UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE TELECOMUNICACIONES

HIBRIDACIÓN DE MODELOS SUPERVISADOS PARA LA DETECCIÓN DE INTRUSIONES EN REDES DE COMUNICACIÓN

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Telecomunicaciones

AUTORES: SHIRLEY SOFIA SINCHIGUANO PANJON
SANTIAGO ALEXANDER AULES REYES

TUTOR: JUAN CARLOS DOMÍNGUEZ AYALA

Quito-Ecuador

2025

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, SHIRLEY SOFIA SINCHIGUANO PANJON con documento de identificación N° 1754614319 y SANTIAGO ALEXANDER AULES REYES con documento de identificación N° 1726289646; manifestamos que:

Somos las autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 22 de julio del año 2025

Atentamente,

Shirley Sofia Sinchiguano Panjon
1754614319

Santiago Alexander Aules Reyes
1726289646

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Shirley Sofía Sinchiguano Panjon con documento de identificación No. 1754614319 y Santiago Alexander Aules Reyes con documento de identificación No. 1726289646, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autoras del artículo académico: "Hibridación de modelos supervisados para la detección de intrusiones en redes de comunicación", el cual ha sido desarrollado para optar por el título de: Ingeniero en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 22 de julio del año 2025

Atentamente,

Shirley Sofia Sinchiguano Panjon
1754614319

Santiago Alexander Aules Reyes
1726289646

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Juan Carlos Domínguez Ayala con documento de identificación N° 1713195590, docente de la Universidad Politécnica Salesiana declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: HIBRIDACIÓN DE MODELOS SUPERVISADOS PARA LA DETECCIÓN DE INTRUSIONES EN REDES DE COMUNICACIÓN, realizado por Shirley Sofía Sinchiguano Panjon con documento de identificación N° 1754614319 y por Santiago Alexander Aules Reyes con documento de identificación N° 1726289646, obteniendo como resultado final el trabajo de titulación bajo la opción artículo académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 22 de julio del año 2025

Atentamente,

_____

Ing. Juan Carlos Domínguez Ayala MSc.
1713195590

DEDICATORIA

Dedico este proyecto, en primer lugar, a Dios, por darme la sabiduría, fortaleza y salud para llegar hasta aquí. A mis padres, Roberto y Sofía, y a toda mi familia, por ser mi mayor pilar de apoyo, por su amor incondicional, su paciencia y por creer en mí incluso cuando yo dudaba. Este logro es tan mío como de ustedes, porque sin su guía y compañía, no habría sido posible.

Shirley Sofía Sinchiguano Panjon

El presente trabajo lo dedico primero a Dios, por darme la fortaleza para no rendirme y guiarme con su inmensa sabiduría, a mis padres Patricio y Esmeralda por su amor, paciencia y apoyo incondicional. A mí mismo por no decaer y seguir este camino hasta el final.

Santiago Alexander Aules Reyes

AGRADECIMIENTO

En primer lugar, a Dios, por brindarme la fuerza, la sabiduría y la salud para alcanzar esta meta. A mis padres, Sofía y Roberto, por su amor incondicional, sus enseñanzas y su apoyo constante en cada paso que he dado. A mis hermanos, por ser mi compañía, mi motivación diaria y por compartir conmigo cada logro y dificultad a lo largo de este camino. A mis abuelitos Ligia, Walter y Segundo, por su cariño inmenso y por ser un ejemplo de fortaleza y sabiduría. De manera especial, recuerdo con amor a mi abuelita Luz quien, aunque ya no está físicamente, sigue presente en mi corazón y en cada logro que alcanzo. No puedo dejar de mencionar a mis tías, por su cercanía, sus consejos y por estar siempre pendientes de mí. A toda mi familia, por ser un pilar fundamental en mi vida. A mis docentes, por su dedicación y por compartir con generosidad sus conocimientos. A quienes me acompañaron durante la etapa universitaria, gracias por su amistad, por los momentos compartidos y por hacer de este proceso una experiencia valiosa. Y de manera muy especial, a mi novio Jonathan, gracias por tu amor, paciencia y por estar a mi lado en cada momento, animándome a seguir adelante con fe y confianza.

Shirley Sofia Sinchiguano Panjon

A Dios por sobre todas las cosas, por su sabiduría, y fortaleza para no rendirme, a mis padres Patricio y Esmeralda por su amor, el siempre acompañarme, su apoyo incondicional y su guía en este largo camino, a mi abuelita María por sus consejos, a mis tíos Napoleón y Consuelo, de manera especial a mi parroquia Cristo Resucitado, un lugar en el cuál encontré un soporte espiritual en el momento más difícil del desarrollo de este trabajo. A mis docentes, por los conocimientos compartidos, en especial a mi tutor Juan Carlos Domínguez, pilar fundamental para el desarrollo de esta Tesis. A todos los compañeros que me acompañaron en esta etapa de la vida, gracias por su amistad, los momentos memorables y experiencias compartidas.

Santiago Alexander Aules Reyes

HIBRIDACIÓN DE MODELOS SUPERVISADOS PARA LA DETECCIÓN DE INTRUSIONES EN REDES DE COMUNICACIÓN

Santiago Aules[1], Shirley Sinchiguano[2], and Juan Carlos Domínguez[3]

[1] Universidad Politécnica Salesiana, Quito, Ecuador
saulesr@est.ups.edu.ec
[2] Universidad Politécnica Salesiana, Quito, Ecuador
sshinchiguanop@est.ups.edu.ec
[3] Universidad Politécnica Salesiana, Quito, Ecuador
jdominguez@ups.edu.ec

Resumen El presente trabajo evaluó el desempeño de un modelo híbrido de Machine Learning (ML) en el contexto de un Sistema de Detección de Intrusos en Redes (IDS) basado en anomalías. Para ello, se utilizó el conjunto de datos CIC-IDS2017, el cual incluye una variedad representativa de ataques contemporáneos como DDoS, Infiltration y Botnet, así como tráfico benigno. Se seleccionaron y combinaron los algoritmos Random Forest (RF) y K Nearest-Neighbors (KNN), y se compararon con modelos individuales como Support Vector Machine (SVM), XGBoost y Multi-Layer Perceptron (MLP), siguiendo la metodología CRISP-DM.
El proceso incluyó la limpieza de datos, codificación de variables y el balanceo de clases mediante Synthetic Minority Over-Sampling Technique (SMOTE), aplicado exclusivamente a los conjuntos de entrenamiento generados por validación cruzada. Se evaluaron los modelos utilizando métricas como Accuracy, Recall, F1-Score, Precision y ROC-AUC.
Como resultado, el modelo híbrido basado en votación soft logró un F1-Score de 99.43 %, superando a los algoritmos individuales. Este rendimiento evidenció la capacidad del enfoque propuesto para detectar ataques poco frecuentes, manteniendo un equilibrio entre rendimiento y eficiencia computacional, lo cual se refleja en la consistencia de los resultados obtenidos a través de los diferentes conjuntos de validación.

Palabras clave: Machine Learning, Detección de intrusos, CIC-IDS2017, Modelo híbrido.

Abstract. The present work evaluated the performance of a hybrid Machine Learning (ML) model in the context of an anomaly-based Intrusion Detection System (IDS). For this purpose, the CIC-IDS2017 dataset was used, which includes a representative variety of contemporary attacks such as DDoS, Infiltration, and Botnet, as well as benign traffic. Random

Forest (RF) and K Nearest Neighbors (KNN) algorithms were selected and combined, and compared with individual models such as Support Vector Machine (SVM), XGBoost, and Multi-Layer Perceptron (MLP), following the CRISP-DM methodology.
The process included data cleaning, variable encoding, and class balancing using the Synthetic Minority Over-Sampling Technique (SMOTE), applied exclusively to the training sets generated by cross-validation. The models were evaluated using metrics such as Accuracy, Recall, F1-Score, Precision, and ROC-AUC.
As a result, the hybrid model based on soft voting achieved an F1-Score of 99.43%, outperforming the individual algorithms. This performance demonstrated the capability of the proposed approach to detect rare attacks while maintaining a balance between performance and computational efficiency, as reflected in the consistency of the results obtained across different validation sets.

Keywords: Machine Learning, Intrusion Detection, CIC-IDS2017, Hybrid model.

1 Introducción

Con el constante crecimiento de los dispositivos conectados y una complejidad cada vez mayor del tráfico de red han surgido nuevos retos para la seguridad informática. En consecuencia, los IDS se han convertido en herramientas fundamentales para la protección de infraestructuras de red. No obstante, el software basado en firmas presenta limitaciones significativas: no pueden detectar ataques desconocidos (zero-day), dependen de actualizaciones constantes y suelen fallar ante técnicas de evasión avanzadas. Es por estas limitaciones que se ha impulsado el uso de estrategias más dinámicas como el ML, que permite identificar patrones anómalos en el tráfico sin requerir definiciones previas [1,2].

El principal desafío al aplicar Machine Learning (ML) en sistemas de detección de intrusos (IDS) es la identificación precisa de ataques poco frecuentes. Estos ataques, aunque escasos, pueden tener un impacto crítico en la seguridad del sistema, especialmente en entornos sensibles como el Internet de las Cosas (IoT) o los servicios financieros [3]. En conjuntos de datos como CIC-IDS2017, la distribución de clases está fuertemente desbalanceada, predominando el tráfico benigno frente a un número reducido de instancias maliciosas[4]. Esta asimetría afecta el rendimiento de los modelos tradicionales, que tienden a centrarse en las clases mayoritarias, disminuyendo su capacidad para detectar amenazas relevantes como Heartbleed o WebSQL Injection. Por consiguiente, el problema se agrava cuando el modelo implementado no es sensible a patrones locales o no ha sido adecuadamente ajustado a las características del conjunto de datos, incluyendo el desbalance de clases y los ataques menos frecuentes.

A nivel práctico, otro reto importante radica en los requerimientos computacionales. Algunos algoritmos, aunque potentes en términos teóricos, exigen una carga completa de datos en memoria o tiempos de entrenamiento prolongados, lo cual puede limitar su aplicabilidad en entornos reales con recursos restringidos [5].

Varios trabajos han explorado modelos supervisados como Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP) y XGBoost, mostrando resultados prometedores cuando se ajustan sus hiperparámetros. En [6] se logró un F1-score de 0.99 en RF al optimizar la cantidad de árboles y su profundidad, mientras que [7] reportó mejoras sustanciales en XGBoost mediante optimización bayesiana. Otras investigaciones reportan precisiones de hasta 98.3 % en KNN al elegir adecuadamente el número de vecinos y la métrica de distancia [8], y F1-scores cercanos a 0.96 en MLP utilizando técnicas de búsqueda aleatoria [9]. También se ha demostrado que modelos combinados como RF y XGBoost pueden alcanzar precisiones superiores al 99.9 % [10].

No obstante, muchos de estos trabajos se enfocan en contextos experimentales controlados y no consideran de forma explícita el impacto de las clases minoritarias ni los requisitos prácticos de implementación. En particular, algoritmos como SVM y KNN, aunque potentes, requieren recursos significativos de memoria, ya que cargan todo el conjunto en RAM. Por su parte, XGBoost, si bien más eficiente, sigue siendo computacionalmente demandante [11].

En este trabajo se seleccionaron cinco algoritmos supervisados ampliamente utilizados en la literatura para tareas de detección de intrusos: Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP) y XGBoost. Esta selección obedece a que cada uno representa una familia distinta de enfoques en aprendizaje supervisado, permitiendo contrastar su rendimiento sobre un mismo conjunto de datos. RF y XGBoost son modelos basados en árboles de decisión, reconocidos por su robustez ante ruido, su capacidad para manejar datos no lineales y su eficiencia en problemas multiclase. SVM ha sido ampliamente utilizado en contextos de clasificación binaria con márgenes máximos, siendo útil en escenarios con fronteras claras entre clases. KNN, por su parte, es un clasificador no paramétrico basado en distancia, eficaz para detectar patrones locales en datos con distribuciones complejas. MLP representa la aproximación tradicional de redes neuronales, capaz de modelar relaciones no lineales cuando se entrena correctamente. La evaluación de estos modelos sobre el conjunto CIC-IDS2017 permite explorar su efectividad frente a un conjunto de datos realista y actualizado, que incluye múltiples tipos de ataques y refleja condiciones de tráfico contemporáneo. Además, al aplicar técnicas uniformes de preprocesamiento, validación cruzada y ajuste de hiperparámetros, se garantiza una comparación justa entre los modelos evaluados [20].

Ante este panorama, se propuso un modelo híbrido basado en la combinación de RF y KNN mediante votación suave (Voting Classifier). Esta decisión no responde únicamente a su rendimiento empírico, sino a sus propiedades complementarias. Random Forest, como ensamblado de árboles de decisión, ofrece baja varianza, buena generalización y tolerancia al ruido, gracias a su estructura aleatoria y capacidad para manejar datos desbalanceados. KNN, por su parte, es un modelo basado en instancias que detecta patrones locales, lo que lo vuelve especialmente sensible a clases minoritarias, compensando la tendencia de RF a subestimar dichas clases [12].

Ambos algoritmos presentan ventajas complementarias en términos de eficiencia computacional: RF puede entrenarse en paralelo, lo cual reduce significativamente el tiempo de entrenamiento en arquitecturas multihilo; mientras que KNN no requiere la construcción de un modelo explícito, ya que su inferencia se basa en comparaciones directas con las instancias del conjunto de entrenamiento, lo que simplifica su implementación. La combinación de estos modelos permite equilibrar la sensibilidad de KNN hacia clases minoritarias con la robustez de RF frente al sobreajuste, mejorando así la capacidad general del sistema para detectar correctamente tanto ataques frecuentes como raros [13].Para cumplir con este objetivo se utilizó la metodología CRISP-DM [14], la cual incluye la limpieza de valores nulos, normalización de datos mediante z-score y codificación de etiquetas mediante Label Encoder, seguido del balance de clases usando SMOTE y el entrenamiento del modelo aplicando validación cruzada.

A diferencia de otros trabajos, esta investigación no solo evalúa métricas generales como Accuracy, Precision, Recall y F1-score, sino que también analiza el

comportamiento del sistema ante clases desbalanceadas y considera una posible implementación en entornos con tráfico real.

2 Trabajos Relacionados

Para contextualizar esta investigación, se examinan contribuciones recientes agrupadas en tres líneas relevantes: (i) algoritmos supervisados tradicionales aplicados a la detección de intrusos, (ii) enfoques híbridos que combinan modelos o técnicas de mejora del rendimiento, y (iii) estudios que emplean el conjunto de datos CIC-IDS2017 como base experimental. Esta revisión permite identificar los enfoques más utilizados, sus ventajas y limitaciones, así como establecer las motivaciones del presente trabajo. A continuación, se detallan los métodos supervisados más representativos y su desempeño reportado en la literatura.

2.1 Algoritmos de Aprendizaje Supervisado

Modelos clásicos como K-Nearest Neighbors (KNN), Naïve Bayes (NB) y Árboles de Decisión (DT) siguen siendo una opción gracias a su simplicidad, bajo consumo de recursos y facilidad para ser implementados[15].En [16], estos algortimos se probaron en un entorno controlado y lograron tasas de detección por encima del 96 %.Sin embargo, aunque al principio los resultados parecen buenos estos modelos no funcionan tan bien cuando los datos están desbalanceados o tienen errores,lo que hace que no sean tal útiles en casos reales.

En [17],se compararon SVM, RF y DT específicamente sobre el dataset CIC-IDS2017. Los autores evaluaron el rendimiento utilizando métricas como accuracy, precision, recall y F1-score. Random Forest sobresalió por su capacidad de mantener un rendimiento estable incluso en presencia de ruido, superando a SVM en precisión y estabilidad, por otro lado [18]incorporó RF, NB y una arquitectura basada en redes convolucionales (CNN). Aunque la CNN obtuvo métricas más altas (por ejemplo, un F1-score de 0.99), el consumo de recursos fue considerablemente mayor, lo cual limitó su aplicabilidad práctica. Esta comparación resalta que, en contextos con restricciones de procesamiento, modelos como RF continúan siendo una opción eficaz y escalable.

2.2 Modelos Híbridos y Optimización de Hiperparámetros

La necesidad de mejorar la precisión sin incurrir en sobreajuste ha impulsado el desarrollo de modelos híbridos. En [19], se propuso un sistema que integra algoritmos genéticos con SVM, mejorando tanto la selección de características como la calibración de hiperparámetros. Esta combinación redujo la tasa de falsos positivos en más del 30 % en comparación con el SVM estándar, lo que evidencia el impacto positivo de la optimización evolutiva.

2.3 Uso del Dataset CIC-IDS2017

El conjunto de datos CIC-IDS2017 ha sido objeto de numerosos estudios debido a su nivel de realismo y la diversidad de ataques que contiene, abarcando tanto amenazas frecuentes como poco usuales.En [20] se optimizó una variante del modelo Random Forest, conocida como Random Subspace Random Forest (RS-RF), mediante Grid Search y ajuste de hiperparámetros, alcanzando una precisión del 98.9 %. De manera complementaria, en [21] se propuso usar técnicas de transferencia de aprendizaje, como redes convolucionales (CNN) preentrenadas sobre el dataset CSE-CIC-IDS2018 y luego adaptadas a CIC-IDS2017, logrando F1-scores cercanos a 0.9999. Aunque ambos enfoques alcanzaron resultados notables, la complejidad computacional de modelos más avanzados como las CNN representa una limitación práctica frente a soluciones más simples y escalables como RF.

3 Metodología

3.1 Descripción del Conjunto de Datos

En esta investigación se seleccionó el dataset CIC-IDS2017 proporcionado por el Canadian Institute for Cybersecurity por su capacidad de representar el tráfico normal de red y ataques como DoS, Brute Force, Web SQL Injection, HeartBleed, entre otros. Contiene más de 2.8 millones de registros y 122 características por registro, tales como duración de conexión, cantidad de paquetes, tamaño en bytes, tiempos entre paquetes, flags TCP, entre otras lo que lo hace ideal para entrenar algoritmos de ML aplicados a sistemas IDS [21] . Como se observa en la Figura 1, existe una marcada predominancia de la clase Benign, seguida por un grupo reducido de clases minoritarias, lo que evidencia un fuerte desbalance de clases.

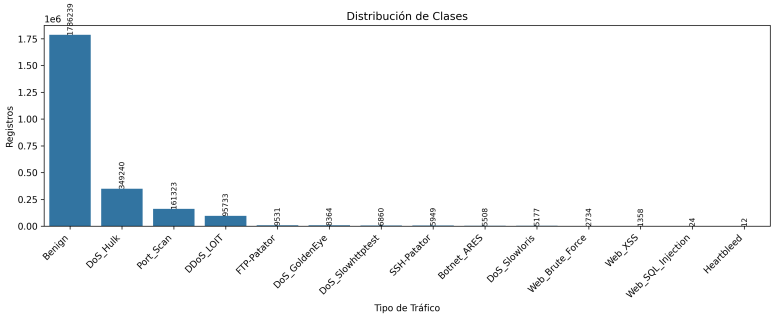

Figura 1. Distribución de clases en el conjunto de datos CIC-IDS2017.

3.2 Limpieza y Procesamiento de Datos

Dado el gran tamaño del dataset y la presencia de columnas poco útiles, se realizó una limpieza iterativa en bloques (chunks) de 10.000 registros para optimizar el uso de memoria. Las columnas con más del 50 % de valores nulos, constantes o identificadores (como Timestamp, IP, Port, etc.) fueron descartadas. Las columnas numéricas restantes fueron imputadas con la mediana en caso de valores faltantes.

3.3 Codificación y Filtrado Final

Como parte del preprocesamiento, se codificaron las etiquetas (label) utilizando LabelEncoder, generando una nueva columna denominada 'label' con valores numéricos. Además, se eliminaron los registros que contenían valores nulos o infinitos.

Para asegurar el correcto desempeño sin comprometer la viabilidad computacional, se aplicó una limpieza iterativa en bloques (chunks) de 10.000 registros, generando un nuevo archivo CSV. Se utilizó muestreo estratificado para mantener la proporción original entre clases mayoritarias y minoritarias, lo que garantiza estadísticamente una representación proporcional de cada clase en cada subconjunto. Esta técnica reduce el sesgo en la estimación de las métricas de rendimiento del modelo, ya que preserva la distribución de la variable objetivo durante el muestreo[22].

3.4 Normalización de Características

Se normalizaron todas las variables numéricas usando la técnica Z-score, aplicando el proceso en bloques de 10.000 registros para garantizar eficiencia computacional sin saturar la memoria disponible. Posteriormente, el conjunto de datos fue dividido en dos subconjuntos de forma estratificada: 80 % para entrenamiento y 20 % para prueba, manteniendo la proporción de clases. Esta estrategia de procesamiento por bloques y división estratificada permitió equilibrar el uso de recursos y la representatividad de las clases durante el entrenamiento del modelo.

3.5 Balanceo durante el Entrenamiento

Aunque el conjunto completo no fue balanceado globalmente, se empleó la técnica SMOTE (Synthetic Minority Over-sampling Technique) unicamente en los subconjuntos de entrenamiento generados durante cada fold de validación cruzada. Esta estrategia evita el data leakage (filtrado de datos) que podría ocurrir si se generan muestras sintéticas antes de la separación de entrenamiento y prueba [23]. En cada división de validación cruzada con 5 folds, se aplicó SMOTE solamente al conjunto de entrenamiento, dejando intacto el conjunto de prueba para mantener su distribución original.Esta decisión garantiza que las métricas obtenidas no estén sobreestimadas por el uso de datos sintéticos y represente de forma más precisa el rendimiento del modelo con nuevos datos.

3.6 Diagrama de Flujo del Proceso

La Figura 2 muestra el flujo general del proceso metodológico, el cual sigue la estructura CRISP-DM desde la comprensión del problema hasta la evaluación final del modelo híbrido.

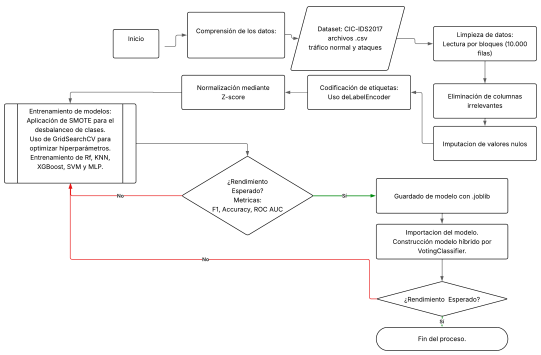

Figura 2. Diagrama de flujo - Proceso de entrenamiento.

4 Evaluación Experimental

Para evaluar el rendimiento de los modelos, se entrenaron cinco algoritmos de aprendizaje supervisado; con la validación cruzada, se obtuvó los mejores hiperparámetros que se detallan a continuación en el Cuadro 1.

Modelo	**Hiperparámetros**
Random Forest	`n_estimators=100`, `max_depth=10`, `min_samples_split=2`, `random_state=42`
KNN	`n_neighbors=3`, `metric='manhattan'`, `weights='uniform'`, `n_jobs=-1`
SVM	`kernel='rbf'`, `C=1.0`, `gamma='scale'`, `probability=True`, `random_state=42`
XGBoost	`tree_method='hist'`, `objective='multi:softprob'`, `eval_metric='mlogloss'`, `n_estimators=100`, `learning_rate=0.3`, `max_depth=6`, `n_jobs=-1`
MLP	`hidden_layer_sizes=(128, 64)`, `activation='relu'`, `solver='adam'`, `max_iter=300`
Ensemble (Voting)	`voting='soft'`, `estimators=[RF, KNN]`, `weights=None`, `n_jobs=None`

Cuadro 1. Hiperparámetros utilizados para cada modelo

Los modelos fueron evaluados sobre un conjunto de prueba con proporciones estratificadas de clases. Se utilizaron métricas estándar como Accuracy, Precision, Recall, F1 Score macro y ROC AUC macro.El Cuadro 2 resume los resultados obtenidos. A partir de esos resultados se determinó la combinación de modelos para la construcción del modelo híbrido basado en votación mayoritaria (Ensemble Voting).

Modelo	**Accuracy**	**F1 Score (Macro)**	**Precisión**	**Recall**	**ROC AUC**
KNN	0.9996	0.9933	0.9935	0.9931	N/A
XGBoost	1.0000	1.0000	1.0000	1.0000	1.0000
SVM	0.9999	0.9184	0.9276	0.9108	0.9483
Random Forest	0.9914	0.7744	0.7406	0.8519	0.9637
MLP	1.0000	0.9679	0.9935	0.9497	1.0000
Ensemble Voting	**0.9999**	**0.9943**	**0.9923**	**0.9963**	**N/A**

Cuadro 2. Comparación de métricas de evaluación para cada modelo

Las variables de entrada corresponden a las 122 características numéricas extraídas del conjunto CIC-IDS2017, mientras que la variable de salida fue la etiqueta codificada de clase. Estas características incluyen estadísticas de flujo de red como duración de la conexión, número de paquetes, tamaños, y tiempos entre paquetes, entre otras. La evaluación se realizó sobre un conjunto de prueba con proporciones estratificadas, asegurando una representación justa de cada clase.

La métrica Accuracy refleja el porcentaje global de predicciones correctas, aunque puede no ser suficiente ante un desbalance de clases. Por esta razón, se empleó también el F1 Score (macro), que calcula el promedio del F1 por clase, ponderando tanto la precisión (Precision) como la sensibilidad (Recall). La precisión indica la proporción de verdaderos positivos entre todas las predicciones positivas realizadas, mientras que el recall representa la proporción de verdaderos positivos detectados frente al total de verdaderos positivos existentes. Finalmente, ROC AUC macro evalúa la capacidad del modelo para distinguir entre clases, incluso en presencia de desbalance, considerando el área bajo la curva ROC promedio.

Los resultados muestran que XGBoost y MLP alcanzaron una precisión perfecta en todas las métricas, lo cual podría indicar sobreajuste. Por el contrario, Random Forest obtuvo resultados significativamente menores, en especial en F1 Score y precisión, posiblemente debido a la distribución de clases o a la sensibilidad del modelo frente a las características del conjunto. El modelo híbrido basado en votación (Ensemble Voting) logró combinar los puntos fuertes de KNN y RF, obteniendo un equilibrio destacable entre todas las métricas evaluadas, lo que justifica su elección como propuesta final.

4.1 KNN

Mediante la aplicación de los hiperparámetros del cuadro1 Knn logró una precisión del 99.35 % y un F1 Score (macro) de 99.33 %, destacándose por su equilibrio entre rendimiento y simplicidad. Fue el modelo más efectivo sin mostrar signos de sobreajuste. Como se observa en la Figura 3 su clasificación es elevada en la mayoría de las clases, con predicciones acertadas especialmente en Benign, DoS Hulk y Port Scan. Sin embargo, algunas etiquetas como Heartbleed y Web SQL Injection, al estar poco representadas, presentaron dificultades para ser correctamente identificadas.

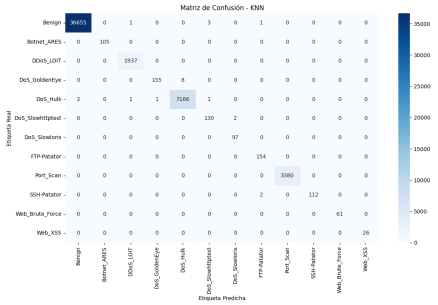

Figura 3. Matriz de confusión del modelo Nearest Neighbors (KNN)

.

4.2 XGBoost

Con el ajuste de sus hiperparámetros, XGboost alcanzó una clasificación perfecta, con valores de precisión, recall y F1-score iguales a 1.0 en todas las clases como se observó en el cuadro 2 . Este rendimiento sobresaliente abarcó tanto clases mayoritarias como minoritarias, incluyendo etiquetas como Web_XSS y Heartbleed. No obstante, estos resultados deben interpretarse con cautela, ya que podrían estar influenciados por una posible sobreadaptación al conjunto de entrenamiento o validación como se observa en la clasificación de clases de la Figura. 4.

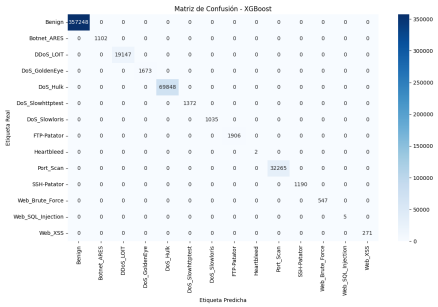

Figura 4. Matriz de confusión del modelo XGBoost.

4.3 SVM

Con su respectivo ajuste, el modelo presentó un rendimiento muy alto,con métricas de precisión, recall y F1-score superiores al 90 %. Estos resultados son consistentes incluso en etiquetas minoritarias como Web_XSS y SSH_Patator, lo que sugiere una alta capacidad del modelo para generalizar. Sin embargo, debe tenerse en cuenta que el entrenamiento se realizó sobre un subconjunto reducido del total de datos, lo cual podría influir en la aparente perfección del rendimiento como se observa en la Figura 5

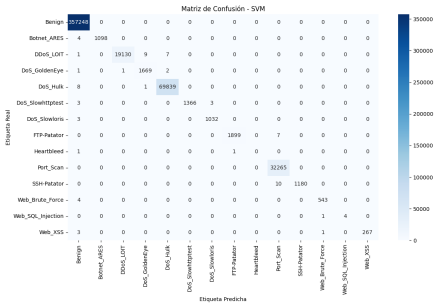

Figura 5. Matriz de confusión del modelo SVM

4.4 Random Forest

A pesar de usar los hiperparámetros del cuadro 1 RF obtuvo resultados relativamente bajos, con un F1 Score de 0.7564 y una precisión del 71.93 % en comparación con el rendimiento de 93.8 % de [17]. Su estabilidad demostrada en 10 y su bajo costo de implementación lo convirtieron en un candidato clave para la hibridación, muestra un rendimiento sólido en clases predominantes como Benign,DoS_Hulk y Port_Scan, alcanzando una exactitud global elevada. No

obstante, se observan errores significativos en clases poco representadas como Heartbleed,Web_Brute_Force y Web_SQL_Injection, lo que indica una sensibilidad limitada a la detección de ataques minoritarios como se observa en la Figura 6

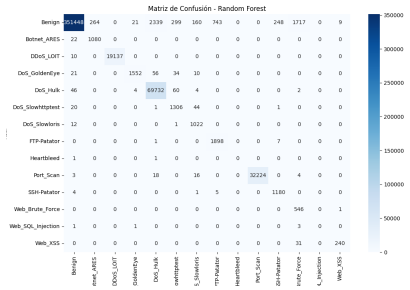

Figura 6. Matriz de confusión el modelo Random Forest (RF)

4.5 MLP

Como se observa en la Figura 7, el modelo MLP clasificó de manera casi perfecta, eso se evidencia en el resultado de F1-Score con 96.79 %, pero no se valida con la figura 10 el escenario de clasificación perfecto en clases minoritarias levanto la sospecha de memorización del modelo antes que de generalización.

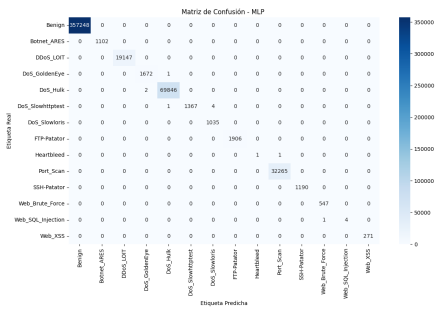

Figura 7. Matriz de confusión generada por el modelo Multi-Layer Perceptron (MLP)

4.6 Modelo Híbrido Ensemble Voting

Finalmente, el modelo híbrido Ensemble Voting, basado en la combinación de KNN y RF, alcanzó un F1 Score (macro) de 0.9943, junto con una precisión del 99.23 % y un recall del 99.63 %. Estos resultados demuestran que la combinación

de modelos bien entrenados y optimizados puede superar a cada clasificador individual en términos de rendimiento general y robustez como se observa en la Figura 8. El modelo fue capaz de clasificar correctamente la mayoría de etiquetas, incluidas clases minoritarias como Botnet_ARES, Web_XSS y Web_Brute_Force.

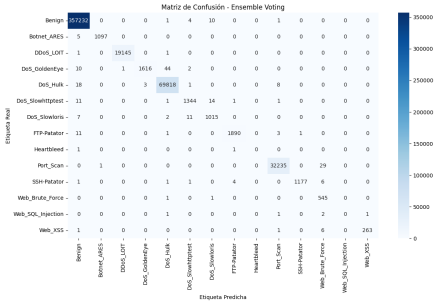

Figura 8. Matriz de confusión obtenida por el esquema de ensamblado Voting Soft Ensemble

4.7 Comparación y Análisis Estadístico

Estos resultados también se reflejan como se observa en la Figura 9, donde el modelo híbrido mantuvo un desempeño consistente y superior a los modelos base KNN y Random Forest.Esto demuestra que el modelos mantiene un buen rendimiento en diferentes particiones de validación, lo que indica que generalizó de manera satisfactoria cada clase de ataque.

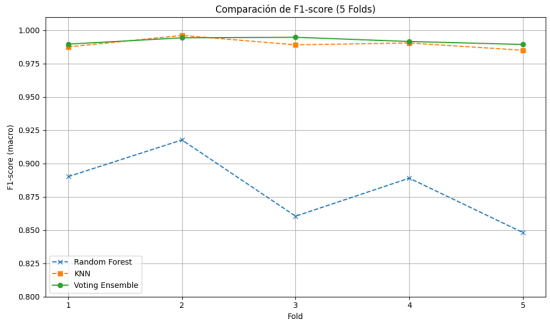

Figura 9. Comparación de F1-score promedio entre los modelos evaluados.

Además, la prueba estadística t-test muestra diferencias significativas como se observa en la Figura 10, respaldando que el rendimiento del modelos híbrido frente a KNN y RF es estadísticamente superior.Este análisis cuantitativo aporta

rigor y justificación a la elección del esquema de ensamblado como propuesta final.

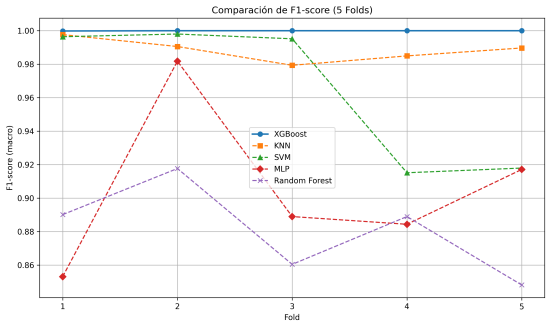

Figura 10. Prueba t-test sobre los F1-score macro en 5 folds para comparar el modelo híbrido con los individuales.

5 Discusión y Comparación

Durante la fase de evaluación, se identificaron diferencias sustanciales entre los modelos analizados, tanto en su rendimiento global como en su comportamiento frente a clases específicas del conjunto CIC-IDS2017. Modelos como XGBoost y SVM obtuvieron métricas generales cercanas al rendimiento perfecto, lo cual podría interpretarse inicialmente como un resultado altamente favorable. Sin embargo al analizar la Figura 10 se observa que XGBoost pudiera estar con sobreajuste, dado que mantiene un F1-score constante. Por otra parte KNN demostró ser el modelo más estable, mantieniendo un F1-score entre 0,98 y 0,99 lo que fue crucial al momento de tomar la decisión para en ensamble final. En cuanto al modelo SVM presenta una caída notable en el fold 4 lo que hace que se desestime un sobreajuste. Como contraparte esta su demanda computacional, este modelo se demoró 8 horas en entrenar lo que no lo hace viable en un escenario de desición rápida[24]. Observando el desempeño de MLP, se determinó que el modelo era muy inestable aun cuando se aplicó validación cruzada, los resultados obtenidos no respaldan el F1-Score alcanzado; en consecuencia, el modelo fue descartado para la selección final por tener indicios de memorización en lugar de generalización de clases. El rendimiento de Random Forest, aunque más bajo, se mantuvo estable con una buena generalización en clases minoritarias. Su F1-score no fue el más alto, pero, tiene concordancia con la estabilidad entre folds. Su comportamiento en predicción y la manera en que representa ataques inusuales, justificaron la inclusión en el modelo final. El modelo final, *Ensemble voting soft*, demostró que, mediante la hibridación de los clasificadores individuales, se puede obtener un desempeño superior, superando el desbalance de clases y mejorando

la detección de ataques poco frecuentes. Gracias a la estabilidad de Random Forest y la alta precisión de KNN, se redujo tanto la tasa de falsos positivos como de falsos negativos, consolidándose como un modelo estable en contexto de IDS.

6 Conclusión y Trabajos Futuros

Este trabajo evaluó el desempeño de cinco modelos supervisados para la detección de intrusos en redes: K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), XGBoost, y un modelo híbrido basado en votación mayoritaria (Ensemble Voting). Los resultados experimentales demostraron que, aunque modelos como XGBoost y MLP alcanzaron métricas perfectas (Accuracy y F1 Score cercanos al 100 %), mostraron indicios claros de sobreajuste debido al uso intensivo de datos sintéticos generados por SMOTE, lo cual limita su generalización en entornos reales. Por otro lado, SVM tuvo una variabilidad considerable en su rendimiento (F1 Score variando entre 0.918 y 0.948), dificultando su uso estable en condiciones operativas reales.

El algoritmo KNN presentó un desempeño muy estable (Accuracy de 99.35 % y F1 Score de 99.33 %), mostrando una excelente capacidad para clasificar correctamente incluso ataques poco frecuentes, como Heartbleed y WebSQL Injection. Random Forest obtuvo métricas menores (Accuracy de 99.14 % y F1 Score de 77.44 %), especialmente afectado por el desbalance extremo del dataset, lo que se reflejó en su baja precisión en clases minoritarias.

La implementación del modelo híbrido Ensemble Voting permitió aprovechar la estabilidad de KNN y compensar las debilidades específicas de Random Forest, obteniendo así un rendimiento equilibrado y robusto con un Accuracy de 99.99 % y un F1 Score macro de 99.43 %. Esta combinación demostró ser especialmente efectiva para mejorar la detección de ataques con escasa representación en los datos.

Sin embargo, a pesar de estos buenos resultados, todavía persisten desafíos por resolver. El desbalance significativo entre clases (más del 80 % de registros clasificados como Benign) continuó afectando negativamente el desempeño de varios modelos, lo que se refleja especialmente en las métricas de Precision y Recall en ataques minoritarios como Heartbleed o WebSQL Injection. Este impacto queda evidenciado por la baja precisión (menos del 75 %) que mostró Random Forest frente a estas clases poco frecuentes.

Para trabajos futuros, se propone explorar métodos adicionales que aborden el desbalance y reduzcan aún más el riesgo de sobreajuste. En concreto, sería interesante evaluar técnicas avanzadas como reducción de dimensionalidad mediante PCA o autoencoders, lo que permitiría entrenamientos más rápidos sin perder eficacia. Además, incorporar métodos modernos de explicabilidad como SHAP o LIME facilitaría una mejor interpretación y confianza en los resultados, aumentando la posibilidad de adopción del modelo en entornos operativos reales. Finalmente, dado el creciente uso de dispositivos IoT, sería prometedor

evaluar enfoques ligeros basados en modelos cuánticos o grandes modelos de lenguaje (LLMs) adaptados a tráfico IoT, así como desplegar y validar el modelo en entornos con tráfico real, analizando su desempeño bajo condiciones no controladas.

Referencias

1. S. Mittal, M. Wazid, D. P. Singh, A. K. Das, and M. S. Hossain, "A Deep Learning Ensemble Approach for Malware Detection in Internet of Things Utilizing Explainable Artificial Intelligence," *Engineering Applications of Artificial Intelligence*, vol. 139, p. 109560, 2025.
2. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
3. M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 4, pp. 2412–2425, 2020.
4. A. H. Lashkari, G. Dastghaibyfard, and A. A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection," *Computers*, vol. 10, no. 4, p. 44, 2021.
5. M. Alomari and Z. A. Jalil, "Lightweight Machine Learning Model for Efficient Intrusion Detection in Resource-Constrained Environments," *IEEE Access*, vol. 10, pp. 7342–7355, 2022.
6. E. V. Jaimes Bastidas, *Detección de amenazas en redes IoT empleando modelo híbrido de Machine Learning y Deep Learning*, Tesis doctoral, Universidad de los Andes, Bogotá, Colombia, 2022. [En línea]. Disponible en: https://repositorio.uniandes.edu.co/handle/1992/60545
7. M. A. Talukder, "A Dependable Hybrid Machine Learning Model for Network Intrusion Detection," arXiv preprint, 2023. [Online]. Available: https://doi.org/10.31224/2716
8. P. Chamoso, A. González-Briones, F. De La Prieta, and J. M. Corchado, "Deep Learning and Machine Learning Models for Edge Computing in Smart Cities: A Case Study for Traffic Prediction," *IEEE Access*, vol. 12, pp. 16015–16030, 2024, doi: 10.1109/ACCESS.2024.3359619.
9. O. I. Falowo, M. Ozer, C. Li, and J. B. Abdo, "Evolving Malware and DDoS Attacks: Decadal Longitudinal Study," *IEEE Access*, vol. 12, pp. 1–1, 2024.
10. Y. Zhu *et al.*, "Optimization of the Random Forest Hyperparameters for Power Industrial Control Systems Intrusion Detection Using an Improved Grid Search Algorithm," *Applied Sciences*, vol. 12, no. 18, p. 8954, 2022.
11. P. Chimphlee and T. Chimphlee, "Hyperparameters Optimization XGBoost for Network Intrusion Detection," *IAES Int. J. Artif. Intell.*, vol. 13, no. 1, pp. 223–232, 2024.
12. S. Dey, M. F. Mridha, and M. Rahman, "Comparative Performance Analysis of Machine Learning Algorithms for Intrusion Detection," *Electronics*, vol. 10, no. 23, p. 3003, 2021.
13. M. A. Baig, M. A. Baig, and S. A. Madani, "A Lightweight Machine Learning Model for Efficient Intrusion Detection in Edge-Enabled IoT Networks," *IEEE Access*, vol. 9, pp. 97728–97742, 2021.

14. A. Mariscal, O. Marbán, and C. Fernández, "A survey of data mining and knowledge discovery process models and methodologies," *Knowledge Engineering Review*, vol. 25, no. 2, pp. 137–166, 2020.
15. S. Umer, M. Sher, and F. Aadil, "Comparative Evaluation of Machine Learning Techniques for Intrusion Detection," *IEEE Access*, vol. 8, pp. 134246–134260, 2020.
16. A. Sharma, S. Mukhopadhyay, and R. Ghosh, "A Comparative Study of Lightweight Supervised Algorithms for Real-Time Intrusion Detection," *J. Cyber Secur. Technol.*, vol. 5, no. 3, pp. 201–215, 2021.
17. M. Almiani, J. Alzubi, A. Alwakeel, and W. Boulila, "Performance Comparison of Supervised Learning Techniques on CIC-IDS2017," *Sensors*, vol. 20, no. 22, p. 6668, 2020.
18. M. A. Rahman and M. R. Islam, "Evaluating Tree-based and Deep Learning Models for Network Intrusion Detection," *Comput. Sci. Rev.*, vol. 38, p. 100309, 2020.
19. B. Farahani, A. Shams, and J. Rezazadeh, "Hybrid Intrusion Detection System Using Genetic Algorithm and SVM for Feature Selection and Classification," *Wireless Netw.*, vol. 27, no. 1, pp. 479–491, 2021.
20. M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection," *IEEE Trans. Netw. Serv. Manage.*, vol. 17, no. 4, pp. 2412–2425, 2020.
21. I. Aljarrah, M. Aldwairi, and E. Basha, "Deep Transfer Learning for Intrusion Detection on CIC-IDS2017 and CSE-CIC-IDS2018," *J. Inf. Secur. Appl.*, vol. 66, p. 103146, 2022.
22. Baeldung, "Stratified Sampling in Machine Learning," tutorial, Apr.2024.
23. H. He and E. A. Garcia, "Learning from imbalanced data: Open challenges and future directions with applications," *IEEE Transactions on Knowledge and Data Engineering*, 2021. doi: https://doi.org/10.1109/TKDE.2021.3097634
24. M. Choi, J. Lee, and Y. Kim, "Real-Time Constraints in Machine Learning: A Case Study in Industrial Control Systems," *IEEE Access*, vol. 8, pp. 110123–110135, 2020.