El Comité de Disertación de Armando Isaac Bolívar Velazco
certifica que esta es la versión aprobada de la siguiente disertación:

Método Sistemático para Mitigar el Problema de las Clases No Balanceadas con Alta Dimensionalidad y Solapamiento en Big Data

Comité:

Dr. Vicente García Jiménez, Asesor

Dr. Rogelio Florencia Juárez, Coasesor

Dr. Roberto Alejo Eleuterio, Coasesor

Dr. Osslan Osiris Vergara Villegas

Dr. Gilberto Rivera Zárate

Método Sistemático para Mitigar el Problema de las Clases No Balanceadas con Alta Dimensionalidad y Solapamiento en Big Data

por

M. en I.E. Armando Isaac Bolívar Velazco

DISERTACIÓN

Presentada al Comité Tutotal de

la Universidad Autónoma de Ciudad Juárez

como Requisito Parcial para Obtener el Grado de

DOCTOR EN CIENCIAS

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Noviembre 2025

Resumen

Método Sistemático para Mitigar el Problema de las Clases No Balanceadas con Alta Dimensionalidad y Solapamiento en Big Data

Publicación No. _____

Armando Isaac Bolívar Velazco, Doctor en Ciencias
Universidad Autónoma de Ciudad Juárez, 2025

Asesores: Dr. Vicente García Jiménez
Dr. Rogelio Florencia Juárez
Dr. Roberto Alejo Eleuterio

Uno de los principales desafíos en el análisis de datos radica en la magnitud de la información involucrada, tanto en el número de observaciones (instancias) como en la cantidad de variables (atributos). El término *big data* hace referencia a volúmenes de datos que superan la capacidad de procesamiento de un sistema informático individual. Por su parte, la alta dimensionalidad se relaciona con la existencia de un número tan elevado de atributos que dificulta el análisis efectivo y la adecuada generalización de los modelos.

Adicionalmente, los conjuntos de datos presentan otras complejidades, como la presencia de clases no balanceadas, en las que un número significati-

vamente menor de instancias es de particular interés para la predicción, y el solapamiento entre clases, que complican la identificación precisa de instancias.

En esta tesis, se realizó una investigación exhaustiva sobre las técnicas empleadas tanto en entornos de *big data* como en datos tradicionales (es decir, aquellos que no necesariamente implican grandes volúmenes de instancias) con el propósito de examinar cómo abordan los problemas de alta dimensionalidad, desbalance y solapamiento de clases. A partir de este análisis, se propone un método sistemático, conformado por una secuencia de técnicas aplicadas en un orden específico, diseñado para mitigar los efectos adversos de estas complejidades en conjuntos de datos utilizados en sistemas de clasificación.

En primer lugar, se emplean distancias fraccionarias en espacios de disimilitud para abordar los efectos de la alta dimensionalidad. Posteriormente, se implementa una búsqueda del vecino más cercano de manera distribuida en distintos nodos de cómputo, que sirve de base para una versión adaptada de SMOTE orientada a entornos *big data*, con el propósito de realizar sobremuestreo y de esta forma tratar el problema del desbalance de clases. Asimismo, se incorpora una implementación de la edición de Wilson para *big data*, desarrollada en esta tesis, destinada a reducir el solapamiento entre clases.

El método propuesto no solo reduce la complejidad y el tamaño de los datos, sino que también mejora de forma estadísticamente significativa las tasas de clasificación.

Abstract

Systematic Method to Mitigate the Problem of Class Imbalance with High Dimensionality and Overlap in Big Data

Publication No. _____

Armando Isaac Bolívar Velazco, Ph.D.
Universidad Autónoma de Ciudad Juárez, 2025

Supervisors: Dr. Vicente García Jiménez
Dr. Rogelio Florencia Juárez
Dr. Roberto Alejo Eleuterio

One of the main challenges in data analysis lies in the magnitude of the information involved, both in terms of the number of observations (instances) and the number of variables (attributes). The term "big data" refers to data volumes that exceed the processing capacity of a single computer system. Conversely, high dimensionality refers to the presence of such a large number of attributes that it hinders effective analysis and model generalization.

In addition, data sets often exhibit additional complexities, such as class imbalance, where a significantly smaller number of instances is of partic-

ular interest for prediction, and class overlap, which complicates the accurate discrimination between classes.

In this thesis, an exhaustive research was conducted on the techniques e,ployed in both big data and traditional data scenarios (i.e., those not necessarily involving massive volumes of instances), with the aim of examining how they address issues related to high dimensionality, class imbalance, and class overlap. Based on this analysis, a systematic method is proposed, consisting of a sequence of techniques applied in a specific order, designed to mitigate the adverse effects of these complexities on data sets used in classification systems.

First, fractional distances in dissimilarity spaces are used to address the effects of high dimensionality. Next, a distributed nearest neighbor search is implemented across multiple computing nodes, which supports an adapted version of SMOTE for big data environments to perform oversampling and thus address class imbalance. Additionally, a Big Data-compatible implementation of Wilson's editing, developed in this thesis, is incorporated to reduce class overlap.

The proposed method not only reduces data complexity and volume but also leads to statistically significant improvements in classification performance.

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Capítulo 1

Introducción

Desde el influyente trabajo de Fayyad et al. (1996), se ha subrayado la importancia de obtener conocimiento de manera sistemática a partir de los datos, empleando diversas teorías y herramientas de inteligencia artificial. Tecnologías emergentes como el Internet de las Cosas, la Web 2.0, la Industria 4.0 y las redes sociales generan diariamente un volumen masivo de datos que son almacenados por organizaciones tanto privadas como gubernamentales. El continuo incremento ha dado origen al paradigma conocido como *big data*, también referido como datos masivos o macrodatos.

Western Digital señala que, en 2020, cada persona generó un promedio de 1.7 megabytes de datos por segundo (Digital, 2020). Este volumen masivo de información presenta para las empresas una oportunidad crucial de crecimiento económico a través del análisis de grandes datos (Kubina et al., 2015). Un ejemplo de ello es *Hearst Corporation*, que procesa más de 30 terabytes de datos diarios para identificar tendencias en sus propiedades digitales (Amazon, 2021). El análisis de datos no solo se limita al sector empresarial, sino que también se ha empleado en áreas tan diversas como la detección de fraudes financieros, los diagnósticos médicos y la predicción de la calidad de los pro-

ductos (Mazurowski et al., 2008; Brennan, 2012; Zareapoor y Shamsolmoali, 2015; Jain et al., 2017).

Aunque no existe una definición precisa de big data, Beyer y Laney (2012) introdujeron las denominadas "Vs" para describir sus características fundamentales. Estas "Vs" se refieren inicialmente a tres propiedades físicas de los datos: volumen, velocidad y variedad. Con el tiempo, se añadieron dos más: valor y veracidad, términos que destacan la importancia de la calidad de los datos (Herrera, 2016; Luengo et al., 2020).

La generación y acumulación de grandes volúmenes de datos no garantiza resultados valiosos; por el contrario, puede dificultar la extracción de información útil. Por lo tanto, es fundamental obtener datos de alta calidad mediante procesos rigurosos de selección y preprocesamiento que permitan la construcción de algoritmos de inteligencia artificial eficientes. Este proceso se conoce como *Smart Data* (Luengo et al., 2020), y en el contexto de big data, va más allá de simplemente reducir la cantidad de datos. Implica garantizar que los datos seleccionados aporten valor y veracidad, evitando así la acumulación indiscriminada de información sin utilidad añadida.

Actualmente, las principales aplicaciones y herramientas de Inteligencia Artificial, hacen uso del aprendizaje automático o *Machine Learning*, no obstante, el aprendizaje automático presenta un desafío común en aplicaciones del mundo real, como la medicina (Lamba et al., 2021) y las finanzas (Brennan, 2012; Zareapoor y Shamsolmoali, 2015), el problema de las clases no balanceadas. Este fenómeno ocurre cuando existe una disparidad significativa

en el número de instancias entre diferentes clases, lo que lleva a que los algoritmos de clasificación tiendan a favorecer la clase predominante, ignorando las clases minoritarias que suelen ser las más relevantes para la problemática en cuestión. El costo de clasificar incorrectamente instancias de estas clases minoritarias es considerablemente mayor que el de las clases mayoritarias (Leevy et al., 2018). Este problema se agrava en conjuntos de datos masivos, donde emergen desafíos adicionales como la escalabilidad, la eficiencia computacional, cuestiones relacionadas con datos faltantes o redundantes, pequeños disyuntos, solapamiento y alta dimensionalidad. Aunque todas estas problemáticas han sido abordadas en la literatura, Ramírez-Gallego et al. (2021) señalan que el desafío específico de la alta dimensionalidad ha experimentado un crecimiento exponencial en los últimos años.

Tradicionalmente, las estrategias más comunes para abordar la disparidad en el tamaño de las clases han sido: (i) soluciones a nivel de datos (Fernández et al., 2017), y (ii) soluciones a nivel de algoritmos (López et al., 2015). Sin embargo, dada la complejidad de los datos en el contexto de big data, la comunidad científica ha desarrollado nuevas estrategias que consideran aspectos como la alta dimensionalidad y el solapamiento entre clases.

Diversas técnicas han sido presentadas para tratar el problema de las clases no balanceadas en big data. Por ejemplo, Fernández et al. (2017) proponen el uso de técnicas como el sobremuestreo aleatorio (ROS, *Random Oversampling*), el submuestreo aleatorio (RUS, *Random Undersampling*) y la técnica de sobremuestreo que genera instancias minoritarias artificiales basada

en la regla del vecino más cercano, conocida como SMOTE (*Syntetic Minority Oversampling Technique*). Por su parte, Juez-Gil et al. (2021a) sugieren una solución basada en sistemas múltiples de clasificación (SMC), que se considera una solución a nivel de algoritmos. Sin embargo, diseñar y aplicar técnicas enfocadas únicamente en la diferencia numérica sin tener en cuenta otras complejidades no asegura una solución efectiva al problema (Kuncheva et al., 2019; Rendón et al., 2020). Por ejemplo, el solapamiento puede tener un impacto más negativo en las tasas de clasificación que el propio desbalance (García et al., 2006). Además, la alta dimensionalidad puede dificultar que modelos de aprendizaje basados en distancia clasifiquen correctamente las instancias de las diferentes clases, ya que las distancias pierden relevancia en espacios de alta dimensionalidad.

No existe consenso en la literatura sobre un número específico de atributos que defina la alta dimensionalidad. Algunos autores, como Tahvili y Hatvani (2022), sugieren que un conjunto de datos se considera de alta dimensionalidad cuando el número de atributos supera al de ejemplos. Otros, como Kuo y Sloan (2005) y Fan y Lv (2010), sostienen que la alta dimensionalidad ocurre cuando el incremento en el número de atributos alcanza cientos o miles de atributos, lo que aumenta la complejidad del problema. En entornos de big data, donde la capacidad de procesar y almacenar datos es mayor, algunos investigadores se refieren a este fenómeno como *Big Dimensionality* (Zhai et al., 2014; Thudumu et al., 2020).

Para abordar el problema de las clases no balanceadas, considerando

las complejidades adicionales, la comunidad científica ha optado por enfoques que combinan diversas técnicas (Jiang y Ma, 2016; Tian et al., 2018; Deng et al., 2020). Estas estrategias aprovechan las fortalezas de diversos enfoques para aumentar el rendimiento y mejorar la capacidad de generalización de los modelos de aprendizaje automático. Un ejemplo de ello es el uso de técnicas de condensación y edición, aplicadas para eliminar instancias de la clase mayoritaria. Aunque estas técnicas no buscan equilibrar numéricamente las clases, sí contribuyen a mejorar la calidad de los datos al reducir el solapamiento, eliminando instancias ruidosas de la clase mayoritaria.

Esta tesis doctoral se enfoca en el problema de la clasificación en entornos de big data, con especial énfasis en la mitigación de los efectos adversos de las clases no balanceadas, el solapamiento entre clases y la alta dimensionalidad. En específico, se propone un método sistemático que se compone de un conjunto de técnicas de preprocesamiento adaptadas para big data aplicadas en un orden específico para mejorar las tasas de clasificación.

El método sistemático propuesto consta de tres fases:

1. Espacios de similitud para reducir la dimensionalidad. En esta fase se transforma el espacio de características a un espacio de disimilitud, utilizando distancias fraccionarias como métrica de similitud. Esta transformación genera un conjunto de datos con menor dimensionalidad.

2. Tratar el problema del desbalance de clases por medio de técnicas de sobre muestreo como SMOTE. En esta fase se aplica SMOTE en este

nuevo espacio transformado para balancear las clases, en el cual se usan distancias fraccionarias para mejorar la efectividad de la técnica.

3. Emplear técnicas de edición de datos para reducir el solapamiento entre clases por medio de la eliminación de instancias cercanas a la frontera de decisión. Finalmente, en esta fase, el conjunto de datos balanceado se refina mediante la técnica de edición ENN (*Edited Nearest Neighbor*), que elimina ruido y mitiga el solapamiento. Para mejorar el desempeño de la ENN se utilizan distancias fraccionarias. Los conjuntos de datos preprocesados resultantes se utilizan para construir el clasificador.

1.1 Antecedentes

En esta sección se desarrollan los fundamentos teóricos esenciales para la presente investigación. En primer lugar, se examina el concepto de big data y su creciente relevancia en el contexto tecnológico actual. A continuación, se presentan los principios del análisis de datos y del aprendizaje automático, resaltando su importancia para el procesamiento eficiente y la extracción de conocimiento de grandes volúmenes de información. Seguidamente, se analizan la complejidad y las características intrínsecas que pueden presentar los conjuntos de datos, así como los métodos utilizados para evaluar dichas complejidades. Por último, se exploran los enfoques para estimar el rendimiento de los algoritmos de clasificación, con el objetivo de evaluar su comportamiento cuando los conjuntos de datos presentan diversas problemáticas.

1.1.1 Big Data

El concepto de big data ha sido definido de diversas maneras en la literatura, en términos generales, se refiere a grandes volúmenes de datos generados por dispositivos conectados a internet, como los pertenecientes al Internet de las Cosas (IoT, *Internet of Things*), la Web 3.0, redes sociales, fotografías, videos, entre otros (Hassib et al., 2020). Una de las principales discusiones en torno a big data se centra en los desafíos de escalabilidad para el análisis de estas enormes cantidades de datos (Ramírez-Gallego et al., 2021). No obstante, los problemas asociados con el análisis de grandes volúmenes de datos no surgieron de manera repentina, este fenómeno ha evolucionado a lo largo de varios años debido a que la generación de datos ha superado con creces la capacidad de extraer información valiosa de ellos (Tsai et al., 2015).

Aunque big data implica la existencia de grandes volúmenes de datos, esto no garantiza que estos datos sean útiles. Los datos pueden ser ambiguos, contener anomalías o estar afectados por ruido. El verdadero valor de big data radica en la capacidad de extraer patrones, a menudo inesperados, y obtener información valiosa mediante la aplicación adecuada de técnicas de análisis de datos (Triguero et al., 2018). Por esta razón, es esencial revisar y adaptar las metodologías tradicionales de minería de datos y aprendizaje automático al contexto de big data (Tsai et al., 2015).

Uno de los primeros desafíos en big data es el volumen de los datos, que puede saturar los sistemas tradicionales de análisis de datos. A diferencia de los métodos convencionales, el cuello de botella se encuentra en las

etapas de procesamiento, comunicación y almacenamiento de los datos generados. Asimismo, la velocidad a la que se generan los datos introduce el desafío de procesar grandes cantidades de información en un tiempo limitado, lo que puede superar la capacidad de los sistemas de análisis. La variedad de los datos, es decir, la diversidad en su estructura y formato, representa otro obstáculo importante, ya que los sistemas de análisis deben lidiar con datos incompletos o heterogéneos, lo que requiere un preprocesamiento adecuado para garantizar resultados precisos.

El análisis de big data ha impulsado avances tecnológicos en diversos sectores, como la medicina, los negocios, el transporte y la energía. No obstante, los desafíos que plantea el procesamiento de grandes volúmenes de datos exigen el desarrollo de nuevas técnicas o métodos, además de plataformas de alto rendimiento capaces de realizar análisis en paralelo o distribuido (Tsai et al., 2015; García et al., 2016; Tsai et al., 2016). Una de las primeras plataformas ampliamente adoptadas para el procesamiento de grandes volúmenes de datos fue *MapReduce*, que sentó las bases para sistemas de código abierto como Apache Hadoop. Sin embargo, *MapReduce* presenta limitaciones de rendimiento, particularmente en la ejecución de ciclos iterativos, comunes en técnicas de preprocesamiento y aprendizaje automático, ya que requiere frecuentes operaciones de lectura y escritura en disco, lo que ralentiza el proceso de análisis de datos.

En respuesta a estas limitaciones, se desarrolló Apache Spark, una plataforma de segunda generación que mejora significativamente el rendimiento

al almacenar los datos en memoria, evitando así las operaciones frecuentes de lectura y escritura en disco. Esto permite a Spark ejecutar algoritmos iterativos hasta 100 veces más rápido que Hadoop (Singh et al., 2019). Por otra parte, Apache Flink, una herramienta de tercera generación también desarrollada por la Fundación Apache, ofrece avances adicionales, especialmente en el procesamiento de flujos de datos, proporcionando mayor flexibilidad y eficiencia en comparación con Hadoop y Spark.

En big data, las técnicas tradicionales enfrentan varios desafíos:

1. **Escalabilidad**: Los algoritmos deben ser capaces de gestionar conjuntos de datos de gran escala de manera eficiente, manteniendo su rendimiento sin degradación (Maillo et al., 2017).

2. **Alta dimensionalidad**: Las técnicas basadas en distancias euclidianas, comúnmente empleadas para equilibrar los conjuntos de datos, pueden volverse ineficaces cuando los datos tienen un gran número de atributos, también conocidos como características o dimensiones (Maldonado et al., 2019).

3. **Combinación de problemáticas**: Existe una carencia de propuestas que aborden simultáneamente otras características intrínsecas de los datos, como el solapamiento y la alta dimensionalidad.

4. **Arquitecturas de big data**: Los nuevos entornos de computación, como Hadoop y Spark, presentan particularidades que pueden influir en

el desempeño de los algoritmos, por ejemplo, el número de particiones en las que se divide el conjunto de datos para su distribución en el clúster (Basgall et al., 2019).

1.1.2 Aprendizaje Automático

El aprendizaje automático se refiere a los procesos que permiten a las computadoras extraer información relevante de manera automática de los datos, con el fin de generar predicciones o identificar posibles asociaciones y patrones en los datos. Este campo multidisciplinar combina aspectos de ciencias computacionales, estadística, inteligencia artificial y teoría de la información (Ma et al., 2014). Las técnicas de aprendizaje automático se dividen principalmente en dos categorías: aprendizaje supervisado y aprendizaje no supervisado. Existe también una tercera categoría, el aprendizaje semisupervisado, que combina elementos de ambos enfoques. El aprendizaje supervisado se enfoca en predecir resultados a partir de datos de entrada, basándose en un conjunto de ejemplos etiquetados. El objetivo es descubrir relaciones entre los atributos de entrada (características) y los de salida (clases), lo que permite la creación de modelos que describen patrones ocultos en los datos y predicen valores futuros. Los datos de entrenamiento contienen clases conocidas, mientras que el modelo resultante busca predecir las clases de nuevos datos (García et al., 2016). El conjunto de datos de prueba se representa como vectores de valores y atributos, sobre los cuales se intenta predecir la clase correcta. Estos atributos pueden ser categóricos, con cardinalidad infinita, o numéricos,

delimitados por límites inferiores y superiores. En este contexto, el concepto de "espacio de instancias" se refiere al conjunto de posibles valores de entrada, definido por el producto cartesiano de todos los atributos, mientras que el "espacio universal de las instancias" incluye tanto atributos de entrada como de salida (Rahmati et al., 2020).

El objetivo principal de un clasificador es diferenciar entre distintos datos para realizar predicciones confiables, lo cual es su principal aplicación. Una vez que se ha desarrollado un modelo que se ajusta a los datos históricos o conocidos, este modelo puede realizar predicciones precisas para nuevos datos, siempre que sean similares a los datos previos (Yu et al., 2018).

En el aprendizaje supervisado, los problemas de regresión buscan encontrar un modelo para predecir valores continuos, mientras que el análisis de series temporales, como la predicción de precios de acciones o tendencias de mercado, considera la evolución de los datos a lo largo del tiempo (Rahmati et al., 2020).

Por otro lado, el aprendizaje no supervisado no utiliza etiquetas predefinidas para entrenar al modelo. Su objetivo es descubrir patrones y relaciones subyacentes en los datos, lo que se logra a través de técnicas como el agrupamiento (*clustering*) y las reglas de asociación. Estas técnicas permiten identificar regularidades, anomalías, relaciones y similitudes sin la necesidad de intervención humana directa (Hassib et al., 2019).

Finalmente, el aprendizaje semisupervisado es un enfoque de apren-

dizaje automático que aprovecha tanto los datos etiquetados como los no etiquetados, abordando el desafío de los conjuntos de datos etiquetados limitados, donde la obtención de etiquetas es costosa o lleva mucho tiempo (Chapelle et al., 2010). Al combinar una pequeña cantidad de datos etiquetados con un conjunto más grande de datos no etiquetados, el aprendizaje semisupervisado permite que los modelos mejoren su precisión sin requerir grandes esfuerzos de etiquetado (Zhu y Goldberg, 2009).

1.1.3 Características Intrínsecas de los Datos

La calidad de los conjuntos de datos es un factor crucial que influye en el rendimiento de los algoritmos de aprendizaje automático. Los datos pueden presentar diversos problemas, entre los que se incluyen la alta dimensionalidad, el ruido, el solapamiento, el desplazamiento de los datos, la presencia de pequeños disyuntos y, particularmente, las clases no balanceadas. En el contexto de clases no balanceadas, el solapamiento es uno de los problemas más críticos para la correcta clasificación de instancias (García et al., 2006). El solapamiento ocurre cuando, en ciertas regiones del espacio de características, la probabilidad de pertenencia a diferentes clases es casi idéntica, lo que dificulta la correcta clasificación (Guzmán-Ponce et al., 2020).

García et al. (2006) demostraron que, a medida que aumenta el solapamiento entre clases, los efectos del desbalance de clases se intensifican, afectando negativamente la capacidad del clasificador para generalizar. Otros estudios también destacan que el solapamiento es un factor determinante en

el desempeño de los clasificadores en problemas de clases no balanceadas. En presencia de solapamiento, los clasificadores tienden a asignar las instancias minoritarias en la región solapada a la clase mayoritaria o a clasificarlas como ruido, lo que conduce a fronteras de decisión difusas y complejas que los algoritmos de aprendizaje automático no pueden aprender adecuadamente (Saez et al., 2019).

Otro reto relevante es la alta dimensionalidad que se relaciona con la geometría, la topología y la densidad del espacio de características. A medida que aumenta el número de variables de entrada, también lo hace el volumen del espacio de datos, lo que conlleva una dispersión que dificulta su análisis. Este fenómeno, conocido como la "maldición de la dimensionalidad" (Thudumu et al., 2020), provoca que las métricas de distancia, como la distancia euclidiana, pierdan su capacidad discriminatoria. Dos efectos importantes asociados con la alta dimensionalidad son la concentración de distancias y el *hubness*[1]. La concentración de distancias ocurre cuando todos los puntos del espacio parecen estar aproximadamente a la misma distancia entre sí (Flexer y Schnitzer, 2015), lo que reduce la efectividad de las métricas de distancia para distinguir entre instancias. El hubness, descrito por Tomasev et al. (2014), es la tendencia de ciertas instancias a aparecer repetidamente como vecinos más cercanos de muchas otras instancias, lo que sesga el proceso de clasificación.

Para mitigar los problemas derivados de la concentración de distancias

[1]No se encontró una traducción para esta palabra; en la literatura en español, normalmente no se traduce. Por lo tanto, este concepto se manejará en inglés.

en espacios de alta dimensionalidad, Aggarwal et al. (2001) proponen el uso de distancias fraccionarias basadas en la norma-p, que también se puede referir con el símbolo ℓ_p. La norma-p, para un vector $\mathbf{x}$ con n elementos denotada como $|\mathbf{x}|$, se define formalmente de la siguiente manera:

$$|\mathbf{x}| := \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \quad \text{para} \quad p \geq 1, \tag{1.1}$$

La métrica de distancia de Minkowski, basada en la norma-p, para dos vectores $\mathbf{x}$ y $\mathbf{y}$ denotada como $d(\mathbf{x}, \mathbf{y})$, se expresa de la siguiente manera:

$$d(\mathbf{x}, \mathbf{y}) := \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \quad \text{para} \quad p \geq 1 \tag{1.2}$$

Las distancias fraccionarias se aplican cuando $0 < p < 1$, aunque no son formalmente definidas como distancias debido a que violan la desigualdad del triángulo (Mirkes et al., 2020).

A pesar de su utilidad en contrarrestar la concentración de distancias, la selección del valor óptimo de p no es trivial, y su cálculo puede ser computacionalmente costoso (Aggarwal et al., 2001; Mirkes et al., 2020). Estudios de Cormode et al. (2002), Flexer y Schnitzer (2015) y Gorban et al. (2018) proponen técnicas para determinar el valor de p que maximice la efectividad de la clasificación basada en el vecino más cercano.

1.1.4 Complejidad de los Datos

La complejidad de los datos es un factor crucial que afecta el rendimiento de los algoritmos de aprendizaje automático. Por ello, una línea de investigación propone que las problemáticas presentes en los datos pueden cuantificarse mediante el uso de métricas que caracterizan aspectos específicos de cada una de las complejidades presentes en los datos. Lorena et al. (2020) agrupan estas métricas en seis categorías principales:

1. Métricas basadas en las características: Cuantifican la capacidad de los atributos para separar eficazmente las clases, según la información que proporcionan.

2. Métricas de linealidad: Cuantifican la capacidad de las clases para ser separadas de manera lineal.

3. Métricas de vecindad: Caracterizan la densidad de clases similares o diferentes en términos de la proximidad local.

4. Métricas basadas en grafos: Extraen información estructural de los conjuntos de datos utilizando representaciones en forma de grafos.

5. Métricas de dimensionalidad: Evalúan la escasez de datos en relación con el número de instancias necesarias según la dimensionalidad.

6. Métricas de desbalance: Miden la proporción entre los números de ejemplos de las diferentes clases.

Para calcular las métricas de complejidad en este estudio, consideremos un conjunto de datos T compuesto por n ejemplos en d dimensiones. Cada ejemplo x_i, con $i = 1, \ldots, n$, se representa como un vector $\mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$. La etiqueta de clase asociada a x_i se denota como $y_j \in \Omega$, donde $j = 1, \ldots, c$, y Ω es el conjunto de las c posibles etiquetas de clase.

F1: Máximo Ratio Discriminante de Fisher

Estima el solapamiento entre los atributos de diferentes clases, evaluando qué tan cercanas están las clases al analizar cada característica individualmente, es decir, cuantifica la efectividad de cada característica para separar las clases y selecciona la característica con el valor más alto. $F1$ se calcula de la siguiente manera:

$$F1 = \arg\max_{i=1}^{d}(f_i) \tag{1.3}$$

donde f_i está definido por:

$$f_i = \frac{(\mu_{c_1} - \mu_{c_2})^2}{\sigma_{c_1}^2 + \sigma_{c_2}^2} \tag{1.4}$$

siendo μ_{c_1} y μ_{c_2} las medias, y $\sigma_{c_1}^2$ y $\sigma_{c_2}^2$ las varianzas de la característica f_i para las clases c_1 y c_2, respectivamente.

Dado que el valor de $F1$ se encuentra en el rango $[0, \infty]$, se puede obtener una versión normalizada en $[0, 1]$, utilizando la siguiente fórmula:

$$F1_{norm} = \frac{1}{F1 + 1} \tag{1.5}$$

Cuanto mayor sea el valor de $F1_{norm}$, más complejo es el conjunto de datos. Un problema se considera sencillo si al menos una característica es capaz de separar completamente ambas clases (Barella et al., 2021).

F2: Volumen de la Región de Solapamiento

Cuantifica el volumen de la región de solapamiento entre clases utilizando los valores máximos y mínimos de cada característica. Estos valores se normalizan utilizando los rangos de los atributos (Barella et al., 2021). Un valor de $F2$ igual a cero indica que las clases son disyuntas (Basu y Ho, 2006), mientras que un valor más alto de F2 señala un mayor solapamiento entre las clases. Sin embargo, $F2$ tiende a disminuir a medida que aumenta la dimensionalidad del conjunto de datos, lo que dificulta la comparación de valores de $F2$ en problemas con alta dimensionalidad frente a aquellos con menos características (Lorena et al., 2020). La métrica se calcula de la siguiente manera:

$$F2 = \prod_{i=1}^{d} \frac{\min\max(f_i) - \max\min(f_i)}{\max\max(f_i) - \min\min(f_i)} \tag{1.6}$$

donde:

$$\min\max(f_i) = \min(\max(f_i, c_1), \max(f_i, c_2)) \tag{1.7}$$

$$\max \min(f_i) = \max(\min(f_i, c_1), \min(f_i, c_2)) \tag{1.8}$$

$$\max \max(f_i) = \max(\max(f_i, c_1), \max(f_i, c_2)) \tag{1.9}$$

$$\min \min(f_i) = \min(\min(f_i, c_1), \min(f_i, c_2)) \tag{1.10}$$

IR: Ratio de Desbalance

Es una métrica de complejidad que mide la disparidad entre el número de ejemplos de las diferentes clases en un conjunto de datos. Para un problema binario, IR se define como la proporción entre el número de ejemplos de la clase mayoritaria y la clase minoritaria:

$$IR = \frac{N_{mayor}}{N_{menor}} \tag{1.11}$$

donde N_{mayor} representa el número de ejemplos en la clase mayoritaria y N_{menor} representa el número de ejemplos en la clase minoritaria. Un valor de IR igual a 1 indica un conjunto de datos perfectamente balanceado, mientras que un valor mayor de IR refleja un mayor desbalance entre las clases.

1.1.5 Estimación del Rendimiento

La evaluación del rendimiento de un clasificador se realiza comúnmente en una matriz de confusión. En esta matriz cuadrada, cada entrada (i, j) representa el número de predicciones correctas e incorrectas realizadas por el clasificador (Japkowicz y Shah, 2011). La Tabla 1.1 muestra un ejemplo de

matriz de confusión de 2×2 para un problema de clasificación binaria, en la cual las columnas corresponden a las predicciones del clasificador y las filas representan las clases reales. Los elementos en la diagonal principal corresponden a las predicciones correctas para las clases positiva y negativa, mientras que las demás entradas representan los errores cometidos.

Tabla 1.1: Matriz de confusión para un problema de dos clases.

	Predicción Positiva	*Predicción Negativa*
Real Positiva	Verdaderos Positivos (TP)	Falsos Negativos (FN)
Real Negativa	Falsos Positivos (FP)	Verdaderos Negativos (TN)

La métrica más utilizada para evaluar la efectividad de un clasificador es la exactitud, que se define como:

$$A = \frac{TP + TN}{TP + FN + TN + FP} \tag{1.12}$$

donde TP y TN representan los verdaderos positivos y negativos, respectivamente, y FN y FP corresponden a los falsos negativos y positivos. Sin embargo, se ha demostrado que la exactitud no es adecuada para conjuntos de datos que no están balanceados, lo que ha llevado a la adopción de métricas alternativas más representativas (Japkowicz y Shah, 2011). Entre estas métricas destacan las que evalúan el rendimiento por clase:

- Tasa de verdaderos positivos (TPR, *True Positive Rate*):

$$TPR = \frac{TP}{TP + FN} \tag{1.13}$$

que representa la proporción de ejemplos positivos correctamente clasificados.

- Tasa de verdaderos negativos (TNR, *True Negative Rate*):

$$TNR = \frac{TN}{TN + FP} \tag{1.14}$$

que representa la proporción de ejemplos negativos correctamente clasificados.

- Tasa de falsos negativos (FNR, *False Negative Rate*):

$$FNR = \frac{FN}{TP + FN} \tag{1.15}$$

que es la proporción de ejemplos positivos clasificados incorrectamente.

- Tasa de falsos positivos (FPR, *False Positive Rate*):

$$FPR = \frac{FP}{TN + FP} \tag{1.16}$$

que es la proporción de ejemplos negativos clasificados incorrectamente.

Una métrica que sintetiza las tasas individuales por clase es la media geométrica:

$$G - Mean = \sqrt{TPR \times TNR} \tag{1.17}$$

que maximiza la efectividad de cada clase mientras mantiene un equilibrio entre ellas, lo que la hace adecuada en escenarios con clases no balanceadas.

Además de estas métricas, los clasificadores pueden evaluarse mediante métodos gráficos, que permiten una visualización más clara de su efectividad (Prati et al., 2011). Uno de los enfoques más utilizados es la curva ROC (*Receiver Operating Characteristics*), que es un gráfico bidimensional que representa la relación entre TPR en el eje Y y FPR en el eje X. A partir de esta curva, se puede calcular el área bajo la curva ROC, una métrica escalar definida como (Fawcett, 2006):

$$AUC - ROC = \frac{TPR + TNR}{2} \tag{1.18}$$

Esta métrica es ampliamente utilizada en problemas con clases no balanceadas debido a su capacidad para medir el rendimiento global del clasificador en estos escenarios.

Otra representación gráfica es la curva *Precision-Recall*, que traza la precisión en el eje Y y el *recall* en el eje X. La precisión se define como (Prati et al., 2011):

$$P = \frac{TP}{TP + FP} \tag{1.19}$$

Aunque este método es popular en problemas no balanceados, se ha demostrado que las métricas basadas en la *precision* y el *recall* tienden a enfocarse principalmente en la clase minoritaria, ignorando el rendimiento en la clase mayoritaria (Sokolova y Lapalme, 2009). En escenarios altamente desbalanceados, la *precision* puede verse afectada por la presencia de un gran número de FP, lo

que produce bajas tasas (Daskalaki et al., 2006; Landgrebe et al., 2006). Por lo tanto, al utilizar la *precision* en la evaluación, requiere una ponderación que compense el sesgo introducido por los FP (Daskalaki et al., 2006; Landgrebe et al., 2006).

En este trabajo, se priorizará el uso de métricas que permitan una evaluación equilibrada de la efectividad del clasificador, teniendo en cuenta los resultados en ambas clases para garantizar una medición representativa del rendimiento general del modelo.

1.2 Problema de Investigación

El problema de investigación que se aborda en esta tesis doctoral se centra en la compleja interrelación entre el desbalance de clases, el solapamiento entre clases y la alta dimensionalidad en ambientes big data. El estudio de las clases no balanceadas ha sido un tema de investigación por más de dos décadas. Un ejemplo temprano es el trabajo de Pazzani et al. (1994), quienes propusieron un algoritmo basado en costos. A lo largo de los años, el interés por desarrollar nuevas estrategias para abordar este problema ha crecido, evidenciado por la existencia de hasta 85 variantes del algoritmo SMOTE (Kovács, 2019). El desbalance de clases puede ocasionar que los modelos de clasificación desarrollen un sesgo hacia las clases mayoritarias, ignorando las clases minoritarias que, en muchos casos, son de mayor interés. Este problema se ve exacerbado en el contexto del big data, donde la escala y la complejidad de los datos introducen nuevos desafíos, como la escalabilidad de los algoritmos y

la necesidad de infraestructuras de cómputo avanzadas como GPU o clústeres de computadoras (Sleeman IV y Krawczyk, 2021b).

Además, el solapamiento entre clases, que ocurre cuando las instancias de diferentes clases se superponen en el espacio de características, complica aún más la tarea de clasificación. Por otro lado, la alta dimensionalidad, que se refiere a la presencia de un gran número de atributos en los datos, plantea un desafío adicional. Las técnicas convencionales de clasificación, basadas en distancias euclidianas, tienden a perder efectividad en entornos de alta dimensionalidad, donde las distancias entre instancias se vuelven menos significativas. Sin embargo, la evidencia muestra que las soluciones propuestas hasta ahora se han centrado en problemas donde el número de atributos es relativamente bajo. Por ejemplo, en un estudio de Juez-Gil et al. (2021a), el conjunto de datos con mayor número de atributos contaba con 893 características. En contraste, en repositorios como LIBSVM existen conjuntos de datos masivos con un número significativamente mayor de atributos, como la base de datos KDD CUP 2012[2], que presenta 54,686,422 características. Esta situación resalta la urgente necesidad de desarrollar o combinar técnicas que puedan abordar la combinación de desafíos presentes en conjuntos de datos masivos. Por lo tanto, este estudio busca explorar y desarrollar estrategias que integren técnicas para abordar las clases no balanceadas, el solapamiento y la alta dimensionalidad en el contexto del big data. Se propone un método que no solo sea escalable y eficiente, sino que también sea capaz de manejar

[2] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

las complejidades inherentes a los conjuntos de datos masivos, mejorando así la capacidad de los modelos de aprendizaje automático para realizar clasificaciones precisas y significativas. Este enfoque integral es esencial para avanzar en la comprensión y solución de los problemas que surgen en el análisis de datos en la era del big data.

1.2.1 Preguntas de Investigación

El enfoque principal de este trabajo es desarrollar un método sistemático de preprocesamiento para el tratamiento de conjuntos de datos masivos caracterizados por las clases no balanceadas, el solapamiento entre clases y la alta dimensionalidad. Para guiar esta investigación, se han planteado las siguientes preguntas, las cuales abordan diferentes aspectos clave del problema de estudio:

1. ¿De qué manera influye la aplicación de algoritmos de preprocesamiento basados en la distancia euclidiana en el rendimiento de los modelos de aprendizaje automático en conjuntos de datos masivos no balanceados y de alta dimensionalidad?

2. ¿Cuál es el orden más adecuado para implementar el método sistemático de preprocesamiento que permita abordar eficazmente las problemáticas de clases no balanceadas, solapamiento y alta dimensionalidad en conjuntos de datos masivos?

3. ¿Qué beneficios, en términos de mejora en la efectividad de clasificación,

ofrece la incorporación de distancias fraccionarias en los algoritmos de preprocesamiento en comparación con los basados en distancia euclidiana en problemas de alta dimensionalidad?

4. ¿En qué medida la transformación al espacio de disimilitud contribuye a la reducción de la dimensionalidad en conjuntos de datos con alta dimensionalidad?

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un método sistemático escalable de preprocesamiento que permita mejorar de manera significativa el rendimiento predictivo en la clasificación de conjuntos de datos masivos medido por la métrica AUC-ROC con respecto a los datos sin preprocesamiento, abordando de manera integral los desafíos asociados a las clases no balanceadas, la alta dimensionalidad y el solapamiento entre clases.

1.3.2 Objetivos Específicos

1. Analizar las técnicas existentes en la literatura científica para abordar el problema de las clases no balanceadas con solapamiento en conjuntos de datos masivos y de alta dimensionalidad.

2. Reproducir y evaluar algoritmos de preprocesamiento existentes en la literatura científica para el tratamiento de clases no balanceadas, cen-

trándose en su efectividad en escenarios de big data.

3. Desarrollar un algoritmo escalable de sobremuestreo que incorpore el uso de distancias fraccionarias para mejorar el equilibrio entre clases en entornos de alta dimensionalidad.

4. Desarrollar un algoritmo escalable para la transformación del espacio de características con alta dimensionalidad hacia un espacio de disimilitud con una dimensionalidad reducida, optimizando la separabilidad entre clases, utilizando distancias fraccionarias.

5. Desarrollar un algoritmo escalable de selección de instancias que mejore la clasificación en conjuntos de datos masivos no balanceados con solapamiento.

6. Evaluar el desempeño del enfoque desarrollado frente a otras técnicas, utilizando métricas de efectividad y complejidad en la clasificación.

7. Realizar análisis estadísticos para determinar la significancia de los resultados obtenidos, y validar o refutar las hipótesis relacionadas con la mejora en el rendimiento del método propuesto.

1.4 Variables de Investigación

Las variables independientes identificadas en este proyecto son:

1. Método sistemático propuesto. Se refiere a la combinación de técnicas de preprocesamiento y modelado, que incluyen:

- Transformación al espacio de disimilitud, los conjuntos de datos con alta dimensionalidad, mediante el uso de distancias fraccionarias, y así reducir su dimensionalidad y mejorar su manejo.
- Técnica de sobremuestreo para abordar el desbalance de clases.
- Técnica de selección de instancias o edición para reducir el solapamiento entre clases.

2. Conjunto de datos. Se refiere a los distintos conjuntos big data utilizados en los experimentos, que presentan características como:
   - Desbalance de clases (diferencia significativa en el número de instancias entre clases).
   - Alta dimensionalidad (gran número de atributos o características).
   - Solapamiento entre clases (instancias de diferentes clases que se superponen en el espacio de características).
3. Parámetros del algoritmo. Parámetros ajustables del método sistemático, como:
   - Valor de fracción en distancias fraccionarias.
   - Tasa de sobremuestreo o submuestreo.

La variable dependiente es el rendimiento del clasificador, medido a través de:

1. Efectividad del clasificador: evaluada en términos de la media geométrica (*G-Mean*) y el área bajo la curva ROC (*AUC-ROC*).

1.5 Hipótesis

Si se definen los siguientes parámetros:

- C: Clasificador.
- D: Conjunto big data, con solapamiento y alta dimensionalidad, sin preprocesamiento.
- D_{Prep}: Conjunto big data, con solapamiento y alta dimensionalidad, preprocesados con el método sistemático.
- R_C: Rendimiento del clasificador.
- H_0: Hipótesis nula.
- H_1: Hipótesis alternativa.

Entonces, las hipótesis de este estudio se formulan de la siguiente manera:

- $H_0 : R_C(C, D_{Prep}) \leq P_C(C, D)$ (No se mejora significativamente el rendimiento predictivo del clasificador cuando se entrena con el conjunto de datos preprocesado en comparación con el conjunto de datos original).

- $H_1 : R_C(C, D_{Prep}) > R_C(C, D)$ (Se mejora significativamente el rendimiento predictivo del clasificador cuando se entrena con el conjunto de datos preprocesado que con el conjunto de datos original).

1.6 Alcances y Delimitaciones

Para el desarrollo de esta tesis, se establecen los siguientes alcances y delimitaciones:

1. **Alcances**:

   (a) El estudio se centró en la creación de un método sistemático que combine técnicas de preprocesamiento avanzadas como la transformación al espacio de disimilitud mediante distancias fraccionarias, técnicas de sobremuestreo y selección de instancias. Este enfoque se diseñará específicamente para mejorar la clasificación en entornos de big data con desbalance de clases, alta dimensionalidad y solapamiento entre clases.

   (b) La investigación aplicó el enfoque propuesto en conjuntos de datos masivos que presenten las problemáticas mencionadas.

   (c) El rendimiento del enfoque desarrollado se evaluó y comparó con otras técnicas existentes en la literatura, utilizando métricas de efectividad como la G-Mean y el AUC-ROC.

   (d) Se llevaron a cabo pruebas estadísticas para verificar la significancia de los resultados obtenidos, lo que permitirá aceptar o rechazar las

hipótesis planteadas y validar la superioridad del enfoque propuesto en términos del rendimiento del clasificador.

2. **Delimitaciones**:

   (a) Los experimentos estuvieron limitados a dos conjuntos de datos específicos que presenten las problemáticas de clases no balanceadas, solapamiento y alta dimensionalidad. Aunque se utilizaron datos de diferentes dominios, los resultados podrían no ser generalizables a todos los tipos de big data.

   (b) El estudio se centró en la combinación de técnicas específicas, como distancias fraccionarias, sobremuestreo y selección de instancias (edición). No se evaluarán todas las posibles combinaciones de técnicas, lo que puede limitar el alcance en términos de posibles enfoques alternativos.

   (c) No se realizó la búsqueda de la distancia fraccionaria óptima y solo se utilizaron las distancias fraccionarias 0.75, 0.66, 0.50, 0.33 y 0.25.

   (d) La implementación del algoritmo se realizó utilizando Apache Spark versión 3.1.1 y Scala versión 2.12. Las técnicas utilizadas en la investigación pueden no ser aplicables en entornos con limitaciones de hardware, tiempo real u otra versión de Apache Spark y Scala.

   (e) El estudio abarcó la problemática en escenarios de dos clases. Una generalización a más de dos clases está fuera del alcance de este trabajo.

Capítulo 2

Estado del Arte

En este capítulo se presenta el estado del arte de las técnicas y métodos utilizados para abordar las problemáticas de las clases no balanceadas, el solapamiento y la alta dimensionalidad en conjuntos de datos masivos. La revisión se organiza en cuatro secciones. En la Sección 2.1 se describe la metodología empleada para la revisión sistemática de la literatura. En la Sección 2.1.1 se realiza un análisis bibliométrico de las publicaciones científicas seleccionadas. En la Sección 2.2 se presentan las principales técnicas o métodos utilizados para tratar el problema de las clases no balanceadas. Finalmente, en la Sección 2.3 se ofrece un resumen y análisis de las diferentes estrategias desarrolladas específicamente para el manejo de datos masivos.

2.1 Metodología Sistemática de Revisión de Bibliografía

Para analizar el estado actual de las investigaciones sobre las clases no balanceadas con solapamiento y alta dimensionalidad se empleó la metodología PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analysis*) (Page et al., 2021). Esta metodología consta de cuatro etapas:

1. **Identificación**: Se realizó una búsqueda de artículos publicados en

bases de datos científicas (Springer, IEEEXplore, ScienceDirect y otras fuentes). Las palabras clave utilizadas fueron “Big Data”, “Class Imbalance”, “Class Overlapping” y “High Dimensionality”. El filtro se configuró para buscar en los campos de título, resumen y palabras clave de los artículos. Los resultados se limitaron a publicaciones realizadas entre 2016 y 2024. Esto produjo un total de 655 artículos: 492 de Springer, 39 de IEEEXplore, 68 de ScienceDirect y 56 de otras fuentes. La diferencia en la cantidad de artículos encontrados en Springer puede atribuirse al funcionamiento de su motor de búsqueda, ya que la mayoría de los artículos no incluían ambos términos de búsqueda.

2. **Selección**: Se descartaron los artículos que no cumplían con los parámetros de inclusión, por ejemplo, aquellos que trataban sobre clases no balanceadas en conjuntos de datos tradicionales, i.e., con una cantidad de ejemplos reducida y que no son procesados por múltiples nodos de cómputo, en lugar de en big data. Este filtro redujo el total a 100 artículos.

3. **Elegibilidad**: Se llevó a cabo una lectura rápida de los resúmenes y conclusiones de los 100 artículos. Se excluyeron 23 documentos que no proponían una solución al problema o que solo se enfocaban en comparaciones de desempeño sin nuevos enfoques metodológicos.

4. **Inclusión**: Finalmente, se seleccionaron 60 artículos relevantes, que constituyen la base para la revisión del estado del arte en esta investigación.

2.1.1 Análisis Bibliométrico

La bibliometría es una herramienta fundamental para entender cómo se organiza y desarrolla el conocimiento en un campo de investigación, utilizando métodos matemáticos y estadísticos para identificar tendencias y avances clave. En esta tesis, se analizan los 60 artículos seleccionados en función de seis aspectos relevantes:

1. Estrategias de solución.
2. Problemáticas en los datos.
3. Clasificadores.
4. Métricas de efectividad.
5. Año de publicación del artículo.
6. Revista de publicación.

La Figura 2.1 muestra los enfoques utilizados para abordar el problema de las clases no balanceadas en big data. Las técnicas de sobremuestreo y submuestreo se mencionan en 13 y 9 artículos, respectivamente, mientras que las soluciones a nivel de algoritmo aparecen en 12 estudios. Además, se observa una tendencia clara hacia el uso de técnicas combinadas, que se presentan en 26 artículos. Este patrón sugiere un interés sostenido en la combinación de múltiples enfoques para mejorar el rendimiento de los clasificadores en problemas de desbalance de clases.

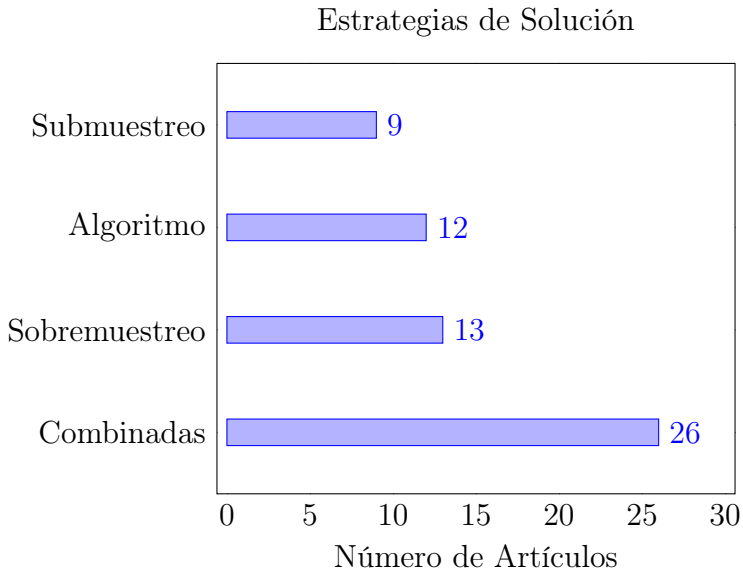


Figura 2.1: Artículos organizados por estrategia utilizada.

En cuanto a las problemáticas presentes en los datos, los artículos se agruparon según su enfoque exclusivo en el problema de clases no balanceadas o si también abordan otros desafíos, como el número de atributos, datos perdidos o solapamiento. Esta clasificación permite identificar áreas de oportunidad para futuras investigaciones. Como se observa en la Figura 2.2, la mayoría de los estudios se concentran en el problema del desbalance en dos clases, mientras que un número menor de publicaciones exploran otros desafíos, como la alta dimensionalidad, la clasificación multiclase, el solapamiento y los datos faltantes. Este desequilibrio sugiere la necesidad de más investigaciones que consideren problemas combinados, especialmente en el contexto de big data.

La Figura 2.3 presenta los clasificadores utilizados en los artículos analizados. Cabe destacar que algunos estudios emplearon más de un clasificador,

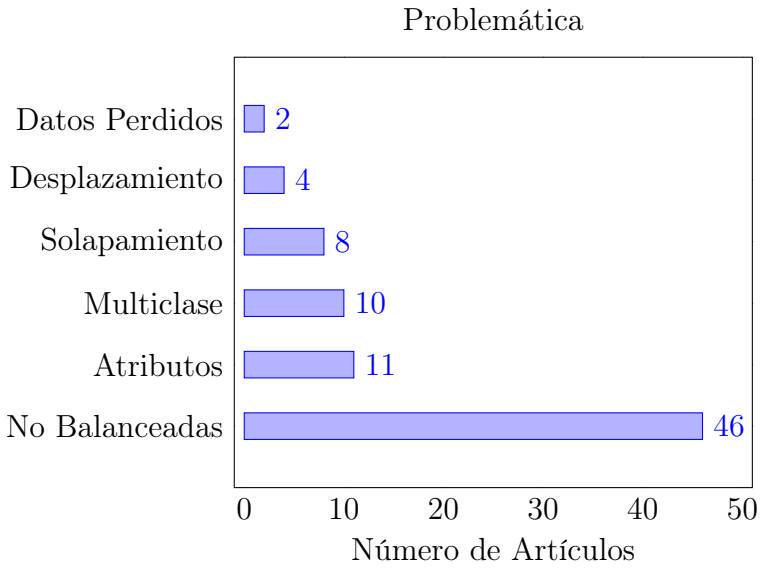


Figura 2.2: Artículos por problemática abordada.

lo que explica que la suma total no coincida con los 60 artículos seleccionados. Es interesante observar que no se percibe una preferencia clara por un clasificador específico. El más utilizado es el Sistema Múltiple de Clasificación (SMC), mencionado en 22 artículos, seguido por los árboles de decisión (DT, *Decision Tree*) con 20 menciones, k-vecinos más cercanos (k-NN, *k-Nearest Neighbor*) con 15, máquinas de vectores de soporte (SVM, *Support Vector Machine*) con 14, redes neuronales artificiales (ANN, *Artificial Neural Network*) con 13, y *Naive Bayes* (NB) con 10 menciones.

En relación con las métricas de efectividad, los resultados muestran que varios estudios emplearon más de una métrica para evaluar el rendimiento de los clasificadores. La Figura 2.4 indica que la métrica más utilizada es el AUC-ROC, con 35 menciones, seguida por la media geométrica y la métrica F, ambas

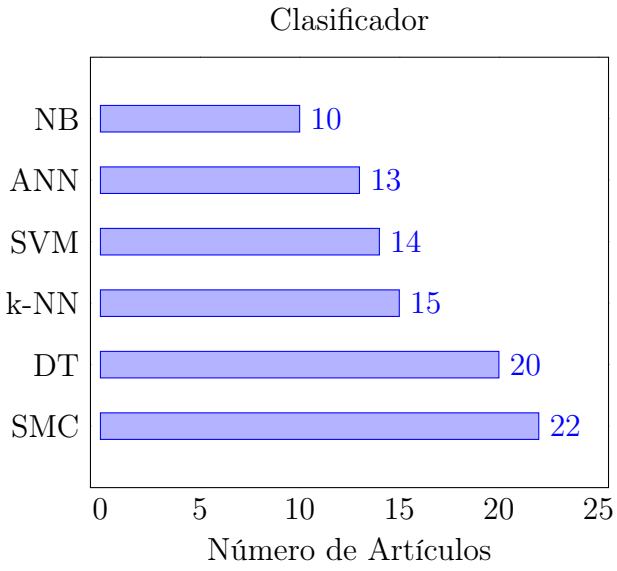


Figura 2.3: Artículos por tipo de clasificador.

con 24 menciones. Otras métricas relevantes incluyen la TPR mencionada en 19 artículos, la precisión con 14 menciones, y la TNR utilizada en 6 estudios.

Adicionalmente, se realizó un análisis de los años de publicación (Figura 2.5) y de las revistas en las que se publicaron los artículos (Figura 2.6). El análisis temporal muestra que ha habido un flujo constante de publicaciones desde 2016, con un promedio de 5 artículos por año. Los años 2020 y 2022 destacan como los de mayor producción, con 10 artículos en cada uno. En cuanto a las revistas, aunque la mayoría de los artículos provienen de conferencias, destacan el *Journal of Big Data* con 14 publicaciones, *IEEE Access* con 5, y *Soft Computing* con 3 (Figura 2.6), se han omitido las revistas que cuentan con menos de tres publicaciones.

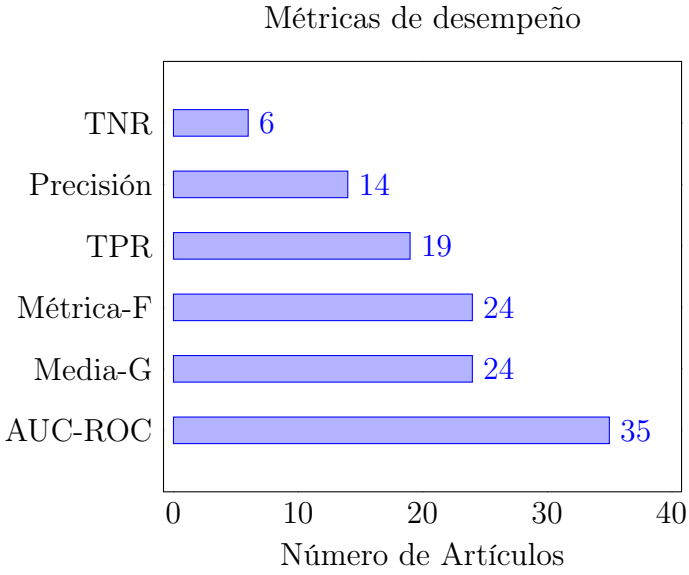


Figura 2.4: Artículos por métricas de efectividad utilizadas.

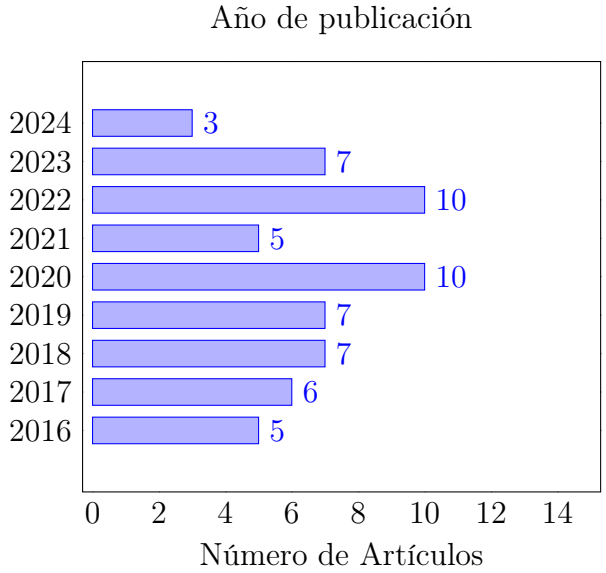


Figura 2.5: Artículos por año de publicación.

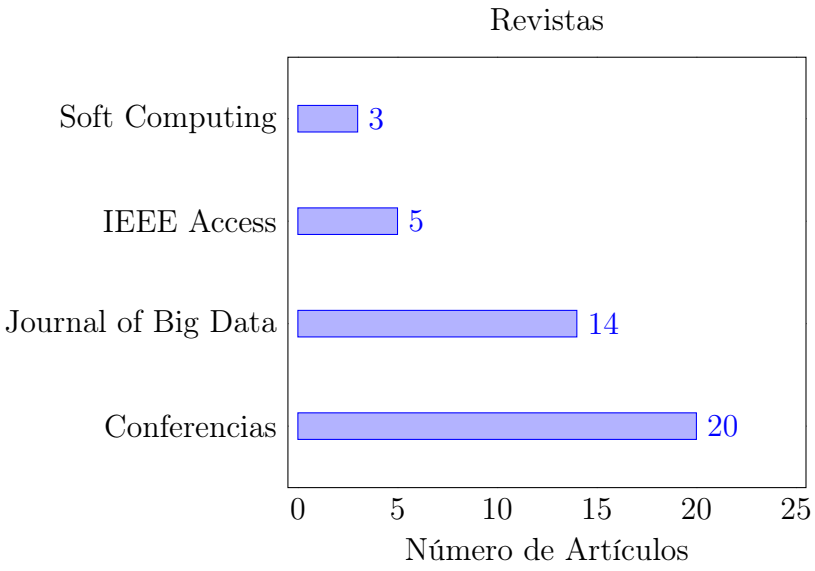


Figura 2.6: Número de artículos por revista.

2.2 Clases No Balanceadas y Big Data

El problema de las clases no balanceadas ha sido ampliamente estudiado en conjuntos de datos tradicionales (Kovács, 2019; Brennan, 2012; Jain et al., 2017; García et al., 2006; Ali et al., 2015; Saez et al., 2019; Pengfei et al., 2014). Sin embargo, hay pocos estudios que aborden esta problemática en el contexto de big data, donde los efectos adversos del desbalance son más evidentes (Bauder et al., 2018; Hasanin et al., 2020). En conjuntos de datos masivos, la desigualdad entre clases tiende a ser más extrema, lo que provoca una sobre-representación de la clase mayoritaria. Esta diferencia numérica dificulta el aprendizaje de la clase minoritaria por parte de los algoritmos de clasificación, convirtiéndose en un reto significativo (Leevy et al., 2018).

Las estrategias para abordar las clases no balanceadas en datos tradi-

cionales y big data son similares. Estas soluciones se agrupan en dos grandes enfoques: (i) técnicas a nivel de datos y (ii) técnicas a nivel de algoritmos (Ali et al., 2015). No obstante, algunos autores proponen otras categorías, como los sistemas múltiples de clasificación, clasificadores de una sola clase y enfoques combinados que integran varias técnicas (Sleeman IV y Krawczyk, 2021b).

Las soluciones a nivel de datos modifican el conjunto de datos original, creando o eliminando instancias para equilibrar el tamaño de las clases. Una estrategia adicional es la selección de atributos, que reduce el conjunto de datos en términos de columnas, mejorando la separabilidad entre clases (Sharifai y Zainol, 2020). En esta investigación se considera que la reducción del conjunto de datos por selección de atributos también pertenece al grupo de soluciones a nivel de datos.

Los métodos de remuestreo se dividen en dos grandes grupos: submuestreo y sobremuestreo. Ambos pueden ser aleatorios o dirigidos, siendo que en este último caso los algoritmos incorporan alguna heurística. El submuestreo, elimina instancias de la clase mayoritaria, mientras que en el sobremuestreo se incrementa el tamaño de la clase minoritaria mediante la generación de nuevas instancias artificiales o la duplicación de instancias existentes.

Una de las técnicas de sobremuestreo más utilizadas es SMOTE, que genera instancias sintéticas de la clase minoritaria (Chawla et al., 2002). Para cada ejemplo de la clase minoritaria (también conocido como ejemplo positivo), SMOTE calcula sus k-vecinos más cercanos y elige uno de estos vecinos

de forma aleatoria para generar nuevos datos mediante la interpolación entre la instancia original y la seleccionada. Este proceso se repite hasta alcanzar un equilibrio deseado entre las clases. Sin embargo, SMOTE utiliza la distancia euclidiana para calcular la similitud entre ejemplos, lo cual puede ser problemático en situaciones de alta dimensionalidad (Elreedy y Atiya, 2019; Maldonado et al., 2019). Este fenómeno, conocido como la maldición de la dimensionalidad, ocasiona una concentración de distancias, por lo que ejemplos diferentes pueden ser considerados como similares, dificultando la diferenciación entre ellos. Blagus y Lusa (2013) demostraron que en espacios de 1,000 dimensiones, SMOTE presenta un desempeño inferior al RUS debido a esta concentración de distancias. Para mitigar este efecto, Maldonado et al. (2019) sugieren el uso de métricas de distancia alternativas en SMOTE.

En lo que respecta a los métodos de selección de atributos, varios estudios han demostrado que seleccionar un subconjunto de atributos mejora la separabilidad entre clases, reduciendo los efectos adversos del desbalance entre clases (Leevy et al., 2018; Maldonado et al., 2014; Sharifai y Zainol, 2020).

Entre las soluciones a nivel de algoritmos, destacan los algoritmos basados en costos, que asignan un mayor peso a las instancias clasificadas incorrectamente. Por otro lado, los SMC combinan varios clasificadores que operan sobre un mismo conjunto de datos. Los métodos de *bagging* y *boosting* son los más comunes en este enfoque (Leevy et al., 2018). En el *bagging*, se minimiza la varianza de las predicciones generando varios subconjuntos de entrenamiento a partir del conjunto original, de modo que cada clasificador se entrena con uno

de estos subconjuntos. La decisión final se toma en función de la predicción de todos en la salida combinada de todos los clasificadores.

En el *boosting*, también se utilizan diferentes conjuntos de entrenamiento, asignando distintos pesos a cada clasificador de manera iterativa, considerando sus errores de clasificación. La combinación de los clasificadores individuales y sus respectivos pesos determina el clasificador final.

Finalmente, las soluciones combinadas utilizan distintos métodos para mejorar el rendimiento en situaciones de desbalance de clases. Un ejemplo es la versión de la técnica SMOTE combinada con una técnica de edición (Xu et al., 2020).

2.2.1 Sobremuestreo

Pengfei et al. (2014) utilizaron como base Borderline-SMOTE (Han et al., 2005), una modificación de la técnica SMOTE que genera instancias artificiales para los ejemplos cercanos a las fronteras de decisión, con el objetivo de abordar conjuntos de datos masivos de dos clases no balanceados. Esta propuesta demostró que mejora el desempeño de los algoritmos de clasificación en comparación con SMOTE y ROS, especialmente en distribuciones de datos masivas, donde las fronteras de decisión son ambiguas y los clasificadores experimentan una mayor pérdida de rendimiento. Además, los autores sugieren el uso del submuestreo como complemento, argumentando que, aunque pueden eliminar instancias valiosas de la clase mayoritaria, esta pérdida es aceptable, ya que los ejemplos de la clase minoritaria son más importantes para definir

fronteras claras de decisión. Al reducir las instancias de la clase mayoritaria, se espera que los ejemplos minoritarios se clasifiquen con mayor precisión debido a la menor densidad en las zonas de decisión. Sin embargo, los experimentos se realizaron con bases de datos relativamente pequeñas tomadas de la UCI, como el conjunto de datos Stigma (6,435 ejemplos y 36 atributos), lo que limita su aplicabilidad en entornos big data. Es importante destacar que en esta investigación se utilizó la distancia euclidiana en todos los experimentos.

Galpert et al. (2015) abordaron el problema de conjuntos de datos masivos no balanceados para la detección de ortólogos[1] en especies de levadura. Los autores integraron información genética adicional de las proteínas a los conjuntos de datos que se utilizan habitualmente, lo que incrementó tanto la dimensionalidad y el número de instancias. Esta adición de datos provocó que los algoritmos de clasificación, utilizados tradicionalmente en este dominio, no se desempeñaran adecuadamente debido al tamaño de los conjuntos de datos. Para abordar esta problemática, utilizaron tres técnicas de preprocesamiento en ambientes big data: RF sensibles al costo (RF-BDCS, *Random Forest - Big Data Cost Sensitive*), ROS combinado con RF (ROS + RF-BD, *Random Over-Sampling + Random Forest - Big Data*) y SVM (SVM-BD, *Support Vector Machine - Big Data*) implementadas en Spark, combinadas con ROS en MapReduce. Estos algoritmos fueron comparados con métodos tradicionales para la detección de ortólogos, como RBH (*Reciprocal Best Hits*), RSD (*Re-*

[1]Dos o más secuencias genéticas homólogas.

ciprocal Smallest Distance) y OMA[2]. Los resultados mostraron que SVM-BD con ROS superó a los métodos tradicionales. Aunque no se consideraron técnicas heurísticas como SMOTE. Además, el método con mejores resultados fue implementado en dos plataformas diferentes, lo cual podría dificultar la comparación debido al uso de arquitecturas distintas.

Lin et al. (2017) aplicaron modelos predictivos para el análisis de potenciales clientes en una compañía de seguros en China. Empleando un clasificador RF en Spark, que superó a SVM y LR en términos de rendimiento, medido con la métrica F y la media geométrica. Asimismo, para balancear las clases utilizaron SMOTE, cuyos resultados mostraron que al equilibrar las clases, los tres clasificadores obtuvieron mejores tasas de efectividad. Sin embargo, esto implicó un aumento en los tiempos de ejecución, obteniendo el menor tiempo el RF.

Gonzalez-Lopez et al. (2018) compararon tres estrategias para calcular el vecino más cercano en conjuntos de datos con más de dos etiquetas. La primera estrategia se basó en la transición iterativa de instancias, la segunda utilizó estructuras de índices distribuidas basadas en árboles, y la tercera empleó tablas de *hash* para agrupar instancias. Al igual que Juez-Gil et al. (2021b), concluyeron que las estrategias basadas en índices de árboles ofrecen mejor rendimiento y tiempos de ejecución más bajos. En este estudio, los autores reportan una reducción del tiempo de hasta 266 veces en velocidad en

[2]OMA es un proyecto que comprende diferentes algoritmos para la identificación de ortólogos y mejora la inferencia ortológica.

el conjunto de datos más grande. Sin embargo, algunos de los conjuntos de datos utilizados son relativamente pequeños para ser considerados big data, por ejemplo, el conjunto de datos *Flags* contenía solo 194 instancias. El mayor número de instancias en un conjunto de datos fue de 120,919 en *IMDB*, en términos de columnas, el conjunto de datos con el mayor número fue *Bookmarks* con 2,150 atributos.

Patil y Sonavane (2020) propusieron una técnica de sobremuestreo basada en métodos de agrupamiento, denominada CMSOT (*Clustering Minority Samples Oversampling Technique*). Los autores mencionan que esta técnica no solo aborda el problema de las clases no balanceadas, sino que también mitiga la problemática de los datos disyuntos. Para evaluar la afectividad de la técnica, utilizaron clasificadores RF y MLP, obteniendo un 6% de mejora en comparación con Borderline-SMOTE, ADASYN y MWMOTE, todas implementadas en MapReduce. Aunque los autores afirman que CMSOT es adecuada para problemas desbalanceados, no proporcionaron evidencia suficiente que respalde esta conclusión.

Chen et al. (2021) proponen una técnica de sobremuestreo diseñada para crear nuevos ejemplos de la clase minoritaria que pueden ubicarse en cuatro regiones del espacio: (i) región fronteriza de la clase minoritaria, (ii) región segura de la clase minoritaria, (iii) región fronteriza de la clase mayoritaria y (iv) región segura formada exclusivamente por ejemplos de la clase mayoritaria. Esta técnica consta de cuatro etapas: (i) se divide el espacio de entrada en cinco regiones, (ii) eliminan los ejemplos ruidosos de la clase

minoritaria, (iii) se forman grupos utilizando las instancias de la clase minoritaria y, finalmente, (iv) se realiza un proceso de sobremuestreo. El objetivo de esta técnica es incrementar el número de ejemplos de la clase minoritaria en la región fronteriza, donde suelen clasificarse incorrectamente. Al implementar una etapa de agrupamiento de la clase minoritaria, se garantiza que los ejemplos generados pertenezcan a su región correspondiente. Los resultados, comparados con SMOTE, ADASYN y Borderline-SMOTE, mostraron mejoras en términos de la media geométrica y la métrica F, aunque los conjuntos de datos utilizados eran relativamente pequeños. El conjunto de datos con más ejemplos tiene 5,472, y el de mayor número de atributos tiene 19.

Juez-Gil et al. (2021b) proponen una implementación de SMOTE, denominada *Aprox-SMOTE*, que emplea un cálculo aproximado al vecino más cercano mediante *Spill Trees*, una variación de los *Metric Trees*, que divide el espacio de tal manera que las instancias ubicadas en las mismas ramas del árbol se consideran vecinas. Aprox-SMOTE obtuvo resultados más rápidos en comparación con otras implementaciones de SMOTE, como SMOTE-BD de Basgall et al. (2019), que utiliza el cálculo exacto del vecino más cercano. Los autores destacan que su algoritmo ofrece una mayor escalabilidad sin comprometer la efectividad de los resultados. Los experimentos se llevaron a cabo en la infraestructura de Google, utilizando conjuntos de datos con hasta 7.2 millones de filas, aunque con un número reducido de columnas (28 en total). Los datos generados mediante SMOTE-BD y Aprox-SMOTE se utilizaron para entrenar un clasificador de bosques aleatorios. En términos de la métrica F y

la media geométrica, Aprox-SMOTE obtuvo mejores resultados que SMOTE-BD. Además, mostró una mejora significativa en los tiempos de ejecución, siendo hasta 28 veces más rápido que SMOTE-BD.

Sleeman IV y Krawczyk (2021a) llevaron a cabo una revisión exhaustiva de las técnicas de sobremuestreo aplicadas a conjuntos de big data con clases no balanceadas. Esta revisión abarcó los algoritmos para clases binarias y multiclase ofreciendo una comparación entre 14 algoritmos que los autores consideran representativos del estado del arte en técnicas de sobremuestreo. De estos 14 algoritmos, 7 son variantes de SMOTE. Los experimentos se realizaron utilizando 26 conjuntos de datos, ninguno de los cuales presentaba alta dimensionalidad, siendo el conjunto de datos con mayor número de atributos con 115 columnas. Para la evaluación comparativa, emplearon tres clasificadores: SVM, RF y NB. Los autores centraron su análisis en las dificultades asociadas con el manejo de instancias a nivel local al diseñar algoritmos para clases no balanceadas en big data, con un especial énfasis en la estrategia de partición que emplean las plataformas como Spark. En este sentido, los autores sugieren que una posible solución a estas dificultades es el desarrollo de enfoques combinados que integren técnicas de limpieza de datos, como la selección de instancias, o la combinación de métodos de submuestreo para mejorar el rendimiento de los algoritmos de sobremuestreo.

Maldonado et al. (2022) exploran las limitaciones de SMOTE en conjuntos de datos de alta dimensionalidad, particularmente debido al uso de la distancia euclidiana. Los autores proponen el uso de una métrica ponderada de

Minkowski que no asume que todas las características tienen la misma importancia. Con esta métrica, buscan definir de manera más precisa la vecindad de los ejemplos de la clase minoritaria, asignando mayor peso a las características más relevantes para la clasificación. La técnica fue evaluada en otros problemas, como el solapamiento, donde obtuvo mejores resultados en comparación con la técnica SMOTE tradicional. Sin embargo, la técnica no fue aplicada en conjuntos de datos masivos debido a los largos tiempos de ejecución. En este estudio, el conjunto de datos con la mayor cantidad de atributos contenía 17,404, aunque solo incluía 98 ejemplos. En sus conclusiones, los autores destacan la importancia de estudiar esta técnica en escenarios de big data.

Rodríguez-Torres et al. (2022) propusieron un método de sobremuestreo específicamente diseñado para conjuntos de datos masivos, con el objetivo principal de reducir los tiempos de ejecución. Los autores reportaron que su técnica es al menos dos veces más rápida en comparación con las técnicas mencionadas en la literatura. A diferencia de los métodos convencionales, que se basan en la distancia entre ejemplos de la clase minoritaria, su enfoque utilizó la moda, junto con los valores máximos y mínimos de cada característica, para generar nuevos ejemplos. Esta elección se justificó por el hecho de que la combinación de la moda con los valores extremos de las características produjo mejores resultados de sobremuestreo en comparación con el uso de la media, mediana, varianza o desviación estándar. Aquí, el conjunto de datos más grande empleado en los experimentos contó con 583,250 ejemplos y 78 características. Sin embargo, no evaluaron si esta nueva forma de generar ejemplos artificiales

es afectada por la maldición de la dimensionalidad.

2.2.2 Submuestreo

Triguero et al. (2016) aplicaron un algoritmo de submuestreo evolutivo (EUS, *Evolutionary Under-Sampling*) para tratar conjuntos de datos masivos con desigualdad extrema entre clases en el entorno de Apache Spark. El algoritmo base utilizado fue DT versión C4.5 y el proceso de submuestreo evolutivo se llevó a cabo mediante un algoritmo genético con una función de ajuste para seleccionar prototipos. Esta función busca equilibrar la reducción de los datos de entrenamiento con el desempeño de los clasificadores. La implementación del algoritmo en Spark mostró tiempos de ejecución más rápidos en comparación con una versión anterior desarrollada en MapReduce (Triguero et al., 2015). Asimismo, aunque EUS superó a RUS en términos de rendimiento, presentando tiempos de ejecución significativamente más rápidos, los autores no compararon su algoritmo con otras estrategias comúnmente utilizadas para abordar el problema de las clases no balanceadas, tales como SMOTE, SMC o los clasificadores sensibles al costo.

Ahlawat et al. (2019) propusieron un algoritmo de submuestreo que incorpora dos técnicas de agrupamiento, K-medias y C-medias, combinadas con lógica difusa. A diferencia de otras técnicas de submuestreo, esta propuesta no solo reduce el número de instancias de la clase mayoritaria, sino que también aplica submuestreo a la clase minoritaria, siempre que existan suficientes ejemplos. Si ambas clases cuentan con al menos 1,500 ejemplos, el submuestreo se

realiza en ambas, resultando en un total de 3,000 centros o medias de agrupamiento que conforman el conjunto preprocesado. En los casos donde la clase minoritaria no tiene suficientes ejemplos, solo se reduce la clase mayoritaria. Con esta técnica, los autores buscan equilibrar el número de ejemplos entre las clases mayoritaria y minoritaria a un ratio de desbalance de 1. Para demostrar la efectividad de la propuesta, los autores llevaron a cabo un estudio comparativo usando SMOTE, en donde la técnica propuesta fue superior en la mayoría de los casos. No obstante, es importante comentar que el estudio no abarcó conjuntos de datos con alta dimensionalidad, lo que limita su aplicabilidad a escenarios donde la distancia euclidiana presenta desventajas.

Jeon y Lim (2020) desarrollaron un método de submuestreo denominado PSU (*Particle Stacking Undersampling*), cuyo objetivo es minimizar la pérdida de información asociada con la eliminación de ejemplos, maximizando la distancia entre ellos. PSU emplea un índice de aptitud (*fitness index*), que se define como la suma de las distancias entre los datos muestreados y los originales, dividida por la suma de las distancias entre los propios datos muestreados. Cuanto más pequeño es el índice, menor es la pérdida de información. PSU fue comparado contra cuatro métodos de submuestreo: (i) RUS, (ii) agrupación de centroides (CC), y los algoritmos (iii) *NearMiss-1* y (iv) *NearMiss-2*. Los resultados, en términos de las métricas AUC-ROC y G-Mean, mostraron que, cuando el índice de aptitud es bajo, el rendimiento del método mejora significativamente en comparación con los otros enfoques. A pesar de que los autores afirman que su algoritmo puede escalarse a conjuntos

de datos grandes, los experimentos se realizaron utilizando conjuntos de datos clásicos y no consideraron el impacto de la partición de datos en entornos de big data.

Viloria et al. (2020) realizaron una comparación entre técnicas tradicionales de muestreo, como ROS, RUS y SMOTE, para mejorar la efectividad de las de redes neuronales profundas en conjuntos de datos de microarreglos de expresiones genéticas. Estos conjuntos de datos se caracterizan por tener una alta dimensionalidad y un número reducido de instancias. Por ejemplo, el conjunto de datos *Breast* contiene 25,410 características y 100 ejemplos. Para acrecentar el desbalance en los conjuntos de datos, los autores eliminaron aleatoriamente ejemplos de la clase minoritaria. Los resultados mostraron que tanto ROS como SMOTE mejoraron la efectividad del clasificador, en términos de AUC-ROC, mientras que RUS produjo resultados similares o inferiores en comparación con los datos originales. A pesar de que el fenómeno de la maldición de la dimensionalidad es una preocupación importante al utilizar SMOTE en conjuntos de datos con un gran número de atributos, este fenómeno no se observó en los experimentos. Sin embargo, los autores no mencionan ni discuten este aspecto, a pesar de que constituye uno de los principales desafíos al aplicar SMOTE en contextos de alta dimensionalidad.

Leevy et al. (2023) centraron su estudio en preprocesamiento y análisis del conjunto de datos "Credit Card Fraud", publicado en Kaggle[3], apli-

[3]`https://www.kaggle.com/`

cando algoritmos de clasificación para la detección de fraudes. El conjunto de datos original consta de 284,807 instancias, con 28 de 30 variables independientes transformadas mediante PCA, a excepción de las variables "Tiempo" y "Monto". La variable "Monto" fue normalizada para el estudio, mientras que "Tiempo" fue ignorada. El estudio incluyó modelos de aprendizaje automático como CatBoost, XGBoost, RF y LR, empleando la librería *Python Imblearn* para ajustar las proporciones de clases en el entrenamiento mediante RUS. Además de RUS, los autores propusieron el uso de un umbral para asignar etiquetas de clase a las puntuaciones de probabilidad de salida del modelo, bajo la restricción de $TPR \geq TNR$. La evaluación de los modelos se llevó a cabo mediante validación cruzada estratificada y la optimización de umbrales para mejorar la precisión en la clasificación de transacciones fraudulentas o legítimas. Los resultados indicaron que el ajuste de las proporciones de clases mediante submuestreo no mejoró significativamente el desempeño de los clasificadores, lo que sugiere que equilibrar artificialmente las clases puede aumentar la dificultad para separarlas correctamente. Por el contrario, la optimización de umbrales se mostró como una estrategia eficaz para mejorar el rendimiento del modelo, especialmente cuando no se equilibran desproporcionadamente las clases. Los mejores resultados se obtuvieron sin necesidad de ajustar las proporciones de clases, lo que resalta la importancia de seleccionar adecuadamente los umbrales de decisión en la detección de fraudes. Asimismo, el estudio sugiere que futuras investigaciones podrían aplicar este método en otros dominios y evaluar el uso de restricciones adicionales en la optimización de umbrales.

2.3 Técnicas Combinadas (Alta Dimensionalidad, Desbalance y Solapamiento de clases)

Uno de los primeros trabajos que combinan estrategias de diversas naturalezas es el de Fernández et al. (2017), quienes analizaron el comportamiento de ROS, RUS y SMOTE para tratar las clases no balanceadas en conjuntos de datos masivos. El estudio se llevó a cabo en dos subconjuntos extraídos del conjunto de datos ECBDL14[4]. Los resultados indicaron que SMOTE tuvo un rendimiento inferior en comparación con RUS y ROS, en donde, RUS mostró mejores resultados cuando se emplearon menos particiones en los datos, mientras que ROS se desempeñó mejor con un mayor número de particiones. RUS mostró mejores resultados cuando se utilizaron menos particiones en los datos, mientras que ROS se desempeñó mejor con un mayor número de particiones. No obstante, el estudio se limitó a comparar la G-Mean, sin utilizar otras métricas de desempeño como AUC-ROC. Además, la implementación de ROS y RUS se realizó en Spark utilizando el paquete *Imb-sampling-ROS-and-RUS*, mientras que SMOTE se implementó en Hadoop, lo que introduce una falta de consistencia en la comparación de los algoritmos, dificultando una evaluación precisa de su rendimiento. Finalmente, las conclusiones del estudio se basaron únicamente en el conjunto de datos ECBDL14, lo cual impide determinar si los resultados obtenidos son generalizables o si están influenciados por las características específicas de este conjunto de datos.

[4]Evolutionary Computation for Big Data and Big Learning Workshop Data Mining Competition 2014

En un enfoque que aborda no solo el desbalance entre clases, sino también el solapamiento, Abdel-Hamid et al. (2018) propusieron un algoritmo distribuido de muestreo de datos, denominado SBMF (*Spark Based Mining Framework*), que combina submuestreo y sobremuestreo en las fronteras de las clases para abordar ambas complejidades. El algoritmo consta de dos etapas principales: primero, elimina instancias de la clase mayoritaria que tienen un menor impacto en la clasificación; segundo, genera nuevas instancias para la clase minoritaria. SBMF se basa en el cálculo del vecino más cercano utilizando la distancia euclidiana. Los autores evaluaron el comportamiento del algoritmo tanto en conjuntos de datos masivos como de tamaño estándar. Los resultados demostraron que SBMF mejora los tiempos de ejecución y las tasas de clasificación en términos de AUC-ROC, métrica F y G-Mean, en comparación con otras técnicas utilizadas en este tipo de datos. Sin embargo, los autores no probaron su algoritmo en conjuntos de datos con alta dimensionalidad, lo que podría hacerlo vulnerable al fenómeno de la concentración de distancias, ya que el conjunto de datos con mayor dimensionalidad utilizado en el estudio, KDD99, cuenta con solo 42 atributos.

Rendón et al. (2020) y Johnson y Khoshgoftaar (2020) investigan el uso de remuestreo de datos como estrategia para mejorar el rendimiento de redes neuronales artificiales en conjuntos de datos masivos con clases no balanceadas. Johnson y Khoshgoftaar (2020) exploran el remuestreo aleatorio utilizando ROS, RUS y una combinación de ambos, encontrando que ROS ofrece los mejores resultados. Por su parte, Rendón et al. (2020) realizan un

estudio más detallado combinando técnicas de sobremuestreo y edición. En su enfoque, primero equilibran el conjunto de datos utilizando SMOTE y luego aplican técnicas de edición en la salida de la red neuronal para identificar y gestionar datos ruidosos. Esta estrategia es innovadora, ya que la edición se lleva a cabo en el espacio transformado por la red neuronal, en lugar del espacio de características original. Aunque las pruebas estadísticas (Friedman e Iman-Davenport) realizadas entre la técnica propuesta SMOTE+ENN* y otras variantes como ROS, SMOTE, SMOTE+ENN, y SMOTE+TL no mostraron mejoras significativas, los autores destacan la importancia de aplicar una estrategia a nivel de datos en la salida de la red neuronal para mejorar su rendimiento. Aunque el enfoque principal de este artículo es mejorar el rendimiento de las redes neuronales artificiales, también resalta la relevancia de desarrollar técnicas de preprocesamiento específicas para conjuntos de datos masivos. Este estudio subraya la necesidad de considerar enfoques innovadores en el preprocesamiento de datos para abordar los desafíos específicos de los conjuntos de datos masivos y desbalanceados.

Bauder y Khoshgoftaar (2020) se enfocan en problemas de clasificación binaria para la detección de fraudes en datos de seguros médicos (*Medicare*). Los autores combinan datos de cuatro años consecutivos (2012-2015), seleccionando los atributos más comunes. El conjunto de datos resultante presenta un desbalance extremo entre las clases (casos válidos frente a fraudes). Para mitigar la degradación del desempeño del clasificador debido a este desbalance, se emplean técnicas de remuestreo aleatorio como ROS y RUS. De estas, RUS

mejoró los resultados en una mayor cantidad de modelos en comparación con ROS. Sin embargo, este estudio no incorpora métodos de muestreo inteligente de datos.

Bagui y Li (2021) aplican métodos combinados de muestreo de datos para tratar conjuntos de datos no balanceados en sistemas de detección de intrusos en redes informáticas. Utilizan diversas combinaciones de técnicas de remuestreo, como ROS, RUS, la combinación de ROS con RUS, RUS con SMOTE, y RUS con ADASYN. Los conjuntos de datos resultantes se clasifican utilizando una ANN. Los experimentos se llevan a cabo en un clúster EMR de Amazon AWS utilizando Spark. De los resultados, se extraen dos conclusiones principales: (i) la combinación de ROS y RUS ofrece el mejor rendimiento para clasificar la clase minoritaria en escenarios de alto desbalance y (ii) el remuestreo de datos no demuestra una diferencia significativa en la clasificación cuando los datos no están extremadamente desbalanceados.

Sleeman IV y Krawczyk (2021b) proponen un marco de trabajo orientado a problemas multiclase que incorpora una modificación de SMOTE, considerando las características locales del conjunto de datos. Además, esta propuesta incluye un proceso de submuestreo. Los experimentos se realizaron utilizando cinco conjuntos de datos masivos con diferentes números de clases. Sin embargo, el conjunto de datos con el mayor número de atributos tenía solo 55, lo que limita el alcance de la evaluación en términos de alta dimensionalidad.

Pokhrel y Wang (2020) adaptan un algoritmo de agrupamiento para

datos de alta dimensionalidad denominado vecino más cercano compartido (*Shared Nearest Neighbor, SNN*), que utiliza árboles k-d (*k-Dimensional Tree*). Esta técnica de partición del espacio se emplea para realizar búsquedas, y en este caso, para calcular el vecino más cercano en Spark, aprovechando la partición del conjunto de datos para mejorar los tiempos de cómputo. El algoritmo divide los datos en el espacio utilizando árboles k-d, realiza agrupamientos locales en las particiones y, finalmente, combina todos los resultados de los agrupamientos locales en agrupamientos globales. El algoritmo propuesto mostró mejoras de hasta 10 veces en los tiempos de ejecución. Aunque se mencionan las bases de datos utilizadas en el estudio (*Bristol, Complex9 y TLC Trip*), no se documentan sus características.

Juez-Gil et al. (2021a) llevan a cabo un estudio comparativo entre diferentes métodos de remuestreo y SMC. En total, se utilizaron cinco métodos de remuestreo y tres sistemas múltiples de clasificación. Una característica destacada de la investigación es el uso de 16 conjuntos de datos masivos, de los cuales dos presentan una dimensionalidad de 893 atributos. Las conclusiones del estudio resultan algo ambiguas en comparación con lo que se sabía en problemas estándar (no big data), donde la combinación de remuestreo y SMC solía ser una estrategia altamente efectiva. Sin embargo, los resultados experimentales indicaron que, en el contexto de big data, las técnicas de remuestreo pueden afectar negativamente el desempeño del SMC.

Gong et al. (2021) proponen calcular el vecino más cercano en cada una de las particiones de datos realizadas por Spark, en lugar de hacerlo sobre

todo el conjunto de datos. Esta estrategia incrementa la velocidad del cálculo de distancias. El algoritmo propuesto se utiliza para la selección de instancias en conjuntos de datos de hasta 11 millones de filas. No obstante, al igual que en otros estudios revisados, la cantidad de columnas en los conjuntos de datos es relativamente baja, con el conjunto de datos más extenso presentando solo 129 columnas.

El estudio de Petinrin et al. (2023) examina el uso de técnicas de preprocesamiento y análisis para seis conjuntos de datos de expresión génica asociados a diferentes tipos de cáncer, aplicando métodos avanzados de aprendizaje automático para abordar la alta dimensionalidad y las clases no balanceadas, características inherentes a este tipo de datos. Los conjuntos de datos incluyen muestras de cáncer de cerebro, colon, leucemia, linfoma, próstata y tumor de células redondas azules pequeñas (SBRCT), cada uno con características específicas y preprocesamientos detallados en estudios anteriores. Para gestionar la alta dimensionalidad, se emplearon métodos de reducción de dimensiones como el PCA, la Descomposición de Valor Singular Truncado (TSVD) y la Incrustación Estocástica de Vecinos Distribuidos en t (tSNE). Además, se aplicó la normalización *min-max* para equilibrar las características antes del análisis, lo cual es crucial para asegurar una contribución equitativa de cada característica. En cuanto al desbalance de clases, se implementó un sobremuestreo utilizando la técnica SVMSMOTE, que se enfoca en generar nuevas muestras en las fronteras de decisión de las clases minoritarias. El estudio también incorporó métodos de clasificación para la selección de características, utilizando

algoritmos como el RF y la LR, que demostraron ser efectivos para identificar las características más relevantes. Los modelos se evaluaron utilizando métricas de precisión, *recall*, exactitud y métrica F1. La mejora en el rendimiento tras aplicar técnicas de sobremuestreo y reducción de dimensiones fue notable, indicando que estas estrategias permiten una mejor interpretación y robustez de los modelos. Los resultados sugieren que PCA y TSVD superaron a tSNE, que mostró un rendimiento generalmente inferior y es recomendado principalmente para la visualización de datos. El estudio destaca la efectividad de la técnica de sobremuestreo SVMSMOTE y de los métodos de reducción de dimensiones para mejorar la precisión y la interpretabilidad de los modelos de clasificación en conjuntos de datos de expresión génica. Los métodos de selección de características basados en clasificadores también demostraron ser valiosos, ofreciendo un enfoque efectivo para manejar tanto la alta dimensionalidad como las clases no balanceadas en datos médicos y bioinformáticos. Finalmente, los autores proponen que futuras investigaciones exploren otras técnicas de sobremuestreo y reducción de dimensiones en distintos conjuntos de datos de microarreglos para seguir mejorando la precisión y robustez de los modelos de clasificación en estos contextos.

Abdelkhalek y Mashaly (2023) abordan el problema de las clases no balanceadas en el conjunto de datos NSL-KDD, con un enfoque en mejorar la detección de clases minoritarias en sistemas de detección de intrusiones (NIDS). El método propuesto combina técnicas de sobremuestreo y submuestreo, utilizando la creación de muestras sintéticas adaptativas (ADASYN) seguida de

Tomek Links para eliminar redundancias. Se eligió el conjunto de datos NSL-KDD, una mejora del KDD'99, por su estructura optimizada que evita registros redundantes y duplicados, facilitando un entrenamiento más preciso. Este conjunto de datos incluye registros etiquetados como tráfico normal y diversos tipos de ataques, distribuidos de manera desigual, lo que representa un reto significativo para la detección eficaz de ataques menos frecuentes. La estrategia de preprocesamiento de datos incluyó la codificación *One-Hot* de atributos no numéricos y la normalización mediante *MinMaxScaler* para adecuar los datos al entrenamiento de modelos de aprendizaje profundo. Los modelos implementados abarcaron desde MLP hasta redes neuronales convolucionales (CNN, *Convolutional Neural Network*) y combinaciones de CNN con LSTM bidireccional (CNN-BLSTM, *Convolutional Neural Network-Bidirectional Long-short Term Memory*), seleccionados por su eficacia demostrada en tareas similares. Estos modelos se evaluaron utilizando métricas como precisión, *recall* y métrica F, comparando los rendimientos antes y después de aplicar las técnicas de remuestreo. Los resultados indicaron mejoras significativas en la detección de clases minoritarias al utilizar técnicas de remuestreo, en comparación con los modelos que no las emplearon. Este estudio demuestra que la combinación de ADASYN y Tomek Links puede mejorar sustancialmente la detección de ataques en NIDS, especialmente en el contexto de clases minoritarias en conjuntos de datos no balanceados. Los hallazgos sugieren que este método no solo aumenta la precisión del modelo, sino que también sienta las bases para la implementación de sistemas NIDS de dos etapas que primero detectan la

presencia de un ataque y luego clasifican su tipo. Los autores proponen que futuras investigaciones exploren la aplicación de esta técnica en otros conjuntos de datos de detección de intrusiones con mayores desbalances y experimenten con una variedad más amplia de técnicas de remuestreo y arquitecturas de aprendizaje profundo, con el objetivo de optimizar aún más la detección de ataques.

Vairetti et al. (2024) introducen SMOTENN, una técnica avanzada para el remuestreo combinado de datos en contextos de clasificación no balanceada, que combina en una sola pasada técnicas de sobremuestreo y submuestreo. El objetivo principal de SMOTENN es mejorar la eficiencia y escalabilidad del remuestreo de datos, utilizando un marco de trabajo distribuido para la carga y procesamiento de datos, una métrica de distancia rápida y escalable, y un algoritmo que integra las técnicas de sobremuestreo de SMOTE con submuestreo basado en edición. SMOTENN se basa en el marco MapReduce, que facilita el manejo de grandes volúmenes de datos mediante la distribución de tareas en múltiples sistemas, y en Spark para proporcionar una abstracción de alto nivel que permita un procesamiento de datos distribuido eficiente. Además, utiliza un algoritmo de vecinos más cercanos aproximado para definir la vecindad de las muestras, adaptando así la métrica de distancia al remuestreo de datos combinado. Este método permite una reducción significativa en la complejidad del aprendizaje al eliminar el ruido antes del sobremuestreo y generar ejemplos sintéticos para la clase minoritaria a través de la interpolación, manteniendo un equilibrio entre las muestras de las clases mayoritaria y minoritaria en cada

vecindad definida. En los experimentos realizados, SMOTENN demostró ser efectivo en 35 conjuntos de datos de diferentes tamaños, con un conjunto de datos que contenía hasta 5,578,255 instancias y otro con un máximo de 54 atributos. Los resultados mostraron que SMOTENN tiene un rendimiento superior en términos de media geométrica en comparación con otras técnicas de remuestreo conocidas. Su principal ventaja radica en su capacidad para reducir eficazmente el ruido en las regiones fronterizas y obtener resultados robustos bajo diferentes configuraciones de parámetros. El análisis sugiere que, aunque técnicas simples como el submuestreo aleatorio pueden ser efectivas en configuraciones a gran escala, el remuestreo combinado inteligente tiene un papel valioso, especialmente en entornos de big data.

La Tabla 2.1 presenta un resumen de los principales artículos revisados en este capítulo, lo que permite observar la relevancia del problema de las clases no balanceadas en el contexto de big data. Además, se destaca que, en la mayoría de los casos, este problema no se aborda de manera conjunta con otras complejidades como la alta dimensionalidad o el solapamiento lo que señala una oportunidad de investigación en la integración de soluciones para múltiples desafíos en conjuntos de datos masivos.

Tabla 2.1: Resumen de técnicas y complejidades.

Referencia	Tipo de Solución	Complejidad Tratada	Número de Atributos	Número de Renglones
Pengfei et al. (2014)	Sobremuestreo	Clases no balanceadas	36	6,435
Galpert et al. (2015)	Técnicas Combinadas	Clases no balanceadas	6	29,887,416
Triguero et al. (2016)	Submuestreo	Clases no balanceadas	631	17,445,419
Fernández et al. (2017)	Técnicas Combinadas	Clases no balanceadas	90	12,000,000
Lin et al. (2017)	Técnicas Combinadas	Clases no balanceadas	16	500,000
Abdel-Hamid et al. (2018)	Técnicas Combinadas	Clases no balanceadas y solapamiento	42	4,856,151
Gonzalez-Lopez et al. (2018)	Algoritmo	Clases no balanceadas y multi-clase	2,150	87,856
Ahlawat et al. (2019)	Submuestreo	Clases no balanceadas	11	56,252
Bauder y Khoshgoftaar (2020)	Técnicas Combinadas	Clases no balanceadas	No específica	3,692,555
Jeon y Lim (2020)	Submuestreo	Clases no balanceadas	42	2,233
Johnson y Khoshgoftaar (2020)	Técnicas Combinadas	Clases no balanceadas	125	4,692,370
Patil y Sonavane (2020)	Sobremuestreo	Clases no balanceadas y datos disjuntos	4,932	12,678
Pokhrel y Wang (2020)	Algoritmo	Clases no balanceadas y agrupamientos	No específica	No específica
Rendón et al. (2020)	Técnicas Combinadas	Clases no balanceadas	224	111,104
Viloria et al. (2020)	Técnicas Combinadas	Clases no balanceadas	15,214	260
Bagui y Li (2021)	Técnicas Combinadas	Clases no balanceadas	No específica	151,887
Chen et al. (2021)	Sobremuestreo	Clases no balanceadas y ruido	19	2,308
Gong et al. (2021)	Técnicas Combinadas	Clases no balanceadas y selección de instancias	129	11,000,000
Juez-Gil et al. (2021b)	Sobremuestreo	Clases no balanceadas	28	7,284,166
Juez-Gil et al. (2021a)	Algoritmo	Clases no balanceadas	893	9,998,491
Sleeman IV y Krawczyk (2021a)	Sobremuestreo	Clases no balanceadas	116	No específica
Sleeman IV y Krawczyk (2021b)	Técnicas Combinadas	Clases no balanceadas	115	3,000,000
Maldonado et al. (2022)	Sobremuestreo	Clases no balanceadas y alta dimensionalidad	17,404	98
Rodríguez-Torres et al. (2022)	Sobremuestreo	Clases no balanceadas	78	583,250
Leevy et al. (2023)	Submuestreo	Clases no balanceadas	30	284,807
Abdelkhalek y Mashaly (2023)	Técnicas Combinadas	Clases no balanceadas y alta dimensionalidad	41	251,946
Petinrin et al. (2023)	Técnicas Combinadas	Clases no balanceadas y alta dimensionalidad	6,034	102
Vairetti et al. (2024)	Técnicas Combinadas	Clases no balanceadas y alta dimensionalidad	54	464,677

Capítulo 3

Metodología

Este capítulo expone el enfoque propuesto en esta investigación, así como los aspectos metodológicos necesarios para entender su desarrollo y proceso experimental. Se comienza con el diseño, estructura y descripción del "Método Sistemático" propuesto, que es detallado en la Sección 3.1. Asimismo, se especifica la infraestructura, las características de los conjuntos de datos (los cuales incluyen las problemáticas de clases no balanceadas, alta dimensionalidad y solapamiento), las técnicas y los métodos utilizados en los experimentos (secciones 3.3, 3.4).

3.1 Método Sistemático

Se realizó un análisis exhaustivo, sobre el problema de las clases no balanceadas con solapamiento y alta dimensionalidad en entornos de big data, donde se identificaron sus características específicas y los desafíos asociados. Incluyó una revisión detallada de las técnicas de sobremuestreo, de las que atienden la problemática de la alta dimensionalidad y el solapamiento entre clases, y se exploraron soluciones propuestas mediante tecnologías de cómputo distribuido. Se reprodujeron algoritmos destacados en la literatura científica (en entornos de big data) que atienden o tratan con los problemas antes mencionados, destacándose aquellos que utilizan el método de los k-vecinos más cercanos (edición de Wilson y SMOTE), así como las distancias fraccionarias para reducir la dimensionalidad de los datos. Esta reproducción permitió una comprensión más profunda de los fundamentos de dichas técnicas y sirvió como punto de referencia para el desarrollo del método propuesto.

1. **Espacios de similitud para reducir la dimensionalidad**. Los espacios de disimilitud propuesto por Pękalska et al. (2006) y Duin y Pękalska (2012), representa las instancias en función de su relación con un conjunto de prototipos, lo que genera un espacio transformado en el que las nuevas variables ya no son atributos originales, sino medidas de similitud (o distancia) a dichos prototipos, y permite la reducción directa de la dimensionalidad de los datos (ver sección 3.2). Este enfoque preserva relaciones estructurales entre los datos, al reformular el problema en un

espacio donde la representación está determinada por la relación con múltiples referencias, y se favorece la separabilidad entre clases, facilitando el trabajo de los clasificadores. Además, este tipo de transformación puede ser útil también para manejar relaciones no lineales entre clases. En este trabajo, se estudiaron espacios de disimilitud con dimensiones de 10, 50, 100, 200, 500, 1000 y 2000 variables.

Por otro lado, en espacios de alta dimensionalidad, la distancia euclidiana pierde su capacidad de discriminar efectivamente entre instancias, ya que las diferencias relativas entre distancias tienden a reducirse, un fenómeno conocido como la concentración de distancias. Aggarwal et al. (2001) y Francois et al. (2007) demostraron que las distancias fraccionarias mitigan este efecto, ya que aumentan el contraste entre instancias cercanas y lejanas. Este enfoque es particularmente valioso cuando los algoritmos dependen explícitamente de métricas de distancia, ya que mejora la sensibilidad del modelo a las relaciones locales entre datos en espacios de alta dimensionalidad. Por tanto, su incorporación contribuye directamente a enfrentar el problema de la alta dimensionalidad. Debido a las características de big data y alta dimensionalidad de los conjuntos de datos utilizados, no se hizo una búsqueda de la distancia fraccionaria óptima, ya que representaría requerimientos de cómputo más extensos (Francois et al., 2007; Flexer y Schnitzer, 2015). Para analizar el comportamiento de diferentes distancias fraccionarias se utilizó como base la distancia propuesta por Aggarwal et al. (2001) de 0.50, junto a las distancias adi-

cionales de 0.75, 0.66, 0.33 y 0.25 (a ± 0.25 y ± 0.16 aproximadamente de 0.50).

2. **Tratar el problema del desbalance de clases por medio de técnicas de sobremuestreo**. El desbalance de clases afecta el rendimiento de los clasificadores, al sesgarlos hacia la clase mayoritaria. SMOTE al generar instancias sintéticas interpolando entre ejemplos cercanos de la clase minoritaria, mejora la representación de esta clase sin duplicar instancias, como en el sobremuestreo aleatorio (Chawla et al., 2002; Blagus y Lusa, 2013; Fernandez et al., 2018). En este trabajo, aplicar SMOTE dentro del espacio de disimilitud ofrece ventajas adicionales: la generación sintética se realiza sobre representaciones más discriminativas y con menor dimensionalidad, donde las fronteras de clase están mejor definidas, lo cual reduce el riesgo de introducir ruido o solapamiento artificial (sección 3.3).

3. **Emplear técnicas de edición de datos para reducir el solapamiento entre clases**. La edición de datos propuesta por Wilson (Wilson, 1972), es una técnica de limpieza de datos que elimina instancias mayoritarias mal clasificadas por un clasificador de primer orden (generalmente k-NN), asumiendo que estas se ubican en regiones de solapamiento o ruido. Esta técnica ha demostrado mejorar la precisión de clasificación al suavizar las fronteras entre clases, lo cual es especialmente útil en problemas con clases parcialmente solapadas o fronteras difusas (Wilson y Martinez,

1997). Al igual que con SMOTE, su aplicación en el espacio de disimilitud maximiza su eficacia, ya que las distancias entre instancias reflejan con mayor fidelidad la similitud estructural, y las regiones ambiguas están más claramente delimitadas, además de tener una menor dimensionalidad. Así, la edición de Wilson contribuye tanto a limpiar las fronteras de clase como a preparar un conjunto de datos más adecuado para la aplicación de modelos de clasificación (para mayor detalle, véase la sección 3.4.2).

3.2 Espacios de Disimilitud en Big Data

En el espacio de disimilitud, las dimensiones se definen mediante vectores que cuantifican la similitud entre los ejemplos y ciertos elementos o prototipos pertenecientes a un conjunto de representación denominado "R". Es fundamental que R incluya ejemplos de todas las clases, los cuales pueden seleccionarse a través de diversas estrategias. Por ejemplo, R puede definirse como el conjunto de entrenamiento completo, un subconjunto derivado mediante un mecanismo heurístico, una selección aleatoria, o incluso elementos extraídos del conjunto de datos de prueba o entrenamiento (Duin y Pękalska, 2012). Para los experimentos descritos en esta sección, el conjunto R fue conformado mediante una selección aleatoria del conjunto de entrenamiento, ya que este enfoque ha demostrado ser efectivo en la mayoría de los contextos (Pękalska et al., 2006).

La conversión del espacio de características original al espacio de dis-

imilitud se realiza mediante el cálculo de una métrica de distancia, como la distancia euclidiana, entre el conjunto R y cada uno de los ejemplos de los conjuntos de entrenamiento y prueba (Pękalska y Duin, 2001, 2002). En este nuevo espacio, las dimensiones no representan atributos originales, sino la disimilitud entre los ejemplos y los prototipos en R. El objetivo de esta transformación es convertir el problema de clasificación a un espacio donde las relaciones entre clases sean más evidentes, facilitando su discriminación.

La lógica subyacente de esta transformación reside en la capacidad de las métricas de disimilitud para reflejar las diferencias intrínsecas entre clases de una manera más efectiva. En un espacio de disimilitud bien estructurado, las distancias entre los ejemplos de la misma clase deberían ser relativamente pequeñas, mientras que las distancias entre ejemplos de diferentes clases deberían ser más grandes. Esto mejora el rendimiento de los clasificadores tradicionales en problemas de alta dimensionalidad o con características complejas, al reducir la influencia de atributos irrelevantes y facilitar la separación entre clases (Pękalska et al., 2006; García et al., 2015).

Formalmente, si definimos $\mathbf{R} = \{r_1, r_2, ..., r_k\}$ y el conjunto de entrenamiento como $\mathbf{E}$, donde e es cualquier ejemplo de $\mathbf{E}$, entonces cualquier vector en el espacio de disimilitud se denota como d_e, donde:

$$d_e = [d(e, r_1), d(e, r_2), ..., d(e, r_k)]$$

en la que $d()$ es la métrica de disimilitud aplicada. Este vector d_e representa las distancias entre el ejemplo e y cada uno de los prototipos $r_i \in R$. Así, el espacio

de disimilitud asociado al conjunto de entrenamiento $\mathbf{E}$ se describe mediante la matriz de disimilitud $\mathbf{D_E}$, que agrupa todos los vectores de disimilitud d_e correspondientes a los ejemplos en $\mathbf{E}$.

Una ventaja clave de este enfoque es su capacidad para abordar problemas de alta dimensionalidad, donde las relaciones entre atributos pueden ser difíciles de capturar mediante métodos tradicionales. La transformación al espacio de disimilitud reduce la dimensionalidad en los datos al centrarse en las distancias con respecto a los prototipos, lo que puede mejorar la eficiencia computacional y la capacidad del modelo para generalizar en problemas de clasificación complejos. Además, el uso de métricas de distancia más sofisticadas, como las distancias fraccionarias, permitiría una mejor construcción del espacio de disimilitud en problemas específicos de alta dimensionalidad, reforzando su aplicabilidad en escenarios donde la distancia euclidiana tradicional es ineficiente.

Como se discutió anteriormente en la Sección 1.1.3, el uso de la distancia euclidiana presenta limitaciones importantes en espacios de alta dimensionalidad. En consecuencia, para los experimentos en esta sección, se optó por utilizar distancias fraccionarias, dadas las características de alta dimensionalidad de los conjuntos de datos. Las distancias fraccionarias seleccionadas, que son consistentes con las utilizadas en la Sección 3.3, incluyen valores de 0.25, 0.33, 0.50, 0.66 y 0.75. Esta elección se fundamenta en la búsqueda de métricas de distancia que proporcionen una interpretación más adecuada en contextos de alta dimensionalidad, facilitando un análisis más efectivo y preciso de los

datos.

Asimismo, se crearon conjuntos R de diferentes tamaños para generar conjuntos de datos con vectores de disimilitud de distintas dimensionalidades. La creación de conjuntos de datos con diferentes tamaños de dimensionalidad permitirá analizar el impacto del balanceo de las clases en el nuevo espacio transformado en función del número de atributos. Los tamaños seleccionados para R fueron: 10, 50, 100, 200, 500, 1000 y 2000. En cada caso, R se configuró para asegurar una representación equitativa de ejemplos tanto de la clase mayoritaria como de la minoritaria.

El método adoptado para realizar la transformación al espacio de disimilitud se describe en detalle en el Algoritmo 1.

Algoritmo 1: Transformación al espacio de disimilitud.

Resultado: Conjuntos de datos $\mathbf{D_E}$ y $\mathbf{D_P}$ en el espacio de disimilitud
Entrada: Número n de ejemplos a tomar de cada clase para $\mathbf{R}$, coeficiente para el cálculo de distancias *distCoef*, conjuntos de datos de entrenamiento $\mathbf{E}$ y prueba $\mathbf{P}$

```
// Función para calcular la distancia de Minkowski de dos vectores con cualquier
   coeficiente distCoef.
```

Función `distancia(`*x, y, distCoef*`):`

```
    // Inicializar la variable suma en 0.0.
```

suma = 0.0 ;

```
    // Obtener el número de elementos del vector x.
```

dimensiones = *x.longitud*;

```
    // Ejecutar el ciclo For para todos los elementos de x y y.
```

para $i \leftarrow 0$ **a** *dimensiones* **hacer**

```
        // Acumular en suma la diferencia de cada elemento de x y y y elevarlo a la
           potencia de distCoef.
```

$suma+ = |x(i) - y(i)|^{distCoef}$;

fin

```
    // Elevar a la potencia de 1/distCoef el resultado de suma.
```

devolver $suma^{1/distCoef}$;

```
// Función para calcular la disimilitud entre dos elementos.
```

Función `disimilitud(`*D*`):`

```
    // Obtener el número de renglones de R, R es accedida por medio del Broadcast.
```

longitud = *número de renglones de* $\mathbf{R}$;

```
    // Inicializar una variable tipo arreglo vacía para almacenar el resultado de
       disimilitud.
```

datosDisimilitud = Arreglo *de* Vectores *vacío*;

```
    // Realizar el ciclo For para cada elemento de D.
```

para cada $d \in D$ **hacer**

```
        // Inicializar un vector vacío, para ir acumulando la disimilitud entre cada
           elemento.
```

vectorTemp = Vector *de tamaño* longitud;

```
        // Realizar ciclo For por cada elemento de R.
```

para $i \leftarrow 0$ **a** *longitud* **hacer**

```
            // Calcular la disimilitud entre d y el elemento R[i].
```

temp = *llamar a la función* distancia(d,R[i],distCoef);

agregar *temp* al vector *vectorTemp*;

fin

agregar *vectorTemp* al arreglo *datosDisimilitud* con etiqueta de clase de d;

fin

devolver *datosDisimilitud*;

$\mathbf{R} \leftarrow$ Seleccionar aleatoriamente n instancias de cada clase del conjunto de datos de entrenamiento;

Realizar *Broadcast* de $\boldsymbol{R}$ a todos los nodos del clúster;

$\mathbf{D_E} \leftarrow$ Llamar a la función *disimilitud*($\mathbf{E}$);

$\mathbf{D_P} \leftarrow$ Llamar a la función *disimilitud*($\mathbf{P}$);

Guardar $\mathbf{D_E}$ y $\mathbf{D_P}$ en archivos con formato *LIBSVM*;

3.3 Clases No Balanceadas y Alta Dimensionalidad

En el ámbito del big data, los algoritmos de sobremuestreo suelen adoptar un enfoque de "dividir y conquistar", utilizando plataformas como Hadoop y Spark. Estas herramientas permiten segmentar grandes volúmenes de datos en particiones, que son procesadas de manera independiente en múltiples nodos de cómputo. Este enfoque facilita la gestión de datos a gran escala que serían inabordables en un sistema único. No obstante, también introduce desafíos para los métodos originalmente diseñados para operar en un entorno centralizado, dado que estos no pueden evaluar el conjunto de datos completo. En particular, técnicas como SMOTE presentan ciertas limitaciones en este paradigma, dado que el cálculo de los vecinos más cercanos se restringe a las instancias contenidas dentro de la misma partición. Esto puede reducir la efectividad del método al no considerar ejemplos relevantes de otras particiones, comprometiendo el desempeño de dicho algoritmo (Basgall et al., 2019).

Para mitigar la anterior limitante, Maillo et al. (2017) desarrollaron una solución innovadora denominada kNN-IS, que facilita la determinación de los vecinos más cercanos en Spark, y que fue posteriormente empleada por Basgall et al. (2018) para implementar el algoritmo SMOTE-BD. El método kNN-IS utiliza la distancia euclidiana para medir la similitud entre ejemplos, aunque no aborda explícitamente los desafíos asociados con la alta dimensionalidad y su impacto en el cálculo de dichas distancias.

En esta sección, se exploran y adaptan las técnicas de ROS, SMOTE-BD y kNN-IS, utilizando Spark 3.1 para el procesamiento de datos en formato

LIBSVM. Se introdujeron modificaciones específicas para calcular distancias mediante distancias fraccionarias, con el objetivo de superar las restricciones impuestas por la alta dimensionalidad en el cálculo de los vecinos más cercanos. Asimismo, se detallan los datos utilizados en esta investigación, los algoritmos de muestreo, clasificadores y la infraestructura de hardware y software con la cual se contó para el desarrollo experimental. ROS se incluyó como una técnica base que permita comparar la efectividad del método SMOTE-BD desarrollado en esta tesis para entornos de alta dimensionalidad y big data.

3.3.1 Conjuntos de Datos

Los experimentos se realizaron conjuntos de datos modificados del original, KDD 2010, en los cuales se redujeron el número de instancias y atributos. KDD 2010 es un conjunto de datos de baja densidad, es decir, que la mayor parte de sus datos contienen valores de 0. Por lo tanto, para superar esta deficiencia se utilizó la técnica de PCA, la cual permitió generar conjuntos de datos densos.

El conjunto de datos KDD 2010 fue extraído del repositorio de LIBSVM[1], el cual contiene dos clases, 19,264,097 instancias y 1,163,024 atributos. Para simular un escenario de clases no balanceadas, se modificó el conjunto de datos original para obtener una proporción de 1:10 entre la clase minoritaria y la clase mayoritaria. Debido a las limitaciones en la infraestructura, se redujo el conjunto de datos a 30,000 instancias para los experimentos. Con el fin de

[1] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

investigar el impacto de la dimensionalidad en la efectividad de SMOTE, se generaron conjuntos de datos con diferentes cantidades de atributos, variando desde 50 hasta 900, en incrementos de 100, manteniendo constante el ratio de desbalance.

Además, para obtener un conjunto de datos de alta densidad, se aplicó el PCA, el cual facilita la extracción de características identificando los vectores propios (eigen-vectores) correspondientes a los mayores valores propios (eigen-valores) derivados de la matriz de covarianza (Wold et al., 1987). Estos eigen-vectores se utilizan para proyectar los datos en un nuevo subespacio que conserva las características esenciales de los datos originales, manteniendo la misma o una menor dimensionalidad. Tras aplicar PCA, los atributos se transformaron en valores densos: por ejemplo, el conjunto de datos de 100 dimensiones, que inicialmente tenía solo un 2.88% de valores diferentes de cero, pasó a tener todos sus valores distintos de cero.

Para reducir el número de filas y columnas del conjunto de datos, se empleó el Algoritmo 2.

Algoritmo 2: Obtención de conjuntos de datos reducidos.

Resultado: Conjunto de datos reducido.
Entrada: dirArchivo, semilla, renglones, columnas, dirNuevoArchivo

1 *datosOriginales* ← leer el archivo LIBSVM original en *dirArchivo*;
2 *datosOriginales* ← mezclar datos utilizando *semilla*;
3 *datosReducidos* ← primeros *renglones* de *datosOriginales*;
4 *datosReducidos* ← primeras *columnas* de *datosReducidos*;
5 grabar *datosReducidos* en *dirNuevoArchivo* con formato LIBSVM;

En una segunda etapa, los conjuntos de datos anteriores se modificaron

para lograr una proporción de desbalance 1:10, utilizando el Algoritmo 3.

Algoritmo 3: Obtención de conjuntos de datos no balanceados.

Resultado: Conjunto de datos con una proporción de desbalance de 1:10.

Entrada: dirArchivo, semilla, ratioDesbalance, dirNuevoArchivo, etiquetaMayoritaria, etiquetaMinoritaria

1 *datosOriginales* ← leer el archivo LIBSVM en *dirArchivo*;
2 *datosClaseMayoritaria* ← instancias de *datosOriginales* con *etiquetaMayoritaria*;
3 *datosClaseMinoritaria* ← instancias de *datosOriginales* con *etiquetaMinoritaria*;
`// Calcular la cantidad de instancias que debe tener la clase minoritaria. El ratioDesbalance es un número flotante; para un desbalance 1:10, es 0.1.`
4 *renglonesParaClaseMin* ← tamaño de *datosClaseMayoritaria* multiplicado por *ratioDesbalance*;
`// Calcular cuántas instancias deben tener la clase minoritaria para lograr el desbalance deseado.`
5 *renglonesClaseMinDesbalance* ← tamaño de *datosClaseMinoritaria* menos *renglonesParaClaseMin*;
`// Tomar la cantidad de instancias necesarias de la clase minoritaria para lograr el desbalance deseado.`
6 *nuevosDatosClaseMin* ← primeros *renglonesClaseMinDesbalance* de *datosClaseMinoritaria*;
`// Juntar en un DataFrame la clase mayoritaria y la clase minoritaria reducida.`
7 *datosDesbalanceados* ← unión de *datosClaseMayoritaria* con *nuevosDatosClaseMin*;
8 grabar *datosDesbalanceados* en *dirNuevoArchivo* con formato LIBSVM;

Una vez modificados los conjuntos de datos para lograr el desbalance, la densidad y las distintas dimensionalidades, se procedió a balancearlos mediante: ROS y SMOTE-BD. ROS se incluye como una técnica que nos permitirá comparar la efectividad de la versión de SMOTE-BD desarrollada en este tra-

bajo, con respecto a un método clásico y simple, pero de un buen desempeño, como se ha evidenciado en la literatura especializada en desbalance de clases.

En la implementación de SMOTE-BD, se evaluaron distintas métricas de distancia, incluidas la distancia euclidiana y distancias fraccionarias, con valores de 0.75, 0.66, 0.50, 0.33 y 0.25. Para ROS, la generación de copias de las instancias minoritarias fue realizada empleando la función *sample* proporcionada por Spark, utilizando el lenguaje de programación Scala.

3.3.2 Algoritmos de Sobremuestreo y Clasificadores

SMOTE-BD integra de manera eficaz las capacidades de distribución de datos de Spark para calcular los vecinos más cercanos mediante el algoritmo kNN-IS basado en el paradigma de *MapReduce*. En la fase de *Map*, el método kNN-IS distribuye los datos de entrenamiento en varias particiones, donde calcula las distancias y determina las clases de los vecinos más cercanos para cada instancia en el conjunto de datos de prueba. Posteriormente, en la fase de *Reduce*, consolidan las distancias obtenidas en cada partición para formar una tabla definitiva de vecinos.

Para adaptar SMOTE-BD al procesamiento de archivos en formato LIBSVM, se realizaron modificaciones, ya que originalmente estaba optimizado solo para bases de datos del repositorio Keel[2] (Algoritmo 4). Además, el código fue ajustado para permitir su ejecución en Apache Zeppelin como notebook

[2] `https://sci2s.ugr.es/keel/datasets.php`

interactivo.

Algoritmo 4: Técnica de sobremuestreo SMOTE-BD.

Resultado: Conjunto de datos con sobremuestreo.
Entrada: dirArchivo, dirNuevoArchivo, porcentajeSobremuestreo, tipoDistancia, kVecinos, dimensiones

```
// Función para calcular la distancia de Minkowski de dos vectores con cualquier coeficiente distCoef.
Función distancia(x, y, distCoef):
    // Inicializar la variable suma en 0.0.
    suma = 0.0 ;
    // Obtener el número de elementos del vector x.
    tamano = x.tamano;
    // Ejecutar el ciclo For para todos los elementos de x y y.
    para i ← 0 a tamano hacer
        // Acumular en suma la diferencia de cada elemento de x y y y elevarlo a la potencia de distCoef.
        suma+ = |x(i) − y(i)|^distCoef;
    fin
    // Elevar a la potencia de 1/distCoef el resultado de suma.
    devolver suma^(1/distCoef);
datos ← leer archivo LIBSVM en dirArchivo;
datos ← expandir vectores sparse a dense con dimensiones número de atributos;
enviar broadcast de datos a todos los nodos;
// Calcular los k vecinos más cercanos en cada partición de datos hechos en Spark.
para cada partición p en datos hacer
    calcular k-NN de p en broadcast de datos utilizando la función distancia;
fin
// Combinar los resultados de los k vecinos más cercanos en un DataFrame.
kNNGlobal ← combinar los resultados del k-NN;
enviar broadcast de kNNGlobal a todos los nodos;
// Generar ejemplos sintéticos utilizando SMOTE en cada una de las particiones de datos creadas Spark.
para cada partición p en datos hacer
    crear ejemplos sintéticos de kNNGlobal utilizando SMOTE;
fin
// Recolectar los datos sintéticos generados en cada partición en un nuevo DataFrame.
datosSinteticos ← recolectar datos con SMOTE;
// Crear un nuevo DataFrane con las instancias originales y sintéticas.
datosConSmote ← unir datos sintéticos con datos;
grabar datosSobreMuestreo en dirNuevoArchivo con formato LIBSVM;
```

El sobremuestreo de los conjuntos de datos se realizó al 100%, logrando conjuntos balanceados. A continuación, se entrenaron el DT (*Decision Tree*) y la SVM (*Support Vector Machine*), utilizando las implementaciones disponibles de la biblioteca MLlib de Spark. Para DT, se empleó la versión CART (*Classification and Regression Trees*) con la medida de impureza *GINI*, una profundidad máxima de 5 y 32 *bins*. En el caso de la SVM, la versión empleada fue SVC (*Support Vector Classifier*) con un kernel lineal, configurado con un máximo de 100 iteraciones, una tolerancia de $1e-6$ y la opción *fitIntercept* activada.

El desempeño de los clasificadores se evaluó con las métricas TPR, TNR y AUC-ROC. Para esto, utilizando el 30% del conjunto original para pruebas y el 70% restante para entrenamiento, permitiendo una evaluación no sesgada bajo condiciones controladas.

3.3.3 Infraestructura

Se configuraron dos clústeres en Spark para llevar a cabo los experimentos. El primer clúster se estableció en Microsoft Azure aprovechando una licencia académica, utilizando un nodo maestro y ejecutor de tipo *standard_e2as_v4* (2 vCPU, 16 GB RAM) y dos nodos ejecutores de tipo *standard_d2ds_v4* (2 vCPU, 8 GB RAM cada uno). Todos los servidores emplearon Linux Debian 10.8, Java 11, Scala 2.12 y Spark 3.1.1. Para facilitar el desarrollo y la ejecución de scripts, se instaló Zeppelin 0.9.0 en el nodo maestro, lo que permitió el uso de cuadernos interactivos (*notebooks*). Esta

infraestructura fue destinada para el análisis de SMOTE aplicado a conjuntos de datos de baja densidad.

Las restricciones de la licencia académica en Azure, que incluían un límite de 6 vCPU y un máximo de dos instancias del mismo tipo de servidor (por ejemplo, más de dos instancias *standard_d2ds_v4*), motivaron la implementación de un segundo clúster en Google Cloud, con un nodo maestro y cuatro nodos ejecutores *e2-highmem-4*, cada uno equipado con 4 vCPU y 32 GB de RAM. Este clúster utilizó Debian 11, Java 11, Scala 2.12 y Spark 3.1.2. De manera similar, se instaló Zeppelin 0.10.0 en el nodo maestro para la gestión de cuadernos interactivos, y fue empleado para evaluar el rendimiento de SMOTE en conjuntos de datos de alta dimensionalidad con baja y alta densidad.

3.4 Clases No Balanceadas con Alta Dimensionalidad y Solapamiento

La implementación de técnicas de preprocesamiento adecuadas es fundamental para abordar las complejidades inherentes a los conjuntos de datos, mitigar sus impactos negativos y desarrollar modelos de clasificación robustos y precisos. Esta parte de la tesis se enfoca en explicar cómo se abordaron las tres complejidades en su conjunto: (i) alta dimensionalidad, (ii) clases no balanceadas y (iv) solapamiento entre clases.

En la Sección 3.3, se describieron los requerimientos necesarios para el desarrollo de un análisis de los desafíos que plantean las clases no balanceadas

en presencia de alta dimensionalidad en big data. En la presente sección, se agrega al método propuesto una etapa más, en la cual se aborda de manera específica el solapamiento entre clases, generando un método integral diseñado para mitigar los efectos adversos de la dimensionalidad de los datos, las clases solapadas y no balanceadas.

El solapamiento de los datos, representa un reto significativo, especialmente en conjuntos de datos no balanceados (García et al., 2020), donde, los efectos negativos de este fenómeno se intensifican con el incremento del volumen de datos, ya que un aumento en la cantidad de datos suele conllevar una mayor probabilidad de incorporar información errónea o ruido, lo que deteriora el desempeño de los modelos de clasificación.

En esta sección, se hace un análisis de la complejidad del conjunto de datos preprocesado para determinar si el método sistemático mitiga la complejidad de solapamiento de las clases, para ello se usan las métricas $F1_{norm}$ y $F2$ descritas en la sección 1.1.4 y se hacen pruebas estadísticas para determinar si las mejoras obtenidas en las tasas de clasificación son significativas.

3.4.1 Conjuntos de Datos

Para los experimentos presentados en esta sección, se utilizó una base de datos compuesta por 21,025 instancias, 24,832 características y 17 clases, etiquetadas numéricamente del 0 al 16. El conjunto de datos original es conocido como *Indian Pines* del Grupo de Inteligencia Computacional (GIC)[3].

[3] `https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`

Aunque conjuntos de datos con alta dimensionalidad están disponibles en diferentes repositorios, algunos tienen complejidades que están fuera del alcance de esta tesis, como un número pequeño de instancias o baja densidad de datos. Tomando como referencia el trabajo de Rendón et al. (2020) donde se utilizó una ANN para transformar los datos y encontrar nuevas características, se utilizó una MLP que fue entrenada con el conjunto de datos completo y se extrajeron los datos de la penúltima capa del conjunto de datos transformado que contó con 24,832 características.

Dado que el enfoque de este estudio se centra en la clasificación binaria, fue necesario adoptar una estrategia de uno contra todos (OVA, *One-Versus-All*) para transformar el problema de clasificación multiclase en un problema de dos clases (Galar et al., 2011; Anand et al., 1995; Lorena et al., 2008). Esta estrategia consiste en diferenciar una clase de todas las demás, es decir, se considera una clase como positiva y el resto como negativas, lo que resulta en conjuntos de datos no balanceados de dos clases.

La generación de los conjuntos de datos desbalanceados se llevó a cabo de la siguiente manera: (i) se eliminaron las instancias pertenecientes a clases que representaban menos del 1.5% del total de datos (vea Tabla 3.1), ya que su inclusión resultaría en un ratio de desbalance excesivamente alto, reduciendo el número total de instancias a 20,838 y (ii) se calculó el ratio de desbalance para las clases restantes utilizando el método OVA (Tabla 3.2).

Tabla 3.1: Número de instancias por clase y porcentaje con respecto al total de datos.

Clase	**Instancias**	**Porcentaje**
0	10,776	51.25%
1	46	0.22%
2	1,428	6.79%
3	830	3.95%
4	237	1.13%
5	483	2.30%
6	730	3.47%
7	28	0.13%
8	478	2.27%
9	20	0.10%
10	972	4.62%
11	2,455	11.68%
12	593	2.82%
13	205	0.98%
14	1,265	6.02%
15	386	1.84%
16	93	0.44%
Total:	21,025	100%

Tabla 3.2: Nuevos conjuntos de datos desbalanceados creados mediante la estrategia OVA.

Datos	**Ej. Mayoritaria**	**Ej. Minoritaria**	**Atributos**	**IR**
2 vs todos	19,410	1,428	24,832	13.59
3 vs todos	20,008	830	24,832	24.11
5 vs todos	20,355	483	24,832	42.14
6 vs todos	20,108	730	24,832	27.55
8 vs todos	20,360	478	24,832	42.59
10 vs todos	19,866	972	24,832	20.44
12 vs todos	20,245	593	24,832	34.14
14 vs todos	19,573	1,265	24,832	15.47
15 vs todos	20,452	386	24,832	52.98

3.4.2 Edición de Wilson

Una vez obtenido el conjunto de datos descrito en la sección 3.4.1, se procedió a su transformación a un espacio de disimilitud (para mayor detalle véase la sección 3.2), posteriormente fue balanceado utilizando SMOTE como se describió en la Sección 3.3.2. La siguiente etapa consistió en mitigar los efectos adversos del solapamiento utilizando la técnica ENN (ver Algoritmo 5). ENN es un método diseñado para mejorar la calidad de los conjuntos de datos antes de aplicar algoritmos de clasificación, depurando el conjunto de datos mediante la eliminación de instancias que son identificadas como ruido o que son atípicas. Este proceso se lleva a cabo evaluando cada instancia del conjunto de datos en relación con sus vecinos más cercanos. Si una instancia no coincide con la mayoría de las clases de sus vecinos más cercanos, se considera un valor atípico o ruidoso; por lo tanto, se elimina. Este enfoque no solo reduce el ruido en los datos, sino que también puede contribuir a equilibrar las clases, eliminando instancias de la clase mayoritaria. Como resultado, ENN puede mejorar el rendimiento y la capacidad de generalización de los modelos de clasificación al trabajar con conjuntos de datos más limpios y equilibrados (Wilson, 1972).

En consonancia con lo descrito en la Sección 3.3.2, se utilizó el cálculo de los cinco vecinos más cercanos, eliminando las instancias en las que tres o más de sus vecinos pertenecían a la clase opuesta. Este proceso de edición de datos se describe en el Algoritmo 5.

Algoritmo 5: Edición de Wilson (*Edited Nearest Neighbor*, ENN) para entornos Big Data.

Resultado: Conjunto de datos de entrenamiento preprocesado con edición $\mathbf{D_{E-enn}}$.

Entrada: Conjuntos de datos de entrenamiento $\mathbf{D_E}$ y número de vecinos más cercanos k.

1 Importar librería con kNN-IS;
2 *datos* ← leer el archivo con $\mathbf{D_E}$;
3 *vecinos* ← obtener los k vecinos más cercanos de *datos* utilizando *kNN-IS.setup().calculatekNeighbours()*;
`// Obtener solo los índices de las instancias y su clase junto con el índice de sus vecinos más cercanos.`
4 $idxVecinos$ ← obtener de *vecinos* un arreglo de datos reducido solo con los índices numéricos de las instancias y los índices de sus k vecinos;
5 Realizar *Broadcast* de $idxVecinos$ a todos los nodos;
6 Utilizar *Map* para marcar las instancias de *idxVecinos* en donde la mayoría de los vecinos no correspondan a la clase de la instancia;
7 D_{E-enn} ← utilizar *Join* para remover las instancias marcadas para borrar en *idxVecinos* de *datos*;
8 Guardar D_{E-enn} en archivo con formato *LIBSVM*;

3.4.3 Infraestructura

Para llevar a cabo los experimentos descritos en esta sección, se utilizaron créditos en la nube Google proporcionados a través del programa "Google para la Educación[4]". El clúster de Spark se configuró con un nodo maestro, equipado con 32 GB de memoria y 8 vCPUs, junto con tres nodos esclavos, cada uno con 64 GB de memoria y 16 vCPUs. Esta infraestructura permitió manejar de manera eficiente los requisitos computacionales necesar-

[4] `https://cloud.google.com/edu/`

ios para procesar grandes volúmenes de datos y ejecutar los experimentos de manera escalable y efectiva.

Capítulo 4

Resultados Experimentales

Este capítulo presenta los resultados experimentales obtenidos del estudio de la aplicación del método sistemático propuesto en esta tesis (para mayor detalle véase la sección 3.1), el cual se centra en tratar tres desafíos de big data: el problema del desbalance de clases, la alta dimensionalidad y el solapamiento de clases. Para la evaluación de la efectividad del método propuesto se usan las métricas de efectividad TPR, TNR, AUC-ROC y G-Mean, descritas en la sección 1.1.5 y la metodología presentada en el capítulo 3. Asimismo, se analiza la complejidad de los datos preprocesados con el método propuesto con las métricas $F1_{norm}$ y $F2$ enunciadas en la sección 1.1.4 y se comprueba la hipótesis de esta tesis por medio de una prueba estadística. Finalmente, se ofrece una interpretación de los resultados, destacando su relevancia para la mejora de los modelos de clasificación en escenarios complejos donde existe desbalance de clases, alta dimensionalidad y solapamiento de datos, y se aportan conclusiones sobre la efectividad del enfoque propuesto.

En la primera parte de este capítulo, se presentan los resultados de evaluar la efectividad de las técnicas propuestas para tratar la dimensionalidad de los datos y el desbalance de clases (sección 4.1). La segunda parte se

enfoca en mostrar los resultados de abordar no solo el desbalance de clases y la dimensionalidad de los datos, sino que incluye una tercera complejidad: el solapamiento de clases (sección 4.2). En la tercera parte se presenta un análisis estadístico no paramétrico que permite la comprobación de la hipótesis propuesta en este trabajo (sección 4.3). Finalmente, en la sección 4.4 se realiza una discusión detallada de los principales resultados experimentales presentados en este capítulo.

4.1 Clases No Balanceadas y Alta Dimensionalidad

Para evaluar el rendimiento de los clasificadores DT y SVM, se utilizaron diversas métricas de efectividad, aplicadas tanto al conjunto original como a los conjuntos balanceados. Estos últimos se analizaron en escenarios de baja y alta densidad. Los conjuntos de datos de baja densidad se refieren a aquellos que cuentan con la mayor parte de sus valores en ceros y los de alta densidad contienen un mínimo de ceros en sus valores.

Los resultados mostraron que, en los conjuntos sin preprocesamiento, ambos clasificadores obtuvieron una TPR cercana a 0 y una TNR igual a 1. Además, la métrica AUC-ROC se mantuvo en torno a 0.5, lo que indica un desempeño equivalente al de un clasificador aleatorio y una incapacidad para identificar la clase minoritaria.

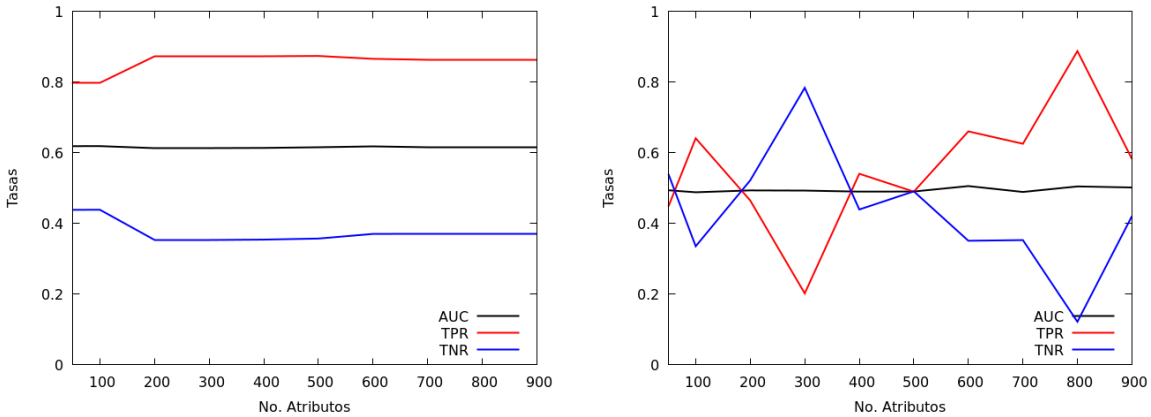
Las Figuras 4.1 y 4.2 presentan los resultados obtenidos tras aplicar las técnicas de balanceo ROS y SMOTE. En los conjuntos de baja densidad, ROS permitió que los clasificadores alcanzaran una TPR de al menos 0.8, indepen-

dientemente del número de atributos. De manera notable, SMOTE-BD logró un comportamiento similar a partir de los 400 atributos, lo que desafía estudios previos que sugieren que la alta dimensionalidad afecta negativamente la eficacia de métodos basados en distancia euclidiana para abordar el desbalance de clases (Elreedy y Atiya, 2019).

En contraste, en los conjuntos de alta densidad, los resultados fueron consistentes con la literatura: la efectividad de las técnicas basadas en distancia euclidiana disminuyó a medida que aumentaba la dimensionalidad. Este patrón también se observó en SVM. Sin embargo, es relevante señalar que, si bien la efectividad de SMOTE-BD se redujo en los conjuntos de baja densidad, la disminución fue menos pronunciada en comparación con los conjuntos de alta densidad.

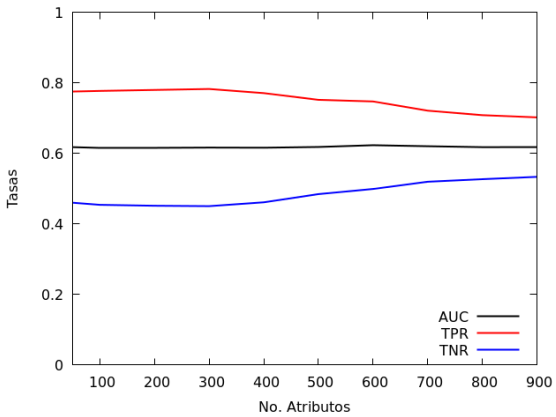
Para evaluar el impacto de las distancias fraccionarias en SMOTE-BD, se realizaron experimentos con distintos valores de $p = 2.00, 0.75, 0.66, 0.50, 0.33$ y 0.25 Los resultados de la TPR utilizando el clasificador DT, en escenarios de baja y alta densidad, se presentan en la Figura 4.3.

En los conjuntos de baja densidad, el uso de distancias fraccionarias no mostró diferencias significativas en comparación con la distancia euclidiana. Sin embargo, en los conjuntos de alta densidad, estas distancias mejoraron notablemente la clasificación de la clase minoritaria, destacando la distancia con $p = 0.33$ como la más efectiva. Este incremento en la TPR, al favorecer la identificación de la clase minoritaria, conllevó una reducción en la tasa de reconocimiento de la clase mayoritaria, como se observa en la Figura 4.4. Como

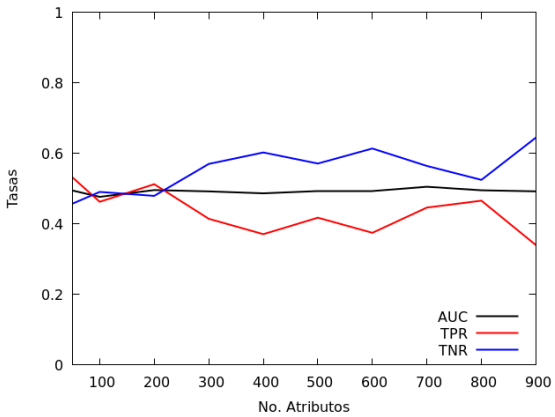


(a) ROS Baja Densidad con DT

(b) ROS Alta Densidad con DT



(c) ROS Baja Densidad con SVM

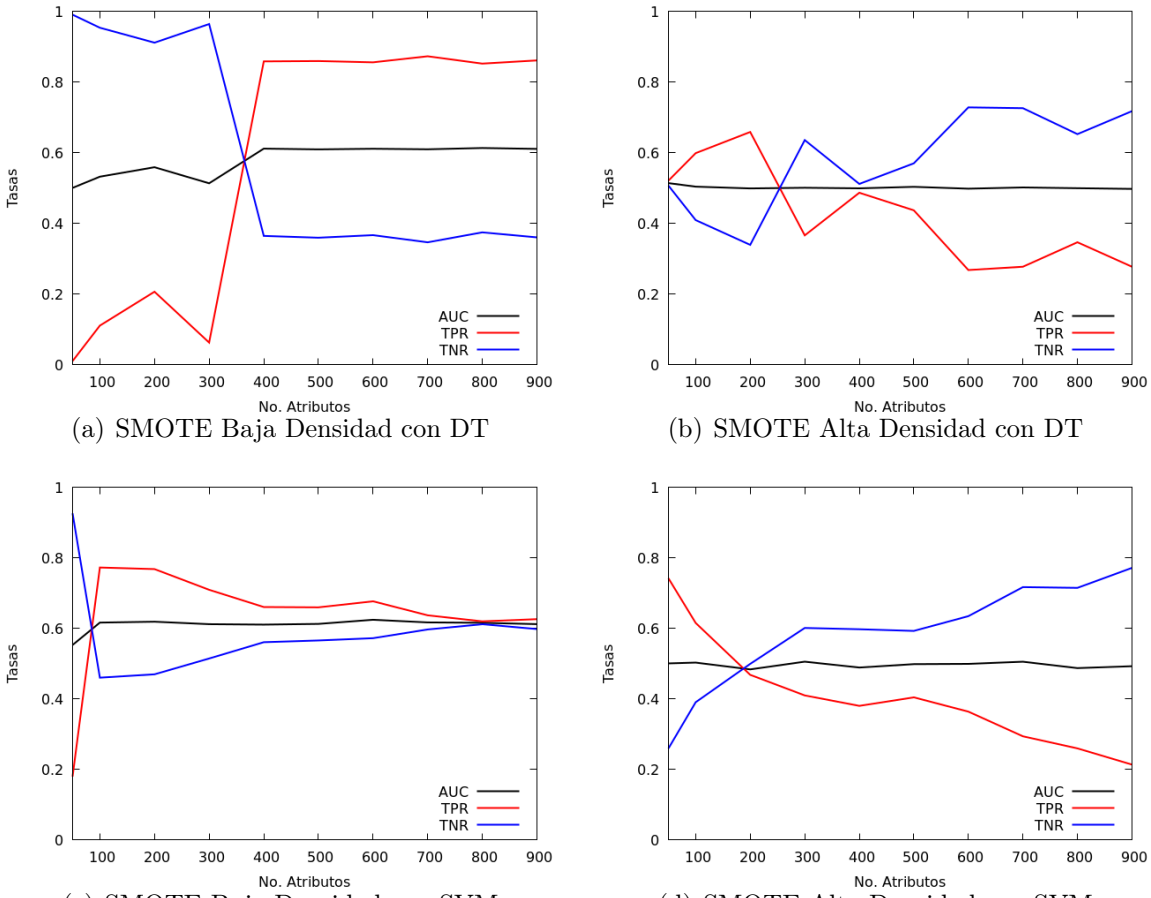


(d) ROS Alta Densidad con SVM

Figura 4.1: Resultados de clasificación en términos de TPR, TNR y AUC-ROC para DT en datos balanceados de baja y alta densidad utilizando ROS.

resultado, el valor de AUC-ROC disminuyó, lo que indica que el balance global del modelo no mejoró sustancialmente, como se muestra en la Figura 4.5.

Al aplicar el clasificador SVM, se observaron resultados similares, como se detalla en las Figuras 4.6, 4.7 y 4.8.



(a) SMOTE Baja Densidad con DT

(b) SMOTE Alta Densidad con DT

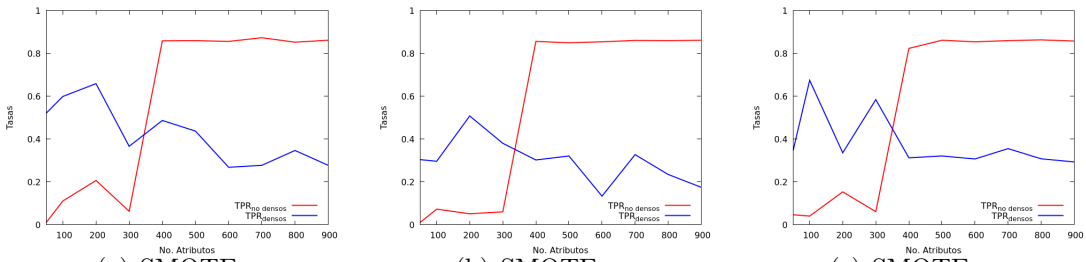
(c) SMOTE Baja Densidad con SVM

(d) SMOTE Alta Densidad con SVM

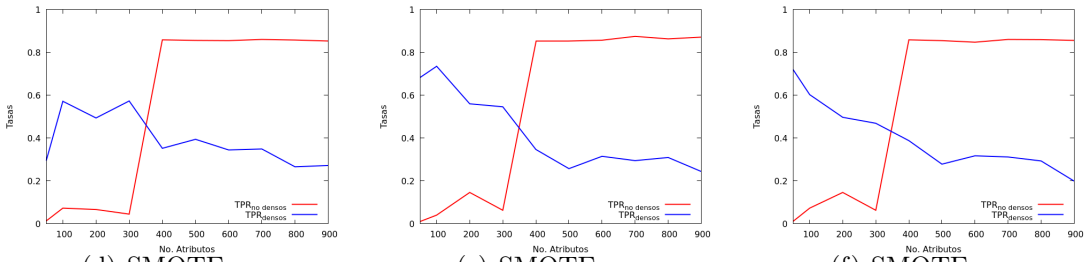
Figura 4.2: Resultados de clasificación en términos de TPR, TNR y AUC-ROC para DT en datos balanceados de baja y alta densidad utilizando SMOTE-BD.

Es importante destacar que el objetivo del sobremuestreo en problemas de clasificación binaria es equilibrar la correcta clasificación de ambas clases. Aunque los experimentos con la base de datos KDD 2010 evidenciaron un aumento en la TPR debido al sobremuestreo, este efecto se logró a expensas de una reducción considerable en la TNR, alcanzando valores inferiores a 0.5,

como se observa en las Figuras 4.4 y 4.7. Esta relación inversa entre TPR y TNR, con un AUC-ROC en el rango de 0.5 a 0.6, sugiere un comportamiento similar al de un clasificador aleatorio. Este fenómeno podría deberse a características intrínsecas de la base de datos utilizada, más que a las técnicas de sobremuestreo en sí, como se desprende de los resultados obtenidos con ROS.

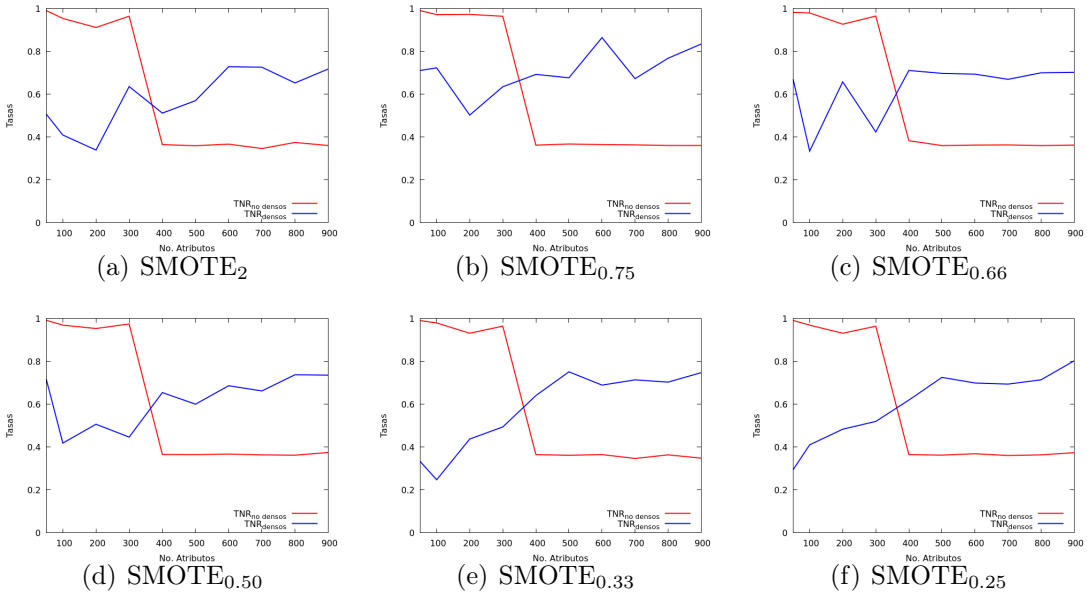


(a) SMOTE_{2} (b) $\text{SMOTE}_{0.75}$ (c) $\text{SMOTE}_{0.66}$



(d) $\text{SMOTE}_{0.50}$ (e) $\text{SMOTE}_{0.33}$ (f) $\text{SMOTE}_{0.25}$

Figura 4.3: Resultados de TPR para DT en datos balanceados de baja y alta densidad usando SMOTE-BD con distancias fraccionarias.



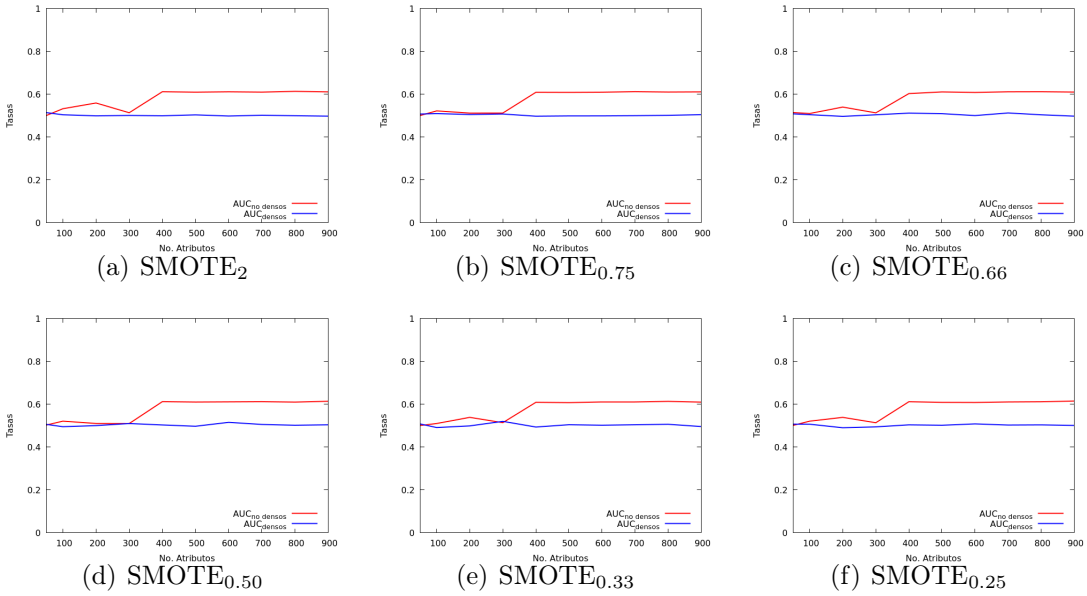
(a) $SMOTE_2$ (b) $SMOTE_{0.75}$ (c) $SMOTE_{0.66}$

(d) $SMOTE_{0.50}$ (e) $SMOTE_{0.33}$ (f) $SMOTE_{0.25}$

Figura 4.4: Resultados de TNR para DT en datos balanceados de baja y alta densidad usando SMOTE-BD con distancias fraccionarias.

El análisis de los resultados obtenidos en los conjuntos de datos, tanto en su versión original desbalanceada como tras la aplicación de PCA para generar conjuntos de alta densidad, revela patrones distintivos en las métricas de clasificación. De manera inesperada, los datos de baja densidad mostraron un incremento en la TPR cuando la dimensionalidad superaba los 400 atributos, tras la aplicación de SMOTE-BD. Este hallazgo sugiere que SMOTE-BD posee una capacidad de adaptación particular en escenarios de alta dimensionalidad dentro de conjuntos de baja densidad.

En contraste, en los conjuntos de alta densidad, la TPR disminuyó a medida que aumentaba el número de dimensiones, en concordancia con la



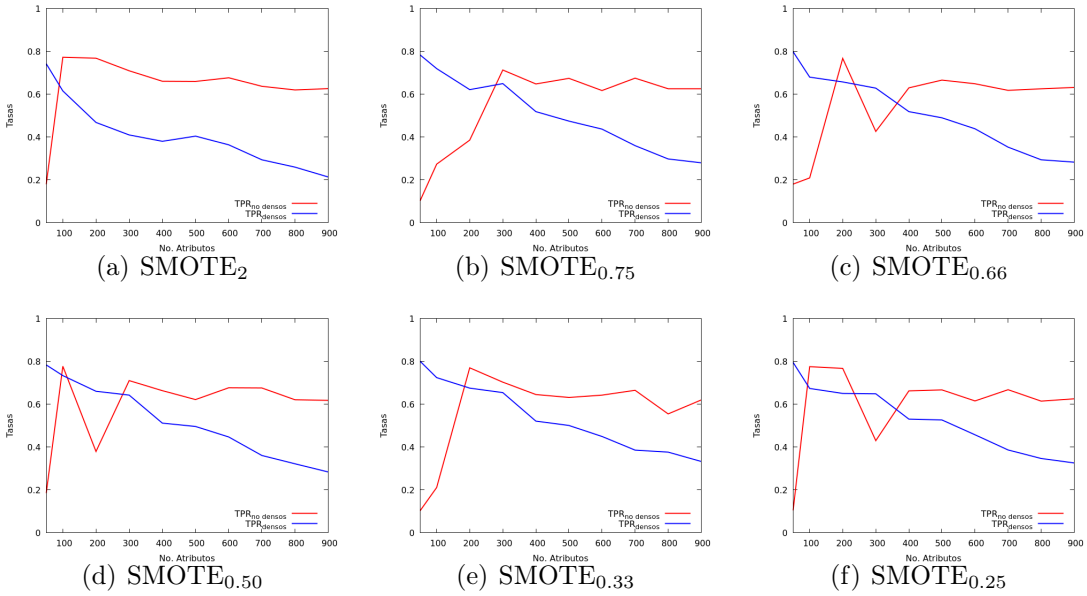
(a) SMOTE_{2} (b) $\text{SMOTE}_{0.75}$ (c) $\text{SMOTE}_{0.66}$

(d) $\text{SMOTE}_{0.50}$ (e) $\text{SMOTE}_{0.33}$ (f) $\text{SMOTE}_{0.25}$

Figura 4.5: Resultados de AUC-ROC para DT en datos balanceados de baja y alta densidad usando SMOTE-BD con distancias fraccionarias.

literatura, que señala una reducción en la efectividad de los métodos de clasificación en espacios de alta dimensionalidad. Sin embargo, el uso de distancias fraccionarias demostró ser una estrategia efectiva para mitigar estos efectos adversos, superando el desempeño de la distancia euclidiana tradicional.

Un aspecto crítico identificado fue la reducción en la tasa de clasificación de la clase mayoritaria al mejorar la TPR, un efecto observado tanto en las aplicaciones de SMOTE-BD con distancias fraccionarias como en el uso de ROS. Esto sugiere que, más allá de los desafíos propios de la alta dimensionalidad y el desbalance de clases, la base de datos KDD 2010 podría presentar problemáticas subyacentes que afectan la eficacia de la clasificación.



(a) $SMOTE_2$ (b) $SMOTE_{0.75}$ (c) $SMOTE_{0.66}$

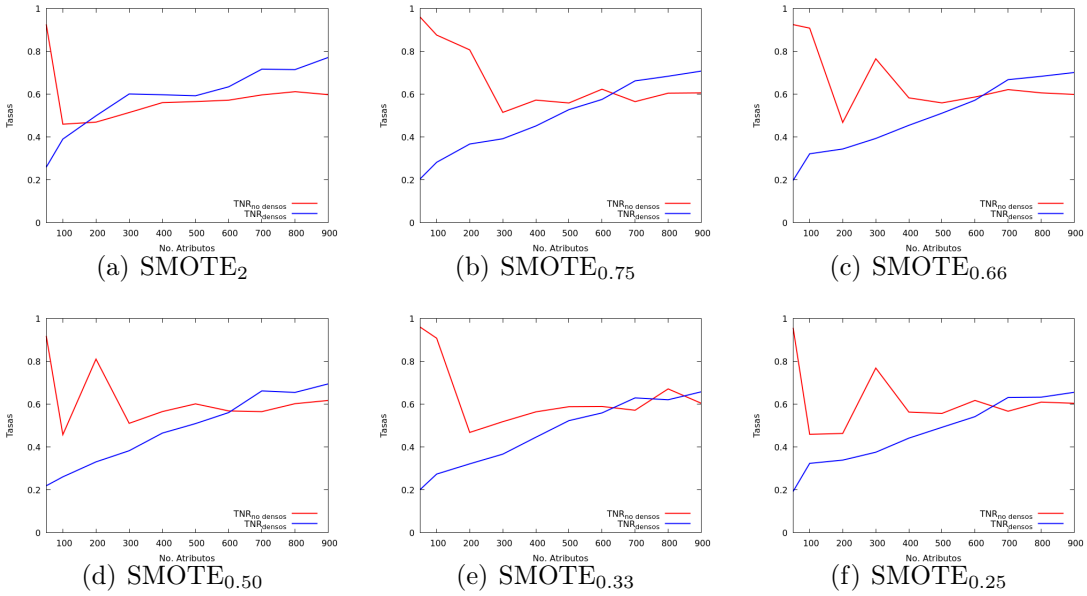
(d) $SMOTE_{0.50}$ (e) $SMOTE_{0.33}$ (f) $SMOTE_{0.25}$

Figura 4.6: Resultados de TPR para SVM en datos balanceados de baja y alta densidad usando SMOTE-BD con distancias fraccionarias.

Por lo tanto, es fundamental abordar estos factores adicionales para lograr un equilibrio adecuado en el rendimiento del clasificador para ambas clases.

4.2 Clases No Balanceadas con Alta Dimensionalidad y Solapamiento

Para evaluar la efectividad de la estrategia propuesta, se empleó un árbol de decisión (DT) con los parámetros especificados en la Sección 3.3.2. Los conjuntos de datos OVA, descritos en la Tabla 3.2, se utilizaron tanto en su versión original (sin preprocesamiento) como en su versión balanceada para construir y evaluar el DT. Los resultados, presentados en la Tabla 4.1,



(a) SMOTE_{2} (b) $\text{SMOTE}_{0.75}$ (c) $\text{SMOTE}_{0.66}$

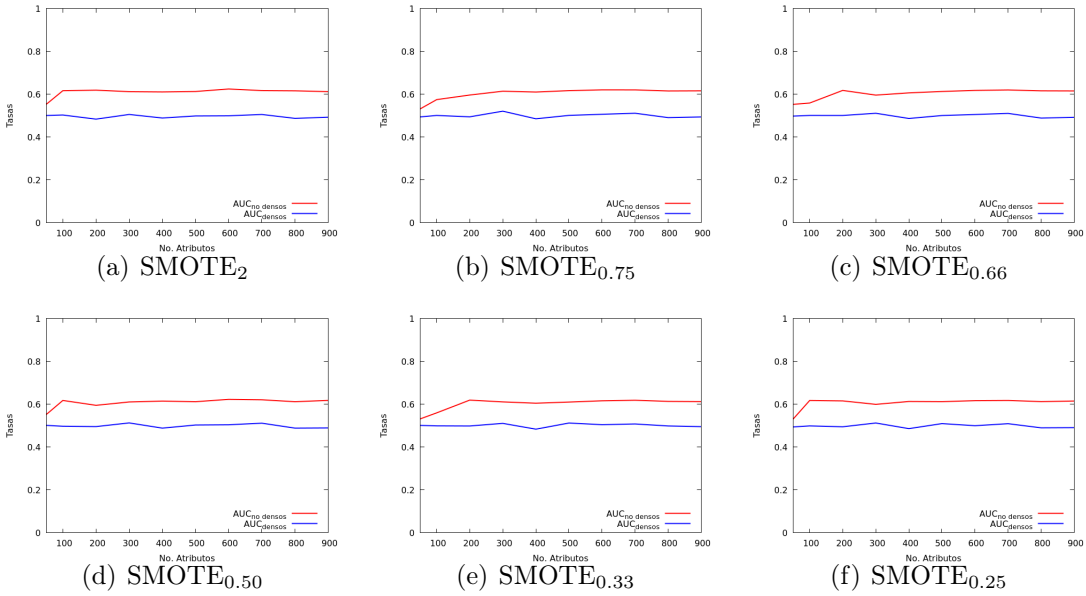
(d) $\text{SMOTE}_{0.50}$ (e) $\text{SMOTE}_{0.33}$ (f) $\text{SMOTE}_{0.25}$

Figura 4.7: Resultados de TNR para SVM en datos balanceados de baja y alta densidad usando SMOTE-BD con distancias fraccionarias.

corresponden a un sobremuestreo del 100% con un parámetro $k = 5$. Se observó que el uso de SMOTE de la librería *imbalanced-learn*[1] (en adelante, SMOTE_{il}) mejora las métricas de clasificación en comparación con los datos sin preprocesar, estableciendo así una referencia para la comparación con el método sistemático propuesto.

Posteriormente, se evaluaron los conjuntos de datos preprocesados mediante dos enfoques: Disimilitud + SMOTE (Figuras 4.9, 4.10 y 4.11) y Disimilitud + SMOTE + ENN (Figuras 4.12, 4.13 y 4.14). En estas figuras, los resultados obtenidos (eje Y) con el método propuesto (líneas sólidas) se

[1] `https://imbalanced-learn.org/`



(a) SMOTE_{2} (b) $\text{SMOTE}_{0.75}$ (c) $\text{SMOTE}_{0.66}$

(d) $\text{SMOTE}_{0.50}$ (e) $\text{SMOTE}_{0.33}$ (f) $\text{SMOTE}_{0.25}$

Figura 4.8: Resultados de AUC-ROC para SVM en datos balanceados de baja y alta densidad usando SMOTE-BD con distancias fraccionarias.

comparan con los datos sin preprocesar (línea punteada negra) y con SMOTE estándar basado en distancia euclidiana (línea punteada roja) para diferentes niveles de dimensionalidad (eje X).

En general, los clasificadores entrenados con los datos preprocesados mostraron un mejor desempeño. Sin embargo, las diferencias entre las distintas distancias fraccionarias varían según el conjunto de datos, como se ha reportado previamente (Aggarwal et al., 2001). En algunos casos, la variación en AUC-ROC entre las distancias fraccionarias es mínima, como se observa en las Figuras 4.9(a)(f)(h) y 4.12. No obstante, estas diferencias son más evidentes al analizar la TPR, como en las Figuras 4.10(a)(f)(h) y 4.13. Final-

Tabla 4.1: Resultados de clasificación de los conjuntos de datos sin preprocesamiento y con SMOTE$_{il}$.

Conjunto de Datos	Preprocesamiento	TPR	TNR	AUC-ROC	G-Mean
2 vs todos	Ninguno	0.51402	**0.97645**	0.74524	0.70846
	SMOTE$_{il}$	**0.85047**	0.86215	**0.85631**	**0.85629**
3 vs todos	Ninguno	0.45977	**0.98814**	0.72395	0.67403
	SMOTE$_{il}$	**0.75096**	0.93049	**0.84073**	**0.83592**
5 vs todos	Ninguno	0.61268	**0.99672**	0.80470	0.78145
	SMOTE$_{il}$	**0.80282**	0.95937	**0.88109**	**0.87761**
6 vs todos	Ninguno	0.57078	**0.99121**	0.78099	0.75217
	SMOTE$_{il}$	**0.86758**	0.96665	**0.91711**	**0.91578**
8 vs todos	Ninguno	0.92373	**0.99461**	0.95917	0.95852
	SMOTE$_{il}$	**0.94915**	0.99249	**0.97082**	**0.97058**
10 vs todos	Ninguno	0.59375	**0.98456**	0.78915	0.76458
	SMOTE$_{il}$	**0.86111**	0.91658	**0.88885**	**0.88841**
12 vs todos	Ninguno	0.44767	**0.99177**	0.71972	0.66633
	SMOTE$_{il}$	**0.78488**	0.91916	**0.85202**	**0.84937**
14 vs todos	Ninguno	0.54595	**0.96886**	0.75740	0.72728
	SMOTE$_{il}$	**0.85676**	0.91797	**0.88736**	**0.88684**
15 vs todos	Ninguno	0.06838	**0.99445**	0.53141	0.26076
	SMOTE$_{il}$	**0.67521**	0.84989	**0.76255**	**0.75754**

mente, los tres mejores resultados en términos de AUC-ROC se presentan en la Tabla 4.2.

Inicialmente, la combinación Disimilitud + SMOTE mostró un ligero aumento en AUC-ROC en comparación con SMOTE en la mayoría de los casos, con la excepción del escenario "2 vs todos" (véase Figura 4.9 y Tabla 4.2). Al examinar los valores de TPR en la Figura 4.10, se observa una mejora respecto a SMOTE$_{il}$, especialmente en el conjunto de datos "15 vs todos" (Figura 4.10(i)). En este caso, la variabilidad de los resultados entre las distintas distancias fraccionarias es más pronunciada, y en la mayoría de los casos, estas distancias superan el desempeño de la distancia euclidiana.

Un caso particular se presenta en la Figura 4.10(e) para el conjunto de datos "8 vs todos". A pesar de tener un IR de 42.59, los datos sin preprocesar alcanzan una TPR superior a 0.9, un fenómeno no observado en ningún otro

conjunto de datos. Esto sugiere que, a pesar del desbalance, las clases en este conjunto podrían ser intrínsecamente separables, lo que reduciría el impacto del IR en la clasificación.

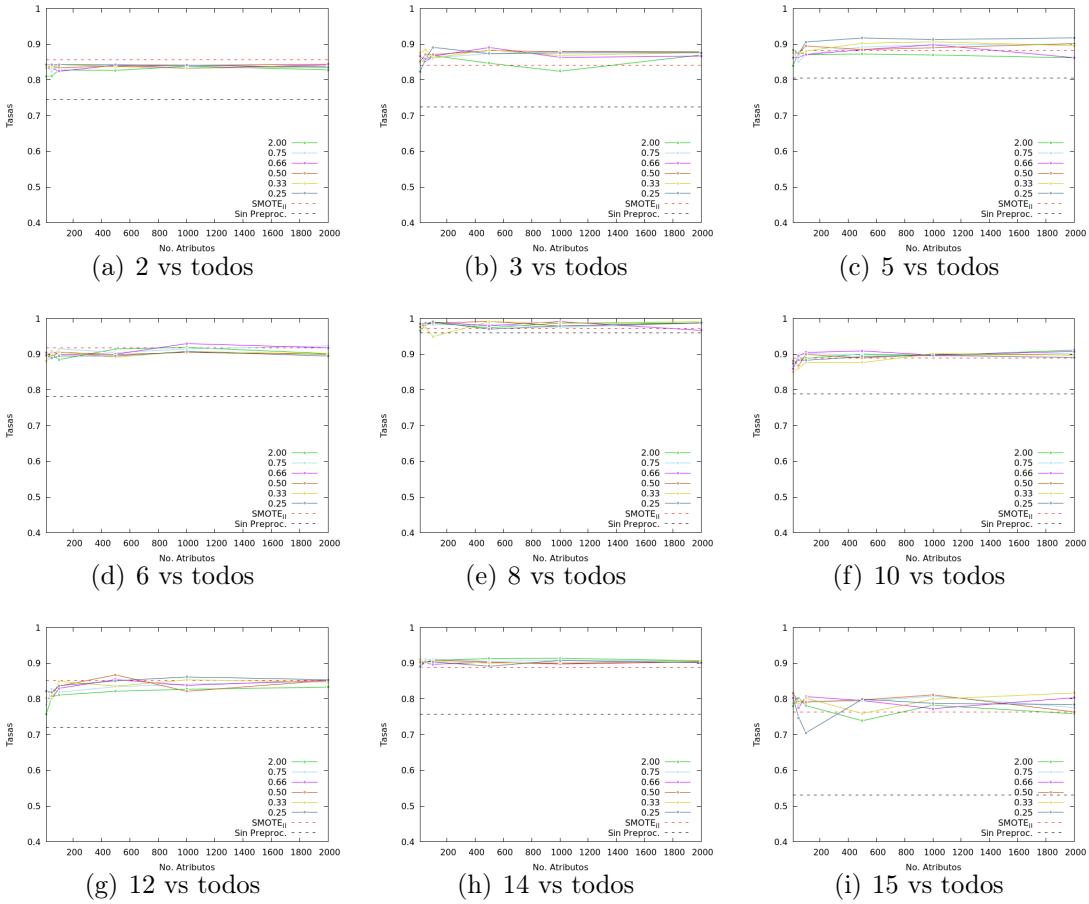
En términos generales, las distancias fraccionarias demostraron un rendimiento superior y más consistente en AUC-ROC, lo que refuerza su capacidad para mitigar los efectos de la maldición de la dimensionalidad. De manera destacada, la mayoría de los mejores resultados se lograron con 500 o menos atributos, lo que representa solo el 2.01% de los 24,832 atributos originales. Esto no solo mejora la eficiencia del modelo, sino que también reduce significativamente los tiempos de ejecución, aunque un análisis detallado de este aspecto queda fuera del alcance del presente estudio.

Finalmente, se analizaron los resultados del método sistemático completo, que incorpora ENN a los datos preprocesados con Disimilitud + SMOTE. Las Figuras 4.12, 4.13 y 4.14 ilustran estos resultados en comparación con los datos sin preprocesar, mientras que los mejores valores de AUC-ROC se presentan en la Tabla 4.3. Este método sistemático mejoró significativamente el desempeño de clasificación en la mayoría de los casos, mostrando una eficacia particularmente alta en el conjunto de datos "15 vs todos" (Figuras 4.12(i) y 4.13(i), correspondientes a AUC-ROC y TPR, respectivamente).

Al comparar los valores de TPR entre Disimilitud + SMOTE (Figura 4.10) y Disimilitud + SMOTE + ENN (Figura 4.13), se observa que la inclusión de ENN generalmente estabiliza los resultados, con un efecto particularmente notable en las Figuras 4.10(b), 4.10(g) y 4.10(i), así como en las Fig-

Tabla 4.2: Mejores Resultados de Disimilitud + SMOTE.

Conjunto de Datos	Distancia	Atributos	TPR	TNR	AUC-ROC	G-Mean
2 vs todos	0.50	2000	0.89252	0.79477	**0.84365**	0.84223
	0.75	100	0.87383	**0.81213**	0.84298	**0.84242**
	0.33	100	**0.92523**	0.76057	0.84290	0.83887
3 vs todos	0.25	100	0.93870	**0.84378**	**0.89124**	**0.88997**
	0.66	500	**0.96169**	0.81988	0.89078	0.88796
	0.33	50	0.94636	0.82172	0.88404	0.88184
5 vs todos	0.25	2000	**0.90141**	0.93381	**0.91761**	**0.91747**
	0.25	500	0.88732	**0.94725**	0.91729	0.91680
	0.25	1000	0.88028	0.94545	0.91286	0.91228
6 vs todos	0.66	1000	**0.96804**	0.89132	**0.92968**	**0.92889**
	0.75	2000	0.95434	0.89182	0.92308	0.92255
	2.00	1000	0.94064	**0.89730**	0.91897	0.91871
8 vs todos	0.50	500	**0.99153**	**0.99184**	**0.99168**	**0.99168**
	0.66	1000	**0.99153**	0.99119	0.99136	0.99136
	0.33	500	**0.99153**	0.99054	0.99103	0.99103
10 vs todos	2.00	2000	**0.93056**	0.89225	**0.91140**	**0.91120**
	0.66	500	0.91319	0.90500	0.90910	0.90909
	0.66	2000	0.90972	**0.90567**	0.90770	0.90770
12 vs todos	0.50	500	**0.96512**	0.76934	**0.86723**	**0.86169**
	0.25	1000	0.91279	**0.81017**	0.86148	0.85995
	0.66	500	0.92442	0.78367	0.85404	0.85114
14 vs todos	2.00	1000	0.96486	0.86266	**0.91376**	**0.91233**
	2.00	500	0.95946	**0.86624**	0.91285	0.91166
	0.75	100	**0.98108**	0.84139	0.91123	0.90855
15 vs todos	0.33	2000	0.91453	**0.71888**	**0.81670**	**0.81082**
	0.50	10	**0.94872**	0.68364	0.81618	0.80534
	0.66	10	0.92308	0.70224	0.81266	0.80512



(a) 2 vs todos (b) 3 vs todos (c) 5 vs todos

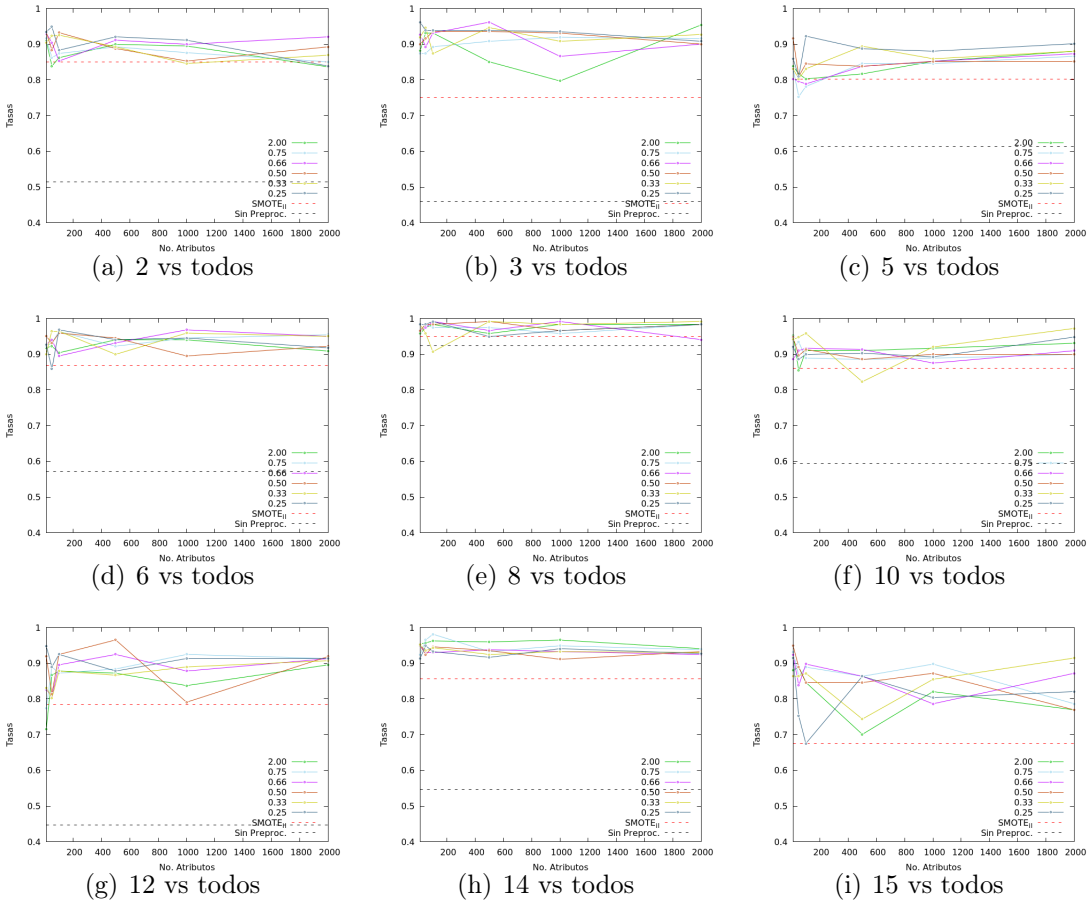
(d) 6 vs todos (e) 8 vs todos (f) 10 vs todos

(g) 12 vs todos (h) 14 vs todos (i) 15 vs todos

Figura 4.9: Resultados de AUC-ROC de los diferentes conjuntos de datos OVA sin preprocesamiento, con SMOTE y Disimilitud + SMOTE.

uras 4.13(b), 4.13(g) y 4.13(i). No obstante, la mejor TPR para "8 vs todos" en la Tabla 4.3 alcanza un valor de 1.0, lo que podría indicar un sobreajuste del modelo, especialmente considerando que este conjunto ya muestra buenos resultados de clasificación sin preprocesamiento.

La comparación de los mejores resultados entre ambos métodos revela



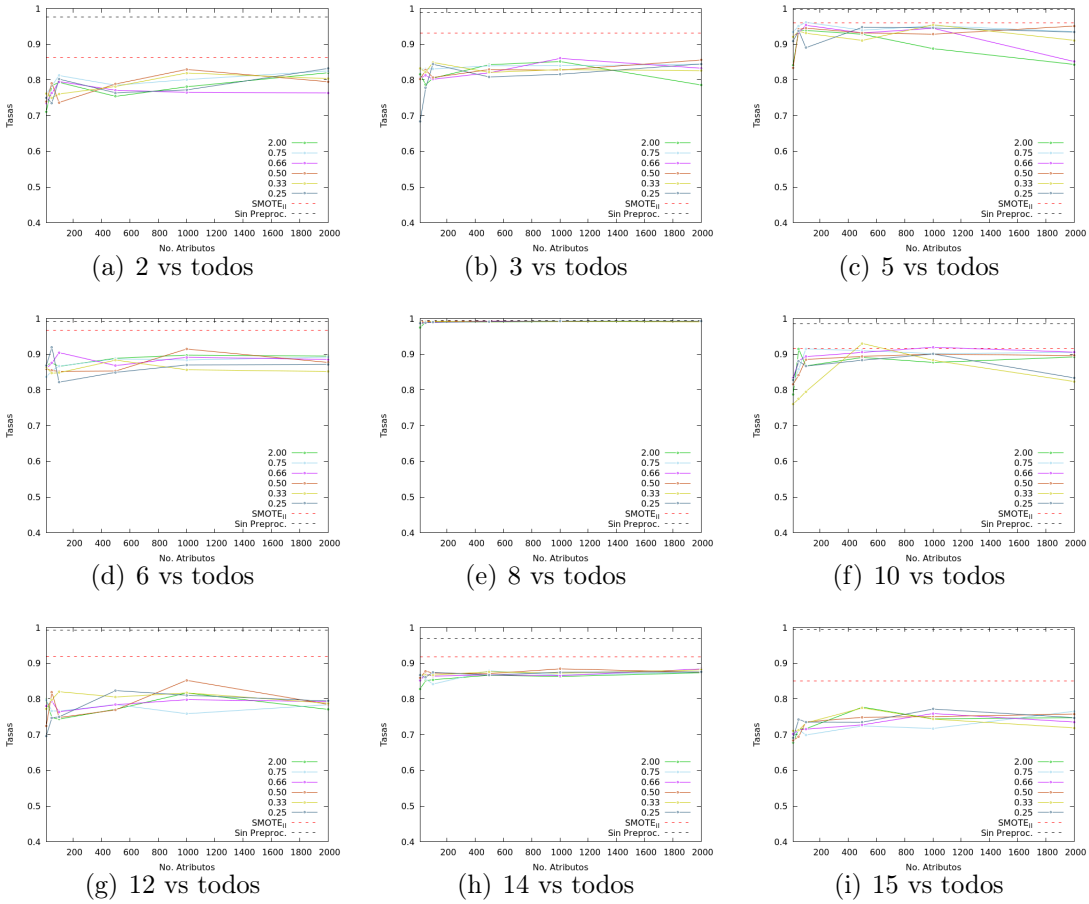
(a) 2 vs todos (b) 3 vs todos (c) 5 vs todos

(d) 6 vs todos (e) 8 vs todos (f) 10 vs todos

(g) 12 vs todos (h) 14 vs todos (i) 15 vs todos

Figura 4.10: Resultados de TPR de los diferentes conjuntos de datos OVA sin preprocesamiento, con SMOTE y Disimilitud + SMOTE.

que Disimilitud + SMOTE + ENN obtiene un desempeño superior o muy similar al de Disimilitud + SMOTE, pero con una cantidad menor de atributos. Por ejemplo, en el conjunto "15 vs todos", el mejor resultado con Disimilitud + SMOTE se obtuvo con 2000 atributos, alcanzando una TPR de 0.91453 y un AUC-ROC de 0.81670. En cambio, Disimilitud + SMOTE + ENN logró



(a) 2 vs todos (b) 3 vs todos (c) 5 vs todos

(d) 6 vs todos (e) 8 vs todos (f) 10 vs todos

(g) 12 vs todos (h) 14 vs todos (i) 15 vs todos

Figura 4.11: Resultados de TNR de los diferentes conjuntos de datos OVA sin preprocesamiento, con SMOTE y Disimilitud + SMOTE.

su mejor desempeño con solo 50 atributos, obteniendo una TPR de 0.94872 y un AUC-ROC de 0.81944. Aunque la mejora en AUC-ROC es modesta, el incremento en TPR es considerable y se logra con una reducción significativa en la cantidad de atributos.

En dos conjuntos de datos, “5 vs todos” y “6 vs todos”, Disimilitud +

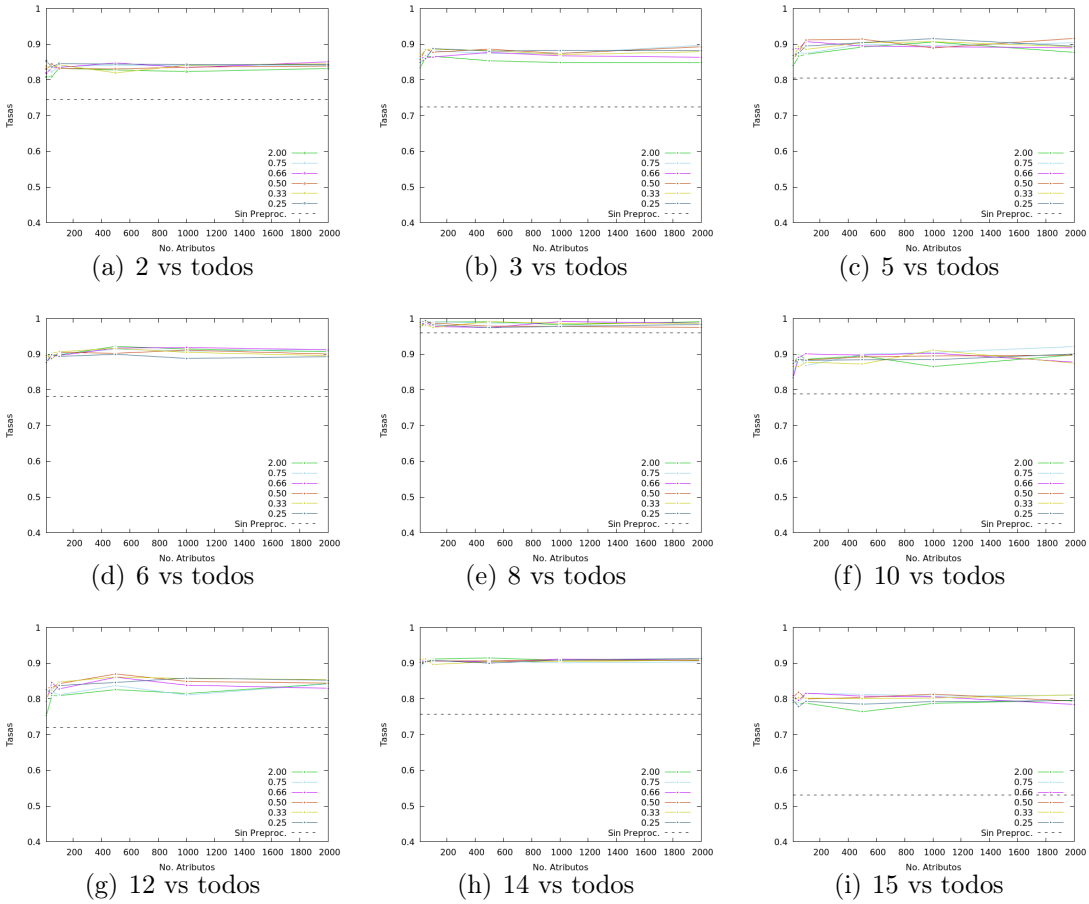
Tabla 4.3: Mejores Resultados de Disimilitud + SMOTE + ENN.

Conjunto de Datos	Distancia	Atributos	TPR	TNR	AUC-ROC	G-Mean
2 vs todos	0.25	10	**0.95327**	0.75180	**0.85254**	0.84657
	0.66	2000	0.92056	0.78034	0.85045	**0.84755**
	0.66	500	0.90654	**0.78824**	0.84739	0.84533
3 vs todos	0.75	2000	**0.96169**	**0.83208**	**0.89688**	**0.89454**
	0.50	2000	**0.96169**	0.82389	0.89279	0.89013
	0.25	100	**0.96169**	0.81370	0.88769	0.88460
5 vs todos	0.50	2000	0.90141	**0.93087**	**0.91614**	**0.91602**
	0.25	1000	**0.90845**	0.92267	0.91556	0.91553
	0.50	500	0.90141	0.92628	0.91384	0.91376
6 vs todos	2.00	500	0.95890	**0.88452**	**0.92171**	**0.92096**
	0.66	1000	**0.96804**	0.86926	0.91865	0.91732
	0.66	500	0.95890	0.87523	0.91707	0.91611
8 vs todos	0.25	50	**1.00000**	0.98760	**0.99380**	**0.99378**
	0.66	1000	0.99153	**0.99102**	0.99128	0.99128
	0.33	500	0.99153	0.99086	0.99119	0.99119
10 vs todos	0.75	2000	**0.95486**	**0.88771**	**0.92129**	**0.92068**
	0.33	1000	0.94444	0.87798	0.91121	0.91061
	0.75	1000	0.92361	0.88604	0.90482	0.90463
12 vs todos	0.50	500	**0.97093**	0.76869	**0.86981**	**0.86391**
	0.33	500	0.93023	**0.79206**	0.86115	0.85837
	0.66	500	0.93023	0.79108	0.86065	0.85784
14 vs todos	2.00	500	**0.97838**	0.85058	**0.91448**	**0.91224**
	0.25	2000	0.97027	0.85636	0.91332	0.91154
	0.33	50	0.95946	**0.86453**	0.91200	0.91076
15 vs todos	0.50	50	**0.94872**	0.69016	**0.81944**	0.80918
	0.66	100	0.92308	**0.70941**	0.81625	**0.80922**
	0.75	100	0.92308	0.70729	0.81519	0.80801

SMOTE obtuvo un AUC-ROC superior al de Disimilitud + SMOTE + ENN. Sin embargo, la diferencia fue mínima, y en ambos casos el método sistemático logró una mejor TPR con un menor número de atributos. En general, Disimilitud + SMOTE + ENN mostró los mejores resultados de clasificación y una mayor consistencia. Además, su implementación en Spark lo hace un método escalable y adecuado para entornos de big data.

Métricas de Complejidad

En esta sección se analiza la complejidad de los conjuntos de datos mediante las métricas $F1_{norm}$ y $F2$, con el objetivo de evaluar si el método



(a) 2 vs todos (b) 3 vs todos (c) 5 vs todos

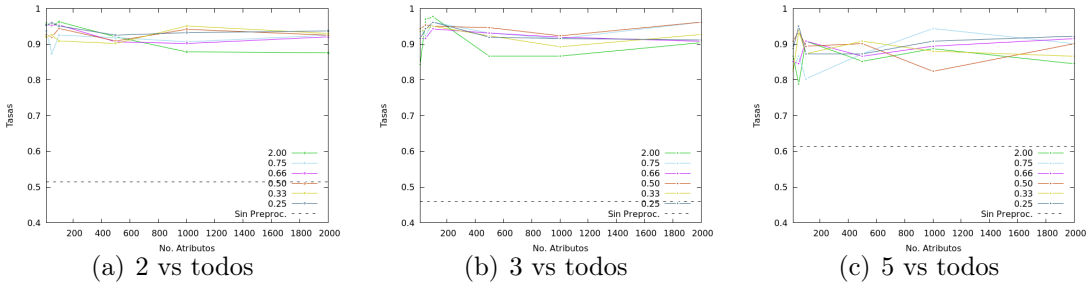
(d) 6 vs todos (e) 8 vs todos (f) 10 vs todos

(g) 12 vs todos (h) 14 vs todos (i) 15 vs todos

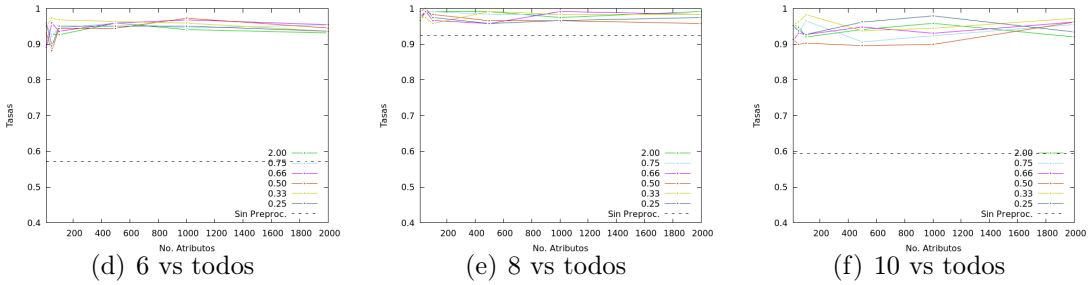
Figura 4.12: Resultados de AUC-ROC de los diferentes conjuntos de datos OVA sin preprocesamiento y Disimilitud + SMOTE + ENN.

sistemático propuesto contribuye a reducir el solapamiento entre clases.

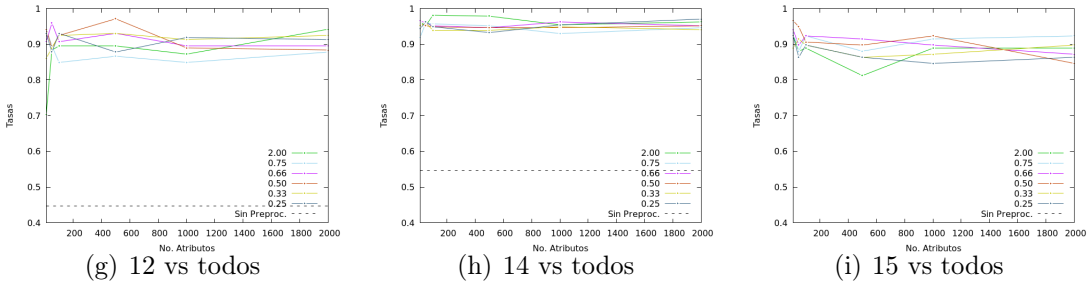
A partir de los mejores resultados presentados en la Tabla 4.3, se calcularon las métricas de complejidad en cada una de las etapas del método sistemático: (i) Disimilitud, (ii) Disimilitud + SMOTE y (iii) Disimilitud + SMOTE + ENN. Además, se obtuvieron los valores de $F1_{norm}$ y $F2$ para el



(a) 2 vs todos (b) 3 vs todos (c) 5 vs todos



(d) 6 vs todos (e) 8 vs todos (f) 10 vs todos

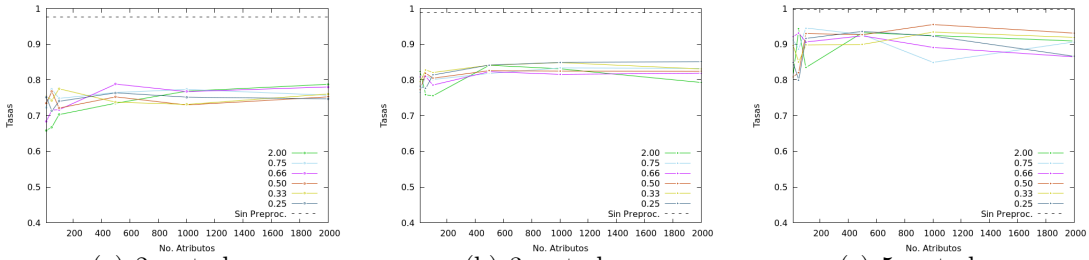


(g) 12 vs todos (h) 14 vs todos (i) 15 vs todos

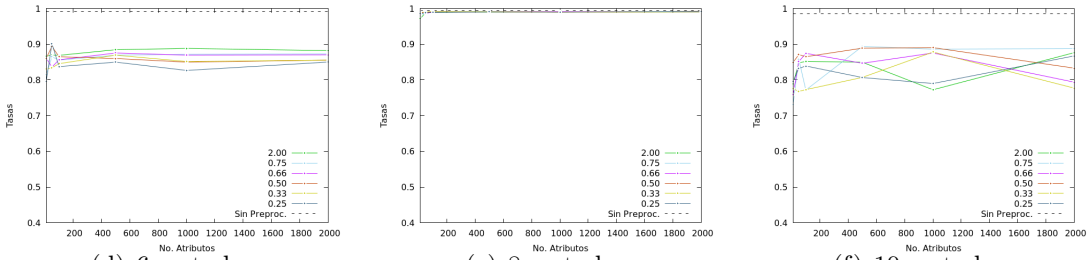
Figura 4.13: Resultados de TPR de los diferentes conjuntos de datos OVA sin preprocesamiento y Disimilitud + SMOTE + ENN.

conjunto de datos sin preprocesar como referencia.

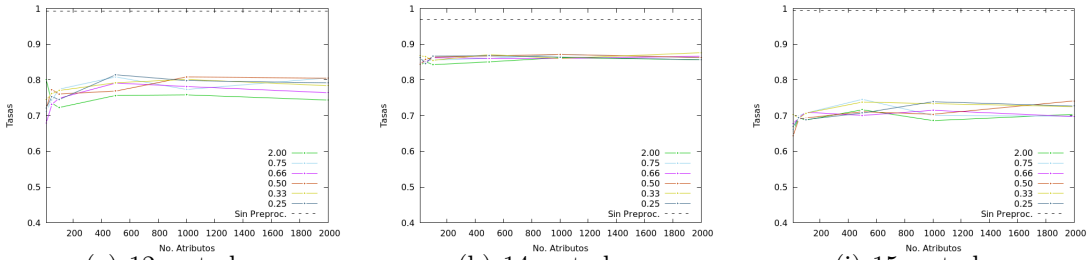
Los resultados, presentados en la Tabla 4.4, resaltan en negritas los valores más favorables. En general, se observa que el método propuesto contribuye a reducir la complejidad de los datos en la mayoría de los casos. No obstante, en los conjuntos de datos “5 vs todos” y “15 vs todos”, el mejor valor



(a) 2 vs todos (b) 3 vs todos (c) 5 vs todos



(d) 6 vs todos (e) 8 vs todos (f) 10 vs todos



(g) 12 vs todos (h) 14 vs todos (i) 15 vs todos

Figura 4.14: Resultados de TNR de los diferentes conjuntos de datos OVA sin preprocesamiento y Disimilitud + SMOTE + ENN.

de $F2$ se obtuvo en los datos preprocesados únicamente con disimilitud.

Estos hallazgos, junto con las mejoras en las tasas de clasificación al aplicar el método sistemático, respaldan su eficacia para abordar los desafíos de solapamiento y alta dimensionalidad analizados en esta investigación.

Tabla 4.4: Comparación de las métricas de complejidad con diferente preprocesamiento.

Conjunto de Datos	Preprocesamiento	Distancia	Atributos	$\mathbf{F1_{norm}}$	F2
2 vs todos	Ninguno	N/A	24832	0.953816	0.054276
	Disimilitud	0.25	10	0.914146	0.061335
	Disim. + SMOTE	0.25	10	0.639925	0.089869
	Disim. + SMOTE + ENN	0.25	10	**0.526725**	**0.036671**
3 vs todos	Ninguno	N/A	24832	0.978682	0.029194
	Disimilitud	0.75	2000	0.963123	0.031250
	Disim. + SMOTE	0.75	2000	0.700579	0.060579
	Disim. + SMOTE + ENN	0.75	2000	**0.629412**	**0.020730**
5 vs todos	Ninguno	N/A	24832	0.979122	0.012199
	Disimilitud	0.50	2000	0.978573	**0.011102**
	Disim. + SMOTE	0.50	2000	0.763306	0.027331
	Disim. + SMOTE + ENN	0.50	2000	**0.746300**	0.011407
6 vs todos	Ninguno	N/A	24832	0.991923	0.023986
	Disimilitud	2.00	500	0.976838	0.025905
	Disim. + SMOTE	2.00	500	0.779219	0.036680
	Disim. + SMOTE + ENN	2.00	500	**0.745823**	**0.013480**
8 vs todos	Ninguno	N/A	24832	0.992368	0.007333
	Disimilitud	0.25	50	0.968495	0.004797
	Disim. + SMOTE	0.25	50	0.616588	0.003548
	Disim. + SMOTE + ENN	0.25	50	**0.609130**	**0.000566**
10 vs todos	Ninguno	N/A	24832	0.966343	0.033375
	Disimilitud	0.75	2000	0.945492	0.037692
	Disim. + SMOTE	0.75	2000	0.643044	0.054285
	Disim. + SMOTE + ENN	0.75	2000	**0.568197**	**0.024166**
12 vs todos	Ninguno	N/A	24832	0.982738	0.025151
	Disimilitud	0.50	500	0.981833	0.025699
	Disim. + SMOTE	0.50	500	0.814656	0.048126
	Disim. + SMOTE + ENN	0.50	500	**0.767193**	**0.020528**
14 vs todos	Ninguno	N/A	24832	0.917240	0.056469
	Disimilitud	2.00	500	0.885820	0.058868
	Disim. + SMOTE	2.00	500	0.537600	0.061327
	Disim. + SMOTE + ENN	2.00	500	**0.436504**	**0.016636**
15 vs todos	Ninguno	N/A	24832	0.995374	0.018983
	Disimilitud	0.50	50	0.989847	**0.018914**
	Disim. + SMOTE	0.50	50	0.808780	0.105146
	Disim. + SMOTE + ENN	0.50	50	**0.732539**	0.036361

4.3 Prueba Estadística no Paramétrica

Para determinar si las diferencias en los resultados obtenidos son estadísticamente significativas, se aplicó la prueba no paramétrica de Friedman, seguida de la prueba *post-hoc* de Dunn-Bonferroni. Estas pruebas permiten identificar qué algoritmos presentan un rendimiento significativamente superior, inferior o similar en comparación con los demás.

El análisis estadístico se realizó utilizando los valores de AUC-ROC de cuatro escenarios: los datos sin preprocesar, los datos preprocesados con SMOTE_{il} y los conjuntos de datos que obtuvieron el mejor AUC-ROC tras aplicar el método sistemático (ver Tabla 4.3).

Los resultados de la prueba de Friedman, presentados en la Tabla 4.5, indican que el preprocesamiento con mejor desempeño tiene el rango más bajo, mientras que el de peor desempeño presenta el rango más alto (García et al., 2010; Demšar, 2006). En este contexto, el método sistemático Disimilitud + SMOTE + ENN obtuvo el mejor desempeño con un rango de 1.1111, seguida de Disimilitud + SMOTE con 2.2222. Por otro lado, la disimilitud sin preprocesamiento adicional mostró el peor rendimiento, incluso por debajo de los datos sin preprocesar.

Estos resultados resaltan la importancia de combinar múltiples estrategias de preprocesamiento para abordar conjuntos de datos complejos, mejo-

rando así la efectividad de los modelos de clasificación.

Tabla 4.5: Rangos promedio de los datos preprocesados con las diferentes técnicas y combinaciones de técnicas.

Preprocesamiento	**Rango**
Ninguno	4.0000
SMOTE_{il}	2.6667
Disimilitud	5.0000
Disim. + SMOTE	2.2222
Disim. + SMOTE + ENN	1.1111

La prueba de Friedman, considerando la distribución χ^2 con 4 grados de libertad, arrojó un valor estadístico de 33.422222 y un valor de P de 9.788572×10^{-7}.

Asimismo, al aplicar la estadística de Iman-Davenport, basada en la distribución F con 4 y 32 grados de libertad, se obtuvo un valor de 103.724138 y un valor de P de 7.572429×10^{-18}. Dado que ambos valores de P son menores que el nivel de significancia $\alpha = 0.05$, se rechaza la hipótesis nula, lo que indica la existencia de diferencias significativas entre las técnicas y combinaciones de técnicas de preprocesamiento evaluadas. Esto sugiere que al menos una de las técnicas proporciona un rendimiento de clasificación significativamente distinto en comparación con las demás.

Para identificar qué pares de técnicas o combinaciones de técnicas presentan diferencias significativas, se llevó a cabo el análisis *post-hoc* de Dunn-Bonferroni. Este análisis es fundamental en la comparación de múltiples métodos, ya que permite determinar específicamente cuáles presentan diferencias estadísticas (Shaffer, 1995; García y Herrera, 2008).

Los resultados de las pruebas estadísticas se presentan en la Tabla 4.6. En este análisis, el valor z representa la diferencia entre los rangos promedio de dos técnicas, normalizada por el error estándar. En este contexto, R_0 corresponde al rango promedio de la mejor técnica o combinación de técnicas observada, R_i representa el rango promedio de la técnica comparada y EE indica el error estándar de la diferencia entre los rangos. Un valor z cercano a cero sugiere que no hay diferencias significativas entre las técnicas comparadas, mientras que valores elevados, ya sean positivos o negativos, indican diferencias significativas. De manera análoga, los valores con $p < \alpha$ se consideran estadísticamente distintos.

En la Tabla 4.6 se observa que Disimilitud + SMOTE + ENN muestra un desempeño significativamente superior en comparación con los datos sin preprocesar (i = 9), con SMOTE$_{il}$ (i = 5) y con los datos preprocesados únicamente con disimilitud (i = 10). Sin embargo, no se identificaron diferencias significativas entre Disimilitud + SMOTE y Disimilitud + SMOTE + ENN (i = 3). No obstante, el análisis de métricas de complejidad (ver Tabla 4.4) sugiere que la inclusión de ENN contribuye a reducir la complejidad de los datos en términos de $F1norm$ y $F2$.

Asimismo, no se detectaron diferencias significativas entre SMOTE$_{il}$ y Disimilitud + SMOTE (i = 1). Sin embargo, es importante destacar que los resultados obtenidos con Disimilitud + SMOTE se lograron utilizando menos del 10% del número total de atributos, lo que resalta la eficiencia en términos de reducción de dimensionalidad.

Tabla 4.6: Tabla de valores de P para $\alpha = 0.05$.

i	Preprocesamiento	$\mathbf{z = (R_0 - R_i)/EE}$	p
10	Disimilitud vs. Disim. + SMOTE + ENN	5.217492	0.000000
9	Sin Preprocesar vs. Disim. + SMOTE + ENN	3.875851	0.000106
8	Disimilitud vs. Disim. + SMOTE	3.726780	0.000194
7	SMOTE_{il} vs. Disimilitud	3.130495	0.001745
6	Sin Preprocesar vs. Disim. + SMOTE	2.385139	0.017073
5	SMOTE_{il} vs. Disim. + SMOTE + ENN	2.086997	0.036888
4	Sin Preprocesar vs. SMOTE_{il}	1.788854	0.073638
3	Disim. + SMOTE vs. Disim. + SMOTE + ENN	1.490712	0.136037
2	Sin Preprocesar vs. Disimilitud	1.341641	0.179712
1	SMOTE_{il} vs. Disim. + SMOTE	0.596285	0.550985

En resumen y a partir de la información obtenida de la prueba estadística no paramétrica mostrada en esta sección, se puede confirmar que "Se mejora significativamente el rendimiento predictivo del clasificador cuando se entrena con el conjunto de datos preprocesado, (es decir, aquel en el que se trataron la dimensionalidad, el desbalance y solapamiento de clases con el método propuesto) que con el conjunto de datos original". En otras palabras, se rechaza la hipótesis nula y se acepta la alternativa (para mayor detalle sobre la hipótesis de este trabajo, véase la sección 1.5).

4.4 Discusión

En este estudio se evaluaron diversas técnicas y combinaciones de técnicas de preprocesamiento y clasificación para abordar los desafíos de datos no balanceados, con alta dimensionalidad y solapamiento, en entornos de big data. Se utilizaron clasificadores DT junto con métodos de sobremuestreo y

edición para la mejora de datos.

El uso de SMOTE$_{il}$ demostró ser eficaz, mejorando las métricas de clasificación en comparación con los conjuntos de datos sin preprocesar, lo que reafirma la utilidad de SMOTE como una herramienta valiosa de preprocesamiento en escenarios con clases no balanceadas. La implementación de Disimilitud + SMOTE y Disimilitud + SMOTE + ENN en un entorno distribuido mostró mejoras significativas, especialmente en conjuntos de datos con un alto IR. Estas técnicas no solo mejoraron el AUC-ROC, sino que también lograron incrementos notables en la TPR, demostrando su capacidad para manejar de manera efectiva grandes volúmenes de datos y alta dimensionalidad.

Las técnicas de Disimilitud + SMOTE superaron a SMOTE$_{il}$ en casi todos los casos evaluados. Esto sugiere que la incorporación de la disimilitud como parte del proceso de preprocesamiento es altamente beneficiosa para la clasificación en espacios de alta dimensionalidad. Además, la integración de ENN en el flujo de trabajo de preprocesamiento con Disimilitud + SMOTE aportó una mejora adicional en el rendimiento de clasificación, reduciendo la complejidad de los datos, como se observa en la Tabla 4.4, y disminuyendo el número de atributos necesarios para alcanzar métricas de desempeño comparables o superiores. Los conjuntos de datos con mayor IR, como "15 vs todos", "8 vs todos" y "5 vs todos", que presentaban una mayor complejidad $F1_{norm}$, mostraron una notable mejora tras ser preprocesados con la combinación de técnicas que conforman el método sistemático propuesto, logrando además mejores tasas de clasificación. Este hallazgo destaca la efectividad del método

sistemático para reducir tanto la dimensionalidad como la complejidad de los datos, mejorando así significativamente el rendimiento de los clasificadores.

Los resultados obtenidos con la combinación de Disimilitud + SMOTE + ENN indican que el método sistemático no solo es capaz de mejorar las métricas de clasificación, sino que también lo hace de manera eficiente en términos de recursos computacionales, gestionando eficazmente la maldición de la dimensionalidad.

Estos hallazgos fueron respaldados por las pruebas estadísticas realizadas, específicamente la prueba de Friedman y la de Iman-Davenport, cuyos resultados indicaron diferencias significativas entre las técnicas de preprocesamiento evaluadas. La prueba *post-hoc* de Dunn-Bonferroni confirmó que la combinación de técnicas Disimilitud + SMOTE + ENN obtuvo los mejores resultados en comparación con los datos sin preprocesar, SMOTE_{il} y Disimilitud.

El rechazo de la hipótesis nula y las diferencias significativas no solo justifica el esfuerzo adicional y los recursos computacionales involucrados en la implementación del método sistemático de preprocesamiento, sino que también subraya la importancia de elegir estrategias adecuadas para el manejo de datos complejos. Los hallazgos destacan la relevancia de seleccionar y combinar técnicas de preprocesamiento y clasificación de manera estratégica para abordar problemas de datos no balanceados con alta dimensionalidad y solapamiento, especialmente en contextos de big data.

Las técnicas y combinaciones de técnicas evaluadas en este estudio se presentan como soluciones prometedoras para mejorar la precisión de los modelos predictivos en escenarios desafiantes y representan un avance significativo hacia la implementación eficiente y escalable de algoritmos de aprendizaje automático en grandes conjuntos de datos. Investigaciones futuras podrían explorar la adaptación de este método sistemático a otros tipos de modelos clasificatorios y su aplicación en diferentes dominios de datos para validar y ampliar su aplicabilidad y robustez.

Capítulo 5

Conclusiones

Este capítulo ofrece una síntesis de los hallazgos más relevantes de la investigación, contextualizando los resultados en relación con las preguntas de investigación planteadas. Se discute cómo estos hallazgos contribuyen al avance del conocimiento en el ámbito del preprocesamiento de datos y el aprendizaje automático en entornos de big data. Además, se resaltan las contribuciones específicas de este estudio y se proponen direcciones para futuras investigaciones, orientadas a profundizar y ampliar los descubrimientos actuales.

5.1 Síntesis de los Hallazgos

En esta investigación, se propuso un método sistemático para mejorar el rendimiento de los modelos de clasificación, utilizando DT para evaluar las estrategias de preprocesamiento en la resolución de problemas de datos no balanceados, alta dimensionalidad y solapamiento en entornos de big data. El método propuesto está compuesto de las siguientes tres etapas:

1. **Tratamiento de la alta dimensionalidad**: Se transformó el espacio de características del conjunto de datos al espacio de disimilitud, resul-

tando en una reducción de la dimensionalidad en comparación con el espacio original. Debido a la alta dimensionalidad del conjunto de datos en el espacio de características, se utilizaron distancias fraccionarias en lugar de la distancia euclidiana para calcular las disimilitudes, mitigando así los efectos de la maldición de la dimensionalidad. Este cálculo de disimilitudes se implementó de manera distribuida en la plataforma Apache Spark.

2. **Balance de clases**: Para resolver el problema del desbalance de clases, se implementó el algoritmo SMOTE-DB usando kNN-IS en Spark para la búsqueda de los k-vecinos más cercanos en ambientes distribuidos, el cual fue aplicado al conjunto de datos transformado a un espacio de disimilitud. Esta técnica de sobremuestreo permitió balancear las clases en el conjunto de datos de manera efectiva.

3. **Solapamiento de clases**: La última etapa del método sistemático involucró ENN, utilizando kNN-IS, lo cual ayudó a reducir el solapamiento de clases. Esto se corroboró mediante el cálculo de las métricas de complejidad $F1_{norm}$ y $F2$, y al mejorar las tasas de clasificación.

La implementación del método sistemático en un entorno distribuido demostró mejoras significativas en las métricas de desempeño del clasificador DT, especialmente en conjuntos de datos con un alto IR. Este método no solo incrementó el AUC-ROC, sino que también lograron aumentos notables en la TPR, lo que refleja su capacidad para manejar eficientemente grandes

volúmenes de datos con alta dimensionalidad. Además, el método sistemático mejoró el rendimiento de clasificación al alcanzar resultados superiores con un menor número de atributos, lo que resalta su eficacia para reducir la dimensionalidad. Al estar implementada en Spark, se garantiza su escalabilidad para aplicarse a conjuntos de datos aún más grandes.

Los análisis estadísticos, que incluyeron las pruebas de Friedman e Iman-Davenport, seguidos del análisis *post-hoc* de Dunn-Bonferroni, confirmaron que las diferencias en el rendimiento entre las técnicas y combinaciones de técnicas evaluadas eran estadísticamente significativas. En particular, la combinación de las técnicas de Disimilitud + SMOTE + ENN demostró ser superior a las técnicas tradicionales, validando su efectividad y justificando la adopción del método propuesto en el procesamiento de grandes volúmenes de datos con características complejas.

5.2 Relación con las Preguntas de Investigación

En relación con las preguntas de investigación planteadas en la Sección 1.2.1, se pueden concluir los siguientes puntos:

1. **¿Cómo afecta la aplicación de algoritmos de preprocesamiento basados en distancia euclidiana el rendimiento en conjuntos de datos masivos no balanceados con alta dimensionalidad?**

   La investigación revela que los algoritmos de preprocesamiento basados en la distancia euclidiana, como SMOTE, mejoran significativamente las

métricas de clasificación en conjuntos de datos no balanceados en comparación con aquellos que no han sido preprocesados. Sin embargo, su efectividad se ve limitada en escenarios de alta dimensionalidad, debido a la maldición de la dimensionalidad, que dificulta la captura adecuada de las relaciones relevantes entre los datos. Además, las técnicas tradicionales enfrentan restricciones significativas relacionadas con la capacidad de un solo nodo de cómputo, lo que impide su aplicación a conjuntos de datos masivos una vez que se supera su capacidad de procesamiento. Estos hallazgos subrayan la necesidad de explorar alternativas que manejen de manera más efectiva la complejidad de los datos en entornos de big data.

2. **¿Cuál es el orden óptimo para abordar las problemáticas de clases no balanceadas, solapamiento y alta dimensionalidad en el preprocesamiento de datos?**

   El método sistemático propuesto en esta investigación establece un orden específico que ha demostrado ser efectivo para abordar estas problemáticas. Primero, se realiza la transformación al espacio de disimilitud, lo que reduce la dimensionalidad y mitiga los problemas relacionados con espacios de alta dimensión. A continuación, se aplica SMOTE-BD para balancear las clases, utilizando distancias fraccionarias en lugar de la distancia euclidiana. Finalmente, se emplea la ENN para reducir el solapamiento entre clases. Este enfoque secuencial no solo mejora las métricas de clasificación, sino que también facilita el manejo eficiente

de grandes volúmenes de datos, sugiriendo que esta secuencia es óptima para el preprocesamiento en entornos de big data.

3. **¿Qué beneficios se obtienen al incorporar distancias fraccionarias en los algoritmos de preprocesamiento tradicionalmente basados en distancia euclidiana para problemas de alta dimensionalidad?**

   La incorporación de distancias fraccionarias en lugar de la distancia euclidiana en los algoritmos de preprocesamiento presenta varios beneficios importantes. En esta investigación, el uso de distancias fraccionarias en la transformación al espacio de disimilitud mitigó los efectos de la maldición de la dimensionalidad, mejorando la capacidad de los algoritmos para diferenciar entre clases en espacios de alta dimensión. Esto se reflejó en un rendimiento superior del clasificador, con mejoras en métricas como AUC-ROC y TPR. Los resultados indican que las distancias fraccionarias son más efectivas para capturar relaciones significativas en datos de alta dimensionalidad, contribuyendo así a una clasificación más precisa y robusta.

4. **¿En qué medida la transformación al espacio de disimilitud contribuye a la reducción de la dimensionalidad en conjuntos de datos con alta dimensionalidad?**

   La investigación demuestra que la transformación al espacio de disimilitud contribuye a la reducción de la dimensionalidad y se logra un mejor

desempeño de clasificación con 10% o menos de las dimensiones originales del conjunto de datos. Este enfoque permite representar los datos en un espacio de menor número de dimensiones, donde las relaciones entre instancias se capturan a través de vectores de disimilitud, utilizando distancias fraccionarias. Esto no solo reduce el número de atributos necesarios, sino que también se mantiene o mejora la capacidad de los clasificadores para diferenciar entre clases. Al mejorar la eficiencia computacional y facilitar el manejo de grandes volúmenes de datos, esta transformación contribuye a una clasificación más efectiva y escalable en contextos de big data.

5.3 Objetivos Específicos

Se cumplió con los objetivos específicos de la siguiente manera:

1. **Analizar las técnicas existentes en la literatura científica para abordar el problema de las clases no balanceadas con solapamiento en conjuntos de datos masivos y de alta dimensionalidad.** En la Sección 2 se presentó una revisión exhaustiva de la literatura científica existente sobre el problema de las clases no balanceadas con solapamiento en conjuntos de big data y de alta dimensionalidad. Se realizó un análisis crítico de las diversas técnicas propuestas, evaluando sus fortalezas, debilidades y aplicabilidad a diferentes escenarios. Además, se llevó a cabo un análisis bibliométrico para identificar las tendencias de investigación y las principales líneas de desarrollo en este campo.

2. **Reproducir y evaluar algoritmos de preprocesamiento existentes en la literatura científica para el tratamiento de clases no balanceadas, centrándose en su efectividad en escenarios de big data.** Se llevó a cabo la reproducción de los algoritmos kNN-IS de Maillo et al. (2017) y SMOTE-BD de Basgall et al. (2018). Este último algoritmo fue seleccionado por su relevancia en el tratamiento de clases desbalanceadas en escenarios de big data. Los experimentos realizados permitieron evaluar su efectividad en términos de AUC-ROC sobre un conjunto de big data. Por otro lado, se intentó reproducir el algoritmo de selección de características BELIEF de Ramírez-Gallego et al. (2021), sin embargo, se encontraron dificultades al aplicarlo a datos de alta dimensionalidad, lo cual sugiere que este algoritmo podría requerir ajustes o modificaciones para su uso en este tipo de escenarios.

3. **Desarrollar un algoritmo escalable de sobremuestreo que incorpore el uso de distancias fraccionarias para mejorar el equilibrio entre clases en entornos de alta dimensionalidad.** Con el objetivo de abordar el desequilibrio de clases en entornos big data con alta dimensionalidad, se propuso una extensión del algoritmo SMOTE-BD que incorpora el concepto de distancias fraccionarias en la búsqueda de vecinos cercanos. Esta modificación busca mejorar la generación de instancias sintéticas en espacios de alta dimensionalidad. Los resultados experimentales, presentados en la Sección 3.3, demuestran la efectividad de este enfoque en términos de AUC-ROC.

4. **Desarrollar un algoritmo escalable para la transformación del espacio de características con alta dimensionalidad hacia un espacio de disimilitud con una dimensionalidad reducida, optimizando la separabilidad entre clases, utilizando distancias fraccionarias.** Se propuso un algoritmo que combina las distancias fraccionarias con la eficiencia del procesamiento de datos distribuidos en Spark para transformar el espacio de características en un espacio de disimilitud con una dimensionalidad reducida. El uso de distancias fraccionarias permitió obtener mejoras en términos de AUC-ROC.

5. **Desarrollar un algoritmo escalable de selección de instancias que mejore la clasificación en conjuntos de datos masivos no balanceados con solapamiento.** Se desarrolló el algoritmo distribuido ENN para la selección de instancias basado en el algoritmo de Edición de Wilson. La implementación se realizó utilizando computación paralela con Spark para aprovechar la potencia de cálculo de múltiples procesadores. Para la búsqueda de vecinos cercanos, se utilizó el algoritmo kNN-IS. Se demostró en el apartado *Métricas de Complejidad* de la sección 4.2 que la Edición de Wilson fue eficaz en la identificación de instancias ruidosas o solapadas al mejorar las métricas $F1_{norm}$ y $F2$.

6. **Evaluar el desempeño del enfoque desarrollado frente a otros métodos o técnicas, utilizando métricas de efectividad y complejidad en la clasificación.** Para evaluar la efectividad del método

sistemático propuesto, se llevó a cabo un estudio comparativo con el algoritmo SMOTE, ampliamente utilizado en la literatura para tratar el desbalance de clases. Se utilizaron las métricas de AUC-ROC, G-Mean, TPR y TNR para evaluar el rendimiento de ambos métodos. Además, se utilizaron las métricas de complejidad $F1_{norm}$ y $F2$, para evaluar la complejidad de los datos antes y después de ser preprocesados. Los resultados obtenidos muestran que el método sistemático propuesto supera significativamente a SMOTE en términos de AUC-ROC, G-Mean y TPR. De igual manera, se mejoraron las métricas $F1_{norm}$ y $F2$ sugiriendo que los datos preprocesados tienen una menor complejidad. Se intentó comparar los resultados con SMOTE-BD en el espacio de características original, y la técnica de selección de características BELIEF, sin embargo, debido a la alta dimensionalidad del conjunto de datos y a posibles limitaciones en la implementación de estas técnicas, no fue posible preprocesar el conjunto de datos con ellos. Estos hallazgos sugieren que el método sistemático propuesto es una alternativa prometedora para abordar el problema de las clases no balanceadas con solapamiento y alta dimensionalidad en big data.

7. **Realizar análisis estadísticos para determinar la significancia de los resultados obtenidos, y validar o refutar las hipótesis relacionadas con la mejora en el rendimiento y la eficiencia del modelo propuesto.** En el apartado de *Prueba Estadística* de la sección 4.3, se presentan los resultados de las pruebas estadísticas realizadas

para evaluar la significancia de las mejoras obtenidas con el método sistemático propuesto. Se emplearon las pruebas de Friedman e Iman-Davenport, además de la prueba *post-hoc* Dunn-Bonferroni para comparar el rendimiento del método sistemático con el algoritmo SMOTE y las tasas de clasificación de los datos sin preprocesar. Los resultados muestran que el método sistemático obtuvo un valor de p inferior a 0.05 para las métricas de AUC-ROC, lo que indica que las diferencias en el desempeño son estadísticamente significativas. Estos hallazgos soportan nuestra hipótesis de que el método sistemático propuesto mejora significativamente la clasificación en conjuntos big data con clases no balanceadas, solapamiento y alta dimensionalidad.

5.4 Contribuciones al Campo

Las contribuciones de esta investigación al campo del preprocesamiento de datos en entornos de big data y aprendizaje automático son diversas. A continuación, se detallan las principales contribuciones:

1. Impacto de la densidad de datos en el preprocesamiento: Se demostró experimentalmente que la densidad de datos es un factor predominante en técnicas de preprocesamiento basadas en la distancia euclidiana como SMOTE, inclusive, tiene mayor influencia que la dimensionalidad del conjunto de datos en el cálculo de la distancia euclidiana.

2. Se introduce un método sistemático de preprocesamiento que combina

diferentes técnicas: la transformación al espacio de disimilitud, SMOTE y ENN en el entorno distribuido de Spark. Esta combinación aborda de manera eficaz los problemas de clases no balanceadas, alta dimensionalidad y solapamiento, superando las limitaciones de las técnicas tradicionales de preprocesamiento para manejar conjuntos de datos masivos.

3. Se valida que las distancias fraccionarias son más efectivas que la distancia euclidiana en escenarios de alta dimensionalidad. Esto contribuye al campo proponiendo un cambio de enfoque en las métricas de distancia utilizadas en los espacios de disimilitud y para el preprocesamiento de datos, alentando a la adopción de distancias alternativas que mejoren el rendimiento en espacios de alta dimensión. Sugiriendo que estas métricas de distancia deben considerarse seriamente en el desarrollo de nuevas técnicas de preprocesamiento y clasificación para problemas de alta dimensionalidad.

4. La validación de la secuencia específica de preprocesamiento (transformación al espacio de disimilitud, seguida de SMOTE y finalmente ENN) proporciona una guía práctica y fundamentada en la selección del orden de aplicación de técnicas de preprocesamiento en ambientes distribuidos.

5. La investigación muestra que la combinación de disimilitud, SMOTE y ENN en big data no solo mejora las métricas de clasificación, sino que también reduce el solapamiento de clases demostrado por medio del cálculo de las métricas de complejidad $F1_{norm}$ y $F2$, lo cual es un desafío

común en conjuntos de datos no balanceados. Este enfoque contribuye a la literatura existente sobre cómo manejar eficazmente el solapamiento para mejorar la calidad del modelo.

6. La utilización de pruebas estadísticas no paramétricas como Friedman e Iman-Davenport, junto con análisis *post-hoc* de Dunn-Bonferroni, proporciona una validación sólida del método sistemático propuesto. Esto no solo refuerza la credibilidad de los resultados obtenidos experimentalmente, sino que también establece un estándar de rigor metodológico para futuras investigaciones en el campo.

7. Aunque la investigación se centró en dos conjuntos de datos no balanceados de alta dimensionalidad, el método sistemático propuesto tiene el potencial de ser adaptado a otros conjuntos de datos. Esto amplía el impacto de los hallazgos, haciendo que las contribuciones sean relevantes en una amplia gama de aplicaciones industriales y científicas.

8. La investigación establece una base sólida para la exploración de nuevas técnicas de preprocesamiento y para la investigación de métricas de distancia no convencionales. También sugiere que futuras investigaciones podrían enfocarse en la combinación de técnicas similares con otros tipos de clasificadores o en diferentes configuraciones de aprendizaje automático.

9. En Bolívar et al. (2022) se publicó el estudio de datos con baja y alta densidad en ambientes big data y como la densidad impacta en el de-

sempeño de SMOTE y en Bolívar et al. (2024) se presentó el método sistemático de preprocesamiento utilizando espacios de disimilitud con distancias fraccionarias y SMOTE en ambientes big data.

5.5 Futuras Investigaciones

Las direcciones para futuras investigaciones a partir de los hallazgos de esta investigación son variadas y ofrecen múltiples oportunidades para profundizar y expandir el conocimiento en el campo del preprocesamiento de datos, aprendizaje automático y big data. A continuación, se presentan algunas direcciones prometedoras.

1. Exploración de otras métricas de distancia:

    - Si bien la investigación actual se centró en algunas distancias fraccionarias, futuros estudios podrían explorar una gama más amplia de estas métricas para identificar aquellas que sean aún más efectivas en espacios de alta dimensionalidad y en la mejora del rendimiento de clasificación. Además, sería interesante investigar cómo diferentes distancias fraccionarias se comportan con distintos tipos de datos y dominios.
    - Investigar métricas de distancia alternativas y más sofisticadas, como la distancia de Mahalanobis, distancias basadas en aprendizaje (*learning-based distances*), o métricas de distancia no lineales,

podría proporcionar nuevas maneras de mejorar la discriminación entre clases en espacios de alta dimensionalidad.

2. Aplicación en otros tipos de datos y dominios:

   - Futuros estudios podrían investigar cómo se comporta el método sistemático cuando se aplica al procesamiento de lenguaje natural (NLP), o datos secuenciales (como series de tiempo). Estos dominios presentan desafíos únicos de alta dimensionalidad y complejidad que podrían beneficiarse.
   - Dado que los conjuntos de datos en biomedicina y genómica son a menudo de alta dimensionalidad y presentan problemas de clases no balanceadas, aplicar y adaptar el método propuesto a dichos dominios podría mejorar la precisión de modelos predictivos en diagnósticos médicos y en investigaciones de salud.

3. Mejoras en la eficiencia computacional:

   - Futuros trabajos podrían centrarse en la optimización de las implementaciones actuales de SMOTE y ENN para hacer un uso más eficiente de los recursos computacionales en entornos distribuidos, reduciendo los tiempos de procesamiento y mejorando la escalabilidad.
   - Investigar cómo implementar el método sistemático de preprocesamiento propuesto en un marco de tiempo real, permitiendo la integración con sistemas de análisis de flujo de datos (*streaming data*),

lo que sería crucial para aplicaciones que requieren decisiones rápidas basadas en grandes volúmenes de datos en constante cambio.

4. Investigación en la combinación de otras técnicas de preprocesamiento.

    - Explorar la combinación de más técnicas de preprocesamiento, como técnicas de reducción de dimensionalidad basadas en PCA, t-SNE, UMAP o técnicas de normalización avanzadas con las ya utilizadas en el método sistemático propuesto. Esto podría llevar a descubrimientos de combinaciones sinérgicas que no solo mejoren el rendimiento de clasificación, sino también la interpretabilidad del modelo.

5. Estudios de caso en la industria y evaluaciones de impacto.

    - Colaborar con industrias para implementar este método sistemático en problemas del mundo real, como la detección de fraude, personalización de marketing, mantenimiento predictivo o análisis de riesgos. Evaluar el impacto en la toma de decisiones empresariales y en la eficiencia operativa podría proporcionar una retroalimentación valiosa y oportunidades para ajustar y mejorar las técnicas.

Bibliografía

Abdel-Hamid, N. B., ElGhamrawy, S., Desouky, A. E. y Arafat, H. A dynamic spark-based classification framework for imbalanced big data. *Journal of Grid Computing*, vol. 16(4), páginas 607–626, 2018.

Abdelkhalek, A. y Mashaly, M. Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. *The Journal of Supercomputing*, vol. 79(10), página 10611–10644, 2023. ISSN 1573-0484.

Aggarwal, C. C., Hinneburg, A. y Keim, D. A. On the surprising behavior of distance metrics in high dimensional space. En *Database Theory — ICDT 2001*, páginas 420–434. Springer Berlin Heidelberg, 2001.

Ahlawat, K., Chug, A. y Singh, A. P. Benchmarking framework for class imbalance problem using novel sampling approach for big data. *International Journal of System Assurance Engineering and Management*, vol. 10(4), páginas 824–835, 2019.

Ali, A., Shamsuddin, S. M. y Ralescu, A. Classification with class imbalance problem: A review. *International Journal of Advances in Soft Computing and its Application*, vol. 7(3), páginas 176–204, 2015.

Amazon. Big data en aws. https://aws.amazon.com/es/big-data/use-cases/, 2021. Visitado el 28/01/2021.

Anand, R., Mehrotra, K., Mohan, C. y Ranka, S. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, vol. 6(1), página 117–124, 1995. ISSN 1045-9227.

Bagui, S. y Li, K. Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data*, vol. 8(1), 2021.

Barella, V. H., Garcia, L. P., de Souto, M. C., Lorena, A. C. y de Carvalho, A. C. Assessing the data complexity of imbalanced datasets. *Information Sciences*, vol. 553, páginas 83–109, 2021.

Basgall, M. J., Hasperué, W., Naiouf, M., Fernández, A. y Herrera, F. Smote-bd: An exact and scalable oversampling method for imbalanced classification in big data. En *VI Jornadas de Cloud Computing & Big Data (JCC&BD)(La Plata, 2018)*. 2018.

Basgall, M. J., Hasperué, W., Naiouf, M., Fernández, A. y Herrera, F. An analysis of local and global solutions to address big data imbalanced classification: A case study with smote preprocessing. En *Cloud Computing and Big Data* (editado por M. Naiouf, F. Chichizola y E. Rucci), páginas 75–85. Springer International Publishing, Cham, 2019. ISBN 978-3-030-27713-0.

Basu, M. y Ho, T. K., editores. *Data Complexity in Pattern Recognition*. Springer London, 2006.

Bauder, R. A. y Khoshgoftaar, T. M. A study on rare fraud predictions with big medicare claims fraud data. *Intelligent Data Analysis*, vol. 24(1), páginas 141–161, 2020.

Bauder, R. A., Khoshgoftaar, T. M. y Hasanin, T. An empirical study on class rarity in big data. En *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, páginas 785–790. 2018.

Beyer, M. y Laney, D. The importance of big data: A definition. Informe técnico, Gartner, 2012.

Blagus, R. y Lusa, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, vol. 14(106), páginas 1–16, 2013.

Bolívar, A., García, V., Alejo, R., Florencia-Juárez, R. y Sánchez, J. S. Data-centric solutions for addressing big data veracity with class imbalance, high dimensionality, and class overlapping. *Applied Sciences*, vol. 14(13), página 5845, 2024. ISSN 2076-3417.

Bolívar, A., García, V., Florencia, R., Alejo, R., Rivera, G. y Sánchez-Solís, J. P. *A Preliminary Study of SMOTE on Imbalanced Big Datasets When Dealing with Sparse and Dense High Dimension-*

ality, página 46–55. Springer International Publishing, 2022. ISBN 9783031077500.

Brennan, P. *A comprehensive survey of methods for overcoming the class imbalance problem in fraud detection*. Proyecto Fin de Carrera, Institute of Technology Blanchardstown, Dublin, Ireland, 2012.

Chapelle, O., Scholkopf, B. y Zien, A., editores. *Semi-Supervised Learning*. Adaptive Computation and Machine Learning series. MIT Press, London, England, 2010.

Chawla, N. V., Bowyer, K. W., Hall, L. O. y Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, vol. 16, páginas 321–357, 2002.

Chen, L., Jiang, J. y Zhang, Y. HSDP: A hybrid sampling method for imbalanced big data based on data partition. *Complexity*, vol. 2021, páginas 1–9, 2021.

Cormode, G., Indyk, P., Koudas, N. y Muthukrishnan, S. Fast mining of massive tabular data via approximate distance computations. En *Proceedings 18th International Conference on Data Engineering*. IEEE Comput. Soc, 2002.

Daskalaki, S., Kopanas, I. y Avouris, N. EVALUATION OF CLASSIFIERS FOR AN UNEVEN CLASS DISTRIBUTION PROBLEM. *Applied Artificial Intelligence*, vol. 20(5), páginas 381–417, 2006.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, vol. 7, página 1–30, 2006. ISSN 1532-4435.

Deng, Y., Wang, B. y Lu, Z. A hybrid model based on data preprocessing strategy and error correction system for wind speed forecasting. *Energy Conversion and Management*, vol. 212, página 112779, 2020.

Digital, W. How to make sense of big data. `https://www.westerndigital.com/solutions/big-data`, 2020. Visitado el 2021-01-28.

Duin, R. P. y Pękalska, E. The dissimilarity space: Bridging structural and statistical pattern recognition. *Pattern Recognition Letters*, vol. 33(7), páginas 826–832, 2012.

Elreedy, D. y Atiya, A. F. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. *Information Sciences*, vol. 505, páginas 32–64, 2019. ISSN 0020-0255.

Fan, J. y Lv, J. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, páginas 101–148, 2010.

Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*, vol. 27(8), páginas 861–874, 2006.

Fayyad, U., Piatetsky-Shapiro, G. y Smyth, P. From data mining to knowledge discovery in databases. *AI magazine*, vol. 17(3), páginas 37–54, 1996.

Fernandez, A., Garcia, S., Herrera, F. y Chawla, N. V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, vol. 61, páginas 863–905, 2018.

Fernández, A., del Río, S., Chawla, N. V. y Herrera, F. An insight into imbalanced big data classification: outcomes and challenges. *Complex & Intelligent Systems*, vol. 3(2), páginas 105–120, 2017.

Flexer, A. y Schnitzer, D. Choosing ℓp norms in high-dimensional spaces based on hub analysis. *Neurocomputing*, vol. 169, páginas 281–287, 2015.

Francois, D., Wertz, V. y Verleysen, M. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, vol. 19(7), página 873–886, 2007. ISSN 1041-4347.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H. y Herrera, F. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, vol. 44(8), página 1761–1776, 2011. ISSN 0031-3203.

Galpert, D., del Río, S., Herrera, F., Ancede-Gallardo, E., Antunes, A. y Agüero-Chapin, G. An effective big data supervised imbalanced classification approach for ortholog detection in related yeast species. *BioMed Research International*, vol. 2015, páginas 1–12, 2015.

García, S., Fernández, A., Luengo, J. y Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, vol. 180(10), página 2044–2064, 2010. ISSN 0020-0255.

García, S. y Herrera, F. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research - JMLR*, vol. 9, 2008.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M. y Herrera, F. Big data preprocessing: methods and prospects. *Big Data Analytics*, vol. 1(9), páginas 1–22, 2016.

García, V., Alejo, R., Sánchez, J. S., Sotoca, J. M. y Mollineda, R. A. Combined effects of class imbalance and class overlap on instance-based classification. En *Intelligent Data Engineering and Automated Learning – IDEAL 2006*, páginas 371–378. 2006.

García, V., Sánchez, J., Marqués, A., Florencia, R. y Rivera, G. Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. *Expert Systems with Applications*, vol. 158, página 113026, 2020.

García, V., Sánchez, J. S., Domínguez, H. J. O. y Cleofas-Sánchez, L. Dissimilarity-based learning from imbalanced data with

small disjuncts and noise. En *Pattern Recognition and Image Analysis*, páginas 370–378. Springer International Publishing, 2015.

Gong, C., gang Su, Z., hong Wang, P., Wang, Q. y You, Y. Evidential instance selection for k-nearest neighbor classification of big data. *International Journal of Approximate Reasoning*, vol. 138, páginas 123–144, 2021.

Gonzalez-Lopez, J., Ventura, S. y Cano, A. Distributed nearest neighbor classification for large-scale multi-label data on spark. *Future Generation Computer Systems*, vol. 87, páginas 66–82, 2018.

Gorban, A. N., Mirkes, E. M. y Zinovyev, A. Data analysis with arbitrary error measures approximated by piece-wise quadratic PQSQ functions. En *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018.

Guzmán-Ponce, A., Valdovinos, R. M., Sánchez, J. S. y Marcial-Romero, J. R. A new under-sampling method to face class overlap and imbalance. *Applied Sciences*, vol. 10(15), páginas 1–22, 2020.

Han, H., Wang, W.-Y. y Mao, B.-H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. En *Lecture Notes in Computer Science*, páginas 878–887. 2005.

Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L. y Bauder, R. A. Investigating class rarity in big data. *Journal of Big Data*, vol. 7(23),

páginas 1–17, 2020.

Hassib, E. M., El-Desouky, A. I., El-Kenawy, E.-S. M. y El-Ghamrawy, S. M. An imbalanced big data mining framework for improving optimization algorithms performance. *IEEE Access*, vol. 7, páginas 170774–170795, 2019.

Hassib, E. M., El-Desouky, A. I., Labib, L. M. y El-kenawy, E.-S. M. WOA + BRNN: An imbalanced big data classification framework using whale optimization and deep neural network. *Soft Computing*, vol. 24(8), páginas 5573–5592, 2020.

Herrera, F. Big data: Preprocesamiento y calidad de datos. *novática*, vol. XLII(237), páginas 17–23, 2016.

Jain, A., Ratnoo, S. y Kumar, D. Addressing class imbalance problem in medical diagnosis: A genetic algorithm approach. En *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, páginas 1–8. 2017.

Japkowicz, N. y Shah, M. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

Jeon, Y.-S. y Lim, D.-J. PSU: Particle stacking undersampling method for highly imbalanced big data. *IEEE Access*, vol. 8, páginas 131920–131927, 2020.

Jiang, P. y Ma, X. A hybrid forecasting approach applied in the electrical power system based on data preprocessing, optimization and artificial intelligence algorithms. *Applied Mathematical Modelling*, vol. 40(23-24), páginas 10631–10649, 2016.

Johnson, J. M. y Khoshgoftaar, T. M. The effects of data sampling with deep learning and highly imbalanced big data. *Information Systems Frontiers*, vol. 22(5), páginas 1113–1131, 2020.

Juez-Gil, M., Arnaiz-González, A. y aínd César García-Osorio, J. J. R. Experimental evaluation of ensemble classifiers for imbalance in big data. *Applied Soft Computing*, vol. 108, página 107447, 2021a. ISSN 1568-4946.

Juez-Gil, M., Arnaiz-González, A., Rodríguez, J. J., López-Nozal, C. y García-Osorio, C. Approx-SMOTE: fast SMOTE for big data on apache spark. *Neurocomputing*, 2021b.

Kovács, G. SMOTE-variants: A python implementation of 85 minority oversampling techniques. *Neurocomputing*, vol. 366, páginas 352–354, 2019.

Kubina, M., Varmus, M. y Kubinova, I. Use of big data for competitive advantage of company. *Procedia Economics and Finance*, vol. 26, páginas 561–565, 2015.

Kuncheva, L. I., Arnaiz-González, Á., Díez-Pastor, J.-F. y Gunn, I. A. D. Instance selection improves geometric mean accuracy: a study on imbalanced data classification. *Progress in Artificial Intelligence*, vol. 8(2), páginas 215–228, 2019.

Kuo, F. Y. y Sloan, I. H. Lifting the curse of dimensionality. *Notices of the AMS*, vol. 52(11), páginas 1320–1328, 2005.

Lamba, R., Gulati, T., Al-Dhlan, K. A. y Jain, A. A systematic approach to diagnose parkinson's disease through kinematic features extracted from handwritten drawings (in press). *Journal of Reliable Intelligent Environments*, vol. 0(0), páginas 1–10, 2021.

Landgrebe, T., Paclik, P. y Duin, R. Precision-recall operating characteristic (p-ROC) curves in imprecise environments. En *18th International Conference on Pattern Recognition (ICPR'06)*. IEEE, 2006.

Leevy, J. L., Johnson, J. M., Hancock, J. y Khoshgoftaar, T. M. Threshold optimization and random undersampling for imbalanced credit card data. *Journal of Big Data*, vol. 10(1), 2023. ISSN 2196-1115.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A. y Seliya, N. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, vol. 5(42), páginas 1–30, 2018.

Lin, W., Wu, Z., Lin, L., Wen, A. y Li, J. An ensemble random forest algorithm for insurance big data analysis. *IEEE Access*, vol. 5, páginas 16568–16575, 2017.

López, V., del Río, S., Benítez, J. M. y Herrera, F. Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets and Systems*, vol. 258, páginas 5–38, 2015.

Lorena, A. C., de Carvalho, A. C. P. L. F. y Gama, J. M. P. A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, vol. 30(1–4), página 19–37, 2008. ISSN 1573-7462.

Lorena, A. C., Garcia, L. P. F., Lehmann, J., Souto, M. C. P. y Ho, T. K. How complex is your classification problem? *ACM Computing Surveys*, vol. 52(5), páginas 1–34, 2020.

Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S. y Herrera, F. Final thoughts: From big data to smart data. En *Big Data Preprocessing*, páginas 183–186. Springer International Publishing, 2020.

Ma, C., Zhang, H. H. y Wang, X. Machine learning for big data analytics in plants. *Trends in Plant Science*, vol. 19(12), páginas 798–808, 2014.

Maillo, J., Ramírez, S., Triguero, I. y Herrera, F. kNN-IS: An iterative spark-based design of the k-nearest neighbors classifier for big data. *Knowledge-Based Systems*, vol. 117, páginas 3–15, 2017.

Maldonado, S., López, J. y Vairetti, C. An alternative SMOTE oversampling strategy for high-dimensional datasets. *Applied Soft Computing*, vol. 76, páginas 380–389, 2019.

Maldonado, S., Vairetti, C., Fernandez, A. y Herrera, F. FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification. *Pattern Recognition*, vol. 124, página 108511, 2022.

Maldonado, S., Weber, R. y Famili, F. Feature selection for high-dimensional class-imbalanced data sets using support vector machines. *Information Sciences*, vol. 286, páginas 228–246, 2014.

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A. y Tourassi, G. D. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, vol. 21(2-3), páginas 427–436, 2008.

Mirkes, E. M., Allohibi, J. y Gorban, A. Fractional norms and quasinorms do not help to overcome the curse of dimensionality. *Entropy*, vol. 22(10), página 1105, 2020.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hrobjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P. y Moher, D. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *Systematic Reviews*, vol. 10, páginas 1–11, 2021.

Patil, S. y Sonavane, S. Investigation of imbalanced big data set classification: Clustering minority samples over sampling technique. En *Advances in Intelligent Systems and Computing*, páginas 299–310. Springer Singapore, 2020.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. y Brunk, C. Reducing misclassification costs. En *Machine Learning Proceedings 1994* (editado por W. W. Cohen y H. Hirsh), páginas 217–225. Morgan Kaufmann, San Francisco (CA), 1994. ISBN 978-1-55860-335-6.

Pękalska, E. y Duin, R. P. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, vol. 23(8), páginas 943–956, 2002.

Pękalska, E., Duin, R. P. y Paclík, P. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, vol. 39(2), páginas

189–208, 2006.

Pękalska, E. y Duin, R. P. W. On combining dissimilarity representations. En *Multiple Classifier Systems*, páginas 359–368. Springer Berlin Heidelberg, 2001.

Pengfei, J., Chunkai, Z. y Zhenyu, H. A new sampling approach for classification of imbalanced data sets with high density. En *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, páginas 217–222. 2014.

Petinrin, O. O., Saeed, F., Salim, N., Toseef, M., Liu, Z. y Muyide, I. O. Dimension reduction and classifier-based feature selection for oversampled gene expression data and cancer classification. *Processes*, vol. 11(7), página 1940, 2023. ISSN 2227-9717.

Pokhrel, A. R. y Wang, S. Design of fast and scalable clustering algorithm on spark. En *Proceedings of the 2020 4th International Conference on Cloud and Big Data Computing*. ACM, 2020.

Prati, R. C., Batista, G. E. A. P. A. y Monard, M. C. A survey on graphical methods for classification predictive performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, vol. 23(11), páginas 1601–1618, 2011.

Rahmati, F., Nezamabadi-pour, H. y Nikpour, B. A gravitational density-based mass sharing method for imbalanced data classification.

SN Applied Sciences, vol. 2(260), páginas 1–11, 2020.

Ramírez-Gallego, S., García, S., Xiong, N. y Herrera, F. BELIEF: A distance-based redundancy-proof feature selection method for big data (in press). *Information Sciences*, vol. 0, páginas 1–28, 2021.

Rendón, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J. y Granda-Gutiérrez, E. E. Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, vol. 10(4), páginas 1–15, 2020.

Rodríguez-Torres, F., Martínez-Trinidad, J. F. y Carrasco-Ochoa, J. A. An oversampling method for class imbalance problems on large datasets. *Applied Sciences*, vol. 12(7), página 3424, 2022.

Saez, J. A., Galar, M. y Krawczyk, B. Addressing the overlapping data problem in classification using the one-vs-one decomposition strategy. *IEEE Access*, vol. 7, páginas 83396–83411, 2019.

Shaffer, J. P. Multiple hypothesis testing. *Annual Review of Psychology*, vol. 46(1), página 561–584, 1995. ISSN 1545-2085.

Sharifai, G. A. y Zainol, Z. Feature selection for high-dimensional and imbalanced biomedical data based on robust correlation based redundancy and binary grasshopper optimization algorithm. *Genes*, vol. 11(717), páginas 1–26, 2020.

Singh, A., Khamparia, A. y Luhach, A. K. Performance comparison of apache hadoop and apache spark. En *Proceedings of the Third International Conference on Advanced Informatics for Computing Research*, ICAICR - 2019. ACM, 2019.

Sleeman IV, W. C. y Krawczyk, B. Imbalanced big data oversampling: Taxonomy, algorithms, software, guidelines and future directions. *CoRR*, vol. abs/2107.11508, 2021a.

Sleeman IV, W. C. y Krawczyk, B. Multi-class imbalanced big data classification on spark. *Knowledge-Based Systems*, vol. 212, página 106598, 2021b. ISSN 0950-7051.

Sokolova, M. y Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, vol. 45(4), páginas 427–437, 2009.

Tahvili, S. y Hatvani, L. Transformation, vectorization, and optimization. En *Artificial Intelligence Methods for Optimization of the Software Testing Process*, páginas 35–84. Elsevier, 2022.

Thudumu, S., Branch, P., Jin, J. y Singh, J. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, vol. 7(42), páginas 1–30, 2020.

Tian, C., Hao, Y. y Hu, J. A novel wind speed forecasting system based on hybrid data preprocessing and multi-objective optimization.

Applied Energy, vol. 231, páginas 301–319, 2018.

Tomasev, N., Radovanovic, M., Mladenic, D. y Ivanovic, M. The role of hubness in clustering high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, vol. 26(3), páginas 739–751, 2014.

Triguero, I., Galar, M., Merino, D., Maillo, J., Bustince, H. y Herrera, F. Evolutionary undersampling for extremely imbalanced big data classification under apache spark. En *2016 IEEE Congress on Evolutionary Computation (CEC)*, páginas 640–647. IEEE, 2016.

Triguero, I., Galar, M., Vluymans, S., Cornelis, C., Bustince, H., Herrera, F. y Saeys, Y. Evolutionary undersampling for imbalanced big data classification. En *2015 IEEE Congress on Evolutionary Computation (CEC)*, páginas 715–722. IEEE, 2015.

Triguero, I., García-Gil, D., Maillo, J., Luengo, J., García, S. y Herrera, F. Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9(2), páginas 1–24, 2018.

Tsai, C.-F., Lin, W.-C. y Ke, S.-W. Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. *Journal of Systems and Software*, vol. 122, páginas 83–92, 2016.

Tsai, C.-W., Lai, C.-F., Chao, H.-C. y Vasilakos, A. V. Big data analytics: a survey. *Journal of Big Data*, vol. 2(21), páginas 1–32, 2015.

Vairetti, C., Assadi, J. L. y Maldonado, S. Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification. *Expert Systems with Applications*, vol. 246, página 123149, 2024. ISSN 0957-4174.

Viloria, A., Lezama, O. B. P. y Mercado-Caruzo, N. Unbalanced data processing using oversampling: Machine learning. *Procedia Computer Science*, vol. 175, páginas 108–113, 2020.

Wilson, D. L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2(3), página 408–421, 1972. ISSN 2168-2909.

Wilson, D. R. y Martinez, T. R. Improved heterogeneous distance functions. *J. Artif. Int. Res.*, vol. 6(1), página 1–34, 1997. ISSN 1076-9757.

Wold, S., Esbensen, K. y Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, vol. 2(1-3), páginas 37–52, 1987.

Xu, Z., Shen, D., Nie, T. y Kou, Y. A hybrid sampling algorithm combining m-smote and enn based on random forest for medical imbal-

anced data. *Journal of Biomedical Informatics*, vol. 107, página 103465, 2020.

Yu, L., Zhou, R., Tang, L. y Chen, R. A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. *Applied Soft Computing*, vol. 69, páginas 192–202, 2018.

Zareapoor, M. y Shamsolmoali, P. Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia Computer Science*, vol. 48, páginas 679–685, 2015.

Zhai, Y., Ong, Y.-S. y Tsang, I. W. The emerging "big dimensionality". *IEEE Computational Intelligence Magazine*, vol. 9(3), páginas 14–26, 2014.

Zhu, X. y Goldberg, A. B. *Introduction to Semi-Supervised Learning*. Springer International Publishing, 2009. ISBN 9783031015489.