



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TOLUCA

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

IMPLEMENTACIÓN DE TÉCNICAS PARA TRATAR EL DESBALANCE DE CLASES, EN LA CLASIFICACIÓN DE IMÁGENES DE VIOLENCIA FÍSICA REALIZADA CON REDES NEURONALES ARTIFICIALES

TESIS QUE PRESENTA

DANIEL CERVANTES AMBRIZ

NO. DE CONTROL: M23281647

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA INGENIERÍA

DIRECTOR: **DR. FEDERICO DEL RAZO LÓPEZ**

CODIRECTOR: **DR. JUAN ALBERTO ANTONIO VELÁZQUEZ**

METEPEC, ESTADO DE MÉXICO, MÉXICO

OCTUBRE 2025



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Toluca
División de Estudios de Posgrado e Investigación

Metepec, Edo. de México, 16/JUNIO/2025
DEPI-3200-326/2025

C. DANIEL CERVANTES AMBRIZ
CANDIDATO AL GRADO DE MAESTRO
EN CIENCIAS DE LA INGENIERÍA

De acuerdo con los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México y las disposiciones de este Instituto, habiendo cumplido con todas las indicaciones que la Comisión Revisora realizó con respecto a su trabajo de Tesis titulado **"IMPLEMENTACIÓN DE TÉCNICAS PARA TRATAR EL DESBALANCE DE CLASES, EN LA CLASIFICACIÓN DE IMÁGENES DE VIOLENCIA FÍSICA REALIZADA CON REDES NEURONALES ARTIFICIALES"**, la División de Estudios de Posgrado e Investigación de este Instituto, concede la autorización para proceder a la publicación del trabajo final a través del repositorio institucional.

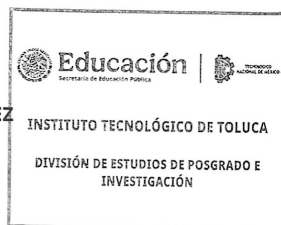
Sin otro particular por el momento, me despido de usted.

ATENTAMENTE

Excelencia en Educación Tecnológica®

"Educación, integridad y ciencia"

FRANCISCO JAVIER ILLESCAS MARTÍNEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



ccp. Archivo
FJM/mcdh



2025
Año de
La Mujer
Indígena

Av. Tecnológico S/N, Col. Agrícola Bellavista, Metepec, Edo. de México,
C.P. 52149, Tels. Dirección: 7222087205, Conmut.: 7222087200
e-mail: info@toluca.tecnm.mx tecnm.mx | toluca.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Toluca
División de Estudios de Posgrado e Investigación

Metepec, Edo. de México, 28/mayo/2025

DEPI-3200-287/2025

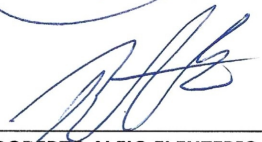
FRANCISCO JAVIER ILLESCAS MARTÍNEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
PRESENTE

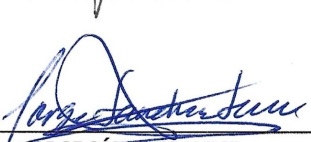
Por este medio comunicamos a usted que la Comisión Revisora designada para analizar la tesis denominada **"IMPLEMENTACIÓN DE TÉCNICAS PARA TRATAR EL DESBALANCE DE CLASES, EN LA CLASIFICACIÓN DE IMÁGENES DE VIOLENCIA FÍSICA REALIZADA CON REDES NEURONALES ARTIFICIALES"**, ha dictaminado que dicho trabajo reúne las características de contenido y calidad para proceder a la publicación del trabajo final a través del repositorio institucional. Lo anterior como parte de los requisitos para obtener el grado académico de Maestro en Ciencias de la Ingeniería que presenta el **C. DANIEL CERVANTES AMBRIZ** con número de control **M23281647** para sustentar el acto de Recepción Profesional.

ATENTAMENTE


FEDERICO DEL RAZO LÓPEZ
DIRECTOR DE TESIS


JUAN ALBERTO ANTONIO VELÁZQUEZ
CODIRECTOR DE TESIS


ROBERTO ALEJO ELEUTERIO
REVISOR DE TESIS


JORGE SÁNCHEZ JAIME
REVISOR DE TESIS

ccp. Archivo
mcdh



2025
Año de
La Mujer
Indígena

Av. Tecnológico S/N, Col. Agrícola Bellavista, Metepec, Edo. de México,
C.P. 52149, Tels. Dirección: 7222087205, Conmut.: 7222087200
e-mail: info@toluca.tecnm.mx tecnm.mx | toluca.tecnm.mx



Agradecimientos

Esta investigación fue financiada parcialmente por el **CONAHcyT** (México) con el No. CVU:
1288917.

Dedicatoria

A mi madre.

Resumen

En la presente tesis de maestría se abordó el problema del desbalance de clases en la clasificación de imágenes de violencia física mediante redes neuronales artificiales. Este estudio implementa y evalúa diversas técnicas para tratar el desbalance, incluyendo métodos tradicionales (sobremuestreo y submuestreo), generación de datos sintéticos con modelos generativos, y técnicas de aumento de datos. La metodología se basa en la introducción de desbalance controlado (1 %, 5 %, 10 %, 25 %, 50 % y 75 %) en cinco conjuntos de datos públicos, para evaluar la efectividad de cada técnica en diferentes escenarios. Para la clasificación se utilizó una arquitectura ResNet18, evaluando su rendimiento mediante métricas como precisión, sensibilidad y principalmente g-mean. Los resultados muestran que el método Random Over Sampling destaca en escenarios de desbalance extremo (1 %-25 %), mientras que One Sided Selection y Tomek Links presentan mejor desempeño en escenarios de 50 % y 75 % respectivamente. El análisis estadístico mediante el Test de Friedman confirma que la elección de la técnica de balanceo debe adaptarse al grado de desbalance del conjunto de datos para optimizar la capacidad del modelo para identificar correctamente los casos de violencia física.

Abstract

In this thesis for a master's degree, the problem of class imbalance in the classification of physical violence images using artificial neural networks was addressed. This study implements and evaluates various techniques to treat imbalance, including traditional methods (oversampling and undersampling), synthetic data generation with generative models, and data augmentation techniques. The methodology is based on introducing controlled imbalance (1%, 5%, 10%, 25%, 50%, and 75%) in five public datasets to assess the effectiveness of each technique in different scenarios. A ResNet18 architecture was used for classification, evaluating its performance through metrics such as precision, sensitivity, and primarily g-mean. The results show that the Random Over Sampling method stands out in extreme imbalance scenarios (1%-25%), while One Sided Selection and Tomek Links perform better in 50% and 75% scenarios, respectively. Statistical analysis using the Friedman Test confirms that the choice of balancing technique should be adapted to the degree of imbalance in the dataset to optimize the model's capacity to correctly identify cases of physical violence.

Índice general

1. Introducción	1
1.1. Planteamiento del Problema	1
1.1.1. Definición del problema	1
1.1.2. Delimitación del problema	2
1.2. Objetivos	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos	2
1.3. Hipótesis	3
1.4. Justificación	3
1.5. Organización del documento	3
2. Estado del Arte	5
2.1. Antecedentes	5
2.2. Trabajos Relacionados	7
3. Marco Teórico	13
3.1. Machine Learning	13
3.2. Redes Neuronales	15
3.2.1. Inspiración Biológica	16
3.2.2. Neurona artificial	16
3.3. <i>ResNet18</i>	17
3.4. <i>DCCGAN</i>	19
3.5. <i>DeepSMOTE</i>	20
3.6. Desbalance de Clases	22

4. Metodología	25
4.1. Conjuntos de datos	32
4.2. Redes Neuronales	34
4.2.1. <i>ResNet18</i>	34
4.2.2. <i>DCGAN</i>	35
4.2.3. <i>DeepSMOTE</i>	36
4.3. Optimización del modelo	36
4.4. Configuración del Entorno de Ejecución	37
5. Resultados y discusión	39
5.1. Resultados	39
5.2. Discusión	46
6. Conclusiones y recomendaciones	48
6.1. Conclusiones	48
6.2. Recomendaciones	48
Referencias	50
Apéndice A	53

Índice de figuras

3.1. Neurona biológica.	16
3.2. Neurona artificial.	17
3.3. Arquitectura de la red neuronal <i>ResNet18</i>	19
3.4. Diagrama de submuestreo.	23
3.5. Diagrama de sobremuestreo.	24
4.1. Etapas de la metodología.	25
4.2. Muestras de violencia generadas mediante métodos tradicionales de sobremuestreo (a),b),c)) y no violencia para el caso de los métodos tradicionales de submuestreo (d),e),f),g) y h)).	28
4.3. Muestras creadas por modelos generativos (<i>DeepSMOTE</i> y <i>DCCGAN</i>).	30
4.4. Muestras creadas por Aumento de Datos.	32
4.5. Muestras de los conjuntos de datos sin procesar.	33
5.1. Comparación del <i>g-mean</i> promedio de todos los métodos de balanceo en un escenario de 1 % de balance.	41
5.2. Comparación del <i>g-mean</i> promedio de todos los métodos de balanceo en un escenario de 5 % de balance.	42
5.3. Comparación del <i>g-mean</i> promedio de todos los métodos de balanceo en un escenario de 10 % de balance.	43
5.4. Comparación del <i>g-mean</i> promedio de todos los métodos de balanceo en un escenario de 25 % de balance.	44
5.5. Comparación del <i>g-mean</i> promedio de todos los métodos de balanceo en un escenario de 50 % de balance.	45
5.6. Comparación del <i>g-mean</i> promedio de todos los métodos de balanceo en un escenario de 75 % de balance.	46

Índice de tablas

2.1. Resumen de trabajos relacionados.	11
4.1. Distribución de imágenes por conjunto de datos.	34
4.2. Desglose de muestras por porcentaje de balance con división en violencia y no violencia.	34
4.3. Hiperparámetros utilizados para el ajuste fino de <i>ResNet18</i>	35
4.4. Hiperparámetros utilizados para la implementación de la red <i>DCCGAN</i>	36
4.5. Hiperparámetros utilizados para la implementación de <i>DeepSmote</i>	36
4.6. Configuración del entorno de ejecución	38
5.1. Resumen de Resultados de <i>Tests</i> de Friedman por escenario de balance.	47
1. Resultados del modelo en diferentes métodos de balanceo.	53
3. Rangos promedio de los algoritmos para el escenario de desbalance de 1 %. . .	58
4. Rangos promedio de los algoritmos para el escenario de desbalance de 5 %. . .	58
5. Rangos promedio de los algoritmos para el escenario de desbalance de 10 %. . .	58
6. Rangos promedio de los algoritmos para el escenario de desbalance de 25 %. . .	59
7. Rangos promedio de los algoritmos para el escenario de desbalance de 50 %. . .	59
8. Rangos promedio de los algoritmos para el escenario de desbalance de 75 %. . .	59

Capítulo 1

Introducción

La clasificación de datos desbalanceados representa un reto frecuente en el ámbito del aprendizaje automático, particularmente en aplicaciones críticas como la detección de violencia física a través de imágenes. Este problema surge cuando ciertas categorías del conjunto de datos tienen una representación significativamente menor en comparación con otras, lo que genera un sesgo en los modelos predictivos y compromete su capacidad de clasificación. En este trabajo se propone abordar este desafío implementando estrategias innovadoras que potencien el rendimiento de las redes neuronales, enfatizando en la identificación precisa de eventos violentos, una tarea de gran relevancia tecnológica y social.

1.1. Planteamiento del Problema

En esta sección se explora cómo el desbalance de clases afecta significativamente la eficacia de los algoritmos de aprendizaje automático, con un enfoque particular en las redes neuronales debido a su alta sensibilidad a este problema.

1.1.1. Definición del problema

El desbalance anteriormente mencionado, ocurre cuando ciertas categorías dentro de un conjunto de datos están menos representadas que otras. En situaciones como la violencia física retratada en imágenes, el desbalance es especialmente problemático ya que los casos reales de violencia son mucho menos frecuentes que los no violentos.

Supóngase que el conjunto de datos D está compuesto por dos subconjuntos, $D_{violencia}$ y $D_{no_violencia}$, los cuales comprenden respectivamente las instancias de las clases con violencia y no violencia. El desbalance de clases se puede expresar mediante la siguiente desigualdad: $\frac{|D_{violencia}|}{|D_{no_violencia}|} \neq 1$. Donde $|D_{violencia}|$ y $|D_{no_violencia}|$ denotan la cardinalidad (número de elementos) de los subconjuntos $D_{violencia}$ y $D_{no_violencia}$ respectivamente. El problema surge cuando el

resultado de esta división difiere significativamente de 1, es decir, cuando una clase está mucho menos representada que la otra en el conjunto de datos. Cuanto más se aleje este cociente de 1, mayor será el desbalance entre las clases.

Por otra parte, la predicción del modelo, que está representada por $\hat{y} = f_{\theta}(\mathbf{x})$, donde se toma una entrada $\mathbf{x}_i$ y produce una predicción $\hat{y}$. La calidad y la naturaleza de la entrada $\mathbf{x}$ son fundamentales para la precisión de la predicción $\hat{y}$. Sin embargo, si $\mathbf{x}$ contiene ruido, errores o simplemente no representa las características relevantes de la categoría a la que pertenece, las predicciones resultantes pueden ser engañosas o incorrectas.

1.1.2. Delimitación del problema

La presente investigación se centra en la evaluación de estrategias de balanceo de clases, para determinar su efectividad en mejorar la capacidad de los modelos para identificar correctamente incidentes de violencia física en distintos escenarios de balance. Así mismo, no se establece de antemano si una muestra específica del conjunto de datos de entrenamiento pertenece a la clase de *violencia* o *no violencia*, debido a que la clasificación previa ha sido realizada por expertos en la materia y los autores de los conjuntos de datos, cuya eficacia ya ha sido demostrada.

1.2. Objetivos

1.2.1. Objetivo general

Implementar y evaluar técnicas de balanceo para contrarrestar el desbalance de clases en la clasificación de imágenes de violencia física realizada con redes neuronales artificiales aplicando diferentes escenarios de balance.

1.2.2. Objetivos específicos

- Investigar métodos de balance tradicionales, modelos generativos y técnicas de aumento de datos.
- Obtener conjuntos de datos específicos relacionados con imágenes de violencia física.
- Realizar un preprocesamiento exhaustivo de los datos para su análisis.
- Introducir distintos escenarios de desbalance controlado en los conjuntos de datos con proporciones del 1 %, 5 %, 10 %, 25 %, 50 % y 75 %.
- Aplicar y comparar técnicas de balanceo sobre los conjuntos de datos desbalanceados.
- Implementar un clasificador binario utilizando redes neuronales artificiales.

- Llevar a cabo experimentos con los conjuntos de datos en escenarios variados de desbalance.
- Evaluar y analizar los resultados obtenidos para determinar la efectividad de las estrategias aplicadas.
- Identificar la técnica de balanceo de clases más efectiva para cada escenario de desbalance, evaluando los resultados obtenidos mediante un análisis estadístico no paramétrico.

1.3. Hipótesis

La implementación de técnicas de balanceo de clases, incluyendo métodos tradicionales como el sobremuestreo y el submuestreo, así como enfoques avanzados como la generación de datos sintéticos y el aumento de datos, incrementará significativamente la precisión y sensibilidad de los modelos de clasificación en la identificación de imágenes de violencia física. Se espera que estas técnicas sean eficaces en una amplia variedad de escenarios, desde condiciones de desbalance extremo hasta casos más equilibrados

1.4. Justificación

Esta investigación aborda un problema fundamental que es el impacto que el desequilibrio en los datos tiene sobre los sistemas que clasifican incidentes, especialmente en situaciones tan críticas como la violencia física. Mediante el uso de métodos de muestreo avanzados para equilibrar estos datos, no solo se busca perfeccionar la precisión de estos sistemas de clasificación, sino también destacar las serias implicaciones de los errores cometidos al clasificar tales incidentes. El objetivo final es lograr una mejora al identificar estos casos, asegurando que las víctimas sean reconocidas y apoyadas adecuadamente, y fomentando una respuesta más equitativa y efectiva por parte de la sociedad.

1.5. Organización del documento

En esta sección, se presenta la estructura de la tesis y se proporciona la descripción en 6 capítulos. En el capítulo 1 se establece una descripción del problema del desbalance de clases y sus implicaciones en sistemas automatizados para la clasificación de violencia. Se presentan los objetivos que orientan la investigación y se justifica la relevancia del estudio en el contexto actual. En el capítulo 2, se incluye una revisión exhaustiva del estado del arte, abarcando tanto enfoques convencionales como estrategias avanzadas para afrontar el desbalance de datos. Posteriormente el capítulo 3, profundiza en los fundamentos teóricos, ofreciendo un análisis detallado de conceptos clave como aprendizaje automático, redes neuronales y técnicas de balanceo de clases. Por su parte en el capítulo 4, se detalla la metodología utilizada. Se

describen minuciosamente los conjuntos de datos seleccionados y se analizan las técnicas implementadas para mitigar el desbalance. Estas técnicas se evalúan en diversos escenarios que varían desde condiciones de desbalance extremo hasta casos más equilibrados, permitiendo un análisis completo de su efectividad. Así mismo, el capítulo 5 presenta un análisis riguroso de los resultados experimentales, evaluando el impacto de las distintas técnicas de balanceo en el rendimiento de los modelos. Los hallazgos destacan las ventajas y limitaciones de cada enfoque en términos de métricas como precisión, sensibilidad y equilibrio entre clases, especialmente en escenarios con alto grado de desbalance. Finalmente, el capítulo 6 recopila las conclusiones del trabajo y propone posibles líneas de investigación futura, incluyendo la exploración de enfoques híbridos que combinen técnicas tradicionales y generativas para optimizar la clasificación en situaciones de desbalance extremo. Esta organización proporciona una estructura coherente que facilita la comprensión progresiva del estudio, desde sus fundamentos teóricos hasta los resultados empíricos y sus implicaciones.

Capítulo 2

Estado del Arte

Este capítulo, presenta una visión actual de la literatura, y se explora las investigaciones más recientes, así como conceptos clave, delineando así un entendimiento amplio que capta tanto los progresos como las aplicaciones modernas del aprendizaje automático en la gestión de bases de datos desbalanceadas.

2.1. Antecedentes

La Organización Mundial de la Salud ([OMS, 2002](#)), define la *violencia* como un acto deliberado, intencional y no accidental, que implica el uso de la fuerza física o de algún tipo de poder. Este puede dirigirse hacia uno mismo, hacia otras personas o incluso hacia comunidades completas. Las consecuencias de este uso de la fuerza son graves e incluyen lesiones físicas, pérdida de vidas, daños psicológicos y privaciones, abarcando desde daños emocionales hasta la pérdida de recursos esenciales o derechos fundamentales.

Entre las formas más significativas de violencia se encuentra la *violencia física*, caracterizada por el uso intencionado de la fuerza que puede provocar heridas, traumatismos o incluso la muerte. Este tipo de violencia puede ocurrir en diversos contextos, como en el entorno familiar, entre parejas (hombre-mujer) y en escenarios de conflicto como en peleas multitudinarias.

De acuerdo con la Organización Panamericana de la Salud ([Pan American Health Organization, 2024](#)), la violencia física en América tiene un impacto alarmante, cobrando la vida de aproximadamente 500 personas diariamente. Esto convierte a la región en una de las más afectadas a nivel global por homicidios, con una tasa que triplica el promedio mundial.

Por otra parte, el aprendizaje automático ([Mehta y cols., 2019](#)), por sus siglas en inglés (*ML, Machine Learning*), es un campo de estudio dentro de la inteligencia artificial (*IA*), dedicada al desarrollo de algoritmos que aprenden de manera autónoma a partir de los datos. Estos algoritmos están diseñados para que los sistemas puedan entre otras cosas, identificar objetos

y predecir comportamientos en su entorno, lo que les permite tomar decisiones basadas en la información adquirida. A diferencia de los métodos estadísticos tradicionales, que se centran más en la estimación, el ML se orienta hacia la predicción y clasificación, además es particularmente útil en situaciones complejas o con datos de alta dimensionalidad.

Así mismo, (Inglés Muñoz, 2023) considera que una de las aplicaciones más destacadas del aprendizaje supervisado, es el enfoque utilizado en la clasificación de imágenes. A la hora de clasificar, surge un desafío notable, conocido como *Desbalance de Clases*. Este fenómeno modifica la forma en como aprenden los clasificadores debido a una notable diferencia en la cantidad de información disponible para cada categoría o clase. Esta desigualdad hace que el modelo preste más atención a los tipos de casos que aparecen con más frecuencia, reduciendo su capacidad para reconocer y clasificar adecuadamente aquellos que son menos comunes.

Tal y como menciona (Morales y Pizarro, 2022), en el problema de la desigualdad de muestras se utilizan técnicas como el sobremuestreo de la clase minoritaria y el submuestreo de la clase mayoritaria.

El **sobremuestreo** consiste en aumentar la representación de las clases minoritarias dentro de un conjunto de datos. Esto se logra, por un lado, duplicando ejemplos ya existentes, y por otro, generando nuevos datos de manera sintética. Este enfoque busca equilibrar las clases menos representadas frente a las mayoritarias. Entre las técnicas más utilizadas para este propósito, se encuentran las siguientes:

- ADASYN (*Adaptive Synthetic Sampling Approach for Imbalanced Learning*): Este enfoque se centra en crear más muestras sintéticas justo donde los ejemplares de la clase minoritaria son más difíciles de identificar.
- SMOTE (*Synthetic Minority Over-sampling Technique*): Es una técnica avanzada que genera ejemplos sintéticos en lugar de duplicar ejemplos existentes.
- ROS (*Random Over Sampling*): Consiste en replicar de manera aleatoria las muestras de la clase minoritaria.

Por otro lado, el **submuestreo** busca reducir la predominancia de los grupos mayoritarios al eliminar parte de sus muestras, evitando así que dominen el análisis. Dentro de los mecanismos más importantes del submuestreo están:

- RUS (*Random Under Sampling*): elimina al azar instancias de la clase mayoritaria sin reemplazo.
- NM (*Near Miss*): Este método selecciona muestras de la clase mayoritaria a partir de su proximidad a las de la clase minoritaria.

- TL (*Tomek Links*): Elimina instancias de la clase mayoritaria que junto con una de la minoritaria forman un *Tomek link*, que no es más que un par de muestras muy cercanas entre sí pero que pertenecen a categorías distintas.
- ENM (*Edited Nearest Neighbours*): Elimina las muestras de la clase mayoritaria cuyos vecinos más cercanos no son en su mayoría de la misma categoría.

2.2. Trabajos Relacionados

Primeramente, el trabajo de investigación de ([Afzal y cols., 2019](#)) aborda la problemática del desequilibrio de clases en la clasificación automática de las etapas del *Alzheimer* mediante imágenes de resonancia magnética. El estudio implementa una metodología basada en el aumento de datos complementada con técnicas de transferencia de aprendizaje utilizando la arquitectura *AlexNet*. El análisis se fundamenta en el conjunto de datos *OASIS*, caracterizado por una distribución asimétrica entre cuatro categorías clínicas (*CDR*: 0, 0.5, 1, 2), con notable escasez de muestras en las etapas avanzadas. Para mitigar este desequilibrio, cada muestra de las clases minoritarias fue sometida a transformaciones geométricas y fotométricas, generando 28 muestras sintéticas adicionales por cada instancia original. Los resultados experimentales demuestran la efectividad de la metodología propuesta. El modelo basado en la vista axial alcanzó una precisión del 98.41 % con aumento de datos, en contraste con el 85.11 % obtenido sin este tratamiento. De manera similar, el enfoque multivista tridimensional logró una precisión del 95.11 % con aumento de datos, superando el 86.35 % obtenido mediante el entrenamiento convencional. Esta aproximación metodológica supera los enfoques previos que, aunque reportaron precisiones entre 79.08 % y 99.4 % en clasificaciones binarias, no abordaron eficazmente el problema multiclase con distribuciones severamente desequilibradas. La contribución principal radica en la integración efectiva de técnicas de aumento de datos con arquitecturas neuronales pre entrenadas para optimizar la clasificación diagnóstica en condiciones de limitada disponibilidad de casos.

Por otra parte, en el estudio realizado por ([Alejo, Rendón, y González-Bárceñas, 2020](#)), se evaluó el desempeño de tres modelos de redes neuronales artificiales: *Multilayer Perceptron* (MLP), *SVM* y *Radial Basis Function* (RBF). El objetivo principal fue clasificar a los estudiantes en categorías de alto y bajo rendimiento, además de identificar aquellos con riesgo de abandonar la carrera. La base de datos utilizada se obtuvo de estudios previos relacionados con la carrera de Ingeniería en Sistemas Computacionales (ISC) del Tecnológico de Estudios Superiores de Jocotitlán (TESJo). Este conjunto de datos presentaba un marcado desbalance en la distribución de clases. Para abordarlo, se aplicaron métodos como *ROS*, *RUS* y *SMOTE*, reconocidos por su efectividad en estos escenarios. *RUS* seleccionó aleatoriamente ejemplos de la clase mayoritaria para equilibrar las clases, mientras que *ROS* replicó ejemplos de la clase minoritaria. Por otro lado, *SMOTE* generó ejemplos sintéticos de la clase minoritaria interpolando entre muestras cercanas. Los resultados experimentales demostraron que los modelos lograron una precisión superior al 95 % en la clasificación de los estudiantes. En particular, el

modelo *RBF* destacó con un 97.4 % de precisión utilizando *ROS*, seguido por un 94.2 % con *SMOTE* y un 86.3 % con *RUS*. Estos hallazgos subrayan la importancia de seleccionar métodos de balanceo efectivos para optimizar el rendimiento de los clasificadores en contextos de datos desbalanceados.

Por su parte ([Rendón, Alejo, Castorena, Isidro-Ortega, y Granda-Gutiérrez, 2020](#)), su artículo se centra en el potencial y las capacidades de las técnicas de muestreo heurístico en el contexto de redes neuronales de aprendizaje profundo y *big data*. Así mismo, los autores hacen énfasis en explorar estrategias de limpieza dentro de este marco. Dicho trabajo se desarrolla sobre datasets desequilibrados y con varias clases a partir de imágenes de tele detección hiperespectral; todo esto con la finalidad de estudiar la efectividad de un enfoque híbrido en estos conjuntos de datos. Como resultado se tiene que *ROS* da mejores resultados que *SMOTE*, esto en escenarios de aprendizaje automático, por otra parte, *SMOTE* tradicionalmente funciona mejor que *ROS*, pero en el contexto de *big data* y aprendizaje profundo.

Así mismo, el trabajo desarrollado por ([Ojer Baztán, 2020](#)), se enfocó en implementar, analizar y comparar el rendimiento de *DERs-Boost* (*Differential Evolutionary Resampling Boosting*) frente a otros métodos de aprendizaje automático. Esta propuesta se diseñó para resolver problemas de clasificación en conjuntos de datos desbalanceados, combinando técnicas de re muestreo con algoritmos basados en cálculo diferencial evolutivo. En el estudio, se presenta una descripción detallada de la teoría subyacente y del funcionamiento del algoritmo propuesto. Además, se realizaron experimentos con distintos conjuntos de datos, evaluando los resultados en términos de precisión y tiempos de ejecución al comparar *DERs-Boost* con métodos más conocidos para clasificar datos desbalanceados. Por un lado, los resultados mostraron que el submuestreo implementado en *DERs-Boost* logró mejoras en comparación con técnicas tradicionales como *RUS*, *TL* y *ENN*. Por otro lado, se identificó que la combinación de *Evolución Diferencial* (DE) con *SMOTE* para generar datos sintéticos no aportó beneficios significativos al modelo. Finalmente, *DERs-Boost* obtuvo un *F1-score* de 51.98 %, destacándose como una alternativa prometedora para el manejo de conjuntos de datos desbalanceados.

En el trabajo de investigación desarrollado por ([Solís Fonseca, 2020](#)), se presenta un diseño experimental enfocado en evaluar el impacto del balanceo de clases en conjuntos de datos desiguales, utilizando como métrica principal la puntuación *F1-score* de un clasificador. Para evaluarlo, se aplicó un análisis estadístico con el objetivo de verificar si equilibrar las clases mejora el desempeño del modelo. En este caso, se empleó una máquina de soporte vectorial (SVM, por sus siglas en inglés) como clasificador. Los resultados obtenidos indican que el balanceo de clases mediante submuestreo produce un mejor rendimiento en comparación con un conjunto de datos sin balancear. De manera similar, se demostró que el sobremuestreo mejora aún más los resultados del modelo, logrando valores de *F1-Score* de 0.41 y 0.55, en contraste con el 0.35 obtenido sin aplicar técnicas de balanceo.

Con respecto al artículo realizado por ([Chávez-Bosquez, Vasquez, Hernandez, y Hernández-Ocaña, 2020](#)), el objetivo principal fue evaluar qué algoritmo de sobremuestreo mejora de

manera más efectiva el rendimiento de los clasificadores. Para ello, se analizaron tres técnicas aplicadas a la clase minoritaria: *ROS*, *SMOTE* y *ADASYN*. Los resultados mostraron que equilibrar los datos no solo mejora la precisión, sino también el rendimiento general de los modelos predictivos. El desempeño de estas técnicas se evaluó utilizando la curva *ROC* como métrica principal. Entre los métodos probados, *SMOTE* sobresalió con una precisión del 88.26 %, seguido de *ROS* con un 84.40 % y *ADASYN* con 78.82 %. Este análisis destaca la importancia de seleccionar técnicas de balanceo adecuadas para optimizar el rendimiento de los clasificadores en contextos de datos desbalanceados.

En el caso de la tesis de ([García Abad, 2021](#)), se realizó un trabajo comparativo de métodos de balanceo de clases en una aplicación real para la predicción de fuga de clientes. Concretamente se centra en las técnicas tales como; *RUS*, *ENN*, *TL*, *Near Miss V3 (NM3)*, *ROS*, *SMOTE*, *ADASYN* y *SMOTE + Tomek Links (SMOTE_TL)*. Dentro de las cuales aplicando una regresión logística se tiene que *RUS* es el mejor con 95 %. Mediante la clasificación por bosques aleatorios se tiene que *RUS* con 96 %, sigue siendo el mejor. Por último, en la aplicación de *gradient boost*, se tiene que el mejor método de balance es *TL* con 83 %. Cabe aclarar que se menciona que la elección entre una técnica u otra está en función al problema, por lo que no hay una técnica en general que se acople a todos los problemas.

Por su parte, en el artículo de ([Ljubic, Martinovic, y Volaric, 2022](#)), se evidencia que *DCGAN* emerge como una herramienta particularmente eficaz para la generación de imágenes sintéticas en diversos dominios de aplicación. En el contexto del conjunto de datos *CVPPP 2017 LCC*, *DCGAN* demostró capacidad para sintetizar imágenes de plantas con resolución de 128x128 píxeles. La calidad de estas representaciones sintéticas alcanzó un nivel de realismo tal que resultaron prácticamente indistinguibles de las muestras originales, constituyendo un recurso valioso para el entrenamiento de sistemas de visión computacional. En aplicaciones biomédicas, *DCGAN* generó imágenes sintéticas a partir de un corpus de 48,000 resonancias magnéticas cerebrales. Si bien se identificaron ciertas limitaciones relacionadas con el contraste entre sustancia gris y blanca, las imágenes generadas contribuyeron a optimizar significativamente los modelos de diagnóstico automático, según validación de especialistas médicos. Para el conjunto *ISIC 2017*, que comprende imágenes dermatológicas, *DCGAN* produjo representaciones de 192x256 píxeles que equilibraron eficazmente la distribución de clases, incrementando la precisión en tareas de segmentación de melanoma con una mejora del 7 % en el coeficiente de *Jaccard*. Adicionalmente, en el *BeesBook project dataset* (conjunto de datos de imágenes de alta resolución de abejas melíferas marcadas con códigos QR), la integración de *DCGAN* con modelado tridimensional facilitó la creación de imágenes sintéticas de abejas con variaciones controladas en iluminación y fondo, potenciando el entrenamiento de algoritmos de clasificación y detección. En síntesis, *DCGAN* constituye una aproximación metodológica robusta para la generación de imágenes sintéticas en múltiples dominios, optimizando tanto la calidad como la representatividad de los datos. No obstante, para aplicaciones que requieren mayor diversidad o resolución, arquitecturas alternativas como *WGAN*, *PGGAN* o *SAGAN* pueden ofrecer ventajas comparativas significativas.

En el trabajo de investigación realizada por ([Magnolia, 2023](#)), se propone una metodología integral basada en aprendizaje automático para la detección de enfermedades respiratorias agudas a partir del análisis de señales acústicas de la tos. El enfoque principal de esta tesis fue analizar el impacto de diferentes técnicas de segmentación y balanceo de clases en el diagnóstico de estas enfermedades mediante el procesamiento de grabaciones de audio. Aunque el trabajo se centra en casos relacionados con el COVID-19, la metodología es extensible a otras enfermedades respiratorias agudas que pueden ser identificadas utilizando grabaciones de tos. Este enfoque resulta crucial porque, en todas estas enfermedades, la segmentación precisa de los períodos de interés en las grabaciones es un paso esencial para lograr un análisis más detallado y efectivo. En este contexto, la investigación proporciona un aporte significativo al facilitar el diagnóstico y tratamiento de enfermedades respiratorias a través de herramientas más rápidas y precisas. Además, se evaluaron diversas estrategias de balanceo de clases, como el sobremuestreo, el submuestreo y el aumento de datos, para mejorar la efectividad de los modelos de clasificación. Entre los hallazgos más relevantes, se identificó que técnicas convencionales como *SMOTE* alcanzaron una especificidad del 78.01 %. Por otra parte, la aplicación de métodos de aumento de datos incrementó aún más esta métrica, demostrando su utilidad frente a conjuntos de datos desbalanceados. Por ejemplo, el aumento aleatorio elevó la especificidad al 94.67 %, mientras que el método *Pitch scaling* mejoró la sensibilidad al 55.24 %. Sin embargo, técnicas como *Time Stretch* y *Pitch scaling*, cuando se evaluaron tanto en términos de sensibilidad como de especificidad, lograron un mejor rendimiento comparado con métodos como *Noise* y *Polarity inversion*.

En el trabajo de investigación de ([Dablain, Krawczyk, y Chawla, 2023](#)), se propone un enfoque innovador para abordar el problema de los datos desbalanceados, combinando la técnica de sobremuestreo *SMOTE* con una arquitectura de redes neuronales profundas basada en el modelo *encoder-decoder*. Los resultados experimentales obtenidos, DeepSMOTE mostró mejoras significativas en comparación con el uso del conjunto de datos desbalanceado original. En el conjunto de datos *MNIST*, el *accuracy* aumentó de 87.4 % a 95.2 %, mientras que el *F1-Score* mejoró de 85.2 % a 92.3 %. Para *CIFAR-10*, el *accuracy* pasó de 68.5 % a 78.9 %, y el *F1-Score* subió de 65.3 % a 76.8 %. En el caso de *CelebA*, el *accuracy* mejoró de 74.2 % a 82.7 %, y el *F1-Score* aumentó de 72.1 % a 82.4 %. En *SVHN*, el *accuracy* creció de 85.3 % a 91.8 %, y el *F1-Score* pasó de 83.6 % a 90.5 %. Finalmente, en *Fashion-MNIST*, el *accuracy* mejoró de 79.8 % a 88.6 %, y el *F1-Score* aumentó de 77.5 % a 86.9 %. Estos resultados demuestran que DeepSMOTE no solo equilibra los conjuntos de datos desbalanceados, sino que también mejora significativamente el rendimiento de los modelos en términos de precisión y capacidad para manejar clases minoritarias.

Finalmente, en el artículo de ([Jalali Mojahed, Moattar, y Ghaffari, 2024](#)), se propone un enfoque novedoso para mejorar la clasificación en conjuntos de datos desbalanceados. Este método aborda los desafíos que surgen cuando las clases mayoritarias y minoritarias presentan tamaños significativamente desiguales, lo que afecta negativamente el desempeño de clasificadores como *k-nearest neighbors (KNN)* y *SVM*. El enfoque se basa en un aprendizaje de métricas que utiliza la distancia de *Bhattacharyya* para maximizar la separación entre las

distribuciones de clases. Las clases son modeladas como combinaciones de distribuciones gaussianas multi variantes, y su estructura interna es capturada mediante el algoritmo de *clustering DBSCAN*, junto con estimaciones de máxima a posteriori. Esta métrica se optimiza incrementando la distancia entre las clases distintas y reduciéndola entre los componentes de una misma clase, lo que mejora la capacidad de los clasificadores para diferenciarlas, especialmente en escenarios donde las clases minoritarias están sobre presentadas. Evaluado en 15 conjuntos de datos desbalanceados y comparado con otros métodos tradicionales utilizando *KNN*, el enfoque demostró ser significativamente más eficiente, incluso en casos de alto desbalance. En conclusión, el artículo introduce una nueva métrica basada en la distancia de *Bhattacharyya* que, en combinación con modelos de mezcla gaussiana, mejora notablemente la separación entre clases y el rendimiento en tareas de clasificación con clases desiguales.

La Tabla 2.1 es una síntesis de investigaciones relacionadas con el desbalance de clases en problemas de clasificación, donde se presentan los autores, títulos, modelos implementados, técnicas de balanceo, conjuntos de datos evaluados y resultados experimentales significativos de cada contribución. Esto se ha realizado con la finalidad de proporcionar un marco comparativo que fundamenta la aportación de la presente investigación y contextualiza sus aportaciones dentro del estado actual del conocimiento en este dominio específico del *Machine Learning*.

Tabla 2.1: Resumen de trabajos relacionados.

Autor(es)	Título	Modelo	Técnica	Conjunto de datos	Resultados
(Afzal y cols., 2019)	A Data Augmentation-Based Framework to Handle Class Imbalance Problem for Alzheimers Stage Detection	AlexNet	Data Augmentation	OASIS (resonancias magnéticas)	Accuracy aumentó de 85.11 % a 98.41 %
(Rendón y cols., 2020)	Data Sampling Methods to Deal With the Big Data Multi-Class Imbalance Problem	MLP y entornos big data	Métodos de muestreo y de edición	Imágenes hiperespectrales	ROS mejor en ML tradicional, SMOTE destacó en Big Data
(Solís Fonseca, 2020)	Efecto del balanceo de clases al evaluar el F1-Score para un clasificador de texto en análisis de sentimiento	SVM	Submuestreo y sobremuestreo	Datos de texto desbalanceados	F1-Score sin balanceo: 0.35, con ENN: 0.41, con SMOTE: 0.55
(Ojer Baztán, 2020)	Implementación y estudio del rendimiento del clasificador DERS-Boost, para problemas con conjuntos de datos no balanceados	Boosting	Combinación de Métodos de muestreo tradicionales	Datos de múltiples dominios	SMOTE-ENN-Boost con F1-Score de 51.98 %
(Chávez-Bosquez y cols., 2020)	Impacto de los algoritmos de sobremuestreo en la clasificación de subtipos principales del síndrome de Guillain-Barré	SVM	ROS, SMOTE, ADASYN	Datos clínicos de Guillain-Barré	SMOTE obtuvo una precisión de (88.26 %)
(Alejo y cols., 2020)	Procesamiento de bases de datos escolares por medio de redes neuronales artificiales	MLP, SVM y RBF	ROS, RUS y SMOTE	Datos académicos del rendimiento de estudiantes	MLP: 96.8 % de precisión con ROS, SVM: 93.5 % con SMOTE y RBF: 97.4 % con ROS
(García Abad, 2021)	Comparativa de técnicas de balanceo de datos. Aplicación a un caso real para la predicción de fuga de clientes	Regresión logística, Random Forest y Gradient Boosting	RUS, ENN, Tomek Links, Near Miss, ROS, SMOTE, ADASYN	Datos de clientes (Privado)	RUS fue el mejor método en Regresión Logística (95 %) y Random Forest (96 %)
(Ljubic y cols., 2022)	Augmenting data with generative adversarial networks: An overview	DCGAN	N/A	CVPPP 2017 LCC, ISIC 2017 y BeesBook Project Dataset	Mejoró segmentación de melanomas en ISIC 2017 (+7 % Jaccard)
(Magnolia, 2023)	Análisis acústico de las tos para detección de COVID con énfasis en el balanceo de clases	CNN	SMOTE, sobremuestreo y submuestreo	Grabaciones de tos de pacientes con COVID-19	SMOTE alcanzó 78.01 %
(Dablain y cols., 2023)	DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data	DeepSMOTE	N/A	MNIST, CIFAR-10, CelebA, SVHN, Fashion-MNIST	MNIST: Accuracy 95.2 %, CIFAR-10: 78.9 %, CelebA: 82.7 %, SVHN: 91.8 %, Fashion-MNIST: 88.6 %
(Jalali Mojahed y cols., 2024)	Supervised Density-Based Metric Learning Based on Bhattacharya Distance for Imbalanced Data Classification Problems	SVM y K-NN	Distancia de Bhattacharya, DBSCAN y Estimación de máxima a posteriori	Pima, Glass0, Ecoli1, Ecoli2, Ecoli3, Yeast 2 vs 4, Yeast 1 vs 7, Newthyroid1, Glass6, Winequality red 8 vs 6 7, Winequality white 3 9 vs 5, Heart, WDBC	KNN (3NN) Promedio global: F1-Score medio de 66.05 %, SVM Promedio global de 62.3 %.

Capítulo 3

Marco Teórico

La presente sección del capítulo presenta el marco teórico, proporcionando una estructura firme y bien organizada que fundamenta la investigación. Aquí se indagan y analizan los modelos y técnicas del aprendizaje automático que son fundamentales para comprender y mejorar el tratamiento de datos desbalanceados. El análisis comienza con definir la red neuronal, su aplicación práctica en la clasificación de imágenes con o sin violencia. Posteriormente se revisan métodos de balance de clases como el sobremuestreo, el submuestreo y técnicas combinadas, así como modelos generativos y aumento de datos. Cada uno con sus méritos y limitaciones en contextos específicos.

3.1. Machine Learning

En ([Raschka, Liu, y Mirjalili, 2022](#)), se menciona que el aprendizaje automático (*ML*, *Machine Learning*), se centra en desarrollar sistemas capaces de aprender por sí mismos en base a los datos. Este aprendizaje se traduce en la habilidad de identificar patrones y tendencias ocultas en extensos conjuntos de datos, algo simple para las personas, pero desafiante para las máquinas. Esta disciplina aprovecha conocimientos de ciencias de la computación, ingeniería y estadística para ofrecer soluciones prácticas en diversos ámbitos.

El enfoque de aprendizaje a partir de datos destaca por su flexibilidad, lo que lo hace especialmente valioso en escenarios donde las variables o condiciones no están completamente definidas desde el inicio. Esta metodología permite ajustes y adaptaciones conforme surgen nuevos datos, facilitando así la gestión de situaciones imprevistas.

Un modelo de aprendizaje automático, como ya se había mencionado anteriormente, es un sistema diseñado para identificar patrones, relaciones o características en conjuntos de datos. A continuación, se explica el proceso de cómo se desarrolla un modelo de aprendizaje automático:

1. **Elección del Tipo de Modelo:** Se selecciona el modelo ($f_\theta(\mathbf{x})$) en función al problema que se necesita abordar, como clasificación y regresión.
2. **Recopilación y Preparación de Datos:** Este paso involucra la recolección de datos pertinentes y su preparación para el entrenamiento. Así mismo en este paso se deben limpiar los datos para eliminar errores y finalmente se normalizan para obtener mejores resultados, según sea el caso. En términos matemáticos, un conjunto de datos se define tal y como se puede observar en la Ecuación 3.1.

$$D = \left\{ \begin{array}{l} ((x_{11}, x_{12}, \dots, x_{1n}), y_1), \\ ((x_{21}, x_{22}, \dots, x_{2n}), y_2), \\ \dots, \\ ((x_{m1}, x_{m2}, \dots, x_{mn}), y_m) \end{array} \right\} \quad (\text{Ecuación 3.1})$$

3. **Entrenamiento del Modelo:** Durante el entrenamiento, el modelo ajusta sus parámetros θ para minimizar errores en las predicciones, usando un conjunto de datos designado como conjunto de entrenamiento. Supóngase el modelo $f_\theta(\mathbf{x})$. En la Ecuación 3.2, se puede observar formalmente el proceso de entrenamiento. Así mismo en la Ecuación 3.3 se ha sustituido $\hat{y}$, para simplificar la expresión.

$$\min_{\theta} \sum_{i=1}^N \ell(f_\theta(x_i), y_i) \quad (\text{Ecuación 3.2})$$

sustituyendo $\hat{y}$:

$$\min_{\theta} \sum_{i=1}^N \ell(\hat{y}_i, y_i) \quad (\text{Ecuación 3.3})$$

donde:

- $\hat{y}$: es la salida predicha por el modelo.
- θ : es el conjunto de parámetros θ del modelo.
- N : es el número total de muestras o puntos de datos en el conjunto de datos.
- $\ell(f_\theta(x_i), y_i)$: Esta es la función de pérdida evaluada para cada par de entrada y salida, que mide qué tan buena es la predicción $f_\theta(x_i)$ comparada con la etiqueta verdadera y_i :
 - $f_\theta(x_i)$: Es el modelo o función de predicción parametrizado por θ , evaluado en la entrada x_i . Esta función genera la predicción del modelo para la entrada dada.

- y_i : Es la etiqueta o el valor verdadero correspondiente a la entrada x_i .
- ℓ : Es la función de pérdida.

El objetivo de esta expresión es para optimizar los parámetros del modelo para minimizar la suma total de la función de pérdida, sobre todos los ejemplos del conjunto de datos. Esto implica hacer que las predicciones del modelo sean lo más cercanas posible a los valores verdaderos, ajustando los parámetros del modelo de manera adecuada.

Una estrategia ampliamente utilizada para resolver este tipo de problemas de optimización es el método del gradiente descendente. Como se puede ver formalmente en la Ecuación 3.4, este enfoque consiste en actualizar iterativamente los parámetros del modelo con base en la dirección de mayor descenso de la función de pérdida, calculada en función de θ . De este modo, el modelo ajusta sus parámetros de manera eficiente para minimizar el error en las predicciones.

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{1}{N} \sum_{i=1}^N \nabla_{\theta_t} \ell(f_{\theta_t}(x_i), y_i) \quad (\text{Ecuación 3.4})$$

donde:

- θ_t son los parámetros en la iteración t .
- η es la tasa de aprendizaje, que controla el tamaño del paso en la actualización.
- ∇_{θ} denota el gradiente respecto a los parámetros θ de la función de pérdida $\ell(f_{\theta}(x_i), y_i)$, donde $f_{\theta}(x_i)$ es la predicción del modelo ($\hat{y}$) para la entrada x_i , y y_i es el valor objetivo correspondiente.

4. **Evaluación del Modelo:** Una vez entrenado, el rendimiento del modelo se evalúa con un conjunto de datos de prueba. Esto determina si el modelo puede generalizar lo aprendido a nuevas situaciones y no solo memorizar los datos con los que fue entrenado.

3.2. Redes Neuronales

Las redes neuronales artificiales son modelos matemáticos que imitan la forma en que el cerebro humano procesa la información. Tienen la increíble capacidad de aprender a partir de datos, lo que les permite resolver una amplia gama de problemas en diversas áreas. Su adaptabilidad y capacidad de aprendizaje las convierten en una herramienta poderosa en el campo de la inteligencia artificial.

3.2.1. Inspiración Biológica

Según (Qamar y Zardari, 2023), las redes neuronales artificiales (RNA), son estructuras matemáticas diseñadas para simular el comportamiento del cerebro humano en el análisis y procesamiento de datos. Estas redes se destacan por su capacidad de identificar patrones y aprender de manera autónoma a partir de los datos proporcionados, lo que les permite ofrecer soluciones a problemas complejos en múltiples disciplinas. Su flexibilidad y eficiencia las han posicionado como un recurso clave en el desarrollo de tecnologías dentro del campo de la inteligencia artificial.

Las RNA emulan esta estructura mediante modelos matemáticos que simulan el comportamiento de las neuronas y sus conexiones. Es fundamental comprender que las RNA son modelos simplificados que capturan solo algunos aspectos del funcionamiento neuronal. No obstante, esta inspiración en la biología ha sido clave para el desarrollo de las RNA y su aplicación en diversas áreas del *ML*. La Figura 3.1 muestra una neurona biológica con sus componentes fundamentales: a) núcleo, b) cuerpo, c) axón y d) dendritas.

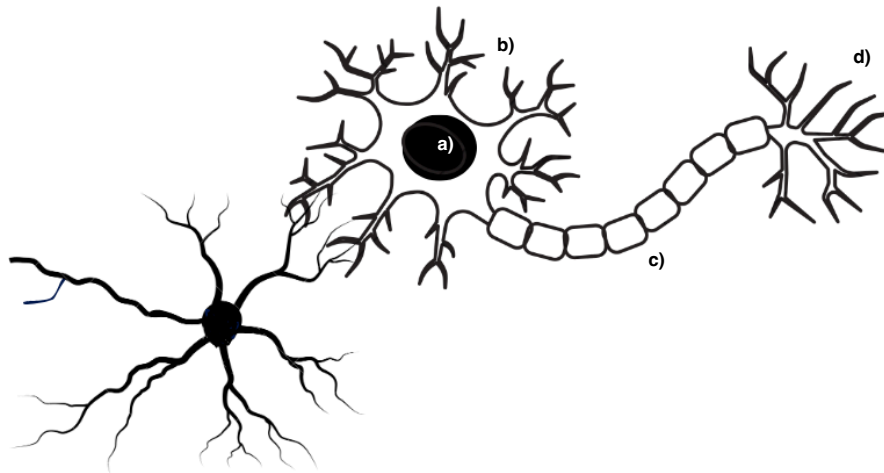


Figura 3.1: Neurona biológica.

3.2.2. Neurona artificial

El modelo de McCulloch-Pitts, que emula el funcionamiento de las neuronas biológicas, igualmente reciben señales de entrada, que pueden ser excitatorias o inhibitorias, y las integran. Si la suma total de estas señales supera un cierto límite, la neurona se activa y produce una respuesta.

Así mismo y tal como se representa en la Figura 3.2, recibe diversas señales de entrada, representadas como $x_1, x_2, \dots, x_n$, para cada una de estas señales tiene un peso asociado, w_1 ,

$w_2, \dots, w_n$, que es la importancia que la neurona le da a cada mensaje. A su vez, es como si algunos mensajes fueran más relevantes que otros. Como ya se había mencionado y al igual con su antagonista biológico, existe un umbral que es un nivel mínimo de activación que la neurona necesita para activarse y enviar una señal de salida. Este umbral se suma a las señales de entrada ponderadas. Posteriormente, todas las señales y pesos se combinan en un sumador, que es el lugar donde se juntan todos los mensajes y se decide si la neurona se activará o no. Finalmente, esta salida pasa por una función de activación, que ajusta la intensidad de la respuesta.

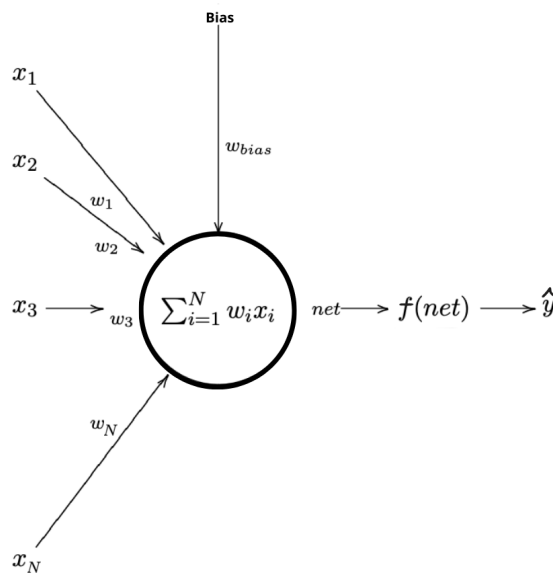


Figura 3.2: Neurona artificial.

3.3. ResNet18

Es una arquitectura de red neuronal convolucional mostrada en el artículo de ([K. He, Zhang, Ren, y Sun, 2016](#)), como parte de la familia de redes *Residual Networks* (*ResNet*). Esta red tiene una buena capacidad de evitar el desvanecimiento de gradiente, común en arquitecturas profundas. La arquitectura se compone por 18 capas de convolución.

La operación de convolución es fundamental en el procesamiento de imágenes y en la arquitectura de redes neuronales convolucionales, ya que permite la extracción eficiente de características y patrones relevantes indispensables para clasificar imágenes. En la Ecuación 3.5 se presenta formalmente la expresión matemática que define dicha operación.

$$f_{\theta}(x_i) = \mathbf{x} \otimes \mathbf{w} \quad (\text{Ecuación 3.5})$$

donde:

- $f_{\theta}(x_i)$ corresponde a la función de activación resultante tras aplicar la operación de convolución sobre el vector de entrada x_i .
- x representa el vector de entrada que contiene los datos de la imagen a procesar.
 - w denota el filtro o *kernel* convolucional cuyos parámetros son ajustados durante el proceso de entrenamiento.
 - $\otimes$ simboliza específicamente la operación matemática de convolución.
 - θ hace referencia al conjunto de parámetros aprendibles del modelo.

Retomando el tema central de esta sección, la arquitectura de *ResNet18*, como se ilustra en la Figura 3.3, es una red neuronal convolucional que pertenece a la familia de redes Residual Networks (ResNet). Esta arquitectura consta de 18 capas entrenables organizadas en bloques residuales, los cuales son fundamentales para superar problemas comunes en redes profundas, como el desvanecimiento del gradiente.

Los bloques residuales permiten que la información fluya directamente entre capas mediante conexiones de identidad, reduciendo la pérdida de información y facilitando el entrenamiento de redes más profundas. Cada bloque está compuesto por dos capas convolucionales seguidas de una normalización por lotes (Batch Normalization) y una función de activación ReLU, lo que contribuye a una extracción de características más efectiva.

En su diseño, *ResNet18* inicia con una capa convolucional seguida de una capa de agrupamiento (*max pooling*) para reducir las dimensiones de la entrada. Posteriormente, se disponen cuatro bloques residuales con un número creciente de filtros (64, 128, 256 y 512), adaptándose a las características de mayor complejidad a medida que avanza la red. Finalmente, la arquitectura incluye una capa de agrupamiento promedio (*average pooling*) y una capa completamente conectada (*fully connected*) que utiliza la función SoftMax para la clasificación. Este diseño permite que *ResNet18* sea altamente eficiente en tareas de reconocimiento de patrones y clasificación de imágenes, destacándose por su capacidad para aprovechar redes profundas sin comprometer su desempeño.

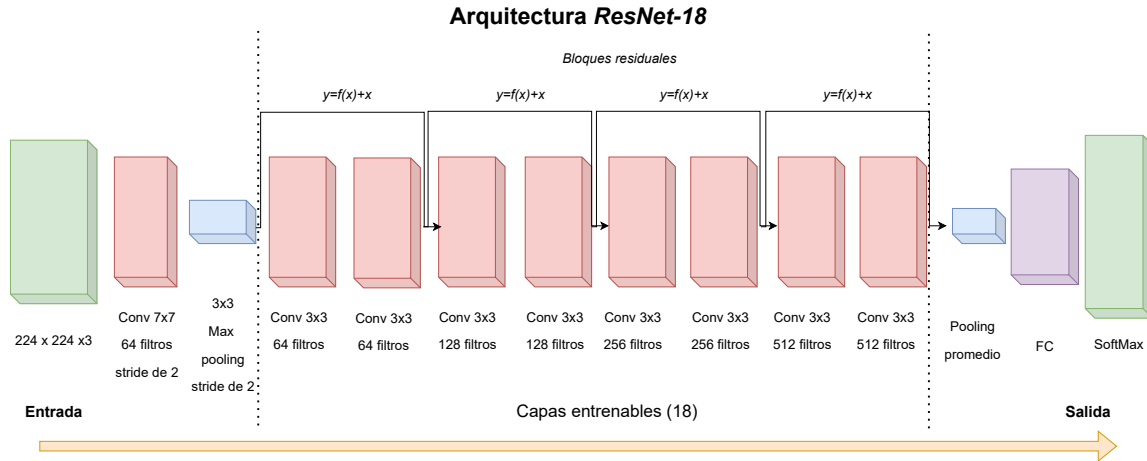


Figura 3.3: Arquitectura de la red neuronal ResNet18.

3.4. DCGAN

Según lo mencionado en el trabajo de investigación de (Foster, 2024), las Redes Generativas Adversarias Convolucionales Profundas (DCGAN, por sus siglas en inglés), constituyen un paradigma arquitectónico significativo en el campo del aprendizaje profundo generativo. Este enfoque metodológico, propuesto inicialmente por (Radford, Metz, y Chintala, 2015), representa una adaptación especializada de las redes GAN que incorpora principios convolucionales para abordar las limitaciones inherentes a las implementaciones previas. La arquitectura DCGAN se fundamenta en un sistema adversarial compuesto por dos entidades neurales que operan en contraposición estratégica. En la Ecuación 3.6 se expresa formalmente generador que se encarga de transformar los vectores de ruido multidimensional en representaciones visuales sintéticas.

$$G_{\theta}(z) \rightarrow \mathbf{x}_{\text{sintética}} \quad (\text{Ecuación 3.6})$$

donde:

- G_{θ} es el generador, parametrizado por θ .
- z es un vector de ruido aleatorio, generalmente muestreado de una distribución normal $z \sim \mathcal{N}(0, I)$, donde I corresponde a la matriz de covarianza que garantiza que no exista correlación entre las diferentes dimensiones del vector de ruido, permitiendo que el modelo aprenda a mapear eficientemente este espacio latente hacia el espacio de las imágenes sintéticas.
- $\mathbf{x}_{\text{sintética}}$ es la muestra generada por G_{θ} , que intenta imitar las muestras reales.

Por otro lado, y como se muestra en la Ecuación 3.7, el discriminador evalúa dichas representaciones contra ejemplos provenientes de la distribución de datos reales. Esta dinámica adversarial impulsa un proceso de refinamiento iterativo que converge hacia la producción de imágenes con características visuales coherentes y estructuralmente similares a las muestras auténticas.

$$D_{\theta}(x) \rightarrow [0, 1] \quad (\text{Ecuación 3.7})$$

donde:

- D_{θ} es el discriminador, parametrizado por θ .
- x es una muestra que puede ser real o generada.
- La salida $[0, 1]$ es un valor escalar que representa la probabilidad de que x provenga del conjunto de datos reales. Si el resultado es cercano a 1, el discriminador cree que x es real; si es cercano a 0, cree que es sintética.

Cabe aclarar que el discriminador evalúa tanto imágenes reales como sintéticas, ajustando sus parámetros para maximizar su capacidad discriminatoria. Simultáneamente, el generador optimiza sus parámetros para producir representaciones que engañen al discriminador. Esta dinámica competitiva establece un mecanismo de retroalimentación donde el discriminador proporciona señales de gradiente que permiten al generador aproximarse progresivamente a la distribución de datos objetivo. El proceso tiene como propósito llegar a un equilibrio donde las imágenes sintéticas resultan indistinguibles de las reales.

3.5. DeepSMOTE

DeepSMOTE es una técnica de modelado generativo que combina el clásico algoritmo de sobremuestreo SMOTE con arquitecturas profundas, presentada en el trabajo de (Dablain y cols., 2023). A diferencia de las Redes Generativas Antagónicas (GANs), este método elimina la necesidad de un discriminador, utilizando en su lugar un esquema de *codificador-decodificador* (autoencoder) junto con una función de pérdida mejorada que incluye un término de penalización. Este enfoque no solo preserva la distribución original de las clases, sino que también genera imágenes sintéticas de alta calidad, enriqueciendo los conjuntos de datos sin introducir ruido o distorsiones. Además, evita problemas comunes de las GANs, como el colapso de modo, donde el modelo genera solo un subconjunto limitado de ejemplos, perdiendo diversidad y afectando la generalización.

El principal componente de *DeepSMOTE* es una arquitectura de autoencoder, diseñada para aprender una representación compacta de los datos en un espacio de características de menor dimensión. Esta estructura se compone de dos partes principales:

- **Codificador:** En la Ecuación 3.8 se describe el proceso de codificación en el modelo *DeepSMOTE*. En este contexto, x representa la imagen de entrada, como podría ser una fotografía de un dígito en el conjunto de datos. Los parámetros θ_E corresponden a los pesos y sesgos de la red, los cuales se ajustan durante el entrenamiento para optimizar la representación de las imágenes. Este paso es fundamental, ya que permite trabajar con una versión simplificada de la imagen en lugar de manejar directamente el espacio de píxeles, lo que facilita la generación de nuevas muestras sintéticas en el espacio latente. Esta capacidad es esencial para equilibrar conjuntos de datos desbalanceados, como cuando ciertas clases tienen muchas menos muestras que otras.

$$z = E(x; \theta_E) \quad (\text{Ecuación 3.8})$$

donde:

- $E(\cdot)$ es el codificador, que en sí mismo es una red neuronal diseñada para transformar la imagen x en una representación latente z .
 - z es el espacio latente, es decir, es una versión comprimida de la imagen original, que conserva las características más relevantes.
 - θ_E corresponden a los pesos y parámetros internos de la red, los cuales se ajustan durante el entrenamiento para optimizar la representación de las imágenes.
- **Decodificador:** Es la parte del modelo que se encarga de permite generar imágenes sintéticas $\hat{x}$ de alta calidad a partir de las muestras creadas en el espacio latente z , lo que es clave para equilibrar conjuntos de datos desbalanceados. Este proceso se describe formalmente en la Ecuación 3.9.

$$\hat{x} = D(z; \theta_D) \quad (\text{Ecuación 3.9})$$

donde:

- $D(\cdot)$ corresponde al decodificador, que toma el vector latente z y lo transforma de nuevo en una imagen $\hat{x}$.
- θ_D son los pesos y ajustes internos de la red que componen el decodificador.

El objetivo principal del autoencoder es minimizar la pérdida de reconstrucción, la cual se calcula como el error cuadrático medio (*MSE*) entre la imagen original x y su reconstrucción $\hat{x}$. En la Ecuación 3.10 se presenta formalmente esta pérdida.

$$\ell_{rec} = \frac{1}{N} \sum_{i=1}^N \|x_i - D(E(x_i))\|^2 \quad (\text{Ecuación 3.10})$$

3.6. Desbalance de Clases

En (Abhishek y Abdelaziz, 2023), se hace referencia que el desequilibrio de clases es un desafío complicado, dado que los modelos de *ML* tienden a favorecer la clase con más ejemplos, lo que puede perjudicar el rendimiento en la clase con menos ejemplos, que suele ser la que más nos interesa. En el caso de las redes neuronales artificiales, ralentiza el aprendizaje de las clases minoritarias, lo que resulta en un modelo que no puede generalizar bien.

Por otra parte, el porcentaje de balance es una métrica esencial en el análisis de desbalance de clases, ya que permite cuantificar la representación relativa de la clase minoritaria en comparación con el total de ejemplos presentes en el conjunto de datos. Este indicador proporciona una visión clara sobre el nivel de equilibrio entre las clases, facilitando la evaluación del impacto del desbalance en el rendimiento de los modelos de aprendizaje automático. Formalmente, el porcentaje de balance se define como la proporción de ejemplos de la clase menos representada respecto al número total de observaciones en todas las clases, y puede expresarse formalmente tal y como se puede ver en la Ecuación 3.11.

$$PB = \frac{|D_{violencia}|}{|D_{no_violencia}| + |D_{violencia}|} \times 100 \quad (\text{Ecuación 3.11})$$

donde:

- *PB*: Porcentaje de Balance.
- $|D_{violencia}|$: Número de ejemplos de la clase minoritaria.
- $|D_{no_violencia}| + |D_{violencia}|$: Total de ejemplos en todas las clases, que es equivalente a $|D|$.

Un valor alto de *PB* indica un conjunto de datos más equilibrado, mientras que un valor bajo refleja un desbalance significativo. Por ejemplo, un porcentaje de balance del 25 % indica que la clase minoritaria representa el 25 % del total de los ejemplos en el conjunto de datos.

Este indicador es útil para evaluar y diseñar estrategias de balanceo, como el *sobremuestreo* o el *submuestreo*, que buscan mitigar los efectos negativos del desbalance de clases en el entrenamiento de modelos de aprendizaje automático.

El **submuestreo**, es una técnica que busca equilibrar el conjunto de datos reduciendo la prevalencia de la clase mayoritaria hasta que su representación sea comparable a la de la clase minoritaria. La Figura 3.4, ejemplifica el proceso de submuestreo, donde la clase **a** tiene mayor número de casos que la clase **b** y posterior al aplicar el submuestreo quedan aproximadamente con la misma cantidad de muestras.

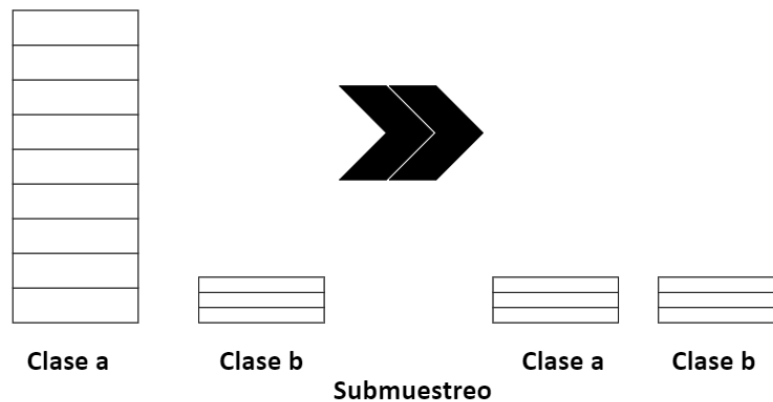


Figura 3.4: Diagrama de submuestreo.

Este enfoque puede ser implementado con los siguientes métodos:

- **RUS:** Se enfoca en reducir el número de ejemplos de la clase mayoritaria mediante la eliminación aleatoria de algunos de ellos. El objetivo es lograr un equilibrio entre la clase mayoritaria y la minoritaria, mejorando así el rendimiento de los modelos de aprendizaje automático que pueden verse afectados por la desproporción en la cantidad de ejemplos de cada clase.
- **Near Miss:** Es un enfoque de submuestreo que identifica y selecciona ejemplos de la clase mayoritaria que se encuentran próximos a los ejemplos de la clase minoritaria. Existen tres variantes de este método, cada una con un criterio específico para determinar qué ejemplos eliminar. *NearMiss-1* se basa en la distancia promedio a los tres vecinos más cercanos de la clase minoritaria, mientras que *NearMiss-2* utiliza la distancia promedio a los tres vecinos más lejanos. Por otro lado, *NearMiss-3* selecciona un número determinado de vecinos de la clase mayoritaria para cada ejemplo de la clase minoritaria.
- **Tomek Links:** Por su parte, identifica pares de ejemplos muy cercanos que pertenecen a clases diferentes, denominados *Tomek Links*. Para abordar el desequilibrio, se elimina el ejemplo de la clase mayoritaria del par, o en algunos casos, ambos ejemplos. Este método se basa en la premisa de que los ejemplos cercanos deberían pertenecer a la misma clase, por lo que los *Tomek Links* probablemente representan ejemplos ruidosos o ambiguos.
- **Edited Nearest Neighbors:** elimina los ejemplos de la clase mayoritaria que son mal clasificados por sus vecinos más cercanos. Para ello, se entrena un clasificador *K-NN* en el conjunto de datos y se eliminan los ejemplos de la clase mayoritaria que el clasificador etiqueta incorrectamente. El objetivo de este método es crear un conjunto de datos más preciso y refinar los límites de decisión entre las clases.

En contra parte, el **sobremuestreo** tiene como principal ventaja mantener toda la información de la clase mayoritaria, además de mejorar el desempeño del modelo en la clase minoritaria.

Como se puede apreciar en la Figura 3.5, este proceso replica los casos de la clase minoritaria (**b**), hasta aproximarse al tamaño de la clase mayoritaria (**a**). Sin embargo, su aplicación puede incrementar significativamente el tiempo de entrenamiento y los recursos computacionales necesarios, e incluso, en algunos casos, derivar en problemas de sobreajuste (*Overfitting*).

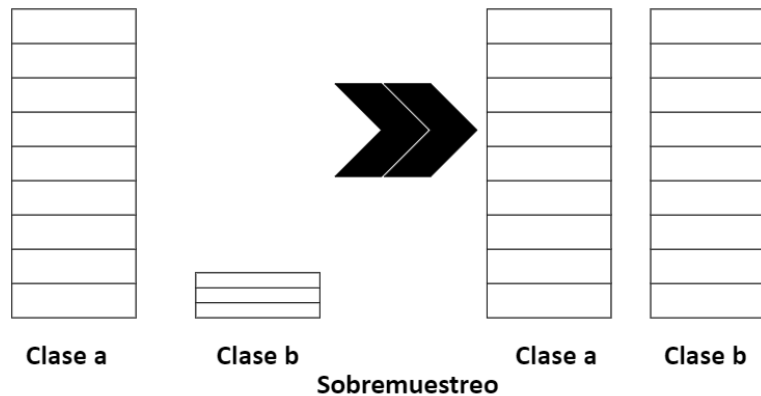


Figura 3.5: Diagrama de sobremuestreo.

Existen algunos métodos de sobremuestreo:

- **ROS:** Es la técnica más sencilla para equilibrar un conjunto de datos, duplicando ejemplos de la clase minoritaria hasta igualar el número de la clase mayoritaria. Aunque fácil de implementar, puede llevar a un sobreajuste al repetir los mismos ejemplos y no considerar la distribución de los datos.
- **SMOTE:** Aborda este problema generando ejemplos sintéticos en lugar de duplicar los existentes. Aunque suele evitar el sobreajuste, puede ser más costosa computacionalmente y no tener en cuenta la distribución de la clase mayoritaria, generando posibles superposiciones.
- **ADASYN:** Es similar a SMOTE, pero se adapta a la distribución de los datos, generando más ejemplos sintéticos en regiones donde la clase minoritaria es más difícil de aprender. Aunque se enfoca en áreas problemáticas para el modelo, también puede ser costoso computacionalmente y generar superposiciones con la clase mayoritaria.

Estas técnicas pueden ser combinadas con enfoques de aprendizaje supervisado para reforzar la capacidad predictiva del clasificador frente a conjuntos de datos desequilibrados, mejorando no solo la precisión global, sino también la sensibilidad y especificidad para cada clase individual. La implementación de estas técnicas requiere una consideración cuidadosa de las propiedades inherentes de la base de datos y del entorno de la problemática a abordar.

Capítulo 4

Metodología

En este capítulo, se exploran y evalúan diversas técnicas de balanceo de imágenes, con el objetivo de optimizar la capacidad de las redes neuronales para clasificar con precisión imágenes que retratan violencia física. Para ello, se han seleccionado cinco conjuntos de datos (datasets), los cuales se detallan más adelante. En cada uno de estos conjuntos, se busca identificar las estrategias más efectivas para abordar el desbalance de clases, un problema recurrente en aplicaciones de visión por computadora y especialmente crítico en contextos de detección de eventos esporádicos, como la violencia física. Para este propósito, se ha modificado la distribución de clases en los conjuntos de datos, introduciendo desbalance de forma controlada al 1 %, 5 %, 10 %, 25 %, 50 % y 75 %. Estos niveles de desbalance fueron seleccionados para abarcar un rango amplio de escenarios, desde condiciones de alta escasez de eventos de violencia (1 %) hasta situaciones más equilibradas (75 %). Esta manipulación permite evaluar el rendimiento del clasificador frente a distintos grados de desbalance, así como determinar qué técnicas de balanceo resultan más eficaces en cada escenario. De esta forma, se busca reflejar las condiciones reales que suelen presentarse en sistemas de monitoreo o vigilancia, donde los eventos de violencia física son considerablemente menos frecuentes que los eventos sin violencia. El objetivo de esta manipulación de clases es asegurar que el modelo pueda generalizar correctamente y clasificar imágenes de violencia física, a pesar de su escasa representación en el conjunto de datos de entrenamiento. Al evaluar el modelo en condiciones de desbalance, se pretende identificar aquellas técnicas de balanceo que no solo maximizan la precisión, sino que también optimizan la sensibilidad del modelo hacia eventos de violencia, crucial en aplicaciones donde la detección oportuna es prioritaria. Este flujo de trabajo completo de la metodología, incluyendo las etapas de preparación, balanceo y evaluación, se muestra en la Figura 4.1.

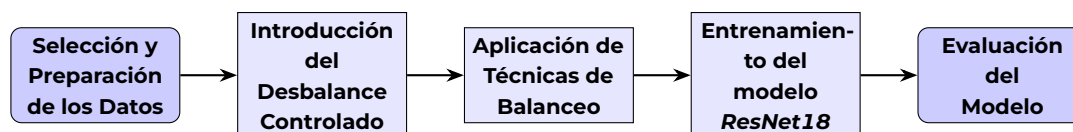


Figura 4.1: Etapas de la metodología.

Para abordar el problema del desbalance de clases en la clasificación, se implementan cuatro enfoques: *Datos Originales* (escenarios de desbalance y una versión completamente balanceada al 100 %), *Métodos tradicionales de balanceo*, *Generación de datos sintéticos mediante Redes Neuronales*, y *Aumento de datos (Data Augmentation)*. Cada uno de estos enfoques tiene como objetivo mejorar la representatividad de la clase minoritaria en el conjunto de datos, incrementando así la capacidad de generalización del modelo de red neuronal y su precisión en la detección de eventos de violencia.

1. **Datos Originales:** Este enfoque utiliza el conjunto de datos en su forma original, sin aplicar técnicas de balanceo, para evaluar el rendimiento del modelo en condiciones de desbalance natural. Además, incluye un escenario completamente balanceado al 100 %, donde se iguala la cantidad de muestras entre clases, permitiendo una comparación directa con otros métodos de balanceo. Esta combinación ofrece una referencia clara del impacto del desbalance en el desempeño del modelo.
2. **Métodos tradicionales de balanceo:** Este enfoque incluye técnicas ampliamente utilizadas para abordar problemas de desbalance de clases mediante el ajuste de la distribución de las muestras en el conjunto de datos. En primer lugar, se implementan métodos de sobremuestreo, como *ROS*, que consiste en duplicar aleatoriamente instancias de la clase minoritaria, y *SMOTE* y *ADASYN*, que generan ejemplos sintéticos para aumentar la representatividad de la clase minoritaria sin duplicación directa. Por otro lado, se aplican técnicas de submuestreo, como *RUS*, que reduce el número de ejemplos de la clase mayoritaria al eliminar instancias de forma aleatoria. En adición a esto, dentro de las técnicas de submuestreo, se incluye la selección de ejemplos, la cual busca mejorar el balance de clases al eliminar o seleccionar instancias de la clase mayoritaria de manera estratégica. Esta subcategoría abarca métodos como *NM*, *ENN*, *TL* y *OSS*. Por ejemplo, *NM* selecciona muestras de la clase mayoritaria que están más cercanas a la clase minoritaria, mientras que *TL* elimina pares de muestras de diferentes clases que están muy cercanas entre sí, ayudando a definir mejor los límites de decisión entre clases. En el Algoritmo 1 que muestra de forma clara como se evalúan los modelos que fueron entrenados en escenarios de desbalance, aplicando métodos tradicionales para mitigar dicho desbalance aplicado.

Algoritmo 1: Proceso de balanceo usando métodos tradicionales

Entrada: Conjuntos de datos $D = (\text{AirLab}, \text{SCVD}, \text{RLVS}, \text{UBI-Fights}, \text{RWJF2000})$,

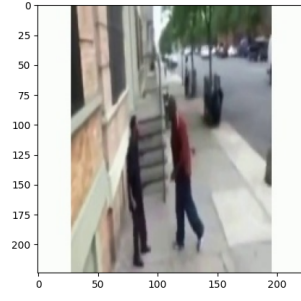
Niveles de desbalance $N = (1\%, 5\%, 10\%, 25\%, 50\%, 75\%)$,

Métodos de balance $M = (\text{ROS}, \text{SMOTE}, \text{ADASYN}, \text{RUS}, \text{ENN}, \text{OSS}, \text{TL}, \text{NM})$

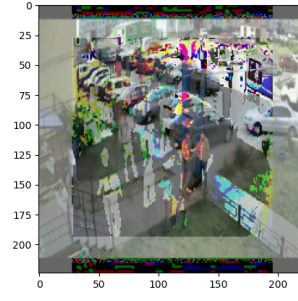
Salida: Modelos entrenados y evaluados $f_{\theta}(\mathbf{x})$ para cada combinación

```
1 Definir los resultados  $R = \{\}$  para almacenar métricas de evaluación;
2 //Iteración sobre conjuntos de datos;
3 para cada conjunto de datos  $D_{\text{actual}}$  en  $D$  hacer
4     Dividir  $D_{\text{actual}}$  en subconjuntos: train y test;
5     //Iteración sobre niveles de desbalance;
6     para cada nivel de desbalance  $N_{\text{desbalance}}$  hacer
7         Aplicar  $N_{\text{desbalance}}$  al subconjunto train, generando  $D_{\text{train\_desbalanceado}}$ ;
8         //Iteración sobre métodos de balance;
9         para cada método de balance  $M_{\text{balance}}$  hacer
10             Aplicar  $M_{\text{balance}}$  a  $D_{\text{train\_desbalanceado}}$ , generando  $D_{\text{train\_balanceado}}$ ;
11             //Entrenamiento del modelo;
12             Entrenar el modelo  $f_{\theta}(\mathbf{x})_{\text{actual}}$  usando  $D_{\text{train\_balanceado}}$ ;
13             //Evaluación del modelo;
14             Evaluar  $f_{\theta}(\mathbf{x})_{\text{actual}}$  utilizando  $D_{\text{test}}$  para calcular las métricas finales;
15             Almacenar los resultados de  $f_{\theta}(\mathbf{x})_{\text{actual}}$  en  $R$ ;
16 //Finalización del proceso;
17 Guardar los modelos entrenados y los reportes de evaluación para todas las
    combinaciones en  $R$ ;
```

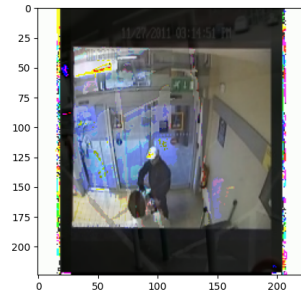
En la Figura 4.2 se ilustran las muestras generadas por métodos tradicionales de balanceo de datos. Específicamente, las subfiguras (a) - (h) muestran los resultados de aplicar métodos como *ROS*, *SMOTE*, *ADASYN*, *RUS*, *TL*, *ENN*, *OSS* y *NM*, cada una representando un enfoque distintivo. Esta visualización proporciona un marco comparativo de cómo diferentes métodos pueden transformar y equilibrar los conjuntos de datos.



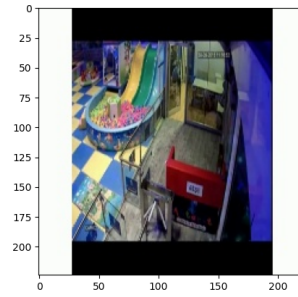
(a) ROS



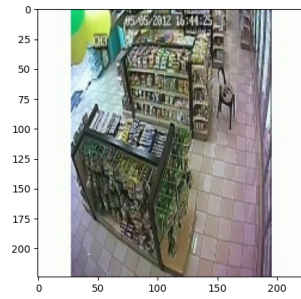
(b) SMOTE



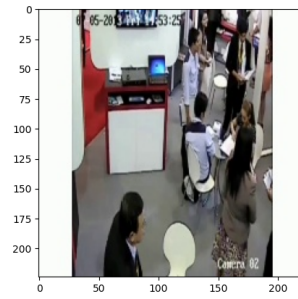
(c) ADASYN



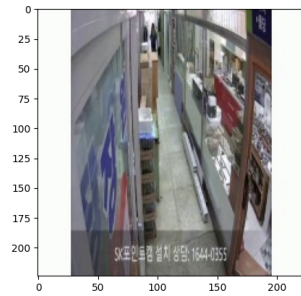
(d) RUS



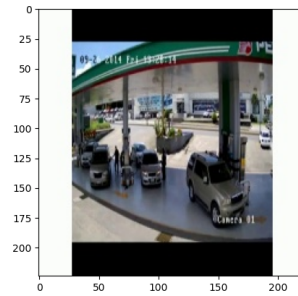
(e) TL



(f) ENN



(g) OSS



(h) NM

Figura 4.2: Muestras de violencia generadas mediante métodos tradicionales de sobremuestreo (a),b),c)) y no violencia para el caso de los métodos tradicionales de submuestreo (d),e),f),g) y h)).

-
3. **Generación de datos sintéticos mediante Modelos Generativos:** Este enfoque utiliza redes neuronales generativas para crear imágenes sintéticas de la clase minoritaria, es decir, escenas de violencia física, con el fin de complementar el conjunto de datos original y mejorar la capacidad del modelo para manejar escenarios desbalanceados. Se emplean dos modelos generativos tales como: *DCGAN* y *DeepSMOTE*. La primera permite generar imágenes artificiales que capturan patrones visuales característicos de la violencia física, incrementando la diversidad en el conjunto de datos sin necesidad de duplicar instancias. La segunda, *DeepSMOTE*, combina la generación de datos sintéticos con una arquitectura que garantiza la variedad en las nuevas muestras, ayudando a que el modelo pueda generalizar mejor en condiciones de desbalance extremo.

Tal y como se puede observar en el Algoritmo 2, ambas técnicas se utilizan para evaluar si los datos sintéticos mejoran la precisión y sensibilidad del modelo en la detección de violencia física en diferentes escenarios de desbalance aplicado, comparando su impacto en la capacidad de generalización del clasificador.

Algoritmo 2: Proceso de balanceo mediante muestras sintéticas generadas a partir de modelos generativos

Entrada: Conjuntos de datos $D = (\text{AirLab}, \text{SCVD}, \text{RLVS}, \text{UBI-Fights}, \text{RWJF2000})$,

Niveles de desbalance $N = (1\%, 5\%, 10\%, 25\%, 50\%, 75\%)$,

Modelos generativos $M = (\text{DCGAN}, \text{DeepSMOTE})$

Salida: Modelos entrenados y evaluados $f_{\theta}(\mathbf{x})$ para cada combinación

- 1 Definir los resultados $R = \{\}$ para almacenar métricas de evaluación;
 - 2 **//Iteración sobre conjuntos de datos;**
 - 3 **para** cada conjunto de datos D_{actual} en D **hacer**
 - 4 Dividir D_{actual} en subconjuntos: *train* y *test*;
 - 5 **//Iteración sobre niveles de desbalance;**
 - 6 **para** cada nivel de desbalance $N_{\text{desbalance}}$ **hacer**
 - 7 Aplicar $N_{\text{desbalance}}$ al subconjunto *train*, generando $D_{\text{train_desbalanceado}}$;
 - 8 **//Iteración sobre modelos generativos de balance;**
 - 9 **para** cada método de balance M_{balance} **hacer**
 - 10 Aplicar M_{balance} a $D_{\text{train_desbalanceado}}$, generando $D_{\text{train_balanceado}}$;
 - 11 **//Entrenamiento del modelo;**
 - 12 Entrenar el modelo $f_{\theta}(\mathbf{x})_{\text{actual}}$ usando $D_{\text{train_balanceado}}$;
 - 13 **//Evaluación del modelo;**
 - 14 Evaluar $f_{\theta}(\mathbf{x})_{\text{actual}}$ utilizando D_{test} para calcular las métricas finales;
 - 15 Almacenar los resultados de $f_{\theta}(\mathbf{x})_{\text{actual}}$ en R ;
 - 16 **//Finalización del proceso;**
 - 17 Guardar los modelos entrenados y los reportes de evaluación para todas las combinaciones en R ;
-

En la Figura 4.3, se muestra claramente, como mejora la calidad de las imágenes generadas por modelos generativos como *DCGAN* y *DeepSMOTE* a medida que aumenta el porcentaje de balance.



Figura 4.3: Muestras creadas por modelos generativos (*DeepSMOTE* y *DCGAN*).

-
4. **Aumento de datos:** Este enfoque utiliza transformaciones de las imágenes originales, dentro de estas se encuentran las rotaciones y cambios de escala, que simulan diferentes ángulos de visión y tamaños de los objetos en las imágenes, permitiendo que el modelo se adapte a variaciones comunes en escenarios de vigilancia. También se aplican ajustes de brillo y contraste, lo cual ayuda a que el modelo responda a diversas condiciones de iluminación, como las que se encuentran en cámaras de seguridad en exteriores o lugares poco iluminados.

Si bien el Data Augmentation no corrige directamente el desbalance de clases, amplía de manera significativa el número de representaciones de la clase minoritaria (violencia) al generar variaciones de las muestras existentes. Esto permite que el modelo se entrene con una gama más diversa de ejemplos de violencia sin necesidad de crear nuevas instancias de datos reales, lo cual fortalece su habilidad para reconocer patrones relevantes en situaciones poco comunes. Además, el aumento de datos ayuda a reducir el riesgo de sobreajuste, al exponer al modelo a una mayor diversidad de escenarios.

En conjunto, estas técnicas permiten que el modelo no solo mejore en su generalización en distintos escenarios de desbalance, sino que también trate de manera más equitativa ambas clases, optimizando su rendimiento en contextos donde la clase de violencia es significativamente menos representada. Véase el Algoritmo 3.

Algoritmo 3: Proceso de balanceo mediante aumento de datos

Entrada: Conjuntos de datos $D = (\text{AirLab}, \text{SCVD}, \text{RLVS}, \text{UBI-Fights}, \text{RWJF2000})$,

Niveles de desbalance $N = (1\%, 5\%, 10\%, 25\%, 50\%, 75\%)$,

Salida: Modelos entrenados y evaluados $f_{\theta}(\mathbf{x})$ para cada combinación

- 1 Definir los resultados $R = \{\}$ para almacenar métricas de evaluación;
 - 2 **//Iteración sobre conjuntos de datos;**
 - 3 **para cada conjunto de datos D_{actual} en D hacer**
 - 4 Dividir D_{actual} en subconjuntos: `train` y `test`;
 - 5 **//Iteración sobre niveles de desbalance;**
 - 6 **para cada nivel de desbalance $N_{\text{desbalance}}$ hacer**
 - 7 Aplicar $N_{\text{desbalance}}$ al subconjunto `train`, generando $D_{\text{train_desbalanceado}}$;
 - 8 **//Aumento de datos;**
 - 9 Aplicar **Aumento de datos** a $D_{\text{train_desbalanceado}}$, generando $D_{\text{train_balanceado}}$;
 - 10 **//Entrenamiento del modelo;**
 - 11 Entrenar el modelo $f_{\theta}(\mathbf{x})_{\text{actual}}$ usando $D_{\text{train_balanceado}}$;
 - 12 **//Evaluación del modelo;**
 - 13 Evaluar $f_{\theta}(\mathbf{x})_{\text{actual}}$ utilizando D_{test} para calcular las métricas finales;
 - 14 Almacenar los resultados de $f_{\theta}(\mathbf{x})_{\text{actual}}$ en R ;
 - 15 **//Finalización del proceso ;**
 - 16 Guardar los modelos entrenados y los reportes de evaluación para todas las combinaciones en R ;
-

La efectividad de cada técnica se evalúa entrenando un modelo de red neuronal convolucional (ResNet18) con los diferentes conjuntos de datos balanceados y aumentados. Para esto, se emplean varias métricas: Accuracy, que mide la precisión general del modelo; Precision y Recall, que cuantifican el rendimiento en cada clase; F1 Score, una métrica armónica entre precisión y sensibilidad; y G-Mean, que es particularmente relevante para conjuntos de datos desbalanceados. Esta última métrica se considera especialmente importante en este estudio, ya que refleja un balance entre las tasas de clasificación de las distintas clases, mitigando el sesgo hacia la clase mayoritaria.

En la Figura 4.4, se aprecian las muestras resultantes después de aplicar el Aumento de datos.

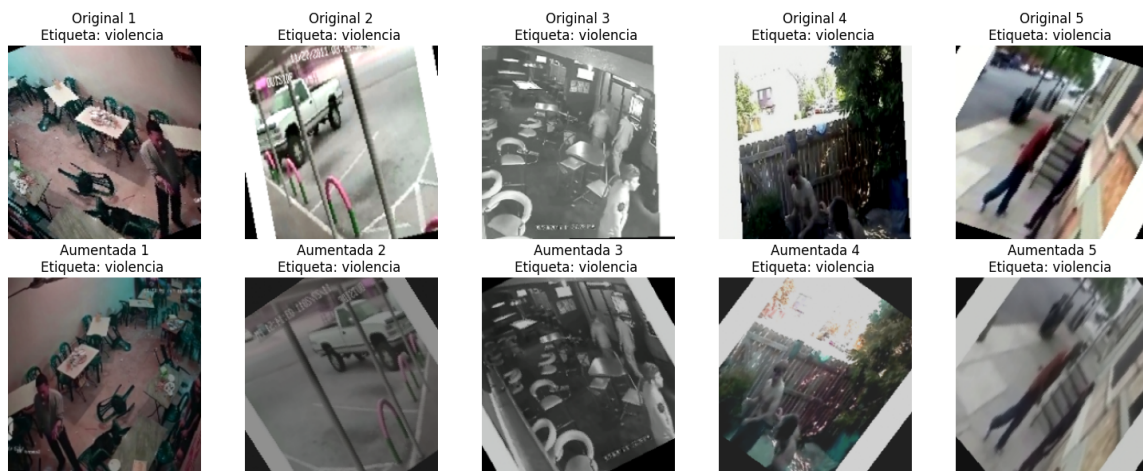


Figura 4.4: Muestras creadas por Aumento de Datos.

4.1. Conjuntos de datos

Para el presente trabajo de investigación, se usaron los siguientes conjuntos de datos (*data-sets*); *Airtlab*¹ (Bianculli y cols., 2020), *SCVD*² (Aremu, Zhiyuan, Alameeri, Khan, y Saddik, 2024), *RWF2000*³ (Cheng, Cai, y Li, 2021), *RLVSD*⁴ (Soliman y cols., 2019), *UBI-Fights*⁵ (Degardin y Proença, 2020). Estos mismos son secuencias de videos, que posteriormente se extraerán los fotogramas de cada video.

Como se muestra en la Figura 4.5, los *Datasets SCVD_a* y *RWF2000_b* retratan escenas de violencia y no violencia en distintos ambientes urbanos, con grabaciones que provienen principalmente de cámaras de seguridad (CCTV) y videos de internet, respectivamente. Estas

¹ Disponible en: <https://github.com/airtlab/A-Dataset-for-Automatic-Violence-Detection-in-Videos>

² Disponible en: <https://www.kaggle.com/datasets/toluwaniamemu/smartcity-cctv-violence-detection-dataset-scvd>

³ Disponible en: <https://github.com/mchengny/RWF2000-Video-Database-for-Violence-Detection>

⁴ Disponible en: <https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset>

⁵ Disponible en: <http://socia-lab.di.ubi.pt/EventDetection/>

fuentes capturan la diversidad de escenarios, condiciones de iluminación y calidad de imagen que se encuentran en situaciones reales, aunque dificultan el etiquetado preciso de las acciones violentas. Por otro lado, *Airtlab_e* se centra en un escenario controlado con escenas de violencia simulada por actores, ofreciendo imágenes de alta calidad y etiquetado preciso, pero con menor realismo y diversidad de acciones violentas. *RLVSD_d* se caracteriza por su enfoque en la diversidad de acciones violentas, incluyendo no solo peleas, sino también robos y agresiones en diferentes contextos, brindando una representación más completa de la violencia real. Finalmente, *UBI – Fights_e* se enfoca en la detección de eventos en videos, con especial atención a las peleas, pero la información disponible sobre este conjunto de datos es limitada, lo que dificulta una evaluación completa de sus características.



(a) SCVD



(b) RWF-2000



(c) Airtlab



(d) RLVSD



(e) UBI-Fights

Figura 4.5: Muestras de los conjuntos de datos sin procesar.

En la Tabla 4.1 se encuentra de forma detallada el número de imágenes extraídas por Dataset.

Tabla 4.1: Distribución de imágenes por conjunto de datos.

Airlab	SCVD	RWF2000	RLVSD	UBI-Fights
5200	54368	20000	16960	30000

Posteriormente y en base a los estándares en la estructuración de *datasets*, se segmentaron en tres subconjuntos, lo cuales son; entrenamiento (*train*), validación (*validation*) y pruebas (*test*). Los cuales están constituidos por el 70 % en el caso de entrenamiento. Por su parte, el conjunto de validación va arraigado al de entrenamiento, siendo constituido por un 20 % de este. Finalmente, el conjunto de pruebas está compuesto por un 30 % de la totalidad del conjunto de datos.

Adicionalmente y como se puede ver en la tabla 4.2, los subconjuntos de entrenamiento se han desbalanceado *aleatoriamente*, con la finalidad de ver el impacto del desbalance sobre la eficiencia del modelo de red neuronal. Asimismo, ver que técnica mitiga mejor el error en función al escenario desbalance correspondiente.

Tabla 4.2: Desglose de muestras por porcentaje de balance con división en violencia y no violencia.

Porcentaje de balance	Airlab		SCVD		RWF2000		UBI-Fights		RLVSD	
	NV	V	NV	V	NV	V	NV	V	NV	V
1 %	1819	18	20910	209	7000	70	10500	105	5936	59
5 %		90		1045		350		525		296
10 %		181		2091		700		1050		593
25 %		454		5227		1750		2625		1484
50 %		909		10455		3500		5250		2968
75 %		1364		15682		5250		7875		4452
100 %		1819		20910		7000		10500		5936

4.2. Redes Neuronales

En este apartado, al igual que en el apartado anterior, se especifican las arquitecturas y los parámetros utilizados en las redes neuronales implementadas. Esto abarca tanto los modelos utilizados en los experimentos como los propuestos para resolver el problema planteado. Se describen las configuraciones específicas, incluyendo el tipo de capas, la cantidad de neuronas, las funciones de activación, así como los hiperparámetros clave como la tasa de aprendizaje y el número de épocas. Estos elementos fueron seleccionados cuidadosamente para optimizar el rendimiento de los modelos y garantizar su adecuación a los datos y objetivos del estudio.

4.2.1. ResNet18

ResNet18 es una red neuronal convolucional, conocida por ser ligera, ya que, como su nombre lo indica, tiene 18 capas. Es eficiente en comparación con otras redes más complejas. A pesar

de tener solo 18 capas, logra excelentes resultados en tareas de visión por computadora, como la clasificación de imágenes. Gracias a su diseño compacto, ofrece un buen equilibrio entre precisión y velocidad, lo que la convierte en una opción ideal para aplicaciones donde los recursos computacionales son limitados, pero se necesitan resultados de alta calidad.

Para este caso, y dado que es una red preentrenada, se ha conservado el conocimiento previo, lo que permite congelar las capas convolucionales, un proceso conocido como transferencia de aprendizaje. Al congelar estas capas, solo se entrena la capa completamente conectada (FC), lo que reduce el tiempo de entrenamiento. Además, al requerir pocas épocas para ajustarse a nuevas tareas, esta técnica sigue proporcionando excelentes resultados. Este enfoque aprovecha las características previamente aprendidas y facilita la adaptación a diferentes aplicaciones de manera rápida y eficiente.

En el presente trabajo de investigación se ha tratado de seleccionar rigurosamente los hiperparámetros que guían el entrenamiento, tales como la tasa de aprendizaje, el tamaño del lote, el número de épocas y el método de optimización. En la tabla 4.3, se describen los hiperparámetros clave utilizados al ajustar una *ResNet18* pre entrenada, con el objetivo que el modelo se ajuste bien.

Tabla 4.3: Hiperparámetros utilizados para el ajuste fino de *ResNet18*.

Hiperparámetro	Implementación
Pesos pre-entrenados	IMAGENET1K_V1
Tasa de aprendizaje	0.001
Tamaño de lote	32
Número de épocas	2
Método de optimización	Adam
Capas congeladas	Todas excepto layer4 y fc
Función de pérdida	CrossEntropyLoss

4.2.2. DCGAN

Esta red neuronal, pertenece al grupo denominado como *modelado generativo*, que, a diferencia de la *ResNet18*, esta no es discriminante, sino que es usada para generar muestras sintéticas a partir de ruido aleatorio. Cabe hacer hincapié que se entrena con el mismo conjunto de datos de entrenamiento, con la que entrena el clasificador. Como tal el trabajo de investigación no se centra en esta herramienta, sino que más bien se implementa para comparar resultados frente a la propuesta de investigación. Así mismo está compuesta por un *generador* y un *discriminador* que compiten entre sí durante el entrenamiento, por decirlo de alguna manera. En la tabla 4.4, se describen los hiperparámetros utilizados para este modelo.

Tabla 4.4: Hiperparámetros utilizados para la implementación de la red DCGAN.

Hiperparámetro	Implementación
Tasa de aprendizaje	0.002
Tamaño de lote	64
Número de épocas	300
Método de optimización	Adam
Parámetro de ruido	0.1
Tamaño del espacio latente	100

4.2.3. DeepSMOTE

DeepSMOTE es una técnica que mejora el equilibrio de clases en datos desbalanceados, combinando la potencia de redes neuronales profundas y la generación de muestras sintéticas. A través del uso de redes neuronales. Esta técnica se enfoca en captar relaciones más complejas entre las clases, superando las limitaciones de métodos más simples como *SMOTE* tradicional. En la tabla 4.5, se describen los hiperparámetros utilizados para este método.

Tabla 4.5: Hiperparámetros utilizados para la implementación de *DeepSmote*.

Hiperparámetro	Implementación
Tasa de aprendizaje	0.0002
Tamaño de lote	8
Número de épocas	500
Método de optimización	Adam
Parámetro de ruido	0.1
Tamaño del espacio latente	600

4.3. Optimización del modelo

El proceso de optimización del modelo clasificador se llevó a cabo mediante la configuración de un modelo base, en este caso *ResNet18*, adaptado para la tarea de clasificación binaria. Tal y como se puede ver en el Algoritmo 4, primero se cargó el modelo y se ajustó su capa de salida para adaptarse a la naturaleza del problema de detección de violencia en imágenes. Se congelaron las capas iniciales para aprovechar el aprendizaje previo, permitiendo que solo las capas finales ajusten sus parámetros durante el entrenamiento. A continuación, se definió la función de pérdida como entropía cruzada, y se seleccionó el optimizador Adam con una tasa de aprendizaje de 0.001. El modelo se entrenó durante un número de épocas E , donde en cada época, se procesaron lotes de datos de entrenamiento, enviándolos a la *GPU* para acelerar el cálculo. Para cada lote, se calculó la salida del modelo, se evaluó la pérdida y se aplicaron los ajustes de parámetros mediante *backpropagation*. Al final de cada época, se

evaluó el rendimiento en el conjunto de validación para monitorear el progreso. Finalmente, al término de todas las épocas, se guardó el modelo optimizado para su uso futuro.

Algoritmo 4: Procedimiento para entrenar y optimizar el modelo de red neuronal $f_{\theta}(\mathbf{x})$

Entrada: $f_{\theta}(\mathbf{x})$: Modelo, D_{train} : datos de entrenamiento, D_{val} : datos de validación, D_{test} : datos de prueba, E : número de épocas, α : tasa de aprendizaje

Salida: $f_{\theta}(\mathbf{x})_{\text{opt}}$: Modelo optimizado

- 1 **1. Configuración del Modelo clasificador;**
 - 2 Cargar el modelo $f_{\theta}(\mathbf{x})$ y ajustar la última capa para clasificación binaria;
 - 3 Congelar capas iniciales para aprovechar el aprendizaje previo y permitir el ajuste de las últimas capas;
 - 4 Mover el modelo ($f_{\theta}(\mathbf{x})$) a la *GPU*, si está disponible;
 - 5 **2. Definir la Función de Pérdida y el Optimizador;**
 - 6 Utilizar `CrossEntropyLoss` como función de pérdida;
 - 7 Configurar `Adam` como optimizador con tasa de aprendizaje $\alpha = 0.001$;
 - 8 **3. Entrenamiento por Épocas;**
 - 9 **para** $e = 1$ **to** E **hacer**
 - 10 **para** cada lote en los datos de entrenamiento **hacer**
 - 11 Enviar el lote a la *GPU*;
 - 12 Calcular la salida del modelo $\hat{y} = f_{\theta}(\mathbf{x})$;
 - 13 Calcular la pérdida $L = \ell(y, \hat{y})$;
 - 14 Realizar retropropagación y actualizar los parámetros de $f_{\theta}(\mathbf{x})$ con el optimizador `Adam`;
 - 15 Evaluar $f_{\theta}(\mathbf{x})$ en los datos de validación, calculando métricas de rendimiento;
 - 16 **4. Guardado del Modelo;**
 - 17 Guardar el modelo optimizado $f_{\theta}(\mathbf{x})_{\text{opt}}$ para uso futuro;
-

4.4. Configuración del Entorno de Ejecución

Para garantizar la reproducibilidad de los experimentos y un rendimiento óptimo en el procesamiento de datos, los experimentos se llevaron a cabo en un entorno de *hardware* y *software* cuidadosamente configurado. El modelo principal se desarrolló en PyTorch, aunque en casos específicos de generación de datos sintéticos (*DCGAN* y *DeepSMOTE*) se utilizó *TensorFlow*. En la Tabla 4.6, se detalla la configuración de *hardware* y las principales bibliotecas utilizadas.

Tabla 4.6: Configuración del entorno de ejecución

Componente	Especificación
Sistema Operativo	Ubuntu 22.04
Procesador	Intel Xeon E-2186G (12 núcleos) a 4.700 GHz
GPU	NVIDIA GeForce RTX 3060 con 12 GB de VRAM
Memoria RAM	64 GB
Aceleración por GPU	CUDA Toolkit 11.8 y cuDNN
Frameworks	PyTorch (principal), TensorFlow 2.14.0 (para DCGAN y DeepSMOTE)
Bibliotecas	NumPy 1.23.5, Pandas 2.1.0, scikit-learn 1.3.0, Matplotlib, Seaborn, Torchvision

Capítulo 5

Resultados y discusión

En este capítulo se detallan los resultados obtenidos a partir de los experimentos realizados. La Tabla 1 ilustra el comportamiento del modelo entrenado con todos los *datasets* para cada nivel de porcentaje de la clase minoritaria. Conforme se incrementa la proporción de la clase minoritaria. Se observa que, en escenarios de desbalance alto (1 % y 5 %), los valores tanto de *Recall* como de *g-mean* se mantienen relativamente bajos. Sin embargo, dichas métricas mejoran sustancialmente aplicando métodos de balanceo, ya sea generando datos sintéticos o mediante técnicas de balanceo más tradicionales, y estrategias de aumento de datos. Conforme el desbalance se reduce (10 %, 25 %, 50 % y 75 %), el modelo alcanza valores elevados en todas las métricas, destacando en particular el *g-mean*, lo que sugiere una mayor capacidad de generalización del clasificador. Las Figuras 5.1 a 5.6 permiten observar de manera clara, cómo varía el *g-mean* promedio a medida que se modifican tanto la técnica de balanceo como la representación de la clase minoritaria.

5.1. Resultados

En este trabajo, se ha empleado la media geométrica (*g-mean*). Según (Kubat y Matwin, 1997; H. He y Garcia, 2009) es una métrica de evaluación particularmente relevante para escenarios de clasificación con desbalance de clases. Esta medida resulta óptima para conjuntos de datos altamente asimétricos debido a su sensibilidad al rendimiento individual por clase. La Ecuación 5.1 expresa formalmente el *g-mean* como la media geométrica de precisiones parciales por clase:

$$g-mean = \sqrt[J]{\prod_{j=1}^J acc_j} \quad (\text{Ecuación 5.1})$$

donde:

- *J* representa el total de clases.

- acc_j la exactitud en la clase j , definida como $1_{hit} = (\text{instancias correctamente clasificadas de clase } j) / (\text{total de instancias de clase } j)$.

Esta formulación garantiza que el $g - mean$ solo alcance valores elevados cuando todas las clases presentan alta exactitud, penalizando eficazmente el rendimiento desequilibrado y proporcionando una evaluación robusta en distribuciones asimétricas donde la capacidad discriminatoria debe mantenerse consistente independientemente de la prevalencia de cada clase.

Las Figuras 5.1 a 5.6 permiten observar de manera clara, cómo varía el $g - mean$ promedio a medida que se modifican tanto la técnica de balanceo como la representación de la clase minoritaria. En la Figura 5.1 se contemplan doce métodos de balanceo, incluido el conjunto Original con solo 1 % de la clase minoritaria respecto a la mayoritaria, *ROS* logra el valor de $g - mean$ promedio más elevado (0.750). Este resultado supera a métodos más complejos o sofisticados, como *SMOTE* (0.706) y *DCGAN* (0.687). En contraste, *Near Miss* evidencia el desempeño más bajo (0.131), reflejando las limitaciones críticas de los esquemas de sub-muestreo basados en distancias cuando se trabaja con una clase minoritaria sumamente reducida. Por ende, las metodologías de sobremuestreo parecen presentar ventajas notables en escenarios con desbalance extremo, mientras que la eliminación de casos de la clase mayoritaria podría comprometer la representatividad del espacio de características y afectar la capacidad de clasificación.

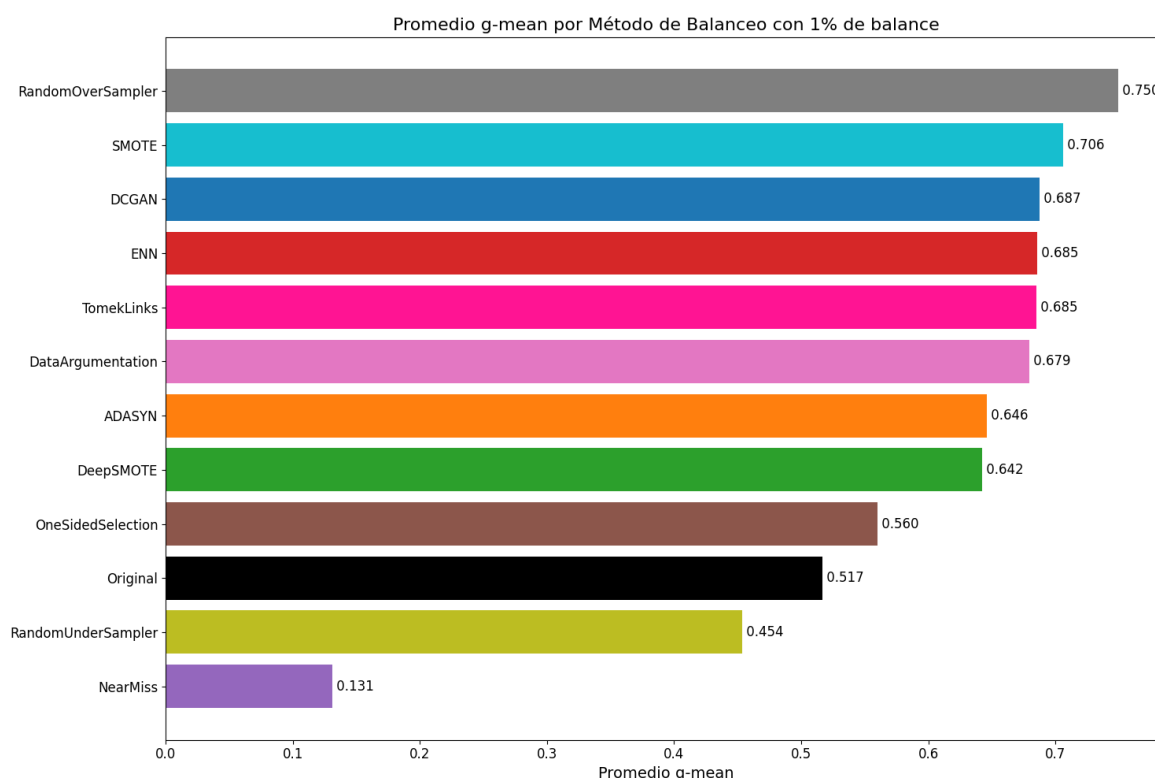


Figura 5.1: Comparación del *g-mean* promedio de todos los métodos de balanceo en un escenario de 1 % de balance.

Por su parte, la Figura 5.2 muestra resultados con un escenario de balance de 5 % en la clase minoritaria, donde claramente se observa que el *g-mean* promedio aumenta de manera notable, respecto al escenario anterior; *ROS* se sigue manteniendo a la cabeza (0.922), seguido de *Tomek Links* (0.904) y *Data Augmentation* (0.883). Dichos valores subrayan la relevancia de las estrategias de sobremuestreo en situaciones donde la asimetría de clases es moderada. A pesar de lograr una mejora relativa, *Near Miss* registra de nuevo la métrica más baja (0.398), lo que sugiere que los métodos de submuestreo basados en proximidad enfrentan dificultades para preservar la información esencial de la clase minoritaria.

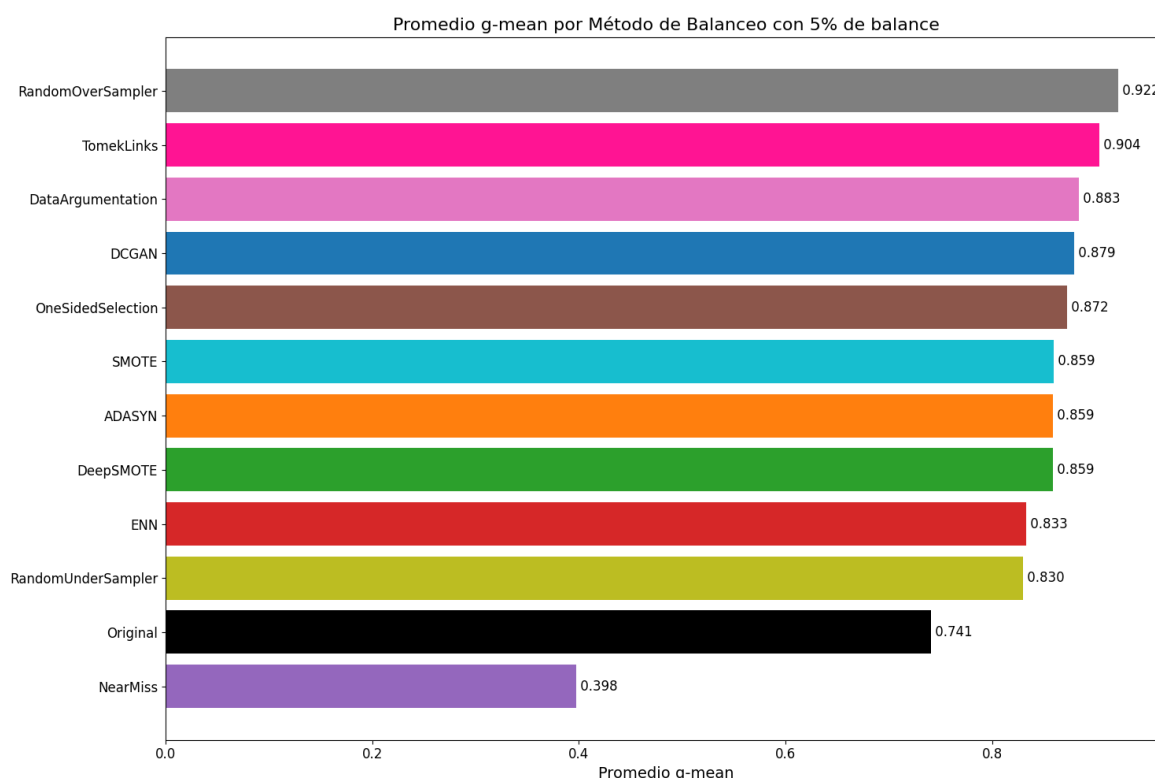


Figura 5.2: Comparación del *g-mean* promedio de todos los métodos de balanceo en un escenario de 5 % de balance.

Cuando se la proporción de la clase minoritaria al 10 %, se puede observar un incremento notable en la eficacia del modelo. En este caso, *ROS* conserva el *g-mean* promedio más alto (0.961), mientras que las arquitecturas generativas *DCGAN* (0.951) y *Deep SMOTE* (0.941) ofrecen un desempeño cercano. Estos resultados experimentales respaldan la idea de que la generación sintética de ejemplos minoritarios, apoyada en redes neuronales, puede salvaguardar las características fundamentales de la distribución de datos en escenarios de balance no tan alto. Aun cuando *Near Miss* exhibe cierta mejoría (0.497), continúa rezagado en comparación con los métodos de sobremuestreo, lo que sigue confirmando que los enfoques generativos y de sobremuestreo brindan ventajas notorias sobre la reducción selectiva de ejemplos mayoritarios para problemas de clasificación con desbalance moderado.

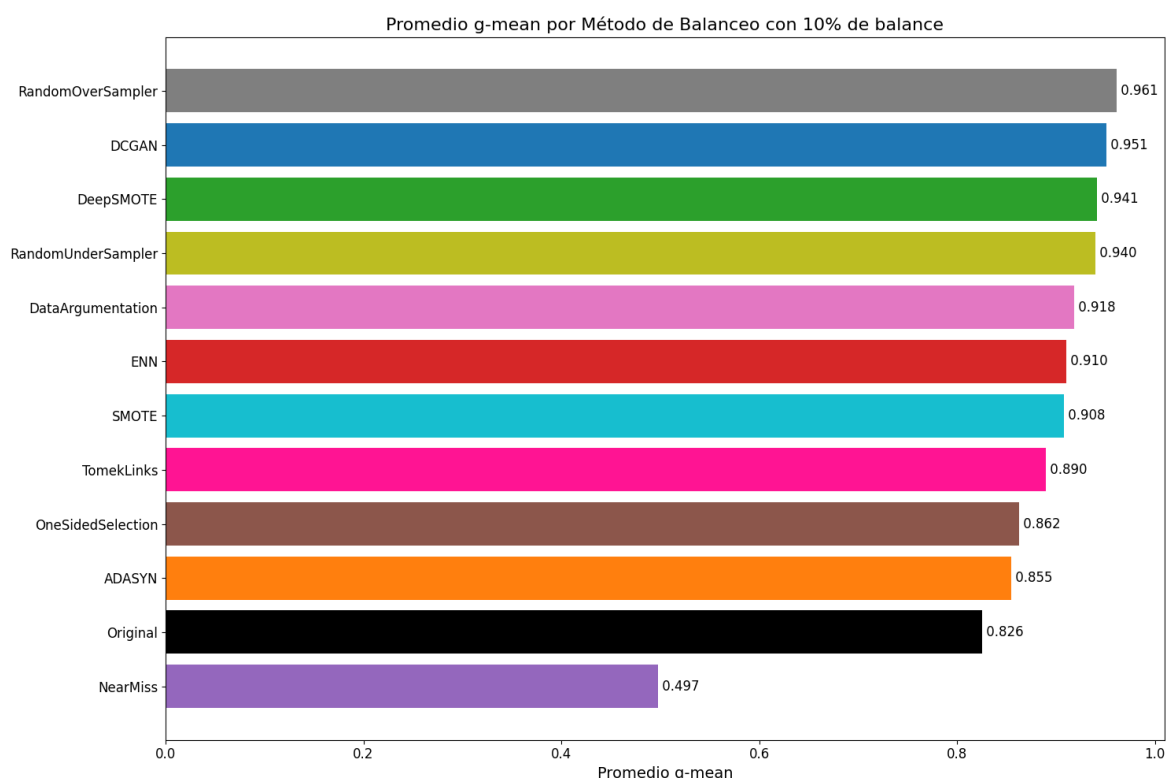


Figura 5.3: Comparación del *g-mean* promedio de todos los métodos de balanceo en un escenario de 10 % de balance.

Cuando el porcentaje de la clase minoritaria aumenta a 25 %, tal como se presenta en la Figura 5.4, se observan resultados relativamente altos respecto a los escenarios de balance anteriores. La mayoría de los métodos superan el 0.90 de *g-mean* promedio, con *ROS* nuevamente a la cabeza (0.979) y *DCGAN* cerca de su desempeño (0.976). Por otro lado, *RUS* (0.959) confirma que, en niveles de desbalance moderado, los procedimientos de submuestreo pueden aproximarse al rendimiento de técnicas de sobremuestreo. Aun así, *Near Miss* (0.661) permanece por debajo de alternativas como *SMOTE* (0.962) y *Data Augmentation* (0.964). De esta forma, aun se sigue corroborando que los métodos basados en criterios de distancia presentan limitaciones, incluso cuando la clase minoritaria adquiere mayor representación dentro del conjunto de datos.

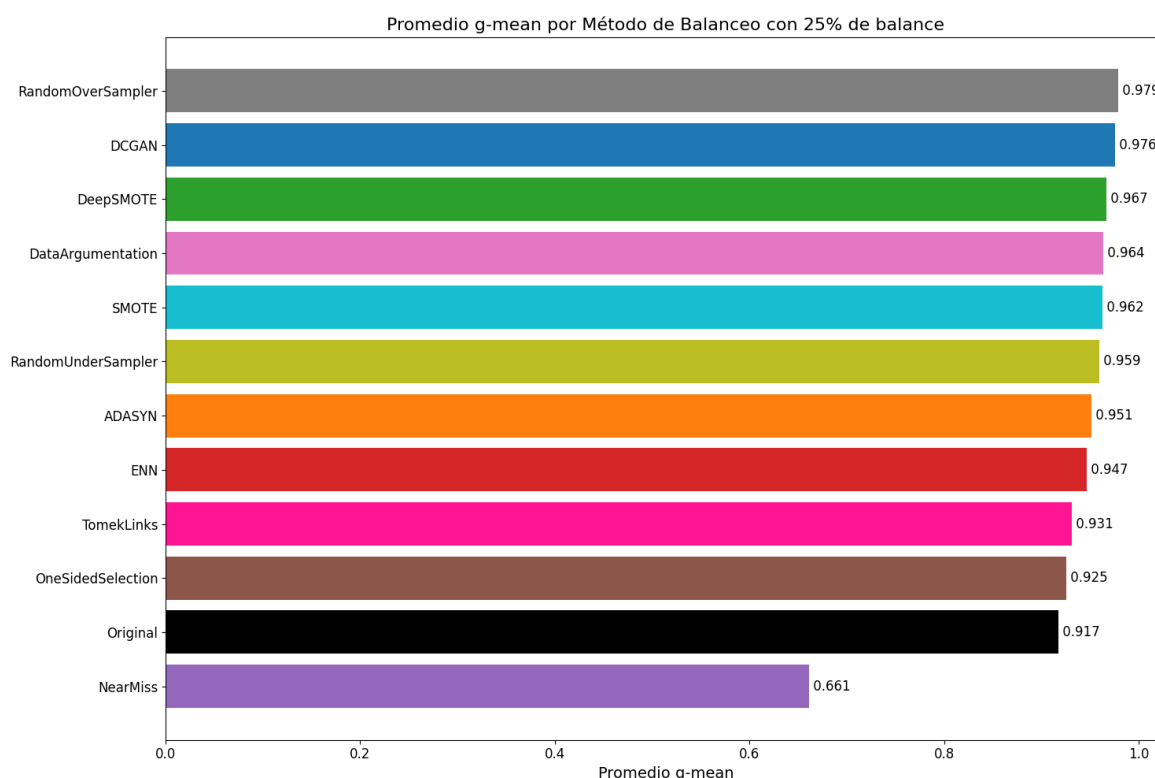


Figura 5.4: Comparación del *g-mean* promedio de todos los métodos de balanceo en un escenario de 25 % de balance.

En los experimentos donde se equilibra el conjunto de entrenamiento al 50 %, según lo reflejado en la Figura 5.5, se manifiesta una notoria convergencia en los valores de *g-mean* promedio de los distintos métodos empleados. *ROS* y *OSS* comparten la primera posición (0.984), confirmando que las estrategias híbridas y el simple sobremuestreo pueden lograr rendimientos equivalentes cuando la distribución de clases se acerca a la paridad en escenarios de balance considerable. Asimismo, *Data Augmentation* y *DCGAN* muestran un desempeño muy similar (0.981), lo que indica que, conforme se reduce la disparidad de clases, las diferencias entre las técnicas tienden a diluirse. Por último, *Near Miss* obtiene un incremento considerable (0.892) respecto a configuraciones previas, lo que sugiere que las desventajas de este enfoque basado en distancia se aminoran cuando la clase minoritaria se acerca en cantidad a la mayoritaria.

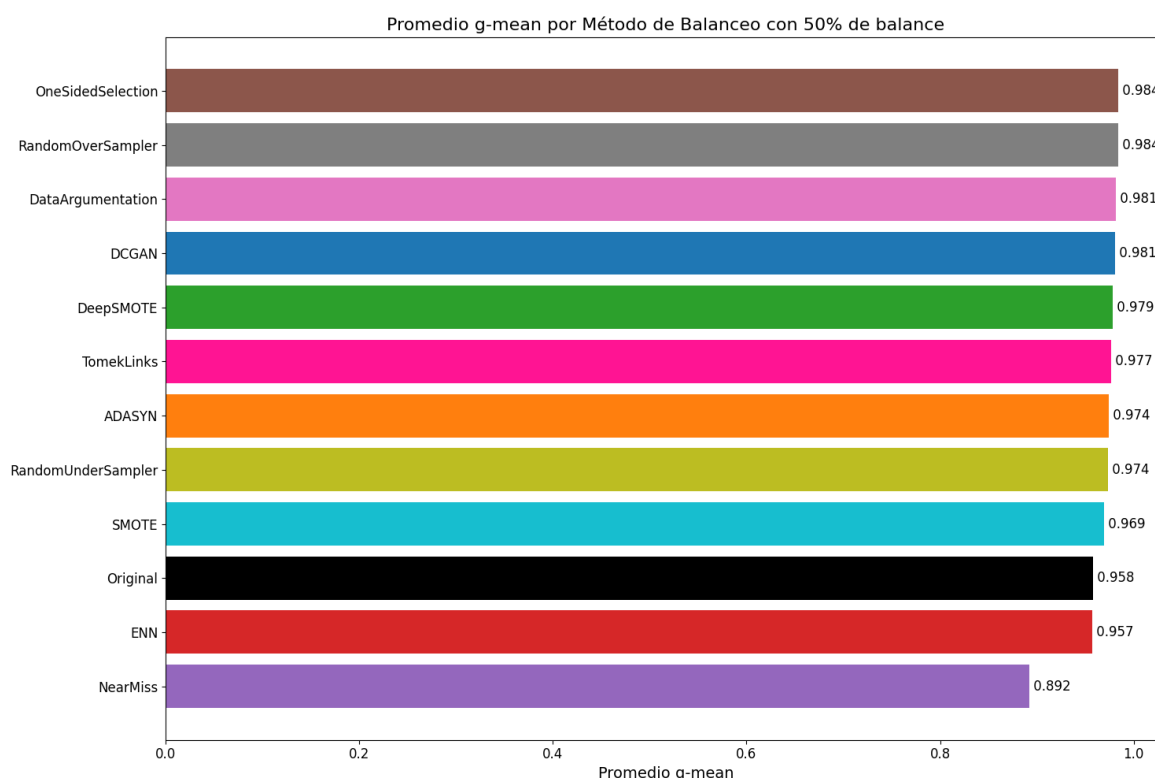


Figura 5.5: Comparación del *g-mean* promedio de todos los métodos de balanceo en un escenario de 50 % de balance.

Finalmente, para una distribución aún más favorable, con el 75 % de la clase minoritaria, todos los métodos evaluados registran valores de *g-mean* promedio superiores o iguales a 0.96. De acuerdo con la Figura 5.6, las técnicas de modelado generativo como *DCGAN* (0.989) y *DeepSMOTE* (0.988) alcanzan los mejores índices, aunque la diferencia respecto a métodos tradicionales como *ROS* (0.987) y *Data Augmentation* (0.987) es mínima. Incluso *Near Miss* logra valores aceptables (0.960), lo que confirma que, a mayor equilibrio en la distribución de clases, las brechas de rendimiento entre los diferentes métodos de balanceo se atenúan. Esto implica que, bajo estas condiciones, la elección de la metodología específica pierde relevancia comparada con escenarios de mayor desbalance.

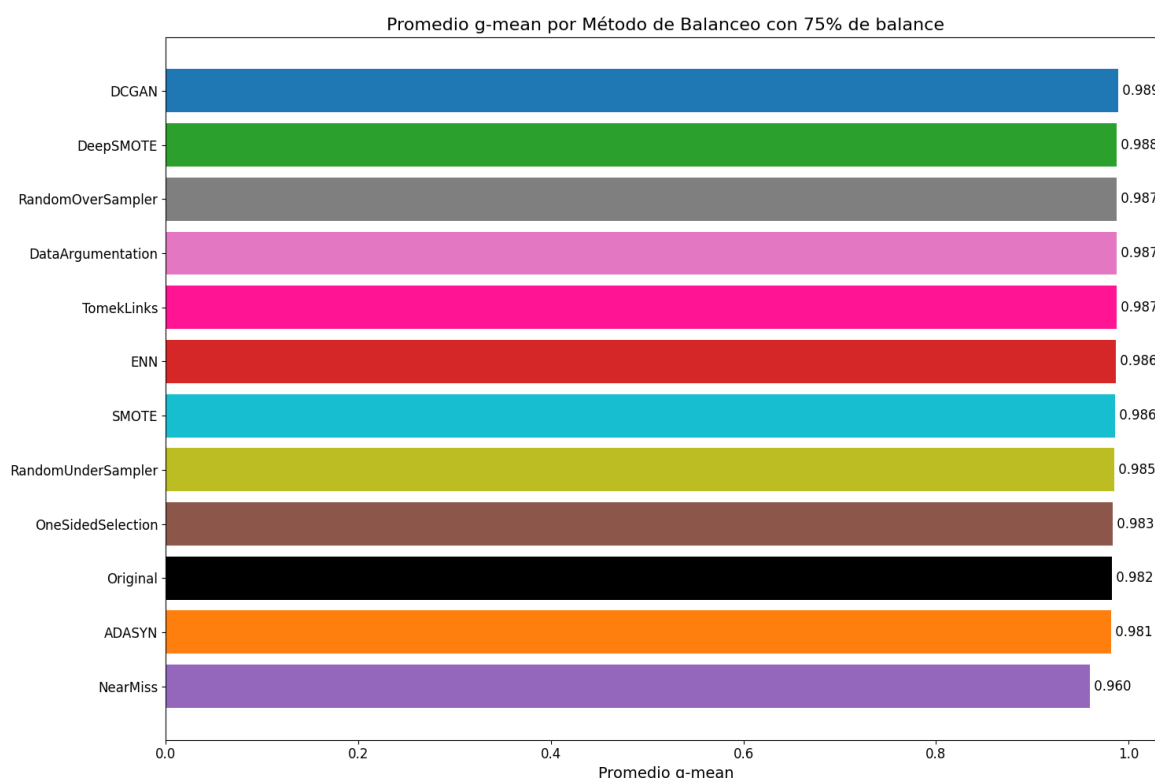


Figura 5.6: Comparación del *g-mean* promedio de todos los métodos de balanceo en un escenario de 75 % de balance.

Dado que el volumen de resultados es extenso y complejo para analizar de manera directa, se opta por aplicar el *Test de Friedman* para evaluar estadísticamente la efectividad de las técnicas de balanceo en distintos escenarios de desbalance. Este *test* no paramétrico es útil para comparar múltiples métodos en diferentes condiciones experimentales, como es el caso de los escenarios de desbalance considerados en este estudio. A diferencia de otros *tests*, el *Test de Friedman* es particularmente adecuado para datos de tipo ordinal y para casos en los que las condiciones de normalidad no se cumplen. Los resultados de este *test* se muestran en las tablas 3, 4, 5, 6, 7, 8.

5.2. Discusión

En la Tabla 5.1 se presentan, de forma resumida, los resultados obtenidos mediante los *tests de Friedman*, que muestran las técnicas de balanceo con mejor desempeño en cada escenario de desbalance. Este análisis sugiere que la elección de la técnica de balanceo debe adaptarse al grado de desbalance del conjunto de datos para optimizar la precisión y sensibilidad del modelo.

Por ejemplo, *ROS* es la técnica más eficaz en escenarios de desbalance extremo (1 %, 5 %, 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 %, 95 %, 99 %).

10 %, y 25 %), donde la clase minoritaria tiene muy poca representación. Por otro lado, OSS sobresale en el escenario de 50 % de balance, ya que combina la eliminación de instancias de la clase mayoritaria con la selección de ejemplos representativos. Este enfoque resulta útil en situaciones de desbalance moderado, pues reduce el sesgo hacia la clase mayoritaria al tiempo que mantiene una buena diversidad en los datos. Finalmente, *TL* es la técnica que obtiene mejores resultados en el escenario de 75 %, donde las clases están relativamente equilibradas. En este caso, la eliminación de pares de muestras de clases distintas que se encuentran muy cercanas entre sí ayuda a mejorar la precisión del modelo en contextos de balance cercano, al refinar la frontera de decisión entre clases.

Tabla 5.1: Resumen de Resultados de *Tests* de Friedman por escenario de balance.

Escenario de Balance	Mejor Técnica
1 %	ROS
5 %	ROS
10 %	ROS
25 %	ROS
50 %	OSS
75	TL

Capítulo 6

Conclusiones y recomendaciones

6.1. Conclusiones

El estudio realizado demuestra que el desbalance de clases en conjuntos de datos de clasificación afecta negativamente el desempeño de los modelos de redes neuronales, reduciendo su capacidad para identificar con precisión los casos menos representados. La aplicación de técnicas de balanceo, tanto de generación de datos sintéticos como de modificación de la distribución de clases, permite mejorar notablemente la sensibilidad del modelo hacia la clase minoritaria. En este contexto, las metodologías basadas en métodos tradicionales de balanceo y la generación de imágenes sintéticas muestran ventajas significativas en términos de precisión y generalización, especialmente cuando se trabaja con conjuntos de datos altamente desbalanceados.

Finalmente, la evaluación de diferentes técnicas también evidenció que no existe una solución única que funcione óptimamente para todos los escenarios de desbalance. Cada método tiene limitaciones y beneficios específicos que dependen del contexto del problema y de las características particulares de los datos. Esta investigación, por lo tanto, subraya la importancia de seleccionar el método de balanceo adecuado según el grado de desbalance y la naturaleza del conjunto de datos.

6.2. Recomendaciones

Para investigaciones futuras, se recomienda primeramente llevar a cabo estudios comparativos con otros modelos de redes neuronales para la clasificación de imágenes, por ejemplo; *MobileNet*, *VGG-16*, *VGG-19*, *Inception*, *EfficientNet*, incluso otras variantes de *ResNet*. Esto es con la finalidad de que se realice un enfoque comparativo del impacto del desbalance de clases en la clasificación de imágenes que retraten violencia física, en función a los modelos a implementar. Adicionalmente a esto, se recomienda realizar un análisis de sensibilidad de los

modelos con la finalidad de encontrar una mejor configuración de hiperparámetros θ , que conlleve a mejores resultados en la clasificación. En adición a lo mencionado anteriormente, se sugiere abordar el problema del desbalance implementando otras arquitecturas más sofisticadas como lo son los *transformers* para la clasificación de imágenes, en esta investigación no fue posible implementar por falta de la infraestructura requerida para correr dicho modelo.

Adicionalmente, se podrían implementar otros modelos generativos para mitigar el desbalance en el número de casos de la clase minoritaria. En esta investigación sólo se implementaron algunos de los modelos generativos más populares para el desbalance de clases como lo son; *DeepSMOTE* y *DCGAN*, pero aún queda un área de oportunidad extensa donde se analicen otras *GAN's*, inclusive también se pueda hacer una variación en sus parámetros para refinar la calidad de las imágenes generadas.

Finalmente, con base en la experimentación realizada en este trabajo de investigación, se recomienda abordar el desbalance de clases de manera más eficiente aplicando primero, en el espacio de características, algún método de sobremuestreo como *ROS* para equilibrar las clases. En segundo lugar, se sugiere aplicar técnicas de selección como *Tomek Links* o *ENN* con el fin de eliminar muestras ruidosas. Este proceso mejoraría la clasificación del modelo implementado, ya que combinaría las ventajas del sobremuestreo y el submuestreo de forma equilibrada.

Todas estas recomendaciones podrían llevar a futuras investigaciones relacionadas, tener mayor impacto científico y una mejor aportación al estado del arte.

Referencias

- Abhishek, K., y Abdelaziz, M. (2023). *Machine learning for imbalanced data: Tackle imbalanced datasets using machine learning and deep learning techniques*. Packt Publishing. Descargado de <https://imbalanceddata.com/>
- Afzal, S., Maqsood, M., Nazir, F., Khan, U., Aadil, F., Awan, K. M., ... Song, O.-Y. (2019). A data augmentation-based framework to handle class imbalance problem for alzheimers stage detection. *IEEE Access*, 7, 115528-115539. doi: 10.1109/ACCESS.2019.2932786
- Alejo, R., Rendón, E., y González-Bárcenas, V. (2020, 10). Procesamiento de bases de datos escolares por medio de redes neuronales artificiales. *CIENCIA ergo sum*, 27. doi: 10.30878/ces.v27n3a11
- Alejo Eleuterio, R. (2010). *Análisis del error en redes neuronales: Corrección de los datos y distribuciones no balanceadas*. Universitat Jaume I.
- Aremu, T., Zhiyuan, L., Alameeri, R., Khan, M., y Saddik, A. E. (2024). Ssvd-net: A novel salient super image classification and detection technique for weaponized violence. En K. Arai (Ed.), *Intelligent computing* (pp. 16–35). Cham: Springer Nature Switzerland.
- Arroyo, R., y Valladares, L. (2009). Derechos humanos y violencia sexual contra las mujeres. *Ramiro Ávila; Judith Salgado; Lola Valladares (comps.), El género en el derecho. Ensayos críticos*. Quito: Ministerio de Justicia y Derechos Humanos, 397–464.
- Bianculli, M., Falcionelli, N., Sernani, P., Tomassini, S., Contardo, P., Lombardi, M., y Dragoni, A. F. (2020). A dataset for automatic violence detection in videos. *Data in Brief*, 33, 106587. Descargado de <https://www.sciencedirect.com/science/article/pii/S2352340920314682> doi: <https://doi.org/10.1016/j.dib.2020.106587>
- Chen, W., Su, L., Chen, X., y Huang, Z. (2023). Rock image classification using deep residual neural network with transfer learning. *Frontiers in Earth Science*, 10. Descargado de <https://www.frontiersin.org/articles/10.3389/feart.2022.1079447> doi: 10.3389/feart.2022.1079447
- Cheng, M., Cai, K., y Li, M. (2021). Rwf-2000: An open large scale video database for violence detection. En *2020 25th international conference on pattern recognition (icpr)* (p. 4183-4190). doi: 10.1109/ICPR48806.2021.9412502
- Chávez-Bosquez, O., Vasquez, M., Hernandez, J., y Hernández-Ocaña, B. (2020, 12). Impacto de los algoritmos de sobremuestreo en la clasificación de subtipos principales del síndrome de guillain-barré. *Ingenius*, 20-31. doi: 10.17163/ings.n25.2021.02
- Cisneros, J., PAREDES, M., y Cevallos Uve, G. (2024, 11). I symposium "science, technology and innovation". *Revista de Investigación Científica TSE DE*, 7(L9), Pág. 1 138. doi: 10.60100/tsede.v7iL9.218
- Dablain, D., Krawczyk, B., y Chawla, N. V. (2023). Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9), 6390-6404. doi: 10.1109/TNNLS.2021.3136503
- Degardin, B., y Proença, H. (2020). Human activity analysis: Iterative weak/self-supervised learning frameworks for detecting abnormal events. En *2020 ieee international joint conference on biometrics (ijcb)* (p. 1-7). doi: 10.1109/IJCB48548.2020.9304905

- Foster, D. (2024). *Deep Learning Generativo: Enseñar a las máquinas a pintar, escribir, componer y jugar*. España: ANAYA MULTIMEDIA.
- García Abad, J. (2021). *Comparativa de técnicas de balanceo de datos. aplicación a un caso real para la predicción de fuga de clientes* (Trabajo Fin de Máster, Universidad de Oviedo). Descargado de <https://digibuo.uniovi.es/dspace/handle/10651/60629>
- García Jiménez, V. (2010, 02). Distribuciones de clases no balanceadas: Métricas, análisis de complejidad y algoritmos de aprendizaje. *También depositat en: https://www.educacion.es/teseo/mostrarRef.do?ref=859311 tesis en Teseo.*
- Guevara Cruz, R., y Augusto Delrieux, C. (2023, jul.). Aplicación de redes neuronales densas y convolucionales para detección de covid_19 en imágenes de rayos x. *CONNECTIVIDAD*, 4(2), 19-32. Descargado de <https://revista.ister.edu.ec/ojs/index.php/ISTER/article/view/78> doi: 10.37431/conectividad.v4i2.78
- Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). Pearson. Descargado de <https://www.pearson.com/en-us/subject-catalog/p/neural-networks-and-learning-machines/P200000003278/9780133002553>
- He, H., y Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284. doi: 10.1109/TKDE.2008.239
- He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *2016 IEEE conference on computer vision and pattern recognition (cvpr)* (p. 770-778). doi: 10.1109/CVPR.2016.90
- Inglés Muñoz, G. (2023). *Técnicas de aprendizaje profundo para la detección automática de objetos en imágenes sonar* [Trabajo Fin de Grado]. Cartagena, España. Descargado de <https://repositorio.upct.es/entities/publication/f8c08fe3-d73c-4900-8833-252ac459a6c5> (Dirigido por [Nombre del director, si lo conoces])
- Jalali Mojahed, A., Moattar, M. H., y Ghaffari, H. (2024). Supervised density-based metric learning based on bhattacharya distance for imbalanced data classification problems. *Big Data and Cognitive Computing*, 8(9). Descargado de <https://www.mdpi.com/2504-2289/8/9/109> doi: 10.3390/bdcc8090109
- Klevens, J. (2001, 02). Violencia física contra la mujer en santa fe de bogotá: prevalencia y factores asociados. *Revista Panamericana de Salud Pública*, 9. doi: 10.1590/S1020-49892001000200004
- Kubat, M., y Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. *Proceedings of the Fourteenth International Conference on Machine Learning*, 179-186.
- Ljubic, H., Martinovic, G., y Volaric, T. (2022). Augmenting data with generative adversarial networks: An overview. *Intelligent Data Analysis*, 26(3), 361-378.
- López, O. C. (2017). La violencia económica y/o patrimonial contra las mujeres en el ámbito familiar. *Persona y familia*, 1(6), 39-58.
- Magnolia, A. G. A. (2023). *Análisis acústico de la tos para detección de covid con énfasis en el balanceo de clases*. Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California.
- Mehta, P., Bukov, M., Wang, C.-H., Day, A. G., Richardson, C., Fisher, C. K., y Schwab, D. J. (2019, mayo). A high-bias, low-variance introduction to machine learning for physicists. *Physics*

- Reports, 810, 1124. Descargado de <http://dx.doi.org/10.1016/j.physrep.2019.03.001> doi: 10.1016/j.physrep.2019.03.001
- Morales, D. I. B., y Pisarro, W. S. T. (2022). *Análisis comparativo del rendimiento de algoritmos de clasificación binaria en un conjunto de datos desbalanceados* (B.S. thesis, Universidad Politécnica Salesiana). Descargado de <https://dspace.ups.edu.ec/handle/123456789/22246> (Repositorio Institucional de la Universidad Politécnica Salesiana)
- Ojer Baztán, C. (2020). *Implementación y estudio del rendimiento del clasificador ders-boost, para problemas con conjuntos de datos no balanceados*. Pamplona, España. Descargado de <https://academica-e.unavarra.es/handle/2454/39108> (Trabajo Fin de Grado dirigido por José Antonio Sanz Delgado)
- OMS. (2002). *Informe mundial sobre la violencia y la salud: resumen*. Washington, D.C.: Organización Panamericana de la Salud. Descargado de https://iris.who.int/bitstream/handle/10665/43431/9275324220_spa.pdf
- Pan American Health Organization. (2024). *Prevención de la violencia*. Descargado de <https://www.paho.org/es/temas/prevencion-violencia>
- Poggi, F. (2019, dic.). Sobre el concepto de violencia de género y su relevancia para el derecho. *DOXA. Cuadernos de Filosofía del Derecho*(42), 285307. Descargado de <https://doxa.ua.es/article/view/2019-n42-sobre-el-concepto-de-violencia-de-genero-y-su-relevanci> doi: 10.14198/DOXA2019.42.12
- Qamar, R., y Zardari, B. (2023, 08). Artificial neural networks: An overview. *Mesopotamian Journal of Computer Science*, 2023, 130-139. doi: 10.58496/MJCSC/2023/015
- Radford, A., Metz, L., y Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*. Descargado de <https://arxiv.org/abs/1511.06434> doi: 10.48550/arXiv.1511.06434
- Raschka, S., Liu, Y. H., y Mirjalili, V. (2022). *Machine learning with pytorch and scikit-learn: Develop machine learning and deep learning models with python*. Packt Publishing. Descargado de <https://www.amazon.com/Machine-Learning-PyTorch-Scikit-Learn-learning-ebook/dp/B09NW48MR1>
- Rendón, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J., y Granda-Gutiérrez, E. E. (2020). Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, 10(4). Descargado de <https://www.mdpi.com/2076-3417/10/4/1276> doi: 10.3390/app10041276
- Safranoff, A. (2017). Violencia psicológica hacia la mujer: ¿cuáles son los factores que aumentan el riesgo de que exista esta forma de maltrato en la pareja? *Salud colectiva*, 13, 611-632.
- Soliman, M. M., Kamal, M. H., El-Massih Nashed, M. A., Mostafa, Y. M., Chawky, B. S., y Khattab, D. (2019). Violence recognition from videos using deep learning techniques. En *2019 ninth international conference on intelligent computing and information systems (icicis)* (p. 80-85). doi: 10.1109/ICICIS46948.2019.9014714
- Solís Fonseca, C. F. (2020). *Efecto del balanceo de clases al evaluar el f1-score para un clasificador de texto en análisis de sentimiento* (Tesis de Master, Universidad de Costa Rica, San José, Costa Rica). Descargado de <https://hdl.handle.net/10669/82700> (Tesis de Maestría)

Apéndice A

Resultados de Experimentos

Tabla 1: Resultados del modelo en diferentes métodos de balanceo.

Dataset	Balance	Método Balance	Accuracy	Precision	Recall	F1 Score	G-Mean
airtlab	1	ADASYN	75.800	1.000	0.516	0.681	0.718
airtlab	1	DataAugmentation	80.730	1.000	0.610	0.761	0.780
airtlab	1	DCGAN	76.820	1.000	0.536	0.698	0.732
airtlab	1	DeepSMOTE	75.290	1.000	0.505	0.671	0.711
airtlab	1	ENN	80.470	1.000	0.610	0.757	0.781
airtlab	1	NearMiss	50.000	0.500	1.000	0.667	0.000
airtlab	1	OneSidedSelection	79.260	1.000	0.585	0.738	0.765
airtlab	1	Original	71.250	1.000	0.425	0.597	0.652
airtlab	1	RandomOverSampler	69.850	1.000	0.397	0.568	0.630
airtlab	1	RandomUnderSampler	50.060	0.556	0.006	0.013	0.080
airtlab	1	SMOTE	75.030	1.000	0.501	0.667	0.708
airtlab	1	TomekLinks	87.960	0.998	0.761	0.863	0.872
airtlab	5	ADASYN	89.500	1.000	0.790	0.883	0.889
airtlab	5	DataAugmentation	92.250	1.000	0.840	0.916	0.919
airtlab	5	DCGAN	91.610	1.000	0.832	0.908	0.912
airtlab	5	DeepSMOTE	94.110	1.000	0.882	0.937	0.939
airtlab	5	ENN	93.280	1.000	0.866	0.928	0.930
airtlab	5	NearMiss	64.790	0.595	0.924	0.724	0.586
airtlab	5	OneSidedSelection	92.380	0.961	0.884	0.921	0.923
airtlab	5	Original	90.970	1.000	0.820	0.901	0.905
airtlab	5	RandomOverSampler	94.490	1.000	0.890	0.942	0.943
airtlab	5	RandomUnderSampler	82.010	0.948	0.677	0.790	0.808
airtlab	5	SMOTE	92.960	1.000	0.859	0.924	0.927
airtlab	5	TomekLinks	93.980	1.000	0.880	0.936	0.938
airtlab	10	ADASYN	98.210	1.000	0.964	0.982	0.982
airtlab	10	DataAugmentation	96.410	1.000	0.928	0.962	0.963
airtlab	10	DCGAN	97.760	1.000	0.955	0.977	0.977
airtlab	10	DeepSMOTE	97.120	1.000	0.942	0.970	0.970
airtlab	10	ENN	97.950	1.000	0.959	0.979	0.979
airtlab	10	NearMiss	62.680	0.576	0.965	0.721	0.527
airtlab	10	OneSidedSelection	87.840	0.838	0.937	0.885	0.876
airtlab	10	Original	95.770	1.000	0.916	0.956	0.957
airtlab	10	RandomOverSampler	97.820	1.000	0.956	0.978	0.978
airtlab	10	RandomUnderSampler	95.770	0.964	0.951	0.958	0.958
airtlab	10	SMOTE	97.820	1.000	0.956	0.978	0.978
airtlab	10	TomekLinks	96.030	1.000	0.921	0.959	0.960
airtlab	25	ADASYN	96.930	1.000	0.938	0.968	0.969
airtlab	25	DataAugmentation	97.820	1.000	0.956	0.977	0.977
airtlab	25	DCGAN	98.210	1.000	0.964	0.981	0.981
airtlab	25	DeepSMOTE	98.720	0.998	0.975	0.987	0.987
airtlab	25	ENN	97.120	0.999	0.944	0.970	0.971
airtlab	25	NearMiss	64.980	0.588	0.996	0.740	0.550
airtlab	25	OneSidedSelection	98.660	1.000	0.973	0.986	0.986
airtlab	25	Original	97.630	1.000	0.953	0.976	0.976
airtlab	25	RandomOverSampler	98.850	1.000	0.977	0.988	0.988
airtlab	25	RandomUnderSampler	98.020	0.999	0.962	0.980	0.980
airtlab	25	SMOTE	98.660	1.000	0.973	0.986	0.986
airtlab	25	TomekLinks	98.270	1.000	0.965	0.982	0.983
airtlab	50	ADASYN	98.980	0.999	0.981	0.990	0.990
airtlab	50	DataAugmentation	98.590	1.000	0.971	0.985	0.985
airtlab	50	DCGAN	98.590	1.000	0.971	0.985	0.985
airtlab	50	DeepSMOTE	99.170	0.996	0.987	0.991	0.991

Continúa en la siguiente página

Dataset	Balance	Método Balance	Accuracy	Precision	Recall	F1 Score	G-Mean
airtlab	50	ENN	94.110	0.901	0.991	0.944	0.940
airtlab	50	NearMiss	97.500	0.956	0.996	0.976	0.975
airtlab	50	OneSidedSelection	98.910	0.999	0.980	0.989	0.989
airtlab	50	Original	98.720	1.000	0.974	0.987	0.987
airtlab	50	RandomOverSampler	98.980	0.999	0.981	0.990	0.990
airtlab	50	RandomUnderSampler	99.100	1.000	0.982	0.991	0.991
airtlab	50	SMOTE	99.300	1.000	0.986	0.993	0.993
airtlab	50	TomekLinks	99.360	1.000	0.987	0.994	0.994
airtlab	75	ADASYN	99.620	1.000	0.992	0.996	0.996
airtlab	75	DataAugmentation	99.490	1.000	0.989	0.994	0.994
airtlab	75	DCGAN	99.620	1.000	0.992	0.996	0.996
airtlab	75	DeepSMOTE	99.490	0.998	0.991	0.994	0.994
airtlab	75	ENN	99.230	0.999	0.986	0.992	0.992
airtlab	75	NearMiss	99.040	0.991	0.990	0.990	0.990
airtlab	75	OneSidedSelection	99.170	1.000	0.983	0.992	0.992
airtlab	75	Original	99.550	0.999	0.992	0.996	0.996
airtlab	75	RandomOverSampler	99.680	1.000	0.994	0.997	0.997
airtlab	75	RandomUnderSampler	99.620	1.000	0.992	0.996	0.996
airtlab	75	SMOTE	99.620	0.999	0.994	0.996	0.996
airtlab	75	TomekLinks	99.680	0.999	0.995	0.997	0.997
airtlab	100	Original	99.550	1.000	0.991	0.996	0.996
rlvsd	1	ADASYN	86.010	0.999	0.720	0.837	0.848
rlvsd	1	DataAugmentation	88.400	0.998	0.769	0.869	0.876
rlvsd	1	DCGAN	84.960	0.998	0.700	0.823	0.836
rlvsd	1	DeepSMOTE	74.040	1.000	0.480	0.649	0.693
rlvsd	1	ENN	78.950	0.996	0.580	0.734	0.761
rlvsd	1	NearMiss	60.930	0.561	0.998	0.718	0.469
rlvsd	1	OneSidedSelection	84.340	0.986	0.696	0.816	0.830
rlvsd	1	Original	58.060	1.000	0.161	0.277	0.401
rlvsd	1	RandomOverSampler	88.990	0.998	0.781	0.876	0.883
rlvsd	1	RandomUnderSampler	50.000	0.000	0.000	0.000	0.000
rlvsd	1	SMOTE	83.100	0.998	0.663	0.796	0.813
rlvsd	1	TomekLinks	90.000	0.996	0.802	0.889	0.894
rlvsd	5	ADASYN	95.070	0.992	0.908	0.948	0.949
rlvsd	5	DataAugmentation	95.360	0.999	0.907	0.951	0.952
rlvsd	5	DCGAN	92.530	0.985	0.863	0.920	0.923
rlvsd	5	DeepSMOTE	83.510	0.999	0.670	0.802	0.818
rlvsd	5	ENN	82.430	0.981	0.661	0.790	0.807
rlvsd	5	NearMiss	60.930	0.561	0.997	0.718	0.469
rlvsd	5	OneSidedSelection	89.190	0.984	0.796	0.880	0.886
rlvsd	5	Original	65.570	0.975	0.319	0.481	0.562
rlvsd	5	RandomOverSampler	96.720	0.997	0.937	0.966	0.966
rlvsd	5	RandomUnderSampler	72.800	0.977	0.466	0.631	0.679
rlvsd	5	SMOTE	91.000	0.998	0.821	0.901	0.905
rlvsd	5	TomekLinks	97.440	0.970	0.978	0.974	0.974
rlvsd	10	ADASYN	74.250	0.997	0.486	0.653	0.696
rlvsd	10	DataAugmentation	96.520	0.998	0.931	0.964	0.964
rlvsd	10	DCGAN	96.500	0.995	0.935	0.964	0.965
rlvsd	10	DeepSMOTE	91.860	0.966	0.867	0.914	0.917
rlvsd	10	ENN	87.970	0.967	0.785	0.867	0.874
rlvsd	10	NearMiss	62.300	0.570	0.997	0.725	0.498
rlvsd	10	OneSidedSelection	72.580	0.999	0.452	0.622	0.672
rlvsd	10	Original	79.830	0.996	0.598	0.748	0.772
rlvsd	10	RandomOverSampler	97.960	0.980	0.978	0.979	0.979
rlvsd	10	RandomUnderSampler	95.950	0.957	0.961	0.959	0.959
rlvsd	10	SMOTE	94.160	0.994	0.888	0.938	0.940
rlvsd	10	TomekLinks	93.440	0.952	0.913	0.932	0.934
rlvsd	25	ADASYN	93.910	0.985	0.891	0.936	0.937
rlvsd	25	DataAugmentation	98.170	0.998	0.964	0.981	0.981
rlvsd	25	DCGAN	98.290	0.997	0.969	0.983	0.983
rlvsd	25	DeepSMOTE	97.310	0.993	0.953	0.972	0.972
rlvsd	25	ENN	93.650	0.985	0.885	0.933	0.935
rlvsd	25	NearMiss	66.120	0.596	0.999	0.746	0.568
rlvsd	25	OneSidedSelection	84.100	0.770	0.972	0.859	0.830
rlvsd	25	Original	85.160	0.967	0.727	0.830	0.842
rlvsd	25	RandomOverSampler	98.660	0.987	0.985	0.987	0.986
rlvsd	25	RandomUnderSampler	98.010	0.973	0.987	0.980	0.980
rlvsd	25	SMOTE	97.390	0.995	0.951	0.973	0.973
rlvsd	25	TomekLinks	81.900	0.981	0.650	0.782	0.801
rlvsd	50	ADASYN	97.800	0.965	0.990	0.978	0.977
rlvsd	50	DataAugmentation	98.620	0.997	0.975	0.986	0.986
rlvsd	50	DCGAN	98.820	0.996	0.979	0.988	0.988
rlvsd	50	DeepSMOTE	98.310	0.968	0.998	0.983	0.982
rlvsd	50	ENN	91.630	0.867	0.982	0.921	0.913
rlvsd	50	NearMiss	87.340	0.800	0.994	0.887	0.865
rlvsd	50	OneSidedSelection	98.550	0.984	0.986	0.985	0.985
rlvsd	50	Original	95.180	0.932	0.974	0.952	0.951
rlvsd	50	RandomOverSampler	99.350	0.996	0.990	0.993	0.993
rlvsd	50	RandomUnderSampler	97.880	0.982	0.975	0.978	0.978
rlvsd	50	SMOTE	97.780	0.975	0.979	0.977	0.977
rlvsd	50	TomekLinks	97.410	0.960	0.988	0.974	0.973
rlvsd	75	ADASYN	98.880	0.996	0.981	0.988	0.988

Continúa en la siguiente página

Dataset	Balance	Método Balance	Accuracy	Precision	Recall	F1 Score	G-Mean
rlvsd	75	DataAugmentation	99.840	0.998	0.998	0.998	0.998
rlvsd	75	DCGAN	99.840	0.998	0.998	0.998	0.998
rlvsd	75	DeepSMOTE	99.880	0.997	1.000	0.998	0.998
rlvsd	75	ENN	98.760	0.979	0.996	0.987	0.987
rlvsd	75	NearMiss	97.980	0.983	0.975	0.979	0.979
rlvsd	75	OneSidedSelection	98.580	0.982	0.989	0.985	0.985
rlvsd	75	Original	98.290	0.967	0.999	0.983	0.982
rlvsd	75	RandomOverSampler	98.920	0.980	0.999	0.989	0.989
rlvsd	75	RandomUnderSampler	98.490	0.972	0.998	0.985	0.984
rlvsd	75	SMOTE	98.390	0.969	0.999	0.984	0.983
rlvsd	75	TomekLinks	98.900	0.984	0.984	0.988	0.988
rlvsd	100	Original	99.860	0.997	1.000	0.998	0.998
rwf2000	1	ADASYN	56.480	0.978	0.133	0.234	0.364
rwf2000	1	DataAugmentation	50.680	1.000	0.014	0.026	0.116
rwf2000	1	DCGAN	54.150	0.933	0.089	0.163	0.297
rwf2000	1	DeepSMOTE	51.070	1.000	0.021	0.041	0.146
rwf2000	1	ENN	52.320	0.993	0.047	0.089	0.216
rwf2000	1	NearMiss	50.200	0.501	0.998	0.667	0.075
rwf2000	1	OneSidedSelection	50.020	1.000	0.000	0.001	0.018
rwf2000	1	Original	50.000	0.000	0.000	0.000	0.000
rwf2000	1	RandomOverSampler	61.980	0.974	0.246	0.393	0.495
rwf2000	1	RandomUnderSampler	65.870	0.673	0.618	0.644	0.657
rwf2000	1	SMOTE	54.280	0.996	0.086	0.158	0.293
rwf2000	1	TomekLinks	50.000	0.000	0.000	0.000	0.000
rwf2000	5	ADASYN	75.500	0.964	0.530	0.684	0.721
rwf2000	5	DataAugmentation	70.800	0.985	0.422	0.591	0.647
rwf2000	5	DCGAN	69.550	0.968	0.404	0.570	0.631
rwf2000	5	DeepSMOTE	68.030	0.981	0.367	0.604	0.604
rwf2000	5	ENN	68.750	0.966	0.389	0.554	0.619
rwf2000	5	NearMiss	51.070	0.506	0.949	0.660	0.263
rwf2000	5	OneSidedSelection	70.370	0.987	0.413	0.582	0.641
rwf2000	5	Original	61.130	0.990	0.225	0.367	0.474
rwf2000	5	RandomOverSampler	78.470	0.986	0.578	0.728	0.757
rwf2000	5	RandomUnderSampler	81.320	0.794	0.846	0.819	0.812
rwf2000	5	SMOTE	71.620	0.992	0.436	0.606	0.659
rwf2000	5	TomekLinks	77.080	0.960	0.565	0.711	0.743
rwf2000	10	ADASYN	81.930	0.984	0.649	0.782	0.801
rwf2000	10	DataAugmentation	75.380	0.989	0.513	0.676	0.714
rwf2000	10	DCGAN	85.400	0.899	0.797	0.845	0.852
rwf2000	10	DeepSMOTE	85.570	0.944	0.755	0.839	0.849
rwf2000	10	ENN	81.870	0.985	0.647	0.781	0.800
rwf2000	10	NearMiss	53.270	0.518	0.939	0.668	0.344
rwf2000	10	OneSidedSelection	87.020	0.960	0.773	0.856	0.865
rwf2000	10	Original	70.200	0.992	0.407	0.578	0.637
rwf2000	10	RandomOverSampler	88.120	0.986	0.773	0.867	0.874
rwf2000	10	RandomUnderSampler	84.480	0.878	0.801	0.838	0.844
rwf2000	10	SMOTE	85.380	0.988	0.716	0.830	0.843
rwf2000	10	TomekLinks	88.030	0.916	0.837	0.875	0.879
rwf2000	25	ADASYN	88.300	0.988	0.776	0.869	0.876
rwf2000	25	DataAugmentation	88.750	0.987	0.784	0.874	0.881
rwf2000	25	DCGAN	92.450	0.925	0.923	0.924	0.924
rwf2000	25	DeepSMOTE	90.830	0.960	0.851	0.902	0.906
rwf2000	25	ENN	90.920	0.971	0.844	0.903	0.907
rwf2000	25	NearMiss	78.520	0.734	0.895	0.806	0.777
rwf2000	25	OneSidedSelection	92.370	0.956	0.888	0.921	0.923
rwf2000	25	Original	86.930	0.982	0.752	0.852	0.861
rwf2000	25	RandomOverSampler	93.350	0.959	0.906	0.932	0.933
rwf2000	25	RandomUnderSampler	86.730	0.945	0.780	0.855	0.863
rwf2000	25	SMOTE	91.750	0.976	0.856	0.912	0.915
rwf2000	25	TomekLinks	90.750	0.981	0.831	0.900	0.904
rwf2000	50	ADASYN	94.670	0.965	0.927	0.946	0.946
rwf2000	50	DataAugmentation	94.780	0.951	0.944	0.947	0.947
rwf2000	50	DCGAN	94.730	0.969	0.923	0.946	0.947
rwf2000	50	DeepSMOTE	94.150	0.979	0.901	0.939	0.940
rwf2000	50	ENN	94.320	0.952	0.933	0.943	0.943
rwf2000	50	NearMiss	88.950	0.863	0.926	0.893	0.889
rwf2000	50	OneSidedSelection	95.750	0.964	0.950	0.957	0.958
rwf2000	50	Original	94.100	0.953	0.928	0.940	0.941
rwf2000	50	RandomOverSampler	96.080	0.978	0.943	0.960	0.961
rwf2000	50	RandomUnderSampler	93.020	0.952	0.906	0.928	0.930
rwf2000	50	SMOTE	95.620	0.963	0.949	0.956	0.956
rwf2000	50	TomekLinks	95.200	0.974	0.928	0.951	0.952
rwf2000	75	ADASYN	94.770	0.969	0.925	0.946	0.947
rwf2000	75	DataAugmentation	95.330	0.944	0.963	0.954	0.953
rwf2000	75	DCGAN	95.620	0.957	0.954	0.956	0.956
rwf2000	75	DeepSMOTE	95.080	0.949	0.952	0.951	0.950
rwf2000	75	ENN	95.820	0.954	0.963	0.958	0.958
rwf2000	75	NearMiss	91.050	0.957	0.859	0.906	0.909
rwf2000	75	OneSidedSelection	94.720	0.972	0.921	0.946	0.947
rwf2000	75	Original	94.880	0.974	0.922	0.947	0.948
rwf2000	75	RandomOverSampler	95.630	0.955	0.957	0.956	0.956
rwf2000	75	RandomUnderSampler	94.750	0.943	0.953	0.948	0.948

Continúa en la siguiente página

Dataset	Balance	Método Balance	Accuracy	Precision	Recall	F1 Score	G-Mean
rwf2000	75	SMOTE	95.500	0.969	0.940	0.954	0.955
rwf2000	75	TomekLinks	95.680	0.956	0.958	0.957	0.957
rwf2000	100	Original	96.120	0.978	0.943	0.960	0.961
scvd	1	ADASYN	82.230	1.000	0.645	0.784	0.803
scvd	1	DataAugmentation	89.880	1.000	0.797	0.887	0.893
scvd	1	DCGAN	86.360	1.000	0.727	0.842	0.852
scvd	1	DeepSMOTE	89.780	1.000	0.795	0.886	0.892
scvd	1	ENN	90.720	1.000	0.814	0.898	0.902
scvd	1	NearMiss	50.500	0.502	0.997	0.668	0.113
scvd	1	OneSidedSelection	72.780	1.000	0.456	0.626	0.675
scvd	1	Original	87.370	1.000	0.747	0.855	0.864
scvd	1	RandomOverSampler	88.960	1.000	0.779	0.876	0.883
scvd	1	RandomUnderSampler	90.310	0.953	0.848	0.897	0.901
scvd	1	SMOTE	90.580	0.999	0.812	0.896	0.901
scvd	1	TomekLinks	86.010	0.993	0.725	0.838	0.849
scvd	5	ADASYN	98.650	1.000	0.973	0.986	0.986
scvd	5	DataAugmentation	97.230	1.000	0.944	0.971	0.971
scvd	5	DCGAN	98.240	0.983	0.980	0.982	0.982
scvd	5	DeepSMOTE	98.050	0.993	0.966	0.980	0.980
scvd	5	ENN	97.520	1.000	0.950	0.975	0.975
scvd	5	NearMiss	54.030	0.521	0.997	0.684	0.289
scvd	5	OneSidedSelection	97.670	0.994	0.959	0.976	0.977
scvd	5	Original	97.550	1.000	0.951	0.975	0.975
scvd	5	RandomOverSampler	98.210	1.000	0.964	0.982	0.982
scvd	5	RandomUnderSampler	97.340	0.990	0.957	0.973	0.973
scvd	5	SMOTE	97.180	0.999	0.944	0.971	0.971
scvd	5	TomekLinks	98.660	1.000	0.973	0.986	0.986
scvd	10	ADASYN	99.590	0.999	0.993	0.996	0.996
scvd	10	DataAugmentation	98.680	1.000	0.973	0.986	0.986
scvd	10	DCGAN	98.740	0.996	0.978	0.987	0.987
scvd	10	DeepSMOTE	99.240	1.000	0.984	0.992	0.992
scvd	10	ENN	99.490	0.997	0.993	0.995	0.995
scvd	10	NearMiss	58.390	0.546	0.999	0.706	0.411
scvd	10	OneSidedSelection	98.280	0.981	0.985	0.983	0.983
scvd	10	Original	88.850	1.000	0.777	0.874	0.882
scvd	10	RandomOverSampler	99.740	1.000	0.995	0.997	0.997
scvd	10	RandomUnderSampler	99.350	1.000	0.987	0.993	0.993
scvd	10	SMOTE	98.960	1.000	0.979	0.990	0.990
scvd	10	TomekLinks	97.820	1.000	0.956	0.978	0.978
scvd	25	ADASYN	99.780	1.000	0.996	0.998	0.998
scvd	25	DataAugmentation	99.560	1.000	0.991	0.995	0.995
scvd	25	DCGAN	99.940	1.000	0.998	0.999	0.999
scvd	25	DeepSMOTE	99.610	0.998	0.993	0.996	0.996
scvd	25	ENN	99.860	1.000	0.997	0.999	0.999
scvd	25	NearMiss	65.670	0.593	0.998	0.744	0.561
scvd	25	OneSidedSelection	98.810	0.991	0.986	0.988	0.988
scvd	25	Original	96.920	1.000	0.938	0.968	0.969
scvd	25	RandomOverSampler	99.920	1.000	0.998	0.999	0.999
scvd	25	RandomUnderSampler	99.550	0.998	0.993	0.996	0.996
scvd	25	SMOTE	99.840	1.000	0.997	0.998	0.998
scvd	25	TomekLinks	99.580	1.000	0.992	0.996	0.996
scvd	50	ADASYN	99.800	1.000	0.996	0.998	0.998
scvd	50	DataAugmentation	99.290	1.000	0.985	0.992	0.992
scvd	50	DCGAN	99.890	1.000	0.997	0.998	0.998
scvd	50	DeepSMOTE	99.880	0.997	0.999	0.998	0.998
scvd	50	ENN	99.780	1.000	0.996	0.998	0.998
scvd	50	NearMiss	82.250	0.739	0.998	0.849	0.804
scvd	50	OneSidedSelection	99.930	1.000	0.999	0.999	0.999
scvd	50	Original	96.040	0.997	0.923	0.959	0.960
scvd	50	RandomOverSampler	99.850	1.000	0.996	0.998	0.998
scvd	50	RandomUnderSampler	99.370	0.990	0.997	0.994	0.994
scvd	50	SMOTE	99.860	1.000	0.998	0.999	0.999
scvd	50	TomekLinks	98.980	0.988	0.992	0.990	0.990
scvd	75	ADASYN	99.970	0.999	1.000	0.999	0.999
scvd	75	DataAugmentation	99.430	1.000	0.988	0.994	0.994
scvd	75	DCGAN	99.940	0.999	0.999	0.999	0.999
scvd	75	DeepSMOTE	99.810	1.000	0.996	0.998	0.998
scvd	75	ENN	99.980	1.000	0.999	0.999	0.999
scvd	75	NearMiss	94.220	0.897	0.999	0.945	0.940
scvd	75	OneSidedSelection	99.880	0.998	0.999	0.999	0.999
scvd	75	Original	99.390	0.990	0.998	0.994	0.994
scvd	75	RandomOverSampler	99.940	1.000	0.999	0.999	0.999
scvd	75	RandomUnderSampler	99.780	1.000	0.996	0.998	0.998
scvd	75	SMOTE	99.940	1.000	0.999	0.999	0.999
scvd	75	TomekLinks	99.940	0.999	1.000	0.999	0.999
scvd	100	Original	99.910	0.999	0.998	0.999	0.999
ubi-fights	1	ADASYN	62.330	0.990	0.249	0.398	0.498
ubi-fights	1	DataAugmentation	76.730	1.000	0.534	0.696	0.731
ubi-fights	1	DCGAN	75.870	1.000	0.517	0.681	0.719
ubi-fights	1	DeepSMOTE	79.500	0.997	0.591	0.742	0.768
ubi-fights	1	ENN	79.410	0.998	0.589	0.741	0.767
ubi-fights	1	NearMiss	50.000	0.000	0.000	0.000	0.000

Continúa en la siguiente página

Dataset	Balance	Método Balance	Accuracy	Precision	Recall	F1 Score	G-Mean
ubi-fights	1	OneSidedSelection	63.020	0.998	0.261	0.414	0.511
ubi-fights	1	Original	72.200	0.999	0.444	0.615	0.667
ubi-fights	1	RandomOverSampler	86.740	0.998	0.736	0.847	0.857
ubi-fights	1	RandomUnderSampler	69.480	0.967	0.403	0.569	0.631
ubi-fights	1	SMOTE	83.140	0.999	0.664	0.797	0.814
ubi-fights	1	TomekLinks	81.570	0.895	0.715	0.795	0.809
ubi-fights	5	ADASYN	78.010	0.999	0.561	0.718	0.749
ubi-fights	5	DataAugmentation	93.140	0.999	0.863	0.926	0.928
ubi-fights	5	DCGAN	94.940	0.999	0.899	0.946	0.948
ubi-fights	5	DeepSMOTE	95.340	0.991	0.915	0.951	0.952
ubi-fights	5	ENN	84.610	1.000	0.692	0.818	0.832
ubi-fights	5	NearMiss	56.720	0.537	0.987	0.695	0.381
ubi-fights	5	OneSidedSelection	93.520	0.989	0.881	0.931	0.934
ubi-fights	5	Original	81.030	0.999	0.622	0.766	0.788
ubi-fights	5	RandomOverSampler	96.290	0.995	0.930	0.962	0.962
ubi-fights	5	RandomUnderSampler	87.590	0.867	0.889	0.877	0.876
ubi-fights	5	SMOTE	84.640	0.999	0.693	0.819	0.832
ubi-fights	5	TomekLinks	88.430	0.988	0.778	0.871	0.878
ubi-fights	10	ADASYN	81.930	0.999	0.639	0.780	0.799
ubi-fights	10	DataAugmentation	96.610	0.999	0.932	0.964	0.965
ubi-fights	10	DCGAN	97.480	0.999	0.949	0.974	0.974
ubi-fights	10	DeepSMOTE	97.800	0.998	0.957	0.977	0.977
ubi-fights	10	ENN	90.640	0.998	0.814	0.897	0.902
ubi-fights	10	NearMiss	74.240	0.666	0.972	0.791	0.706
ubi-fights	10	OneSidedSelection	91.870	0.997	0.840	0.912	0.915
ubi-fights	10	Original	88.690	0.995	0.778	0.873	0.880
ubi-fights	10	RandomOverSampler	97.860	0.981	0.976	0.979	0.979
ubi-fights	10	RandomUnderSampler	94.520	0.932	0.961	0.946	0.945
ubi-fights	10	SMOTE	81.230	0.999	0.625	0.769	0.791
ubi-fights	10	TomekLinks	74.370	0.996	0.489	0.656	0.699
ubi-fights	25	ADASYN	97.520	0.984	0.966	0.975	0.975
ubi-fights	25	DataAugmentation	98.560	0.998	0.972	0.985	0.985
ubi-fights	25	DCGAN	99.300	0.999	0.986	0.992	0.992
ubi-fights	25	DeepSMOTE	97.440	0.999	0.949	0.973	0.974
ubi-fights	25	ENN	92.540	0.995	0.855	0.920	0.923
ubi-fights	25	NearMiss	85.540	0.796	0.956	0.869	0.850
ubi-fights	25	OneSidedSelection	90.310	0.999	0.807	0.893	0.898
ubi-fights	25	Original	93.920	0.997	0.881	0.935	0.937
ubi-fights	25	RandomOverSampler	98.890	0.993	0.985	0.989	0.989
ubi-fights	25	RandomUnderSampler	97.690	0.989	0.964	0.977	0.977
ubi-fights	25	SMOTE	94.000	0.998	0.881	0.936	0.938
ubi-fights	25	TomekLinks	97.280	0.995	0.951	0.972	0.973
ubi-fights	50	ADASYN	95.970	0.927	0.998	0.961	0.959
ubi-fights	50	DataAugmentation	99.700	0.997	0.996	0.996	0.997
ubi-fights	50	DCGAN	98.630	0.999	0.972	0.986	0.986
ubi-fights	50	DeepSMOTE	98.290	0.970	0.995	0.983	0.982
ubi-fights	50	ENN	99.160	0.986	0.997	0.992	0.992
ubi-fights	50	NearMiss	92.960	0.927	0.932	0.930	0.930
ubi-fights	50	OneSidedSelection	99.040	0.987	0.994	0.990	0.990
ubi-fights	50	Original	95.290	1.000	0.906	0.951	0.952
ubi-fights	50	RandomOverSampler	97.870	0.999	0.958	0.978	0.978
ubi-fights	50	RandomUnderSampler	97.630	0.997	0.956	0.976	0.976
ubi-fights	50	SMOTE	92.470	0.998	0.851	0.919	0.922
ubi-fights	50	TomekLinks	97.500	0.999	0.951	0.974	0.975
ubi-fights	75	ADASYN	97.640	0.998	0.955	0.976	0.976
ubi-fights	75	DataAugmentation	99.710	0.999	0.994	0.997	0.997
ubi-fights	75	DCGAN	99.880	0.998	0.998	0.999	0.998
ubi-fights	75	DeepSMOTE	99.810	0.998	0.998	0.998	0.998
ubi-fights	75	ENN	99.540	0.991	1.000	0.995	0.995
ubi-fights	75	NearMiss	98.040	0.964	0.998	0.981	0.980
ubi-fights	75	OneSidedSelection	99.430	0.989	0.999	0.994	0.994
ubi-fights	75	Original	99.110	0.999	0.983	0.991	0.991
ubi-fights	75	RandomOverSampler	99.470	0.996	0.994	0.995	0.995
ubi-fights	75	RandomUnderSampler	99.770	0.998	0.998	0.998	0.998
ubi-fights	75	SMOTE	99.520	0.999	0.992	0.995	0.995
ubi-fights	75	TomekLinks	99.480	0.991	0.999	0.995	0.995
ubi-fights	100	Original	99.910	0.998	1.000	0.999	0.999

Test de *Friedman*

Rangos promedio obtenidos al aplicar el procedimiento de *Friedman*.

Tabla 3: Rangos promedio de los algoritmos para el escenario de desbalance de 1 %.

Técnica	Rango Promedio
RandomOverSampler	4.2
ENN	4.4
DataAugmentation	4.8
SMOTE	5
TomekLinks	5.1
DCGAN	5.8
DeepSMOTE	6.4
ADASYN	6.8
RandomUnderSampler	7
OneSidedSelection	8.2
Original	9.3
NearMiss	11

Tabla 4: Rangos promedio de los algoritmos para el escenario de desbalance de 5 %.

Técnica	Rango Promedio
RandomOverSampler	2
TomekLinks	2.8
DCGAN	5.4
DeepSMOTE	5.4
OneSidedSelection	6
ADASYN	6.2
DataAugmentation	6.4
SMOTE	6.8
RandomUnderSampler	7.6
ENN	7.8
Original	9.6
NearMiss	12

Tabla 5: Rangos promedio de los algoritmos para el escenario de desbalance de 10 %.

Técnica	Rango Promedio
RandomOverSampler	1.7
DCGAN	4.2
DeepSMOTE	5
RandomUnderSampler	5.6
ENN	5.8
ADASYN	6
SMOTE	6.3
DataAugmentation	6.4
TomekLinks	7.4
OneSidedSelection	8
Original	9.8
NearMiss	11.8

Tabla 6: Rangos promedio de los algoritmos para el escenario de desbalance de 25 %.

Técnica	Rango Promedio
RandomOverSampler	1.2
DCGAN	2.6
SMOTE	4.9
DeepSMOTE	5.2
DataAugmentation	6.2
RandomUnderSampler	6.6
ENN	7.2
ADASYN	7.4
TomekLinks	7.4
OneSidedSelection	7.5
Original	9.8
NearMiss	12

Tabla 7: Rangos promedio de los algoritmos para el escenario de desbalance de 50 %.

Técnica	Rango Promedio
OneSidedSelection	3.4
RandomOverSampler	3.6
DCGAN	5.1
SMOTE	5.3
DataAugmentation	5.6
DeepSMOTE	5.6
TomekLinks	6.4
ADASYN	6.7
RandomUnderSampler	7.1
ENN	8
Original	9.6
NearMiss	11.6

Tabla 8: Rangos promedio de los algoritmos para el escenario de desbalance de 75 %.

Técnica	Rango Promedio
TomekLinks	3.6
DCGAN	3.7
RandomOverSampler	3.7
DeepSMOTE	5.4
SMOTE	5.4
ENN	5.6
DataAugmentation	6.1
RandomUnderSampler	6.8
ADASYN	7.3
OneSidedSelection	9.2
Original	9.4
NearMiss	11.8