InvestiGO
Revista Científica Multidisciplinaria
ISSN 2953-6367
http://revistainvestigo.com

Julio - diciembre 2026
Vol. 7 No. 20, PP. 314-332
https://doi.org/10.56519/bvpms658

ANÁLISIS COMPARATIVO DEL RENDIMIENTO DE MODELOS DE APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE INTRUSIONES USANDO EL DATASET CIC-IDS2017

COMPARATIVE PERFORMANCE ANALYSIS OF MACHINE LEARNING MODELS FOR INTRUSION DETECTION USING CIC-IDS2017 DATASET

Katherine Adriana Merino Villa[1], Juan Carlos Yungán Cazar[2], Edgar Gualberto Salazar Alvarez[3], Diego Alejandro Cáceres Veintimilla[4]

{kathetine.merino@espoch.edu.ec[1], jyungan@espoch.edu.ec[2], edgar.salazar@espoch.edu.ec[3], diego.caceres@espoch.edu.ec[4]}

Fecha de recepción: 14/05/2026 / Fecha de aceptación: 25/06/2026 / Fecha de publicación: 08/07/2026

RESUMEN: El incremento de infraestructuras tecnológicas, de ciberamenazas y del volumen de tráfico de red, ha convertido a la detección de intrusiones como un componente crítico en ciberseguridad. Por lo anterior mencionado, el presente estudio realizó una evaluación comparativa del desempeño de los algoritmos Random Forest, Support Vector Machine y Naive Bayes, con respecto a la clasificación de tráfico. Como marco experimental, se empleó el dataset CIC-IDS2017, bajo un enfoque cuantitativo de tipo experimental, analizando tráfico BENIGN y ataques DDoS y PortScan, se trabajó con más de un millón de registros, los cuales siguieron los procesos de limpieza transformación numérica e imputación de datos faltantes. Posteriormente se seleccionaron 300 000 registros para entrenar los modelos y poder evaluarlos con las métricas de exactitud, precisión, recall y F1-score. Además, se evaluó la interacción e importancia de las variables, matrices de confusión y la correlación entre características del tráfico de red. Los resultados obtenidos evidenciaron que Random Forest presentó el mejor desempeño, alcanzando una exactitud cercana al 99.98 % y mostrando una alta capacidad para diferenciar tráfico. Por otro lado, Support Vector Machine obtuvo una exactitud aproximada del 80.74 %, con limitaciones para identificar ataques PortScan. En contraste, Naive Bayes registró el rendimiento más bajo con una exactitud de 44.62 %, evidenciando dificultades para trabajar con datasets complejos y con variables correlacionadas. El análisis de importancia de características permitió determinar que variables vinculadas al volumen de tráfico, al tamaño de los paquetes y al comportamiento estadístico de las transmisiones son determinantes para distinguir el tipo de tráfico. Estos hallazgos, evidencia que Random Forest es la opción más equilibrada: combina alta capacidad de generalización,

[1]Escuela Superior Politécnica de Chimborazo – Ecuador, https://orcid.org/0009-0001-0616-9611; +593983270118
[2]Escuela Superior Politécnica de Chimborazo – Ecuador, https://orcid.org/0000-0001-5682-0399; +593979338270
[3]Escuela Superior Politécnica de Chimborazo – Ecuador, https://orcid.org/0000-0003-0988-0641; +593984939338
[4]Escuela Superior Politécnica de Chimborazo – Ecuador, https://orcid.org/0000-0003-0498-1240; +593996568359

resistencia al sobreajuste y eficiencia computacional, lo que lo hace adecuado para desplegar sistemas IDS basados en aprendizaje automático en redes reales, entornos donde la correlación entre variables es alta.

Palabras clave: aprendizaje automático, ciberseguridad, CIC-IDS2017, detección de intrusiones, Random Forest, tráfico de red

ABSTRACT: The increase in technological infrastructure, cyber threats, and network traffic volume has made intrusion detection a critical component of cybersecurity. Therefore, this study conducted a comparative evaluation of the performance of the Random Forest, Support Vector Machine, and Naive Bayes algorithms with respect to traffic classification. The CIC-IDS2017 dataset was used as an experimental framework, employing a quantitative experimental approach. Benign traffic, DDoS attacks, and PortScans were analyzed, working with over one million records, which underwent cleaning, numerical transformation, and imputation of missing data. Subsequently, 300,000 records were selected to train the models and evaluate them using metrics such as accuracy, precision, recall, and F1 score. Additionally, the interaction and importance of variables, confusion matrices, and the correlation between network traffic characteristics were assessed. The results showed that Random Forest performed best, achieving an accuracy of nearly 99.98% and demonstrating a high capacity for differentiating traffic. Support Vector Machine, on the other hand, achieved an accuracy of approximately 80.74%, with limitations in identifying PortScan attacks. In contrast, Naive Bayes registered the lowest performance with an accuracy of 44.62%, demonstrating difficulties working with complex datasets and correlated variables. Feature importance analysis revealed that variables related to traffic volume, packet size, and the statistical behavior of transmissions are key to distinguishing traffic types. These findings demonstrate that Random Forest is the most balanced option: it combines high generalization capacity, resistance to overfitting, and computational efficiency, making it suitable for deploying machine learning-based IDS systems in real-world networks, environments where the correlation between variables is high.

Keywords: machine learning, cybersecurity, CIC-IDS2017, intrusion detection, Random Forest, network traffic

INTRODUCCIÓN

El crecimiento de las infraestructuras tecnológicas y la dependencia de estas en sectores estratégicos como salud, finanzas, industria y educación ha permitido la exposición a amenazas cibernéticas. Actualmente, los ataques informáticos son más sofisticados, dinámicos y automatizados, afectando de manera importante a la confidencialidad, integridad y disponibilidad de la información. Por lo que la detección temprana de actividades maliciosas es una necesidad prioritaria dentro de la ciberseguridad moderna (1,2).

En este contexto, los Sistemas de Detección de Intrusiones (IDS) cumplen un rol fundamental en la identificación de comportamientos anómalos dentro de las infraestructuras de red. Tradicionalmente, estos sistemas han utilizado mecanismos basados en firmas y reglas

predefinidas para reconocer amenazas conocidas; sin embargo, este enfoque presenta limitaciones frente a ataques emergentes y variantes nuevas y desconocidas. Frente a estos escenarios, las técnicas de aprendizaje automático han adquirido una relevancia importante en el desarrollo de sistemas IDS inteligentes, gracias a su capacidad para analizar grandes volúmenes de datos y reconocer patrones complejos asociados a tráfico malicioso (3,4).

Varias investigaciones han evidenciado que algoritmos supervisados como Random Forest, Support Vector Machine (SVM) y Naive Bayes presentan relevantes resultados en la clasificación relacionadas con detección de intrusiones. Entre las ventajas principales identificadas de estos modelos destacan la automatización del análisis, la capacidad de generalización y la reducción de falsos positivos frente a métodos tradicionales. Random Forest en cambio ha mostrado un comportamiento robusto en datasets de altas dimensiones debido a su estructura basada en ensamblaje de árboles de decisión, mientras que SVM ha demostrado un rendimiento competitivo en problemas de clasificación complejos. Por otra parte, Naive Bayes es utilizado por su simplicidad matemática y rapidez computacional (5–8).

El desarrollo de sistemas IDS basados en Machine Learning ha impulsado la necesidad de utilizar datasets más realistas que representen el comportamiento actual del tráfico de red. Durante varios años, conjuntos de datos como KDD99 y NSL-KDD fueron ampliamente utilizados en investigaciones de ciberseguridad; sin embargo, diferentes estudios señalaron limitaciones relacionadas con redundancia de registros y baja representación de ataques modernos. Como respuesta a estas limitaciones, el Canadian Institute for Cybersecurity desarrolló el dataset CIC-IDS2017, considerado actualmente uno de los conjuntos de datos más utilizados para investigaciones relacionadas con detección de intrusiones y aprendizaje automático aplicado a ciberseguridad (9,10).

El dataset CIC-IDS2017 está formado por tráfico benigno y múltiples tipos de ataques los cuales fueron generados bajo condiciones similares a entornos reales, incluyendo DDoS, PortScan, Botnet, Brute Force e Infiltration. Además, integra características relacionadas con flujo de paquetes, volumen de transmisión y comportamiento estadístico del tráfico, esto permite evaluar de manera más precisa el desempeño de modelos inteligentes en escenarios complejos y altamente dinámicos similares a los reales(11,12). Además, varios estudios mencionan que los IDS tradicionales basados de en firmas y reglas presentan limitaciones frente a ataques DDoS o PortScan en redes de alta velocidad debido a las grandes cantidades de conexión simultaneas, variabilidad de patrones de tráfico y dificultad de diferenciar actividades maliciosas

En el caso de los ataques PortScan representan un reto para los IDS porque el atacante puede enviar miles de solicitudes a diferentes puertos en pocos segundos, generando un tráfico difícil de diferenciar del comportamiento normal de la red. En redes de alta velocidad, esto puede provocar retrasos en el análisis, pérdida de paquetes y errores en la detección (13).

De manera similar, los ataques DDoS generan grandes volúmenes de tráfico simultáneo que saturan la red y dificultan el análisis en tiempo real. Además, cuando el tráfico malicioso se parece al tráfico legítimo, los métodos tradicionales presentan mayores limitaciones para detectar

comportamientos anómalos, especialmente en redes modernas con alto ancho de banda y numerosos dispositivos conectados (14).

El preprocesamiento y la selección de características son otros aspectos importantes dentro de los sistemas IDS además del modelo de clasificación. De acuerdo con diferentes investigaciones se han evidenciado que la calidad de los datos utilizados influye directamente en la capacidad predictiva de los algoritmos de clasificación durante el entrenamiento. Por lo que procesos como limpieza de variables, imputación de datos faltantes, eliminación de valores inconsistentes y análisis de correlación permiten mejorar la estabilidad de los modelos y reducir errores de clasificación. Asimismo, la identificación de variables relevantes permite el reconocimiento de patrones asociados a actividades maliciosas dentro del tráfico de red (15–18).

Asimismo, investigaciones más recientes han demostrado que el tráfico de red contiene relaciones complejas entre diferentes múltiples variables, especialmente entre métricas asociadas al tamaño de paquetes, volumen de flujo y tiempos de transmisión. Este comportamiento favorece el uso de modelos capaces de manejar relaciones no lineales y estructuras multidimensionales, particularmente algoritmos basados en ensamblaje y técnicas híbridas de aprendizaje automático y aprendizaje profundo (19,20).

Aunque han sido significativos los avances con Machine Learning aplicado a ciberseguridad, todavía existen desafíos relacionados con reducción de falsos positivos, procesamiento eficiente de grandes volúmenes de datos y detección de ataques emergentes. Por esta razón, es importante continuar realizando estudios comparativos que permitan evaluar el comportamiento de diferentes algoritmos bajo condiciones experimentales similares y datasets modernos como CIC-IDS2017 (21,22).

Por esta razón, la presente investigación tuvo como objetivo comparar el rendimiento de los algoritmos Random Forest, Support Vector Machine y Naive Bayes en la detección de intrusiones utilizando tráfico BENIGN, DDoS y PortScan del dataset CIC-IDS2017. Para lo cual, se evaluaron métricas de desempeño como exactitud, precisión, recall y F1-score, además del análisis de importancia de variables y correlaciones entre características del tráfico de red, con el propósito de identificar el modelo con mejor capacidad de clasificación y aportar evidencia técnica sobre la aplicación de Machine Learning en sistemas IDS modernos

MATERIALES Y MÉTODOS

La investigación se desarrolló bajo un enfoque cuantitativo de tipo experimental y comparativo, enfocado en el análisis del rendimiento de modelos de aprendizaje automático aplicados a la detección de intrusiones en redes de datos. El estudio tuvo como objetivo comparar el desempeño de los algoritmos Random Forest, Support Vector Machine (SVM) y Naive Bayes utilizando el dataset CIC-IDS2017, utilizado en investigaciones relacionadas con sistemas de detección de intrusiones y ciberseguridad.

Para el desarrollo de la investigación se utilizaron los archivos “Monday-WorkingHours.pcap_ISCX.csv”, “Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv” y “Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv”, correspondientes al dataset CIC-IDS2017. Los archivos fueron integrados mediante el lenguaje de programación Python utilizando la librería Pandas, obteniéndose un conjunto de datos final compuesto por 1 042 130 registros y 79 características relacionadas con el comportamiento del tráfico de red.

Posteriormente, con el propósito de reducir el costo computacional asociado al entrenamiento de los modelos de aprendizaje automático se realizó un muestreo aleatorio del conjunto de datos. Para ello, se seleccionó una muestra de 300 000 registros utilizando la función sample() de Pandas con una semilla aleatoria de 42, garantizando reproducibilidad experimental. Luego del muestreo, se efectuó el reinicio de índices del conjunto de datos mediante reset_index().

En la etapa de preprocesamiento se realizó primeramente la limpieza de nombres de columnas utilizando la función str.strip(). Luego, se separaron las variables predictoras y la variable objetivo, considerando la columna “Label” como atributo de clasificación. Las variables fueron transformadas a formato numérico mediante la función pd.to_numeric() utilizando el parámetro errors='coerce'. Además, se reemplazaron valores infinitos positivos y negativos por valores nulos utilizando NumPy con la finalidad de evitar inconsistencias durante el entrenamiento de los modelos.

Para los valores faltantes se los trató utilizando la herramienta SimpleImputer de la librería Scikit-learn aplicando la estrategia de imputación basada en la mediana (strategy='median'). Esta acción permitió completar valores faltantes y mantener la estabilidad estadística del conjunto de datos antes de la fase de entrenamiento.

Posteriormente, el dataset fue dividido en conjuntos de entrenamiento y prueba mediante la función train_test_split() de Scikit-learn. El 70 % de los registros se destinó al entrenamiento de los modelos y el 30 % restante a la validación y evaluación experimental. Además, se aplicó estratificación sobre la variable objetivo utilizando el parámetro stratify=y, con el propósito de conservar la proporción original de las clases BENIGN, DDoS y PortScan durante la partición de datos. Como resultado, el conjunto de entrenamiento quedó conformado por 210 000 registros y el conjunto de prueba por 90 000 registros.

El primer algoritmo evaluado fue Random Forest, para este algoritmo se utilizó la clase RandomForestClassifier de Scikit-learn utilizando 100 árboles de decisión (n_estimators=100) y una semilla aleatoria de 42 (random_state=42). El modelo se entrenó aplicando el conjunto de entrenamiento completo para ser evaluado posteriormente sobre el conjunto de datos de prueba. Para la evaluación se consideró métricas de exactitud (accuracy), precisión (precision), sensibilidad (recall) y F1-score utilizando las funciones accuracy_score() y classification_report(). Posteriormente, se generó la matriz de confusión mediante confusion_matrix() y se representó gráficamente utilizando las librerías Matplotlib y Seaborn para un análisis profundo.

El segundo algoritmo analizado fue Naive Bayes utilizando el clasificador GaussianNB de Scikit-learn. El modelo fue entrenado utilizando las mismas particiones de datos empleadas en Random

Forest y posteriormente evaluado mediante las mismas métricas de desempeño y análisis matricial de clasificación. Asimismo, se generó la correspondiente matriz de confusión para analizar el comportamiento del modelo frente a las diferentes categorías de tráfico de red.

Considerando la alta complejidad computacional que presentan los modelos SVM al trabajar con grandes volúmenes de datos y múltiples variables, se decidió utilizar un subconjunto representativo del dataset para el entrenamiento y evaluación del modelo mediante la clase SVC utilizando un kernel radial basis function (kernel='rbf') y una semilla aleatoria de 42. Esta decisión se apoya en estudios previos que señalan que el costo computacional para el algoritmo SVM incrementa conforme aumenta el número de registros, especialmente cuando se emplean kernels no lineales como RBF, afectando el tiempo de entrenamiento y el consumo de memoria (23).

Diversas investigaciones relacionadas con clasificación de datos a gran escala han utilizado muestras representativas o técnicas de reducción de datos para mantener la viabilidad computacional del modelo sin generar cambios significativos en su capacidad predictiva (23,24). Asimismo, se ha reportado que este tipo de estrategias permite conservar los patrones más relevantes del conjunto original, siempre que se mantenga una distribución adecuada de clases y métricas de evaluación consistentes durante el proceso comparativo (24).

Considerando lo anteriormente mencionado, el modelo SVM fue entrenado con 20 000 registros y evaluado con 5 000 registros, procurando siempre conservar la representatividad del dataset CIC-IDS2017. Cabe mencionar que esta decisión no alteró el enfoque metodológico, debido a que el estudio se orienta a comparar el desempeño predictivo y la capacidad de detección de intrusiones de los algoritmos evaluados bajo condiciones técnicamente viables de ejecución, y no a analizar tiempos de procesamiento o escalabilidad computacional. Finalmente, el modelo fue evaluado con las mismas métricas aplicadas a los demás algoritmos, con el fin de garantizar la consistencia del análisis experimental.

Finalmente, se llevó a cabo análisis de importancia de variables medinate el atributo feature_importances_ del modelo Random Forest, lo que permitió identificar las características con mayor influencia en la detección de intrusiones. Las diez variables más relevantes fueron representadas mediante gráficas de barras elaboradas en Matplotlib. De igual manera se construyeron matrices de correlación con la función corr() de Pandas y mapas de calor generados con Seaborn con el propósito de identificar relaciones lineales entre las variables de análisis del dataset y dependencias estructurales entre atributos del tráfico de red.

Todas las pruebas experimentales se desarrollaron en el entorno Google Colab utilizando Python como lenguaje principal de programación. Las librerías utilizadas fueron Pandas y NumPy para procesamiento de datos, Scikit-learn para implementación de modelos de aprendizaje automático y Matplotlib junto con Seaborn para visualización y análisis gráfico de resultados.

RESULTADOS

Para el desarrollo experimental se integraron los archivos que contenían tráfico BENIGN, ataques DDoS y ataques PortScan del dataset CIC-IDS2017. Luego de la integración de los datos se obtuvo un nuevo dataset conformado por 1 042 130 registros y 79 características relacionadas con el comportamiento del tráfico de red. Posteriormente, se aplicó un muestreo aleatorio de 300 000 registros para reducir el costo computacional y facilitar el entrenamiento de los modelos de aprendizaje automático. Luego del proceso de limpieza, conversión numérica e imputación de valores faltantes, los datos fueron divididos en conjuntos de entrenamiento y prueba para el desarrollo de los experimentos. A continuación, se presentan los resultados por cada modelo:

Rendimiento del modelo Random Forest

Los resultados obtenidos evidenciaron el mejor desempeño entre todos los modelos evaluados. Como se observa en la **Tabla 1**, el modelo alcanzó una exactitud de 99.98 %, obteniendo valores cercanos a 1.00 en precisión, recall y F1-score para las tres clases analizadas: BENIGN, DDoS y PortScan.

Tabla 1. Reporte de clasificación del modelo Random Forest.

Clase	Precision	Recall	F1-Score	Support
BENIGN	1.00	1.00	1.00	65210
DDoS	1.00	1.00	1.00	11063
PortScan	1.00	1.00	1.00	13727
Accuracy			1.00	90000
Macro Avg	1.00	1.00	1.00	90000
Weighted Avg	1.00	1.00	1.00	90000

El comportamiento del modelo puede observarse con mayor claridad en la **Figura 1**, correspondiente a la matriz de confusión. En esta representación se evidencia que prácticamente todos los registros fueron clasificados correctamente. Únicamente se registró un error de clasificación del tráfico BENIGN hacia la categoría DDoS y once registros DDoS fueron clasificados como tráfico BENIGN. Por su parte, la clase PortScan fue reconocida correctamente en todos los casos evaluados.

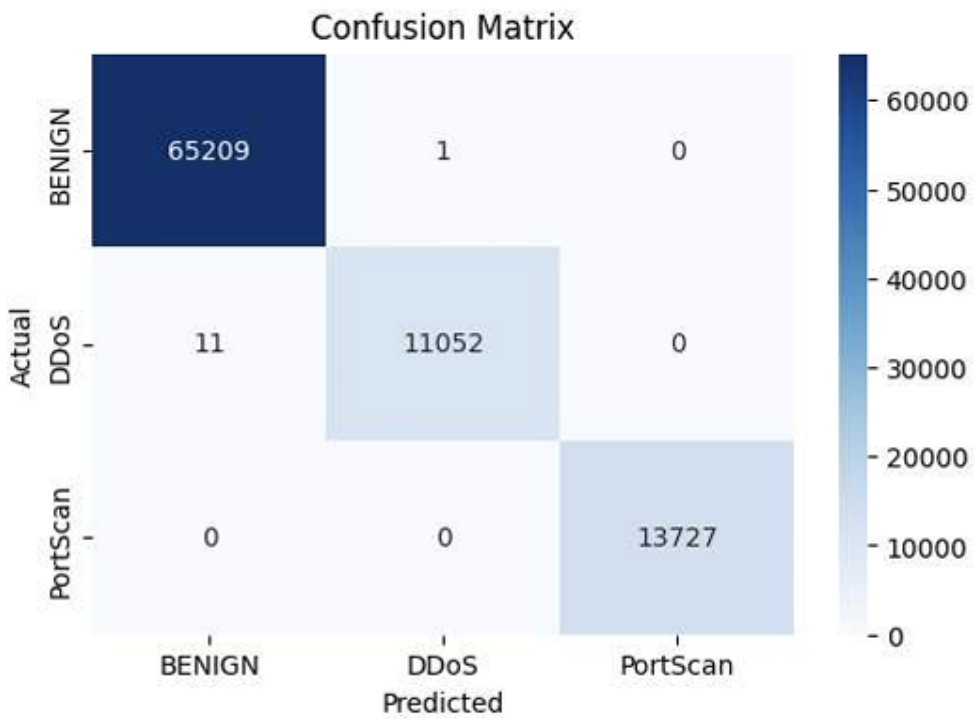

Figura 1. Matriz de confusión Random Forest.

Estos resultados evidencian la elevada capacidad del modelo para reconocer patrones complejos dentro del tráfico de red, manteniendo una clasificación estable tanto para tráfico legítimo como para tráfico malicioso. Además, la baja presencia de falsos positivos y falsos negativos evidencia un comportamiento robusto ante las diferentes categorías analizadas.

Rendimiento del modelo Naive Bayes

Naive Bayes presentó un desempeño inferior respecto a Random Forest. Como se muestra en la **Tabla 2**, el modelo obtuvo una exactitud de 44.62 %, esto quier decir que le modelo tiene limitaciones para clasificar los diferentes tipos de tráfico de red.

Tabla 2. Reporte de clasificación del modelo Naive Bayes.

Clase	Precision	Recall	F1-Score	Support
BENIGN	1.00	0.25	0.40	65210
DDoS	0.70	0.94	0.80	11063
PortScan	0.23	0.99	0.38	13727
Accuracy			0.45	90000
Macro Avg	0.64	0.73	0.52	90000
Weighted Avg	0.84	0.45	0.44	90000

Aunque el modelo logró identificar gran parte de los ataques DDoS y PortScan, presentó dificultades importantes para reconocer correctamente el tráfico BENIGN. Esto puede observarse en la **Figura 2**, donde la matriz de confusión evidencia que una gran cantidad de registros benignos fueron clasificados erróneamente como PortScan.

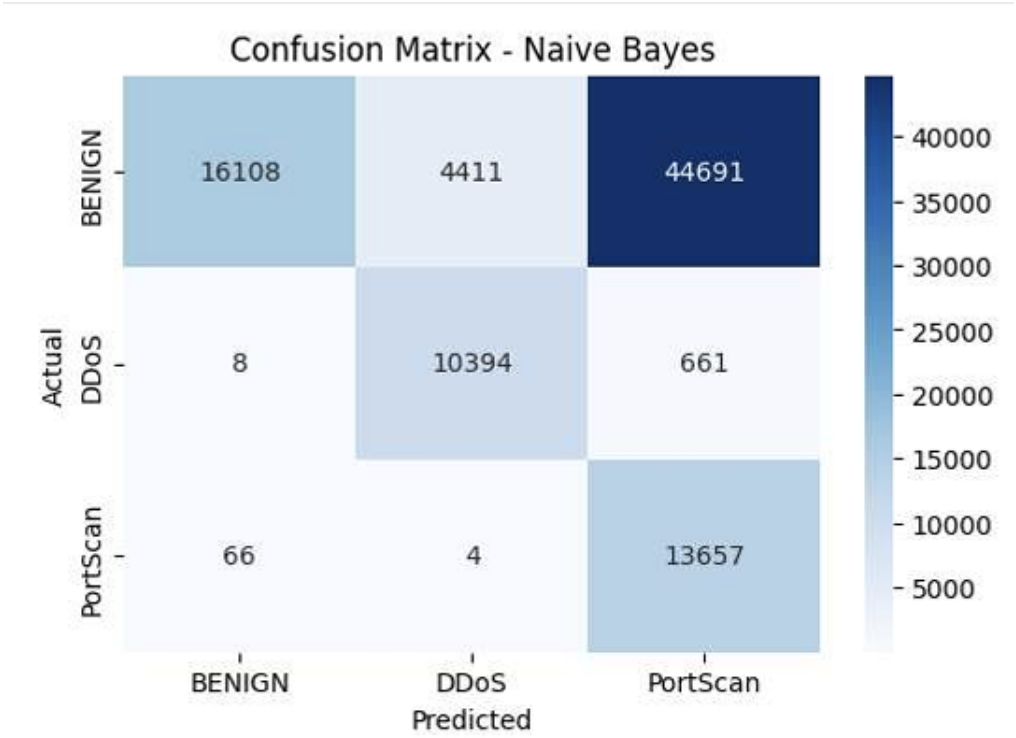

Figura 2.Matriz de confusión Naive Bayes.

De acuerdo con la **Figura 2,** 44 691 registros BENIGN se clasificaron como PortScan, esto generó una gran cantidad de falsos positivos. A pesar de esto, el modelo mostró valores altos de recall para DDoS y PortScan, lo que indica que este modelo tiene mayor facilidad para identificar tráfico malicioso que tráfico legítimo. Este comportamiento evidencia limitaciones que tiene Naive Bayes al momento de trabajar con variables relacionadas y dependencias internas complejas.

Rendimiento del modelo Support Vector Machine

El modelo Support Vector Machine presentó un desempeño intermedio respecto a los demás algoritmos evaluados. Como se observa en la **Tabla 3**, la exactitud alcanzada fue de 80.74 %

Tabla 3. Reporte de clasificación del modelo Support vector machine.

Clase	Precisión	Recall	F1-Score	Support
BENIGN	0.79	0.99	0.88	3598
DDoS	0.92	0.79	0.85	609
PortScan	0.00	0.00	0.00	793

Clase	Precisión	Recall	F1-Score	Support
Accuracy			**0.81**	**5000**
Macro Avg	**0.57**	**0.59**	**0.58**	**5000**
Weighted Avg	**0.68**	**0.81**	**0.74**	**5000**

El análisis de la **Figura 3** permitió observar que el modelo clasificó correctamente la mayoría de los registros BENIGN y DDoS. Sin embargo, presentó dificultades importantes para reconocer ataques PortScan.

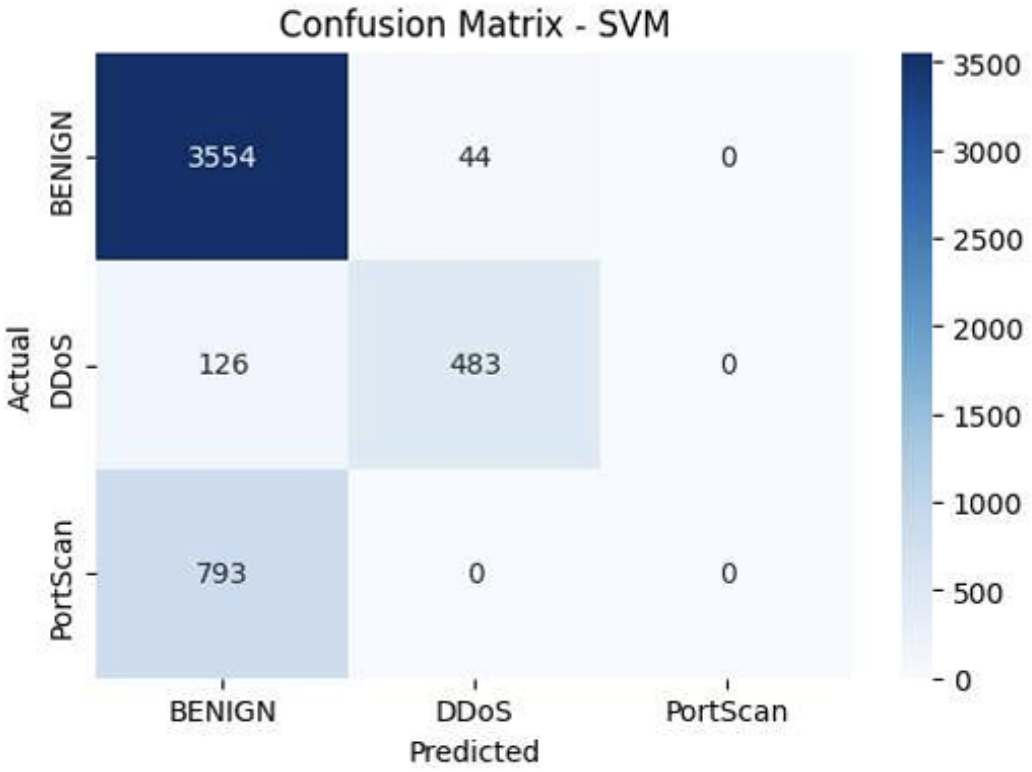

Figura 3.Matriz de confusión Support vector machine.

La matriz de confusión evidenció que 793 instancias de pertenecientes a PortScan se clafificaron como tráfico BENIGN. Esto generó los valores nulos obtenidos en precisión, recall y F1-score. Estos resultados evidencian que el modelo no logró generar una clasificación adecuada para este tipo de tráfico malicioso.

Comparación global de modelos

La **Figura 4** presenta la comparación general de desempeño entre los algoritmos evaluados.

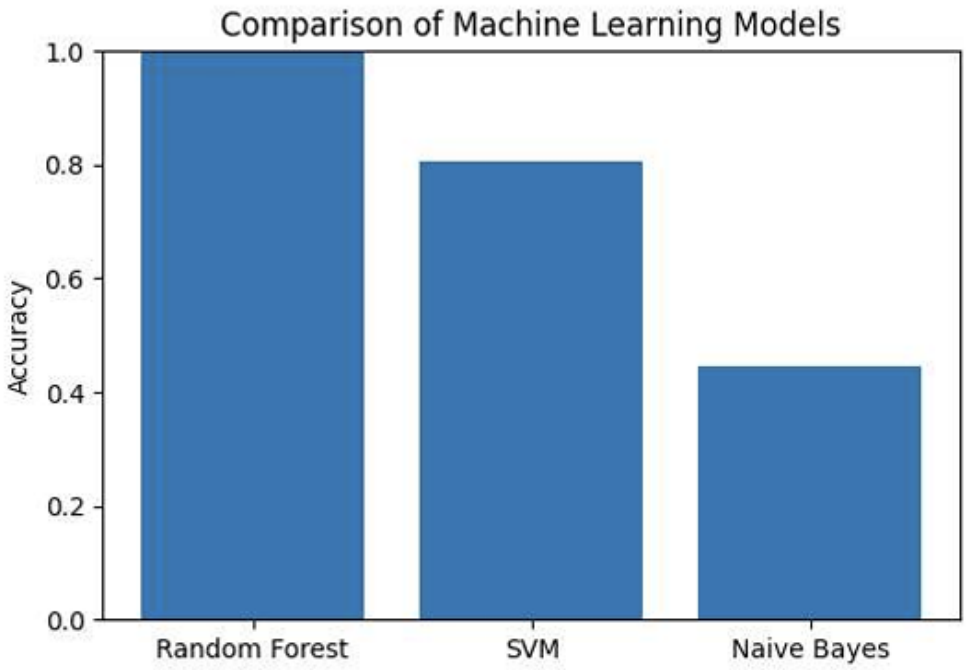

Figura 4. Comparación de los modelos de Machine Learnig.

La **Figura 4** permite concluir que Random Forest alcanzó el mejor rendimiento global, teniendo una exactitud cercana al 99.98 %, superando a SVM con 80.74 % y a Naive Bayes con 44.62 %. Estos resultados permiten incidir que se logran detecciones más precisas y estables con los modelos basados en ensamblaje por su capacidad para procesar datasets complejos como CIC-IDS2017

Importancia de características

El análisis de importancia de variables generado mediante el modelo Random Forest permitió identificar las características con mayor influencia en el proceso de clasificación. Como se observa en la **Figura 5**, las variables más relevantes fueron *Subflow Fwd Bytes*, *Bwd Packets/s* y *Average Packet Size*.

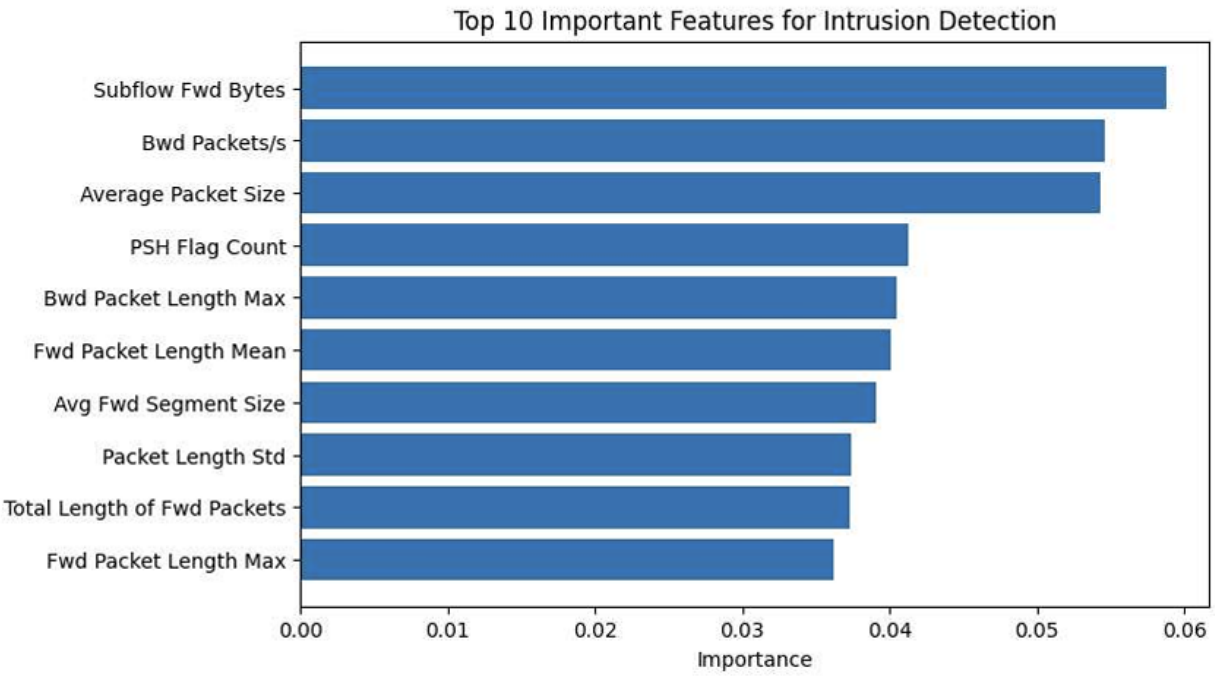

Figura 5. Top 10 características más importantes para la detección de intrusiones.

Los resultados muestran que las métricas relacionadas con volumen de tráfico, frecuencia de paquetes y tamaño promedio poseen una importante capacidad discriminativa para diferenciar ttráfico legítimo y tráfico malicioso. Asimismo, variables asociadas al flujo de paquetes y longitud de transmisión también presentaron una contribución significativa dentro del modelo.

Análisis de correlación de variables

La **Figura 6** presenta la matriz de correlación general del dataset CIC-IDS2017. En ella se identificaron múltiples relaciones lineales entre variables relacionadas con longitud de paquetes, tamaño de segmentos y métricas de flujo de tráfico.

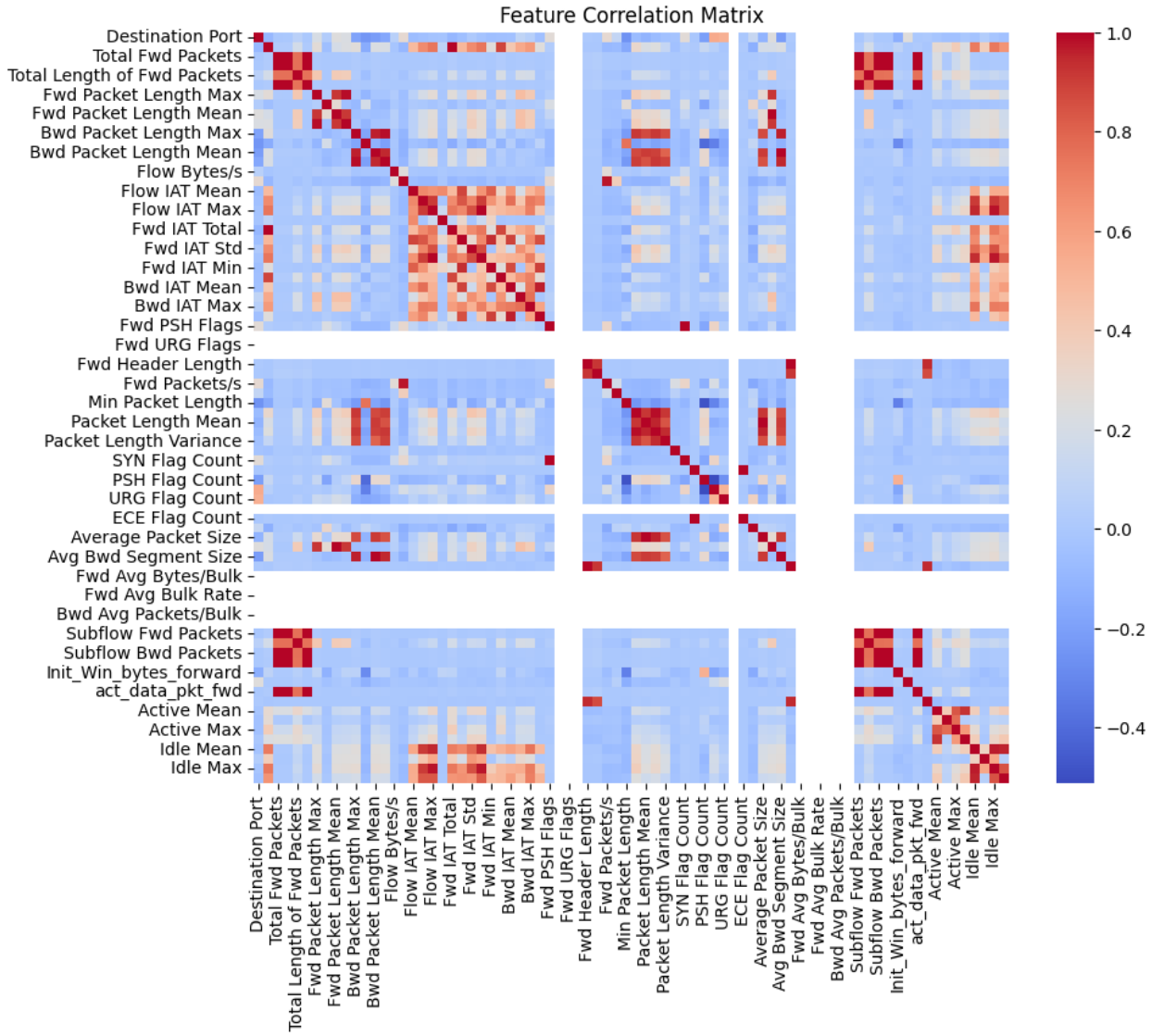

Figura 6. Matriz de correlación de características del dataset CIC-IDS2017.

Correlación de características

La **Figura 7** muestra la correlación existente entre las variables más relevantes identificadas por Random Forest.

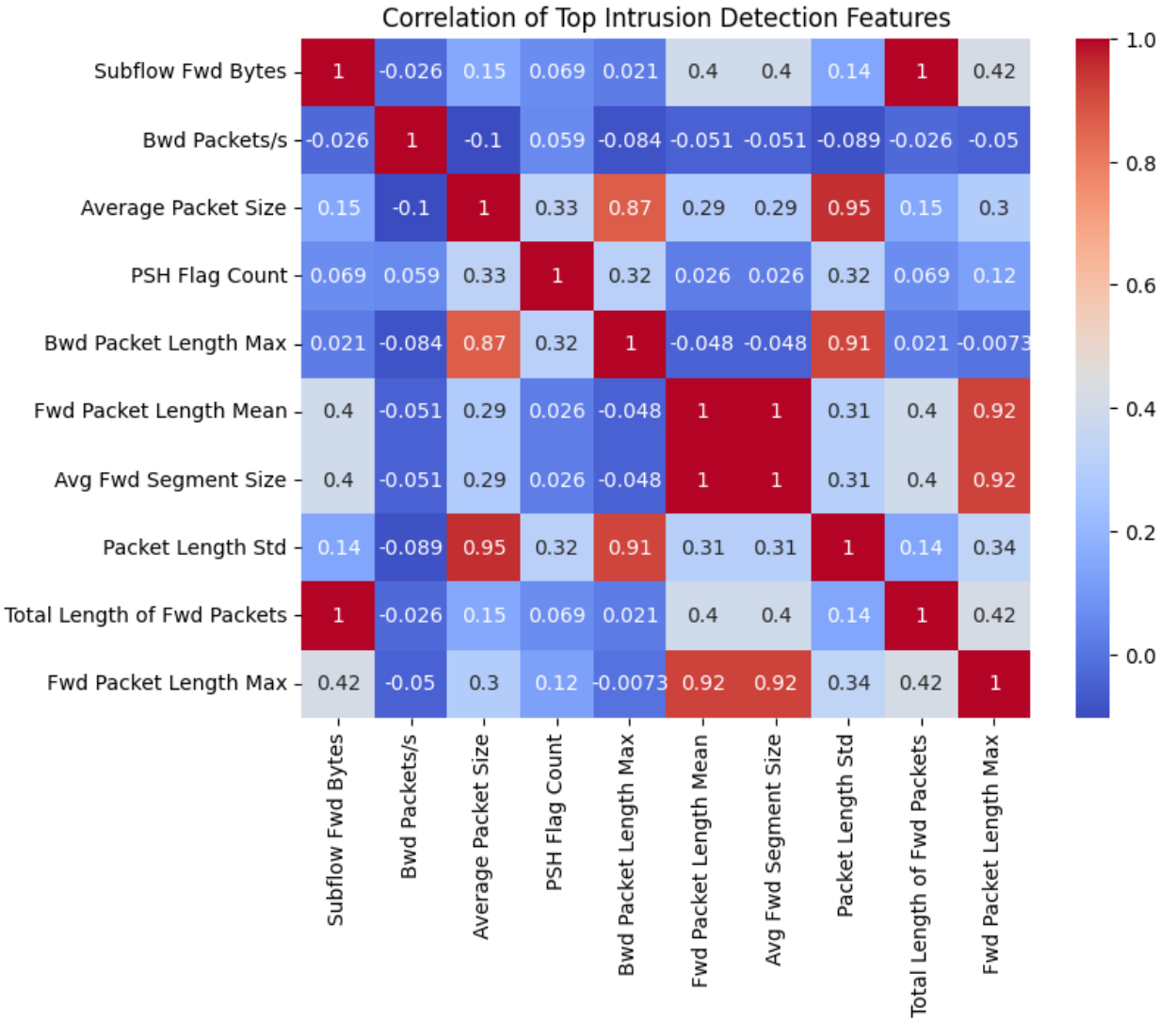

Figura 7. Matriz de correlación de las características más relevantes.

En esta figura se puede observar correlaciones existentes entre variables como *Fwd Packet Length Mean* y *Avg Fwd Segment Size*, así como asociaciones de características entre *Average Packet Size* y *Bwd Packet Length Max*. Estos resultados muestran la existencia de dependencias internas entre las características del tráfico de red, esto significa que Random Forest es un modelo apropiado para estos casos.

DISCUSIÓN

Los resultados obtenidos durante la investigación permitieron identificar diferencias claras en el comportamiento de los modelos de aprendizaje automático utilizados para la detección de intrusiones utilizando el dataset CIC-IDS2017. Entre los algoritmos evaluados, Random Forest destacó ampliamente al alcanzar una exactitud cercana al 99.98 %, superando de manera considerable a Support Vector Machine y Naive Bayes. Este resultado permite identificar la

capacidad que poseen los modelos basados en ensamblaje para trabajar con conjuntos de datos complejos, altamente dimensionales y con múltiples patrones de tráfico de red. Además, el comportamiento observado coincide con investigaciones recientes que reconocen a Random Forest como uno de los algoritmos más robustos para sistemas IDS, principalmente por su capacidad para manejar grandes volúmenes de información y reducir problemas de sobreajuste durante la clasificación (25).

El rendimiento obtenido por Random Forest evidenció una notable capacidad para diferenciar tráfico benigno y tráfico malicioso. La matriz de confusión mostró únicamente errores mínimos de clasificación, especialmente en las categorías BENIGN y DDoS, mientras que la clase PortScan fue identificada correctamente en todos los registros evaluados. Este comportamiento se lo puede relacionar con la estructura interna del algoritmo, ya que está basada en múltiples árboles de decisión que trabajan de manera conjunta para identificar relaciones complejas entre las variables del tráfico de red. Gracias a ello, el modelo logró adaptarse de forma eficiente a patrones no lineales y mantener una clasificación estable incluso en un entorno de datos altamente variable. Estudios previos desarrollados sobre CIC-IDS2017 han reportado comportamientos similares, especialmente en investigaciones donde se analizan métricas relacionadas con flujo de paquetes, tamaño de transmisión y características temporales del tráfico (26).

Otro hallazgo importante estuvo relacionado con la relevancia de determinadas variables dentro del proceso de clasificación. El análisis de importancia de características permitió identificar que atributos como Subflow Fwd Bytes, Bwd Packets/s y Average Packet Size tuvieron una influencia significativa en el comportamiento del modelo. Esto sugiere que las variables asociadas al flujo de tráfico y al comportamiento estadístico de los paquetes contienen información especialmente útil para diferenciar actividades normales de comportamientos maliciosos. Además, los resultados obtenidos muestran que la correcta selección de variables continúa siendo un aspecto fundamental para mejorar la precisión y eficiencia de los sistemas de detección de intrusiones basados en aprendizaje automático (27).

Por el contrario, Naive Bayes presentó el rendimiento más bajo entre los modelos evaluados, alcanzando una exactitud aproximada del 44.62 %. Aunque el algoritmo logró identificar gran parte del tráfico asociado a ataques DDoS y PortScan, tuvo dificultades para reconocer correctamente el tráfico BENIGN lo que generó una elevada cantidad de falsos positivos. Este comportamiento puede explicarse por la composición probabilística del modelo y por el supuesto de independencia entre variables que caracteriza a Naive Bayes. En datasets complejos como CIC-IDS2017, muchas características del tráfico de red presentan relaciones y dependencias internas, situación que limita la predicciám de este tipo de algoritmos. Los mapas de correlación obtenidos durante la investigación confirmaron precisamente la existencia de asociaciones significativas entre múltiples variables relacionadas con tamaño de paquetes, flujo de bytes y segmentos promedio de transmisión.

Por otra parte, los resultados obtenidos con Support Vector Machine mostraron un comportamiento intermedio respecto a los demás modelos evaluados. El algoritmo alcanzó una exactitud de 80.74 %, logrando clasificar correctamente la mayoría de las instancias BENIGN y

DDoS. Sin embargo, presentó una limitación importante en la detección de ataques PortScan, debido a que no logró identificar correctamente ninguna instancia perteneciente a esta categoría. Este comportamiento posiblemente estuvo influenciado por la reducción del conjunto de entrenamiento utilizada durante el proceso experimental como consecuencia del elevado costo computacional que representa SVM en datasets extensos y de alta dimensionalidad. Aunque Support Vector Machine continúa siendo un modelo ampliamente utilizado en tareas de clasificación, su rendimiento puede verse afectado cuando trabaja con conjuntos de datos complejos y altamente variables, especialmente en problemas multiclase relacionados con tráfico de red (28).

Las matrices de correlación permitieron comprender la estructura interna del dataset CIC-IDS2017 ya que se identificaron relaciones lineales entre variables como Fwd Packet Length Mean y Avg Fwd Segment Size, además de asociaciones relevantes entre Average Packet Size y Bwd Packet Length Max.

Asimismo, los resultados obtenidos evidencian la importancia del preprocesamiento de datos. La limpieza de variables, el tratamiento de valores infinitos, la imputación de datos faltantes y la adecuada selección de registros permitieron construir un entorno experimental más estable y consistente previo al entrenamiento de los modelos. Estas acciones resultan importantes en investigaciones relacionadas con sistemas IDS basados en aprendizaje automático (29).

En relación con los resultados obtenidos se puede confirmar que Random Forest por su capacidad para clasificar correctamente tráfico benigno y diferentes tipos de ataques es una alternativa robusta y eficiente para la detección de intrusiones. Además, la investigación evidenció que para que el rendimiento mejore de los modelos de aprendizaje automático aplicados a entornos de ciberseguridad se requieren una adecuada selección de variables y el análisis estructural del tráfico de red

CONCLUSIONES

Los resultados obtenidos evidencian que el algoritmo Random Forest presentó el mejor desempeño para la detección de intrusiones al utilizarse el dataset CIC-IDS2017 para el entrenamiento de este modelo. Alcanzando una exactitud cercana al 99.98 %, esto muestra que el algoritmo presenta una alta capacidad para diferenciar tráfico benigno de tráfico malicioso incluso en un entorno con múltiples variables y patrones complejos de comunicación de red. Asimismo, los valores obtenidos en precisión, recall y F1-score reflejaron un comportamiento estable y consistente frente a las categorías BENIGN, DDoS y PortScan. Estos resultados respaldan la efectividad de los modelos basados en ensamblaje.

Por otra parte, se identificaron diferencias en el comportamiento de los algoritmos evaluados. Aunque Support Vector Machine alcanzó un rendimiento aceptable con una exactitud aproximada del 80.74 %, presentó limitaciones en la detección de ataques PortScan, evidenciando dificultades para adaptarse a determinadas características del tráfico. Al contrario, Naive Bayes registró el rendimiento más bajo, con una exactitud cercana al 44.62 %, debido a las limitaciones

de criterios de asociación, situación que no se ajusta a datasets complejos y altamente correlacionados como lo es CIC-IDS2017.

También se identificó que variables de tráfico como Subflow Fwd Bytes, Bwd Packets/s y Average Packet Size mostraron alta relevancia durante el proceso de detección de intrusiones, evidenciando que el análisis y selección correcta de atributos contribuyen a mejorar la eficiencia y precisión de los IDS basados en inteligencia artificial. Ya que se observaron dentro de la investigación que variables relacionadas con el volumen de tráfico, la longitud de paquetes y el comportamiento estadístico de transmisión influyen directamente en la capacidad de clasificación de los modelos de aprendizaje automático.

Asimismo, el análisis de correlación realizado sobre el dataset evidenció la existencia de relaciones importantes entre múltiples características del tráfico de red del dataset. Esto permitió determinar que los datos no son independientes, sino que presenta patrones de interacción entre variables. En estos casos, modelos que manejen relaciones no lineales y estructuras multidimensionales, como Random Forest son mejores para analizar datasets como CIC-IDS2017.

Finalmente, el estudio permitió confirmar que es importante el proceso de preprocesamiento de datos dentro del desarrollo de modelos de aprendizaje automático aplicados a ciberseguridad. Procesos como la limpieza de variables, tratamiento de valores inconsistentes, imputación de datos faltantes y preparación adecuada del conjunto de entrenamiento contribuyeron de forma directa a mejorar la estabilidad y desempeño de los modelos evaluados. Por todo lo anterior mencionado se puede decir que los resultados obtenidos evidencian que la integración de técnicas de Machine Learning en sistemas de detección de intrusiones constituye una alternativa eficiente y prometedora para fortalecer la protección de infraestructuras tecnológicas frente a amenazas y ataques informáticos cada vez más sofisticados.

REFERENCIAS BIBLIOGRÁFICAS

1. Alauthman M, Aslam N, Al-kasassbeh M, Khan S, Al-Qerem A, Choo KKR. A multiagent and machine learning based hybrid NIDS for intrusion detection. International Journal of Advanced Computer Science and Applications. 2021;12:858–868. doi: 10.14569/IJACSA.2021.0120843.
2. AlSharman SAD, Alenezi MN, Alabdulkarim A. A detailed inspection of machine learning based intrusion detection systems. IoT. 2024;5:34. doi: 10.3390/iot5040034.
3. Belavagi MC, Muniyal B. Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Comput Sci. 2016;89:117–123. doi: 10.1016/j.procs.2016.06.016.
4. Lucas TJ, De Figueiredo IS, Tojeiro CAC, De Almeida AMG, Scherer R, Brega JRF. A comprehensive survey on ensemble learning-based intrusion detection approaches in computer networks. IEEE Access. 2023;11:117699–117728. doi: 10.1109/ACCESS.2023.3328535.

5. Gu J, Lu S. An effective intrusion detection approach using SVM with naïve Bayes feature embedding. Comput Secur. 2021;103:102158. doi: 10.1016/j.cose.2020.102158.
6. Liu C, Gu Z, Wang J. A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning. IEEE Access. 2021;9:75729–75740. doi: 10.1109/ACCESS.2021.3080898.
7. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy. Madeira; 2018. p. 108–116.
8. Maseer ZK, Yusof R, Bahaman N, Mostafa SA, Foozy CFM. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access. 2021;9:22351–22370. doi: 10.1109/ACCESS.2021.3056614.
9. Doménech J, León O, Siddiqui MS, Pegueroles J. Evaluating and enhancing intrusion detection systems in IoMT: The importance of domain-specific datasets. Internet of Things. 2025;32:101631. doi: 10.1016/j.iot.2025.101631.
10. Al-Hawawreh M, Sitnikova E, Aboutorab N. The intrusion detection system by deep learning methods. International Arab Journal of Information Technology. 2022;19:529–538. doi: 10.34028/iajit/19/3A/10.
11. Abbas A, Khan MA, Latif S, Ajaz M, Shah AA, Ahmad J. A new ensemble-based intrusion detection system for internet of things. Arab J Sci Eng. 2022;47:1805–1819. doi: 10.1007/s13369-021-06086-5.
12. Zhao R, Mu Y, Zou L, Wen X. A hybrid intrusion detection system based on feature selection and weighted stacking classifier. IEEE Access. 2022;10:71414–71426. doi: 10.1109/ACCESS.2022.3187631.
13. Ahmim A, Maglaras L, Ferrag MA, Derdour M, Janicke H. A Novel Hierarchical Intrusion Detection System based on Decision Tree and Rules-based Models. 2018;
14. Chen Z, Yu W, Zhou L. ADASYN-Random Forest Based Intrusion Detection Model. 2022; doi: 10.1145/3483207.3483232.
15. Wisanwanichthan T, Thammawichai M. A double-layered hybrid approach for network intrusion detection system using combined naive Bayes and SVM. IEEE Access. 2021;9:138432–138450. doi: 10.1109/ACCESS.2021.3118024.
16. Goeschel K. Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. Proceedings of SoutheastCon. Norfolk; 2016.
17. Dina AS, Siddique AB, Manivannan D. Effect of balancing data using synthetic data on the performance of machine learning classifiers for intrusion detection in computer networks. IEEE Access. 2022;10:96731–96747. doi: 10.1109/ACCESS.2022.3205092.
18. Talukder MA, Islam MM, Uddin MA, Moni MA. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data. 2024;11:32. doi: 10.1186/s40537-024-00886-w.
19. Ahmetoğlu H, Kaçar S, Yıldırım S. Analysis of feature selection approaches in large scale network intrusion detection. 2020 28th Signal Processing and Communications Applications Conference (SIU). Gaziantep; 2020.

20. Adenusi DA, Adedoyin FF, Misra S, Damasevicius R. Data-driven network intrusion detection using optimized machine learning models. Front Appl Math Stat. 2025;11:100339. doi: 10.3389/fams.2025.100339.
21. Du J, Yang K, Hu Y, Jiang L. NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. IEEE Access. 2023;11:24808–24821. doi: 10.1109/ACCESS.2023.3251451.
22. Abid A, Khan MT, Iwendi C, Mittal M. Distributed deep learning approach for intrusion detection systems. Cyber-Physical Systems. 2023;9:275–291. doi: 10.1080/24751839.2023.2239617.
23. Nalepa J, Kawulok M. Selecting training sets for support vector machines: a review. Artif Intell Rev. 2019;52:857–900. doi: 10.1007/s10462-017-9611-1.
24. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing. 2020;408:189–215. doi: 10.1016/j.neucom.2019.10.118.
25. Hamidou ST, Mehdi A. Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks. Machine Learning with Applications. 2025;22:100738. doi: 10.1016/j.mlwa.2025.100738.
26. Chavan P V., Alone N V. Optimizing Intrusion Detection with Random Forest: A High-Accuracy Approach Using CIC-IDS 2017. Int J Comput Appl. 2025;187:17–22. doi: 10.5120/ijca2025924816.
27. Kurniabudi, Stiawan D, Darmawijoyo, Bin Idris MY, Bamhdi AM, Budiarto R. CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection. IEEE Access. 2020;8:132911–132921. doi: 10.1109/ACCESS.2020.3009843.
28. Hamidou ST, Mehdi A. Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks. Machine Learning with Applications. 2025;22:100738. doi: 10.1016/j.mlwa.2025.100738.
29. Saidane S, Telch F, Shahin K, Granelli F. Optimizing Intrusion Detection System Performance Through Synergistic Hyperparameter Tuning and Advanced Data Processing [Internet]. 2024. Available from: https://arxiv.org/abs/2408.01792.