 8 selecciona el orden en el que se aplicarán estos métodos. Dado que en el algoritmo de Evolución Diferencial los individuos se representan con vectores de números reales, para las variables que toman valores discretos se considera solamente la parte entera del valor seleccionado. La forma en que se representan los individuos está establecida por un vector con 8 elementos como se muestra en la Tabla 1. Para oversampling se

Algoritmo 1: Evolución Diferencial

NP: Tamaño de la población
 NG: Número de generaciones
 F: Factor de mutación
 CR: Probabilidad de cruza
 Se genera una población de tamaño NP de forma aleatoria con dimensión D dentro de los límites establecidos
 $G = 0$
mientras $G < NG$ **hacer**
 para $i = 1$ **a** NP **hacer**
 Seleccionar tres individuos X_{r_1} , X_{r_2} y X_{r_3} distintos entre sí y que no sean el elemento i seleccionado
 Inicia Mutación:
 $V_i^G = X_{r_1}^G + F \cdot (X_{r_2}^G - X_{r_3}^G)$
 Inicia Cruza: $ea(1, D)$ es un entero aleatorio entre 1 y D, $a[0, 1)$ es un número aleatorio entre 0 y 1
 $j_{aleatorio} = ea(1, D)$
 para $j = 1$ **a** D **hacer**
 si $a[0, 1) \leq CR$ **o** $j = j_{aleatorio}$ **entonces**
 $U_{i,j}^G = V_{i,j}^G$
 en otro caso
 $U_{i,j}^G = X_{i,j}^G$
 fin
 fin
 Inicia Selección: $f(X)$ es la evaluación de X en la función de desempeño
 si $f(U_i^G) \leq f(X_i^G)$ **entonces**
 $X_i^{G+1} = U_i^G$
 en otro caso
 $X_i^{G+1} = X_i^G$
 fin
 fin
 $G = G + 1$
fin
devolver Mejor individuo de la población

seleccionaron los métodos 1: Random Oversampling, 2: SMOTE y 3: Borderline-SMOTE. Estos tres métodos de balanceo reciben como parámetro el porcentaje de radio de balanceo ($\% \rho_O$) que va a usar, lo que se traduce en el número de instancias por aumentar. El elemento k_O se usa en SMOTE y Borderline-SMOTE, donde representa el número de k vecinos a tomar en cuenta para aplicar el método.

Por último, Borderline-SMOTE usa el parámetro μ para seleccionar el número de puntos a tomar en cuenta para detectar si se encuentra un borde. Para undersampling, se seleccionaron los métodos 1: Random Undersampling, 2: NearMiss y 3: Cluster Centroids. De la misma forma que sucede en oversampling,

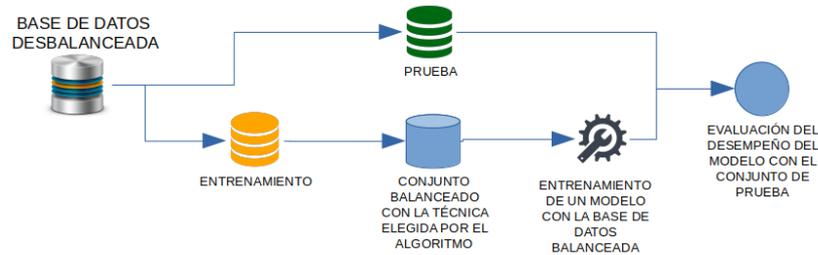


Fig. 1. Diagrama sobre la forma en que se evalúa el desempeño de una solución.

Tabla 2. Matriz de confusión.

	Clasificado Positivo	Clasificado Negativo
Positivo	VP	FN
Negativo	FP	VN

los tres métodos reciben el porcentaje de radio de balanceo ($\% \rho_U$) que se traduce en el número de instancias por disminuir. También se usa el parámetro k_U como el número de vecinos a tomar en cuenta para el método NearMiss.

Finalmente, el parámetro *Orden* toma cuatro valores posibles para indicar el orden de aplicación de los métodos de la siguiente forma: 1 = se aplica oversampling y 2 = se aplica undersampling. Para la aproximación de hybridsampling, se hace notar que el orden en que se aplican las técnicas de oversampling y undersampling pueden tener efectos distintos debido a las diferentes técnicas que se eligieron.

Un ejemplo de esto lo podemos notar si seleccionamos Borderline-SMOTE como técnica de oversampling, puesto que los límites encontrados entre la clase minoritaria y la clase mayoritaria original, pueden ser distintos a los límites entre la clase minoritaria y la clase mayoritaria luego de reducir el número de puntos al aplicar una técnica de undersampling.

Dado esto, se propone tomar los valores para hybridsampling de la siguiente forma: 3 = se aplica primero oversampling y después undersampling, 4 = se aplica primero undersampling y después oversampling. Con esto abordamos las tres aproximaciones posibles de los métodos de balanceo basados en muestreo (oversampling, undersampling, hybridsampling).

2.3. Función de desempeño

Debido a que, como todo algoritmo evolutivo, el algoritmo de Evolución Diferencial requiere una función para evaluar el desempeño de las soluciones elegidas, se hace uso de una medida para evaluar el desempeño de clasificación cuando se realiza el balanceo. Para poder realizar la evaluación de una solución,

se realizan distintos procedimientos. El primer paso es dividir la base de datos en dos conjuntos, un conjunto de entrenamiento y un conjunto de prueba, ambos conjuntos deben contener de forma proporcional ambas clases (minoritaria y mayoritaria).

Tabla 3. Bases de datos seleccionadas.

Nombre	#Atributos	#Instancias	#Ins. Minoritaria	#Ins. Mayoritaria	IR
glass1	9	214	76	138	1.82
pima	8	768	268	500	1.87
iris0	4	150	50	100	2
haberman	3	306	81	225	2.78
vehicle1	18	846	217	629	2.9
ecoli2	7	336	52	284	5.46
yeast0359vs78	8	506	50	456	9.12
ecoli0146vs5	6	280	20	260	13
glass4	9	214	13	201	15.47
yeast1458vs7	8	693	30	663	22.1

El siguiente paso es aplicar el método de balanceo seleccionado al conjunto de entrenamiento, ya sea la aplicación de undersampling, oversampling o ambas aproximaciones en los distintos ordenes disponibles con los respectivos hiperparámetros elegidos. Una vez que se balancea el conjunto de entrenamiento, se usa para entrenar un modelo de clasificación. Por último, se evalúa el desempeño del clasificador introduciendo el conjunto de prueba, al cual no se le aplicó ningún método de balanceo.

Este procedimiento se puede observar en la Figura 1. Una vez obtenidas las clases asignadas por el clasificador, se obtiene la matriz de confusión como la que se muestra en la Tabla 2 donde se incluye el número de elementos que tenían etiqueta positiva y el clasificador les asignó la etiqueta positiva (VP), el número de elementos que tenían etiqueta positiva, pero el clasificador les asignó la etiqueta negativa (FN), el número de elementos que tenían etiqueta negativa, pero el clasificador les asignó la etiqueta positiva (FN) y por último, el número de elementos que tenían etiqueta negativa y que el clasificador les asignó la etiqueta negativa (VN).

La construcción de esta matriz de confusión permite evaluar mediante distintas medidas de desempeño, en nuestro caso se hará uso de la G-Media (G-Mean), la cual combina medidas de sensibilidad (1) y especificidad (2), por lo que existe un balance entre un buen desempeño de clasificación medido tanto para la clase mayoritaria como para la clase minoritaria.

Esta medida indica un buen desempeño en la clasificación mientras más grande sea, en cambio, mientras más pequeño sea el valor de la G-Mediana, el clasificador se desempeña de peor forma en la clasificación de ambas clases.

Esta medida es recomendada para evaluar el desempeño de clasificación en bases de datos desbalanceadas [1,17]:

$$\text{sensitividad} = \frac{VP}{VP + FN}, \quad (1)$$

$$\text{especificidad} = \frac{VN}{VN + FP}, \quad (2)$$

$$\text{G-Media} = \sqrt{\text{sensitividad} \times \text{especificidad}}. \quad (3)$$

3. Experimentos y resultados

En esta sección, se muestran los experimentos realizados con la propuesta presentada anteriormente, así como las configuraciones asignadas y las bases de datos usadas. Las bases de datos seleccionadas para los experimentos corresponden a la sección de bases de datos desbalanceadas pertenecientes al *KEEL-dataset repository* [2]. En la Tabla 3 se pueden apreciar las bases de datos que se seleccionaron, así como sus correspondientes números de instancias pertenecientes a las clases mayoritaria y minoritaria.

Como se puede notar, para realizar los experimentos hemos elegido un número variable de radio de balanceo (IR), por lo que podemos ver el desempeño de nuestra propuesta en diferentes grados de desbalanceo. Para realizar la selección de los valores de los parámetros necesarios para el algoritmo de Evolución Diferencial, se hizo uso del lenguaje de programación R donde se encuentra la paquetería *irace* [11] para la configuración automática del algoritmo.

Luego de la ejecución de *irace*, se obtuvieron los siguientes parámetros: Número de población = 44, Número de generaciones = 139, F = 1.46, CR = 0.06. Con estos parámetros mencionados, se realizaron 10 ejecuciones independientes. Todas estas pruebas fueron realizadas haciendo uso de Python y Jupyter Notebooks en una computadora portátil que cuenta con un procesador Intel Core i7-8750H con 16GB de RAM.

Para la comparación de resultados, se decidieron aplicar los métodos de balanceo individualmente con sus parámetros por defecto, así como agregar dos funciones que cuentan con la aproximación de hybridsampling (SMOTEENN y SMOTETomek), de igual forma, con sus parámetros por defecto. Los dos algoritmos de clasificación que se seleccionaron fueron Árbol de decisión (profundidad = 5) y Naive Bayes (parámetros por defecto). Con el fin de analizar el desempeño de la Evolución Diferencial como método de búsqueda, se hace uso de tres diferentes estrategias: DE/best/1/bin, DE/best/1/exp/ y DE/rand/1/exp. Esto con el fin de analizar los resultados obtenidos por la aplicación de la propuesta con distintas estrategias de búsqueda.

Los resultados obtenidos se presentan en las Tablas 4 y 5. Estos resultados se analizaron con una prueba de Shapiro-Wilk para verificar si provienen de una población que se distribuye de una forma normal, posteriormente se hace uso de la prueba Kruskal-Wallis para analizar si los individuos provienen de

la misma población y confirmar diferencias significativas. Con los resultados obtenidos de las pruebas anteriores, se procedió a realizar la prueba posthoc de Dunn-Sidak. Los resultados se expresan como la media de la G-Media más la diferencia a los cuartiles 25 y 75.

4. Discusión

Como podemos observar en la Tabla 4, existen diferencias de desempeño entre la población y se distinguen dos grupos, un grupo (*a*) formado por todos los métodos con los parámetros por defecto, incluidas las aproximaciones de hybridsampling y el grupo (*b*) donde se encuentra la propuesta que realizamos nosotros con las distintas estrategias de Evolución Diferencial.

Dados estos dos grupos, podemos darnos cuenta que existe una diferencia significativa entre el desempeño del grupo (*a*) y el desempeño del grupo (*b*). En esta tabla podemos observar que en todas las bases de datos, nuestra propuesta obtiene un mejor desempeño en términos de la G-Media.

En la Tabla 5 podemos observar que sucede lo mismo que en la Tabla 4, a excepción de los resultados obtenidos con la base de datos *ecoli0146vs5* donde se crean tres grupos, el grupo (*a*) que se desempeña peor que el grupo (*b*) y el grupo (*ab*) que se desempeña de forma similar a los dos grupos antes mencionados. En el resto de bases de datos, el grupo (*b*) que corresponde a nuestra propuesta, es superior en desempeño.

5. Conclusiones y trabajo futuro

Ya que no existe un criterio específico para la selección de un método de balanceo o sus hiperparámetros, es necesario encontrar las opciones que mejor se adapten a una base de datos dada, ya que no todas las bases de datos cumplen con las mismas condiciones. En este trabajo se presentó una propuesta que es capaz de seleccionar el mejor método de balanceo y sus hiperparámetros mediante Evolución Diferencial para una base de datos en particular, de tal forma que se mejora el desempeño de una medida, en este caso se selecciona un método de balanceo que optimiza la G-Media, lo cual se traduce a un buen desempeño de clasificación tanto en la clase mayoritaria como en la minoritaria.

Aunque se aborda el problema de desbalanceo, sigue siendo un tema extenso, por lo que quedan temas para abordar en el futuro como el estudio en más bases de datos, el estudio de otras medidas distintas a la G-Media, el estudio en bases de datos con más de dos clases y el uso de otros algoritmos de optimización.

Agradecimientos. El primer autor agradece el apoyo de CONACyT mediante una beca para la realización de estudios de posgrado en el Instituto de Investigaciones en Inteligencia Artificial de la Universidad Veracruzana.

Tabla 4. Valores de la G-Media obtenidos haciendo uso del Árbol de decisión como clasificador. Los resultados obtenidos se expresan como el promedio del resultado de 10 evaluaciones. Los valores entre paréntesis representan la diferencia de la media al cuartil 25 de las 10 evaluaciones (lado izquierdo) y la diferencia de la media al cuartil 75 (lado derecho). En las últimas dos columnas se muestran las estadísticas relacionadas a la prueba Kruskal-Wallis. Las etiquetas corresponden a los grupos donde se observan diferencias siguiendo la prueba de Dunn-Sidak.

Nombre	RO	SMOTE	B-SMOTE	RU	NM	CC	SMOTEENN	SMOTEImek	DE/best/1/bin	DE/best/1/exp	DE/rand/1/bin	Estadísticas	
												H	p-value
glass1	0.7338 ^a (-0.0202,+0.0076)	0.7105 ^a (-0.022,+0.0114)	0.6742 ^a (-0.055,+0.0523)	0.7285 ^a (-0.0137,+0.0169)	0.5553 ^a (-0.0149,+0.0143)	0.6431 ^a (-0.0442,+0.0475)	0.7056 ^a (-0.0146,+0.0148)	0.6954 ^a (-0.0851,+0.0434)	0.9247 ^a (-0.0067,+0.0072)	0.9282 ^b (-0.0083,+0.0056)	0.9274^b (-0.0087,+0.005)	86.1079	3.155e-14
pinna	0.7267 ^a (-0.0169,+0.0127)	0.7348 ^a (-0.0277,+0.0274)	0.7366 ^a (-0.0137,+0.019)	0.6977 ^a (-0.0113,+0.0032)	0.7117 ^a (-0.0045,+0.0019)	0.6873 ^a (-0.01,+0.014)	0.7311 ^a (-0.0064,+0.0101)	0.7239 ^a (-0.0259,+0.0253)	0.8281^a (-0.0054,+0.0069)	0.8274 ^a (-0.0053,+0.0044)	0.8275 ^a (-0.0046,+0.005)	77.9891	1.2426e-12
iris0	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	NA	NA
haberman	0.5595 ^a (-0.0421,+0.0296)	0.5787 ^a (-0.0148,+0.0133)	0.5954 ^a (-0.026,+0.0278)	0.5943 ^a (-0.0106,+0.0045)	0.5599 ^a (-0.0271,+0.0253)	0.573 ^a (-0.0168,+0.0038)	0.5592 ^a (-0.0234,+0.0185)	0.5668 ^a (-0.0195,+0.0183)	0.8077^a (-0.0062,+0.0057)	0.8034 ^a (-0.0105,+0.0092)	0.8018 ^a (-0.0031,+0.0036)	72.5177	1.441e-11
vehicle1	0.7461 ^a (-0.0158,+0.0139)	0.75 ^a (-0.0128,+0.0113)	0.7432 ^a (-0.0221,+0.0148)	0.7292 ^a (-0.0125,+0.0172)	0.6738 ^a (-0.0233,+0.0245)	0.6815 ^a (-0.0229,+0.0226)	0.7345 ^a (-0.0042,+0.0079)	0.7481 ^a (-0.0076,+0.0083)	0.8496^a (-0.0073,+0.0061)	0.848 ^a (-0.0076,+0.0009)	0.8435 ^a (-0.0039,+0.0028)	88.3314	1.1665e-14
ecoli2	0.916 ^a (+0.0028,+0.0125)	0.9271 ^a (-0.0077,+0.0028)	0.8377 ^a (-0.0233,+0.0233)	0.886 ^a (-0.0238,+0.0187)	0.6964 ^a (-0.0266,+0.0286)	0.847 ^a (-0.0134,+0.0287)	0.8324 ^a (-0.0122,+0.0113)	0.831 ^a (-0.0109,+0.0142)	1.0^b (-0,+0)	1.0^b (-0,+0)	0.9988 ^b (+0.0011,+0.0011)	99.2312	7.7655e-17
yeast0359vs78	0.6268 ^a (-0.065,+0.04)	0.6636 ^a (-0.0238,+0.0185)	0.6446 ^a (-0.0092,+0.0006)	0.6517 ^a (-0.035,+0.0224)	0.7083 ^a (-0.0072,+0.0048)	0.5359 ^a (-0.0301,+0.0505)	0.7056 ^a (-0.0115,+0.0196)	0.6697 ^a (-0.0414,+0.0383)	0.8919^a (-0.0096,+0.0083)	0.8883 ^a (-0.0106,+0.0044)	0.8836 ^a (-0.006,+0.0054)	87.6765	1.5452e-14
ecoli1046vs5	0.8155 ^a (-0.0095,+0.0095)	0.9511 ^a (-0.0334,+0.0295)	0.9539 ^a (-0.0514,+0.0332)	0.8082 ^a (-0.0292,+0.0457)	0.7679 ^a (-0.0,+0.0)	0.8359 ^a (-0.0038,+0.0038)	0.9409 ^a (-0.0443,+0.0445)	0.9105 ^a (-0.0213,+0.0094)	1.0^b (-0,+0)	1.0^b (-0,+0)	1.0^b (-0,+0)	98.9884	8.6841e-17
glass4	0.6989 ^a (-0.0035,+0.0024)	0.6983 ^a (-0.0029,+0.0029)	0.6995 ^a (-0.0026,+0.0018)	0.761 ^a (-0.0775,+0.0576)	0.866 ^a (-0.0,+0.0)	0.5901 ^a (+0.0122,+0.0372)	0.6942 ^a (-0.0047,+0.0012)	0.696 ^a (-0.006,+0.0038)	1.0^b (-0,+0)	1.0^b (-0,+0)	1.0^b (-0,+0)	92.5082	1.7014e-15
yeast1458vs7	0.4075 ^a (-0.0887,+0.1354)	0.5225 ^a (-0.0313,+0.0238)	0.6068 ^a (-0.0658,+0.0353)	0.4726 ^a (-0.0816,+0.0973)	0.5353 ^a (-0.0437,+0.0396)	0.5255 ^a (-0.0181,+0.0105)	0.5471 ^a (-0.0388,+0.0451)	0.5823 ^a (-0.0373,+0.0141)	0.8862^a (-0.0109,+0.0119)	0.8723 ^a (-0.0132,+0.0099)	0.8751 ^a (-0.0115,+0.0059)	75.23	4.2916e-12

Tabla 5. Valores de la G-Media obtenidos haciendo uso del algoritmo Naive Bayes como clasificador. Los resultados obtenidos se expresan como el promedio del resultado de 10 evaluaciones. Los valores entre paréntesis representan la diferencia de la media al cuartil 25 de las 10 evaluaciones (lado izquierdo) y la diferencia de la media al cuartil 75 (lado derecho). En las últimas dos columnas se muestran las estadísticas relacionadas a la prueba Kruskal-Wallis. Las etiquetas corresponden a los grupos donde se observan diferencias siguiendo la prueba de Dunn-Sidak.

Nombre	RO	SMOTE	B-SMOTE	RU	NM	CC	SMOTEENN	SMOTETomek	DE/best/1/bin	DE/best/1/exp	DE/rand/1/bin	Estadísticas	
												H	p-value
glass1	0.5941 ^a (-0.0268,+0.0318)	0.6113 ^a (-0.0528,+0.054)	0.6129 ^a (-0.0192,+0.0343)	0.6001 ^a (-0.0425,+0.0365)	0.6132 ^a (-0.0037,+0.0422)	0.6129 ^a (-0.0522,+0.0091)	0.6165 ^a (-0.0222,+0.0405)	0.6397 ^{ac} (-0.0446,+0.0512)	0.8402 ^b (-0.0097,+0.0050)	0.8392 ^b (-0.0101,+0.0122)	0.8272 ^b (-0.0063,+0.0049)	67.5398	1.3205e-10
pana	0.6981 ^a (-0.0132,+0.0157)	0.7111 ^a (-0.0169,+0.0144)	0.6833 ^a (-0.0161,+0.0209)	0.7131 ^a (-0.0225,+0.0223)	0.6918 ^a (-0.0243,+0.0206)	0.697 ^{ac} (-0.0287,+0.0216)	0.73 ^a (-0.025,+0.0309)	0.7082 ^a (-0.0077,+0.0123)	0.8290 ^b (-0.0081,+0.0070)	0.8270 ^b (-0.0036,+0.0026)	0.8282^b (-0.004,+0.0021)	70.1248	4.1943e-11
iris0	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	1.0(-0.0,+0.0)	NA	NA
haberman	0.4388 ^a (-0.0443,+0.051)	0.3779 ^a (-0.0802,+0.0684)	0.4616 ^a (-0.0188,+0.0272)	0.4186 ^a (-0.0564,+0.0519)	0.432 ^a (-0.036,+0.0393)	0.3762 ^a (-0.0776,+0.0572)	0.404 ^a (-0.0407,+0.0607)	0.4494 ^a (-0.0379,+0.0639)	0.7083 ^a (-0.0029,+0.0030)	0.7016 ^a (-0.0045,+0.0053)	0.7092^b (-0.0121,+0.0014)	69.7754	4.8991e-11
vehicle	0.6844 ^a (-0.048,+0.0428)	0.687 ^a (-0.0224,+0.0178)	0.6906 ^a (-0.0236,+0.015)	0.6847 ^a (-0.0047,+0.0092)	0.6848 ^a (-0.0148,+0.0193)	0.6738 ^a (-0.0124,+0.0097)	0.6601 ^a (-0.0284,+0.043)	0.6867 ^a (-0.0074,+0.0251)	0.7914 ^a (-0.0107,+0.0079)	0.8118 ^a (-0.0029,+0.0048)	0.8129^b (-0.0061,+0.0044)	67.8269	1.1633e-10
ecoli2	0.5163 ^a (-0.1415,+0.1504)	0.4756 ^a (-0.1346,+0.1557)	0.4858 ^a (-0.1346,+0.1557)	0.5311 ^a (-0.1117,+0.1577)	0.4553 ^a (-0.1153,+0.1089)	0.576 ^a (-0.1238,+0.1102)	0.4076 ^a (-0.0646,+0.0014)	0.4894 ^a (-0.1219,+0.0833)	0.8737 ^a (-0.0039,+0.0049)	0.8838^b (-0.0164,-0.0066)	0.8779 ^a (-0.0051,+0.0006)	67.9555	1.0984e-10
yeast0359vs78	0.351 ^a (-0.0447,+0.0518)	0.349 ^a (-0.0373,+0.0179)	0.3403 ^a (-0.0384,+0.0389)	0.3135 ^a (-0.0368,+0.0362)	0.3018 ^a (-0.0307,+0.04)	0.3518 ^a (-0.042,+0.0065)	0.3338 ^a (-0.0141,+0.0262)	0.3251 ^a (-0.0536,+0.0245)	0.6989 ^a (-0.0233,+0.0233)	0.7129 ^a (-0.0093,+0.0093)	0.7328^b (-0.0106,-0.0025)	69.1378	6.5030e-11
ecoli10_46vs5	0.8974 ^a (-0.0627,+0.0745)	0.8257 ^{ab} (-0.0197,+0.0664)	0.847 ^a (-0.0358,+0.0266)	0.8504 ^{ab} (-0.0278,+0.0551)	0.824 ^a (-0.0638,+0.0463)	0.8362 ^{ab} (-0.038,+0.0392)	0.792 ^a (-0.0345,+0.0434)	0.8138 ^a (-0.0806,+0.0591)	1.0 ^a (-0.0,+0.0)	1.0 ^a (-0.0,+0.0)	1.0 ^a (-0.0,+0.0)	68.3224	9.3368e-11
glass4	0.4834 ^a (-0.3605,+0.3083)	0.3393 ^a (-0.3393,+0.2589)	0.3069 ^a (-0.3069,+0.1848)	0.5123 ^a (-0.028,+0.0123)	0.4919 ^a (-0.37,+0.3153)	0.3434 ^a (-0.3434,+0.234)	0.3762 ^a (-0.2397,+0.1165)	0.5434 ^a (-0.0831,+0.1458)	0.9859^b (-0.0024,+0.0037)	0.9851 ^b (-0.0016,-0.0016)	0.9851 ^b (-0.0016,-0.0016)	70.2971	3.8848e-11
yeast1458vs7	0.292 ^a (-0.0392,+0.0399)	0.2753 ^a (-0.0249,+0.0169)	0.3039 ^a (-0.0204,+0.0111)	0.3249 ^a (-0.0242,+0.0187)	0.2914 ^a (-0.0079,+0.0194)	0.293 ^a (-0.0254,+0.0078)	0.294 ^a (-0.0172,+0.021)	0.2968 ^a (-0.0292,+0.0081)	0.5051 ^a (-0.0039,+0.0031)	0.5082^b (-0.0069,+0.0029)	0.5055 ^a (-0.0042,+0.0044)	69.5501	5.4149e-11

Referencias

1. Akosa, J. S.: Predictive accuracy: A misleading performance measure for highly imbalanced data (2017)
2. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Multiple Valued Log. Soft Comput.*, vol. 17, no. 2-3, pp. 255–287 (2011)
3. Cao, L., Zhai, Y.: Imbalanced data classification based on a hybrid resampling SVM method, pp. 1533–1536 (2015) doi: 10.1109/UIC-ATC-ScalCom-CBDCCom-IoP.2015.275
4. Gao, Z., Zhang, L. F., Chen, M. Y., Hauptmann, A., Zhang, H., Cai, A. N.: Enhanced and hierarchical structure algorithm for data imbalance problem in semantic extraction under massive video dataset. *Multimedia Tools Appl.*, vol. 68, no. 3, pp. 641–657 (2014) doi: 10.1007/s11042-012-1071-7
5. Hodge, V., Austin, J.: A survey of outlier detection methodologies. *Artificial Intelligence Review*, vol. 22, pp. 85–126 (2004) doi: 10.1023/B:AIRE.0000045502.10941.a9
6. Kaur, H., Pannu, H. S., Malhi, A. K.: A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.*, vol. 52, no. 4 (2019) doi: 10.1145/3343440
7. Kaya, E., Korkmaz, S., Sahman, M. A., Cinar, A. C.: DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets. *Expert Systems with Applications*, vol. 169, pp. 114482 (2021) doi: 10.1016/j.eswa.2020.114482
8. Korkmaz, S., Şahman, M. A., Cinar, A. C., Kaya, E.: Boosting the oversampling methods based on differential evolution strategies for imbalanced learning. *Applied Soft Computing*, vol. 112, pp. 107787 (2021) doi: 10.1016/j.asoc.2021.107787
9. Kubat, M., Holte, R., Matwin, S.: A survey of outlier detection methodologies. *Machine Learning*, vol. 30, pp. 195–215 (1998) doi: 10.1023/A:1007452223027
10. Li, J., Fong, S., Wong, R. K., Chu, V. W.: Adaptive multi-objective swarm fusion for imbalanced data classification. *Information Fusion*, vol. 39, pp. 1–24 (2018) doi: 10.1016/j.inffus.2017.03.007
11. López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Stützle, T., Birattari, M.: The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, vol. 3, pp. 43–58 (2016) doi: 10.1016/j.orp.2016.09.002
12. Miranda, P. B., Morais, R. F., Silva, R. M.: Using a many-objective optimization algorithm to select sampling approaches for imbalanced datasets. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–7 (2018) doi: 10.1109/CEC.2018.8477988
13. Nepal, S., Pathan, M.: Security, privacy and trust in cloud systems. Springer Publishing Company, Incorporated (2013)
14. Ramentol, E., Gondres, I., Lajes, S., Bello, R., Caballero, Y., Cornelis, C., Herrera, F.: Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: The SMOTE-FRST-2T algorithm. *Engineering Applications of Artificial Intelligence*, vol. 48, pp. 134–139 (2016) doi: 10.1016/j.engappai.2015.10.009
15. Sahin, Y., Bulkan, S., Duman, E.: A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, vol. 40, no. 15, pp. 5916–5923 (2013) doi: 10.1016/j.eswa.2013.05.021

16. Storn, R., Price, K.: Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 23 (1995)
17. Susan, S., Amitesh: The balancing trick: Optimized sampling of imbalanced datasets—a brief survey of the recent state of the art. *Engineering Reports*, vol. 3 (2021) doi: 10.1002/eng2.12298
18. Telikani, A., Tahmassebi, A., Banzhaf, W., Gandomi, A. H.: Evolutionary machine learning: A survey. *ACM Comput. Surv.*, vol. 54, no. 8 (2021) doi: 10.1145/3467477
19. Upadhyay, K., Kaur, P., Prasad, S.: State of the art on data level methods to address class imbalance problem in binary classification, vol. 8, pp. 975–903 (2021)
20. Wei, W., Li, J., Cao, L., Ou, Y., Chen, J.: Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, vol. 16 (2013) doi: 10.1007/s11280-012-0178-0
21. Woods, K. S., Doss, C. C., Bowyer, K. W., Solka, J. L., Priebe, C. E., Kegelmeyer, W. P.: Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 6, pp. 1417–1436 (1993) doi: 10.1142/S0218001493000698
22. Zhu, Y., Yan, Y., Zhang, Y., Zhang, Y.: EHSO: Evolutionary hybrid sampling in overlapping scenarios for imbalanced learning. *Neurocomputing*, vol. 417, pp. 333–346 (2020) doi: 10.1016/j.neucom.2020.08.060