Universidad Autónoma del Estado de México
Universitat Jaume I

Facultad de Ingeniería
Doctorado en Ciencias de la Ingeniería

Nuevos Algoritmos Basados en Grafos y Clustering para el Tratamiento de Complejidades de los Datos

TESIS EN COTUTELA

QUE PARA OBTENER LOS GRADOS DE:

Doctora en Ciencias de la Ingeniería
Doctora en Informática

PRESENTA:

Angélica Guzmán Ponce

Directores:
Dra. Rosa María Valdovinos Rosas
Dr. José Salvador Sánchez Garreta

Tutores Adjuntos:
Dr. José Raymundo Marcial Romero
Dr. Héctor Miguel Montenegro Monroy

Marzo, 2021

Resumen

Hoy en día existen diversas áreas para las cuales la extracción de conocimiento en conjuntos de datos es esencial en la toma de decisiones. No obstante, los conjuntos de datos comúnmente tienen algunos problemas (complejidades) que decrementan la tasa de efectividad en el proceso de extracción del conocimiento. La distribución no balanceada de los datos entre las clases, así como la presencia de ruido y el traslape de clases incluidas en el conjunto de datos, son complejidades de los datos que a menudo interfieren en la efectividad de la extracción del conocimiento. Esto principalmente es debido a que la mayoría de estos modelos asumen que los datos mantienen una distribución uniforme y libre de otros problemas.

En los últimos años, las complejidades de datos han sido objeto de estudio en áreas como de Reconocimiento de Patrones y Minería de Datos dado el impacto que tienen en el rendimiento de los modelos de aprendizaje. En este sentido, la presente tesis aborda el tratamiento de desbalance de clases, traslape de clases y/o ruido con propuestas orientadas a realizar la reducción y limpieza de la clase mayoritariamente representada.

Dentro de las soluciones para el tratamiento de desbalance de clases, se proponen nuevos algoritmos basados en Teoría de grafos. Este concepto surge del hecho que muchos problemas del mundo real (análisis de redes, modelos químicos, teledetección, entre otros) han sido abordados con estrategias basadas en grafos, en las que el problema es transformado en términos de vértices y aristas. Con esto en mente, las propuestas realizadas en esta tesis se basan en considerar las instancias contenidas en la clase más representada como un grafo completo, al cual bajo criterios guiados de reducción se obtiene un subconjunto reducido de instancias representativas de dicha clase.

Para el escenario de estudio en el que el conjunto de datos puede contener además del desbalance de clases, traslape de clases y/o ruido, las propuestas incluyen la utilización de algoritmos de *clustering* como estrategia de limpieza. Es sabido que este tipo de algoritmos son usados para agrupar las instancias de acuerdo a características semejantes; no obstante, en la propuesta realizada se

aprovecha la capacidad que tienen para discriminar instancias consideradas como ruido. De este modo, la aplicación de un algoritmo de *clustering* es ejecutado previo al tratamiento de desbalance de clases.

Como parte inicial de un estudio que da continuidad a las propuestas realizadas en esta tesis, y dado el constante crecimiento de datos en contextos de *Big Data*, en la parte final de la tesis se presenta un algoritmo basado en grafos como una primera aproximación al tratamiento de desbalance de clases para grandes volúmenes de datos.

Para validar las propuestas antes mencionadas, se ejecutaron una serie de experimentos en conjuntos de datos reales y sintéticos obtenidos del repositorio KEEL, los cuales cuentan con una amplia variedad entre la cantidad de instancias y el grado de desbalance. En la experimentación se consideraron 16 métodos de bajo-muestreo utilizados en el estado del arte, de los cuales seis son basados en vecindad, cuatro del enfoque de ensembles, cinco basados en *clustering* y uno con enfoque evolutivo.

Los resultados experimentales permitieron observar que las propuestas basadas en grafos presentan los mejores porcentajes de rendimiento frente a otros métodos para el tratamiento de desbalance de clases. En tanto que la incorporación de un algoritmo de *clustering* para la limpieza del conjunto de datos, permite solventar de manera adecuada problemas de ruido y/o traslape de clases adicionales al desbalance de clases que es abordado de manera favorable con grafos.

Finalmente, se utilizó el test de Wilcoxon para descubrir diferencias estadísticamente significativas entre cada par de algoritmos. Este estadístico ordena las diferencias del rendimiento de dos algoritmos para cada conjunto de datos, ignorando los signos, y compara los rangos para las diferencias positivas y negativas. En resumen, este test permite concluir que las propuestas basadas en grafos son competitivas cuando se comparan con otras técnicas basadas en ensembles y *clustering* para el tratamiento del desbalance de clases. Adicionalmente, las soluciones basadas en grafos y *clustering* para abordar problemas con presencia de traslape de clases y/o ruido presentan resultados uniformemente competitivos.

Abstract

Nowadays, knowledge extraction from data is an essential task for decision-making in many areas. However, the data sets commonly present some negative problems (complexities) that decrease the performance in the knowledge extraction process. The imbalanced distribution of data between classes and the presence of noise and/or class overlap are data intrinsic characteristics that frequently decrease the performance of the knowledge extraction because data are assumed to keep a uniform distribution and free from any other problem.

All these issues have been studied in Pattern Recognition and Data Mining, because of their impact on the performance of the learning models. Thus this Ph.D. thesis addresses class imbalance, class overlap and/or noise through techniques that reduce and clean the most represented class.

Among the solutions to handle with the class imbalance problem, new algorithms based on graphs are proposed. This idea arises from the fact that many real-world problems (network analysis, chemical models, remote sensing, among others) have been tackled by using graph-based strategies, in which the problem is transformed in terms of vertices and edges. Keeping this in mind, the proposals presented in this Ph.D. thesis consider the most represented class as as a complete graph in such a way that a representative subset of majority class instances is obtained through reduction criteria.

Regarding the data sets with class imbalance and class overlap and/or noise, the proposals include the use of clustering algorithms as a cleaning strategy. It is well known that these algorithms are used to group instances according to similar characteristics; however, the proposal here presented makes use of their ability to detect noisy instances. By this, the application of a clustering algorithm is carried out before facing the class imbalance.

As a further extension to the proposals presented in this Ph.D. thesis and due to the growing interest in Big Data problems, the last part of this report introduces a graph-based algorithm to handle class imbalance in large-scale data sets.

In order to validate the aforementioned proposals, a series of experiments were carried out on real and synthetic data sets from the KEEL repository, which have an extensive variety in the number of instances and imbalance ratio. The experimental results allowed to determinate that proposals based on graphs have the best performance compared with other methods to face the class imbalance problem,while the inclusion of a clustering algorithm to clean the data set allows to adequately solve noise and class overlap when presented together with class overlap, which is rightly faced by using graphs.

Finally, the Wilcoxon's paired signed-rank test was used to find out statistically significant differences between each pair of algorithms. This statistic ranks the differences in performances of two algorithms for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences. In summary, this test allows to conclude that the graph-based proposals are competitive when compared with other techniques based on ensembles and clustering to face the class imbalance problem. In addition, the solutions based on graphs and clustering to tackle problems with class overlap and/or noise showed evenly competitive results.

Índice general

Índice de figuras

Página

Índice de tablas

Página

Índice de Algoritmos

Nomenclatura

BC	*Balance Cascade.*
CBU	*Cluster-based Undersampling.*
CD	*Conjunto de datos.*
CNN	*Condensado de Hart.*
COSS	*ClusterOSS.*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise.*
DMBS	*Data Base Managed System.*
EE	*Easy Esemble.*
ENN	*Edición de Wilson.*
fCBUS	*Fast-CBUS.*
Gmean	*Media Geométrica.*
FN	*Falsos Negativos- False Negative.*
FP	*Falsos Positivos- False Positive.*
G	*Grafo.*
HDFS	*Hadoop Distributed File System.*
IR	*Imbalance Ratio- Grado de Desbalance .*
KDD	*Kwoledge Discovery in Database.*
MST	*Minimun Spanning Tree.*
OSS	*Selección Unilateral.*
RBt	*RUSBoost.*
RDD	*Resilient Distributed Dataset.*
ROS	*Random Over-sampling.*
RUS	*Random Under-sampling.*
TL	*Tomek link.*
TN	*Verdaderos Negativos- True Negative.*
TP	*Verdaderos Positivos- True Positive.*
TNR	*Tasa de Verdaderos Negativos- True Negative Rate- Especificidad.*
TPR	*Tasa de Verdaderos Positivos- True positive Rate- Sensibilidad.*
SMOTE	*Synthetic Minority Oversampling TEchnique.*

Parte I

Introducción y Sustento Teórico

Capítulo 1
Introducción

Dado que actualmente la disponibilidad de datos y el uso generalizado de técnicas de extracción de conocimiento son de suma importancia tanto en ámbitos públicos como privados, el desarrollo de múltiples soluciones para el tratamiento de los datos exige entender la naturaleza intrínseca de éstos. En este sentido, es imposible pensar que los conjuntos de datos se encuentren totalmente libres de ciertas complejidades que pueden afectar a la extracción de conocimiento y, en consecuencia, se hace necesaria la utilización de técnicas que filtren o corrijan esas complejidades de tal modo que los datos valiosos sean aprovechados en el proceso de obtención del conocimiento [1].

Una de estas complejidades de los datos se refiere al problema del desbalance entre clases, el cual ocurre cuando en un conjunto de datos se tiene una desproporción de instancias a lo largo de las clases. Así, cuando una clase se encuentra mínimamente representada en comparación con el resto de las clases, se le asigna el nombre de clase minoritaria, mientras que la clase mayormente representada se conoce como clase mayoritaria. Esta complejidad de los datos ha sido ampliamente estudiada debido a su presencia en diversos problemas del mundo real, tales como financieros, diagnósticos médicos, detección de anomalías, reconocimiento facial, entre otros [2]. Adicionalmente, la desproporción entre el número de instancias de clases mayoritaria y minoritaria da lugar a un importante sesgo en favor de la clase mayoritaria al utilizar los modelos de aprendizaje y, como resultado, un pobre rendimiento a la hora de identificar muestras de la clase minoritaria.

Diversos estudios [3, 4, 5] aseguran que el problema del desbalance entre clases por si sólo no tiene un impacto significativo sobre el modelo de aprendizaje o clasificación, sino que es la presencia de éste junto con otras complejidades como el ruido y/o el solapamiento de clases lo que provoca una dramática disminución del rendimiento. El solapamiento o traslape de clases hace referencia a regiones ambiguas del espacio de características donde la probabilidad de las clases es aproximadamente igual, lo que en general dificulta el proceso de aprendizaje y puede dar lugar a clasificaciones erróneas. Por otro lado, el ruido implica instancias mal etiquetadas, que de igual manera puede conducir a decisiones de

clasificación incorrectas. Ambas complejidades han sido abordadas habitualmente mediante técnicas de filtrado de datos, cuyo objetivo es identificar y eliminar datos mal etiquetados y también remover posibles solapamientos.

Para el tratamiento del desbalance entre clases se han desarrollado diversas estrategias que pueden ser agrupadas en cuatro categorías [6]: *métodos a nivel de algoritmo*, que consisten en introducir un sesgo en los clasificadores para compensar el desbalance; *métodos a nivel de datos*, que consisten en disminuir o incrementar el número de instancias para reducir el nivel de desbalance; *métodos sensibles al costo*, que incorporan costos de clasificación errónea en el proceso de clasificación; y *métodos basados en ensembles*, que consisten en la combinación de alguna de las técnicas anteriores con el uso de ensembles (combinación de clasificadores).

Es esencial considerar que el resultado de la obtención del conocimiento dependerá de la calidad del conjunto de datos. En consecuencia, el tratamiento de complejidades que se derivan de esto, tal como el desbalance de clases, instancias incorrectamente clasificadas, datos ausentes, entre otros, requieren de soluciones que no conlleven pérdida de conocimiento. En este sentido, esta tesis se centra en el desarrollo de nuevos algoritmos que obtengan subconjuntos representativos del conjunto original por medio de técnicas que abstraigan en términos de vértices y aristas la representación del conjunto de clase mayoritaria, de tal forma que se aborde el tratamiento del desbalance entre clases. Por otra parte, para ofrecer una alternativa de solución a la presencia adicional de ruido y/o solapamiento de clases, se incorpora el uso de algoritmos de *clustering* por su capacidad para filtrar instancias consideradas como ruido.

Por lo que se acaba de exponer, en esta tesis se busca aprovechar el funcionamiento de los métodos propios de la teoría de grafos, por medio de la representación de un grafo, y de algoritmos de *clustering* para atender de forma eficiente el procesamiento de datos y afrontar los problemas del desequilibrio entre clases, del solapamiento de clases y del ruido.

1.1. Justificación

Suponer que los conjuntos de datos se encuentran bajo una distribución de clases equilibrada puede derivar en una pérdida de conocimiento [4]. Aunado a este problema, se pueden encontrar datos que aporten un bajo beneficio al cocimiento, como por ejemplo, aquellos que se encuentren mal etiquetados o inclusive estén incompletos o sean inconsistentes. Todos estos problemas se encuentran en temas tales como análisis de imágenes satelitales, identificación de proteínas, reconocimiento de células, diagnósticos médicos, análisis de redes, entre otros [6], por lo que es indispensable proponer estrategias que se encarguen de enfrentar

las complejidades que los datos puedan tener.

Los conjuntos de datos que se encuentran inmersos en estas complejidades deterioran el rendimiento de los modelos de aprendizaje o clasificación, debido principalmente a que los clasificadores asumen que los conjuntos de datos cuentan con una distribución uniforme de las clases y libres de ruido. La presencia tanto de ruido como de desbalance de clases puede afectar significativamente el proceso de clasificación, dado que el incorrecto etiquetado de la clase minoritaria a la mayoritaria aumenta la relación de desequilibrio, en tanto que el incorrecto etiquetado de la clase mayoritaria a la minoritaria produce una dificultad mayor en la identificación de instancias de clase minoritaria. Esto se resume en que los clasificadores se vuelven incapaces de distinguir instancias entre las diferentes clases.

En los últimos años, el uso de algoritmos de *clustering* ha tenido popularidad para solventar las complejidades de los datos, ya que si bien estos algoritmos agrupan elementos con características similares, de manera especifica para el tratamiento del desequilibrio de clases, permite reducir el riesgo de eliminar datos útiles.

Por otro lado, la teoría de grafos ha sido usada de manera predilecta para generar soluciones en problemas de optimización del mundo real dado que permite representar un problema en términos de vértices y aristas para obtener soluciones adecuadas [7, 8, 9, 10]. Por ejemplo, en química, un grafo puede representar la topología de una molécula, en física se puede utilizar para describir el grado de estabilidad termodinámica, en ingeniería eléctrica la teoría de grafos se aplica en la configuración de antenas y sus frecuencias, en áreas urbanas para la planificación en programar rutas de autobús o semáforos, entre otros. En el área de ciencia de datos, la teoría de grafos ha sido usada para extraer conocimiento de representaciones basadas en grafos [11, 12, 13].

1.2. Hipótesis

Con el uso de teoría de grafos y algoritmos de *clustering* se podrá realizar el preprocesado de datos para afrontar problemas de desbalance de clases, el traslape de clases y ruido que se encuentran presentes en los conjuntos de datos, con índices de precisión iguales o superiores a los que reportan los algoritmos más utilizados en el estado del arte.

1.3. Objetivos de la tesis

El objetivo general se centra en proponer nuevos algoritmos basados en grafos y *clustering* atendiendo las complejidades de los datos y, más específicamente, para el tratamiento del desbalance, el solapamiento de clases y el ruido. Para este fin, se establecen los siguientes objetivos específicos:

1. Analizar y desarrollar algoritmos para el tratamiento de la distribución de clases en los conjuntos de datos, donde el desbalance afecta al rendimiento del clasificador; en consecuencia, se realiza una aportación de algoritmos basados en grafos que permitan tratar dicho problema.

2. Analizar y desarrollar algoritmos para el tratamiento del solapamiento de clases y del ruido en conjuntos de datos; dado que el desbalance de clases en algunos casos no se presenta sólo, se requiere de técnicas que adicionalmente limpien los datos de tal modo que se aborde una nueva solución basada en *clustering* como estrategia de limpieza de los datos.

3. Analizar el comportamiento de las propuestas para el manejo de complejidades de datos utilizando conjuntos de datos de repositorios conocidos tales como UCI Machine Learning Database Repository o KEEL, mediante métodos de evaluación comúnmente aceptados por la comunidad de *Machine Learning*, como las matrices de confusión y el análisis de significancia estadística, entre otros.

1.4. Estructura de la tesis

La tesis está organizada en cuatro partes:

1. Parte I Introducción y Sustento Teórico, formada por el primer capítulo donde se incluye la propuesta de investigación, contextualizando la necesidad de desarrollar nuevos algoritmos para el tratamiento de complejidades de los datos, mientras que el marco teórico y estado del arte se abordan en el Capítulo 2.

2. Parte II Propuestas y Metodología, integrada por el Capítulo 3 donde se presentan las propuestas para el tratamiento de desbalance de clases, traslape de clases y ruido, en tanto que en el Capítulo 4 se detalla la metodología a seguir.

3. Parte III Resultados Experimentales, distribuida en cuatro capítulos, donde el Capítulo 5 presenta el escenario de pruebas y las validaciones realizadas a

las propuestas basadas en grafos para el tratamiento de desbalance de clases, mientras que el Capítulo 6 consiste en los resultados obtenidos por DBSCAN como estrategia de limpieza del conjunto de datos. El Capítulo 7 presenta los resultados obtenidos por las propuestas de tratamiento de desbalance de clases, traslape de clases y/o ruido, en tanto que en el Capítulo 8, dado el gran volumen de datos que se generan, se aborda una aproximación a *Big Data*.

4. Parte IV Conclusiones, está integrada por el Capítulo 7, donde las principales conclusiones de investigación y líneas abiertas de estudio son descritas.

De forma adicional, se incluyen cuatro apéndices, en cuyos anexos se puede encontrar en resumen el desempeño de los clasificadores en términos de media geométrica de las propuestas de esta tesis comparadas con técnicas bien conocidas del estado del arte y un anexo de los resultados de rendimiento por clase para las técnicas del estado del arte.

Capítulo 2

Marco teórico y estado del arte

Dos áreas de la Inteligencia Artificial son de interés en esta tesis: *Machine Learning* o Aprendizaje Automático y *Pattern Recognition* o Reconocimiento de Patrones. Por un lado, el Aprendizaje Automático es un área del conocimiento que provee algoritmos que extraen automáticamente conocimiento a partir de cúmulos de datos, mientras que el Reconocimiento de Patrones es el proceso de asignar a un objeto o fenómeno físico (instancia o patrón) una clase o categoría por medio de reglas de decisión automática que transformen medidas sobre un modelo en asignaciones a clases. En este sentido, tanto los algoritmos de Aprendizaje Automático como los de Reconocimiento de Patrones se utilizan comúnmente para cubrir este objetivo.

En Aprendizaje Automático y Reconocimiento de Patrones, un **modelo** es una representación de una instancia. Sea X una instancia, compuesta por d características o atributos $(x_1, x_2, \ldots, x_d)$ que dan lugar a la representación de un objeto del mundo real. El **espacio de representación** es el conjunto de todas las representaciones posibles para un cierto problema, es decir, el universo de trabajo [14].

Un clasificador ζ en términos de C **funciones discriminantes**, $D_i(X)$ se expresa como $\zeta(X) = \omega_i \iff D_i(X) > D_j(X)\ \forall j \neq i$, donde $i, j = 1, ..., C$, por lo que un clasificador asigna a X la clase cuya función discriminate $D_i(X)$ sea mayor [15].

La **frontera de decisión** es la zona en el espacio de características donde $D_i(X) = D_j(X)$, es decir, particiones con similar probabilidad de pertenencia a más de una clase.

Tradicionalmente, los algoritmos de Aprendizaje Automático y Reconocimiento de Patrones requieren de dos etapas en su funcionamiento: aprendizaje y clasificación, las cuales pueden realizarse de forma disjunta. Por un lado, para realizar el proceso de aprendizaje, dependiendo del conjunto de datos y del tipo de análisis, en la literatura se consideran básicamente cuatro enfoques [16]: supervisado, no supervisado, semi-supervisado y aprendizaje profundo. En el primero, a partir de un conjunto de instancias, el modelo aprende una relación entrada-salida (ver

Sección 2.1), mientras que el aprendizaje no supervisado parte de un conjunto de instancias de solo entrada para determinar una "estructura"que las modele (ver Sección 2.2).

Por otro lado, el aprendizaje semi-supervisado se puede realizar de dos maneras principalmente; una de ellas consiste en iniciar con aprendizaje supervisado, utilizando una cantidad mínima de instancias que son representativas en el espacio de decisión para, posteriormente, realizar aprendizaje no supervisado [17], mientras que la segunda estrategia se realiza de manera inversa. Por último, el aprendizaje profundo (*Deep Learning*) sigue un proceso de aprendizaje jerárquico de los datos y es orientado principalmente a un enfoque no supervisado [18]. A diferencia de los otros enfoques de aprendizaje, en éste, tanto la extracción de características como la clasificación de instancias se realizan directamente a partir de los datos, sin la supervisión de un experto humano.

Para el logro de su cometido, en Aprendizaje Automático y Reconocimiento de Formas, podemos encontrar una amplia gama de algoritmos, algunos orientados a tareas de clasificación, regresión o agrupamiento. El contexto de esta investigación se centra en los algoritmos de aprendizaje supervisado y no supervisado, los cuales son abordados en detalle en las Secciones 2.1 y 2.2.

2.1. Aprendizaje supervisado

También conocido como aprendizaje inductivo o métodos de predicción. Este tipo de algoritmos, durante el proceso de aprendizaje utilizan datos de los cuales se conoce a priori la salida deseada o esperada por el clasificador. Es decir, los conjuntos de instancias usadas se presentan al sistema ya etiquetados por un experto humano en el área de estudio [19]. En general, este tipo de métodos intentan descubrir relaciones entre los atributos de entrada (características de una instancia) y un atributo destino (clase), esta relación se representa en un modelo.

En un escenario típico de aprendizaje supervisado, el conjunto de datos se divide en dos subconjuntos, conservando la distribución a priori de las clases, utilizando uno de ellos para entrenamiento del modelo o clasificador y el otro para fines de prueba, con el cual se validará la precisión del modelo al clasificar nuevos casos. En general, un conjunto de datos (CD) está formado por N-instancias con d-características, cada instancia p_N es una tupla $(f_{N,1}, f_{N,2}, \ldots, f_{N,d}, C)$, donde, $f_{N,d}$ es el valor de la d-íesima característica de una instancia p_N. Esta instancia pertenece a una clase C [20]. Los algoritmos de aprendizaje supervisado utilizados en esta tesis se describen en las Secciones 2.1.1, 2.1.2 y 2.1.3.

2.1.1. Aprendizaje basado en instancias

Los tipos de algoritmos basados en instancias se encargan de comparar las nuevas instancias del problema con instancias que ya se encuentran clasificadas y almacenadas en memoria, es decir, produce una etiqueta/predicción de clase basada en la similitud de sus vecinos más cercanos en el conjunto de entrenamiento.

El algoritmo más común y que será usado en este trabajo de tesis es el algoritmo de vecinos más próximos, kNN (k-Nearest Neighbors), ya que es un algoritmo de aprendizaje no paramétrico, lo que significa que no hace ninguna suposición sobre la distribución de los datos. La asignación de clase de una instancia se basa en la pertenencia que la mayoría de sus k vecinos pertenecen a cierta clase. para determinar la cercanía de instancias se aplica la métrica de distancia, la cual, típicamente usada es la distancia Euclídea [21].

2.1.2. Árbol de decisión

Un árbol de decisión, en específico el algoritmo J48, realiza particiones al conjunto de datos repetidamente, con el objetivo de que cada partición contenga elementos de una sola clase. Para realizar el proceso de partición, se prueban todos los posibles valores de las instancias en cada atributo de estas, posteriormente se selecciona la mejor partición de acuerdo a un determinado criterio, comúnmente el de mayor ganancia de información, el cual se entiende como la medida de relevancia que tiene un atributo dentro del conjunto de datos [22].

2.1.3. Aprendizaje lineal

Los modelos de aprendizaje lineal tienen como objetivo asignar a una instancia una clase tomando una decisión basado en el valor cercano a una combinación lineal de las características, las cuales están representadas en una estructura de datos denominada vector. El método comúnmente usado son las máquinas de vectores de soporte (*Support Vector Machine, SVM*) es un método que crea una línea o un hiperplano que separa los datos en clases. SVM es un algoritmo que toma los datos como entrada y genera de ser posible una línea que separe las clases, para determinar una línea ideal que separe el conjunto, se determina por medio de los puntos más cercanos a una línea, estos puntos se denominan vectores de apoyo, posteriormente se calculan las distancias entre la línea y los vectores de soporte, dicha distancia se denomina margen, de tal modo que este margen se maximice para determinar la línea o hiperplano óptimo. Cuando los problemas no pueden ser separados linealmente, SVM agrega una dimensión a la representación lineal para determinar un hiperplano que logre separar las clases [23].

2.2. Aprendizaje no supervisado

El aprendizaje no supervisado, también conocido como aprendizaje deductivo, permite la construcción libre del agrupamiento de las instancias basándose únicamente en relaciones de similitud entre las instancias de su grupo y que, al mismo tiempo, sean diferenciables de los demás grupos [24].

Existen varios tipos de algoritmos dentro de este paradigma de aprendizaje: asociación, modelos de variables latentes y de *clustering* [25]. Los algoritmos de asociación encuentran relaciones o dependencias en el conjunto de datos por medio de reglas asociativas. Por su parte, los modelos de variables latentes se basan en la relación estadística de un conjunto de variables directamente medibles con un conjunto de variables que necesita una variable medible asignada como indicador para probar si está o no presente. Por último, los algoritmos de *clustering* son los de interés en esta tesis, motivo por el cual se hace especial énfasis en ellos.

Dado N instancias, $x_1, x_2, ..., x_N$ contenidas en un espacio S, el proceso de *clustering* formalmente se define como: buscar las regiones $S_1, S_2, ..., S_c$ tales que cada x_i, $i = 1, 2, ..., N$ quede en alguna de estas regiones y x_i no quede simultáneamente en dos regiones [26], es decir:

$$S_1 \cup S_2 \cup S_3 \cup \cdots \cup S_c = S \quad S_i \cap S_j = \emptyset \quad \forall i \neq j \tag{2.1}$$

Los algoritmos de *clustering* se pueden diferenciar en dos tipos: directos (constructivos) o indirectos (de optimización), de acuerdo a si cuentan o no con una función criterio usada para realizar el proceso de *clustering* [26]. Los algoritmos agrupan por asociación natural de similitudes utilizando alguna métrica como la distancia Euclídea o la distancia de Mahalanobis, entre otras.

La Figura 2.1 muestra la gran variedad de algoritmos de aprendizaje no supervisado y supervisado que se puede encontrar en la literatura.

Existen diferentes tipos de métodos para *clustering*, entre los cuales se incluyen los algoritmos de partición, los métodos jerárquicos y las técnicas basadas en densidad.

Los métodos basados en particiones son aquellos en los que los datos se dividen en subconjuntos que no se superponen, de modo que cada instancia se encuentra exactamente en un subconjunto. Aunque existen muchos algoritmos que pertenecen a esta categoría, el más popular y ampliamente utilizado es el algoritmo K-Means. Pese a los buenos resultados que reporta en la literatura, el algoritmo requiere como dato de entrada el número de grupos en los cuales se va a agrupar el conjunto de datos [27]. Este requisito es una de sus principales desventajas, ya que obliga a que se tenga conocimiento a priori de la distribución estimada de los grupos, aspecto que en la vida real es difícil de saber.

Los algoritmos de agrupación jerárquica son abordados desde dos perspecti-

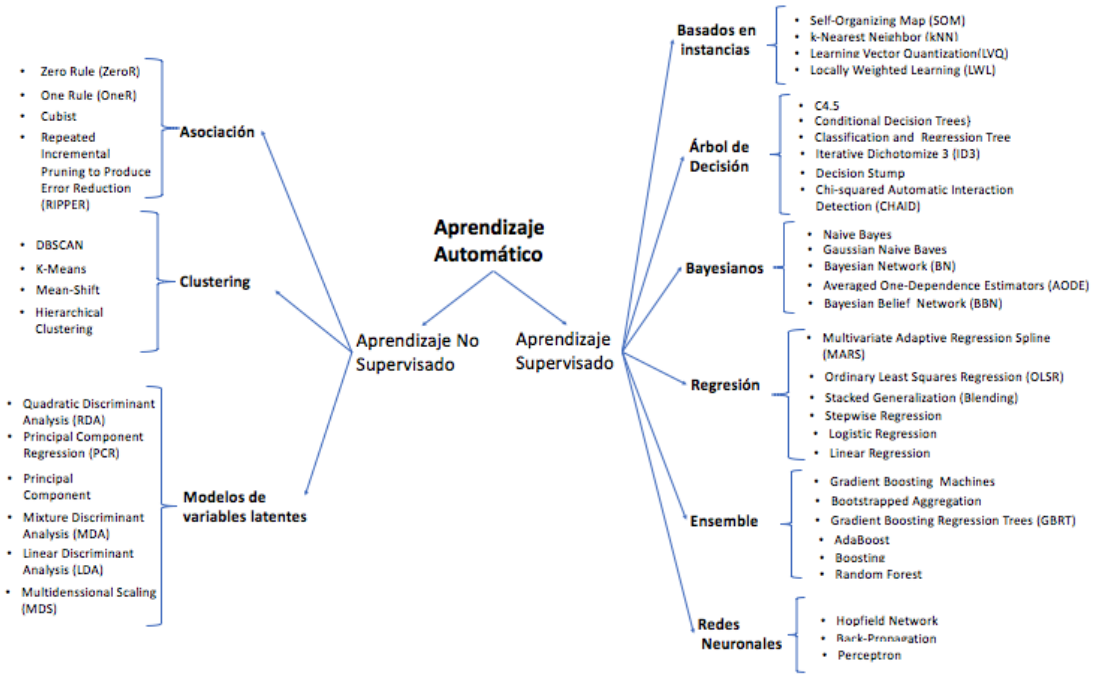


Figura 2.1: Ejemplos de algoritmos de Aprendizaje Automático

vas: aglomerativos y divisivos. Los enfoques aglomerativos o de manera ascendente tratan cada instancia como un solo grupo desde el principio para, posteriormente, juntar sucesivamente pares de grupos hasta que todos los grupos se hayan fusionado en uno. Por el contrario, los métodos divisivos inician asignando un grupo a cada instancia para, posteriormente, realizar divisiones de manera recursiva a medida que se desciende en la jerarquía.

Por su parte, los algoritmos basados en la densidad del conjunto identifican grupos distintivos en los datos basados en la idea de que un grupo en un espacio de datos es una región continua de instancias, separada de otros grupos similares por regiones contiguas de baja densidad, comúnmente consideradas como *ruido*. Dentro de esta categoría, la técnica más popular es el algoritmo DBSCAN, el cual no requiere de un número predefinido de grupos.

Por esto último y por apegarse a los objetivos planteados en esta tesis, se utilizó el algoritmo DBSCAN que no requiere semilla y ha demostrado su robustez en datos con distribución compleja.

2.2.1. DBSCAN

DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [28] es un algoritmo de *clustering* que se basa en la densidad de los datos para encon-

trar el número de grupos (*clusters*) de acuerdo a la distribución en el espacio de representación. Dentro de las fortalezas que presenta DBSCAN, se puede destacar el hecho de encontrar clusters que no son linealmente separables, la capacidad de eliminar instancias ruidosas y no necesitar asumir un número fijo de grupos. Por otro lado, dentro de las desventajas principales, se encuentra la dificultad de determinar una adecuada separación de clusters.

Para comprender mejor el funcionamiento del algoritmo, dado un conjunto de datos, se tienen las siguientes definiciones:

Definición 1 *Densidad de una instancia* X. Es el número de instancias dentro de un círculo de radio ϵ desde una instancia X.
Definición 2 *Región de densidad*. Para cada instancia en el cluster o grupo, la circunferencia con radio ϵ contiene al menos un número mínimo de instancias, denotado como $minPts$.
Definición 3 *Medida de distancia*. Es la distancia entre cualquier $X, Y \in CD$ denotado como $dist(X, Y)$.
Definición 4 ϵ*-vecindad de una instancia*. El ϵ vecindad de X es un subconjunto de CD, tal que $N_\epsilon(X) = \{Y \in CD | dist(X, Y) \leq \epsilon\}$
Definición 5 *Instancia núcleo*. La instancia X es una instancia núcleo, si $|N_\epsilon(X)| \geq minPts$.
Definición 6 *Instancia borde*. Una instancia borde es aquella instancia X que tiene menos de $minPts$ dentro de su $N_\epsilon(X)$, pero se encuentra cerca de otro punto central.
Definición 7 *Ruido*. Es cualquier instancia que no es ni núcleo ni instancia borde.
Definición 8 *Densidad-alcanzable*. Dado $X, Y \in CD$, si $Y \in N_\epsilon(X)$ y X es un punto central, entonces Y es de densidad accesible desde X.
Definición 9 *Densidad-conectada*. Dados $X, Y \in CD$, las instancias X, Y están conectadas por densidad si hay una instancia $Z \in CD$ de modo que tanto X como Y sean accesibles por densidad desde Z.

El Algoritmo 2.1 describe el proceso de DBSCAN, el cual comienza con una instancia que no ha sido visitada. La vecindad de esta instancia se extrae usando una distancia o radio ϵ, es decir, todos las instancias que estén dentro de ϵ son instancias vecinas. Si existen $minPts$ dentro de la vecindad, la instancia actual se convierte en la primer instancia de un nuevo agrupamiento: de lo contrario, la instancia se etiquetará como ruido y es marcada como visitada. Para la primer instancia en el nuevo grupo, las instancias vecinas también se convierten en parte del mismo grupo. Se expande el grupo mediante la comprobación de todas las nuevas instancias revisando si estas también cumplen con los parámetros libres.

Algoritmo 2.1 DBSCAN

Entrada: Conjunto de datos $CD = \{X_1, X_2, \ldots, X_N\}$, ϵ, $minPts$
Salida: Conjunto de datos agrupado

```
1: C ← ∅
2: for cada instancia no visitada X_i in CD do
3:     Marcar X_i como visitada
4:     nbhdP ← N_ε(X_i)
5:     if |nbhdP| < minPts then
6:         Marcar X_i como ruido
7:     else
8:         C= siguiente cluster
9:         C ← X_i
10:        for cada instancia X' in nbhdP do
11:            if X' no está visitada then
12:                Marcar X' como visitada
13:                nbhdP' ← vecindad(ε, X')
14:                if |nbhdP'| ≥ minPts then
15:                    nbhdP = nbhdP unir con nbhdP'
16:                end if
17:            end if
18:            if X' aún no es miembro de cualquier cluster then
19:                C ← X'
20:            end if
21:        end for
22:    end if
23: end for
```

Como ya se ha dicho, DBSCAN depende de dos parámetros libres. Sin embargo, en algunos estudios se ha visto la manera de determinar dichos parámetros dependiendo del conjunto de datos. Abir y Zied [29] proponen las siguientes ecuaciones para determinar $minPts$ y ϵ, basados en los conjuntos de datos, así como en el proceso de realizar en una primera aproximación la generación de *clusters* por el algoritmo *Gaussian Means*, el cual determina un número c por medio de una prueba estadística con la idea de considerar de que el conjunto de datos sigue una distribución gaussiana. Para determinar el cálculo local de $MinPts$, se usa la Ecuación 2.2.

$$MinPts = \frac{\pi \times r_j^2}{TotalVolume_j} \times N_j \tag{2.2}$$

donde N_j es el número de instancias en el *cluster* j, cuyo centro es M_j, Además $TotalVolume_j$ representa el volumen total del *cluster* j y se obtiene de la Ecuación 2.3.

$$TotalVolume = \frac{4}{3} \times \pi \times r^3 \tag{2.3}$$

Por otro lado, para calcular r_j^2 que en general, será el valor de ϵ, se selecciona el valor mínimo obtenido de cada j cluster:

$$r = \sqrt{\frac{\sum_N^{i=1} distancia^2(M_j, x_i j)}{N}} \tag{2.4}$$

donde N es el número de instancias del cluster j y *distancia* hace referencia a la distancia Euclídea.

2.3. Complejidades de los datos

Para las áreas de Reconocimiento de Patrones y Aprendizaje Automático, el rendimiento de un clasificador se ve influenciado por la calidad del conjunto de datos que se utilice [6]. En este sentido, las complejidades de los datos son los problemas que los conjuntos de datos tienen y que, por lo tanto, afectan al rendimiento del clasificador. En esta sección, se abordarán las características intrínsecas de los datos y los métodos de preprocesamiento usados para tratarlos. Por último, teniendo en cuenta que esta tesis se centra en el desbalance de clases, se describen las principales técnicas para el tratamiento de este problema.

2.3.1. Características intrínsecas de los datos

En problemas de la vida real, por diferentes cuestiones, los conjuntos de datos cuentan con peculiaridades que en muchas ocasiones afectan de forma negativa al rendimiento de los algoritmos de extracción del conocimiento.

Una clasificación de las complejidades de los datos se pueden dar en *irregularidades de datos basadas en distribución* e *irregularidades de datos basadas en características* [1]. Dentro del primero grupo se integran problemas tales como el desequilibrio o desbalance entre clases, el solapamiento de clases, los datos atípicos y los datos ruidosos. En el caso de las irregularidades de datos basadas en características, se incluyen complejidades tales como las características ausentes, ruidosas, irrelevantes y redundantes. A continuación, se describen con más detalle algunas de las complejidades que acabamos de mencionar.

- Solapamiento de clases: Se presenta cuando las instancias tienen atributos poco discriminantes y en consecuencia, la frontera de decisión no se encuentra bien definida entre las clases. Para un problema de dos clases, el solapamiento podría observarse en la Figura 2.2a [30], donde el clasificador se ve afectado por la dificultad en la identificación de la frontera de decisión, donde la probabilidad previa de las clases es aproximadamente igual, dicho de otra manera, todas las clases tienen una cantidad similar de datos, en consecuencia, la mayoría de los clasificadores son propensos a producir una clasificación errónea.

- Alta dimensionalidad: Se presenta cuando un conjunto de datos tiene un número significativamente elevado de características por instancia (Figura 2.2b); el número de características puede exceder el número de instancias del conjunto de datos. Por ejemplo, los microarrays que miden la expresión genética pueden tener cientos de instancias, pero cada instancia puede contener decenas de miles de genes [31]. En términos de procesamiento, se busca realizar disminución de

características para reducir de igual manera el costo computacional requerido por el clasificador.

- Instancias atípicas o ruidosas: Las instancias atípicas son las que ya tienen definida una clase, pero son significativamente diferentes al resto de su misma clase. Por ejemplo, una persona sana, que no tiene ninguna lesión ni padece ninguna enfermedad y realiza con normalidad todas sus funciones, difiere del resto de las personas con respecto a tener pie plano (Figura 2.2c). Por otro lado, las instancias ruidosas son aquellas que tiene cierto parecido con instancias de otras clases. Por ejemplo, para el caso de personas identificadas con gastritis, los síntomas en mujeres, en algunos casos son parecidos a los de un embarazo, por lo que se llega a confundir el diagnostico [30].

- Desbalance de clases: Ocurre cuando una de las clases se encuentra en gran parte representada en comparación del resto de las clases, dicho de otro modo, es cuando la relación de instancias de una clase es significativamente alto en comparación de otras clases. Como ejemplo, se tiene un mayor número de personas sanas en relación con las personas enfermas, tal y como se muestra en la Figura 2.2d [31].

  Diversos estudios [32, 33, 34] han mostrando que la mayoría de los clasificadores tienen un bajo rendimiento en el reconocimiento de clases menos representadas, dado que existe una inclinación por la clasificación de clase con un mayor número de instancias, en otras palabras, el clasificador discrimina a instancias de una clase por ser menos representadas.

- Datos ausentes o perdidos: Un dato ausente o perdido se presenta cuando se carece de un valor para ciertos atributos, en algunas o todas las instancias. Esto puede incurrir en una mala inferencia de resultados. En análisis estadístico, la presencia de información faltante en un conjunto de datos conlleva inconvenientes, tales como pérdida de eficiencia, estimaciones sesgadas, así como limitar el proceso de análisis [35].

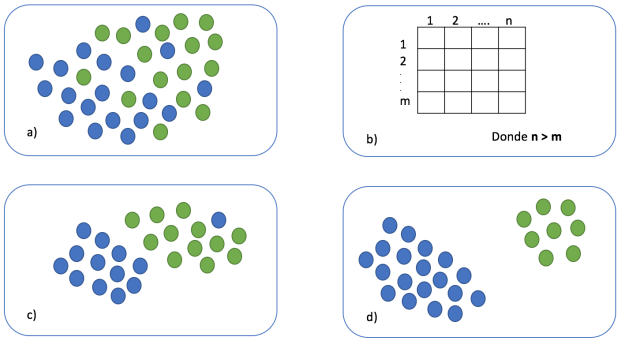


Figura 2.2: Complejidades de los datos: a) Solapamiento de clases, b) Alta dimensionalidad, c) Instancias atípicas, d) Desbalance de clases

2.3.2. Preprocesamiento

Para disminuir el efecto negativo que las complejidades de los datos tengan sobre el clasificador, es importante aplicar alguno de los métodos de preprocesado existentes en la literatura [31], de tal forma que la gestión de la calidad de los datos se asegure. Los algoritmos de preprocesado más utilizados se dividen en dos grandes categorías [36]:

- Preparación de los datos: Son técnicas que ajustan el conjunto de datos de manera apropiada para que algún algoritmo lo use, entre las cuales se destacan:
  - Limpieza de datos. La limpieza de datos es un proceso en el que se identifican datos redundantes, ruido, solapamiento y outliers en el conjunto de datos y busca reparar inconsistencias o conflictos entre datos. Este proceso puede ser una limpieza de datos independiente del domino o bien una limpieza específica del dominio [37]. En este proceso se realiza desde la limpieza de instancias duplicadas, así como también datos incompletos, hasta errores lógicos, problemas de ruido o eliminación de outliers.
  - Transformación de los datos: La transformación de los datos se utiliza como medio de consolidación de una manera apropiada para el análisis de los datos, debido a las peculiaridades que los clasificadores tienen para realizar el procesado. Por ejemplo, si un dato se encuentra de forma categórica y se desea usar una red neuronal, este tipo de dato tendrá que ser cambiado a numérico. Entre las estrategias más comunes se encuentran la normalización y discretización. La normalización es el procesos en el que los atributos se escalan para que éstos se encuentren dentro de un rango especifico, como por ejemplo de -1.0 a 1.0 o de 0 a 1.0. Por su parte, las técnicas de discretización pueden ser usadas para reducir el número de valores de atributo contínuo dividiendo los atributos en un rango de intervalos, donde los valores numéricos se pueden remplazar por etiquetas de intervalos o etiquetas conceptuales.
- Reducción de datos: Estos métodos buscan obtener una representación reducida de los datos sin comprometer la integridad del conjunto de datos original. Es común relacionar la reducción de datos tanto para el volumen como para la dimensionalidad. Algunas de las estrategias de reducción de instancias se encuentran la selección de instancias, que consiste en elegir un subconjunto del total de datos disponibles, los algoritmos seleccionan un subconjunto de instancias mediante el uso de algunas reglas y/o heurísticas. Mientras que las estrategias de selección de características o atributos, reduce el conjunto de datos al eliminar características redundantes, el objetivo es encontrar un conjunto mínimo de atributos para el cual la distribución original sea lo más cercana a la distribución del conjunto original.

2.4. Desbalance de clases

En general, cualquier conjunto de datos es susceptible de sufrir un desequilibrio o desbalance en la distribución de instancias entre las clases [6]. En problemas de dos clases, la clase que se encuentra menos representada se conoce como *minoritaria* o clase *positiva*, mientras que la clase más representada se denomina *mayoritaria* o clase *negativa*. En particular, este problema es parte fundamental del trabajo de esta tesis.

La importancia del tratamiento del desbalance de clases recae en la afectación que llega a tener sobre el rendimiento de los clasificadores tradicionales debido al sesgo que la clase mayoritaria tiene. En consecuencia, el bajo rendimiento se debe al menor número de instancias de la clase minoritaria, las cuales no pueden ser generalizadas por los clasificadores convencionales tales como los árboles de decisión o las redes neuronales [4] porque éstos asumen que los conjuntos de datos están distribuidos de manera uniforme. Como resultado de esta problemática, se han implementado multitud de técnicas para el tratamiento de desbalance de clases, las cuales usualmente se categorizan en cuatro grupos: métodos a nivel algorítmico, métodos a nivel de datos, métodos sensibles al costo y técnicas basadas en ensembles.

2.4.1. Métodos a nivel algorítmico

Las soluciones que se presentan a nivel algorítmico se enfocan en modificar el modelo de entrenamiento, es decir, modifican el algoritmo de aprendizaje del clasificador, sesgando internamente el proceso basado en la discriminación para compensar el desequilibrio de clase. Por lo tanto, las soluciones a nivel algorítmico no realizan cambios en la distribución de los datos, de esta manera pueden ser usados en diferentes conjuntos de datos desequilibrados, sin embargo, las soluciones dependerán de un clasificador determinado.

En este tipo de técnicas se debe comprender qué factores afectan el clasificador con datos desequilibrados, por lo que es necesario analizar por separado la influencia que cada factor tiene, así como su interacción. En el estado del arte, los árboles de decisión son comúnmente afectados por el desbalance de clases, dado que el sesgo a la clase mayoritaria es una consecuencia del criterio de división usado, por consiguiente, se deben estudiar y proponer soluciones en este sentido.

Las soluciones a nivel algorítmico no son tan usuales en el estado del arte, esto puede deberse a que son soluciones que pueden ser difíciles de diseñar e implementar por el nivel de comprensión del algoritmo de aprendizaje a usar, no obstante, se encuentran algunas soluciones. Por ejemplo, Lenca et al. [38] proponen como criterio de división en un árbol de decisión el uso de una entropía descentralizada con una distribución tomada a priori de la distribución de las clases o de la distribución que tenga en cuenta los costos de clasificación errónea, de tal forma que se permite incorporar una relación de desequilibrio en el procedimiento de inducción del árbol.

2.4.2. Métodos a nivel de datos

Las técnicas basadas a nivel de datos consisten en modificar el conjunto de datos desbalanceado, a través de algoritmos que equilibran o adecúan de mejor manera la distribución de los datos para ser usados posteriormente en algoritmos de aprendizaje. Dicho de otro modo, estas técnicas realizan algún tipo de procesamiento previo de los datos con el objetivo de reducir la relación de desequilibrio, conocidos como métodos de re-muestreo. Debido a la importancia que tienen en esta tesis estos métodos, se brinda una mayor explicación de los mismos.

Los métodos a nivel de datos como tratamiento de desbalance de clases se han posicionado como los más comunes, convirtiéndose en técnicas estándar, cuya ventaja es que son independientes del clasificador. En general, los métodos de re-muestro consisten en ajustar el tamaño del conjunto de datos para equilibrar la distribución de la clase, ya sea disminuyendo o aumentando el número de instancias, usualmente se clasifican en tres categorías:

- Bajo-muestreo (*Under-sampling*): Consiste en crear un subconjunto de datos por medio de la eliminación de instancias, usualmente tomadas de la clase mayoritaria, con el fin de reducir el tamaño del conjunto de datos original. En la literatura especializada, un ejemplo, es la técnica de eliminación de instancias aleatoriamente denominado bajo-muestro aleatorio (RUS).

- Sobre-muestreo (*Over-sampling*): Implica la creación o replicación de instancias del conjunto original, comúnmente de la clase minoritaria, una técnica usualmente usada se basa en una réplica aleatoria de instancias, denominado sobre-muestreo aleatorio (ROS).

- Métodos híbridos: Consiste en aplicar tanto técnicas de bajo-muestro como técnicas de sobre-muestreo al conjunto desbalanceado.

Es importante señalar que para técnicas aleatorias existen desventajas, como en ROS la principal desventaja es la posibilidad de descartar datos que sean potencialmente útiles, mientras que para RUS se aumenta la probabilidad de producir un sobreajuste en el clasificador (*overfitting*), dado que se realizan réplicas de instancias ya existentes.

Dada la importancia que tienen los métodos de tratamiento de desbalance a nivel de datos, en el Estado del Arte se precisan las estrategias propuestas bajo este enfoque.

2.4.3. Métodos sensibles al costo

Estas técnicas retoman principios de las orientadas a nivel de algoritmos, con la diferencia de que en lugar de usar una evaluación estándar basada en el error o pérdida de clasificación, en este tipo de técnicas se introduce un costo de clasificación errónea para minimizar el riesgo condicional. Dicho de otro modo, incorporan costos de clasificación errónea en los procesos de clasificación. Al penalizar los errores en algunas clases, aleja

las instancias de la frontera de decisión, en consecuencia manifiesta una mejor generalización del conjunto de datos y minimicen el número de predicciones incorrectas. La función de pérdida en datos desbalanceados se puede minimizar fácilmente centrándose en la clase mayoritaria y pasando por alto (o incluso ignorando por completo) la clase minoritaria.

En la literatura existen dos escenarios que vulneran a los clasificadores sensibles al costo: el costo asociado con las características y el costo asociado con las clases. El primero asume que la adquisición de una determinada característica está relacionada con un costo de prueba. Entiéndase este tipo de costo con respecto al valor en recursos de tiempo, económico o humano para generar las características del conjunto, por lo tanto el objetivo es crear un clasificador que obtenga un mejor rendimiento al usar características que se puedan obtener con el menor costo posible. Esto también puede implicar el uso de algoritmos de selección de características, no obstante, algunos clasificadores sensibles al costo como los árboles de decisión incorporan un procedimiento de optimización de costos.

Por último, el costo asociado con las clases, asume que cometer errores en instancias proviene de ciertas causas asociadas con las clases. En este sentido, las técnicas que se desarrollan tienen como objetivo hacer que el clasificador se centre en las clases que tienen un costo de riesgo alto [6].

2.4.4. Técnicas basadas en ensembles

Un *ensemble* es una estrategia de clasificación en la que un conjunto de clasificadores es visto como uno solo. La idea principal es mejorar el rendimiento de un solo clasificador al entrenar varios clasificadores diferentes y combinar sus resultados para obtener la decisión final. Para aplicar estos métodos a problemas de desbalance comúnmente es necesario combinarlos con alguna de otras técnicas mencionadas anteriormente.

En la literatura, es posible encontrar técnicas basadas en *bagging* o *boosting*. El primero es un modelo de aprendizaje que entrena a cada clasificador con diferentes réplicas extraídas aleatoriamente (con reemplazo) del conjunto de datos original. Mientras que *boosting*, es un modelo de remuestreo adaptativo y combinatorio, en contraste con *bagging*, este último solo reduce la varianza, mientras que *boosting* adicionalmente reduce el sesgo del clasificador. AdaBoost es el algoritmo de *boosting* más representativo [6].

2.5. Teoría de grafos

Los grafos han sido usados para modelar varios de los problemas del mundo real, tales como aplicaciones industriales, modelos químicos, redes sociales, sensores remotos, entre otros problemas, dado que para obtener su solución, el problema puede ser transformado en términos de vértices y aristas. En consecuencia, la teoría de grafos comienza ha ser usada en las áreas de Aprendizaje Automático y de Reconocimiento de Patrones para extraer conocimiento de representaciones dadas por un grafo.

Formalmente un grafo simple no dirigido se define como un par ordenado $G = (V, E)$, donde V es un conjunto de elementos denominados *vértices*, y E es un conjunto de pares no ordenados de vértices $\{v, u\}$ denominados *aristas* [39] (Figura 2.3).

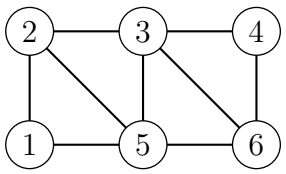


Figura 2.3: Grafo simple.

Sea $\psi_G(e)$ una función de incidencia que asocia con cada arista de G. Si e es una arista, y u, v son vértices tales que la función $\psi_G(e) = \{u, v\}$, entonces se dice que e es una unión de u y v, estos se conocen como *finales* [40].

Definición 10 *Vértices adyacentes.* Dos vértices v y u se denominan *adyacentes*, si hay una arista $\{v, u\} \in E$.

Definición 11 *vecindad.* La *vecindad* de un vértice v en un grafo $G = (V, E)$ es $N(v) = \{\forall u \in V \mid \{v, u\} \in E\}$, es decir, $N(v)$ es el conjunto de todos los vértices adyacentes a v sin el mismo. Por lo tanto, los vértices no vecinos a un vértice v son todos aquellos que no compartan arista con v.

Definición 12 *Grado de un vértice.* Dado un grafo $G = (V, E)$, el *grado* de un vértice $v \in V$, denotado como $\delta(v)$, es $|N(v)|$ (es decir, el número de aristas en las cuales v incide).

Definición 13 *Subgrafo.* Dado un subconjunto de vértices $S \subseteq V$, el subgrafo de G denotado como $G|S$ tiene un conjunto de vértices S y un conjunto de aristas tal que, $E(G|S) = \{\{u, v\} \in E : u, v \in S\}$. Por lo que, $G|S$ se denomina el *subgrafo de G inducido por S*. Se escribe $G - S$ para denotar el grafo $G|(V - S)$. El subgrafo inducido por $N(v)$ es denotado como $H(v) = G|N(v)$, el cual tienen al conjunto $N(v)$ como conjunto de vértices y todas las aristas sobre ellos.

Dado un subgrafo $H \subseteq G$, para cada vértice $u \in V(H)$, dado $\delta_H(u)$ el grado de u en un subgrafo inducido H de G, si $H = G$ entonces $\delta_G(u) = \delta(u)$ y $E_H(u) = \{\{u, v\} \in E(G) : v \in H\}$. De manera similar, $N_H(u)$ denota el conjunto de vértices de H adjacentes a u. Para cualquier subgrafo $H \subseteq G$, $\delta_G(H) = \sum_{u \in H} \delta_G(u)$. Si H es un *conjunto independiente* de G, entonces $\delta_G(H)$ es el número de aristas de G incidentes a cualquier vértice de H.

Definición 14 *Camino.* Un camino del vértice v a un vértice u en un grafo, es una secuencia de aristas:
$v_0v_1, v_1v_2, \ldots, v_{n-1}v_n$, tal que, $v = v_0$, $v_n = u$,v_k es adyacente a v_{k+1} y el longitud del camino es n. Un camino simple, es un camino tal que $v_0, v_1, \ldots, v_{n-1}, v_n$ son todos distintos.

Definición 15 *Ciclo.* Un *ciclo* (con al menos tres vértices) es un camino no vacío cuyos vértices pueden ser organizados en una secuencia cíclica, es decir, un vértice inicial y final se unen por una arista. El grafo de la Figura 2.4 representa un ciclo.

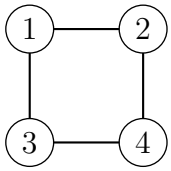


Figura 2.4: Ciclo.

Definición 16 *Árbol.* Es un grafo sin ciclos, es decir un grafo G tal que, para cualquier par de vértices en G hay un solo camino que las une.
Definición 17 *Árbol de expansión.* Un árbol de expansión T contiene todos los vértices del grafo original sin aristas que formen ciclos. En la Figura 2.5 se muestra un posible árbol de expansión del grafo de la Figura 2.3 [41].

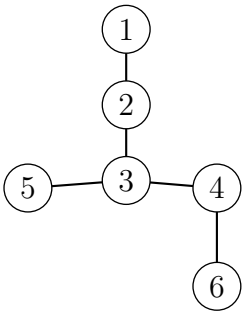


Figura 2.5: Árbol de expansión del grafo Figura 2.3.

Definición 18 *Grafo conectado.* Es un grafo $G = (V, E)$ si cada par de vértices en G tienen un camino entre ellos. Si el grafo es no conectado, cada pieza conectada máxima se denomina *componente*.
Definición 19 *Grafo completo.* Es un grafo no dirigido en el cual cualquier par de vértices está conectado por una única arista, tal y como se muestra en la Figura 2.6.

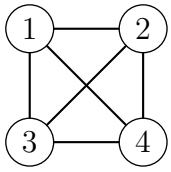
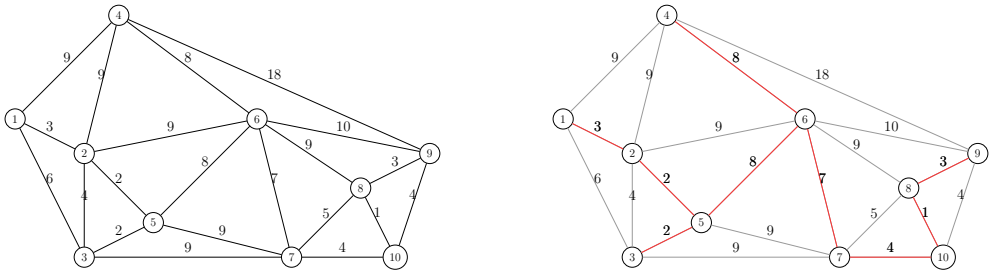


Figura 2.6: Grafo completo.

Definición 20 *Grafo ponderado.* Un grafo ponderado denotado como $G_w = (V, E)$, es un grafo donde cada arista $e \in E$ tiene asociado un número real $w(e)$, denominado *peso*. La matriz de adyacencia de un grafo ponderado G_w es una $V \times V$ matriz, tal que $M_G = (w_{vu})$, donde cada elemento (v_i, v_j) contiene un peso $w(e)$ asignado a la arista $e = v_i, v_j$ o 0 de acuerdo a si los vértices v_i y v_j son adyacentes o no en el grafo.

Definición 21 *Árbol de Expansión Mínimo.* Si H es un subgrafo tal que $H \subset G_w$ el peso $w(H)$ de H es la suma de todos los pesos $\sum w(e)$ en sus aristas. Un Árbol de Expansión Mínimo (*Minimum Spanning Tree*, MST) es un subconjunto de aristas de un grafo no dirigido cuyas aristas tienen un peso (Figura 2.7 (a)), que conecta a todos los vértices, sin ciclos con la condición de tener el mínimo peso total de aristas. Como se muestra en la Figura 2.7 (b), las aristas que se resaltan son aquellas aristas que cumplen con la condición de tener el mínimo peso total.



(a) ***Grafo ponderado*** **(b)** ***MST del grafo ponderado***

Figura 2.7: Grafos ponderados

Prim [42] propone el Algoritmo 2.2 para generar un árbol de expansión mínimo, dada un grafo G ponderado.

Algoritmo 2.2 Árbol de expansión mínimo

Entrada: $G = (V, E)$
Salida: T
1: Inicializar el árbol de expansión mínimo con un vértice elegido al azar, $v \in V$.
2: Encontrar todas las aristas que conectan el árbol con nuevos vértices, con el mínimo coste y agregarlo al árbol T.
3: Repetir el paso 2 hasta obtener un árbol de expansión mínimo.

Por otro lado, un árbol de expansión máxima, de manera análoga que el MST, es un subconjunto de aristas de un grafo no dirigido ponderado, con la condición de encontrar las aristas que tengan el mayor peso. El proceso de esto se encuentra en el Algoritmo 2.3.

Algoritmo 2.3 Árbol de expansión máximo

Entrada: $G = (V, E)$
Salida: T
1: Ordenar las aristas de G de manera decreciente de acuerdo a su peso.
2: Agregar la primera arista a T.
3: Agregar la siguiente arista a T si y sólo si no forma un ciclo en T. Si no hay aristas restantes, terminar.
4: Si T tiene $n - 1$ aristas, donde n es el número de vértices, se termina el proceso; en caso contrario, regresar al paso 3.

2.6. Estado del Arte

A lo largo de los años se han realizado diversas propuestas para el tratamiento del desbalance de clases. Conforme a los objetivos de esta tesis, se describen algunas técnicas de bajo-muestreo en esta sección dedicada al estado del arte. Es importante mencionar que el proceso de bajo-muestreo se realiza únicamente sobre la clase mayoritaria, puesto que reducir instancias de la clase minoritaria implicaría disminuir la representatividad de ésta y, en consecuencia, produciría un mayor desbalance de clases. Los métodos de bajo-muestreo pueden clasificarse en diferentes categorías de acuerdo a los procesos base implementados. A lo largo de esta sección, se presentan algunas técnicas relevantes en el estado del arte agrupadas en métodos basados en el vecino más cercano, técnicas basadas en cómputo evolutivo, técnicas basadas en ensembles y, por último, técnicas basadas en *clustering*.

2.6.1. Técnicas basadas en vecindad

Los algoritmos basados en la vecindad buscan solventar el desequilibrio o desbalance entre las clases mediante la identificación de las instancias vecinas más cercanas.

La mayoría de las propuestas basadas en vecindad implementan el algoritmo los k vecinos más cercanos (kNN) para analizar el espacio de características, con el objetivo de eliminar instancias que se encuentren lejos o cerca de la frontera de decisión. Dentro de las propuestas más comunes, se encuentra la edición de Wilson (ENN) [43], la cual elimina todas aquellas instancias que resultan mal clasificadas por el algoritmo kNN. Otra propuesta frecuentemente usada es el condensado de Hart (CNN) [44], el cual elimina instancias que están lo suficientemente lejos de la frontera de decisión y que, por tanto, pueden considerarse irrelevantes.

De manera similar, Tomek [45] propone eliminar todas aquellas instancias que formen un enlace (TL), ya que alguna de éstas corresponden a ruido o ambas instancias se encuentran en la frontera de decisión. En otro trabajo, Tomek [46] propone dos extensiones del algoritmo ENN mediante los métodos denominados RENN y All-k-NN (ALLk); el primero extiende la edición de Wilson repitiendo el proceso de eliminación hasta no tener más instancias que se puedan eliminar, mientras que el método ALLk retoma el principio de RENN con la diferencia que varía el número de vecinos incrementándolos en cada iteración.

En cambio, Kubat y Matwin [47] proponen una técnica de selección unilateral (OSS) para eliminar únicamente instancias de clase mayoritaria redundantes o ruidosas, es decir, eliminar todas aquellas instancias que se encuentren en la frontera de decisión de la clase minoritaria. La detección de estas instancias es por medio de los enlaces de Tomek, mientras que para detectar instancias redundantes se implementa el algoritmo CNN. Siguiendo la misma línea de OSS, Laurikkala [48] propone una técnica de reglas de limpieza de vecindad (NCL), donde se eliminan todas aquellas instancias de clase mayoritaria para las cuales al menos dos de sus tres vecinos más cercanos son de la clase minoritaria al aplicar el algoritmo ENN. Adicionalmente, este algoritmo también

descarta los vecinos de clase mayoritaria si una instancia de clase minoritaria resulta mal clasificada.

2.6.2. Técnicas basadas en cómputo evolutivo

El cómputo evolutivo es una área que implementa algoritmos heurísticos y, en consecuencia, las soluciones que aportan son aproximadas a una solución óptima. Este tipo de algoritmos se basan en la teoría de la evolución natural y son comúnmente utilizados en problemas de optimización y búsqueda [49].

García et al. [50] proponen un algoritmo memético, el cual surge de la combinación de un algoritmo evolutivo con una búsqueda local denominado SSMA. Este método integra búsquedas globales y locales para seleccionar un subconjunto de instancias correctamente clasificadas por el algoritmo 1NN y que este subconjunto cumpla con el porcentaje de reducción de instancias del conjunto original.

Para afrontar el problema de desbalance de clases, García et al. [51] presentan una técnica denominada EUS basada en cómputo evolutivo. El método propuesto aprovecha los algoritmos evolutivos para seleccionar instancias, considerando la función objetivo asociada a las tasas de clasificación. El objetivo es aumentar la precisión del clasificador mediante la reducción de instancias que pertenecen principalmente a la clase mayoritaria.

Por otra parte, Derrac et al. [52] proponen un algoritmo híbrido para la reducción de datos mediante la selección de instancias y características, denominado IS-RFS. La selección de instancias se realiza por medio de un algoritmo genético donde las posibles soluciones están codificadas por la presencia o ausencia de las instancias que pertenecen a la solución, mientras que la función objetivo se encarga de evaluar la tasa de efectividad del clasificador kNN. Se seleccionan aquellas características que sean capaces de discernir todas las instancias.

Por otro lado, Ha y Lee [53] propusieron un algoritmo denominado GAUS (*Genetic Algorithm based Under-Sampling*), basado en la selección de instancias de clase mayoritaria que maximicen el rendimiento de un clasificador. Este método emplea un algoritmo genético que codifica una posible solución con un número de cromosomas igual al tamaño de la clase minoritaria y, en consecuencia, las posibles soluciones contienen los identificadores de instancias de clase mayoritaria. Por su parte, la función objetivo mide la precisión del clasificador usado.

2.6.3. Técnicas basadas en ensembles

Como anteriormente se ha mencionado, las técnicas basadas en ensembles buscan realizar un tratamiento de la desproporción de clases mediante la combinación de métodos a nivel de datos o a nivel algorítmico con un conjunto de clasificadores, conocidos como ensembles.

De los algoritmos más representativos de este grupo, cabe destacar RUSBoost (RBt). Este algoritmo propuesto por Seiffert et al. [54] combina el algoritmo RUS con

boosting para la construcción de un conjunto de clasificadores. Siguiendo la misma idea de submuestreo, los algoritmos EasyEsemble (EE) y BalanceCascade (BC) [55] dividen los datos en múltiples subconjuntos de la clase mayoritaria y entrenan un *ensemble* construido por cada subconjunto. Para generar el conjunto de datos remuestreado, se fusionan las mejores instancias de cada subconjunto. La principal diferencia entre EE y BC radica en que el primero es una estrategia no supervisada para explorar el conjunto de datos mediante el uso de muestreo aleatorio con reemplazo, en tanto que el segundo explora el conjunto de datos de forma supervisada.

Por otro lado, Galar et al. [56] proponen un algoritmo basado en cómputo evolutivo en combinación con *boosting* denominado EUSBoost. Si bien el algoritmo original EUS tiene como objetivo el aumento de la precisión, en el método EUSBoost se incorpora la diversidad de los diferentes clasificadores adaptando la función objetivo de tal modo que se evalúe la diferencia estadística de los clasificadores.

Por su parte, Rayhan et al [57] proponen combinar un algoritmo basado en clustering y *boosting*, denominado CUSBoost. En general, el método subdivide el conjunto de clase mayoritaria en g subconjuntos por medio del algoritmo K-Means para, posteriormente, eliminar instancias de manera aleatoria y, una vez balanceado el conjunto, implementa *boosting* con un árbol de decisión.

Por último, Kang et al. [58] desarrollaron un método de submuestreo con filtro de ruido (EEKF), donde la clase minoritaria se filtra eliminando instancias consideradas como ruido, cuyos vecinos pertenecen a la clase mayoritaria; después de esto, se aplica el algoritmo EE.

2.6.4. Técnicas basadas en clustering

Las propuestas basadas en *clustering* se convierten en una alternativa ideal dado que pueden mitigar la pérdida de información [59]. Por ejemplo, Yen y Lee [60] proponen un algoritmo basado en agrupamiento denominado SBC, el cual se basa en la idea de que pueden existir varios grupos en un conjunto de datos y cada grupo g_i a su vez puede tener una distribución diferente de clases. Por lo tanto, todas las instancias se agrupan inicialmente en g grupos, para posteriormente seleccionar al azar varias instancias de clase mayoritaria de cada grupo conforme al número de instancias de clase minoritaria que tengan en el grupo. Finalmente, las instancias de clase mayoritaria seleccionadas de cada grupo se combinan con las instancias de clase minoritaria para formar un conjunto balanceado.

Un método similar es el propuesto por Longadge et al. [61], donde de manera inicial se agrupan las instancias de clase mayoritaria en g grupos con el algoritmo K-Means para seleccionar $|C^+| \times IR_i$ instancias de clase mayoritaria de cada grupo, donde IR_i denota la relación de desequilibrio en el grupo i y $|C^+|$ el número de instancias de clase minoritaria. Es importante mencionar que el objetivo de este algoritmo es reducir el desbalance, no obtener una distribución de clases equilibrada.

ClusterOSS (COSS), propuesto por Barella et al. [62], es una mejora del método OSS con la particularidad de iniciar el proceso agrupando instancias de clase mayori-

taria con el algoritmo K-Means. Posteriormente, las instancias más cercanas K-Means al centro de cada grupo son usadas para iniciar el método OSS, de tal manera que se eliminen aquellas instancias cercanas a la frontera de decisión. Siguiendo la línea objetivo de descartar instancias redundantes y ruidosas, Sowah et al. [63] proponen el método conocido como CUST, en el que además de eliminar instancias redundantes, elimina valores atípicos de la clase mayoritaria. Este algoritmo está compuesto por dos etapas: la primera consiste en aplicar el algoritmo TL para remover instancias ruidosas, mientras que la segunda etapa se encarga de eliminar instancias atípicas y redundantes agrupando el conjunto de datos con el algoritmo K-Means.

Por otro lado, Das et al. [64] introducen el método ClusBUS, en el que plantean descartar instancias de clase mayoritaria que se encuentren en regiones traslapadas (solapamiento de clases). Primero, este algoritmo agrupa el conjunto de datos en g grupos por medio de DBSCAN. Luego, para aquellos grupos que contienen instancias de ambas clases, el algoritmo filtra instancias de clase mayoritaria. Siguiendo la línea de agrupamiento y selección de instancias, Tsai et al. [65] proponen la técnica denominada CBIS, en la cual el algoritmo de propagación por afinidad (*Affinity Propagation Algorithm*) agrupa instancias de clase mayoritaria similares en g grupos para, posteriormente, realizar una selección de instancias en cada grupo; una vez reducidos todos los grupos, se unen a las instancias de clase minoritaria para formar un conjunto de datos equilibrado.

Bajo el uso del algoritmo K-Means, Lin et al. [66] propusieron un método basado en dos estrategias, donde la principal diferencia con respecto a los algoritmos anteriormente mencionados versa en la definición del número de grupos. Así, en esta propuesta, el valor g de grupos es igual al tamaño de la clase minoritaria. La primera estrategia de su propuesta consiste en generar g centros para representar la clase mayoritaria (CBU), mientras que la segunda usa los vecinos más cercanos de los centros. Por otro lado, el método Fast-CBUS (fCBUS) propuesto por Ofek et al. [59] realiza un agrupamiento de g grupos de la clase minoritaria y, por cada grupo, se toma un número similar de instancias de la clase mayoritaria; el criterio de selección de instancias de clase mayoritaria se basa en la cercanía a la clase minoritaria.

La Tabla 2.1 recapitula los diferentes métodos descritos en el estado del arte. En ella, se puede observar que las técnicas de bajo-muestro, en su mayoría, se basan en la eliminación de instancias lo más alejadas de o cercanas a la frontera de decisión con el método kNN, el cual dependerá del número k de vecinos. Siguiendo la línea de dependencia, también cabe señalar que algunos de los métodos basados en *clustering* implementan el algoritmo K-Means y, por tanto, una de sus principales desventajas es el hecho de tener que determinar el número de grupos g.

Por su parte, los algoritmos basados en cómputo evolutivo aproximan una solución mediante la efectividad del algoritmo kNN, con lo cual también dependerán de un número de vecinos que deberá fijarse a priori. Finalmente, los métodos basados en ensembles, si bien combinan múltiples clasificadores, la dependencia de determinar a priori un número de grupos o vecinos de acuerdo al método implementado tiene la desventaja de no establecer un valor general para todos los conjuntos de datos que presente altas tasas de efectividad en los clasificadores.

La necesidad de no depender de un valor a priori y de disminuir el uso de procesos aleatorios que pueden derivar en cierta perdida de información abre nuevas líneas de estudio para contemplar el uso de algoritmos que dependan de la propia distribución de los datos, así como de métodos deterministas que permitan obtener invariablemente resultados similares bajo condiciones iniciales similares.

Tabla 2.1: Compendio de métodos de bajo-muestreo del estado del arte.

	Técnica	Características
Aleatorio	RUS [6].	Elimina instancias de manera aleatoria.
vecindad	CNN [44]. ENN [43]. TL [45]. NCL [67]. OSS [47]. RENN [46]. ALLk [46].	Eliminan instancias que se encuentren lejos o cerca de la frontera de decisión, por medio del algoritmo *kNN*.
Evolutivo	EUS [51]. SSMA [50]. ISRFS [52]. GAUS [53].	Solución aproximada mediante búsquedas globales y locales, la función objetivo se basa en las tasa de efectividad del algoritmo *kNN*.
Ensemble	EE [55]. BC [55]. RBT [54]. EEKF [58]. EUSBoost [56]. CUSBoost [57].	Incorporación de métodos como Bagging y Boosting que combinan múltiples clasificadores basados en la técnica de votación con algún otro método de bajo-muestreo como el aleatorio, los basados en vecindad, evolutivos o basados en clustering.
Clustering	SBC [60]. CBU [66]. fCBUS [59]. CBIS [65]. COSS [62]. CUST [63].	Agrupación del conjunto de datos en subgrupos con algoritmos tales como K-Means o DBSCAN, para luego realizar bajo-muestreo a cada grupo, seleccionando instancias lo más lejos o cerca de la frontera de decisión con algoritmos basados en vecindad.

Parte II

Propuestas y Metodología

Capítulo 3

Nuevos algoritmos basados en grafos y clustering

Asumir que los conjuntos de datos se encuentran libres de complejidades implicaría una pérdida de rendimiento en los clasificadores y, por lo tanto, se hace fundamental el desarrollo de estrategias que den solución a dichas implicaciones sin reducir la precisión del clasificador. El trabajo de esta tesis se centra en la creación de algoritmos basados en grafos y clustering como estrategias para solventar el problema del desbalance entre clases y de otras complejidades como el solapamiento de clases y el ruido.

Dado el estado del arte sobre métodos para el tratamiento de desbalance entre clases, es evidente la falta de algoritmos que aprovechen la teoría de grafos para realizar el remuestro del conjunto de datos, así como la facilidad que los algoritmos de *clustering* tienen para eliminar ruido. En consecuencia, las propuestas de esta tesis se resumen en el esquema de la Figura 3.1: en el contexto de *small data*, los métodos se detallan en el resto de este capítulo, mientras que las propuestas en el contexto de *Big Data* se abordan en el Capítulo 8. Cabe destacar que todas estas propuestas también solventan la necesidad de no depender de variables determinadas a priori y la incorporación de métodos deterministas que producirán invariablemente los mismos resultados.

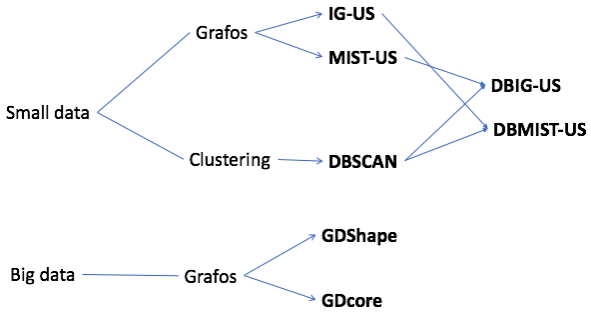


Figura 3.1: Resumen de propuestas basadas en grafos y clustering.

3.1. Tratamiento de desbalance de clases

Una de las complejidades de datos que tiene gran interés hoy en día en los problemas del mundo real, es el desbalance de clases. La Figura 3.2 muestra la propuesta para hacer frente al desbalance de clases. El proceso inicia separando el conjunto de datos en dos conjuntos, C^+ para instancias de clase minoritaria y C^- para la clase mayoritaria. En general, como se observa, el proceso de bajo-muestreo se realiza únicamente para la clase mayoritaria (C^-) para ser considerada un grafo completo ponderado. La ponderación de las aristas se basa en la distancia Euclídea que existe entre un vértice y otro, cada instancia actuará como un vértice en el grafo. Por último, una vez generado un grafo completo ponderado, se obtiene un subconjunto denominado C'^-, siendo una muestra.

La construcción de un grafo completo ponderado $G_w = (V, E)$, está definido por:

- $V(G_w) = \{i \in V(G_w) \mid p_i \in C^-\}$, es decir, un vértice para cada instancia.
- $E(G_w) = \{\{v, u\} \mid v, u \in V(G_w)\}$, y
- $\forall e = \{v, u\} \in E(G_w)$, $w(e) = dist(v, u)$ donde $dist(v, u)$ es la distancia Euclídea entre v y u.

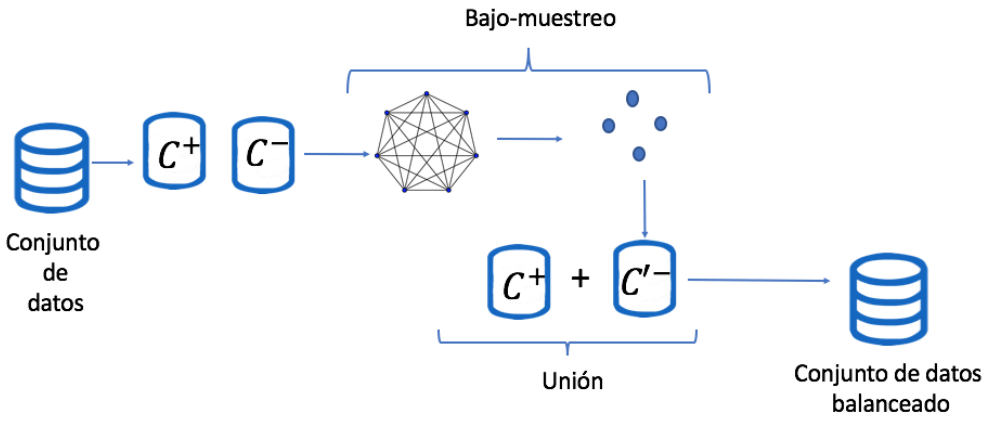


Figura 3.2: Propuesta para bajo-muestreo basada en grafos

Una vez construido el grafo G_w, es posible aplicar alguna de las siguientes estrategias: la construcción de un subgrafo inducido para obtener la silueta de la clase mayoritaria (Sección 3.1.1) o la construcción de un árbol de expansión mínimo (Sección 3.1.2). Es importante mencionar que la estrategia de bajo-muestreo toma únicamente una muestra representativa, por medio de la Ecuación 3.1.

$$f(x) = \begin{cases} \frac{|C_1^-| \quad \sigma^2 Z^2}{e^2(|C_1^-|-1)+\sigma^2 Z^2} & \text{Si } x = 1 \\ \\ \frac{f(x-1) \quad \sigma^2 Z^2}{e^2(|C_1^-|-1)+\sigma^2 Z^2} & \text{de otro modo} \end{cases} \tag{3.1}$$

donde $Z = 1.96$, $\sigma = 0.5$ y $e = 0.05$, para el 95 % de confianza.

3.1.1. Subgrafo inducido (IG-US)

Con la intención de obtener las instancias que sean parte del borde de la clase mayoritaria (Silueta), se propone el Algoritmo 3.4 como una estrategia de bajo-muestreo denominada IG-US (Induced Graph-UnderSampling).

Algoritmo 3.4 IG-US

Entrada: C_1^-, R_{max}, C^+
Salida: C_2^-
1: Construir $G_w = (V, E)$ un grafo completo de C^-.
2: $Sample \leftarrow |C_1^-|$
3: $IR \leftarrow \frac{Sample}{|C^+|}$
4: **while** $IR > maxIR$ **do**
5: $\quad Sample \leftarrow \frac{Sample \quad \sigma^2 Z^2}{e^2(|C_1^-|-1)+\sigma^2 Z^2}$
6: $\quad IR \leftarrow \frac{Sample}{|C^+|}$
7: **end while**
8: $C_2^- \leftarrow []$
9: $M_G \leftarrow$ incidenceMatrix(G_w)
10: **while** $|C_2^-| < Sample$ **do**
11: $\quad Maximum \leftarrow$ GetMaximum(M_G)
12: $\quad$ **for all** $\{v, u\}$ in $Maximum$ **do**
13: $\quad\quad$ **if** v no ha sido visitado **then**
14: $\quad\quad\quad$ Marcar v como visitado
15: $\quad\quad\quad C_2^- \leftarrow C_2^- \cup \{v\}$
16: $\quad\quad$ **end if**
17: $\quad\quad$ **if** u no ha sido visitado **then**
18: $\quad\quad\quad$ Marcar u como visitado
19: $\quad\quad\quad C_2^- \leftarrow C_2^- \cup \{u\}$
20: $\quad\quad$ **end if**
21: $\quad\quad$ Remover $\{v, u\}$ de M_G
22: $\quad$ **end for**
23: **end while**
24: **return** C_2^-

El proceso de bajo-muestreo como medio de obtención de silueta inicia con la generación de un grafo completo ponderado (línea 1) de la clase mayoritaria, posteriormente se calcula el grado de desbalance que tiene el conjunto de datos (línea 3), de tal modo que se determine el tamaño de la muestra (Ecuación 3.1), el cual será el valor que determine el fin del proceso (líneas 4-7), nótese que este valor dependerá del grado de desbalance máximo, asignado por el usuario. El siguiente paso es la generación de la matriz de adyacencia de las instancias pertenecientes a la clase mayoritaria (línea 9).

Por medio de la matriz de adyacencia, se obtienen cada par de vértices cuya distancia Euclídea sea la mayor hasta el momento (línea 11). Posteriormente, cada vértice que no haya sido visitado anteriormente se agrega al subconjunto resultante C_2^- (líneas 12-20), anulando estos vértices en la matriz de adyacencia (línea 21). El proceso finaliza al obtener un subconjunto C_2^- de tamaño menor al del resultado de la Ecuación 3.1.

3.1.2. Árbol de expansión mínimo (MIST-US)

El interés de obtener instancias que se encuentran lo más alejadas del borde de la clase mayoritaria, es para obtener un *core* o "núcleo" representativo. El Algoritmo 3.5 describe la propuesta denominada MIST-US (Minimal Spanning Tree-UnderSampling).

Algoritmo 3.5 MIST-US

Entrada: C^-, IRm grado deseado, C^+
Salida: $C'^- \subset C^-$ conjunto de datos balanceado
1: Construir $G_w = (V, E)$ un grafo completo de C^-.
2: $S \leftarrow \lfloor |C^-| \rfloor$
3: $IR \leftarrow \frac{S}{|C^+|}$
4: $C'^- \leftarrow []$
5: $M_G \leftarrow$incidenceMatrix(G_w)
6: $MST \leftarrow$GetMST(G_w, M_G)
7: **while** $IR > IRm$ **do**
8: $\quad S \leftarrow \frac{S \quad \sigma^2 Z^2}{e^2(|C^-|-1)+\sigma^2 Z^2}$
9: $\quad IR \leftarrow \frac{S}{|C^+|}$
10: **end while**
11: **for all** $\{v, u\}$ in $E(MST)$ **do**
12: $\quad C'^- \cup u$
13: $\quad C'^- \cup v$
14: $\quad$ **if** $|C'^-| > S$ **then**
15: $\quad\quad$ **return** C'^-
16: $\quad$ **end if**
17: **end for**

La propuesta basada en la construcción de un árbol de expansión mínimo inicia con la construcción de un grafo completo ponderado de clase mayoritaria (línea 1), posteriormente se calcula el grado de desbalance que tiene el conjunto de datos (línea 3). El método *GetMST* (línea 6), devuelve un árbol de expansión mínimo generado por el algoritmo de Prim [42] a partir de la gráfica G_w. Posteriormente, se toma un determinado número de instancias calculadas por la Ecuación 3.1 que determine el fin del proceso (líneas 7-10), nótese que este valor dependerá del grado de desbalance máximo, asignado por el usuario. Finalmente (líneas 11-17), se toman las primeras instancias del árbol mínimo de expansión, hasta tener un subconjunto C_2^- de tamaño determinado en las líneas 1-9.

En general, la idea de obtener un núcleo de clase mayoritaria es descartar todas aquellas instancias que estén lo suficiente cerca a la frontera de clase minoritaria.

3.2. DBSCAN como estrategia de limpieza

Los datos, por su propia naturaleza, mantienen complejidades que deben ser tratadas (ver Capítulo 2).

La estrategia de limpieza propuesta hace uso del algoritmo DBSCAN para obtener un subconjunto comprimido de datos de clase mayoritaria, aprovechando las ventajas que ofrece con respecto a la eliminación de datos identificados como ruido.

Al algoritmo original se le realizaron algunas adaptaciones, las cuales consideran: la estimación de ϵ y $minPts$ se basa en las ecuaciones propuestas en [29], con la variación de utilizar la cardinalidad de la clase minoritaria para determinarlos.

$$\epsilon = \sqrt{\frac{\sum_{|C^-|}^{i=1} distancia(m, p_i^-)}{|C^-|}} \tag{3.2}$$

donde m es el vector medio de la clase mayoritaria, p_i^- representa una instancia de clase mayoritaria y $distancia$ se obtiene a partir de la distancia Euclidea.

$$minPts = \frac{\pi \times \epsilon^2}{TotalVolume} \times |C^+| \tag{3.3}$$

donde $TotalVolume = \frac{4}{3} \times \pi \times \epsilon^3$.

El Algoritmo 3.6 muestra el proceso de DBSCAN como estrategia de limpieza de los datos. La propuesta inicia particionando el conjunto de datos en dos subconjutos C^- y C^+ con instancias de clase mayoritaria y clase minoritaria, respectivamente. DBSCAN es aplicado únicamente al conjunto C^-, tomando arbitrariamente una instancia y buscando todas las instancias alcanzables con respecto a ϵ y $minPts$. En caso de no contar con $minPts$ instancias vecinas a una distancia ϵ, la instancia analizada es marcada como *ruido* y eliminada del conjunto C^-. Este proceso se realiza hasta que los parámetros libres no cambien.

Algoritmo 3.6 Propuesta DBSCAN para limpieza

Entrada: $CD = \{X_1, X_2, \ldots, X_N\}$
Salida: $C^- conjuntodeclasemayoritariasinruido$
1: Dividir el conjunto CD en dos subconjuntos tales que C^- contenga instancias de clase mayoritaria y C^+ contenga instancias de clase minoritaria.
2: **repeat**
3: Estimar los valores ϵ y $minPts$.
4: Aplicar DBSCAN al conjunto C^-, eliminando instancias marcadas como ruido.
5: **until** ϵ y $minPts$ no cambien
6: **return** C^-

3.3. Tratamiento de desbalance de clases, traslape de clases y/o ruido

Desafortunadamente, el problema de desbalance de clases no siempre se encuentra por sí solo, sino que en la mayoría de los casos está acompañado de alguna otra complejidad [5, 1].

La propuesta para realizar el tratamiento del problema de desbalance de clases, traslape de clases y/o ruido en los conjuntos de datos se muestra en la Figura 3.3. Como puede verse, la propuesta requiere aplicar el algoritmo DBSCAN modificado 3.6 hasta que los parámetros libres no cambien (Paso de filtrado). Posteriormente, se aplica ya sea la generación de árbol de expansión mínimo (Sección 3.3.2) o la obtención de un subgrafo inducido (Sección 3.3.1) (Bajo-muestreo).

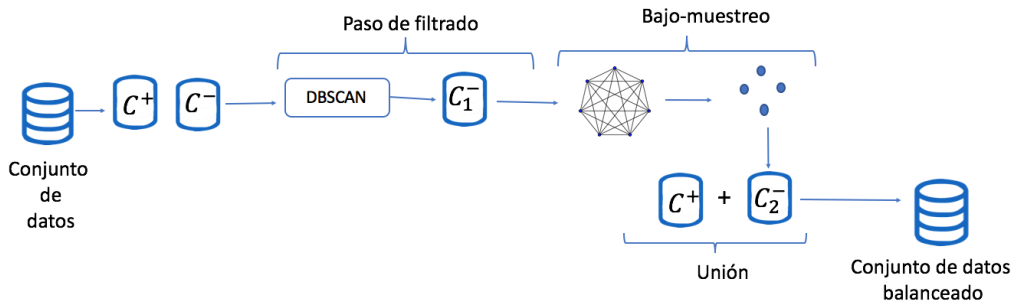


Figura 3.3: Propuesta para el tratamiento de desbalance de clases, traslape de clases y/o ruido

3.3.1. DBIG-US: DBSCAN y subgrafo inducido

Con la intención de afinar el proceso de tratamiento de desbalance de clases basado en la obtención de un subgrafo inducido (Sección 3.1.1), se propone el algoritmo en dos pasos denominado DBIG-US (Algortimo 3.7). En un primer paso para eliminar instancias consideradas como ruido se ejecuta el algoritmo de DBSCAN modificado (Algoritmo 3.6), de este modo se trata de solventar tanto el traslape de clases como instancias mal etiquetadas.

La idea principal es que una vez limpia la frontera de decisión, la obtención de instancias de clase mayoritaria que están lo más lejos unas de otras, sea todas aquellas que están cerca de la frontera de decisión pero libres de ruido.

Algoritmo 3.7 DBIG-US

Entrada: $DB = \{p_1, p_2, \ldots, p_N\}$
Salida: DB' conjunto balanceado
1: Dividir el conjunto DB en dos subconjuntos tales que, C^- contenga instancias de clase mayoritaria y C^+ contenga instancias de clase minoritaria.
2: **repeat**
3: Estimar los valores ϵ y MinPts.
4: Aplicar DBSCAN al conjunto C^-, donde se estima remover instancias marcadas como ruido.
5: **until** ϵ y $minPts$ no cambien
6: $C^- \leftarrow$ IG-US(C^-)
7: $DB' = C^+ \cup C^-$

3.3.2. DBMIST-US: DBSCAN y árbol de expansión mínimo

La estrategia que se denomina DBMIST-US (Algortimo 3.8) se basa en la obtención de un árbol de expansión mínimo (Sección 3.1.2) una vez limpio el conjunto de datos por el algoritmo de DBSCAN modificado (Algoritmo 3.6).

Algoritmo 3.8 DBMIST-US

Entrada: $DB = \{p_1, p_2, \ldots, p_N\}$
Salida: DB' conjunto balanceado
1: Dividir el conjunto DB en dos subconjuntos tales que, C^- contenga instancias de clase mayoritaria y C^+ contenga instancias de clase minoritaria.
2: **repeat**
3: Estimar los valores ϵ y MinPts.
4: Aplicar DBSCAN al conjunto C^-, donde se estima remover instancias marcadas como ruido.
5: **until** ϵ y $minPts$ no cambien
6: $C^- \leftarrow$ MIST-US(C^-)
7: $DB' = C^+ \cup C^-$

En general, el objetivo de este algoritmo es obtener instancias ubicadas en el núcleo de la clase mayoritaria una vez limpio el conjunto de datos. De tal modo que las instancias seleccionadas por el árbol mínimo de expansión no sean parte del traslape de clases o con ruido.

Capítulo 4
Marco metodológico

La metodología que se sigue para dar solución al problema planteado se muestra en la Figura 4.1. Como puede observarse, la metodología de la propuesta se basa en un proceso típico de KDD [68]; si bien este proceso parece ser secuencial, de acuerdo a los resultados obtenidos se debe permitir el flujo hacia atrás o hacia adelante.

El proceso inicia con un conjunto de datos derivado de algún problema del mundo real, seguidamente se preprocesa el conjunto de datos con el fin de solventar algunas complejidades propias de los datos. Es importante resaltar que las propuestas presentadas en esta tesis inciden de manera directa en esta fase. Una vez atendidas las características intrínsecas de los datos, se procesan los conjuntos de datos mediante clasificadores para obtener una generalización del conocimiento. Finalmente, se analizan los resultados con el objetivo de determinar el rendimiento de las propuestas.

Aunque las etapas de la metodología pueden realizarse de forma secuencial, de acuerdo a los resultados que se obtengan de la fase de validación de los resultados, si los índices de precisión de cada clasificador superan o igualan a los resultados de los métodos del estado del arte, se finaliza el tratamiento de complejidades; en caso contrario, se requiere afinar el conjunto de datos atendiendo algunas de las otras características intrínsecas de los datos. En el resto de este capítulo se describe cada una de las tareas realizadas.

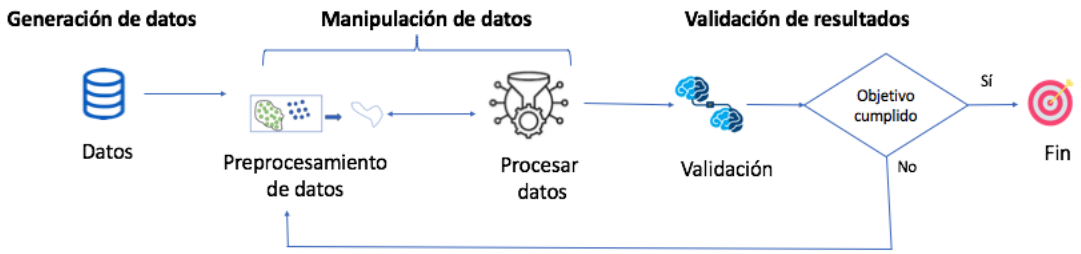


Figura 4.1: Metodología seguida en esta tesis.

4.1. Obtención de datos

La experimentación reportada en esta tesis se realizó con 24 conjuntos de datos, todos ellos con dos clases, obtenidos del repositorio que ofrece la herramienta KEEL (`https://sci2s.ugr.es/keel/imbalanced.php#subA`).

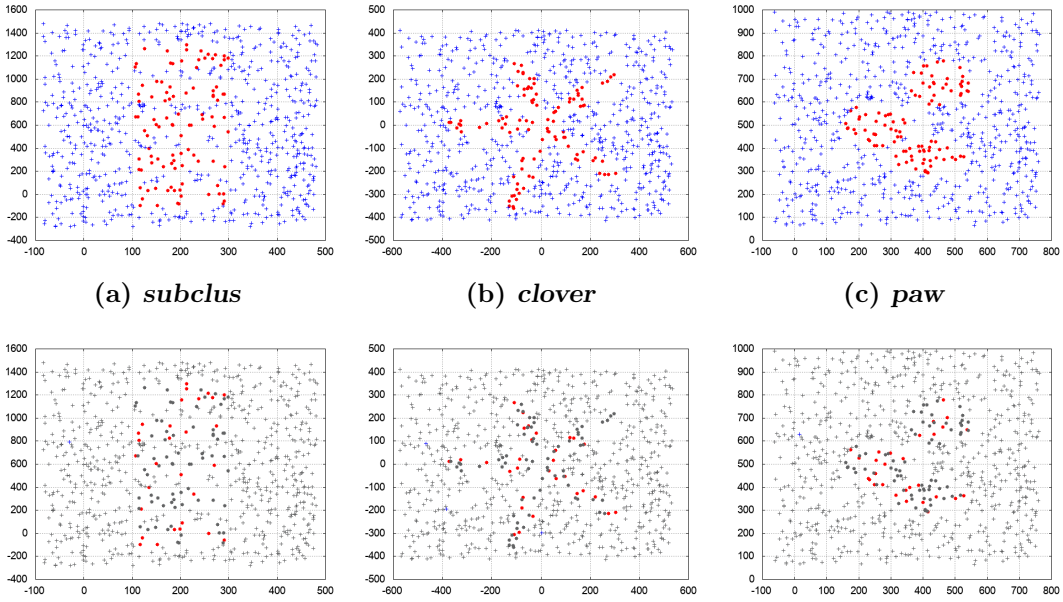
La Tabla 4.1 muestra las características principales de dichos conjuntos de datos: el nombre del conjunto de datos, la distribución a priori de las clases, el número total de instancias y el grado de desbalance (IR). Esta última característica se calcula como la relación entre el número de instancias de clase mayoritaria con respecto al número de instancias de clase minoritaria ($IR = \frac{|C^-|}{|C^+|}$) [69].

Tabla 4.1: Características de los conjuntos de datos reales.

	Conjunto de datos	#Características	Distribución	#Instancias	IR
1	yeast05679v4	8	51 – 477	528	9.35
2	vowel0	13	90 – 898	988	9.98
3	glass016v2	9	17 – 175	192	10.29
4	glass2	9	17 – 197	214	11.59
5	shuttle0v4	9	123 – 1706	1829	13.87
6	yeast1v7	7	30 – 429	459	14.30
7	glass4	9	13 – 201	214	15.47
8	ecoli4	7	20 – 316	336	15.80
9	pagBks13v4	10	28 – 444	472	15.86
10	glass016v5	9	9 – 175	184	19.44
11	shuttle2vs4	9	6 – 123	129	20.50
12	yeast1458vs7	8	30 – 663	693	22.10
13	glass5	9	9 – 205	214	22.78
14	yeast2vs8	8	20 – 462	482	23.10
15	flareF	11	43 – 1023	1066	23.79
16	yeast4	8	51 – 1433	1484	28.10
17	yeast1289v7	8	30 – 917	947	30.57
18	yeast5	8	44 – 1440	1484	32.73
19	ecoli0137v26	7	7 – 274	281	39.14
20	abalone17v78910	8	58 – 2280	2338	39.31
21	yeast6	8	35 – 1449	1484	41.40
22	shuttle2v5	9	49 – 3267	3316	66.67
23	kddBfOfw-b	41	30 – 2203	2233	73.43
24	poker89v5	10	25 – 2050	2075	82.00

De forma adicional, y con la intención de analizar de forma más detallada el comportamiento de los algoritmos propuestos en escenarios controlados, se utilizaron 15 conjuntos de datos sintéticos disponibles en el repositorio de KEEL (`https://sci2s.ugr.es/keel/imbalanced.php#sub50`).

Los conjuntos de datos denominados *subclus*, *clover* y *paw*, constan de 800 instancias cada uno, con IR = 7 y la inclusión de ruido (instancias atípicas) en un 0 %, 30 %, 50 %, 60 %, y 70 %. La Figura 4.2 muestra gráficamente la distribución de puntos que tienen estos conjuntos de datos.



(a) ***subclus*** (b) ***clover*** (c) ***paw***

(d) ***subclus-70*** (e) ***clover-70*** (f) ***paw-70***

Figura 4.2: Dispersión de los conjuntos de datos sintéticos con 0 % y 70 % de ruido (los puntos en color gris son datos ruidosos)

4.2. Preprocesado de datos

En la fase de preprocesado de datos es en la que se aplican los algoritmos propuestos en en esta tesis (ver Capítulo 3) para tratar las complejidades de datos: desbalance de clases, traslape de clases y/o ruido.

Para validar el resultado de los algoritmos propuestos en esta tesis, se seleccionaron algunos de los algoritmos de bajo-muestreo más utilizados en el estado del arte, los cuales se muestran en la Tabla 4.2, divididos de acuerdo al tipo de estrategia que utilizan (Sección 2.4).

Tabla 4.2: Métodos de bajo-muestreo del estado del arte.

	Técnica
Vecindario	RUS [6]. CNN, Condensado de Hart [44]. ENN, Edición de Wilson [43]. TL, Tomek's links [45]. NCL *Neighborhood Cleaning Rule* [67]. OSS, *One-sided selection* [47].
Evolutivo	EUS, *Evolutionary Under-Sampling* [51].
Ensemble	EE, *Easy Ensemble* [55]. BC, *Balance Cascade* [55]. RBT, *ROSBOOST* [54]. EEKF [58].
Clustering	SBC, Undersampling Based on Clustering [60]. CBU [66]. fCBUS, *Fast-CBUS* [59]. CBIS [65]. COSS, *ClusterOSS* [62].

4.3. Clasificación

Para determinar la viabilidad y solidez de los algoritmos de preprocesado propuestos, se evaluó el desempeño de tres modelos de clasificación ampliamente utilizados en Aprendizaje Automático y Reconocimiento de Patrones: la regla del vecino más cercano (1NN), el árbol de decisión C4.5 (J48) y la máquina de vector soporte (*Support Vector Machine*, SVM).

Para la ejecución de las pruebas se utilizó el software de código abierto WEKA [70], utilizando los parámetros que el software brinda por defecto para cada clasificador:

Tabla 4.3: Parámetros por defecto de los clasificadores usados en WEKA.

	Parámetros
1NN	k=1; número de vecinos
J48	confidenceFactor= 0.25 minNumObj = 2 numFolds = 3 reducedErrorPruning = False seed =1 subtreeRaising = True unpruned = False useLaplace = False
SVM	normalizar=True L= $1.0e-3$; parámetro de tolerancia P= $1.0e-12$; épsilon del error de redondeo. seed=1 kernel= PolyKernel modelo de calibración= Logistic (hasta converger)

Para conjunto de datos, se aplicó una validación cruzada de 10 particiones (*10-fold cross-validation*). El conjunto de datos fue dividido en 10 partes iguales, de las cuales nueve particiones fueron utilizadas para entrenamiento del clasificador y la partición restante para fines de prueba.

4.4. Evaluación y análisis estadístico de los resultados

Para validar el rendimiento de las propuestas, el análisis de resultados se realiza desde dos perspectivas: Evaluación de la precisión del clasificador y Análisis de significancia estadística. El primero de ellos, busca identificar el desempeño del clasificador en términos de precisión. Mientras que el segundo análisis, permite determinar las mejoras estadísticamente significativas que un clasificador tiene respecto al resto de métodos, ya que muestra qué tanto los resultados obtenidos respaldan la hipótesis, así como si las conclusiones alcanzadas pueden generalizarse [71].

4.4.1. Evaluación de la clasificación

Comúnmente en las áreas de Aprendizaje Automático y Reconocimiento de Patrones, para validar el rendimiento de un clasificador en términos de precisión, se hace uso de una **matriz de confusión** [2]. La Tabla 4.4 representa la matriz de confusión para un problema de dos clases, donde se representa el número de predicciones de un clasificador para cada clase con respecto al valor real de la clase. Para este caso, se considera la clase minoritaria como positiva, mientras que la clase mayoritaria se denomina negativa.

Tabla 4.4: Matriz de confusión

	Predicción de clase Positiva	Predicción de clase Negativa
Clase positiva	Verdadero Positivo (TP)	Falso Negativo (FN)
Clase negativa	Falso Positive (FP)	Verdadero Negativo (TN)

El *Verdadero Positivo* y el *Verdadero Negativo* se definen como el número de casos positivos o negativos que han resultado clasificados correctamente. Por el contrario, el *Falso Positivo* se refiere al número de casos negativos que se han clasificados como de la clase positiva, mientras que el número de casos que siendo de clase positiva se clasifican como clase negativa se denomina *Falso Negativo*.

A partir de la Tabla 4.4, algunas de las medidas de rendimiento que se pueden obtener son [72]:

- *Precisión general* mide la efectividad a nivel global del clasificador: $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

- *Sensibilidad o tasa de verdaderos positivos* mide el grado de efectividad de un clasificador en el momento en que identifica las instancias positivas: $TPr = \frac{TP}{TP+FN}$.
- *Especificidad o tasa de verdadero negativo* determina el grado de efectividad con la que el clasificador identifica instancias etiquetadas como negativas: $TNr = \frac{TN}{TN+FP}$
- *Media geométrica* maximiza la precisión de cada clase en un clasificador: $Gmean = \sqrt{TPR \cdot TNR}$

Dado que la media geométrica permite evaluar el impacto que tiene la tasa de errores de cada clase por separado, es la medida que se utilizará para análisis de resultados de la clasificación.

4.4.2. Análisis de significancia estadística

Cuando se desarrollan nuevas técnicas y se desea comparar su rendimiento con las existentes en el estado del arte, es recomendable la utilización de pruebas de significancia estadística [73], a fin de determinar las mejoras reales que éstas tienen.

Una de las pruebas de significancia estadística más utilizada es la de Friedman [73], el objetivo de esta prueba es determinar si existe diferencia entre los resultados del clasificador, por cada método de preprocesado y conjunto de datos. El primer paso es convertir los resultados en rangos, es decir, asignar al mejor resultado el rango 1, el segundo mejor el rango 2, así sucesivamente, en caso de empates, se calculan los rangos promedio. El segundo paso, implica calcular los rangos promedio por método evaluado (Ecuación 4.1), bajo la hipótesis nula que establece que todos los algoritmos son equivalentes y que sus *ranks* promedios tienen que ser iguales.

$$R_j = \frac{1}{M} \sum_i r_i^j \tag{4.1}$$

donde r_i^j el *rank* del $j-$ésimo método de μ en el $i-$ésimo conjunto de M conjuntos de datos.

Otra de las pruebas utilizadas para encontrar diferencias estadísticamente significativas entre cada par de clasificadores es la prueba de Wilcoxon [74]. Esta prueba clasifica las diferencias en el desempeño de dos algoritmos para cada conjunto de datos, ignorando los signos, y compara los resultados en la clasificación para las diferencias positivas y negativas.

Dado β_i la diferencia entre los puntajes de rendimiento de los métodos en el $i-$ésimo conjunto de datos de M conjuntos de datos. Las diferencias se clasifican según sus valores absolutos; los rangos promedio son asignados en caso de empate. Dado R^+ la suma de ranks para los conjuntos de datos que el segundo método superó al primer método, y R^- , la suma de los ranks para el caso contrario (Ecuación 4.2); los rangos de $\beta_i = 0$ se dividen equitativamente entre las sumas. En caso de haber un número

impar, uno de ellos se ignora, por este motivo es importante seleccionar un número par de métodos y conjuntos de datos.

$$R^{+} = \sum_{\beta_i > 0} rank(\beta_i) + \frac{1}{2} \sum_{\beta_i = 0} rank(\beta_i) \quad R^{-} = \sum_{\beta_i < 0} rank(\beta_i) + \frac{1}{2} \sum_{\beta_i = 0} rank(\beta_i) \tag{4.2}$$

Sea τ la menor de las sumas, tal que $\tau = min(R^{+}, R^{-})$, las estadísticas se determina con la ecuación 4.3.

$$\iota = \frac{\tau - \frac{1}{4}M(M+1)}{\sqrt{\frac{1}{24}M(M+1)(2M+1)}} \tag{4.3}$$

Si τ es menor o igual que el valor de la distribución de Wilcoxon [75] , se rechaza la hipótesis nula de igualdad de medias, lo que significa que un método dado supera al otro con el $p - valor$ asociado.

Parte III

Resultados Experimentales

Capítulo 5

Resultados de tratamiento de desbalance de clases

Las propuestas desarrolladas en esta tesis se comparan con otros métodos de bajo-muestreo en términos de media geométrica. En esta capítulo, se evalúa el rendimiento de dos métodos de bajo-muestreo para tratar el desbalance de clases: subgrafo inducido (IG-US) y árbol de expansión mínimo (MIST-US). Cabe señalar que aquí únicamente se ha incluido un resumen de los resultados con el objetivo de facilitar su análisis, mientras que la totalidad de estos resultados se pueden consultar en el Apéndice A.

Adicionalmente, para determinar si existen diferencias estadísticamente significativas entre cada par de métodos, se aplica el test de Wilcoxon. En general, la mitad superior de la diagonal de las tablas que se muestran para este análisis corresponde a los resultados para el valor $\alpha = 0.9$ (10 % de probabilidad), mientras que la mitad inferior de la diagonal es para un nivel 0.95. El símbolo "$\bullet$" indica que el método de la fila fue significativamente mejor que el método de la columna, mientras que el símbolo "$\circ$" representa que el método de la columna funcionó significativamente mejor que el método de la fila. La ausencia de cualquiera de los símbolos anteriores implica que los métodos de la columna y de la fila no muestran diferencias estadísticamente significativas.

5.1. Rendimiento del subgrafo inducido (IG-US)

Cabe recordar que el objetivo de esta propuesta es disminuir el tamaño de la clase mayoritaria mediante la identificación y obtención de las instancias que se encuentran cercanas a la frontera de decisión.

La Figura 5.1 muestra el análisis de los resultados en términos de la media geométrica obtenida por el subgrafo inducido en comparación con los métodos de bajo-muestreo del estado del arte. En los resultados se incluye el resultado obtenido con el conjunto de datos original (sin preprocesado) para fines de comparación.

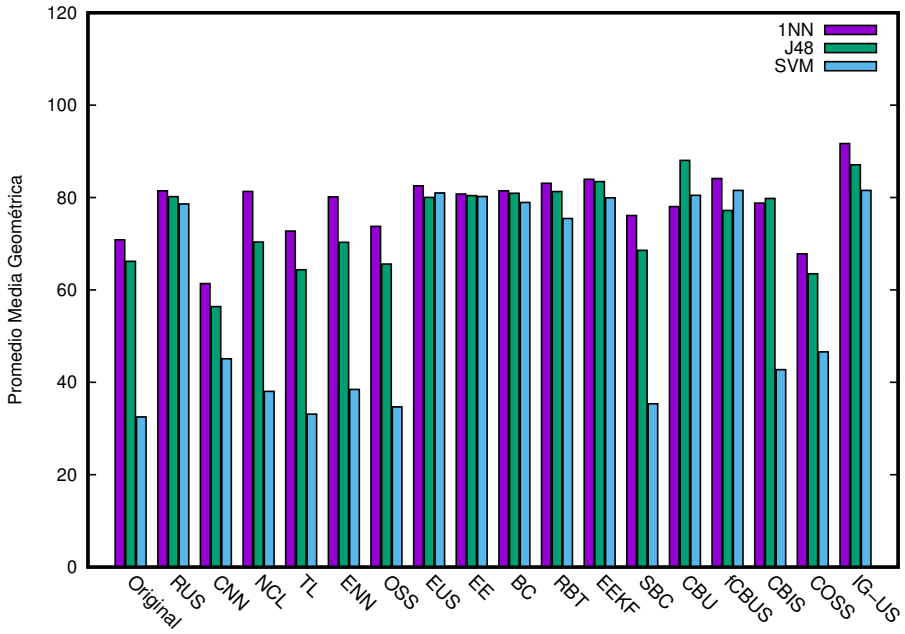
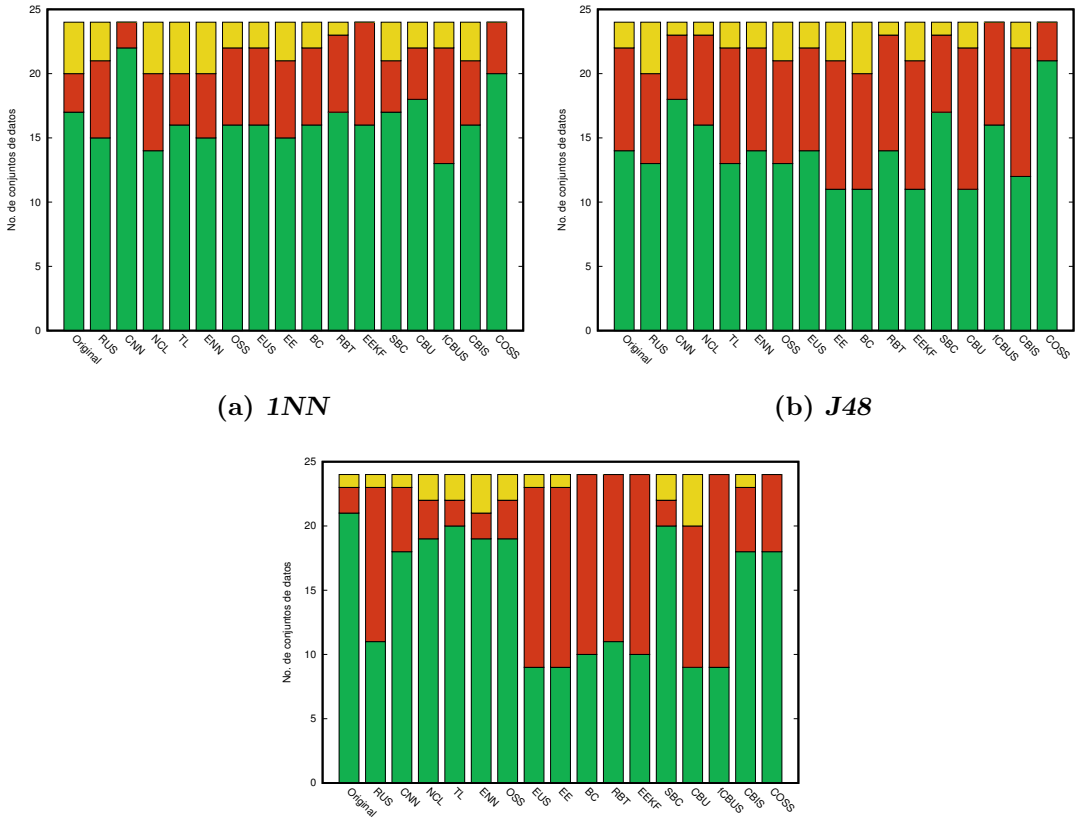


Figura 5.1: Comparativa de IG-US con el resto de métodos con respecto al promedio de la media geométrica obtenida.

Como se observa en la Figura 5.1, independientemente del clasificador, la propuesta IG-US obtiene porcentajes de rendimiento superiores al 70 %, en contraste con técnicas basadas en vecindario (CNN, NCL, TL, ENN y OSS), para las cuales, en específico el clasificador SVM obtienen rendimientos por debajo del 40 %, esto sugiere que conservar instancias lo más cercanas a la frontera de decisión, disminuye la representatividad de clase minoritaria para el SVM. A diferencia de métodos como RUS, EUS, EE, BC, RBT, EEKF, SBC, fCNUS, CBIS, COSS el comportamiento del rendimiento obtenido por el promedio de la media geométrica es similar a la propuesta IG-US.

Las gráficas mostradas en la Figura 5.2 representan el número de conjuntos de datos para los cuales la propuesta IG-US fue mejor (verde), igual (amarillo) o peor (rojo) que el resto de técnicas probadas.



(a) ***1NN***

(b) ***J48***

(c) ***SVM***

Figura 5.2: Comparativa de IG-US con el resto de métodos con respecto al número de conjuntos de datos en los que IG-US fue mejor (verde), igual (amarillo) o peor (rojo).

Como puede observarse, en la Figura 5.2 los clasificadores 1NN y J48, en más de la mitad de los conjuntos de datos probados, IG-US obtuvo la más alta media geométrica por conjunto y método comparado. Sin embargo, como es posible visualizar, el color rojo predomina para el clasificador SVM, determinando que, IG-US obtuvo resultados más bajos e inclusive no obtuvo resultados similares (amarillo) en métodos basados en ensembles (BC, RBT y EEKF) y en un método basado en *clustering* (fCBUS).

A fin de determinar la existencia de mejoras estadísticamente significativas, se aplica la prueba de Wilcoxon para valores de $\alpha = 0.9$ y $\alpha = 0.95$. Las Tablas 5.1-5.3 corresponden a los resultados del test estadístico de Wilcoxon para los clasificadores 1NN, J48 y SVM respectivamente.

Tabla 5.1: Test de Wilcoxon para resultados de IG-US con el clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US
Original	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○
RUS	●	-	●		●												●	○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NCL	●		●	-	●		●						●	●		●	●	○
TL	●	○	●	○	-	○		○	○	○	○	○	○		○	○		○
ENN	●		●		●	-	●						●	●	○		●	○
OSS	●		●	○		○	-	○	○	○	○	○			○			○
EUS	●		●		●		●	-					●	●			●	○
EE	●		●		●				-						○		●	○
BC	●		●		●		●			-							●	○
RBT	●		●		●		●				-			●		●	●	○
EEKF	●		●		●		●					-	●	●		●	●	○
SBC	●		●	○	●	○		○				○	-		○			○
CBU			●	○				○			○	○		-	○			○
fCBUS	●		●		●		●						●	●	-	●	●	
CBIS	●		●		●										○	-		○
COSS				○		○		○	○	○	○	○			○		-	○
IG-US	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	-
$\alpha = 0.9$	1	4	0	8	2	7	2	7	5	5	7	8	3	1	10	3	1	16
$\alpha = 0.95$	1	3	0	7	2	6	2	7	4	5	6	7	3	1	8	3	0	16

Tabla 5.2: Test de Wilcoxon para resultados de IG-US con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US
Original	-	○	●	○		○		○	○	○	○	○		○		○		○
RUS	●	-	●	●	●	●	●						●	○			●	○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		○
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		○
ENN	●	○	●		●	-	●		○	○	○	○		○				○
OSS		○		○		○	-	○	○	○	○	○		○		○		○
EUS	●		●		●		●	-					●	○			●	○
EE	●		●	●	●	●	●		-			○	●	○			●	
BC	●		●	●	●	●	●			-			●	○			●	
RBT	●		●	●	●	●	●				-		●	○			●	
EEKF	●		●	●	●	●	●					-	●	○			●	
SBC		○	●					○	○	○	○	○	-	○		○		○
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●	
fCBUS			●											○	-		●	
CBIS	●		●		●		●						●	○		-	●	
COSS		○						○	○	○	○	○		○	○	○	-	○
IG-US	●		●	●	●	●	●						●				●	-
$\alpha = 0.9$	1	8	0	4	1	4	1	7	8	8	8	9	1	16	3	7	0	10
$\alpha = 0.95$	1	7	0	4	1	4	0	6	8	8	8	8	1	15	2	6	0	8

Tabla 5.3: Test de Wilcoxon para resultados de IG-US con el clasificador SVM.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US
Original	-	○		○		○		○	○	○	○	○		○	○	○	○	○
RUS	●	-	●	●	●	●	●						●			●	●	
CNN		○	-					○	○	○	○	○		○	○			○
NCL	●	○		-	●			○	○	○	○	○		○	○			○
TL		○		○	-	○		○	○	○	○	○		○	○	○	○	○
ENN	●	○			●	-		○	○	○	○	○		○	○	○		○
OSS		○					-	○	○	○	○	○		○	○	○	○	○
EUS	●		●	●	●	●	●	-		●	●		●			●	●	
EE	●		●	●	●	●	●		-		●		●			●	●	
BC	●		●	●	●	●	●			-	●		●			●	●	
RBT	●		●	●	●	●	●	○	○		-		●		○	●	●	
EEKF	●		●	●	●	●	●					-	●			●	●	
SBC		○						○	○	○	○	○	-	○	○	○		○
CBU	●		●	●	●	●	●						●	-		●	●	
fCBUS	●		●	●	●	●	●						●		-	●	●	
CBIS	●	○			●		●	○	○	○	○	○		○	○	-		○
COSS	●	○						○	○	○	○	○		○	○		-	○
IG-US	●		●	●	●	●	●						●			●	●	-
$\alpha = 0.9$	0	9	0	2	0	2	0	11	10	10	9	9	0	9	10	5	3	9
$\alpha = 0.95$	0	9	0	2	0	2	0	10	10	9	9	9	0	9	9	3	1	9

En general, de las Tablas 5.1-5.3, se puede determinar la competitividad del subgrafo inducido IG-US respecto al resto de técnicas probadas. Retomando los resultados de la Figura 5.1, en específico para los clasificadores J48 y SVM, técnicas como CBU y fCBUS obtuvieron altos índices de precisión. Al comparar la propuesta IG-US con CBU para el clasificador J48, la Tabla 5.2 indica que para ambos niveles de significancia, la propuesta IG-US no presenta diferencias estadísticamente significativas, es decir, no existe diferencia entre usar el método IG-US o el algoritmo CBU cuyos índices de precisión pueden considerarse equivalentes. Algo similar sucede comparando la propuesta con el método fCBUS para el clasificador SVM (Tabla 5.3), de tal modo que se puede concluir que la propuesta IG-US es competitiva para el tratamiento del desbalance entre clases.

5.2. Rendimiento del árbol de expansión mínimo (MIST-US)

Con el propósito de determinar si es mejor mantener las instancias cercanas a la frontera de clase o las más alejadas para el tratamiento del desbalance, en esta sección se analiza el rendimiento del método MIST-US con respecto al resto de técnicas del estado del arte para bajo-muestreo.

La gráfica de barras mostrada en la Figura 5.3 muestra los resultados de la media geométrica obtenidos por los métodos de bajo-muestreo, así como los resultados para el conjunto de datos sin preprocesamiento.

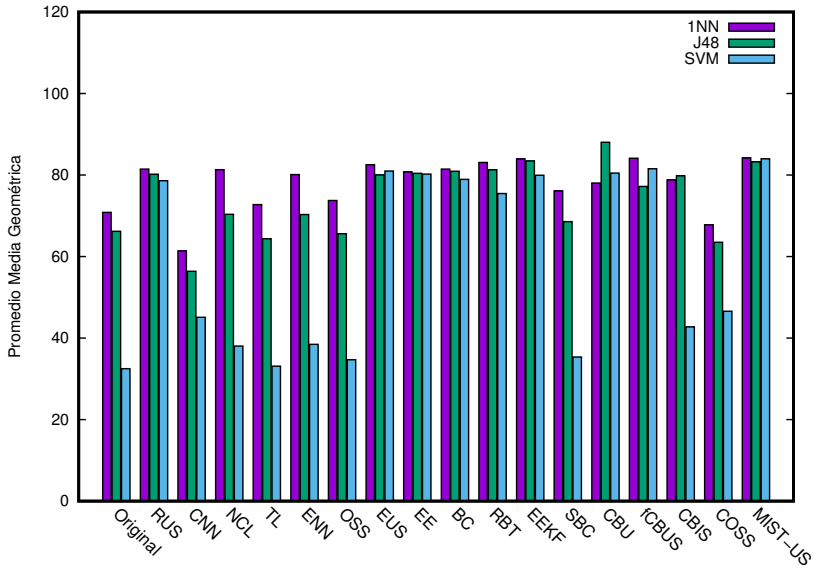


Figura 5.3: Comparativa de MIST-US con el resto de métodos respecto al promedio de la media geométrica obtenida.

De los resultados de la Figura 5.3 se puede observar que de manera general, independientemente del clasificador usado, la propuesta MIST-US genera resultados con precisión mayores al 80 %. Este comportamiento es el mejor para los clasificadores 1NN y SVM, superando a técnicas basadas en vecindario, ensembles y *clustering*, lo que sugiere que, al mantener instancias lo más alejadas de la frontera de clase es capaz de proporcionar una tasa de clasificación suficientemente alta.

Un caso particular se encuentra en el método CBU, ya que para el clasificador J48, el rendimiento obtenido supera a la propuesta MIST-US, esto puede indicar que, el generar un conjunto de datos donde la clase mayoritaria se integra a partir de los centros de los grupos creados por el algoritmo K-means, mantienen una mejor representación de la clase mayoritaria.

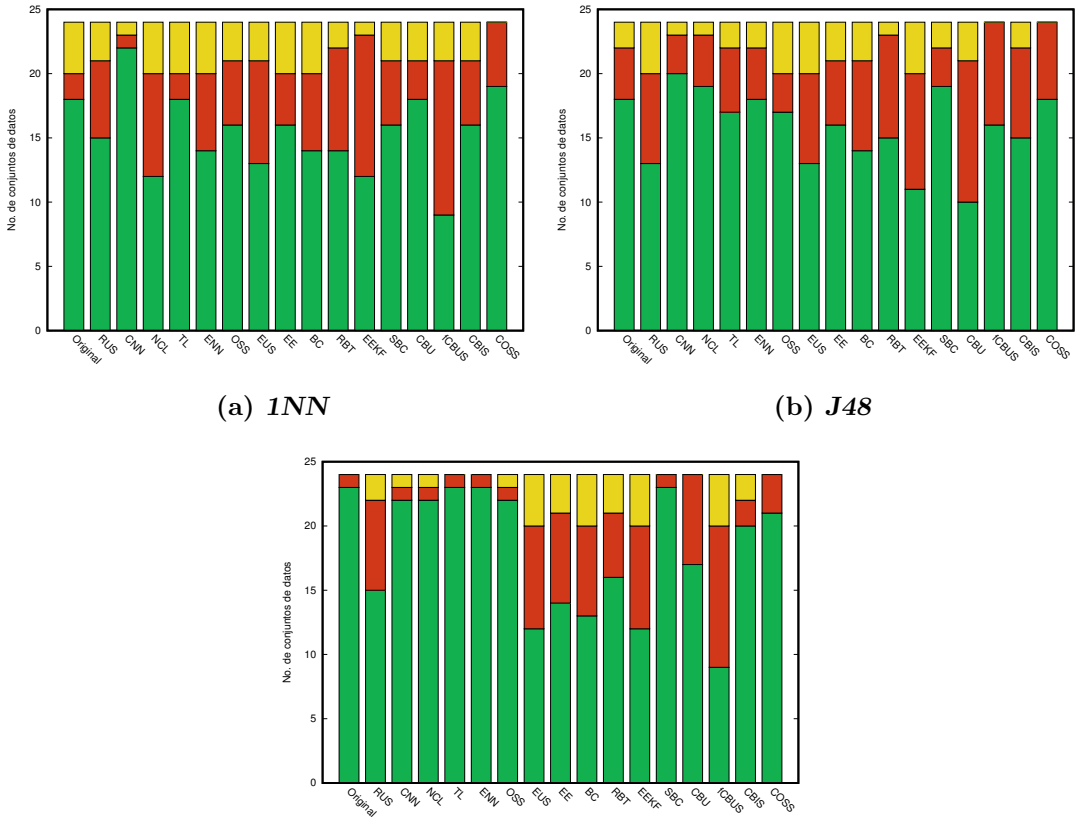
En las gráficas de la Figura 5.4 se presenta en resumen el número de conjuntos de datos para los cuales la propuesta MIST-US se conduce mejor (verde), igual (amarillo) o peor (rojo) comparado con el resto de las técnicas.

De la Figura 5.4 se puede resaltar que para los tres clasificadores utilizados, en la mayoría de métodos comparados, MIST-US es significativamente mejor en la mitad de conjuntos de datos probados con respecto a las técnicas de preprocesamiento empleadas. En particular, se puede observar que en técnicas basadas en vecindario, ensembles e inclusive basadas en *clustering*, MIST-US obtiene en al menos un par de conjuntos resultados similares (amarillo), como es el caso de CNN y EEKF para el clasificador 1NN, y los métodos CNN, NCL y RBT para el clasificador J48.

Para el clasificador SVM, el comportamiento cambia, pues en 11 de los 16 métodos comparados, MIST-US tiene en al menos un par de conjuntos, similares resultados. Un caso peculiar, es la comparativa con respecto al método COSS, pues en todos los clasificadores, en más de la mitad de todos los conjuntos de datos reales probados, MIST-US es significativamente mejor. Otro punto importante, es que en a lo sumo 10 conjuntos de datos MIST-US genera un rendimiento por debajo de fCBUS, en los clasificadores 1NN y SVM.

Esto último sugiere que las técnicas de agrupamiento limpian el conjunto de datos, ayudando al tratamiento de desbalance de clases, no obstante, es evidente la falta de tratamiento de otras complejidades en algunos conjuntos de datos, dado que el sesgo de la clase mayoritaria sigue presente en el proceso de aprendizaje (ver Sección 5.3) .

Para validar las mejoras mencionadas, las Tablas 5.4, 5.5 y 5.6 resumen la prueba de Wilcoxon para cada par de técnicas probadas, para valores de $\alpha = 0.9$ y $\alpha = 0.95$.



(a) ***1NN***

(b) ***J48***

(c) ***SVM***

Figura 5.4: Comparativa de MIST-US con el resto de métodos con respecto al número de conjuntos de datos en los que MIST-US fue mejor (verde), igual (amarillo) o peor (rojo).

Tabla 5.4: Test de Wilcoxon para resultados de MIST-US con el clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	MIST-US
Original	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○
RUS	●	-	●		●												●	
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NCL	●		●	-	●		●						●	●		●	●	
TL	●	○	●	○	-	○		○	○	○	○	○	○		○	○		○
ENN	●		●		●	-	●						●	●	○		●	○
OSS	●		●	○		○	-	○	○	○	○	○			○			○
EUS	●		●		●		●	-					●	●			●	
EE	●		●		●				-						○		●	○
BC	●		●		●		●			-							●	
RBT	●		●		●		●				-			●		●	●	
EEKF	●		●		●		●					-	●	●		●	●	
SBC	●		●	○	●	○		○				○	-		○			○
CBU			●	○				○			○	○		-	○			○
fCBUS	●		●		●		●						●	●	-	●	●	
CBIS	●		●		●										○	-		○
COSS				○		○		○	○	○	○	○			○		-	○
MIST-US	●		●		●		●		●				●	●		●	●	-
$\alpha = 0.9$	1	4	0	8	2	7	2	7	5	5	7	8	3	1	10	3	1	10
$\alpha = 0.95$	1	3	0	7	2	6	2	7	4	5	6	7	3	1	8	3	0	9

Tabla 5.5: Test de Wilcoxon para resultados de MIST-US con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	MIST-US
Original	-	○	●	○		○		○	○	○	○	○		○		○		○
RUS	●	-	●	●	●	●	●						●	○			●	○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		○
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		○
ENN	●	○	●		●	-	●		○	○	○	○		○				○
OSS		○		○		○	-	○	○	○	○	○		○		○		○
EUS	●		●		●		●	-					●	○			●	
EE	●		●	●	●	●	●		-			○	●	○			●	○
BC	●		●	●	●	●	●			-			●	○			●	
RBT	●		●	●	●	●	●				-		●	○			●	
EEKF	●		●	●	●	●	●					-	●	○			●	
SBC		○	●					○	○	○	○	○	-	○		○		○
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●	
fCBUS			●											○	-		●	
CBIS	●		●		●		●						●	○		-	●	
COSS		○						○	○	○	○	○		○	○	○	-	○
MIST-US	●		●	●	●	●	●		●				●				●	-
$\alpha = 0.9$	1	8	0	4	1	4	1	7	8	8	8	9	1	16	3	7	0	10
$\alpha = 0.95$	1	7	0	4	1	4	0	6	8	8	8	8	1	15	2	6	0	9

Tabla 5.6: Test de Wilcoxon para resultados de MIST-US con el clasificador SVM.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	MIST-US
Original	-	○		○		○		○	○	○	○	○		○	○	○	○	○
RUS	●	-	●	●	●	●	●						●			●	●	○
CNN		○	-					○	○	○	○	○		○	○			○
NCL	●	○		-	●			○	○	○	○	○		○	○			○
TL		○		○	-	○		○	○	○	○	○		○	○	○	○	○
ENN	●	○			●	-		○	○	○	○	○		○	○	○		○
OSS		○					-	○	○	○	○	○		○	○	○	○	○
EUS	●		●	●	●	●	●	-		●	●		●			●	●	
EE	●		●	●	●	●	●		-		●		●			●	●	○
BC	●		●	●	●	●	●			-	●		●			●	●	○
RBT	●		●	●	●	●	●	○	○		-		●		○	●	●	○
EEKF	●		●	●	●	●	●					-	●			●	●	
SBC		○						○	○	○	○	○	-	○	○	○		○
CBU	●		●	●	●	●	●						●	-		●	●	
fCBUS	●		●	●	●	●	●						●		-	●	●	
CBIS	●	○			●		●	○	○	○	○	○		○	○	-		○
COSS	●	○						○	○	○	○	○		○	○		-	○
MIST-US	●	●	●	●	●	●	●				●		●			●	●	-
$\alpha = 0.9$	0	9	0	2	0	2	0	11	10	10	9	9	0	9	10	5	3	13
$\alpha = 0.95$	0	9	0	2	0	2	0	10	10	9	9	9	0	9	9	3	1	11

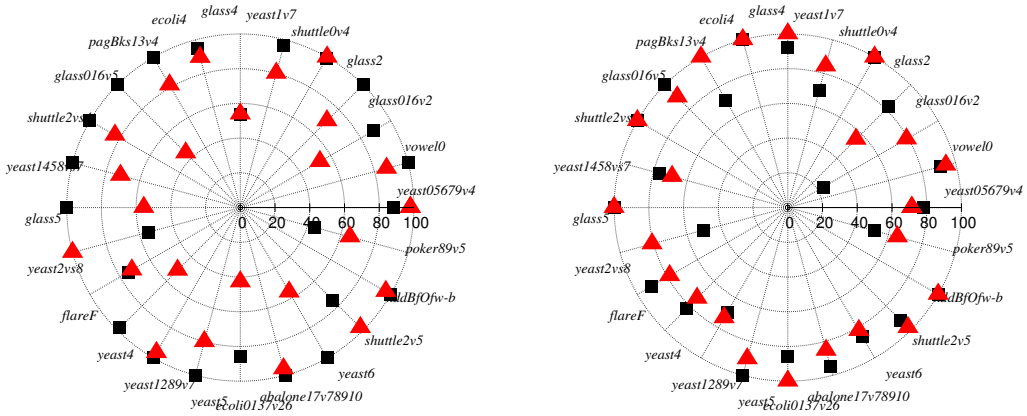
En resumen, se puede observar que en ambos niveles de significancia, con respecto al clasificador 1NN, entre los métodos NCL, EUS y MIST-US no existe diferencia estadísticamente significativa en sus resultados. Algo semejante sucede con algunos métodos basados en ensembles, BC, RBT, EEKF y el basado en *clustering* (fCBUS). En cambio, para el clasificador J48, los métodos con los que no existe diferencia significativa con respecto a MIST-US son, EUS, BC, RBT, EEKF, CBU, fCBUS, CBIS, mientras que para el clasificador SVM, son EUS, EEKF, CBU y fCBUS.

De igual modo, los métodos basados en ensembles (BC y EEKF, específicamente) son métodos que obtienen similares resultados a la propuesta MIST-US, al igual que fCBUS como método representativo basado en *clustering*. Por lo tanto, se deduce que MIST-US es una alternativa diferente, basada en grafos y viable para el tratamiento de desbalance de clases, presentando resultados competitivos a métodos bien conocidos del estado del arte.

5.3. Análisis de rendimiento por clase

Tomando de manera particular los resultados al analizar el comportamiento de los métodos probados con respecto al clasificador SVM (ver Apéndice D), se resalta que para los métodos CNN, NCL, TL, ENN, OSS y SBC, la media geométrica obtenida en la mayoría de los casos fue similar al caso base (conjunto de datos sin preprocesar), este comportamiento indica que estos algoritmos de bajo-muestreo no manejan adecuadamente el problema de desbalance de clases. En contraste, las propuestas basadas en grafos obtienen porcentajes de rendimiento por arriba del 70 %, lo que indica un adecuado equilibro entre las tasas de sensibilidad y especificidad.

En consecuencia, la Figura 5.5 muestra la precisión por clase para el clasificador SVM, en cada conjunto de datos reales, para las propuestas basadas en grafos, mientras que los resultados de métodos del estado del arte, así como caso base se encuentran en el Apéndice D. El resultado de TNR (Especificidad) se ilustra en triángulos rojos y el resultado de TPR (Sensibilidad) en cuadrados negros.



(a) ***IG-US*** **(b)** ***MIST-US***

Figura 5.5: Precisión por clase para conjuntos de datos reales con métodos, IG-US y MIST-US

Como se observa en la Figura 5.5 se obtienen resultados para ambos factores (TNR y TPR) equiparables, lo que indica un adecuado tratamiento del desbalance de clases, sin pérdida de rendimiento en el clasificador SVM, no así para métodos como CNN, NCL, TL, ENN, OSS, RBT, SBC y CBIS el tratamiento de desbalance no es el adecuado, ya que de las figuras mostradas en el Apéndice D se puede observar que el desempeño de la mayoría de los métodos obtienen un TPR igual a 0, lo que implicaría que el modelo SVM clasificó de manera errónea todas las instancias de clase minoritaria.

Capítulo 6

Rendimiento de DBSCAN como estrategia de limpieza

En el estado del arte incluido en el Capítulo 2, se han presentado algunos métodos de bajo-muestreo que se basan en la limpieza del conjunto de datos. Siguiendo esta misma estrategia, se han propuesto varios enfoques basados en *clustering* para reducir el tamaño de la clase mayoritaria, pudiéndose aprovechar de este modo de la eliminación del posible ruido producido por las instancias de esta clase.

En la Sección 3.2 se propuso al algoritmo DBSCAN como estrategia de limpieza del conjunto de datos, de modo que se pueda abordar como técnica para el tratamiento del desbalance entre clases. En consecuencia, el resto de este capítulo se dedica a presentar un análisis comparativo de los resultados obtenidos por DBSCAN con otros algoritmos del estado del arte que tratan el desbalance. Al igual que se hizo en el Capítulo 5, los resultados aquí mostrados se refieren a la precisión en términos de la media geométrica de cada clasificador. La última parte del capítulo presenta una discusión del rendimiento general de las diferentes propuestas de esta tesis para bajo-muestreo, pudiéndose consultar la totalidad de los resultados en el Apéndice B.

6.1. Rendimiento de DBSCAN

El objetivo de esta propuesta es aprovechar las características del algoritmo de *clustering* DBSCAN para eliminar del conjunto de datos instancias consideradas como ruido.

La Figura 6.1 muestra la media geométrica obtenida por los clasificadores después de haber aplicado cada uno de los métodos de bajo-muestreo. Como se puede observar, para los clasificadores 1NN y J48, DBSCAN obtiene un promedio de la media geométrica en los conjuntos de datos superior al 50 %, mejorando al conjunto de datos sin preprocesar. Por otra parte, también presenta un mejor rendimiento que los métodos basados en vecindad como CNN y TL, mientras que para los métodos basados en *clustering* como CBIS, en concreto para el clasificador SVM, DBSCAN obtiene resultados equiparables.

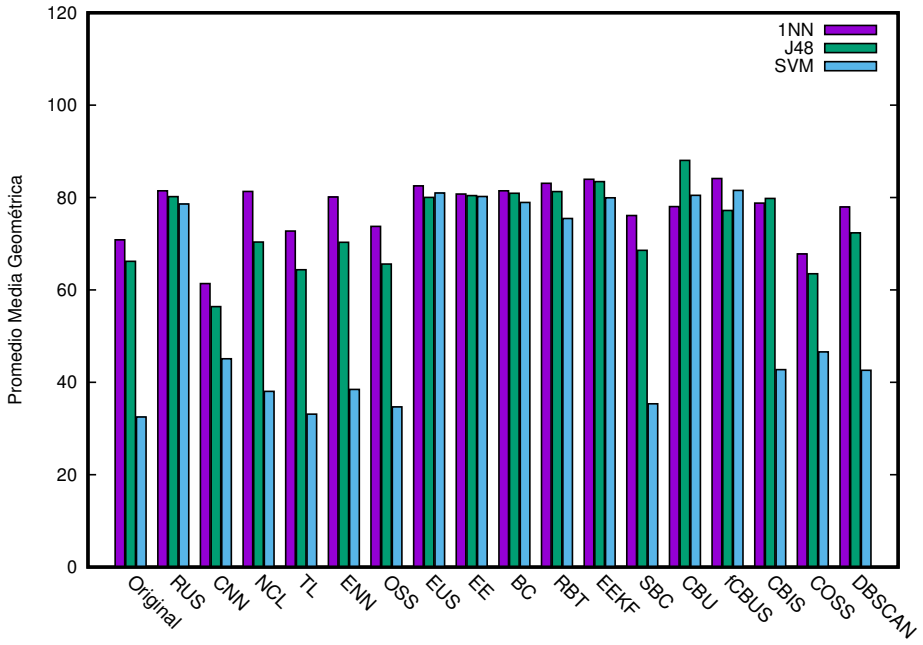
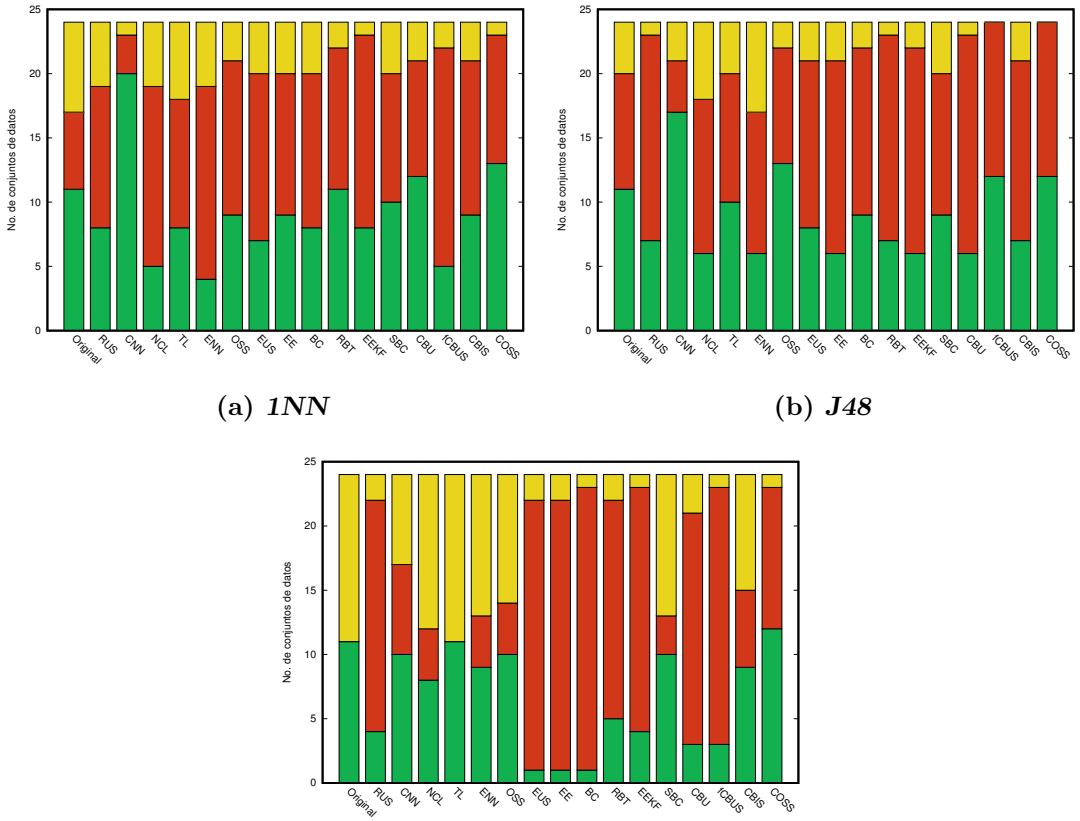


Figura 6.1: Comparativa de DBSCAN con el resto de métodos respecto al promedio de la media geométrica obtenida.

La Figura 6.2 resume el análisis comparativo de DBSCAN con el resto de métodos con respecto al número de conjuntos de datos, para los cuales éste fue mejor (verde), igual (amarillo) o peor (rojo). En estas gráficas, es claramente visible que en más de la mitad de los conjuntos de datos DBSCAN tiene menor precisión respecto a otros métodos. Un ejemplo claro de esto se encuentra en los resultados del clasificador SVM (Figura 6.2 (c)), donde DBSCAN obtuvo mejores resultados que los métodos basados en ensembles como EUS, EE y BC únicamente en dos conjuntos de datos y en otros dos conjuntos produjo resultados similares. Este comportamiento sugiere que los conjuntos de datos tratados por DBSCAN mantienen una densidad en la clase mayoritaria que afecta a la representatividad de la clase minoritaria.



(a) ***1NN***

(b) ***J48***

(c) ***SVM***

Figura 6.2: Comparativa de DBSCAN con el resto de métodos con respecto al número de conjuntos de datos en los que DBSCAN fue mejor (verde), igual (amarillo) o peor (rojo).

Para determinar la significancia estadística de las mejoras reportadas, las Tablas 6.1, 6.2 y 6.3 muestran en resumen los resultado obtenidos por el test de Wilcoxon, para valores de $\alpha = 0.9$ (mitad superior de la diagonal de la tabla) y $\alpha = 0.95$, donde, el símbolo "$\bullet$" indica que el método de la fila fue significativamente mejor que el método de la columna, mientras que, el símbolo "$\circ$" implica que el método de la columna funcionó significativamente mejor que el método de la fila.

Respecto a la capacidad de DBSCAN para tratar el desbalance de clases, se puede deducir que para el clasificador 1NN con varios métodos tales como RUS, TL, OSS, EE, BC, RBT, EEKF, SBC, CBU, CBIS y COSS, DBSCAN es competitivo, pues no presenta diferencia significativa en los resultados que obtiene.

Por otro lado, para el clasificador J48 en métodos como NCL, TL, ENN, OSS, EUS, BC, SBC, fCBUS y COSS se comporta de manera similar al clasificador 1NN. En tanto que con el clasificador SVM, en ambos niveles de significancia en los métodos CNN, NCL, ENN CBIS y COSS, DBSCAN mantiene un resultado similar.

Tabla 6.1: Test de Wilcoxon para resultados de DBSCAN con el clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBSCAN
Original	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		
RUS	●	-	●		●												●	
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NCL	●		●	-	●		●						●	●		●	●	●
TL	●	○	●	○	-	○		○	○	○	○	○	○		○	○		
ENN	●		●		●	-	●						●	●	○		●	●
OSS	●		●	○		○	-	○	○	○	○	○			○			
EUS	●		●		●		●	-					●	●			●	●
EE	●		●		●				-						○		●	
BC	●		●		●		●			-							●	
RBT	●		●		●		●				-			●		●	●	
EEKF	●		●		●		●					-	●	●		●	●	●
SBC	●		●	○	●	○		○				○	-		○			
CBU			●	○				○			○	○		-	○			
fCBUS	●		●		●		●						●	●	-	●	●	●
CBIS	●		●		●										○	-		
COSS				○		○		○	○	○	○	○			○		-	
DBSCAN			●	○		○						○			○			-
$\alpha = 0.9$	1	4	0	9	2	8	2	8	5	5	7	9	3	1	11	3	1	1
$\alpha = 0.95$	1	3	0	8	2	7	2	7	4	5	6	8	3	1	9	3	0	1

Tabla 6.2: Test de Wilcoxon para resultados de DBSCAN con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBSCAN
Original	-	○	●	○		○		○	○	○	○	○		○		○		
RUS	●	-	●	●	●	●	●						●	○			●	●
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		
ENN	●	○	●		●	-	●		○	○	○	○		○				
OSS		○		○		○	-	○	○	○	○	○		○		○		
EUS	●		●		●		●	-					●	○			●	
EE	●		●	●	●	●	●		-			○	●	○			●	●
BC	●		●	●	●	●	●			-			●	○			●	
RBT	●		●	●	●	●	●				-		●	○			●	●
EEKF	●		●	●	●	●	●					-	●	○			●	●
SBC		○	●					○	○	○	○	○	-	○		○		
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●	●
fCBUS			●											○	-		●	
CBIS	●		●		●		●						●	○		-	●	●
COSS		○						○	○	○	○	○		○	○	○	-	
DBSCAN			●						○		○	○		○		○		-
$\alpha = 0.9$	1	9	0	4	1	4	1	7	9	8	9	10	1	17	3	8	0	1
$\alpha = 0.95$	1	7	0	4	1	4	0	6	9	8	9	9	1	16	2	7	0	1

Tabla 6.3: Test de Wilcoxon para resultados de DBSCAN con el clasificador SVM.

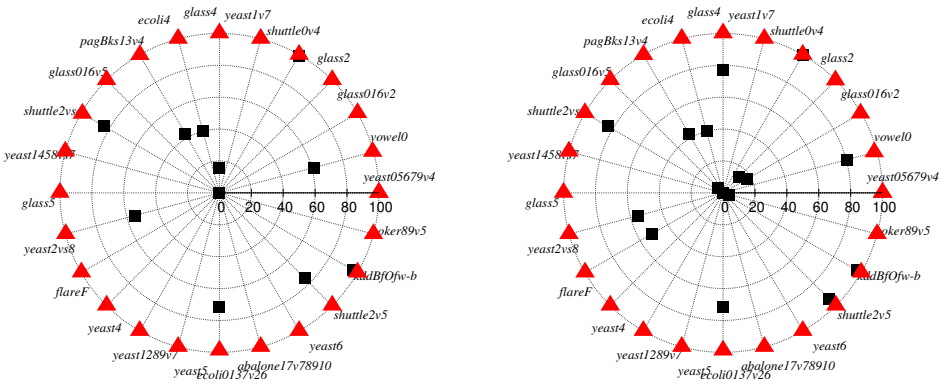
	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBSCAN
Original	-	○		○		○		○	○	○	○	○		○	○	○	○	○
RUS	●	-	●	●	●	●	●						●			●	●	●
CNN		○	-					○	○	○	○	○		○	○			
NCL	●	○		-	●			○	○	○	○	○		○	○			
TL		○		○	-	○		○	○	○	○	○		○	○	○	○	○
ENN	●	○			●	-		○	○	○	○	○		○	○	○		
OSS		○					-	○	○	○	○	○		○	○	○	○	○
EUS	●		●	●	●	●	●	-		●	●		●			●	●	●
EE	●		●	●	●	●	●		-		●		●			●	●	●
BC	●		●	●	●	●	●			-	●		●			●	●	●
RBT	●		●	●	●	●	●	○	○		-		●		○	●	●	●
EEKF	●		●	●	●	●	●					-	●			●	●	●
SBC		○						○	○	○	○	○	-	○	○	○		○
CBU	●		●	●	●	●	●						●	-		●	●	●
fCBUS	●		●	●	●	●	●						●		-	●	●	●
CBIS	●	○			●		●	○	○	○	○	○		○	○	-		
COSS	●	○						○	○	○	○	○		○	○		-	
DBSCAN	●	○			●			○	○	○	○	○		○	○			-
$\alpha = 0.9$	0	10	0	2	0	2	0	12	11	11	10	10	0	10	11	5	3	4
$\alpha = 0.95$	0	10	0	2	0	2	0	11	11	10	10	10	0	10	10	3	1	2

A modo de conclusión, se puede decir que utilizar DBSCAN para limpiar la clase mayoritaria en problemas de desbalance de clases, es una técnica competitiva, pues

se toma como ventaja la eliminación de datos considerados como ruido. No obstante, el problema de desbalance de clases persiste, por lo que se requiere de estrategias adicionales que apoyen a mejorar el rendimiento de DBSCAN.

6.2. Análisis de rendimiento por clase

De igual manera que en el Capítulo 5, se presenta un análisis de los resultados obtenidos por la sensibilidad y especificidad del método DBSCAN para el tratamiento de desbalance de clases. En consecuencia, la Figura 6.3 ilustra el resultado de especificidad (TNR) en triángulos rojos y en cuadrados negros el resultado de Sensibilidad (TPR).



(a) ***Datos reales*** **(b)** ***DBSCAN***

Figura 6.3: Precisión por clase para conjuntos de datos reales sin tratamiento y tratados por el método DBSCAN

A partir de la Figura 6.3, se observa que DBSCAN no afecta la precisión de la clase mayoritaria pese a la disminución de instancias consideradas como ruido, pero se tiene una mejora en precisión de la clase minoritaria dado que se limpia la frontera de decisión. No obstante, puesto que DBSCAN no termina de dar tratamiento al problema de desbalance de clases, se requiere una estrategia adicional para mejorar el rendimiento de DBSCAN, tal como la incorporación de alguna de las técnicas basadas en grafos que se proponen en las Secciones 3.1.1 y 3.1.2.

6.3. Discusión de propuestas de bajo-muestreo

En este trabajo de tesis se han presentado los resultados de las propuestas para manejo del desbalance, dos de ellas basadas en grafos y una en un algoritmo de *clustering*. En el Capítulo 5, se analizó de forma separada el comportamiento de propuestas

basadas en grafos, mientras que en éste se analiza la propuesta basada en *clustering* comparándolas con algunas estrategias de bajo-muestreo presentes en el estado del arte.

Los resultados mostraron lo competitivo que pueden ser. No obstante, al analizar su rendimiento de forma conjunta es posible notar que, las propuestas de esta tesis con respecto a las estrategias del estado del arte, muestran un mejor rendimiento en clasificadores como 1NN y J48, pues se obtienen porcentajes hasta más del 70 %. Mientras que, para el clasificador SVM, las propuestas basadas en grafos logran igualar o superar al método fCBUS que se aprecia es competitivo con IG-US pero MIST-US lo supera.

Esto sugiere que para lograr una mejor generalización del conocimiento, independiente del clasificador, técnicas basadas en grafos son igual o mejor que técnicas bien conocidas como RUS, EUS ,EE, BC, EEKF y fCBUS. No obstante, DBSCAN presenta un rendimiento equiparable con las técnicas anteriormente mencionadas, en clasificadores como 1NN y J48, tal como lo muestra la Figura 6.4.

Por otro lado, comparando únicamente las propuestas de esta tesis para el tratamiento de desbalance de clases, como se aprecia en la Figura 6.4, con los conjuntos de datos utilizados las propuestas basadas en grafos superan a la propuesta basada en *clustering*. Es importante resaltar que a pesar de que IG-US tiene mejor rendimiento en clasificadores 1NN y J48, la propuesta MIST-US independientemente del clasificador usado, genera un comportamiento uniforme, que a su vez presenta tasas de efectividad superiores, lo que sugiere que mantener instancias lo más lejanas de la frontera de decisión, mantiene una representación de clase mayoritaria.

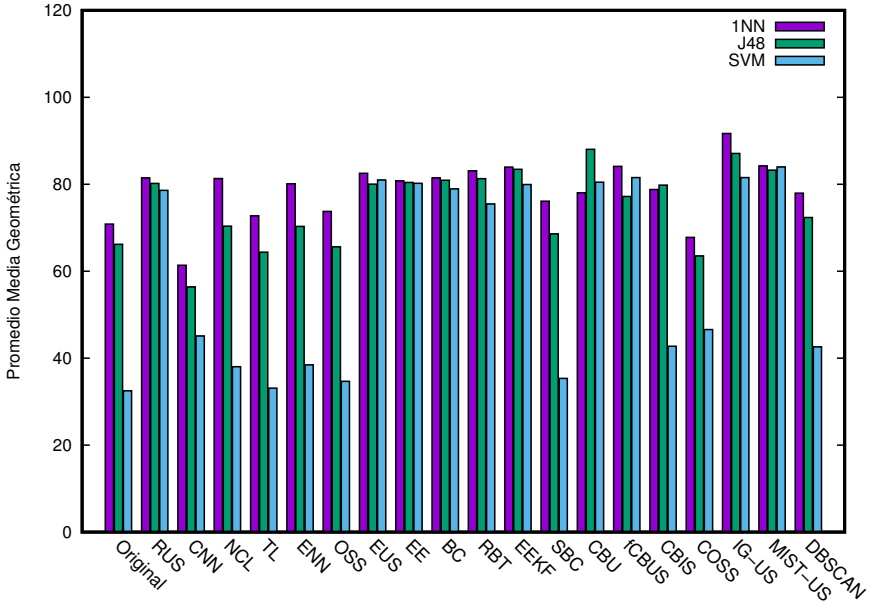


Figura 6.4: Comparativa general de técnicas de bajo-muestreo con respecto a la media geométrica obtenida.

Lo anterior se puede confirmar al aplicar el test de Wilcoxon mostrado en las Ta-

blas 6.4, 6.5 y 6.6, para valores de $\alpha = 0.9$ (diagonal inferior) y $\alpha = 0.95$ (diagonal superior), donde, "$\circ$" indica que el método de la columna funcionó significativamente mejor que el método de la fila y "$\bullet$" representa que el método de la fila fue significativamente mejor que el método de la columna.

En general, se puede observar para el clasificador 1NN, la propuesta IG-US fue significativamente mejor que MIST-US y DBSCAN para un nivel de significancia del 0.9, no así para el nivel 0.95, pues IG-US con respecto a MIST-US son igualmente competitivos. Es importante resaltar que para ambos niveles de significancia, IG-US presenta una importante diferencia significativa en el resto de los métodos probados.

Tabla 6.4: Test de Wilcoxon para resultados de bajo-muestreo con el clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US	MIST-US	DBSCAN
Original	-	$\circ$	$\bullet$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$		$\circ$	$\circ$		$\circ$	$\circ$	
RUS	$\bullet$	-	$\bullet$		$\bullet$												$\bullet$	$\circ$		
CNN	$\circ$	$\circ$	-	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$
NCL	$\bullet$		$\bullet$	-	$\bullet$		$\bullet$						$\bullet$	$\bullet$		$\bullet$	$\bullet$	$\circ$		$\bullet$
TL	$\bullet$	$\circ$	$\bullet$	$\circ$	-	$\circ$		$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$		$\circ$	$\circ$		$\circ$	$\circ$	
ENN	$\bullet$		$\bullet$		$\bullet$	-	$\bullet$						$\bullet$	$\bullet$	$\circ$		$\bullet$	$\circ$	$\circ$	$\bullet$
OSS	$\bullet$		$\bullet$	$\circ$		$\circ$	-	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$			$\circ$			$\circ$	$\circ$	
EUS	$\bullet$		$\bullet$		$\bullet$		$\bullet$	-					$\bullet$	$\bullet$			$\bullet$	$\circ$		$\bullet$
EE	$\bullet$		$\bullet$		$\bullet$				-						$\circ$		$\bullet$	$\circ$	$\circ$	
BC	$\bullet$		$\bullet$		$\bullet$		$\bullet$			-							$\bullet$	$\circ$		
RBT	$\bullet$		$\bullet$		$\bullet$		$\bullet$				-			$\bullet$		$\bullet$	$\bullet$	$\circ$		
EEKF	$\bullet$		$\bullet$		$\bullet$		$\bullet$					-	$\bullet$	$\bullet$		$\bullet$	$\bullet$	$\circ$		$\bullet$
SBC	$\bullet$		$\bullet$	$\circ$	$\bullet$	$\circ$		$\circ$				$\circ$	-		$\circ$			$\circ$	$\circ$	
CBU			$\bullet$	$\circ$				$\circ$			$\circ$	$\circ$		-	$\circ$			$\circ$	$\circ$	
fCBUS	$\bullet$		$\bullet$		$\bullet$		$\bullet$						$\bullet$	$\bullet$	-	$\bullet$	$\bullet$			$\bullet$
CBIS	$\bullet$		$\bullet$		$\bullet$										$\circ$	-		$\circ$	$\circ$	
COSS				$\circ$		$\circ$		$\circ$	$\circ$	$\circ$	$\circ$	$\circ$			$\circ$		-	$\circ$	$\circ$	
IG-US	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$		$\bullet$	$\bullet$	-	$\bullet$	$\bullet$
MIST-US	$\bullet$		$\bullet$		$\bullet$		$\bullet$		$\bullet$				$\bullet$	$\bullet$		$\bullet$	$\bullet$		-	$\bullet$
DBSCAN			$\bullet$	$\circ$		$\circ$						$\circ$			$\circ$			$\circ$	$\circ$	-
$\alpha = 0.9$	1	4	0	9	2	8	2	8	5	5	7	9	3	1	11	3	1	18	11	1
$\alpha = 0.95$	1	3	0	8	2	7	2	7	4	5	6	8	3	1	9	3	0	17	10	1

Para los resultados obtenidos por el clasificador J48 (Tabla 6.5), en ambos niveles de significancia, tanto la propuesta IG-US como MIST-US son competitivos entre ellas y por igual manera con los métodos BC, RBT, EEKF, CBU, fCBUS y CBIS. Adicionalmente IG-US es competitivo con EE y por otro lado, MIST-US es competitivo con EUS. Esto sugiere que para el clasificador SVM, las propuestas IG-US y MIST-US pueden ser usadas para tratar el desbalance de clases y tener un rendimiento similar al método CBU que obtuvo el mejor *rank* (ver Apéndice B Tabla B.5).

Por último, para el clasificador SVM, las propuestas IG-US y MIST-US en ambos niveles de significancia, ambas estrategias obtienen resultados competitivos para los métodos EUS, EEKF y fCBUS. Adicionalmente, IG-US obtiene resultados competitivos con EE, BC, RBT y entre ambas propuestas no existe diferencia significativa. En consecuencia se puede concluir que IG-US es el método más competitivo respecto a técnicas basadas en ensembles, pero ambas propuestas son significativamente mejores en ambos niveles con respecto a técnicas basas en vecindario (NCL, TL, ENN y OSS), no así con DBSCAN que obtiene poca diferencia significativa.

Tabla 6.5: Test de Wilcoxon para resultados de bajo-muestreo con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US	MIST-US	DBSCAN
Original	-	○	●	○		○		○	○	○	○	○		○		○		○	○	
RUS	●	-	●	●	●	●	●						●	○			●	○	○	●
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		○	○	
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		○	○	
ENN	●	○	●		●	-	●		○	○	○	○		○				○	○	
OSS		○		○		○	-	○	○	○	○	○		○		○		○	○	
EUS	●		●		●		●	-					●	○			●	○		
EE	●		●	●	●	●	●		-			○	●	○			●		○	●
BC	●		●	●	●	●	●			-			●	○			●			
RBT	●		●	●	●	●	●				-		●	○			●			●
EEKF	●		●	●	●	●	●					-	●	○			●			●
SBC		○	●					○	○	○	○	○	-	○		○		○	○	
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●			●
fCBUS			●											○	-		●			
CBIS	●		●		●		●						●	○		-	●			●
COSS		○						○	○	○	○	○		○	○	○	-	○	○	
IG-US	●		●	●	●	●	●						●				●	-		●
MIST-US	●		●	●	●	●	●		●				●				●		-	●
DBSCAN			●						○		○	○		○		○		○	○	-
$\alpha = 0.9$	1	9	0	4	1	4	1	7	9	8	9	10	1	17	3	8	0	11	11	1
$\alpha = 0.95$	1	7	0	4	1	4	0	6	9	8	9	9	1	16	2	7	0	9	10	1

Tabla 6.6: Test de Wilcoxon para resultados de bajo-muestreo con el clasificador SVM.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US	MIST-US	DBSCAN
Original	-	○		○		○		○	○	○	○	○		○	○	○	○	○	○	○
RUS	●	-	●	●	●	●	●						●			●	●		○	●
CNN		○	-					○	○	○	○	○		○	○			○	○	
NCL	●	○		-	●			○	○	○	○	○		○	○			○	○	
TL		○		○	-	○		○	○	○	○	○		○	○	○	○	○	○	○
ENN	●	○			●	-		○	○	○	○	○		○	○	○		○	○	
OSS		○					-	○	○	○	○	○		○	○	○	○	○	○	○
EUS	●		●	●	●	●	●	-		●	●		●			●	●			●
EE	●		●	●	●	●	●		-		●		●			●	●		○	●
BC	●		●	●	●	●	●			-	●		●			●	●		○	●
RBT	●		●	●	●	●	●	○	○		-		●		○	●	●		○	●
EEKF	●		●	●	●	●	●					-	●			●	●			●
SBC		○						○	○	○	○	○	-	○	○	○		○	○	○
CBU	●		●	●	●	●	●						●	-		●	●			●
fCBUS	●		●	●	●	●	●						●		-	●	●			●
CBIS	●	○			●		●	○	○	○	○	○		○	○	-		○	○	
COSS	●	○						○	○	○	○	○		○	○		-	○	○	
IG-US	●		●	●	●	●	●						●			●	●	-		●
MIST-US	●	●	●	●	●	●	●				●		●			●	●		-	●
DBSCAN	●	○			●			○	○	○	○	○		○	○			○	○	-
$\alpha = 0.9$	0	10	0	2	0	2	0	12	11	11	10	10	0	10	11	5	3	10	14	4
$\alpha = 0.95$	0	10	0	2	0	2	0	11	11	10	10	10	0	10	10	3	1	10	12	2

Derivado de los análisis anteriores, se concluye que las propuestas IG-US y MIST-US individualmente proporcionan mejores resultados estadísticamente significativos con respecto a técnicas basadas en vecindario, pero competitivas respecto a técnicas basadas en ensembles y *clustering*. Esto último sugiere que al considerar el conjunto de datos como un grafo completo ponderado, logra extraer información relevante para la clasificación. Además, el uso de DBSCAN como técnica de limpieza, favorece los índices de precisión, no obstante, el sesgo de la clase mayoritaria en el proceso de aprendizaje sigue presente.

Capítulo 7

Resultados del tratamiento de desbalance de clases, traslape de clases y/o ruido

La influencia que tienen la presencia de ruido, el solapamiento de clases y el desbalance en los conjuntos datos se encuentra presente en varios problemas del mundo real [76]. Con la intención de mejorar el resultado observado por las estrategias presentadas en el capítulo anterior, en éste se analizan los resultados de las propuestas formadas por etapas: DBIG-US que después de aplicar DBSCAN obtiene un subgrafo inducido y DBMIST-US que después de la ejecución de DBSCAN construye un árbol de expansión mínimo.

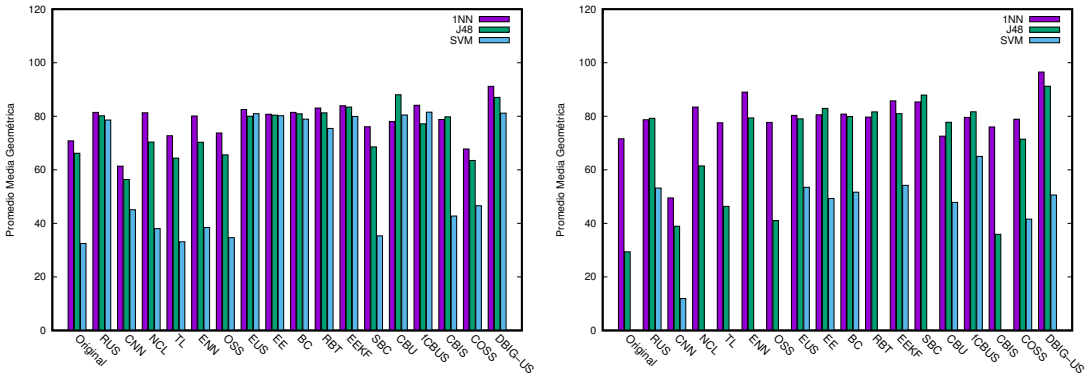
Es importante resaltar que, para determinar si las propuestas son viables sobre conjuntos de datos con problemas de desbalance, solapamiento de clases y/o ruido, además de los 24 conjuntos de datos reales ya utilizados en capítulos anteriores, aquí se emplearon también 12 conjuntos de datos sintéticos. El compendio completo de resultados se encuentran en el Apéndice C.

Para determinar si existen diferencias estadísticamente significativas entre cada par de métodos evaluados, se utiliza la prueba de Wilcoxon, cuyos resultados se han incluido en diversas tablas. En cada una de estas tablas, se muestran dos filas por cada método: la primera se refiere a los resultados obtenidos sobre los conjuntos de datos reales y la segunda a los obtenidos sobre los conjuntos de datos sintéticos. La mitad superior de la diagonal de cada tabla corresponde a los resultados para el valor $\alpha = 0.9$, mientras que la mitad inferior de la diagonal es para un nivel 0.95. El símbolo "$\bullet$" indica que el método de la fila fue significativamente mejor que el método de la columna, mientras que el símbolo "$\circ$" representa que el método de la columna fue significativamente mejor que el método de la fila. La ausencia de cualquiera de los símbolos anteriores implica que los métodos de la columna y de la fila no muestran diferencias estadísticamente significativas.

7.1. Rendimiento de DBIG-US: DBSCAN y subgrafo inducido

DBIG-US es una propuesta que busca el aprovechamiento de DBSCAN para filtrar instancias ruidosas de clase mayoritaria, refinando los conjuntos de datos por medio de la obtención de un subgrafo inducido para equilibrar las clases.

En términos promediales de tasas de rendimiento, las Figura 7.1(a) y 7.1(b) muestran el comportamiento de los métodos con respecto a los conjuntos de datos reales y sintéticos.

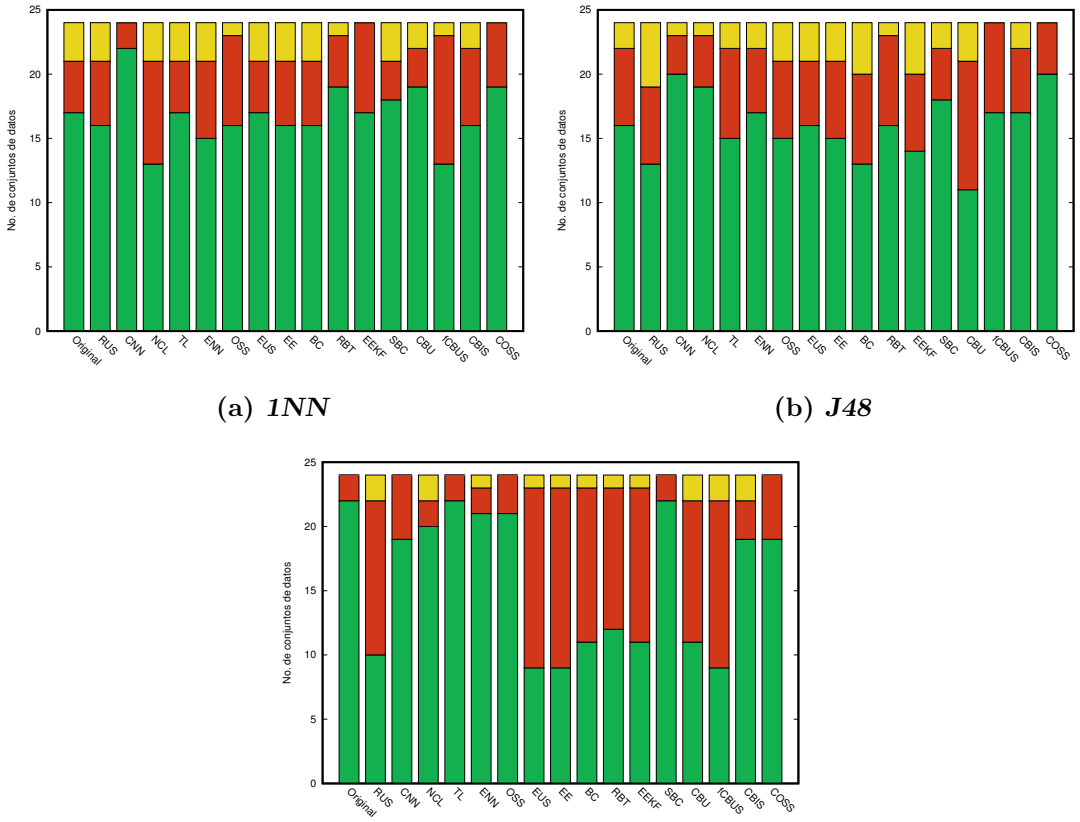


(a) ***Conjuntos de datos reales*** **(b)** ***Conjuntos de datos sintéticos***

Figura 7.1: Comparativa de DBIG-US con el resto de métodos del promedio de la media geométrica obtenida.

De los resultados mostrados en las figuras 7.1(a) y 7.1(b), se puede observar que para los clasificadores 1NN y J48, la propuesta DBIG-US obtiene porcentajes por arriba del 80 %, siendo estos los más altos. No obstante, para el clasificador SVM los resultados logran ser similares a los obtenidos por métodos como EUS, EE y fCBUS, todo esto para conjuntos de datos reales. Mientras que para conjuntos de datos sintéticos, con los clasificadores 1NN y J48, los promedios obtenidos alcanzan más del 85 % de precisión, siendo estos los más altos en comparativa con el resto de métodos probados.

Analizando el comportamiento en cuanto al número de conjuntos para los cuales DBIG-US fue mejor, igual o peor, en las Figuras 7.2 y 7.3 se resume este análisis. Para conjuntos de datos reales, DBIG-US en más del 50 % de los conjuntos de datos reales con respecto a los técnicas probadas es mejor para los clasificadores 1NN y J48. En tanto que, obtiene un rendimiento similar en por lo menos el 12.5 % de conjuntos de datos reales para métodos basados en vecindario (NCL, RL, ENN). De igual forma, para métodos basados en ensembles como EE y BC.



(a) ***1NN***

(b) ***J48***

(c) ***SVM***

Figura 7.2: Cantidad de conjuntos de datos reales en los que DBIG-US fue mejor (verde), igual (amarillo) o peor (rojo).

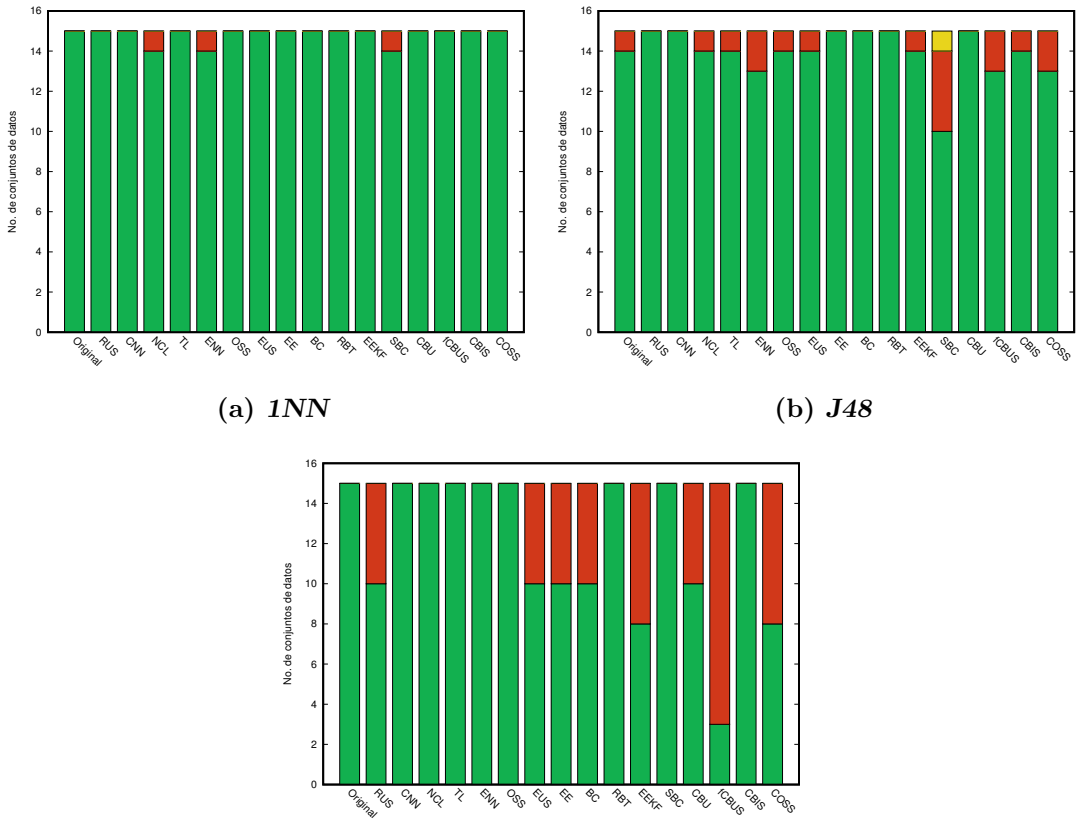
En la gráfica de la Figura 7.2(c) se observa que en métodos basados en ensembles, en más de la mitad de los conjuntos de datos DBIG-US obtuvo un bajo rendimiento, lo que sugiere que DBSCAN pudo haber eliminado algunas instancias útiles para la construcción del grafo.

En general, se muestra que DBIG-US fue mejor que RUSBoost (RBt) y CBU en 19 de los 24 conjuntos de datos reales (79 %), pero fue peor en el resto. DBIG-US también superó a los métodos basados en vecindario, TL y OSS en 17 y 16 conjuntos respectivamente, y fue mejor que RUS en 16 conjuntos de datos (66.6 %). A partir de estos resultados, es posible observar que DBIG-US arrojó un mejor rendimiento de clasificación que técnicas bien conocidas, en la mayoría de conjuntos de datos que utilizan el clasificador 1NN.

Para el clasificador J48 se obtuvieron resultados similares. En primer lugar, DBIG-US fue mejor que NCL en 19 conjuntos de datos reales (79 %). También superó al método SBC en 18 conjuntos de datos (75 %). Mientras que para conjuntos de datos sintéticos, en 66.6 % de los conjuntos de datos tiene mejor desempeño.

Respecto al uso de conjuntos de datos sintéticos, la Figura 7.3 ilustra que DBIG-US obtiene mejor rendimiento, dado que en más del 90 % de los conjuntos de datos sintéticos, DBIG-US obteniene mejor rendimiento para el clasificador 1NN y para el clasificador J48, hasta en un 66 %. No obstante, para el clasificador SVM, en métodos tales como RUS, EUS, EE, BC, EEKF, CBU, fCBUS y COSS, en al menos un 33.3 % de los datos DBIG-US generá un peor rendimiento.

Un caso particular es con respecto al método fCBUS, el cual en sólo un 20 % de los conjuntos de datos tiene el mejor rendimiento, en conjuntos de datos sintéticos. Esto último sugiere que para conjuntos de datos con presencia de ruido o solapamiento de clases, seleccionar instancias cercanas a la frontera de decisión es mejor opción que, seleccionar las más alejadas unas de las otras después de haber limpiado el conjunto de datos.



(a) ***1NN***

(b) ***J48***

(c) ***SVM***

Figura 7.3: Cantidad de conjuntos de datos sintéticos en los que DBIG-US fue mejor (verde), igual (amarillo) o peor (rojo).

El análisis de significancia estadística con el test de Wilcoxon se muestra en las Tablas 7.1, 7.2 y 7.3 para valores de $\alpha = 0.9$ y $\alpha = 0.95$.

El resultado del test para el clasificador 1NN (Tabla 7.1) muestra que la propuesta DBIG-US es significativamente mejor que el resto de los métodos de bajo-muestreo en ambos niveles de significancia tanto para conjuntos de datos reales como para conjuntos de datos sintéticos. Asimismo, tanto el método NCL como ENN son significativamente mejores que RUS, CNN, TL y OSS.

De manera similar que para el clasificador J48, en la Tabla 7.2 se observa que CBU fue el mejor método para $\alpha = 0.95$ estadísticamente igual que DBIG-US para $\alpha = 0.90$. Finalmente, analizando ambos métodos (DBIG-US y CBU) podemos ver que no existen diferencias estadísticas en ambos niveles de significancia. Estos resultados indican que DBIG-US es una propuesta competitiva para enfrentar el problema del desequilibrio de clases.

También cabe mencionar que, para ambos clasificadores, el algoritmo CNN fue significativamente peor que cualquier otro método, lo que revela que la sola eliminación de instancias redundantes de la clase mayoritaria no permite tratar eficazmente el problema del desbalance.

Tabla 7.1: Test de Wilcoxon para resultados del clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US
Original	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○
	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○
RUS	●	-	●		●												●	○
	●	-	●	○		○			○	○		○	○	●		●		○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NCL	●		●	-	●		●						●	●		●	●	○
	●	●	●	-	●	○	●	●	●	●	●	○	○	●	●	●		○
TL	●	○	●	○	-	○		○	○	○	○	○	○		○	○		○
	●		●	○	-	○		○	○	○	○	○	○	●		●		○
ENN	●		●		●	-	●						●	●	○		●	○
	●	●	●	●	●	-	●	●	●	●	●	●	●	●	●	●	●	○
OSS	●		●	○		○	-	○	○	○	○	○			○			○
	●		●	○		○	-			○		○	○	●		●		○
EUS	●		●		●		●	-					●	●			●	○
	●		●	○		○		-				○	○	●		●		○
EE	●		●		●				-						○		●	○
	●	●	●	○		○			-			○	○	●		●		○
BC	●		●		●		●			-							●	○
	●	●	●	○		○				-		○	○	●		●		○
RBT	●		●		●		●				-			●		●	●	○
	●		●	○		○					-	○	○	●		●		○
EEKF	●		●		●		●					-	●	●		●	●	○
	●	●	●		●	○	●	●	●	●	●	-		●	●	●		○
SBC	●		●	○	●	○		○				○	-		○			○
	●	●	●	●	●	○	●	●	●	●	●		-	●	●	●		○
CBU			●	○				○			○	○		-	○			○
		○	●	○	○	○	○	○	○	○	○	○	○	-	○	○		○
fCBUS	●		●		●		●						●	●	-	●	●	
	●		●	○		○						○	○	●	-			○
CBIS	●		●		●										○	-		○
	●		●	○	○	○	○	○	○	○	○	○	○	●		-		○
COSS				○		○		○	○	○	○	○			○		-	○
			●			○											-	○
DBIG-US	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	-
	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-
$\alpha = 0.9$	1	4	0	8	2	7	2	7	5	5	7	8	3	1	10	3	1	16
	1	4	0	12	4	16	4	5	6	7	5	13	13	1	3	3	1	17
$\alpha = 0.95$	1	3	0	7	2	6	2	7	4	5	6	7	3	1	8	3	0	16
	1	3	0	12	4	16	4	4	5	5	4	12	13	1	3	3	1	17

Por último en la Tabla 7.3 se resumen los resultados obtenidos del test de Wilcoxon para el clasificador SVM. En este caso, el mejor método fue fCBUS, sin embargo, cabe resaltar que para el nivel $\alpha = 0.9$, DBIG-US es competitivo en conjuntos de datos reales. También es importante mencionar que, para métodos como RUS, EUS, EE, BC, EEKF

y CBU en ambos niveles de significancia y tanto para conjuntos de datos reales como sintéticos, DBIG-US no presenta diferencia estadísticamente significativa, por lo tanto puede ser competitivo con algoritmos aleatorios y con técnicas basadas en ensembles.

Tabla 7.2: Test de Wilcoxon para resultados de DBIG-US con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US
Original	-	○	●	○		○		○	○	○	○	○		○		○		○
	-	○		○	○	○	○	○	○	○	○	○	○	○	○		○	○
RUS	●	-	●	●	●	●	●						●	○			●	○
	●	-	●	●	●		●		○		○		○		○	●		○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○
		○	-	○		○		○	○	○	○	○	○	○	○		○	○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		○
	●	○	●	-	●	○	●	○	○	○	○	○	○	○	○	●		○
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		○
	●	○		○	-	○		○	○	○	○	○	○	○	○	●	○	○
ENN	●	○	●		●	-	●		○	○	○	○		○				○
	●		●	●	●	-	●						○			●		○
OSS		○		○		○	-	○	○	○	○	○		○		○		○
	●	○		○		○	-	○	○	○	○	○	○	○	○		○	○
EUS	●		●		●		●	-					●	○			●	○
	●		●	●	●		●	-	○		○		○			●		○
EE	●		●	●	●	●	●		-			○	●	○			●	○
	●	●	●	●	●		●	●	-	●			○	●		●	●	○
BC	●		●	●	●	●	●			-			●	○			●	○
	●		●	●	●		●		○	-			○			●		○
RBT	●		●	●	●	●	●				-		●	○			●	○
	●	●	●	●	●		●				-		○	●		●		○
EEKF	●		●	●	●	●	●					-	●	○			●	○
	●		●	●	●		●					-	○	●		●		○
SBC		○	●					○	○	○	○	○	-	○		○		○
	●	●	●	●	●	●	●	●	●	●	●	●	-	●	●	●	●	○
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●	
	●		●	●	●		●		○		○		○	-	○	●		○
fCBUS			●											○	-		●	○
	●		●	●	●		●						○		-	●		○
CBIS	●		●		●		●						●	○		-	●	○
		○		○	○	○		○	○	○	○	○	○	○	○	-	○	○
COSS		○						○	○	○	○	○		○	○	○	-	○
	●		●		●		●						○			●	-	○
DBIG-US	●	●	●	●	●	●	●	●	●		●		●			●	●	-
	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	-
$\alpha = 0.9$	1	8	0	4	1	4	1	7	8	8	8	9	1	16	3	7	0	16
	0	6	0	5	2	6	1	6	11	6	9	7	16	6	8	0	5	17
$\alpha = 0.95$	1	7	0	4	1	4	0	6	8	8	8	8	1	15	2	6	0	13
	0	6	0	5	2	6	1	6	10	6	8	6	16	6	6	0	5	16

Tabla 7.3: Test de Wilcoxon para resultados de DBIG-US con el clasificador SVM.

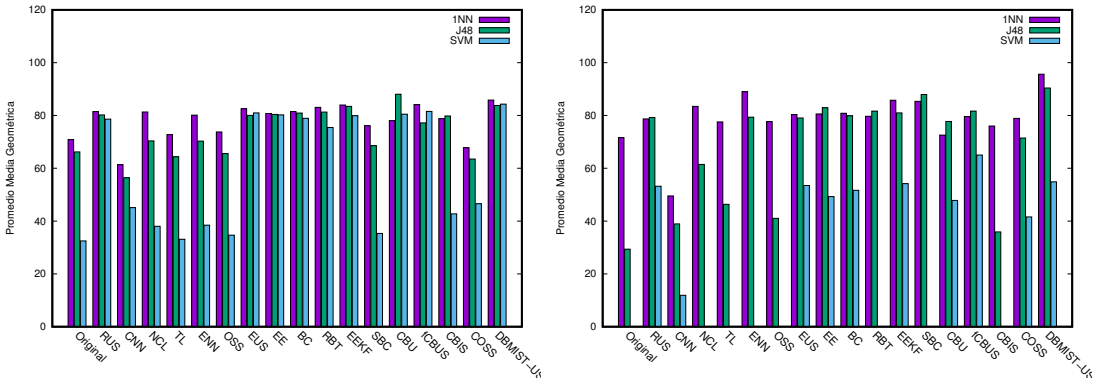
	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US
Original	-	○		○		○		○	○	○	○	○		○	○	○	○	○
	-	○	○					○	○	○		○		○	○		○	○
RUS	•	-	•	•	•	•	•						•			•	•	
	•	-	•	•	•	•	•				•		•	•	○	•		
CNN		○	-					○	○	○	○	○		○	○			○
		○	-	•	•	•	•	○	○	○	•	○	•	○	○	•	○	○
NCL	•	○		-	•			○	○	○	○	○		○	○			○
		○		-				○	○	○		○		○	○		○	○
TL		○		○	-	○		○	○	○	○	○		○	○	○	○	○
		○			-			○	○	○		○		○	○		○	○
ENN	•	○			•	-		○	○	○	○	○		○	○	○		○
		○				-		○	○	○		○		○	○		○	○
OSS		○					-	○	○	○	○	○		○	○	○	○	○
		○					-	○	○	○		○		○	○		○	○
EUS	•		•	•	•	•	•	-		•	•		•			•	•	
	•		•	•	•	•	•	-	•		•		•	•	○	•		
EE	•		•	•	•	•	•		-		•		•			•	•	
	•		•	•	•	•	•	○	-		•		•		○	•		
BC	•		•	•	•	•	•			-	•		•			•	•	
	•		•	•	•	•	•			-	•		•		○	•		
RBT	•		•	•	•	•	•	○	○		-		•		○	•	•	
		○						○	○	○	-	○		○	○		○	○
EEKF	•		•	•	•	•	•					-	•			•	•	
	•		•	•	•	•	•				•	-	•	•	○	•		
SBC		○						○	○	○	○	○	-	○	○	○		○
		○						○	○	○		○	-	○	○		○	○
CBU	•		•	•	•	•	•						•	-		•	•	
	•	○	•	•	•	•	•	○			•	○	•	-	○	•		
fCBUS	•		•	•	•	•	•						•		-	•	•	
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	-	•	•	•
CBIS	•	○			•		•	○	○	○	○	○		○	○	-		○
		○						○	○	○		○		○	○	-	○	○
COSS	•	○						○	○	○	○	○		○	○		-	○
	•		•	•	•	•	•				•		•			•	-	
DBIG-US	•		•	•	•	•	•						•			•	•	-
	•		•	•	•	•	•				•		•		○	•		-
$\alpha = 0.9$	0	9	0	2	0	2	0	11	10	10	9	9	0	9	10	5	3	9
	0	10	8	0	0	0	0	11	9	9	0	10	0	9	17	0	9	9
$\alpha = 0.95$	0	9	0	2	0	2	0	10	10	9	9	9	0	9	9	3	1	9
	0	10	0	0	0	0	0	11	9	9	0	10	0	9	16	0	9	9

7.2. Rendimiento de DBMIST-US: DBSCAN y árbol de expansión mínimo

DBMIST-US es una propuesta que combina la limpieza del conjunto de datos con el algoritmo DBSCAN y la construcción de un árbol mínimo de expansión, para tratar el problema de desbalance de clases.

El rendimiento de los clasificadores en términos de media geométrica, se puede apreciar en la Figura 7.4. Como puede verse, para conjuntos de datos reales, la propuesta DBMIST-US, independientemente del clasificador que se utilice, obtiene porcentajes por arriba del 80 % de precisión.

No obstante, para los conjuntos de datos sintéticos clasificadores como 1NN y J48, DBMIST-US genera los promedios más altos, por arriba del 85 %, no así para el clasificador SVM, donde el método fCBUS obtiene en promedio 65 %, mientras que la propuesta genera un promedio del 54.88 %, siendo el segundo método más alto. Esto indica que, para conjuntos de datos sintéticos, DBMIST-US obtiene resultados no distantes a los mejores (Tabla C.6). Por ejemplo, para el conjunto *subcl0* se reporta para fCBUS 70.7, mientras que para DBMIST-US 67.5, la diferencia no excede a un 5 %, por lo que es importante determinar si DBMIST-US es competitivo.



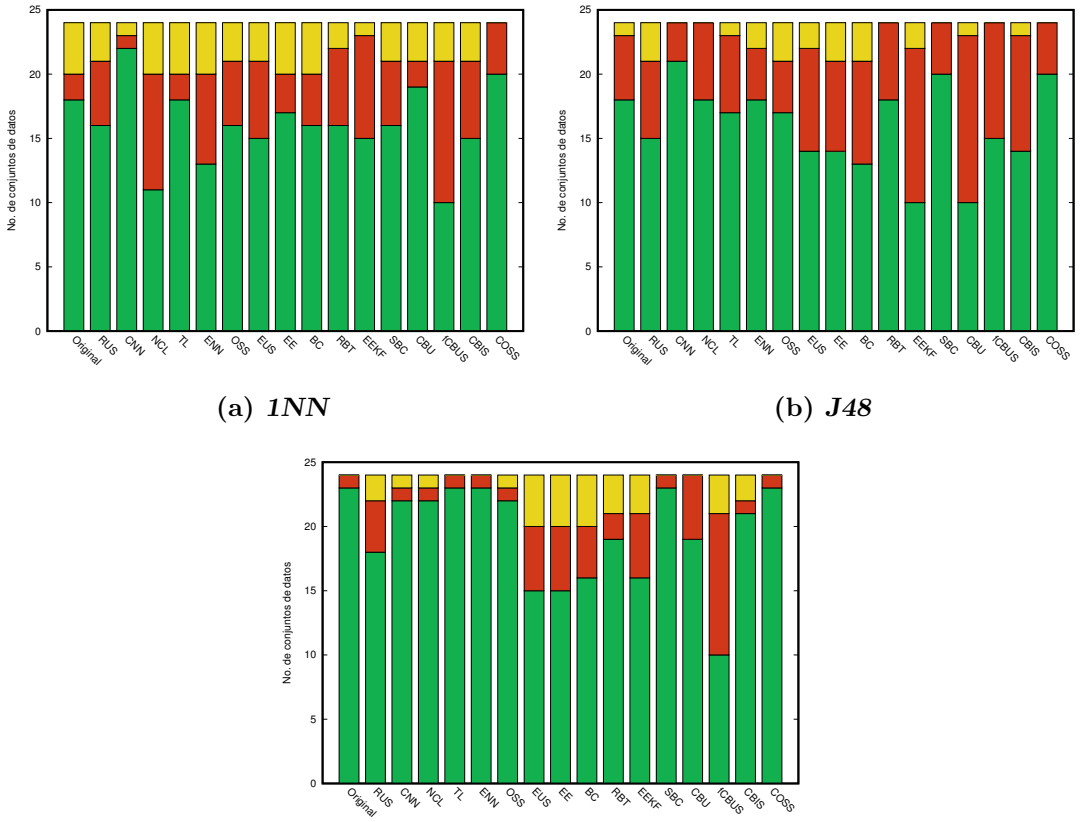
(a) ***Conjuntos de datos reales*** **(b)** ***Conjuntos de datos sintéticos***

Figura 7.4: Comparativa de DBMIST-US con el resto de métodos del promedio de la media geométrica obtenida.

Las Figuras 7.5 y 7.6 incluye los resultados de un análisis de victorias-empates-pérdidas que el método DBMIST-US obtuvo un rendimiento mejor, igual o peor en comparación con el resto de los métodos, para los tres clasificadores usados en el estudio experimental.

En la Figura 7.5 se distingue que independientemente del clasificador usado, en más de la mitad de conjuntos probados, DBMIST-US es mejor con respecto a los métodos basados en vecindario y ensembles. No obstante, para el clasificador J48, los métodos basados en *clustering*, en un 60 % DBMIST-US muestra peor desempeño, no así para el clasificador SVM, donde en métodos como CBU, DBMIS-US obtienen mejores resultados.

Con respecto a los conjuntos de datos sintéticos, es contundente que para el clasificador 1NN, la propuesta DBMIST-US es mejor que la mayoría de métodos. Solo para los métodos NCL, ENN, EEKF y SBC en un solo conjunto de datos se obtuvo peor rendimiento. Este último comportamiento se repite con el clasificador J48, en métodos como RUS, NCL, TL, OSS, EUS y CBIS. Mientras que frente a algunos métodos de ensembles, tales como EE, BC, RBT es estadísticamente significativamente mejor DBMIST-US, sin embargo frente a SBC, DBMIS-US genera en un 53.33 % mejor comportamiento.



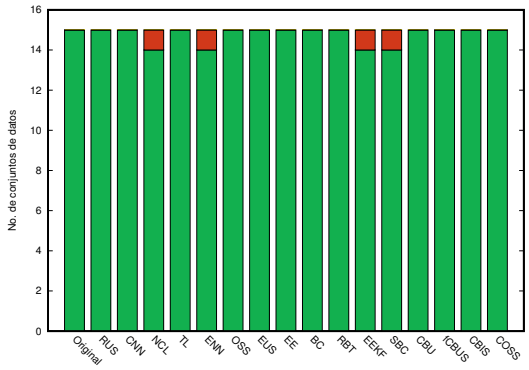
(a) ***1NN***

(b) ***J48***

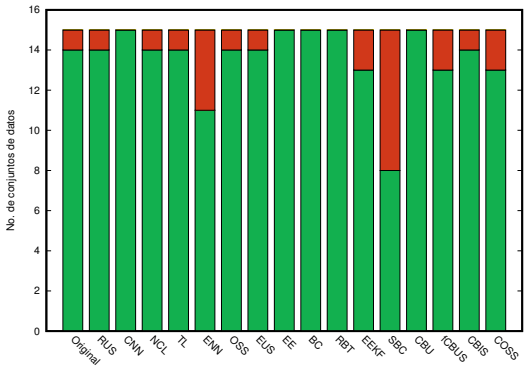
(c) ***SVM***

Figura 7.5: Cantidad de conjuntos de datos reales en los que DBMIST-US fue mejor (verde), igual (amarillo) o peor (rojo).

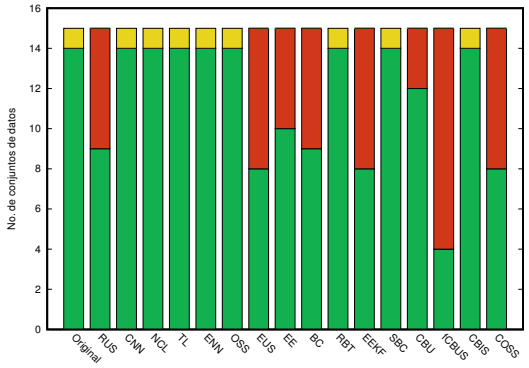
En particular, para el clasificador SVM, en datos sintéticos se aprecia con respecto a métodos basados en ensembles como, EUS, EE, BC y EEKF, en al menos un 53.33 % de los conjuntos de datos, DBMIST-US es mejor, mientras que en el resto de los conjuntos de datos presenta un inferior comportamiento. Un caso particular es se encuentra que frente a fCBUS, DBMIST-US presenta en un 73.3 % menor comportamiento.



(a) ***1NN***



(b) ***J48***



(c) ***SVM***

Figura 7.6: Cantidad de conjuntos de datos sintéticos en los que DBMIST-US fue mejor (verde), igual (amarillo) o peor (rojo).

Dada la comparativa de DBMIST-US en conjuntos de datos sintéticos obtuvo mejores resultados para el clasificador 1NN, pero bajo rendimiento con los clasificadores J48 y SVM.

Para validar si las mejoras son estadísticamente significativas, las Tablas 7.4, 7.5 y 7.6 resumen los resultados del test de Wilcoxon para los clasificadores usados.

Tabla 7.4: Test de Wilcoxon para resultados de DBMIST-US con el clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBMIST-US
Original	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○
	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○
RUS	●	-	●		●												●	○
	●	-	●	○		○			○	○			○	○		●		○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NCL	●		●	-	●		●						●	●		●	●	
	●	●	●	-	●	○	●	●	●	●	●	○	○	●	●	●		○
TL	●	○	●	○	-	○		○	○	○	○	○	○		○	○		○
	●		●	○	-	○		○	○	○	○	○	○	●		●		○
ENN	●		●		●	-	●						●	●	○		●	○
	●	●	●	●	●	-	●	●	●	●	●	●	●	●	●	●	●	○
OSS	●		●	○		○	-	○	○	○	○	○			○			○
	●		●	○		○	-			○		○	○	●		●		○
EUS	●		●		●		●	-					●	●			●	○
	●		●	○		○		-				○	○	●		●		○
EE	●		●		●				-						○		●	○
	●	●	●	○		○			-			○	○	●		●		○
BC	●		●		●		●			-							●	○
	●	●	●	○		○				-		○	○	●		●		○
RBT	●		●		●		●				-			●		●	●	○
	●		●	○		○					-	○	○	●		●		○
EEKF	●		●		●		●					-	●	●		●	●	
	●	●	●		●	○	●	●	●	●	●	-		●	●	●		○
SBC	●		●	○	●	○		○				○	-		○			○
	●	●	●	●	●	○	●	●	●	●	●		-	●	●	●		○
CBU			●	○				○			○	○		-	○			○
		○	●	○	○	○	○	○	○	○	○	○	○	-	○	○		○
fCBUS	●		●		●		●						●	●	-	●	●	
	●		●	○		○						○	○	●	-			○
CBIS	●		●		●										○	-		○
	●		●	○	○	○	○	○	○	○	○	○	○	●		-		○
COSS				○		○		○	○	○	○	○			○		-	○
			●			○											-	○
DBMIST-US	●	●	●		●	●	●	●	●	●	●		●	●		●	●	-
	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-
$\alpha = 0.9$	1	4	0	8	2	7	2	7	5	5	7	8	3	1	10	3	1	14
	1	4	0	12	4	16	4	5	6	7	5	13	13	1	3	3	1	17
$\alpha = 0.95$	1	3	0	7	2	6	2	7	4	5	6	7	3	1	8	3	0	14
	1	3	0	12	4	16	4	4	5	5	4	12	13	1	3	3	1	17

Conforme a los resultados mostrados en la Tabla 7.4, se observa que para conjuntos de datos sintéticos en ambos niveles de significancia, la propuesta DBMIST-US es significativamente mejor. No obstante, para conjuntos de datos reales, en ambos niveles de significancia, frente a métodos como NCL, EEKF y fCBUS, la propuesta es competitiva.

En resumen, se puede concluir que para el clasificador 1NN, la propuesta DBMIST-US es ideal para tratar el problema de desbalance de datos, dado que genera mejores resultados frente a otras técnicas, mientras que para el tratamiento de traslape de clases y/o ruido, DBMIST-US es competitivo frente a técnicas basadas en ensembles y *clustering* pero mejor en técnicas basadas en vecindario.

Tabla 7.5: Test de Wilcoxon para resultados de DBMIST-US con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBMIST-US
Original	-	○	●	○		○		○	○	○	○	○		○		○		○
	-	○		○	○	○	○	○	○	○	○	○	○	○	○		○	○
RUS	●	-	●	●	●	●	●						●	○			●	○
	●	-	●	●	●		●		○		○		○		○	●		○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○
		○	-	○		○		○	○	○	○	○	○	○	○		○	○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		○
	●	○	●	-	●	○	●	○	○	○	○	○	○	○	○	●		○
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		○
	●	○		○	-	○		○	○	○	○	○	○	○	○	●	○	○
ENN	●	○	●		●	-	●		○	○	○	○		○				○
	●		●	●	●	-	●						○			●		○
OSS		○		○		○	-	○	○	○	○	○		○		○		○
	●	○		○		○	-	○	○	○	○	○	○	○	○		○	○
EUS	●		●		●		●	-					●	○			●	○
	●		●	●	●		●	-	○		○		○			●		○
EE	●		●	●	●	●	●		-			○	●	○			●	○
	●	●	●	●	●		●	●	-	●			○	●		●	●	○
BC	●		●	●	●	●	●		-	-			●	○			●	
	●		●	●	●		●		○	-			○			●		○
RBT	●		●	●	●	●	●				-		●	○			●	○
	●	●	●	●	●		●				-		○	●		●		○
EEKF	●		●	●	●	●	●					-	●	○			●	
	●		●	●	●		●					-	○	●		●		○
SBC		○	●					○	○	○	○	○	-	○		○		○
	●	●	●	●	●	●	●	●	●	●	●	●	-	●	●	●	●	
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●	●
	●		●	●	●		●		○		○		○	-	○	●		○
fCBUS			●											○	-		●	
	●		●	●	●		●						○		-	●		○
CBIS	●		●		●		●						●	○		-	●	
		○		○	○	○		○	○	○	○	○	○	○	○	-	○	○
COSS		○						○	○	○	○	○		○	○	○	-	○
	●		●		●		●						○			●	-	○
DBMIST-US	●	●	●	●	●	●	●		●		●		●				●	-
	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	-
$\alpha = 0.9$	1	8	0	4	1	4	1	7	8	8	8	9	1	17	3	7	0	12
	0	6	0	5	2	6	1	6	11	6	9	7	16	6	8	0	5	16
$\alpha = 0.95$	1	7	0	4	1	4	0	6	8	8	8	8	1	15	2	6	0	11
	0	6	0	5	2	6	1	6	10	6	8	6	16	6	6	0	5	16

En cuanto al clasificador J48, con respecto a conjuntos de datos reales, para ambos niveles de α, DMIST-US no presenta diferencias significativas, respecto a métodos tales como EEKF, fCBUS y CBIS. En contraste, para conjuntos de datos sintéticos, con el único método que se presenta competitivo es con SBC, para el resto de métodos, DBMIST-US es estadísticamente significativamente mejor-

Por último, para al clasificador SVM el mejor *rank* promedio fue obtenido por el método fCBUS (ver Apéndice C, Tabla C.6) en conjuntos de datos sintéticos, en la Tabla 7.6 se observa que DMIST-US es competitivo con fCBUS para un nivel de significancia del $\alpha = 0.95$, lo que indica un comportamiento aceptable frente a este método, pero mejor en métodos basados en vecindario y basados en ensembles.

Tabla 7.6: Test de Wilcoxon para resultados de DBMIST-US con el clasificador SVM.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBMIST-US
Original	-	○		○		○		○	○	○	○	○		○	○	○	○	○
	-	○	○					○	○	○		○		○	○		○	○
RUS	•	-	•	•	•	•	•						•			•	•	○
	•	-	•	•	•	•	•				•		•	•	○	•		
CNN		○	-					○	○	○	○	○		○	○			○
		○	-	•	•	•	•	○	○	○	•	○	•	○	○	•	○	○
NCL	•	○		-	•			○	○	○	○	○		○	○			○
		○		-				○	○	○		○		○	○		○	○
TL		○		○	-	○		○	○	○	○	○		○	○	○	○	○
		○			-			○	○	○		○		○	○		○	○
ENN	•	○			•	-		○	○	○	○	○		○	○	○		○
		○				-		○	○	○		○		○	○		○	○
OSS		○					-	○	○	○	○	○		○	○	○	○	○
		○					-	○	○	○		○		○	○		○	○
EUS	•		•	•	•	•	•	-		•	•		•			•	•	○
	•		•	•	•	•	•	-	•		•		•	•	○	•		
EE	•		•	•	•	•	•		-		•		•			•	•	○
	•		•	•	•	•	•	○	-		•		•		○	•		
BC	•		•	•	•	•	•			-	•		•			•	•	○
	•		•	•	•	•	•			-	•		•		○	•		
RBT	•		•	•	•	•	•	○	○		-		•		○	•	•	○
		○						○	○	○	-	○		○	○		○	○
EEKF	•		•	•	•	•	•					-	•			•	•	○
	•		•	•	•	•	•				•	-	•	•	○	•		
SBC		○						○	○	○	○	○	-	○	○	○		○
		○						○	○	○		○	-	○	○		○	○
CBU	•		•	•	•	•	•						•	-		•	•	○
	•	○	•	•	•	•	•	○			•	○	•	-	○	•		○
fCBUS	•		•	•	•	•	•						•		-	•	•	
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	-	•	•	•
CBIS	•	○			•		•	○	○	○	○	○		○	○	-		○
		○						○	○	○		○		○	○	-	○	○
COSS	•	○						○	○	○	○	○		○	○		-	○
	•		•	•	•	•	•				•		•			•	-	
DBMIST-US	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	-
	•		•	•	•	•	•				•		•			•		-
$\alpha = 0.9$	0	9	0	2	0	2	0	11	10	10	9	9	0	9	10	5	3	16
	0	10	8	0	0	0	0	11	9	9	0	10	0	9	17	0	9	10
$\alpha = 0.95$	0	9	0	2	0	2	0	10	10	9	9	9	0	9	9	3	1	16
	0	10	0	0	0	0	0	11	9	9	0	10	0	9	15	0	9	9

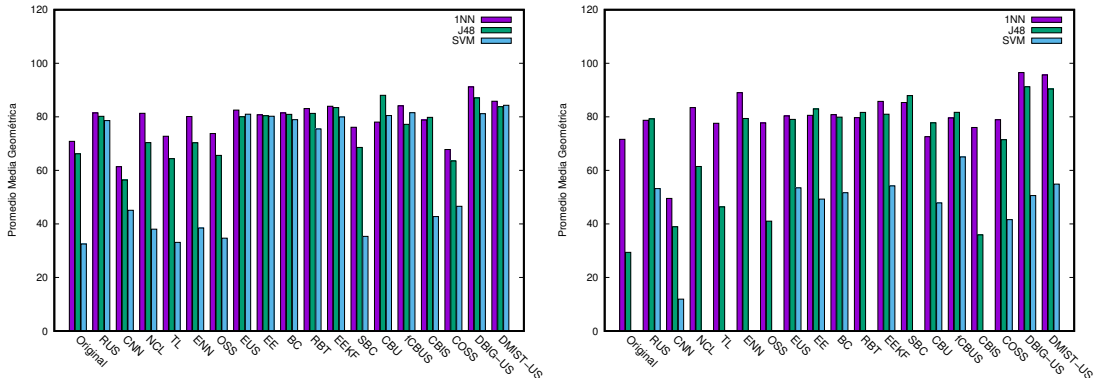
En resumen, por medio de los análisis presentados en esta sección, DBMIST-US supera significativamente al resto de los métodos en la mayoría de los conjuntos de datos para el clasificador 1NN, y mantiene un nivel competitivo para clasificadores como J48 y SVM. En consecuencia, es aconsejable este método como un método para el tratamiento de desbalance y solapamiento de clases y/o ruido.

7.3. Discusión de propuestas para el tratamiento de desbalance de clases, traslape de clases y/o ruido

En esta sección, se incluyen los resultados de dos propuestas para el de tratamiento de desbalance de clases, traslape de clases y/o ruido. Ambas propuestas constituyen un método de dos fases, en el cual primeramente se realiza la limpieza de la clase mayoritaria con el algoritmo modificado de DBSCAN y posteriormente se construye un subgrafo inducido o un árbol mínimo de expansión.

Los resultados reportados en términos de media geométrica mostraron la superioridad de ambas propuestas respecto a otros algoritmos de bajo-muestreo ampliamente

usados en el estado del arte. Esto puede comprobarse en las gráficas mostradas en la Figura 7.7.



(a) ***Conjuntos de datos reales*** **(b)** ***Conjuntos de datos sintéticos***

Figura 7.7: Comparativa general de tratamiento de desbalance de clases, traslape de clases y/o ruido con respecto al promedio de la media geométrica obtenida.

De acuerdo al promedio obtenido de la media geométrica de los métodos DBIG-US y DBMIST-US para el clasificador 1NN en ambos conjuntos de datos se obtienen promedios por arriba del 85 %. Para el clasificador J48 sucede algo similar, ya que DBIG-US obtiene en promedio 87.05 para conjuntos de datos reales y DBMIST-US un 83.78, mientras que para conjuntos de datos sintéticos, DBIG-US presenta un promedio de 91.23 y 90.40 para DBMIST-US (Ver Apéndice C, Tablas C.7 y C.8) .

Particularmente, para el clasificador SVM (Ver Apéndice C, Tabla C.9) en conjuntos de datos reales, las propuestas DBIG-US y DBMIST-US en promedio la media geométrica reportada está por arriba del 80 %, siendo de los más altos. Mientras que, para conjuntos de datos sintéticos, el mejor método es fCBUS, no obstante es seguido por las propuestas DBMIST-US (54.88) y DBIG-US (50.60).

Estos resultados sugieren que ambas propuestas obtienen rendimientos similares al mejor método hasta el momento para el escenario experimental (fCBUS). Para validar las mejoras, la prueba de Wilcoxon es ejecutada y resumida en las Tablas 7.7, 7.8 y 7.9.

De la Tabla 7.7 que resume los resultados de pruebas a pares de los métodos probados para el clasificador 1NN, se pueden extraer algunas conclusiones interesantes, la primera de ellas es que DBIG-US es estadísticamente significativamente mejor que DBMIST-US en conjuntos de datos reales y sintéticos. Esto sugiere que para clasificadores basados en proximidad, la selección de instancias más alejadas unas de las otras generan una silueta representativa de clase mayoritaria.

Por otro lado, es importante señalar que la propuesta DBMIST-US no presenta diferencias significativas frente a métodos como NCL, EEKF y fCBUS en conjuntos de datos reales. En tanto que, para conjuntos de datos sintéticos en ambos niveles de significancia, DBMIST-US es significativamente mejor que métodos basados en vecindarios,

basados en ensembles y *clustering*.

Tabla 7.7: Test de Wilcoxon para el tratamiento de desbalance de clases, traslape de clases y/o ruido con el clasificador 1NN.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US	DBMIST-US
Original	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○	○
	-	○	●	○	○	○	○	○	○	○	○	○	○		○	○		○	○
RUS	●	-	●		●												●	○	○
	●	-	●	○		○			○	○		○	○	●		●		○	○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
NCL	●		●	-	●		●						●	●		●	●	○	
	●	●	●	-	●	○	●	●	●	●	●	○	○	●	●	●		○	○
TL	●	○	●	○	-	○		○	○	○	○	○	○		○	○		○	○
	●		●	○	-	○		○	○	○	○	○	○	●		●		○	○
ENN	●		●		●	-	●						●	●	○		●	○	○
	●	●	●	●	●	-	●	●	●	●	●	●	●	●	●	●	●	○	○
OSS	●		●	○		○	-	○	○	○	○	○			○			○	○
	●		●	○		○	-			○		○	○	●		●		○	○
EUS	●		●		●		●	-					●	●			●	○	○
	●		●	○		○		-				○	○	●		●		○	○
EE	●		●		●				-						○		●	○	○
	●	●	●	○		○			-			○	○	●		●		○	○
BC	●		●		●		●			-							●	○	○
	●	●	●	○		○				-		○	○	●		●		○	○
RBT	●		●		●		●				-			●		●	●	○	○
	●		●	○		○					-	○	○	●		●		○	○
EEKF	●		●		●		●					-	●	●		●	●	○	
	●	●	●		●	○	●	●	●	●	●	-		●	●	●		○	○
SBC	●		●	○	●	○		○				○	-		○			○	○
	●	●	●	●	●	○	●	●	●	●	●		-	●	●	●		○	○
CBU			●	○				○			○	○		-	○			○	○
		○	●	○	○	○	○	○	○	○	○	○	○	-	○	○		○	○
fCBUS	●		●		●		●						●	●	-	●	●		
	●		●	○		○						○	○	●	-			○	○
CBIS	●		●		●										○	-		○	○
	●		●	○	○	○	○	○	○	○	○	○	○	●		-		○	○
COSS				○		○		○	○	○	○	○			○		-	○	○
			●			○											-	○	○
DBIG-US	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	-	●
	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	●
DBMIST-US	●	●	●		●	●	●	●	●	●	●		●	●		●	●	○	-
	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		-
$\alpha = 0.9$	1	4	0	8	2	7	2	7	5	5	7	8	3	1	10	3	1	17	14
	1	4	0	12	4	16	4	5	6	7	5	13	13	1	3	3	1	18	17
$\alpha = 0.95$	1	3	0	7	2	6	2	7	4	5	6	7	3	1	8	3	0	17	14
	1	3	0	12	4	16	4	4	5	5	4	12	13	1	3	3	1	17	17

Al considerar el clasificador J48 (Tabla 7.8), para ambos niveles de significancia, los resultados muestran que no existe diferencia significativa entre ambas propuestas, por lo que son competitivas. Para niveles de $\alpha = 0.95$ ambas propuestas no presentan rivalidad con métodos tales como BC, EEKF, CBU y fCBUS en conjuntos de datos reales, mientras que para conjuntos de datos sintéticos, DBIG-US es significativamente mejor que los métodos basados en vecindario, basados en ensembles y basados en *clustering*.

Tabla 7.8: Test de Wilcoxon para el tratamiento de desbalance de clases, traslape de clases y/o ruido con el clasificador J48.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US	DBMIST-US
Original	-	○	●	○		○		○	○	○	○	○		○		○		○	○
	-	○		○	○	○	○	○	○	○	○	○	○	○	○		○	○	○
RUS	●	-	●	●	●	●	●						●	○			●	○	○
	●	-	●	●	●		●		○		○		○		○	●		○	○
CNN	○	○	-	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○
		○	-	○		○		○	○	○	○	○	○	○	○		○	○	○
NCL	●		●	-	●		●	○	○	○	○	○		○		○		○	○
	●	○	●	-	●	○	●	○	○	○	○	○	○	○	○	●		○	○
TL		○	●	○	-	○		○	○	○	○	○		○	○	○		○	○
	●	○		○	-	○		○	○	○	○	○	○	○	○	●	○	○	○
ENN	●	○	●		●	-	●		○	○	○	○		○				○	○
	●		●	●	●	-	●						○			●		○	○
OSS		○		○		○	-	○	○	○	○	○		○		○		○	○
	●	○		○		○	-	○	○	○	○	○	○	○	○		○	○	○
EUS	●		●		●		●	-					●	○			●	○	○
	●		●	●	●		●	-	○		○		○			●		○	○
EE	●		●	●	●	●	●		-			○	●	○			●	○	○
	●	●	●	●	●		●	●	-	●			○	●		●	●	○	○
BC	●		●	●	●	●	●			-			●	○			●	○	
	●		●	●	●		●		○	-			○			●		○	○
RBT	●		●	●	●	●	●				-		●	○			●	○	○
	●	●	●	●	●		●				-		○	●		●		○	○
EEKF	●		●	●	●	●	●					-	●	○			●	○	
	●		●	●	●		●					-	○	●		●		○	○
SBC		○	●					○	○	○	○	○	-	○		○		○	○
	●	●	●	●	●	●	●	●	●	●	●	●	-	●	●	●	●	○	
CBU	●	●	●	●	●	●	●	●	●	●	●		●	-	●	●	●		●
	●		●	●	●		●		○		○		○	-	○	●		○	○
fCBUS			●											○	-		●	○	
	●		●	●	●		●						○		-	●		○	○
CBIS	●		●		●		●						●	○		-	●	○	
		○		○	○	○		○	○	○	○	○	○	○	○	-	○	○	○
COSS		○						○	○	○	○	○		○	○	○	-	○	○
	●		●		●		●						○			●	-	○	○
DBIG-US	●	●	●	●	●	●	●	●	●		●		●			●	●	-	
	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	-	
DBMIST-US	●	●	●	●	●	●	●		●		●		●				●		-
	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●		-
$\alpha = 0.9$	1	8	0	4	1	4	1	7	8	8	8	9	1	17	3	7	0	16	12
	0	6	0	5	2	6	1	6	11	6	9	7	16	6	8	0	5	17	16
$\alpha = 0.95$	1	7	0	4	1	4	0	6	8	8	8	8	1	15	2	6	0	13	11
	0	6	0	5	2	6	1	6	10	6	8	6	16	6	6	0	5	16	16

Por último, para el clasificador SVM (Tabla 7.9), es importante resaltar que para niveles de $\alpha = 0.95$ en conjuntos de datos reales y sintéticos DBMIST-US presenta competitividad con fCBUS, no así la propuesta DBIG-US, la cual únicamente presenta competitividad con conjuntos de datos reales. Para ambos niveles de significancia, ambas propuestas son competitivas frente a ellas mismas, en conjuntos de datos reales y sintéticos. Por otro lado, la propuesta DBIG-US presenta competitividad frente a métodos como EUS, EE, BC,EEKF y CBU en ambos niveles de α para conjuntos de datos sintéticos y reales.

Estos resultados sugieren que la limpieza de conjuntos de datos con problemas de ruido y/o traslape de clases es adecuadamente tratado con DBSCAN, mientras que el desbalance de clases es abordado de forma favorable con los grafos utilizados.

Tabla 7.9: Test de Wilcoxon para el tratamiento de desbalance de clases, traslape de clases y/o ruido con el clasificador SVM.

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US	DBMIST-US
Original	- -	○ ○	○	○		○		○ ○	○ ○	○ ○	○	○ ○		○ ○	○ ○	○	○ ○	○ ○	○ ○
RUS	• •	- -	• •	• •	• •	• •	• •				•		• •	• •	○	• •	•		○
CNN		○ ○	- -	○ •	○ •	○ •	○ •	○ ○	○ ○	○ ○	○ •	○ ○	• •	○ ○	○ ○	• •	○	○ ○	○ ○
NCL	•	○ ○		- -	• •			○ ○	○ ○	○ ○	○	○ ○		○ ○	○ ○			○ ○	○ ○
TL		○ ○		○	- -	○		○ ○	○ ○	○ ○	○	○ ○		○ ○	○ ○	○	○ ○	○ ○	○ ○
ENN	•	○ ○			• •	- -		○ ○	○ ○	○ ○	○	○ ○		○ ○	○ ○	○	○	○ ○	○ ○
OSS		○ ○					- -	○ ○	○ ○	○ ○	○	○ ○		○ ○	○ ○	○	○ ○	○ ○	○ ○
EUS	• •		• •	• •	• •	• •	• •	- -	• -	•	•		• •	• •	○	• •	•		○
EE	• •		• •	• •	• •	• •	• •	○	- -		•		• •		○	• •	•		○
BC	• •		• •	• •	• •	• •	• •			- -	• •		• •		○	• •	•		○
RBT	•	○	•	•	•	•	•	○ ○	○ ○	○	- -	○	•	○	○ ○	•	• ○	○	○ ○
EEKF	• •		• •	• •	• •	• •	• •				•	- -	• •	• •	○	• •	•		○
SBC		○ ○						○ ○	○ ○	○ ○	○	○ ○	- -	○ ○	○ ○	○	○	○ ○	○ ○
CBU	• •	○	• •	• •	• •	• •	• •	○			•	○	• •	- -	○	• •	•		○ ○
fCBUS	• •	•	• •	• •	• •	• •	• •	•	•	•	•	•	• •	• •	- -	• •	• •	•	•
CBIS	•	○ ○			•		•	○ ○	○ ○	○ ○	○	○ ○		○ ○	○ ○	- -	○	○ ○	○ ○
COSS	• •	○						○	○	○	○ •	○		○	○		- -	○	○
DBIG-US	• •		• •	• •	• •	• •	• •				•		• •		○	• •	•	- -	
DBMIST-US	• •	•	• •	• •	• •	• •	• •	•	•	•	• •	•	• •	•		• •	•		- -
$\alpha = 0.9$	0 0	9 10	0 8	2 0	0 0	2 0	0 0	11 11	10 9	10 9	9 0	9 10	0 0	9 9	10 18	5 0	3 9	9 9	16 10
$\alpha = 0.95$	0 0	9 10	0 0	2 0	0 0	2 0	0 0	10 11	10 9	9 9	9 0	9 10	0 0	9 9	9 16	3 0	1 9	9 9	16 9

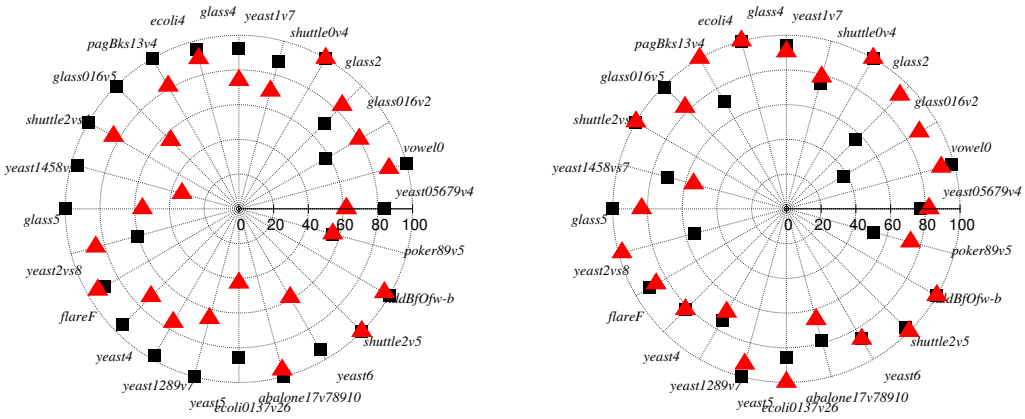
7.4. Análisis de rendimiento por clase

Derivado de la discusión de la Sección 7.3, para determinar el comportamiento de las tasas de verdaderos negativos y verdaderos positivos de las propuestas generadas a partir de la combinación de algoritmos de *clustering* y algoritmos basados en grafos, en las Figuras 7.8 y 7.9 se ilustra la precisión por clase para conjuntos de datos reales y sintéticos respectivamente, generados por las propuestas DBIG-US Y DBMIST-US, únicamente para el clasificador SVM.

El resultado de especificidad se ilustra en triángulos rojos y el resultado de Sensibilidad en cuadrados negros, el resto de métodos se encuentran en el Apéndice D.

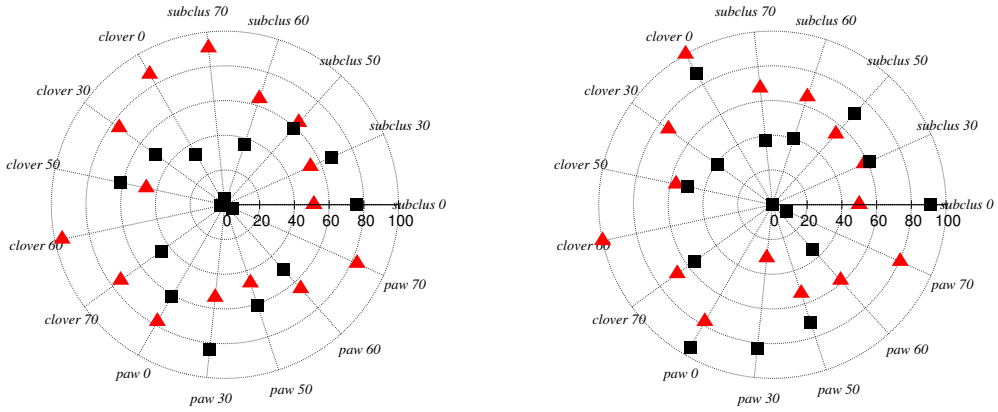
Como se observa en las gráficas de la Figura 7.8 para conjuntos de datos reales, ambas propuestas obtienen resultados para Sensibilidad y Especificidad equiparables, lo que indica un adecuado tratamiento de las complejidades sin perdida de rendimiento en el clasificador.

Para conjuntos de datos sintéticos el comportamiento es similar a los conjuntos de datos reales, es decir, para métodos como CNN, NCL, CNN, NCL, TL, ENN, OSS, RBT, SBC y CBIS se mantienen valores de Sensibilidad cercanos a 0, lo que implica que estos métodos no dan una adecuada solución al tratamiento de desbalance, traslape de clases y/o ruido.



(a) ***DBIG-US*** (b) ***DBMIST-US***

Figura 7.8: Precisión por clase para conjuntos de datos reales con métodos DBIG-US y DBMIST-US.



(a) ***DBIG-US*** (b) ***DBMIST-US***

Figura 7.9: Precisión por clase para conjuntos de datos sintéticos con métodos DBIG-US y DBMIST-US.

Un caso particular es el método COSS, donde el rendimiento de la clase minoritaria mejoró significativamente, no obstante, se generó un decrecimiento severo en la clase mayoritaria, esto se debe a que COSS elimina demasiadas instancias de clase negativa, intercambiando el papel de ambas clases, es decir, la clase mayoritaria se vuelve minoritaria.

Capítulo 8

Desbalance de clases en Big Data

El problema del desbalance de clases ha sido una de las complejidades de los datos ampliamente estudiada. Su importancia recae en el sesgo que tienen ciertos clasificadores en favor de la clase mayoritaria, deteriorando su rendimiento general. Por otro lado, la creciente generación de datos agrava el problema relacionado con la presencia de diversas complejidades inherentes en los datos. Dentro del contexto de *Big Data*, el desafío implica la necesidad de adaptar o crear nuevas técnicas para solventar las limitaciones de escalabilidad.

De manera específica, para técnicas a nivel de datos, los métodos de re-muestreo consisten en ajustar el tamaño del conjunto de datos para equilibrar la distribución de la clase, ya sea disminuyendo el número de instancias de clase mayoritaria (bajo-muestreo) o el aumento del número de instancias de clase minoritaria (sobre-muestreo). En problemas de *Big Data*, el bajo-muestro podría ser la mejor estrategia puesto que permitiría una reducción del volumen, siempre y cuando no se deteriorase la precisión del clasificador.

8.1. Estrategias para el tratamiento del desbalance en Big Data

Al igual que para escenarios tradicionales de Aprendizaje Automático, en la literatura relativa a Big Data existen dos estrategias principales para el tratamiento del desbalance en la distribución entre clases [77]: sobre-muestreo y bajo-muestreo.

8.1.1. Sobre-muestreo

Zhai et al. [78] proponen un algoritmo de sobre-muestreo basado en cuatro fases para un problema de dos clases. En la primera fase, se calcula el centro de las instancias de clase positiva y, posteriormente, se buscan instancias vecinas del centro de la clase positiva. Para cada instancia de clase positiva, se identifican sus k vecinos más cercanos de clase negativa con MapReduce, donde la función Map identifica la instancia de menor distancia por cada partición con respecto al centro, mientras que la función Reduce

compara los resultados obtenidos por Map de las diferentes particiones y obtiene la instancia con distancia mínima global. El proceso de sobre-muestreo se repite de forma iterativa hasta p veces (parámetro libre). En la segunda fase, se crean más instancias sintéticas de la clase positiva hasta obtener el mismo tamaño de instancias de clase negativa. Para cada ronda de muestreo, se generan instancias de clase positiva para obtener l subconjuntos de datos balanceados. Posteriormente, utilizando los subconjuntos balanceados, se aplica *Extreme Learning Machine* (ELM) [1] para fines de clasificación. En la última fase, se integran las decisiones individuales de los clasificadores con una estrategia de votación simple.

Un estudio posterior de Zhai et al. [78] trata el problema de desbalance en tres fases [33]: La primera incluye MapReduce para realizar un sobre-muestreo aleatorio a la clase positiva, basado en el enemigo más cercano (vecino más cercano encontrado en una clase diferente). Para ello, la función Map realiza el cálculo de vecino más cercano y el sobre-muestreo, mientras que la fase de Reduce conjunta los resultados obtenidos del sobre-muestreo en la función Map. La segunda fase está dada por el entrenamiento de redes neuronales de alimentación oculta de capa única, utilizando los conjuntos de datos balanceados. Por último, en la tercera fase, los resultados obtenidos de las redes neuronales son ensamblados utilizando la integral difusa (fuzzy integral).

Utilizando MapReduce, Del Río et al. [77] implementan y analizan varios algoritmos de sobre-muestreo y bajo-muestreo (ROS, RUS y SMOTE), utilizando un cluster de computadoras. En un principio, todos los conjuntos de datos son particionados en bloques de datos independientes, los cuales son replicados a los largo de diferentes nodos del cluster. Para ROS, la función Map balancea aleatoriamente con instancias replicadas de la clase minoritaria, mientras que la función Reduce conjunta todos los resultados generados por cada Map, tomando aleatoriamente instancias para balancear el conjunto de datos. Para RUS, la función Map agrupa todas las instancias por clase, mientras que Reduce recolecta y balancea por eliminación de instancias aleatorias de la clase mayoritaria. Por último, utilizando el algoritmo SMOTE, la función Map sobre-muestrea la clase minoritaria, mientras que la función Reduce recolecta los resultados generados por cada partición y aleatoriamente genera un conjunto de datos balanceado.

Gutiérrez et al. [69] implementaron SMOTE utilizando memorias gráficas (GPU). Dado que SMOTE requiere de la regla de los vecinos más próximos, al ser implementada bajo el esquema de GPU, se divide en dos kernels; el primer kernel construye la matriz de distancias entre el conjunto de entrenamiento y el de prueba, y el segundo realiza la búsqueda de los k vecinos más cercanos.

Otro estudio es el de Hu et al. [34], en el que presentan un algoritmo nombrado NRSBoundary-SMOTE (Neighborhood Rough Set Boundary SMOTE) basado en tres etapas. En la primera etapa se busca la clase minoritaria, generando instancias con

[1] El algoritmo ELM fue propuesto para entrenar redes neuronales de alimentación oculta de capa única (Single-Hidden Layer Feedforward Neural Network, SLFN), los pesos y sesgos de entrada no necesitan ser ajustados, y es posible determinar analíticamente los pesos de salida al encontrar la solución de mínimos cuadrados. La red neuronal se obtiene después de muy pocos pasos con un costo computacional muy bajo [79].

SMOTE y seleccionando las instancias sintéticas que no afecten el espacio de decisión de la clase mayoritaria, todo esto desarrollado en MapReduce. Se emplearon dos tareas MapReduce, donde el primer método MapReduce divide el conjunto de datos, la función Map obtiene los vecinos más cercanos entre las instancias de acuerdo con la clase, el par clave-valor se almacenará, de acuerdo a si una instancia x_i es de la clase positiva se genera el par $(positivo, x_i)$, de lo contrario $(limite, x_i)$. La función Reduce obtiene los resultados de las funciones Map y los agrupa dependiendo del par generado. En la segunda etapa, se realiza el sobre-muestreo de clase minoritaria, donde la función Map obtiene las instancias de la clase minoritaria y obtiene los k vecinos. Posteriormente, selecciona una instancia al azar de los k vecinos, la cual será usada para crear una instancia sintética en la fase de Reduce.

Wang et al. [80] presentan un algoritmo denominado PMIB_SVM, el cual combina el algoritmo IB_SVM con el meta-aprendizaje, es decir, la obtención del aprendizaje por medio de diferentes clasificadores de manera paralela implementado con MapReduce. Para esta implementación, el número de clasificadores es igual al número de *mappers*, es decir, al número de particiones de la función Map. La función Map se encarga de dividir el conjunto de datos y entrenar el respectivo clasificador con cada uno. Al final todos los modelos entrenados se recopilan y almacenan en un archivo de salida.

Para tratar el desbalance en problemas de más de dos clases, Bhagat et al. [81] proponen una metodología para multi-clase utilizando MapReduce, la cual consta de dos fases. La primera fase consiste en Binarization (OVA) para descomponer el conjunto de datos original en subconjuntos de dos clases. En la segunda fase, la función Map balancea la distribución de clases en cada partición utilizando SMOTE, mientras que Reduce se encarga de la selección aleatoria de instancias del conjunto balanceado de datos. Como clasificador, se implementa un árbol de decisión.

Subhash et. al [82] presentan técnicas de sobre-muestreo, todas estas desarrolladas con técnicas de clustering. La primera propuesta se denonima CME (Clustering Minority Examples), la cual sólo involucra las instancias de clases minoritarias para generar instancias sintéticas a través del algoritmo *K-Means*. Otra técnica es UCPM (Update Class Purity Maximazation), la cual se centra en el sobre-muestreo bajo la observación de la clase mayoritaria, encontrando todos los grupos de instancias de clase mayoritaria pura, es decir, aquellas X instancias vecinas del centro. El resto instancias de clase mayoritaria se consideran impuras y necesarias para realizar el sobre-muestreo, debido a que la cardinalidad del conjunto de instancias impuras es considerado como cardinalidad del conjunto de clase minoritaria.

Dado que para MapReduce, existen dos enfoques de programación, Fernádez et al. [83] adaptó desde un enfoque local los algoritmos RUS y ROS nombrados como RUS-BigData y ROS-BigData, básicamente por medio de cada función Map se equilibra la distribución de las clases a través de métodos de bajo-muestreo o sobre-muestreo, debido a que cada Map tiene soluciones independientes por particiones, para crear un conjunto de datos balanceado final, se utiliza una única función Reduce al recopilar los resultados para cada mapper.

Sin embargo, alrededor del método SMOTE existen algunas propuestas, como SMOTE-MR [84], que es una versión local de SMOTE, en la que para calcular los k vecinos de una instancia de la clase minoritaria, SMOTE-MR usa la misma partición a la que pertenece una instancia. Por el contrario, SMOTE-DB [85] se desarrolla como una técnica global, donde el cálculo de k vecinos más cercanos se basó en el método *kNN-IS* [86], dónde mediante múltiples funciones de Reduce se determina cuáles son los k vecinos finales más cercanos de cada mapper. Una función Map calcula para cada fragmento las distancias y las clases correspondientes de los k vecinos más cercanos para cada instancia.

8.1.2. Bajo-muestreo

Para bajo muestro, Triguero et al. [87] presentan un algoritmo evolutivo denominado EUS (Evolutive Under Sampling), cuyo objetivo principal es reducir el conjunto de entrenamiento por medio de la obtención de k vecinos. Si bien la reducción de tamaño puede lograrse mediante RUS (Random Under Sampling), éste puede descartar instancias importantes de la clase mayoritaria, mientras que EUS guía el proceso mediante el equilibrio de preservación de la precisión en ambas clases. Se codifica de manera binaria las instancias, implementando MapReduce; la función Map se encarga de dividir el conjunto de datos en múltiples fragmentos para ser procesados con el algoritmos EUS en los diferentes modos. Como trabajo a futuro se considera el uso de enfoques híbridos de sobre-muestro y bajo-muestreo, así como el incremento del grado del desbalance.

Por otra parte, Jedrzejowicz et al. [88] presentan una versión paralelizada del algoritmo SplitBal, el cual se encarga de dividir el conjunto de datos en bloques que contengan las instancias de la clase minoritaria y parte de la clase mayoritaria de igual tamaño. La implementación se realizó con el algoritmo MapReduce, donde la función Map se encarga de dividir el conjunto de entrenamiento en bloques, además de realizar la clasificación en cada bloque, mientras que la función Reduce se encarga de fusionar los resultados de los clasificadores para obtener una sola decisión.

8.2. Propuesta: Tratamiento del desbalance de clases en Big Data basado en grafos

La falta de métodos de bajo-muestreo para enfrentar el problema de desbalance de clases en contexto de *Big Data* deja una amplia brecha para desarrollar propuestas. Al respecto, la reducción del tamaño del conjunto de datos mediante la selección inteligente de instancias permite tener un mejor rendimiento en modelos de aprendizaje con la característica de usar un número reducido de datos, que de acuerdo al estudio presentado por Maillo et al. [89] no es indispensable contar con un número elevado de instancias para generar resultados de clasificación elevados.

Por otro lado, a condición de que los métodos presentados en el Capítulo 3, se encuentran basados en grafos y obtuvieron resultados favorables y competitivos frente

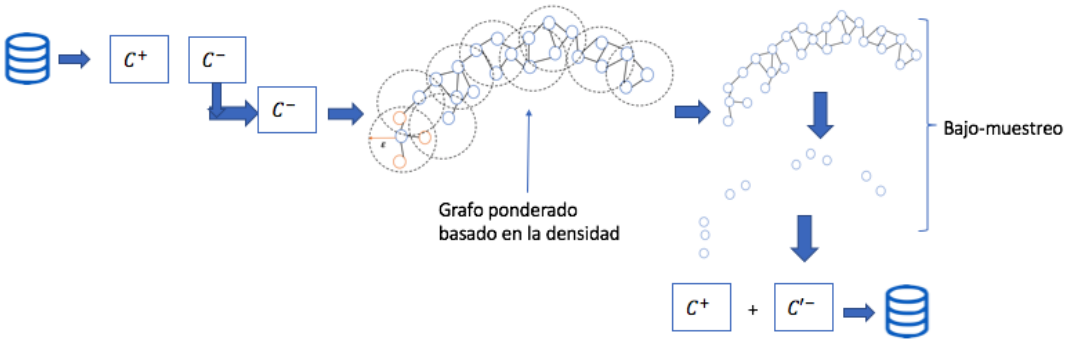


Figura 8.1: Método de bajo-muestreo basado en grafos.

a técnicas de bajo-muestreo, se deduce que la capacidad de los grafos para extraer conocimiento mediante representación de vértices y aristas, son capaces de enfrentar el problema de desbalance de clases y reducir el volumen de los datos. En consecuencia, como una primera aproximación se propone una estrategia de bajo-muestreo basada en grafos por medio de la construcción de un grafo basado en la densidad de los datos para grandes volúmenes de datos.

Dado un conjunto de datos (DS) de dos clases en el ámbito de *Big Data*, se representa a la clase mayoritaria como C^- y a la clase minoritaria como C^+. La propuesta aquí presentada únicamente trabaja con la clase mayoritaria de tal modo que el tamaño de esta reduzca hasta ser similar a la de la clase minoritaria, de este modo se asegura un grado de desbalance igual a uno.

La estrategia basada en grafos se divide en dos pasos importantes:

- Construcción del grafo. En primer lugar, se construye un grafo ponderado basado en la densidad del conjunto de datos (la densidad de un conjunto de datos hace referencia a la distribución de los datos). El grafo que se construye es un grafo disperso (*sparse graph*, definido como un grafo cuyo número de aristas está cerca del número mínimo de aristas) ya que únicamente se consideraran conectadas aquellas instancias vecinas unas de las otras por medio de un radio de vecindad.

- Bajo-muestreo. En este paso se busca equilibrar el conjunto de datos por medio de la reducción de clase mayoritaria, tomando las primeras z instancias de la matriz de adyacencia ($z = |C^+|$), cuyas entradas están en orden descendente o ascendente.

De modo gráfico, la Figura 8.1 presenta el flujo de tareas de la propuesta de bajo-muestreo basada en grafos.

8.2.1. Grafo ponderado basado en la densidad de los datos

Para construir un grafo de acuerdo a la propuesta, es necesario considerar que dada una instancia p_v de clase mayoritaria C^-, que es una tupla $(f_{v,1}, f_{v,2}, \ldots, f_{v,d})$, donde

$f_{v,i} \in \mathbb{R}$ y $1 \leq i \leq d$ es el valor de la i-ésima característica de p_v, un grafo ponderado G_w se construye se la siguiente manera:

- $V(G_w) = \{i \in V(G_w) \mid p_i \in C^-\}$, que es un vértice por cada instancia.
- $E(G_w) = \{\{v, u\} \mid v, u \in V(G_w),\ N_{eps}(p_v) = \{q_u \in C^- | dist(p_v, p_u) \leq eps\}\}$, donde $dist$ es la distancia Euclídea entre p_v y p_u calculada por:

$$dist(p_v, p_u) = \sqrt{(f_{v,1} - f_{u,1})^2 + (f_{v,2} - f_{u,2})^2 + \cdots + (f_{v,d} - f_{u,d})^2} \tag{8.1}$$

- $w(e) = dist(p_v, p_u)$.

El valor *eps* es calculado en base a las ecuaciones propuestas por Smiti y Elouedi [29], con la diferencia de que en nuestra propuesta únicamente se toma el 20 % del valor computado (Ecuación 8.2).

$$eps = \left(\sqrt{\frac{\sum_{v=1}^{|C^-|} dist(m, p_v)}{|C^-|}} \right) * 0.20 \tag{8.2}$$

donde m es el vector medio de C^- (Ecuación 8.3)

$$m = \left(\frac{\sum_{v=1}^{|C^-|} f_{v,1}}{|C^-|}, \frac{\sum_{v=1}^{|C^-|} f_{v,2}}{|C^-|}, \ldots, \frac{\sum_{v=1}^{|C^-|} f_{v,d}}{|C^-|} \right) \tag{8.3}$$

Una vez construido el grafo ponderado G_w, los elementos de su correspondiente matriz de adyacencia $M_G = (w_{vu})$ pueden ser ordenados de manera ascendente o descendente. Para reducir la cardinalidad del conjunto C^- al tamaño de C^+, se toman las primeras z-instancias de la matriz de adyacencia ordenada.

Cuando los elementos de la matriz de adyacencia se ordenan de manera descendente, la propuesta se denominada *GDshape* la cual toma todas aquellas instancias que están lo suficientemente cerca de la frontera de decisión. Por otro lado, cuando los elementos de la matriz de adyacencia se clasifican en orden ascendente, la propuesta es denominada *GDcore* la cual descarta las instancias que están lo suficientemente cerca del borde de decisión.

8.3. Escenario experimental

En términos de grandes volúmenes de datos, para determinar si las propuestas de bajo-muestreo re-direccionan de manera adecuada el tratamiento de desbalance de clases, se realizaron pruebas experimentales con un árbol de decisión, de tal forma que se pueda estimar el beneficio de pre-procesar los conjuntos de datos en contexto de *Big Data*. Los resultados obtenidos se examinan en términos de media geométrica.

Tabla 8.1: Conjuntos de datos ordenados de manera ascendente según su IR.

	Conjunto	Distribution	#Instancias	#Caracterísiticas	IR
1	poker1	346478 - 473529	820007	10	1.37
2	SEA	308518 - 491482	800000	3	1.59
3	Agrawal	262461 - 537538	799999	9	2.05
4	MiniBooNE	29178 - 74870	104048	49	2.57
5	Susy	542434 - 2169739	2712173	18	4.00
6	Click	266159 - 1331769	1597928	11	5.00
7	poker0	39062 - 410960	450022	10	10.52
8	HEPMASS	262435 - 4200169	4462604	28	16.00
9	HIGGS	291417 - 4663335	4954752	28	16.00
10	Covtype	16256 - 448421	464677	54	27.58
11	Credit	403 - 227440	227843	30	564.37
12	RLCP	16901 - 4582252	4599153	4	271.12

8.3.1. Conjuntos de datos utilizados

El estudio experimental se realizó sobre 12 conjuntos de datos de dos clases con problemas de desbalance, todos estos tomados del repositorio *UCI Machine Learning Repository* [90], cuyo grado de desbalance (IR) varía de 1.37 (siendo el más bajo) hasta 271.12 (siendo el más severo). La Tabla 8.1 describe las principales características de los conjuntos de datos, como lo son la distribución de clases, el número de instancias total, el número de características, por último y el grado de desbalance.

Para este estudio experimental, los conjuntos de datos se dividieron usando el esquema *5-fold cross validation*, en el que cada división el 80 % de las instancias se utilizaron para entrenamiento y el 20 % restante para pruebas.

8.3.2. Infraestructura tecnológica

La infraestructura usada en estas pruebas fue la proporcionada por la Universidad de Granada (`http://decsai.ugr.es/gte/`), con un cluster Hadoop, el cual consiste de 14 nodos interconectados vía Gigabit Ethernet. Cada nodo tienen un procesador Intel Core i7-4930K a 3.40GHz de 6 núcleos y 64 GB de RAM. En términos de software, se uso Apache Hadoop 2.9.1 y Apache Spark 2.2.0.

8.4. Resultados experimentales

Con el propósito de analizar el desempeño de clasificación del árbol de decisión de la librería MLlib de Spark, la Tabla 8.2 resume los porcentajes obtenidos de la media geométrica de métodos de preprocesado comúnmente usados en el dominio de *Big Data* (ROS, RUS, and SMOTE-MR) y de las propuestas basadas en grafos. Se incluyen los resultados de clasificación de los conjuntos de datos sin preprocesar, como referencia de evaluación. Se resalta el negrita los mejores resultados por cada conjunto de datos y técnica de remuestreo. Finalmente, la última fila corresponde al rango promedio

Tabla 8.2: Rendimiento basado en la media geométrica obtenida por el árbol de decisión para conjuntos de Big Data.

Conjunto	Original	ROS	RUS	SMOTE-MR	GDshape	GDcore
poker1	33.7	**53.3**	52.9	42.5	51.8	52.7
SEA	82.0	82.9	82.9	82.9	**83.0**	**83.0**
Agrawal	94.4	**95.1**	95.0	94.4	94.8	94.8
MiniBooNE	85.2	87.9	**88.0**	87.9	84.0	84.0
Susy	68.8	**76.8**	76.7	76.3	76.7	76.7
Click	16.2	62.1	**62.2**	56.2	61.9	61.9
poker0	17.0	58.1	56.2	59.7	56.2	**60.0**
HEPMASS	71.9	**83.3**	**83.3**	66.8	83.0	83.0
HIGGS	11.6	66.0	65.8	64.4	**66.1**	**66.1**
Covtype	72.8	**93.2**	**93.2**	92.6	**93.2**	**93.2**
Credit	87.7	91.9	91.3	**93.0**	90.0	90.0
RLCP	10.3	**93.2**	**93.2**	93.0	**93.2**	93.1
Avg. rank	5.71	**2.13**	2.54	4.25	3.29	3.08

obtenido por el test de Friedman.

De la Tabla 8.2, se observa que ROS proporciona el mejor rango promedio de Friedman (el valor más bajo) en comparación con el resto de técnicas, seguido de RUS y la propuesta *GDcore*. Esto sugiere una aparente superioridad del algoritmo ROS, dado que el aumento de instancias de clase minoritaria extiende la frontera de decisión de esta, sin embargo, se incrementa el tamaño del conjunto de datos, en consecuencia la complejidad de clasificación también lo hace.

Si bien RUS obtuvo el segundo lugar en rango promedio de Friedman, el optar por técnicas aleatorias se debe considerar la perdida de información potencialmente significativa, dado que no se tiene una selección guiada de instancias. Al respecto, es preferible optar por técnicas que mantengan instancias por medio de una selección inteligente, como lo son las propuestas basadas en grafos *GDshape* y *GDcore*, las cuales, de acuerdo a los resultados obtenidos son viables, dado que tienen un rendimiento cercado a los métodos aleatorios.

Una ventaja de las propuestas basadas en grafos es la reducción del tamaño del conjunto de datos manteniendo aquellas instancias lo suficientemente cerca de la frontera de decisión, sin decrecer el rendimiento del clasificador.

Considerando los conjuntos HIGGS, SEA y poker0, se observa que tanto *GDshape* como *GDcore* obtuvieron los mejores resultados de clasificación. Este comportamiento sugiere que el uso de una representación por medio de vértices y aristas puede conducir a una mejor extracción de conocimiento debido a que la construcción del grafo se basa en la densidad de los datos, manteniendo esta relación en la selección de instancias a formar parte del conjunto balanceado.

Es interesante resaltar que SMOTE-MR fue el método con peor desempeño en la clasificación, esta situación es completamente diferente en pequeños conjuntos de datos, debido a que existen estudios donde SMOTE mejora el rendimiento [91]. Centrándose en conjuntos como poker1 y poker0, en ambos conjuntos de datos SMOTE obtiene un bajo rendimiento, esto podría suceder como consecuencia del procedimiento de interpolación.

Dado que SMOTE crea instancias sintéticas enfocadas en el espacio de características, y en ambos conjuntos de datos sus características son discretas, en consecuencia las nuevas instancias no brindan suficiente conocimiento sobre conjuntos de datos en contexto de *Big Data*.

En particular, para conjuntos de *Big Data* un método de sobre-muestreo basado en el espacio de características no conduce a mejores resultados, de hecho, es necesario considerar el espacio de datos. Dado que se maneja una gran cantidad de datos, se manifiesta que al igual que en "small data", en grandes volúmenes de datos se mantienen problemas como el solapamiento de clases, donde es evidente que la frontera de decisión no está clara dada la redundancia de instancias que *Big Data* podría tener.

Por otro lado, debido a que ambas propuestas basadas en grafos obtienen en un 33.33 % de los conjuntos de datos un mejor rendimiento. Con el fin de facilitar la comparación entre métodos para identificar si existe una significancia estadística entre los resultados, se aplica la prueba de Wilcoxon, la cual se resume en la Tabla 8.3.

En esta tabla, la mitad superior de la diagonal de la Tabla corresponde a los resultados para el valor $\alpha = 0.9$, mientras que la mitad inferior de la diagonal es para un nivel 0.95. El símbolo "•" indica que el método de la fila fue significativamente mejor que el método de la columna, mientras que el símbolo "◦" representa que el método de la columna funcionó significativamente mejor que el método de la fila.

Tabla 8.3: Test de Wilcoxon para resultados en conjuntos de Big Data.

	Original	ROS	RUS	SMOTE-MR	GDshape	GDcore
Original	-	◦	◦	◦	◦	◦
ROS	•	-			•	•
RUS	•		-			
SMOTE-MR	•			-		◦
GDshape	•	◦			-	
GDcore	•					-
$\alpha = 0.9$	0	3	1	1	1	2
$\alpha = 0.95$	0	2	1	1	1	1

Los resultados de la Tabla 8.3 muestran que ROS se desempeñó significativamente mejor en comparación de técnicas basadas en grafos para un nivel de $\alpha = 0.9$. Sin embargo, para el otro nivel de significancia no hubo diferencias estadísticamente significativas entre ROS y GDcore. En consecuencia, se puede decir que todos estos algoritmos funcionaron por igual para ambos niveles de significancia. Por lo tanto, se puede concluir que los métodos basados en grafos pueden considerarse como una solución competitiva para el tratamiento de desbalance de clases en contexto de *Big Data*.

Parte IV

Conclusiones

Capítulo 9

Conclusiones y trabajo a futuro

Desde hace años, el problema del desbalance en la distribución de clases se ha abordado a partir de diversas estrategias. Sin embargo, este problema sigue siendo hoy día tema de estudio, ya que desafortunadamente en algunos casos, la presencia del desbalance de clases no es el único problema inherente en los datos, pudiendo existir otros problemas como la presencia de ruido o solapamiento entre clases que tienen un efecto negativo sobre el rendimiento de los clasificadores.

En este sentido, la gestión de la calidad de los datos implica utilizar algunas técnicas de preprocesamiento de datos para aumentar la calidad de éstos [92]. Su objetivo es filtrar o corregir las imperfecciones para que los clasificadores puedan usar de manera eficaz un conjunto de datos libre de complejidades [1].

En esta tesis, se desarrollaron nuevos algoritmos que permiten tratar eficientemente el problema del desbalance de clases, así como la presencia de ruido y/o solapamiento entre clases. Las propuestas van encaminadas al aprovechamiento de técnicas basadas en la teoría de grafos y en algoritmos de *clustering*, de tal forma que la combinación de ambos disminuyan las irregularidades de los datos con el fin de incrementar los índices de precisión de los modelos de aprendizaje.

A lo largo de esta tesis se incluyen estudios empíricos para corroborar la eficiencia de los nuevos algoritmos propuestos, de los cuales se pueden destacar algunas contribuciones. En las siguientes secciones, se resumen las aportaciones y se discuten las líneas abiertas de estudio que puedan dar lugar a nuevos enfoques de investigación.

9.1. Conclusiones

El objetivo principal de esta tesis es el desarrollo de nuevos algoritmos basados en *clustering* y grafos para el tratamiento de ciertas complejidades inherentes de los datos, tales como desbalance de clases, solapamiento entre clases y/o ruido.

En primera instancia, para el tratamiento del desbalance de clases, se desarrollaron dos nuevos métodos basados en grafos, denominados IG-US y MIST-US, así como la adaptación del algoritmo de *clustering* DBSCAN, para realizar el bajo-muestreo de la clase mayoritaria.

Comparando el desempeño de las propuestas desarrolladas con métodos de bajo-muestreo ampliamente utilizados en el estado del arte, se logra validar la efectividad de los métodos propuestos. Respecto a los algoritmos basados en grafos, fue posible obtener tasas de precisión por encima del 75 % en todos los modelos de aprendizaje usados (1NN, J48 y SVM). Por su parte, el algoritmo DBSCAN modificado para el tratamiento de desbalance de clases es competitivo para clasificadores como 1NN y J48 al obtener también precisiones superiores al 75 %; por el contrario, al utilizar un clasificador SVM, su comportamiento ha resultado ser deficiente puesto que genera precisiones inferiores al 45 %. Por lo tanto, las propuestas basadas en grafos son una alternativa ideal y novedosa, mientras que el uso de DBSCAN requiere de técnicas adicionales que ayuden a mejorar los resultados.

Para tratar conjuntamente tanto el desbalance entre clases como el ruido y el traslape de clases, se realizaron otras dos propuestas: DBIG-US y DBMIST-US. Estas propuestas constituyen en sí mismos métodos de dos fases, en el que se aprovechan los beneficios ofrecidos por DBSCAN para realizar la limpieza del conjunto de datos y la eficiencia del subgrafo inducido y el árbol mínimo de expansión para condensar la clase mayoritaria.

DBIG-US y DBMIST-US tienen enfoques diferentes en el proceso de refinado, por un lado se obtiene un subconjunto de clase mayoritaria que contenga todas aquellas instancias que están cerca de la frontera de decisión (DBIG-US) y por otro lado, obtener un subconjunto que contenga instancias que representen el núcleo de la clase mayoritaria (DBMIST-US).

Los resultados experimentales mostraron de forma contundente la eficiencia de DBIG-US y DBMIST-US como métodos de bajo-muestreo, en comparación con otros algoritmos probados. A partir de los resultados de la media geométrica, se obtienen las siguientes conclusiones:

- DBIG-US fue el método que obtuvo niveles de precisión más altos para los clasificadores 1NN y J48 en conjuntos de datos reales y sintéticos.
- DBMIST-US tuvo mejor rendimiento para los clasificadores 1NN, J48 y SVM en conjuntos de datos con grado de desbalance moderado a severo.

Para determinar la significancia estadística de los resultados obtenidos por todos los algoritmos propuestos, se utilizó el test de Wilcoxon para valores de $\alpha = 0.9$ y $\alpha = 0.95$. Los resultados mostraron que los algoritmos IG-US, MIST-US y DBSCAN pueden ser considerados como una alternativa de solución para el tratamiento del desbalance de clases, mientras que los métodos de dos fases DBIG-US y DBMIST-US son los más eficientes en la mayoría de los conjuntos de datos utilizados en comparación con los métodos del estado del arte incluidos en las pruebas experimentales. Dado que DBMIST-US obtuvo índices de precisión por encima del 80 % para conjuntos de datos reales y por encima del 50 % para conjuntos de datos sintéticos independientemente del clasificador usado, se puede concluir que esta propuesta es la mejor alternativa de entre los diversos algoritmos basados en *clustering* y grafos, ya que se aprovecha la limpieza

del conjunto de datos por DBSCAN y la conservación de instancias más alejadas de la frontera de decisión a través de un árbol de expansión mínimo.

El desequilibrio de clases sigue siendo un problema relevante, no solo en conjuntos de datos denominados *small data* sino también para conjuntos de gran volumen o *Big Data*. En este sentido, como primera aproximación al tratamiento del desbalance en *Big Data*, se escalaron algunos algoritmos de bajo-muestreo basados en grafos denominados GDshape y GDcore.

Para lograr la reducción de volumen de la clase mayoritaria, la propuesta GDshape mantiene instancias cercanas a la frontera de decisión, mientras que la propuesta GDcore es el caso contrario, mantiene todas aquellas instancias que están alejadas de la frontera de decisión de la clase mayoritaria.

Los resultados experimentales permiten notar que, a pesar de que una técnica de selección aleatoria como ROS tenga el mejor rendimiento, el aumento del volumen de datos la hace prohibitiva para Big Data. Por ello, aun cuando los algoritmos propuestos GDshape y GDcore no sean los que ofrezcan los mejores resultados, sí pueden ser una opción por disminuir el volumen de datos con los que se trabaja, manteniendo un rendimiento aceptable del clasificador.

Por último, teniendo como base la experimentación mostrada en los diferentes capítulos de la tesis, es posible afirmar que el tratamiento de las complejidades de datos sigue siendo una línea abierta de estudio tanto para *small data*, como para *big data*. De igual modo, utilizar estrategias de la teoría de grafos mostró de forma contundente su pertinencia. Además se pudo constatar que realizar la limpieza del conjunto de datos previo al bajo-muestreo es la mejor recomendación derivada de los experimentos mostrados en esta tesis.

9.2. Aportaciones a la ciencia

Las principales aportaciones que se pueden mencionar y son producto de esta tesis son las siguientes:

1. Generación de nuevo conocimiento, mediante la incorporación al estado del arte de siete algoritmos nuevos para el tratamiento del desbalance de clases.

2. Integración de dos áreas del conocimiento para el logro de un problema: Teoría de Grafos y Aprendizaje Automático.

3. Desarrollo de cuatro nuevos métodos de bajo-muestreo basados en grafos:

   a) Subgrafo inducido, IG-US;

   b) Árbol de expansión mínimo, MIST-US;

   c) Método de dos pasos, DBIG-US el cual combina la limpieza que realiza DBSCAN con el uso del subgrafo inducido IG-US;

d) Método de dos pasos, DBMIST-US que combina la limpieza de los datos por DBSCAN con el árbol de expansión mínimo MIST-US.

4. Adaptación del algoritmo de *clustering* DBSCAN para fines de limpieza de datos y al mismo tiempo como estrategia de bajo-muestreo.

5. Desarrollo de propuestas basada en grafos para el tratamiento del desbalance de clases en problemas de *Big Data*: grafo ponderado basado en la densidad de datos con dos variantes GDShape y GDcore.

9.3. Trabajo a futuro

Una vez finalizada la investigación enmarcada en esta tesis, a continuación se describen algunas de las principales líneas abiertas de estudio que podrían abordarse en el futuro:

- Dado que las propuestas basadas en grafos y *clustering* se desarrollaron en un contexto de aprendizaje supervisado, es evidente la necesidad de probar estos algoritmos en problemas de aprendizaje no supervisado e inclusive de aprendizaje semi-supervisado, para poder determinar la viabilidad en futuros escenarios.

- El estudio mostrado en esta tesis se enfocó a problemas de dos clases, por lo que una línea abierta es la necesidad de trasladar la concepción de trabajar en problemas con múltiples clases (más de dos clases).

- El uso de algoritmos de *clustering* como proceso de limpieza en problemas de *Big Data* abre una línea de estudio para determinar la pertinencia del uso de DBSCAN o incorporar alguna otra técnica de *clustering*.

- Analizar la pertinencia de escalar las propuestas DBIG-US y DBMIST-US a escenarios de *Big Data*.

- La utilización de la teoría de grafos en sí misma abre una línea de investigación poco explorada actualmente como solución a otras complejidades de datos tanto para *Big Data* como para *Small Data*.

- En escenarios de *Big Data*, se busca la adopción de estrategias de partición de datos y representación mediante un grafo por medio de modelos centrados en aristas o modelos centrados en vértices, orientados de acuerdo a la localidad de los datos.

- Considerar únicamente el espacio de características en propuestas de tratamiento del desbalance de clases para *Big Data* abre una línea de trabajo para tener en cuenta adicionalmente el espacio de búsqueda. Puesto que el incremento de instancias en un espacio de búsqueda reducido hace que éstas se encuentren muy

próximas entre si, se requiere implementar alguna métrica distinta, tal como la distancia coseno, que permita aportar mayor discrepancia entre instancias pertenecientes a diferentes clases.

9.4. Publicaciones resultantes

Las diferentes aportaciones y resultados obtenidos a lo largo de la investigación presentada en esta tesis han originado un número de contribuciones publicadas tanto en artículos de revistas indexadas como de divulgación científica, así como en congresos nacionales e internacionales:

1. Guzmán-Ponce, A., Valdovinos, R.M., Sánchez, J.S., & Marcial-Romero, J. (2020). *A New Under-Sampling Method to Face Class Overlap and Imbalance*. Applied Sciences [ISSN: 2076-3417], 10(15), Article 5164. `https://doi.org/10.3390/app10155164`

2. Guzmán-Ponce, A., Sánchez, J. S., Valdovinos, R. M., & Marcial-Romero, J. R. (2020). *DBIG-US: A two-stage under-sampling algorithm to face the class imbalance problem*. Expert Systems with Applications [ISSN: 0957-4174], Article 114301. `https://doi.org/10.1016/j.eswa.2020.114301`

3. Guzmán-Ponce A., Valdovinos R.M. & Sánchez J.S. (2020). *A Cluster-Based Under-Sampling Algorithm for Class-Imbalanced Data*. Hybrid Artificial Intelligent Systems. Lecture Notes in Computer Science, 12344, 299-311, Gijón, Spain. `https://doi.org/10.1007/978-3-030-61705-9_25`

4. Guzmán-Ponce, A., Marcial-Romero, J. R., Valdovinos, R. M., & Sánchez, J. S. (2020). *Weighted Complete Graphs for Condensing Data*. Electronic Notes in Theoretical Computer Science [ISSN: 1571-0661], 354, 45-60. `https://doi.org/10.1016/j.entcs.2020.10.005`

5. Guzmán-Ponce, A., Valdovinos, R.M., Montenegro H., Marcial-Romero, J.,& Sánchez, J.S. (2019). *Tendencias del preprocesado en grandes volúmenes de datos*. Komputer Sapiens [ISSN: 2007-0691], 3, 16-20.

6. Guzmán-Ponce, A., Valdovinos, R.M., Marcial-Romero, J., & Alejo-Eleuterio, R. (2018). *Entornos de trabajo para procesamiento de datos masivos y aprendizaje automático*. Research in Computing Science [ISSN 1870-4069], 147(5), 225-237.

Bibliografía

[1] V. García, J. Sánchez, A. Marqués, R. Florencia, and G. Rivera, "Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data," *Expert Systems with Applications*, p. 113026, 2019.

[2] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.

[3] V. López, A. Fernández, J. Moreno-Torres, and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6585–6608, 2012.

[4] V. García, J. S. Sánchez, and R. A. Mollineda, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 13–21, 2012.

[5] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

[6] A. Fernández, S. García, M. Galar, P. R, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*, vol. 1. Cham, Switzerland: Springer International Publishing, 2018.

[7] A. Samaddar, T. Goswami, S. Ghosh, and S. Pal, "An algorithm to input and store wider classes of chemical reactions for mining chemical graphs," in *IEEE International Advance Computing Conference*, (Banglore, India), pp. 1082–1086, 2015.

[8] D. Turvill, L. Barnby, and A. Anjum, "A conceptual framework for the use of graph representation within high energy physics analysis," in *18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, (Washington, DC), pp. 384–385, 2018.

[9] L. Hassani, M. R. Moosavi, and P. Setoodeh, "A Graph Based Approach to Analyse Metabolic Networks for Strain Engineering," in *27th Iranian Conference on Electrical Engineering (ICEE)*, (Yazd, Iran), pp. 1839–1843, 2019.

[10] A. González, E. Barra, A. Beghelli, and A. Leiva, "A sub-graph mapping-based algorithm for virtual network allocation over flexible grid networks," in *17th International Conference on Transparent Optical Networks*, (Budapest, Hungary), pp. 1–4, 2015.

[11] Z. Zhang and E. R. H. , "Localized graph-based feature selection for clustering," in *Image Analysis and Recognition* (A. Campilho and M. Kamel, eds.), (Berlin, Heidelberg), pp. 1–10, 2012.

[12] J. Maillo, J. Luengo, S. García, F. Herrera, and I. Triguero, "A preliminary study on hybrid spill-tree fuzzy k-nearest neighbors for big data classification," in *IEEE International Conference on Fuzzy Systems*, (Rio de Janeiro, Brazil), pp. 1–8, 2018.

[13] J. Kang, R. Fernandez-Beltran, D. Hong, J. Chanussot, and A. Plaza, "Graph relation network: Modeling relations between scenes for multilabel remote-sensing image classification and retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2020.

[14] V. J. Carey, *Machine Learning Concepts and Tools for Statistical Genomics*, pp. 273–292. Springer New York, 2005.

[15] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 1 ed., 2006.

[16] A. L'Heureux, K. Grolinger, H. F. ElYamany, and M. Capretz, "Machine Learning with Big Data: Challenges and Approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017.

[17] R. S. Michalski, J. Carbonell, and T. M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*. Springer-Verlag Berlin Heidelberg, 2013.

[18] M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, pp. 1–21, 2015.

[19] T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, *Machine Learning: A Guide to Current Research*, vol. 12. Springer US, 1986.

[20] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, vol. 1. Cham, Switzerland: Springer International Publishing, 2015.

[21] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 3, 2017.

[22] L. Cruz, a. López-Chau, and J. López, "árbol de decisión c4. 5 basado en entropía minoritaria para clasificación de conjuntos de datos no balanceados," *Research in Computing Science*, vol. 92, pp. 23–34, 2015.

[23] J. Chorowski, J. Wang, and J. M. Zurada, "Review and performance comparison of svm- and elm-based classifiers," *Neurocomputing*, vol. 128, pp. 507–516, 2014.

[24] T. Jebara, *Machine Learning: Discriminative and Generative*, vol. 755. Springer US, 2012.

[25] O. Bousquet, U. V. Luxburg, and G. Rätsch, *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, vol. 3176. Springer, 2011.

[26] S. T. Bow, *Pattern Recognition and Image Preprocessing*. New York: Marcel Dekker, 2002.

[27] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett, "Novel centroid selection approaches for kmeans-clustering based recommender systems," *Information Sciences*, vol. 320, pp. 156–189, 2015.

[28] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *2nd International Conference on Knowledge Discovery and Data Mining*, (Portland, OR), pp. 226–231, AAAI Press, 1996.

[29] A. Smiti and Z. Elouedi, "DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques," in *IEEE 16th International Conference on Intelligent Engineering Systems*, (Lisbon, Portugal), pp. 573–578, 2012.

[30] M. Berry and G. Linoff, *Data Mining Techniques*. John Wiley & Sons, 2009.

[31] L. Cleofas-Sánchez, *Tratamiento de la complejidad de patrones de datos en cúmulos de información, con memorias asociativas*. PhD thesis, Instituto Politécnico Nacional, México, 2017.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[33] J. Zhai, S. Zhang, M. Zhang, and X. Liu, "Fuzzy integral-based ELM ensemble for imbalanced big data classification," *Soft Computing*, vol. 22, no. 11, pp. 3519–3531, 2018.

[34] F. Hu, H. Li, H. Lou, and J. Dai, "A parallel oversampling algorithm based on NRSBoundary-SMOTE," *Journal of Information and Computational Science*, vol. 11, no. 13, pp. 4655–4665, 2014.

[35] N. J. Horton and S. R. Lipsitz, "Multiple Imputation in Practice: Comparison of Software Packages for Regression Models with Missing Variables," *The American Statistician*, vol. 55, no. 3, pp. 244–254, 2001.

[36] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*, vol. 1. Cham, Switzerland: Springer International Publishing, 2015.

[37] X. Dong, H. He, C. Li, Y. Liu, and H. Xiong, "Scene-Based Big Data Quality Management Framework," in *Data Science* (Q. Zhou, Y. Gan, W. Jing, X. Song, Y. Wang, and Z. Lu, eds.), vol. 901, (Zhengzhou, China), pp. 122–139, Springer, 2018.

[38] P. Lenca, S. Lallich, T. Do, and N. Pham, "A comparison of different off-centered entropies to deal with class imbalance for decision trees," in *Advances in Knowledge Discovery and Data Mining*, (Osaka, Japan), pp. 634–643, Springer, 2008.

[39] J. Bondy and U. Murty, *Graph Theory with Applications*, vol. 290. London, UK: Macmillan, 1976.

[40] M. S. Rahman, *Basic Graph Theory*. 1863-7310, Cham, Switzerland: Springer International Publishing, 2017.

[41] J. Gross and J. Yellen, *Graph Theory and Its Applications*. CRC Press, 2005.

[42] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[43] D. L. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408–421, 1972.

[44] P. Hart, "The Condensed Nearest Neighbor Rule," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[45] I. Tomek, "An Experiment with the Edited Nearest-Neighbor Rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 6, pp. 448–452, 1976.

[46] I. Tomek, "Two modifications of CNN," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, no. 11, pp. 769–772, 1976.

[47] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," in *14th International Conference on Machine Learning*, (Nashville, TN), pp. 179–186, Morgan Kaufmann, 1997.

[48] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," in *8th Conference on Artificial Intelligence in Medicine*, (Cascais, Portugal), pp. 63–66, Springer, 2001.

[49] A. Hoffmann, "Artificial and natural computation," in *International Encyclopedia of the Social and Behavioral Sciences* (N. J. Smelser and P. B. Baltes, eds.), pp. 777–783, Oxford: Pergamon, 2001.

[50] S. García, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: A scaling up approach," *Pattern Recognition*, vol. 41, no. 8, pp. 2693–2709, 2008.

[51] S. García and F. Herrera, "Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy," *Evolutionary Computation*, vol. 17, no. 3, pp. 275–306, 2009.

[52] J. Derrac, C. Cornelis, S. GarcÍa, and F. Herrera, "Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection," *Information Sciences*, vol. 186, no. 1, pp. 73–92, 2012.

[53] J. Ha and J. Lee, "A New Under-Sampling Method Using Genetic Algorithm for Imbalanced Data Classification," in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, IMCOM 16, (New York, NY, USA), pp. 1–6, Association for Computing Machinery, 2016.

[54] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[55] X. Liu, J. Wu, and Z. Zhou, "Exploratory Undersampling for Class-Imbalance Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[56] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling," *Pattern Recognition*, vol. 46, no. 12, pp. 3460–3471, 2013.

[57] F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid, "CUSBoost: Cluster-Based Under-Sampling with Boosting for Imbalanced Classification," in *2nd International Conference on Computational Systems and Information Technology for Sustainable Solution*, (Bangalore, India), pp. 1–5, 2017.

[58] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4263–4274, 2017.

[59] N. Ofek, L. Rokach, R. Stern, and A. Shabtai, "Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem," *Neurocomputing*, vol. 243, pp. 88–102, 2017.

[60] S. Yen and Y. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 5718–5727, 2009.

[61] R. Longadge, S. S. Dongre, and L. Malik, "Multi-cluster based approach for skewed data in data mining," *IOSR Journal of Computer Engineering*, vol. 12, no. 6, pp. 66–73, 2013.

[62] V. H. Barella, E. P. Costa, and A. C. P. L. F. Carvalho, "ClusterOSS: a new undersampling method for imbalanced learning," in *Proceedings of the 3rd Brazilian Conference on Intelligent Systems*, (São Carlos, Brazil), pp. 453–458, 2014.

[63] R. A. Sowah, M. A. Agebure, G. A. Mills, K. M. Koumadi, and S. Y. Fiawoo, "New cluster undersampling technique for class imbalance learning," *International Journal of Machine Learning and Computing*, vol. 6, no. 3, p. 205, 2016.

[64] B. Das, N. C. Krishnan, and D. J. Cook, *Handling imbalanced and overlapping classes in smart environments prompting dataset*, pp. 199–219. Berlin, Heidelberg: Springer, 2014.

[65] C.-F. Tsai, W.-C. Lin, Y.-H. Hu, and G.-T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," *Information Sciences*, vol. 477, pp. 47–54, 2019.

[66] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409-410, pp. 17–26, 2017.

[67] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," in *Proceedings of 8th Conference on Artificial Intelligence in Medicine*, (Cascais, Portugal), pp. 63–66, Springer, 2001.

[68] K. J. Cios, R. W. Swiniarski, W. Pedrycz, and L. A. Kurgan, *Data Mining: A Knowledge Discovery Approach*. Springer, Boston, MA, 1 ed., 2007.

[69] P. D. Gutiérrez, M. Lastra, J. M., and F. Herrera, "SMOTE-GPU: Big Data preprocessing on commodity hardware for imbalanced classification," *Progress in Artificial Intelligence*, vol. 6, no. 4, pp. 347–354, 2017.

[70] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. Cambridge, MA: Morgan Kaufmann, 2017.

[71] S. García, A. Fernández, J. Luengo, and F. Herrera, "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, no. 10, pp. 2044–2064, 2010. Special Issue on Intelligent Distributed Information Systems.

[72] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, 2019.

[73] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[74] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[75] J. H. Zar, *Biostatistical Analysis: Pearson New International Edition*. Pearson, 5 ed., 2009.

[76] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a resampling method with filtering," *Information Sciences*, vol. 291, pp. 184–203, 2015.

[77] S. del Río, V. López, J. M. Benítez, and F. Herrera, "On the use of MapReduce for imbalanced big data using Random Forest," *Information Sciences*, vol. 285, pp. 112–137, 2014.

[78] J. Zhai, S. Zhang, and C. Wang, "The classification of imbalanced large data sets based on MapReduce and ensemble of ELM classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 1009–1017, 2017.

[79] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006. Neural Networks.

[80] X. Wang, X. Liu, and S. Matwin, "A distributed instance-weighted SVM algorithm on large-scale imbalanced datasets," in *IEEE International Conference on Big Data*, (Washington, United States), pp. 45–51, 2014.

[81] R. C. Bhagat and S. S. Patil, "Enhanced SMOTE algorithm for classification of imbalanced big-data using Random Forest," in *IEEE International Advance Computing Conference*, (Banglore, India), pp. 403–408, 2015.

[82] S. S. Patil and S. P. Sonavane, "Enhanced Over_Sampling Techniques for Imbalanced Big Data Set Classification," *Journal of Big Data*, vol. 4, no. 49, pp. 49–81, 2017.

[83] A. Fernández, S. del Río, N. V. Chawla, and F. Herrera, "An insight into imbalanced Big Data classification: outcomes and challenges," *Complex & Intelligent Systems*, vol. 3, no. 2, pp. 105–120, 2017.

[84] M. J. Basgall, W. Hasperué, M. Naiouf, A. Fernández, and F. Herrera, "SMOTE-BD: An Exact and Scalable Oversampling Method for Imbalanced Classification in Big Data," *Journal of Computer Science and Technology*, vol. 18, no. 3, pp. 23–28, 2018.

[85] M. J. Basgall, W. Hasperué, M. Naiouf, A. Fernández, and F. Herrera, "An Analysis of Local and Global Solutions to Address Big Data Imbalanced Classification: A Case Study with SMOTE Preprocessing," in *Cloud Computing and Big Data* (M. Naiouf, F. Chichizola, and E. Rucci, eds.), (La Plata, Buenos Aires, Argentina), pp. 75–85, Springer International Publishing, 2019.

[86] J. Maillo, S. Ramírez, I. Triguero, and F. Herrera, "kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data," *Knowledge-Based Systems*, vol. 117, pp. 3–15, 2017. Volume, Variety and Velocity in Data Science.

[87] I. Triguero, M. Galar, D. Merino, J. Maillo, H. Bustince, and F. Herrera, "Evolutionary undersampling for extremely imbalanced big data classification under apache spark," in *IEEE Congress on Evolutionary Computation*, (Vancouver, Canada), pp. 640–647, 2016.

[88] J. Jedrzejowicz, R. Kostrzewski, J. Neumann, and M. Zakrzewska, "Imbalanced data classification using MapReduce and relief," *Journal of Information and Telecommunication*, vol. 2, pp. 217–230, 2018.

[89] J. Maillo, I. Triguero, and F. Herrera, "Redundancy and Complexity Metrics for Big Data Classification: Towards Smart Data," *IEEE Access*, vol. 8, pp. 87918–87928, 2020.

[90] M. Lichman *et al.*, "UCI Machine Learning Repository, 2013," 2013.

[91] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018.

[92] D. Becker, T. D. King, and B. McMullen, "Big data, big data quality problem," in *IEEE International Conference on Big Data*, (Santa Clara, CA, USA), pp. 2644–2653, 2015.

Apéndices

Apéndice A

Resultados de clasificación para propuestas basadas en grafos

Este apéndice resume los resultados de clasificación obtenidos con los algoritmos basados en grafos en comparación con aquellos producidos por métodos de bajo-muestreo del estado del arte, usando los modelos 1NN, J48 y SVM. En las diferentes tablas que se muestran a continuación, los valores en negrita resaltan el mejor resultado para cada par formado por conjunto de datos y algoritmo de bajo-muestreo.

Tabla A.1: Media geométrica obtenida por IG-US con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US
1	61.2	73.5	50.7	83.0	69.0	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	82.1	73.5	58.9	**93.7**
2	**100**	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	**100**
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	85.2	53.1	69.6	47.1	51.7	33.0	**91.1**
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	78.9	66.1	51.9	66.9	**93.9**
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6
6	62.3	74.8	46.2	76.8	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	**94.9**
7	86.6	79.9	65.3	**91.3**	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	88.4
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	94.9
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.1
10	81.0	77.0	65.1	**100**	80.9	**100**	79.3	94.3	94.3	94.3	91.0	92.6	94.0	81.7	**100**	99.5	78.1	88.9
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	**100**	91.3	74.5	**100**	**100**	94.1	91.3
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	72.1	57.0	63.0	61.2	59.8	64.9	**90.6**
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72.0	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	94.3
14	77.0	59.8	58.3	80.6	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	**92.2**
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	**85.5**	26.1	77.8	81.7	74.9	0	76.6
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	84.3	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	**88.4**
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	69.9	65.0	61.5	47.6	51.4	61.2	69.6	69.2	**94.9**
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	86.3
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	**100**	84.2	78.3	**100**	83.0	83.7	65.5
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	92.9	48.7	44.8	**97.2**
21	71.1	69.9	54.8	86.1	79.0	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	88.0	**93.5**
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	98.4
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	66.1	39.8	64.0	87.7
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	84.13	78.80	67.80	**91.68**
Avg. Rank	12.79	9.06	16.10	6.96	11.48	7.79	10.63	7.92	9.02	8.75	8.02	7.58	10.04	10.94	6.65	9.25	12.46	**5.56**

Tabla A.2: Media geométrica obtenida por IG-US con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77.0	70.5	74.3	85.2	85.6	72.9	71.3	**87.9**
2	**96.9**	92.8	80.3	94.4	**96.9**	97.5	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	96.0
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.6	52.0	81.5	33.3	66.1	4.6	**94.1**
4	53.1	90.8	22.4	57.7	33.2	58.0	52.4	55.2	73.0	61.1	65.2	79.4	58.2	94.1	52.7	66.7	65.0	**97.0**
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	**100**	97.0	**100**
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	81.5	67.4	58.3	62.8	78.3	69.8	56.5	68.4	**84.9**
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	**92.3**	**92.3**	89.1	91.7	72.4	80.7	66.7	82.3	9.0	80.7
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	92.5	86.3	95.0	77.1	87.5	91.3	85.8	**93.5**	81.4
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	94.3
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79.0	83.1
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	**100**
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	77.1	64.3	**91.3**
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	83.1
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4.0	79.4	67.1	79.1
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	**91.9**	0	83.7	89.0	74.5	0	78.9
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	**88.2**	82.9	87.3	80.9	82.3
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	79.7	61.2	51.2	68.4	**96.5**
18	86.3	**96.6**	77.8	96.4	89.0	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	87.7
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	**100**	84.3	72.4	57.1
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74.0	75.5	74.1	57.0	75.7	91.4	32.1	45.9	**92.7**
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	90.8
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	**100**
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	95.0
24	0	47.8	0	0	0	0	0	44.0	40	42.3	44.6	56.5	0	68.6	**69.7**	44.7	52.5	56.6
Avg.																		
Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	**88.04**	77.21	79.81	63.50	87.10
Avg. Rank	11.75	8.46	14.62	10.33	11.40	9.54	12.02	8.23	8.10	8.23	8.10	6.56	11.40	**5.75**	9.29	7.94	11.92	7.35

Tabla A.3: Media geométrica obtenida por IG-US con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	80.4	81.4	41.8	30.1	**93.0**
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	94.4	91.3	91.1	**99.0**	77.2	86.2	93.4
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	**72.8**	0	0	0	68.3
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	80.4	61.2	0	29.5	**84.0**
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	99.6
6	0	81.2	0	0	0	0	0	83.3	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	**87.9**
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	84.3	84.3	84.3	**87.6**	82.9	39.2	65.3	81.6	61.8	0	53.8
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	86.6	99.0	92.2
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	90.3
10	0	81.6	73.9	0	0	0	0	**94.3**	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	66.7
11	90.9	91.3	**100**	91.3	90.9	91.3	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	91.3
12	0	60.6	0	0	0	0	0	58.9	67.8	49.4	47.3	65.3	0	58.9	75.0	0	60.5	**84.5**
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	**100**	0	81.6	81.6	0	43.1	74.5
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	74.2
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	**86.6**	0	77.5	81.0	50.3	0	73.2
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	70.8
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	77.5	75.0	0	43.1	**98.3**
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	89.4
19	84.1	78.2	78.7	84.2	84.0	84.2	83.5	92.6	84.5	84.5	84.1	89.4	84.0	84.5	**100**	83.8	82.8	60.6
20	0	74.1	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	94.3	0	0	**98.1**
21	0	78.5	0	0	0	0	0	88.4	81.3	87.1	87.7	**91.4**	0	85.7	88.3	0	86.1	75.0
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	**100**	96.9	99.2	85.9
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	98.4
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	**58.3**	0	2.5	53.6
Avg.																		
Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	**81.55**	42.75	46.58	81.54
Avg. Rank	14.54	6.58	12.69	12.94	14.31	13.25	13.63	**4.54**	4.94	5.29	6.56	5.71	13.67	6.81	4.60	12.17	11.63	7.15

Tabla A.4: Media geométrica obtenida por MIST-US con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	MIST-US
1	61.2	73.5	50.7	**83.0**	69.0	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	82.1	73.5	58.9	75.8
2	**100**	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	98.8
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	**85.2**	53.1	69.6	47.1	51.7	33.0	63.4
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	**78.9**	66.1	51.9	66.9	68.6
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6
6	62.3	74.8	46.2	**76.8**	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	70.7
7	86.6	79.9	65.3	91.3	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	**96.1**
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	**100**
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.2
10	81.0	77.0	65.1	**100**	80.9	**100**	79.3	94.3	94.3	94.3	91.0	92.6	94.0	81.7	**100**	99.5	78.1	94.9
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	**100**	91.3	74.5	**100**	**100**	94.1	92.6
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	**72.1**	57.0	63.0	61.2	59.8	64.9	70.5
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72.0	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	94.9
14	77.0	59.8	58.3	**80.6**	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	65.5
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	**85.5**	26.1	77.8	81.7	74.9	0	82.4
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	**84.3**	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	72.3
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	69.9	65.0	61.5	47.6	51.4	61.2	69.6	69.2	**73.0**
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	97.5
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	**100**	84.2	78.3	**100**	83.0	83.7	87.3
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	92.9	48.7	44.8	**89.9**
21	71.1	69.9	54.8	86.1	79.0	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	**88.0**	79.2
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	**100**
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	**66.1**	39.8	64.0	49.6
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	84.13	78.80	67.80	**84.20**
Avg. Rank	12.83	9.06	16.13	6.88	11.56	7.75	10.65	7.81	9.08	8.71	7.92	7.44	10	10.96	6.5	9.25	12.42	**6.06**

Tabla A.5: Media geométrica obtenida por MIST-US con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	MIST-US
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77.0	70.5	74.3	85.2	**85.6**	72.9	71.3	73.0
2	96.9	92.8	80.3	94.4	96.9	**97.5**	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	96.5
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.64	52.0	**81.5**	33.3	66.1	4.6	33.4
4	53.1	90.8	22.4	57.7	33.2	58.0	52.4	55.2	73.0	61.1	65.2	79.4	58.2	**94.1**	52.7	66.7	65.0	82.4
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	**100**	97.0	**100**
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	**81.5**	67.4	58.3	62.8	78.3	69.8	56.5	68.4	70.9
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	92.3	92.3	89.1	91.7	72.4	80.7	66.7	82.3	9.0	**96.1**
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	92.5	86.3	95.0	77.1	87.5	91.3	85.8	**93.5**	82.8
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	98.2
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79.0	**100**
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	**100**
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	**77.1**	64.3	62.5
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	**100**
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4.0	79.4	67.1	64.7
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	**91.9**	0	83.7	89.0	74.5	0	89.4
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	**88.2**	82.9	87.3	80.9	77.3
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	**79.7**	61.2	51.2	68.4	66.6
18	86.3	**96.6**	77.8	96.4	89.0	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	95.2
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	**100**	84.3	72.4	94.3
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74.0	75.5	74.1	57.0	75.7	**91.4**	32.1	45.9	88.7
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	85.3
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	**100**
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	96.6
24	0	47.8	0	0	0	0	0	44.0	40.0	42.3	44.6	56.5	0	68.6	**69.7**	44.7	52.5	44.4
Avg. Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	**88.04**	77.21	79.81	63.50	83.26
Avg. Rank	11.92	8.46	14.71	10.46	11.56	9.71	12.21	8.23	8.31	8.33	8.15	6.58	11.5	**5.73**	9.29	8.06	11.79	6

Tabla A.6: Media geométrica obtenida por MIST-US con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	MIST-US
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	**80.4**	81.4	41.8	30.1	74.8
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	**94.4**	91.3	91.1	99.0	77.2	86.2	92.7
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	**72.8**	0	0	0	43.1
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	**80.4**	61.2	0	29.5	67.6
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	**100**
6	0	81.2	0	0	0	0	0	**83.3**	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	76.9
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	84.3	84.3	84.3	87.6	82.9	39.2	65.3	81.6	61.8	0	**96.1**
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	86.6	99.0	**100**
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	84.5
10	0	81.6	73.9	0	0	0	0	94.3	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	**94.9**
11	90.9	91.3	**100**	91.3	90.9	91.3	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	**100**
12	0	60.6	0	0	0	0	0	58.9	67.8	49.4	47.3	65.3	0	58.9	75.0	0	60.5	**72.7**
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	**100**	0	81.6	81.6	0	43.1	**100**
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	63.6
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	**86.6**	0	77.5	81.0	50.3	0	84.4
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	78.0
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	**77.5**	75.0	0	43.1	71.6
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	95.0
19	84.1	78.2	78.7	84.2	84.0	84.2	83.5	92.6	84.5	84.5	84.1	89.4	84.0	84.5	**100**	83.8	82.8	92.6
20	0	74.1	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	**94.3**	0	0	89.9
21	0	78.5	0	0	0	0	0	**88.4**	81.3	87.1	87.7	91.4	0	85.7	88.3	0	86.1	83.7
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	**100**	96.9	99.2	94.8
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	**100**
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	**58.3**	0	2.5	**58.3**
Avg. Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	81.55	42.75	46.58	**83.97**
Avg. Rank	14.60	6.77	12.85	13.04	14.40	13.35	13.73	4.73	5.19	5.5	6.83	5.88	13.75	7.06	4.69	12.27	11.75	**4.60**

Apéndice B

Resultados de clasificación para DBSCAN como estrategia de limpieza

Este apéndice resume los resultados de clasificación del algoritmo DBSCAN como proceso de limpieza, comparándolos con los resultados de los métodos de bajo-muestreo del estado del arte, usando los modelos 1NN, J48 y SVM. En las diferentes tablas que se muestran a continuación, los valores en negrita resaltan el mejor resultado para cada par formado por conjunto de datos y algoritmo de bajo-muestreo.

Por otra parte, se incluyen también los resultados correspondientes a las propuestas de esta tesis basadas en grafos y DBSCAN como estrategias de bajo-muestreo.

Tabla B.1: Media geométrica obtenida por DBSCAN con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBSCAN
1	61.2	73.5	50.7	83.0	69.0	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	**82.1**	73.5	58.9	62.9
2	**100**	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	99.9
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	**85.2**	53.1	69.6	47.1	51.7	33.0	67.7
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	**78.9**	66.1	51.9	66.9	64.8
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6
6	62.3	74.8	46.2	**76.8**	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	57.9
7	86.6	79.9	65.3	**91.3**	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	89.4
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	86.3
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.2
10	81.0	77.0	65.1	**100**	80.9	**100**	79.3	94.3	94.3	94.3	91	92.6	94.0	81.7	**100**	99.5	78.1	94.3
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	**100**	91.3	74.54	**100**	**100**	94.1	91.3
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	**72.1**	57.0	63.0	61.2	59.8	64.9	37.3
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72.0	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	94.3
14	77.0	59.8	58.3	**80.6**	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	78.3
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	**85.5**	26.1	77.8	81.7	74.9	0	77.9
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	**84.3**	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	55.4
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	**69.9**	65.0	61.5	47.6	51.4	61.2	69.6	69.2	52.8
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	82.8
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	**100**	84.2	78.3	**100**	83	83.7	83.7
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	**92.9**	48.7	44.8	44.7
21	71.1	69.9	54.8	86.1	79.0	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	**88**	70.9
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	**100**
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	66.1	39.8	64.0	**80.5**
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	**84.13**	78.80	67.80	77.95
Avg. Rank	12.60	8.81	16.04	6.60	11.19	7.35	10.35	7.58	8.79	8.46	7.79	7.27	9.77	10.71	**6.31**	8.96	12.19	10.21

Tabla B.2: Media geométrica obtenida por DBSCAN con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBSCAN
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77.0	70.5	74.3	85.2	**85.6**	72.9	71.3	60.7
2	96.9	92.8	80.3	94.4	96.9	**97.5**	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	97.2
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.64	52.0	**81.5**	33.3	66.1	4.6	63.0
4	53.1	90.8	22.4	57.7	33.2	58.0	52.4	55.2	73.0	61.1	65.2	79.4	58.2	**94.1**	52.7	66.7	65.0	64.0
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	**100**	97.0	99.6
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	81.5	67.4	58.3	62.8	**78.3**	69.8	56.5	68.4	50.8
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	**92.3**	**92.3**	89.1	91.7	72.4	80.7	66.7	82.3	9.0	**92.3**
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	92.5	86.3	**95.0**	77.1	87.5	91.3	85.8	93.5	80.1
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	96.1
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79.0	99.7
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	91.3
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	**77.1**	64.3	0
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	99.4
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4.0	79.4	67.1	21.0
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	**91.9**	0	83.7	89.0	74.5	0	73.5
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	88.2	82.9	**87.3**	80.9	56.0
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	**79.7**	61.2	51.2	68.4	45.2
18	86.3	**96.6**	77.8	96.4	89.0	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	87.3
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	**100**	84.3	72.4	84.4
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74	75.5	74.1	57.0	75.7	**91.4**	32.1	45.9	33.3
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	70.4
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	**100**
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	98.3
24	0	47.8	0	0	0	0	0	44.0	40.0	42.3	44.6	56.5	0	68.6	69.7	44.7	52.5	**73.0**
Avg. Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	**88.04**	77.21	79.81	63.50	72.36
Avg. Rank	11.67	8.15	14.63	10.02	11.31	9.31	12	8	7.90	8.10	7.81	6.33	11.13	**5.52**	9.13	7.75	11.55	10.71

Tabla B.3: Media geométrica obtenida por DBSCAN con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBSCAN
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	80.4	**81.4**	41.8	30.1	0
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	94.4	91.3	91.1	**99.0**	77.2	86.2	89.0
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	**72.8**	0	0	0	34.2
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	**80.4**	61.2	0	29.5	33.3
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	**100**
6	0	81.2	0	0	0	0	0	**83.3**	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	0
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	84.3	84.3	84.3	87.6	82.9	39.2	65.3	81.6	61.8	0	**87.7**
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	86.6	99.0	63.2
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	65.4
10	0	81.6	73.9	0	0	0	0	**94.3**	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	13.3
11	90.9	91.3	**100**	91.3	90.9	91.3	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	91.3
12	0	60.6	0	0	0	0	0	58.9	**67.8**	49.4	47.3	65.3	0	58.9	75.0	0	60.5	0
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	**100**	0	81.6	81.6	0	43.1	0
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	74.2
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	**86.6**	0	77.5	81.0	50.3	0	71.4
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	0
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	**77.5**	75.0	0	43.1	0
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	0
19	84.1	78.2	78.7	84.2	84	84.2	83.5	92.6	84.5	84.5	84.1	89.4	84.0	84.5	**100**	83.8	82.8	84.1
20	0	74.1	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	**94.3**	0	0	0
21	0	78.5	0	0	0	0	0	88.4	81.3	87.1	87.7	**91.4**	0	85.7	88.3	0	86.1	0
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	**100**	96.9	99.2	97.0
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	98.3
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	**58.3**	0	2.5	20.0
Avg. Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	**81.55**	42.75	46.58	42.60
Avg. Rank	14.38	6.31	12.48	12.69	14.17	13	13.42	**4.23**	4.63	4.94	6.35	5.48	13.44	6.54	4.38	11.96	11.40	11.23

Tabla B.4: Comparativa de propuestas de bajo-muestreo con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US	MIST-US	DBSCAN
1	61.2	73.5	50.7	83.0	69	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	82.1	73.5	58.9	**93.7**	75.8	62.9
2	**100**	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	**100**	98.8	99.9
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	85.2	53.1	69.6	47.1	51.7	33.0	**91.1**	63.4	67.7
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	78.9	66.1	51.9	66.9	**93.9**	68.6	64.8
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6	99.6	99.6
6	62.3	74.8	46.2	76.8	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	**94.9**	70.7	57.9
7	86.6	79.9	65.3	91.3	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	88.4	**96.1**	89.4
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	94.9	**100**	86.3
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.1	98.2	98.2
10	81.0	77.0	65.1	**100**	80.9	**100**	79.3	94.3	94.3	94.3	91.0	92.6	94.0	81.7	**100**	99.5	78.1	88.9	94.9	94.3
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	**100**	91.3	74.54	**100**	**100**	94.1	91.3	92.6	91.3
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	72.1	57.0	63.0	61.2	59.8	64.9	**90.6**	70.5	37.3
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72.0	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	94.3	94.9	94.3
14	77.0	59.8	58.3	80.6	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	**92.2**	65.5	78.3
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	**85.5**	26.1	77.8	81.7	74.9	0	76.6	82.4	77.9
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	84.3	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	**88.4**	72.3	55.4
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	69.9	65.0	61.5	47.6	51.4	61.2	69.6	69.2	**94.9**	73.0	52.8
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	86.3	97.5	82.8
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	**100**	84.2	78.3	**100**	83.0	83.7	65.5	87.3	83.7
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	92.9	48.7	44.8	**97.2**	89.9	44.7
21	71.1	69.9	54.8	86.1	79.0	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	88.0	**93.5**	79.2	70.9
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**	**100**	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	98.4	**100**	**100**
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	66.1	39.8	64.0	**87.7**	49.6	80.5
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	84.13	78.80	67.80	**91.68**	84.20	77.95
Avg. Rank	14.23	10.19	17.90	7.85	12.77	8.73	11.79	8.90	10.23	9.83	9.15	8.46	11.27	12.31	7.33	10.42	13.81	**6.35**	6.85	11.63

Tabla B.5: Comparativa de propuestas de bajo-muestreo con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US	MIST-US	DBSCAN
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77.0	70.5	74.3	85.2	85.6	72.9	71.3	**87.9**	73	60.7
2	96.9	92.8	80.3	94.4	96.9	**97.5**	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	96.0	96.5	97.2
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.64	52	81.5	33.3	66.1	4.6	**94.1**	33.4	63.0
4	53.1	90.8	22.4	57.7	33.2	58.0	52.4	55.2	73	61.1	65.2	79.4	58.2	**94.1**	52.7	66.7	65.0	97.0	82.4	64.0
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	100	97.0	**100**	**100**	99.6
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	81.5	67.4	58.3	62.8	78.3	69.8	56.5	68.4	**84.9**	70.9	50.8
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	92.3	92.3	89.1	91.7	72.4	80.7	66.7	82.3	9	80.7	**96.1**	92.3
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	92.5	86.3	95.0	77.1	87.5	91.3	85.8	**93.5**	81.4	82.8	80.1
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	94.3	98.2	96.1
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79	83.1	**100**	99.7
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	**100**	**100**	91.3
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	77.1	64.3	**91.3**	62.5	0
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	83.1	**100**	99.4
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4.0	79.4	67.1	79.1	64.7	21.0
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	**91.9**	0	83.7	89	74.5	0	78.9	89.4	73.5
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	**88.2**	82.9	87.3	80.9	82.3	77.3	56.0
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	79.7	61.2	51.2	68.4	**96.5**	66.6	45.2
18	86.3	**96.6**	77.8	96.4	89	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	87.7	95.2	87.3
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	**100**	84.3	72.4	57.1	94.3	84.4
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74	75.5	74.1	57.0	75.7	91.4	32.1	45.9	**92.7**	88.7	33.3
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	90.8	85.3	70.4
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	**100**	**100**	**100**
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	95.0	96.6	98.3
24	0	47.8	0	0	0	0	0	44.0	40.0	42.3	44.6	56.5	0	68.6	69.7	44.7	52.5	56.6	44.4	**73.0**
Avg. Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	**88.04**	77.21	79.81	63.50	87.10	83.26	72.36
Avg. Rank	13.08	9.40	16.25	11.52	12.65	10.73	13.40	9.25	9.15	9.29	9.06	7.40	12.69	**6.5**	10.46	8.96	13.17	8.19	6.71	12.17

Tabla B.6: Comparativa de propuestas de bajo-muestreo con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	IG-US	MIST-US	DBSCAN
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	80.4	81.4	41.8	30.1	**93.0**	74.8	0
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	94.4	91.3	91.1	**99.0**	77.2	86.2	93.4	92.7	89.0
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	72.8	0	0	0	**68.3**	43.1	34.2
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	80.4	61.2	0	29.5	**84.0**	67.6	33.3
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	99.6	**100**	**100**
6	0	81.2	0	0	0	0	0	83.3	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	**87.9**	76.9	0
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	84.3	84.3	84.3	87.6	82.9	39.2	65.3	81.6	61.8	0	53.8	**96.1**	87.7
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	86.6	99.0	92.2	**100**	63.2
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	90.3	84.5	65.4
10	0	81.6	73.9	0	0	0	0	94.3	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	66.7	**94.9**	13.3
11	90.9	91.3	100	91.3	90.9	91.3	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	91.3	**100**	91.3
12	0	60.6	0	0	0	0	0	58.9	67.8	49.4	47.3	65.3	0	58.9	75.0	0	60.5	**84.5**	72.7	0
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	**100**	0	81.6	81.6	0	43.1	74.5	**100**	0
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	74.2	63.6	74.2
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	**86.6**	0	77.5	81.0	50.3	0	73.2	84.4	71.4
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	70.8	78.0	0
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	77.5	75.0	0	43.1	**98.3**	71.6	0
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	89.4	95.0	0
19	84.1	78.2	78.7	84.2	84.0	84.2	83.5	92.6	84.5	84.5	84.1	89.4	84.0	84.5	**100**	83.8	82.8	60.6	92.6	84.1
20	0	74.1	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	94.3	0	0	**98.1**	89.9	0
21	0	78.5	0	0	0	0	0	88.4	81.3	87.1	87.7	**91.4**	0	85.7	88.3	0	86.1	75.0	83.7	0
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	**100**	96.9	99.2	85.9	94.8	97.0
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	98.4	**100**	98.3
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	**58.3**	0	2.5	53.6	**58.3**	20.0
Avg. Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	81.55	42.75	46.58	81.54	**83.97**	42.60
Avg. Rank	16.23	7.46	14.19	14.46	16	14.81	15.19	5.21	5.67	5.98	7.54	6.48	15.27	7.71	5.21	13.60	13.02	7.94	**5.13**	12.92

Apéndice C

Resultados de clasificación para propuestas de tratamiento de desbalance de clases, ruido y/o traslape de clases

Este apéndice resume los resultados de clasificación de las propuestas DBIG-US y DBMIST-US, comparándolos con los resultados de algunos métodos de bajo-muestreo del estado del arte, usando los modelos 1NN, J48 y SVM. En las diferentes tablas que se muestran a continuación, los valores en negrita resaltan el mejor resultado para cada par formado por conjunto de datos y algoritmo de bajo-muestreo. Adicionalmente, se incluyen los resultados comparativos entre las propuestas de esta tesis.

Tabla C.1: Media geométrica obtenida por DBIG-US con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US
1	61.2	73.5	50.7	**83**	69.0	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	82.1	73.5	58.9	81.9
2	**100**	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	99.9
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	85.2	53.1	69.6	47.1	51.7	33.0	**88.4**
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	78.9	66.1	51.9	66.9	**90.5**
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6
6	62.3	74.8	46.2	76.8	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	**79.3**
7	86.6	79.9	65.3	91.3	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	**96.1**
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	94.9
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.1
10	81.0	77.0	65.1	**100**	80.9	**100**	79.3	94.3	94.3	94.3	91.0	92.6	94.0	81.7	**100**	99.5	78.1	94.3
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	**100**	91.3	74.54	**100**	**100**	94.1	91.3
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	72.1	57.0	63.0	61.2	59.8	64.9	**89.2**
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	94.3
14	77.0	59.8	58.3	80.6	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	**87.7**
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	85.5	26.1	77.8	81.7	74.9	0	**94.2**
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	84.3	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	**91.5**
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	69.9	65.0	61.5	47.6	51.4	61.2	69.6	69.2	**91.5**
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	88.3
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	**100**	84.2	78.3	**100**	83.0	83.7	65.5
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	92.9	48.7	**97.8**	44.8
21	71.1	69.9	54.8	86.1	79	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	**88.0**	85.7
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	98.4
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	66.1	39.8	64.0	**90.0**
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	84.13	78.80	67.80	**91.18**
Avg. Rank	12.77	9.10	16.10	6.90	11.5	7.77	10.60	7.98	9.06	8.77	8.10	7.63	10.08	10.98	6.63	9.23	12.42	**5.38**
1	86.2	89.9	66.1	91.1	90.4	95.1	87.8	94.5	90.4	92.5	89.5	89.7	91.7	85.7	92.3	89.2	88.6	**98.7**
2	72.0	72.5	58.0	78.9	75.3	83.4	75.5	74.5	83.5	77.0	77.3	86.5	83.3	68.6	66.3	77.8	87.2	**91.7**
3	60.5	69.0	46.0	73.6	68.2	81.9	68.3	74.0	71.5	81.0	74.5	73.9	80.6	70.4	67.0	65.5	86.6	**91.3**
4	60.2	74.0	41.7	74.9	63.0	85.3	65.6	69.4	75.5	74.9	70.8	76.4	79.4	65.2	76.4	64.8	78.9	**98.1**
5	52.5	70.5	35.9	72.3	62.7	82.4	62.0	79.8	72.5	73.5	71.6	84.6	74.0	63.8	86.3	64.4	85.5	**94.4**
6	89.1	90.4	59.5	98.7	94.8	96.8	95.1	89.9	89.4	93.4	91.6	91.5	98.1	86.1	98.7	90.2	87.6	**100**
7	80.4	78.0	50.3	86.8	81.6	88.0	81.9	78.5	80.9	76.9	82.6	89.0	87.6	74.0	80.4	80.8	80.4	**96.2**
8	70.0	74.5	48.6	83.1	75.9	87.5	75.6	75.8	75.4	76.9	76.3	86.8	85.9	67.5	70.0	70.8	81.6	**98.4**
9	62.6	79.0	43.4	79.2	68.4	88.6	66.6	71.5	77.5	75.4	74.8	78.7	82.7	67.3	76.1	63.9	89.4	**94.5**
10	62.5	73.9	41.1	80.4	71.9	87.4	75.3	77.5	74.9	77.0	73.4	82.1	78.7	58.8	74.5	67.3	85.0	**96.3**
11	93.5	94.9	60.3	98.4	96.7	98.5	95.6	97.0	93.0	92.4	95.1	95.3	**98.7**	91.8	90.9	94.7	71.9	97.2
12	78.3	77.5	51.9	86.7	83.5	91.8	83.7	83.0	83.5	77.9	83.0	94.7	87.5	75.8	80.4	84.1	79.4	**99.5**
13	74.1	81.5	46.3	86.0	81.8	89.4	83.1	82.8	83.0	87.0	79.9	90.4	86.1	73.4	81.0	80.4	45.9	**97.4**
14	68.1	78.0	48.9	83.4	74.8	89.3	73.4	82.5	78.5	79.0	78.2	83.6	83.9	70.9	74.2	74.2	49.4	**95.1**
15	64.0	77.0	45.0	78.3	74.7	89.8	75.8	74.6	79.0	77.5	76.8	83.3	82.1	69.4	79.4	72.1	86.2	**99.2**
Avg. Gmean	71.60	78.71	49.53	83.45	77.58	89.01	77.69	80.35	80.57	80.82	79.69	85.77	85.35	72.58	79.59	76.01	78.91	**96.53**
Avg. Rank	15.43	11.47	17.93	6.07	11	2.93	10.87	9.43	9.07	8.83	9.97	5.17	4.47	15.53	10.2	12.63	8.8	**1.2**

Tabla C.2: Media geométrica obtenida por DBIG-US con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77.0	70.5	74.3	85.2	**85.6**	72.9	71.3	76.1
2	**96.9**	92.8	80.3	94.4	**96.9**	97.5	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	96.7
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.64	52.0	**81.5**	33.3	66.1	4.6	77.5
4	53.1	90.8	22.4	57.7	33.2	58	52.4	55.2	73.0	61.1	65.2	79.4	58.2	**94.1**	52.7	66.7	65.0	84.3
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	**100**	97.0	**100**
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	**81.5**	67.4	58.3	62.8	78.3	69.8	56.5	68.4	63.3
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	**92.3**	**92.3**	89.1	91.7	72.4	80.7	66.7	82.3	9.0	91.9
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	**92.5**	86.3	95.0	77.1	87.5	91.3	85.8	93.5	81.5
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	94.3
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79.0	**100**
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	**100**
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	77.1	64.3	**94.0**
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	**100**
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4	79.4	67.1	81.6
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	91.9	0	83.7	89.0	74.5	0	**91.8**
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	88.2	82.9	87.3	80.9	**88.6**
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	79.7	61.2	51.2	68.4	**84.3**
18	86.3	**96.6**	77.8	96.4	89.0	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	83.0
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	**100**	84.3	72.4	57.1
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74.0	75.5	74.1	57.0	75.7	91.4	32.1	45.9	**92.7**
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	85.4
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	**100**
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	95.0
24	0	47.8	0	0	0	0	0	44.0	40.0	42.3	44.6	56.5	0	68.6	69.7	44.7	52.5	**70.0**
Avg. Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	**88.04**	77.21	79.81	63.50	87.05
Avg. Rank	11.83	8.48	14.71	10.46	11.48	9.67	12.10	8.33	8.27	8.31	8.19	6.71	11.46	**5.77**	9.33	8.15	11.88	5.88
1	97.8	92.3	77.9	97.8	97.8	**98.3**	97.2	95.5	93.9	88.6	92.3	91.9	96.6	88.6	93.2	97.8	87.3	95.0
2	66.7	75.9	74.0	71.4	70.2	87.5	68.5	75.3	83.2	79.7	83.5	**90.7**	87.3	82.8	73.0	65.5	89.4	87.9
3	26.4	78.0	31.3	74.8	53.4	84.8	68.2	74.7	80.8	82.0	80.9	79.8	83.2	75.3	69.2	49.8	83.7	**84.7**
4	0	77.5	50.1	46.7	0	76.6	0	74.5	81.4	85.7	79.1	75.9	84.3	75.8	78.4	0	81.2	**93.6**
5	0	76.5	0	32.6	24.4	72.6	14.1	77.8	80.7	73.0	79.9	74.4	80.1	73.4	92.1	17.3	83.5	**91.6**
6	63.3	85.6	11.6	65.1	66.7	66.9	66.6	83.4	77.5	78.2	81.5	85.9	92.7	79.5	**87.5**	68.7	82.1	86.7
7	26.4	74.6	59.2	42.9	45.1	71.5	28.2	68.9	79.1	79.6	81.7	74.7	84.0	78.8	82.5	50.2	52.7	**91.2**
8	0	71.8	0	59.1	47.9	64.3	10.0	77.7	82.1	75.5	75.8	79.3	84.2	76.2	71.6	14.1	78.4	**96.2**
9	31.4	77.1	46.1	59.4	43.1	78.1	22.2	68.5	82.3	76.9	76.2	64.0	**91**	73.9	76.0	28.0	83.9	83.1
10	0	68.2	27.2	22.3	0	67.7	0	71.4	80.1	62.6	75.3	72.3	82.2	62.8	76.4	0	80.4	**94.7**
11	80.4	93.9	39.0	65.9	65.1	71.6	80.1	91.2	89.0	90.3	92.2	92.1	**97.2**	91.7	93.9	65.2	35.1	**97.2**
12	19.9	83.4	60.1	79.7	72.5	87.5	60.7	79.2	83.2	82.8	82.3	90.0	87.6	78.1	84.6	38.0	76.0	**91.1**
13	28.2	80.6	52.0	82.5	51.8	87.5	51.5	86.7	86.9	85.6	82.0	89.8	89.0	75.0	86.1	24.3	30.0	**94.3**
14	0	74.7	55.6	57.8	24.4	87.3	14.1	83.7	81.8	79.8	80.2	77.0	**88.8**	75.7	79.2	19.8	51.3	87.6
15	0	78.4	0	63.9	32.9	88.4	34.2	77.3	82.1	78.2	82.0	76.7	90.8	78.6	81.4	0	76.8	**93.5**
Avg. Gmean	29.37	79.23	38.94	61.46	46.35	79.37	41.04	79.05	82.94	79.90	81.66	80.97	87.93	77.75	81.67	35.91	71.45	**91.23**
Avg. Rank	15.9	8.4	15.2	12.37	14.17	6.93	14.93	8.73	6.07	8.43	7.03	7.2	2.83	9.77	6.97	15.1	8.8	**2.17**

Tabla C.3: Media geométrica obtenida por DBIG-US con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	80.4	**81.4**	41.8	30.1	72.1
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	94.4	91.3	91.1	**99.0**	77.2	86.2	94.6
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	72.8	0	0	0	**66.1**
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	**80.4**	61.2	0	29.5	75.6
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	**100**
6	0	81.2	0	0	0	0	0	**83.3**	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	78.4
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	**84.3**	**84.3**	84.3	87.6	82.9	39.2	65.3	81.6	61.8	0	82.4
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	**100**	86.6	99.0	92.2
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	90.3
10	0	81.6	73.9	0	0	0	0	**94.3**	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	74.5
11	90.9	91.3	**100**	91.3	90.9	91.3	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	91.3
12	0	60.6	0	0	0	0	0	58.9	67.8	49.4	47.3	65.3	0	58.9	**75.0**	0	60.5	57.0
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	**100**	0	81.6	81.6	0	43.1	74.5
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	71.7
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	86.6	0	77.5	81.0	50.3	0	**91.5**
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	81.9
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	77.5	75.0	0	43.1	**85.9**
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	80.8
19	84.1	78.2	78.7	84.2	84.0	84.2	83.5	92.6	84.5	84.5	84.1	89.4	84.0	84.5	**100**	83.8	82.8	60.6
20	0	74.1	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	94.3	0	0	**98.1**
21	0	78.5	0	0	0	0	0	88.4	81.3	87.1	87.7	**91.4**	0	85.7	88.3	0	86.1	74.1
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	100	96.9	99.2	**100**
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	98.4
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	**58.3**	0	2.5	56.0
Avg. Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	81.55	42.75	46.58	**81.17**
Avg. Rank	14.56	6.56	12.71	12.98	14.35	13.29	13.67	**4.54**	4.94	5.35	6.63	5.77	13.71	6.85	4.65	12.23	11.67	6.54
1	0	51.3	0	0	0	0	0	54.5	47.0	56.8	0	64.0	0	49.5	**70.7**	0	0	62.4
2	0	41.8	0	0	0	0	0	54.0	41.9	50.4	0	57.2	0	38.5	62.3	0	**76.3**	60.1
3	0	47.6	0	0	0	0	0	50.3	48.5	52.6	0	59.6	0	52.5	60.4	0	**77.3**	61.2
4	0	47.5	0	0	0	0	0	48.0	**55.3**	42.0	0	54.9	0	50.5	68.4	0	22.6	48.5
5	0	52.5	0	0	0	0	0	53.2	43.0	54.8	0	45.7	0	38.4	78.4	0	67.6	18.6
6	0	54.5	0	0	0	0	0	47.1	52.0	48.4	0	41.2	0	48.0	**62.6**	0	29.5	53.7
7	0	55.8	27.2	0	0	0	0	50.5	58.9	58.6	0	40.5	0	40.0	70.5	0	2.2	60.9
8	0	51.2	0	0	0	0	0	58.6	51.4	56.8	0	64.6	0	41.1	63.6	0	**74.5**	53.4
9	0	56.9	0	0	0	0	0	54.7	45.7	47.5	0	52.8	0	48.0	69.5	0	**84.4**	15.6
10	0	57.5	0	0	0	0	0	57.4	57.4	41.4	0	51.3	0	54.7	63.3	0	**80.0**	58.1
11	0	64.2	0	0	0	0	0	47.4	39.4	57.6	0	65.4	0	49.7	52.2	0	0	**68.9**
12	0	57.5	49.6	0	0	0	0	59.8	43.0	45.0	0	51.6	0	50.4	55.0	0	32.4	**67.2**
13	0	61.4	51.3	0	0	0	0	62.0	54.1	56.7	0	**65.3**	0	54.2	61.7	0	0	53.7
14	0	48.8	33.6	0	0	0	0	48.7	46.3	56.5	0	56.2	0	49.8	**71.3**	0	0	57.1
15	0	49.4	17.0	0	0	0	0	55.9	55.3	49.6	0	42.7	0	52.1	**65.4**	0	77	19.6
Avg. Gmean	0.00	53.19	11.91	0.00	0.00	0.00	0.00	53.47	49.28	51.65	0.00	54.20	0.00	47.83	**65.02**	0.00	41.59	50.60
Avg. Rank	14.03	5.33	12.2	14.03	14.03	14.03	14.03	4.97	6.37	5.47	14.03	4.6	14.03	6.53	**2.13**	14.03	6.73	4.4

Tabla C.4: Media geométrica obtenida por DBMITS-US con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBMITS-US
1	61.2	73.5	50.7	**83**	69.0	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	82.1	73.5	58.9	75.4
2	**100**	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	98.4
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	85.2	53.1	69.6	47.1	51.7	33.0	**76.3**
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	**78.9**	66.1	51.9	66.9	76.9
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6
6	62.3	74.8	46.2	**76.8**	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	69.2
7	86.6	79.9	65.3	91.3	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	**92.1**
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	**100**
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.2
10	81.0	77.0	65.1	**100**	80.9	**100**	79.3	94.3	94.3	94.3	91.0	92.6	94.0	81.7	**100**	99.5	78.1	95.4
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	**100**	91.3	74.5	**100**	**100**	94.1	92.6
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	**72.1**	57.0	63.0	61.2	59.8	64.9	61.3
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72.0	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	92.7
14	77.0	59.8	58.3	**80.6**	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	74.5
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	85.5	26.1	77.8	81.7	74.9	0	**87.3**
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	**84.3**	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	79.8
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	**69.9**	65.0	61.5	47.6	51.4	61.2	69.6	69.2	66.1
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	97.1
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	100	84.2	78.3	**100**	83.0	83.7	87.3
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	**92.9**	48.7	44.8	78.9
21	71.1	69.9	54.8	86.1	79.0	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	**88.0**	81.1
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	**100**
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	66.1	39.8	64.0	**80.0**
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	84.13	78.80	67.80	**85.84**
Avg. Rank	12.83	9.10	16.13	6.83	11.56	7.71	10.65	7.90	9.13	8.79	8	7.57	10	11	6.54	9.21	12.46	**5.60**
1	86.2	89.9	66.1	91.1	90.4	95.1	87.8	94.5	90.4	92.5	89.5	89.7	91.7	85.7	92.3	89.2	88.6	**98.1**
2	72.0	72.5	58.0	78.9	75.3	83.4	75.5	74.5	83.5	77.0	77.3	86.5	83.3	68.6	66.3	77.8	87.2	**90.6**
3	60.5	69.0	46.0	73.6	68.2	81.9	68.3	74.0	71.5	81.0	74.5	73.9	80.6	70.4	67.0	65.5	86.6	**89.3**
4	60.2	74.0	41.7	74.9	63.0	85.3	65.6	69.4	75.5	74.9	70.8	76.4	79.4	65.2	76.4	64.8	78.9	**94.9**
5	52.5	70.5	35.9	72.3	62.7	82.4	62.0	79.8	72.5	73.5	71.6	84.6	74.0	63.8	86.3	64.4	85.5	**93.0**
6	89.1	90.4	59.5	98.7	94.8	96.8	95.1	89.9	89.4	93.4	91.6	91.5	98.1	86.1	98.7	90.2	87.6	**100**
7	80.4	78.0	50.3	86.8	81.6	88.0	81.9	78.5	80.9	76.9	82.6	89.0	87.6	74.0	80.4	80.8	80.4	**96.3**
8	70.0	74.5	48.6	83.1	75.9	87.5	75.6	75.8	75.4	76.9	76.3	86.8	85.9	67.5	70.0	70.8	81.6	**98.5**
9	62.6	79.0	43.4	79.2	68.4	88.6	66.6	71.5	77.5	75.4	74.8	78.7	82.7	67.3	76.1	63.9	89.4	**92.7**
10	62.5	73.9	41.1	80.4	71.9	87.4	75.3	77.5	74.9	77.0	73.4	82.1	78.7	58.8	74.5	67.3	85.0	**99.1**
11	93.5	94.9	60.3	98.4	96.7	98.5	95.6	97.0	93.0	92.4	95.1	95.3	98.7	91.8	90.9	94.7	71.9	**97.3**
12	78.3	77.5	51.9	86.7	83.5	91.8	83.7	83.0	83.5	77.9	83.0	**94.7**	87.5	75.8	80.4	84.1	79.4	94.0
13	74.1	81.5	46.3	86.0	81.8	89.4	83.1	82.8	83.0	87.0	79.9	90.4	86.1	73.4	81.0	80.4	45.9	**98.1**
14	68.1	78.0	48.9	83.4	74.8	89.3	73.4	82.5	78.5	79.0	78.2	83.6	83.9	70.9	74.2	74.2	49.4	**94.0**
15	64.0	77.0	45.0	78.3	74.7	89.8	75.8	74.6	79.0	77.5	76.8	83.3	82.1	69.4	79.4	72.1	86.2	**98.5**
Avg. Gmean	71.60	78.71	49.53	83.45	77.58	89.01	77.69	80.35	80.57	80.82	79.69	85.77	85.35	72.58	79.59	76.01	78.91	**95.63**
Avg. Rank	15.43	11.47	17.93	6.07	11	2.93	10.87	9.43	9.07	8.83	9.97	5.1	4.47	15.53	10.2	12.63	8.8	**1.27**

Tabla C.5: Media geométrica obtenida por DBMITS-US con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBMITS-US
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77.0	70.5	74.3	85.2	**85.6**	72.9	71.3	76.7
2	96.9	92.8	80.3	94.4	96.9	97.5	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	**97.5**
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.64	52.0	**81.5**	33.3	66.1	4.6	64.8
4	53.1	90.8	22.4	57.7	33.2	58.0	52.4	55.2	73.0	61.1	65.2	79.4	58.2	**94.1**	52.7	66.7	65.0	72.5
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	**100**	97.0	**100**
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	**81.5**	67.4	58.3	62.8	78.3	69.8	56.5	68.4	69.7
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	92.3	92.3	89.1	91.7	72.4	80.7	66.7	82.3	9.0	**92.3**
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	92.5	86.3	95.0	77.1	87.5	91.3	85.8	**93.5**	82.8
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	98.2
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79.0	92.3
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	**100**
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	**77.1**	64.3	55.0
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	95.8
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4.0	79.4	67.1	73.0
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	**91.9**	0	83.7	89.0	74.5	0	90.5
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	**88.2**	82.9	87.3	80.9	79.3
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	**79.7**	61.2	51.2	68.4	62.8
18	86.3	**96.6**	77.8	96.4	89	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	**96.6**
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	100	84.3	72.4	94.3
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74.0	75.5	74.1	57.0	75.7	**91.4**	32.1	45.9	76.0
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	84.6
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	99.8
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	96.6
24	0	47.8	0	0	0	0	0	44.0	40	42.3	44.6	56.5	0	68.6	**69.7**	44.7	52.5	59.6
Avg. Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	**88.04**	77.21	79.81	63.50	83.78
Avg. Rank	11.90	8.52	14.73	10.40	11.54	9.71	12.19	8.23	8.23	8.29	8.25	6.5	11.5	**5.69**	9.25	8	11.88	6.21
1	97.8	92.3	77.9	97.8	97.8	**98.3**	97.2	95.5	93.9	88.6	92.3	91.9	96.6	88.6	93.2	97.8	87.3	94.4
2	66.7	75.9	74.0	71.4	70.2	87.5	68.5	75.3	83.2	79.7	83.5	**90.7**	87.3	82.8	73.0	65.5	89.4	85.2
3	26.4	78.0	31.3	74.8	53.4	**84.8**	68.2	74.7	80.8	82.0	80.9	79.8	83.2	75.3	69.2	49.8	83.7	82.6
4	0	77.5	50.1	46.7	0	76.6	0	74.5	81.4	85.7	79.1	75.9	84.3	75.8	78.4	0	81.2	**90.9**
5	63.3	76.5	0	32.6	24.4	72.6	14.1	77.8	80.7	73.0	79.9	74.4	80.1	73.4	92.1	17.3	83.5	**86.7**
6	63.3	85.6	11.6	65.1	66.7	66.6	66.6	83.4	77.5	78.2	81.5	85.9	92.7	79.5	**87.5**	68.7	82.1	83.9
7	26.4	74.6	59.2	42.9	45.1	71.5	28.2	68.9	79.1	79.6	81.7	74.7	84.0	78.8	82.5	50.2	52.7	**97.4**
8	0	71.8	0	59.1	47.9	64.3	10.0	77.7	82.1	75.5	75.8	79.3	84.2	76.2	71.6	14.1	78.4	**94.7**
9	31.4	77.1	46.1	59.4	43.1	78.1	22.2	68.5	82.3	76.9	76.2	64.0	**91.0**	73.9	76.0	28.0	83.9	84.5
10	0	68.2	27.2	22.3	0	67.7	0	71.4	80.1	62.6	75.3	72.3	82.2	62.8	76.4	0	80.4	**97.4**
11	80.4	93.9	39.0	65.9	65.1	71.6	80.1	91.2	89.0	90.3	92.2	92.1	**97.2**	91.7	93.9	65.2	35.1	94.6
12	19.9	83.4	60.1	79.7	72.5	87.5	60.7	79.2	83.2	82.8	82.3	90.0	87.6	78.1	84.6	38.0	76.0	**91.1**
13	28.2	80.6	52.0	82.5	51.8	87.5	51.5	86.7	86.9	85.6	82.0	89.8	89.0	75.0	86.1	24.3	30.0	**94.4**
14	0	74.7	55.6	57.8	24.4	87.3	14.1	83.7	81.8	79.8	80.2	77.0	**88.8**	75.7	79.2	19.8	51.3	85.4
15	0	78.4	0	63.9	32.9	88.4	34.2	77.3	82.1	78.2	82.0	76.7	90.8	78.6	81.4	0	76.8	**92.8**
Avg. Gmean	29.37	79.23	38.94	61.46	46.35	79.37	41.04	79.05	82.94	79.90	81.66	80.97	87.93	77.75	81.67	35.91	71.45	**90.40**
Avg. Rank	15.9	8.33	15.2	12.37	14.17	6.8	14.93	8.73	6.07	8.43	7.03	7.13	2.67	9.77	6.97	15.1	8.8	**2.6**

Tabla C.6: Media geométrica obtenida por DBMITS-US con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBMITS-US
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	80.4	**81.4**	41.8	30.1	79.5
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	94.4	91.3	91.1	**99.0**	77.2	86.2	95.4
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	**72.8**	0	0	0	57.4
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	**80.4**	61.2	0	29.5	71.7
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	**100**
6	0	81.2	0	0	0	0	0	**83.3**	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	76.6
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	84.3	84.3	84.3	87.6	82.9	39.2	65.3	81.6	61.8	0	**92.0**
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	86.6	99.0	**100**
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	84.5
10	0	81.6	73.9	0	0	0	0	**94.3**	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	90.2
11	90.9	91.3	**100**	91.3	90.9	91.3	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	**100**
12	0	60.6	0	0	0	0	0	58.9	67.8	49.4	47.3	65.3	0	58.9	**75.0**	0	60.5	62.0
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	100	0	81.6	81.6	0	43.1	**91.2**
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	73.0
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	86.6	0	77.5	81.0	50.3	0	**88.7**
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	82.3
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	**77.5**	75.0	0	43.1	70.9
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	96.3
19	84.1	78.2	78.7	84.2	84.0	84.2	83.5	92.6	84.5	84.5	84.1	89.4	84.0	84.5	**100**	83.8	82.8	92.6
20	0	**74.1**	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	94.3	0	0	72.1
21	0	78.5	0	0	0	0	0	88.4	81.3	87.1	87.7	**91.4**	0	85.7	88.3	0	86.1	86.3
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	**100**	96.9	99.2	98.5
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	**100**
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	58.3	0	2.5	**62.1**
Avg. Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	81.55	42.75	46.58	**84.30**
Avg. Rank	14.60	6.90	12.85	13.04	14.40	13.36	13.73	4.85	5.25	5.63	6.96	6.02	13.75	7.15	4.71	12.31	11.83	**3.67**
1	0	51.3	0	0	0	0	0	54.5	47	56.8	0	64.0	0	49.5	**70.7**	0	0	67.5
2	0	41.8	0	0	0	0	0	54.0	41.9	50.4	0	57.2	0	38.5	**62.3**	0	76.3	59.4
3	0	47.6	0	0	0	0	0	50.3	48.5	52.6	0	59.6	0	52.5	60.4	0	77.3	**62.3**
4	0	47.5	0	0	0	0	0	48.0	55.3	42.0	0	54.9	0	50.5	**68.4**	0	22.6	51.1
5	0	52.5	0	0	0	0	0	53.2	43.0	54.8	0	45.7	0	38.4	**78.4**	0	67.6	50.3
6	0	54.5	0	0	0	0	0	47.1	52.0	48.4	0	41.2	0	48.0	62.6	0	29.5	**93.5**
7	0	55.8	27.2	0	0	0	0	50.5	58.9	58.6	0	40.5	0	40.0	**70.5**	0	2.2	54.0
8	0	51.2	0	0	0	0	0	58.6	51.4	56.8	0	64.6	0	41.1	63.6	0	**74.5**	53.3
9	0	56.9	0	0	0	0	0	54.7	45.7	47.5	0	52.8	0	48.0	69.5	0	**84.4**	0
10	0	57.5	0	0	0	0	0	57.4	57.4	41.4	0	51.3	0	54.7	63.3	0	**80.0**	61.2
11	0	64.2	0	0	0	0	0	47.4	39.4	57.6	0	65.4	0	49.7	52.2	0	0	**85.8**
12	0	**57.5**	49.6	0	0	0	0	59.8	43.0	45.0	0	51.6	0	50.4	55.0	0	32.4	50.6
13	0	61.4	51.3	0	0	0	0	62.0	54.1	56.7	0	**65.3**	0	54.2	61.7	0	0	61.9
14	0	48.8	33.6	0	0	0	0	48.7	46.3	56.5	0	56.2	0	49.8	**71.3**	0	0	45.2
15	0	49.4	17.0	0	0	0	0	55.9	55.3	49.6	0	42.7	0	52.1	**65.4**	0	77	27.1
Avg. Gmean	0.00	53.19	11.91	0.00	0.00	0.00	0.00	53.47	49.28	51.65	0.00	54.20	0.00	47.83	**65.02**	0.00	41.59	54.88
Avg. Rank	14	5.27	12.17	14	14	14	14	4.83	6.37	5.4	14	4.6	14	6.67	**2.2**	14	6.73	4.77

Tabla C.7: Comparativa de propuestas para el tratamiento de desbalance de clases, traslape de clases y/o ruido con el clasificador 1NN

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US	DBMIST-US
1	61.2	73.5	50.7	83.0	69.0	78.9	72.2	78.4	73.2	73.5	76.7	74.3	77.1	74.5	**82.1**	73.5	58.9	81.9	75.4
2	100	97.2	70.7	**100**	**100**	**100**	**100**	98.9	98.3	96.6	98.7	97.8	99.8	97.8	**100**	**100**	89.6	99.9	98.4
3	47.0	74.4	51.0	67.0	52.7	67.9	52.9	68.3	76.2	73.5	64.1	85.2	53.1	69.6	47.1	51.7	33.0	**88.4**	76.3
4	40.7	76.2	51.6	53.1	41.0	67.1	47.4	64.7	60.0	67.6	68.7	75.6	53.1	**78.9**	66.1	51.9	66.9	90.5	76.9
5	99.6	99.6	**100**	99.6	99.6	99.6	**100**	99.6	99.6	99.6	99.6	**100**	99.6	99.6	98.7	99.6	98.5	99.6	99.6
6	62.3	74.8	46.2	76.8	67.6	70.4	69.9	66.6	59.6	58.1	65.9	57.2	67.2	64.5	62.3	68.4	62.4	**79.3**	69.2
7	86.6	79.9	65.3	91.3	86.6	87.7	83.6	80.7	84.3	84.3	87.4	81.7	86.6	80.7	81.7	87.4	56.9	**96.1**	92.1
8	86.3	95.0	81.4	89.4	86.3	92.2	86.2	92.5	**100**	**100**	92.2	92.5	91.9	82.2	**100**	83.4	98.9	94.9	**100**
9	98.2	98.2	94.5	98.2	98.2	98.2	98.2	94.5	94.6	96.4	96.6	87.9	99.9	94.6	**100**	**100**	92.2	98.1	98.2
10	81.0	77.0	65.1	100	80.9	**100**	79.3	94.3	94.3	94.3	91.0	92.6	94.0	81.7	**100**	99.5	78.1	94.3	95.4
11	91.3	91.3	81.6	91.3	91.3	91.3	**100**	**100**	**100**	**100**	98.2	100	91.3	74.54	**100**	**100**	94.1	91.3	92.6
12	44.1	64.5	39.7	62.7	47.9	48.0	54.2	69.7	69.3	60.0	65.1	72.1	57.0	63.0	61.2	59.8	64.9	**89.2**	61.3
13	81.1	94.3	75.2	93.8	81.3	94.0	79.8	88.2	72.0	72.0	91.2	**100**	93.3	77.0	**100**	93.9	44.3	94.3	92.7
14	77.0	59.8	58.3	80.6	77.3	80.5	77.4	54.8	77.5	72.5	75.1	77.5	80.2	64.2	52.9	73.0	71.5	**87.7**	74.5
15	30.1	81.4	30.6	65.9	33.8	72.6	26.1	78.9	76.6	68.5	77.3	85.5	26.1	77.8	81.7	74.9	0	**94.2**	87.3
16	58.8	83.3	43.7	80.1	62.3	71.3	58.9	84.3	77.4	82.4	77.1	73.6	78.6	69.1	81.4	72.0	75.4	**91.5**	79.8
17	47.6	56.6	42.5	65.4	54.2	60.4	60.0	64.5	63.2	69.9	65.0	61.5	47.6	51.4	61.2	69.6	69.2	**91.5**	66.1
18	82.1	**100**	65.5	96.3	87.6	94.1	88.7	93.2	92.0	96.5	94.6	93.1	82.1	89.6	98.3	84.4	92.5	88.3	97.1
19	83.7	78.3	48.8	84.4	84.1	84.4	84.2	92.6	71.4	71.4	81.9	**100**	84.2	78.3	**100**	83.0	83.7	65.5	87.3
20	50.6	77.6	44.8	66.7	45.3	61.4	50.7	73.9	78.2	73.2	78.6	77.3	58.3	63.4	92.9	48.7	44.8	**97.8**	78.9
21	71.1	69.9	54.8	86.1	79	82.7	80.6	81.0	75.1	81.4	83.7	87.1	71.2	75.9	85.5	76.6	**88.0**	85.7	81.1
22	**100**	99.0	96.1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.9	99.0	**100**	**100**	**100**	**100**	99.3	**100**	**100**
23	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	0	98.4	**100**
24	19.9	53.1	14.8	20.0	20.0	20.0	20.0	61.2	45.3	63.5	65.1	43.3	34.5	64.5	66.1	39.8	64.0	**90**	80.0
Avg. Gmean	70.85	81.45	61.37	81.32	72.75	80.11	73.76	82.53	80.75	81.47	83.07	83.95	76.11	78.04	84.13	78.80	67.80	**91.18**	85.84
Avg. Rank	13.60	9.83	17.04	7.44	12.33	8.40	11.33	8.67	9.85	9.52	8.81	8.27	10.81	11.83	7.10	9.92	13.25	**5.71**	6.27
1	86.2	89.9	66.1	91.1	90.4	95.1	87.8	94.5	90.4	92.5	89.5	89.7	91.7	85.7	92.3	89.2	88.6	**98.7**	98.1
2	72.0	72.5	58.0	78.9	75.3	83.4	75.5	74.5	83.5	77.0	77.3	86.5	83.3	68.6	66.3	77.8	87.2	**91.7**	90.6
3	60.5	69.0	46.0	73.6	68.2	81.9	68.3	74.0	71.5	81.0	74.5	73.9	80.6	70.4	67.0	65.5	86.6	**91.3**	89.3
4	60.2	74.0	41.7	74.9	63.0	85.3	65.6	69.4	75.5	74.9	70.8	76.4	79.4	65.2	76.4	64.8	78.9	**98.1**	94.9
5	52.5	70.5	35.9	72.3	62.7	82.4	62.0	79.8	72.5	73.5	71.6	84.6	74.0	63.8	86.3	64.4	85.5	**94.4**	93
6	89.1	90.4	59.5	98.7	94.8	96.8	95.1	89.9	89.4	93.4	91.6	91.5	98.1	86.1	98.7	90.2	87.6	**100**	**100**
7	80.4	78.0	50.3	86.8	81.6	88.0	81.9	78.5	80.9	76.9	82.6	89	87.6	74.0	80.4	80.8	80.4	96.2	**98.3**
8	70.0	74.5	48.6	83.1	75.9	87.5	75.6	75.8	75.4	76.9	76.3	86.8	85.9	67.5	70.0	70.8	81.6	98.4	**98.5**
9	62.6	79.0	43.4	79.2	68.4	88.6	66.6	77.5	77.5	75.4	74.8	78.7	82.7	67.3	76.1	63.9	89.4	**94.5**	92.7
10	62.5	73.9	41.1	80.4	71.9	87.4	75.3	77.5	74.9	77.0	73.4	82.1	78.7	58.8	74.5	67.3	85	96.3	**99.1**
11	93.5	94.9	60.3	98.4	96.7	98.5	95.6	97.0	93.0	92.4	95.1	95.3	**98.7**	91.8	90.9	94.7	71.9	97.2	97.3
12	78.3	77.5	51.9	86.7	83.5	91.8	83.7	83.0	83.5	77.9	83.0	94.7	87.5	75.8	80.4	84.1	79.4	**99.5**	94.0
13	74.1	81.5	46.3	86.0	81.8	89.4	83.1	82.8	83.0	87.0	79.9	90.4	86.1	73.4	81.0	80.4	45.9	97.4	**98.1**
14	68.1	78.0	48.9	83.4	74.8	89.3	73.4	82.5	78.5	79.0	78.2	83.6	83.9	70.9	74.2	74.2	49.4	**95.1**	94.0
15	64.0	77.0	45.0	78.3	74.7	89.8	75.8	74.6	79.0	77.5	76.8	83.3	82.1	69.4	79.4	72.1	86.2	**99.2**	98.5
Avg. Gmean	71.60	78.71	49.53	83.45	77.58	89.01	77.69	80.35	80.57	80.82	79.69	85.77	85.35	72.58	79.59	76.01	78.91	**96.53**	95.63
Avg. Rank	16.43	12.47	18.93	7	12	3.87	11.87	10.43	10.07	9.83	10.97	6.1	5.4	16.53	11.2	13.63	9.8	**1.57**	1.9

Tabla C.8: Comparativa de propuestas para el tratamiento de desbalance de clases, traslape de clases y/o ruido con el clasificador J48

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US	DBMITS-US
1	65.1	70.6	60.2	70.6	65.8	74.5	64.3	73.2	70.5	68.6	77	70.5	74.3	85.2	**85.6**	72.9	71.3	76.1	76.7
2	96.9	92.8	80.3	94.4	96.9	97.5	97.4	95.0	96.1	95.0	95.9	96.1	94.5	95.5	94.7	95.8	77.6	96.7	**97.5**
3	52.3	60.0	43.8	66.7	53.1	53.3	58.3	46.7	57.6	45.6	63.5	57.64	52.0	**81.5**	33.3	66.1	4.6	77.5	64.8
4	53.1	**90.8**	22.4	57.7	33.2	58.0	52.4	55.2	73.0	61.1	65.2	79.4	58.2	94.1	52.7	66.7	65.0	84.3	72.5
5	**100**	**100**	99.6	99.9	**100**	**100**	**100**	**100**	**100**	**100**	99.9	**100**	99.9	99.6	97.5	**100**	97.0	**100**	**100**
6	54.3	64.5	57.2	57.2	65.2	62.6	54.2	71.6	58.3	**81.5**	67.4	58.3	62.8	78.3	69.8	56.5	68.4	63.3	69.7
7	82.2	73.0	73.0	82.1	82.2	86.6	79.3	88.4	92.3	**92.3**	89.1	91.7	72.4	80.7	66.7	82.3	9.0	91.9	**92.3**
8	82.9	87.5	92.3	80.1	82.9	80.1	86.0	76.5	87.5	92.5	86.3	**95.0**	77.1	87.5	91.3	85.8	93.5	81.5	82.8
9	96.2	**100**	94.5	99.8	99.8	96.1	97.4	98.2	96.4	94.5	97.5	92.0	97.1	94.6	95.7	97.5	93.8	94.3	98.2
10	99.4	88.2	95.3	99.7	93.7	99.7	97.1	**100**	94.3	94.3	92.5	92.6	99.7	**100**	81.7	99.5	79.0	**100**	92.3
11	90.9	**100**	81.6	91.3	90.9	91.3	**100**	91.3	**100**	**100**	99.4	**100**	90.9	**100**	66.7	91.3	96.3	**100**	**100**
12	0	59.6	0	0	0	0	0	66.6	60.6	44.7	53.1	60.6	25.8	74.5	61.2	77.1	64.3	**94**	55.0
13	98.8	**100**	85.3	99.7	99.5	99.8	98.5	94.3	94.3	94.3	93.9	**100**	**100**	**100**	81.7	99.3	49.5	**100**	95.8
14	22.4	61.2	69.2	22.3	0	31.6	22.4	74.8	71.4	76.5	72.1	74.3	31.6	**84.4**	4	79.4	67.1	81.6	73.0
15	15.2	85.9	32.7	62.4	0	56.8	0	84.8	86.0	79.5	84.2	**91.9**	0	83.7	89.0	74.5	0	91.8	90.5
16	53.9	78.3	0	60.7	57.4	65.4	53.9	83.3	80.3	84.1	78.2	82.9	68.4	**88.2**	82.9	87.3	80.9	88.6	79.3
17	48.2	53.8	47.9	44.6	44.6	44.6	44.6	59.9	54.2	69.7	67.3	54.3	40.8	79.7	61.2	51.2	68.4	**84.3**	62.8
18	86.3	**96.6**	77.8	96.4	89.0	93.9	88.9	87.4	94.1	89.8	94.2	90.9	88.9	92.0	96.5	88.4	90.2	83.0	**96.6**
19	84.4	71.4	70.4	84.5	84.4	84.5	83.5	70.0	78.2	78.2	74.5	**100**	84.4	85.7	**100**	84.3	72.4	57.1	94.3
20	34.7	75.0	0	43.3	34.6	39.3	32.1	78.8	75.0	74.0	75.5	74.1	57.0	75.7	91.4	32.1	45.9	**92.7**	76.0
21	73.4	72.9	70.2	77.2	73.3	73.5	67.3	82.6	71.4	88.6	81.9	85.7	71.4	**91.3**	82.2	84.4	80.4	85.4	84.6
22	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.0	97.5	**100**	96.8	**100**	99.8
23	98.3	95.0	0	98.3	98.3	98.3	96.6	98.3	98.3	95.0	97.7	98.3	98.3	93.1	**100**	98.3	0	95.0	96.6
24	0	47.8	0	0	0	0	0	44.0	40.0	42.3	44.6	56.5	0	68.6	**69.7**	44.7	52.5	70.0	59.6
Avg. Gmean	66.20	80.20	56.40	70.37	64.37	70.31	65.59	80.04	80.41	80.92	81.29	83.45	68.56	88.04	77.21	79.81	63.50	**87.05**	83.78
Avg. Rank	12.60	9.17	15.58	11.21	12.21	10.46	12.88	8.96	8.92	8.92	8.94	7.17	12.29	**6.21**	9.96	8.75	12.71	6.29	6.79
1	**97.8**	92.3	77.9	**97.8**	**97.8**	98.3	97.2	95.5	93.9	88.6	92.3	91.9	96.6	88.6	93.2	**97.8**	87.3	95.0	94.4
2	66.7	75.9	74.0	71.4	70.2	87.5	68.5	75.3	83.2	79.7	83.5	**90.7**	87.3	82.8	73.0	65.5	89.4	87.9	85.2
3	26.4	78.0	31.3	74.8	53.4	84.8	68.2	74.7	80.8	82.0	80.9	79.8	83.2	75.3	69.2	49.8	83.7	**84.7**	82.6
4	0	77.5	50.1	46.7	0	76.6	0	74.5	81.4	85.7	79.1	75.9	84.3	75.8	78.4	0	81.2	**93.6**	90.9
5	0	76.5	0	32.6	24.4	72.6	14.1	77.8	80.7	73.0	79.9	74.4	80.1	73.4	**92.1**	17.3	83.5	91.6	86.7
6	63.3	85.6	11.6	65.1	66.7	66.9	66.6	83.4	77.5	78.2	81.5	85.9	92.7	79.5	**87.5**	68.7	82.1	86.7	83.9
7	26.4	74.6	59.2	42.9	45.1	71.5	28.2	68.9	79.1	79.6	81.7	74.7	84.0	78.8	82.5	50.2	52.7	91.2	**97.4**
8	0	71.8	0	59.1	47.9	64.3	10.0	77.7	82.1	75.5	75.8	79.3	84.2	76.2	71.6	14.1	78.4	**96.2**	94.7
9	31.4	77.1	46.1	59.4	43.1	78.1	22.2	68.5	82.3	76.9	76.2	64.0	**91.0**	73.9	76.0	28.0	83.9	83.1	84.5
10	0	68.2	27.2	22.3	0	67.7	0	71.4	80.1	62.6	75.3	72.3	82.2	62.8	76.4	0	80.4	94.7	**97.4**
11	80.4	93.9	39.0	65.9	65.1	71.6	80.1	91.2	89.0	90.3	92.2	92.1	**97.2**	91.7	93.9	65.2	35.1	**97.2**	94.6
12	19.9	83.4	60.1	79.7	72.5	87.5	60.7	79.2	83.2	82.8	82.3	90.0	87.6	78.1	84.6	38.0	76.0	**91.1**	**91.1**
13	28.2	80.6	52.0	82.5	51.8	87.5	51.5	86.7	86.9	85.6	82.0	89.8	89.0	75.0	86.1	24.3	30.0	94.3	**94.4**
14	0	74.7	55.6	57.8	24.4	87.3	14.1	83.7	81.8	79.8	80.2	77.0	**88.8**	75.7	79.2	19.8	51.3	87.6	85.4
15	0	78.4	0	63.9	32.9	88.4	34.2	77.3	82.1	78.2	82.0	76.7	90.8	78.6	81.4	0	76.8	**93.5**	92.8
Avg. Gmean	29.37	79.23	38.94	61.46	46.35	79.37	41.04	79.05	82.94	79.90	81.66	80.97	87.93	77.75	81.67	35.91	71.45	**91.23**	90.40
Avg. Rank	16.83	9.33	16.2	13.3	15.1	7.67	15.87	9.67	7.07	9.43	8.03	8.07	3.37	10.77	7.83	16.03	9.67	**2.47**	3.3

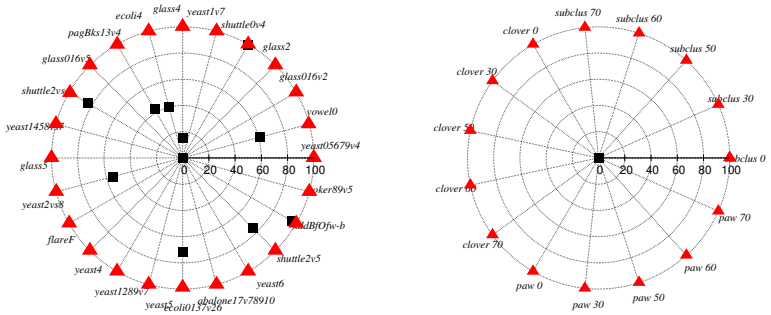
Tabla C.9: Comparativa de propuestas para el tratamiento de desbalance de clases, traslape de clases y/o ruido con el clasificador SVM

	Original	RUS	CNN	NCL	TL	ENN	OSS	EUS	EE	BC	RBT	EEKF	SBC	CBU	fCBUS	CBIS	COSS	DBIG-US	DBMTS-US
1	0	78.3	55.0	42.0	0	37.0	0	78.4	75.3	74.5	77.7	76.4	44.2	**80.4**	81.4	41.8	30.1	72.1	79.5
2	78.0	90.0	45.6	80.7	78.0	82.7	84.5	93.9	96.0	92.8	89.7	94.4	91.3	91.1	99.0	77.2	86.2	**94.6**	95.4
3	0	55.2	0	0	0	0	0	45.6	47.8	44.0	32.1	49.9	0	72.8	0	0	0	**66.1**	57.4
4	0	68.6	0	0	0	0	0	47.1	59.4	54.2	21.8	54.2	0	**80.4**	61.2	0	29.5	75.6	71.7
5	99.6	**100**	99.6	**100**	99.6	99.6	99.6	**100**	**100**	**100**	**100**	**100**	99.6	99.6	**100**	**100**	99.9	**100**	**100**
6	0	81.2	0	0	0	0	0	**83.3**	74.1	74.8	64.5	74.5	0	74.8	64.5	0	12.7	78.4	76.6
7	39.2	68.8	40.7	39.2	39.2	39.2	39.2	84.3	84.3	84.3	87.6	82.9	39.2	65.3	81.6	61.8	0	82.4	**92.0**
8	63.2	92.5	92.3	74.2	63.2	74.2	70.7	97.5	97.5	**100**	94.9	94.9	74.2	92.2	91.3	86.6	99	92.2	**100**
9	65.4	71.9	74.5	65.4	65.4	65.4	65.5	79.4	79.9	75.1	73.1	64.5	65.4	83.0	**100**	70.7	25.6	90.3	84.5
10	0	81.6	73.9	0	0	0	0	**94.3**	81.6	81.6	86.4	84.5	0	81.6	81.6	33.3	42.3	74.5	90.2
11	90.9	91.3	**100**	91.3	90.9	**100**	**100**	**100**	**100**	**100**	**100**	**100**	90.9	91.3	**100**	91.3	87.6	91.3	**100**
12	0	60.6	0	0	0	0	0	58.9	67.8	49.4	47.3	65.3	0	58.9	75.0	0	60.5	57.0	62.0
13	0	88.2	75.2	0	0	0	0	88.2	81.6	81.6	81.8	**100**	0	81.6	81.6	0	43.1	74.5	91.2
14	74.1	72.3	73.4	74.2	74.2	74.2	74.2	74.2	74.2	**77.5**	73.3	72.3	74.2	74.2	56.6	74.0	70.6	71.7	73.0
15	0	78.2	36.2	42.9	15.2	50.4	0	76.8	78.5	76.2	81.7	86.6	0	77.5	81.0	50.3	0	**91.5**	88.7
16	0	82.4	0	0	0	0	0	85.2	82.3	84.3	80.9	79.4	0	81.4	**88.2**	0	24.6	81.9	82.3
17	0	52.4	0	0	0	0	0	69.7	74.5	74.8	56.4	73.0	0	77.5	75.0	0	43.1	**85.9**	70.9
18	0	**98.9**	70.9	33.7	0	39.9	30.2	92.9	94.1	94.1	95.8	94.1	0	90.5	98.3	58.3	92.6	80.8	96.3
19	84.1	78.2	78.7	84.2	84.0	84.2	83.5	**92.6**	84.5	84.5	84.1	89.4	84.0	84.5	100	83.8	82.8	60.6	**92.6**
20	0	74.1	0	0	0	0	0	69.8	75.0	69.8	69.6	75.0	0	64.6	94.3	0	0	**98.1**	72.1
21	0	78.5	0	0	0	0	0	**88.4**	81.3	87.1	87.7	91.4	0	85.7	88.3	0	86.1	74.1	86.3
22	86.9	91.7	67.9	86.9	86.9	86.9	86.9	97.9	97.9	97.9	94.2	90.5	86.9	86.9	**100**	96.9	99.2	**100**	98.5
23	98.3	**100**	98.3	98.3	98.3	98.3	98.3	**100**	**100**	**100**	**100**	**100**	98.3	98.3	**100**	**100**	0	98.4	**100**
24	0	51.8	0	0	0	0	0	44.9	37.9	35.8	30.1	25.0	0	57.7	58.3	0	2.5	56.0	**62.1**
Avg. Gmean	32.49	78.61	45.09	38.04	33.12	38.47	34.69	80.97	80.23	78.93	75.45	79.93	35.34	80.49	81.55	42.75	46.58	81.17	**84.30**
Avg. Rank	15.52	7.35	13.65	13.92	15.31	14.25	14.60	5.25	5.64	6.10	7.48	6.5	14.67	7.65	5.13	13.15	12.63	7.19	**4.02**
1	0	51.3	0	0	0	0	0	54.5	47.0	56.8	0	64.0	0	49.5	**70.7**	0	0	62.4	67.5
2	0	41.8	0	0	0	0	0	54.0	41.9	50.4	0	57.2	0	38.5	62.3	0	76.3	**60.1**	59.4
3	0	47.6	0	0	0	0	0	50.3	48.5	52.6	0	59.6	0	52.5	60.4	0	77.3	61.2	**62.3**
4	0	47.5	0	0	0	0	0	48.0	55.3	42.0	0	54.9	0	50.5	**68.4**	0	22.6	48.5	51.1
5	0	52.5	0	0	0	0	0	53.2	43.0	54.8	0	45.7	0	38.4	78.4	0	**67.6**	18.6	50.3
6	0	54.5	0	0	0	0	0	47.1	52.0	48.4	0	41.2	0	48.0	62.6	0	29.5	53.7	**93.5**
7	0	55.8	27.2	0	0	0	0	50.5	58.9	58.6	0	40.5	0	40.0	**70.5**	0	2.2	60.9	54.0
8	0	51.2	0	0	0	0	0	58.6	51.4	56.8	0	64.6	0	41.1	63.6	0	**74.5**	53.4	53.3
9	0	56.9	0	0	0	0	0	54.7	45.7	47.5	0	52.8	0	48.0	69.5	0	**84.4**	15.6	0
10	0	57.5	0	0	0	0	0	57.4	57.4	41.4	0	51.3	0	54.7	63.3	0	**80.0**	58.1	61.2
11	0	64.2	0	0	0	0	0	47.4	39.4	57.6	0	65.4	0	49.7	52.2	0	0	68.9	**85.8**
12	0	57.5	49.6	0	0	0	0	**59.8**	43.0	45.0	0	51.6	0	50.4	55.0	0	32.4	67.2	50.6
13	0	61.4	51.3	0	0	0	0	62.0	54.1	56.7	0	**65.3**	0	54.2	61.7	0	0	53.7	61.9
14	0	48.8	33.6	0	0	0	0	48.7	46.3	56.5	0	56.2	0	49.8	**71.3**	0	0	57.1	45.2
15	0	49.4	17.0	0	0	0	0	55.9	55.3	49.6	0	42.7	0	52.1	65.4	0	**77.0**	19.6	27.1
Avg. Gmean	0.00	53.19	11.91	0.00	0.00	0.00	0.00	53.47	49.28	51.65	0.00	54.20	0.00	47.83	**65.02**	0.00	41.59	50.60	54.88
Avg. Rank	15	5.93	13.17	15	15	15	15	5.5	7.03	6.07	15	5.13	15	7.33	**2.4**	15	7.27	5	5.17

Apéndice D

Resultados de rendimiento por clase

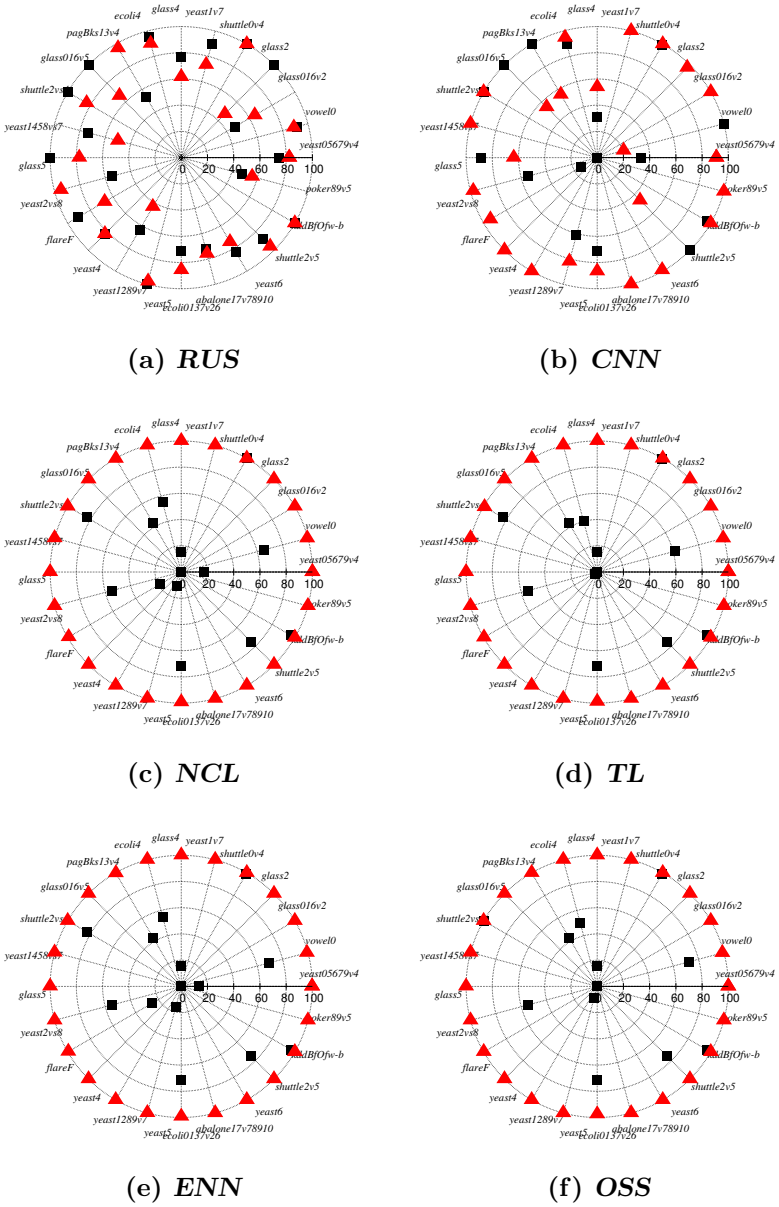
Este apéndice presenta los resultados obtenidos por cada método de bajo-muestreo del estado del arte, con respecto a las tasas de Especificidad (triángulos rojos) y Sensibilidad (cuadros negros) obtenida únicamente por el clasificador SVM, para conjuntos de datos reales (Figuras D.2-D.4) y sintéticos (Figuras D.5-D.7), se incluyen los conjuntos de datos sin preprocesar (Figuras D.1)



(a) ***Datos reales*** **(b)** ***Datos sintéticos***

Figura D.1: Precisión por clase para los resultados del clasificador SVM con conjuntos de datos sin preprocesar.

D.1. Conjuntos de datos reales

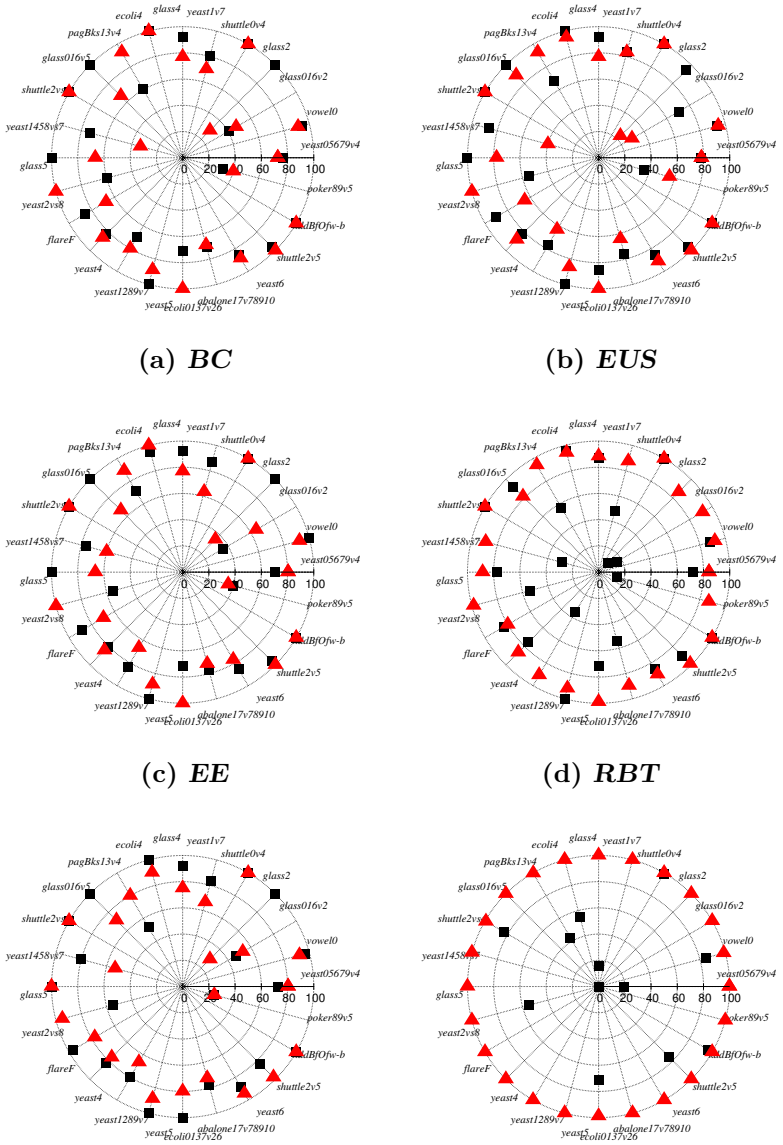


(a) ***RUS*** **(b)** ***CNN***

(c) ***NCL*** **(d)** ***TL***

(e) ***ENN*** **(f)** ***OSS***

Figura D.2: Precisión por clase para conjuntos de datos reales con métodos RUS, CNN, NCL, TL, ENN y OSS.

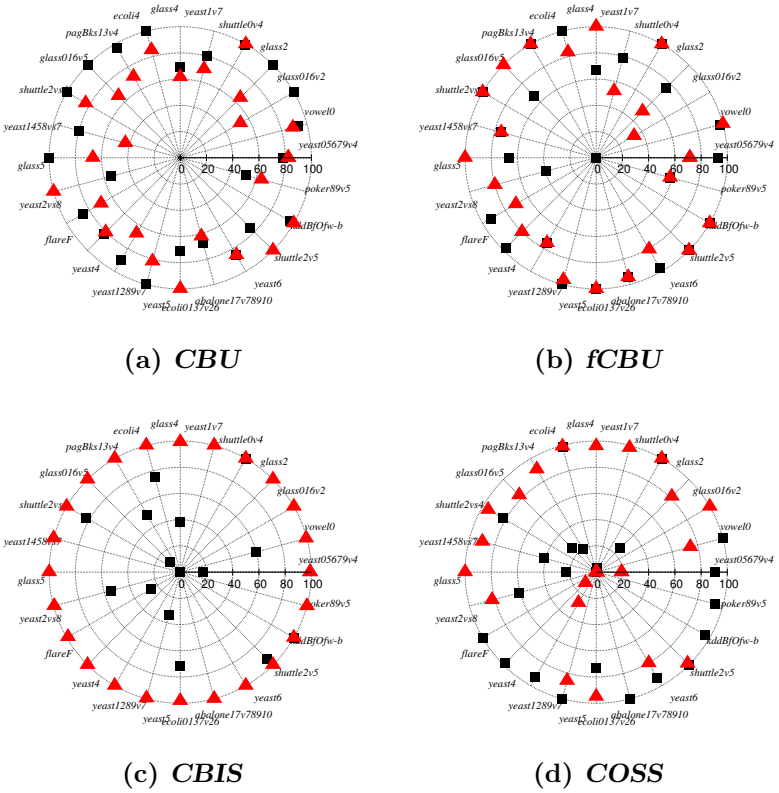


(a) ***BC*** **(b)** ***EUS***

(c) ***EE*** **(d)** ***RBT***

(e) ***EEKF*** **(f)** ***SBC***

Figura D.3: Precisión por clase para conjuntos de datos reales con métodos BC, EUS, EE, RBT, EEKF y SBC.

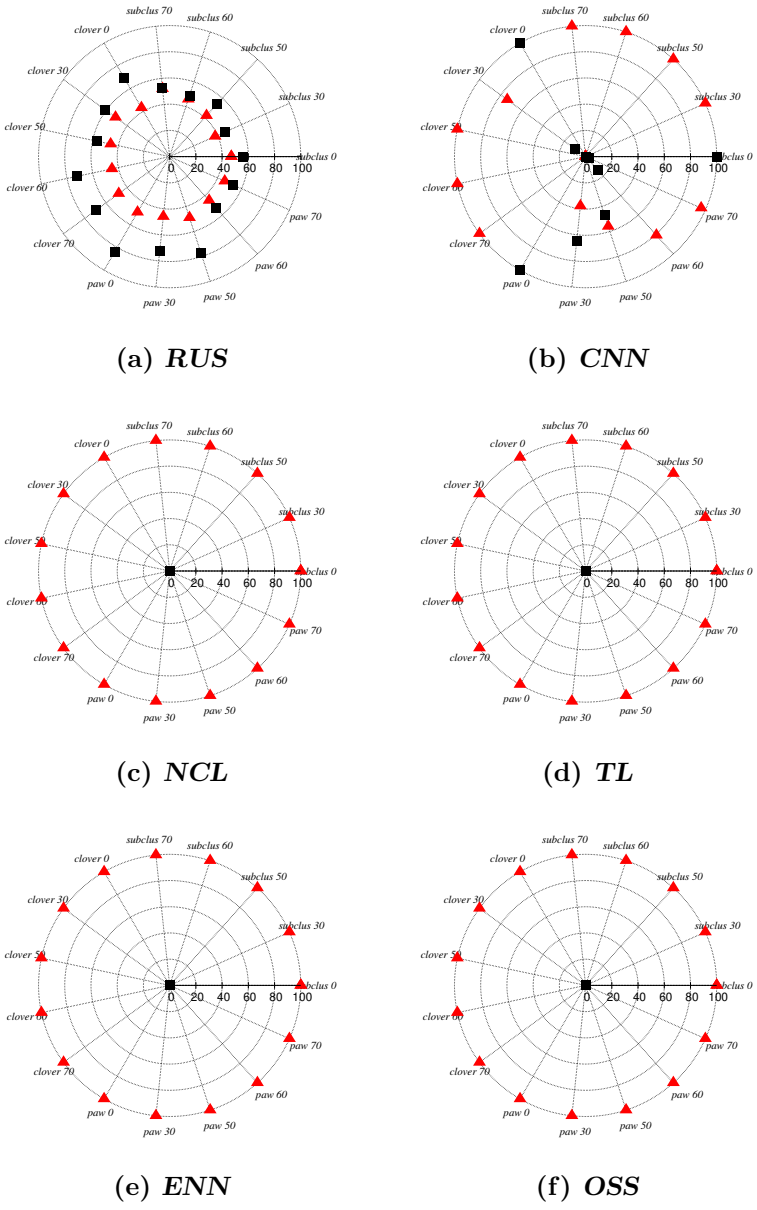


(a) *CBU* **(b) *fCBU***

(c) *CBIS* **(d) *COSS***

Figura D.4: Precisión por clase para conjuntos de datos reales con métodos CBU, fCBU, CBIS y COSS

D.2. Conjuntos de datos sintéticos

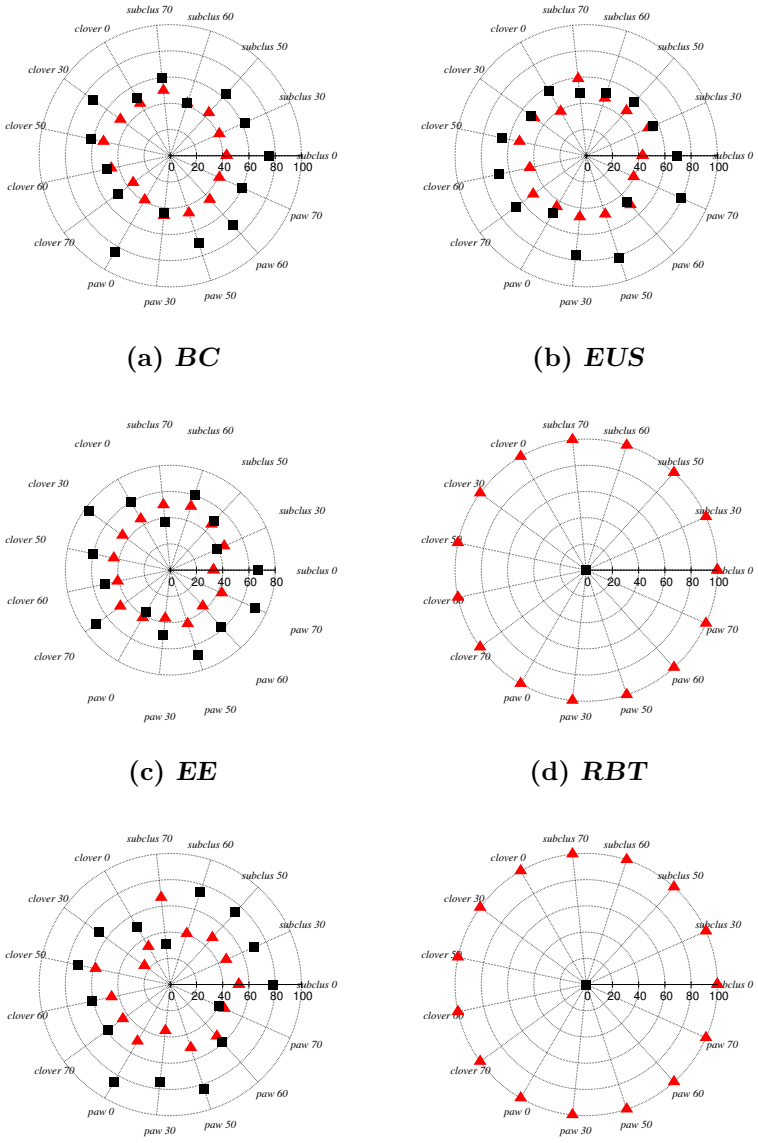


(a) ***RUS*** (b) ***CNN***

(c) ***NCL*** (d) ***TL***

(e) ***ENN*** (f) ***OSS***

Figura D.5: Precisión por clase para conjuntos de datos sintéticos con métodos RUS, CNN, NCL, TL, ENN y OSS.

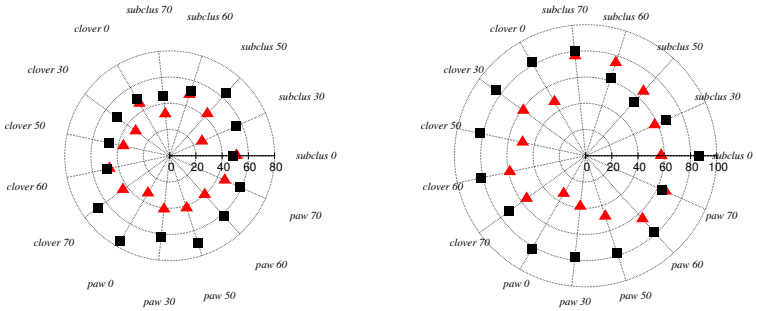


(a) *BC* **(b) *EUS***

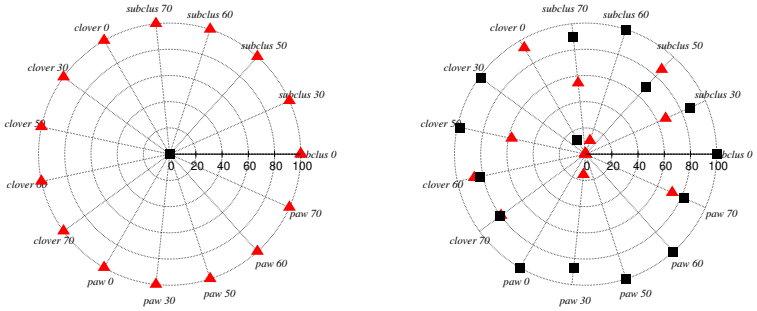
(c) *EE* **(d) *RBT***

(e) *EEKF* **(f) *SBC***

Figura D.6: Precisión por clase para conjuntos de datos sintéticos con métodos BC, EUS, EE, RBT, EEKF y SBC.



(a) *CBU* **(b) *fCBU***



(c) *CBIS* **(d) *COSS***

Figura D.7: Precisión por clase para conjuntos de datos sintéticos con métodos CBU, fCBU, CBIS y COSS.