ISBN: 978-628-96613-3-0. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2026.1.1.2690

Implementación Despliegue Web de un Sistema de Detección de Anomalías para Entornos IoT Mediante la metodología CRISP-ML

Andrés Felipe Yule[1], Juan Camilo Galeano Bucurú[2], Alexander Suarez Gomez[3]

Faculty Mentor: Fernando Gutiérrez Portela[1], Oscar Augusto Diaz Triana[2]

[1] INNOVASIC Research Seedbed, University Cooperative of Colombia, Calle 10 No. 1-120, Ibagué, Tolima; alexander.suarezg@campusucc.edu.co; andres.yule@campusucc.edu.co; juan.galeanob@campusucc.edu.co

[2] AQUA Research Group, University Cooperative of Colombia, Calle 10 No. 1-120, Ibagué, Tolima; fernando.gutierrez@campusucc.edu.co; oscar.diaz@campusucc.edu.co

Resumen*— El Internet de las Cosas (IoT) ha transformado la industria moderna mediante la monitorización y automatización en tiempo real, pero también ha incrementado el riesgo cibernético en implementaciones en el borde con recursos limitados. El artículo presenta la implementación y el despliegue web de un sistema ligero de detección de anomalías para entornos IoT, estructurado mediante el ciclo de vida Cross-Industry Standard Process for Machine Learning with Quality Assurance (CRISP-ML). Mediante entrenamientos y comparaciones con modelos no supervisados (IForest, OCSVM, K-means y Autoencoder) usando el conjunto de datos BoT-IoT, se validó el comportamiento operativo del sistema por medio de capturas de paquetes en tiempo real. Los resultados experimentales demuestran que el sistema logra identificar cambios significativos en el comportamiento del tráfico durante ataques como "TCP RST Flood", activando alertas dinámicas cuando el porcentaje de anomalías supera el umbral operativo definido. Desde una perspectiva práctica, los resultados evidencian que es posible desplegar modelos de detección de anomalías dentro de una plataforma web práctica que facilita el monitoreo de redes IoT en tiempo real. La interfaz desarrollada en Streamlit permite al usuario realizar captura manual y automática de paquetes, cargar archivos CSV para predicción, seleccionar modelos, visualizar métricas de agrupamiento y gráficos estadísticos (predicción, métricas, consumo de recursos computacionales). El enfoque contribuye metodológicamente al demostrar que los modelos de ML pueden integrarse en herramientas accesibles para usuarios no especializados, reduciendo la complejidad de los IDS en IoT. Como limitaciones del estudio tenemos el hecho de que la evaluación se hizo con el conjunto de datos BoT-IoT y escenarios controlados de ataque, lo cual generaría diferencias frente a entornos reales con mayor variabilidad de tráfico. Adicionalmente, la dependencia de TShark con el kernel del dispositivo para poder realizar las capturas de paquetes.

Palabras clave *— Internet de las cosas, detección de anomalías, sistema de detección de intrusiones, CRISP-ML, BoT-IoT*

I. INTRODUCCIÓN

La proliferación de las tecnologías del Internet de las Cosas (IoT) ha permitido la automatización y la monitorización continua en ámbitos como la agricultura, la logística, la manufactura y la infraestructura inteligente. Sin embargo, los ecosistemas del IoT están expuestos a ciberataques debido a la heterogeneidad de los dispositivos, las arquitecturas distribuidas y los limitados presupuestos computacionales y energéticos en el borde. Dichas limitaciones reducen la aplicabilidad de los Sistemas de Detección de Intrusiones (IDS) convencionales, que dependen de un procesamiento intensivo e infraestructuras centralizadas [1].

Los enfoques de IDS basados en anomalías pueden detectar patrones de ataque nunca vistos sin necesidad de firmas, pero su adopción en entornos reales de IoT depende de la eficiencia computacional, la simplicidad de implementación y la capacidad de mantenimiento a lo largo del tiempo [2].

A pesar de los avances en los sistemas de detección de intrusiones basados en el aprendizaje automático [1], [3], [4],[5] gran parte de las investigaciones están más enfocados en la evaluación de modelos en entornos experimentales o con data sets offline[1], [6]. Sin embargo, existe una limitada integración de modelos ML en plataformas prácticas que permitan su uso operativo en redes IoT con limitaciones de hardware [7]. En general, son escasas las soluciones que combinen captura de tráfico en tiempo real, inferencia de modelos y visualización dinámica en una interfaz accesible para el usuario final.

Para abordar el vacío, el presente estudio propone el diseño e implementación de un sistema ligero de detección de anomalías (IDS) para entornos IoT, integrando modelos de aprendizaje

automático no supervisados dentro de una interfaz web[3], [8]. El desarrollo del sistema se estructuró mediante la metodología CRISP ML, la cual permite guiar el ciclo completo desde la preparación de datos hasta el despliegue y monitoreo del modelo[9].

La importancia de la investigación se centra en el desarrollo de una solución práctica para el monitoreo de seguridad en IoT por medio de la integración de modelos de detección de anomalías con una interfaz web interactiva [3], [10], [11], [4]. Los resultados obtenidos aportan a la implementación de herramientas accesibles que facilitan el análisis del tráfico de red y la detección temprana de anomalías en sistemas con recursos limitados[12]. De tal forma, el estudio contribuye a reducir la brecha entre el desarrollo de modelos de ML y su aplicación en sistemas reales de detección de intrusiones.

Las contribuciones del estudio incluyen: (i) una secuencia de datos reproducible para su implementación; (ii) una evaluación comparativa en captura de paquetes reales con los modelos IForest, OCSVM y K-Means en BoT-IoT[12]; y (iii) una implementación en Streamlit que permite la selección interactiva de modelos, la visualización de alertas y métricas.

El resto del artículo se organiza de la siguiente manera: En la **Sección II** se describe la metodología utilizada para el desarrollo del sistema, basada en el ciclo de vida CRISP ML. En la **Sección III** se muestran los resultados experimentales obtenidos a partir de la evaluación de los modelos de detección de anomalías. Posteriormente en la **Sección III.I** se describe la implementación del sistema en escenarios de captura de tráfico y detección de anomalías en redes IoT.

II. METODOLOGÍA

La presente investigación adopta un enfoque cuantitativo y explicativo, estructurado bajo el marco CRISP-ML, con el propósito de garantizar un proceso sistemático, reproducible y orientado al despliegue en entornos reales [9]. A diferencia de enfoques centrados exclusivamente en la optimización del entrenamiento, este trabajo prioriza la validación operativa del sistema de detección de anomalías en escenarios de tráfico real en línea con tendencias recientes en sistemas IDS aplicados a IoT [1], [3].

El proceso de modelado se desarrolló previamente en un entorno experimental utilizando el conjunto de datos BoT-IoT, ampliamente empleado en la evaluación de sistemas de detección de intrusiones en entornos IoT [6], [8]. Este dataset, compuesto por múltiples archivos CSV, fue concatenado y aleatorizado para su procesamiento. Durante la fase de preparación de datos, se eliminaron atributos no informativos como direcciones IP y marcas temporales, y se realizó la conversión de variables categóricas y de tipo texto a representaciones numéricas. Posteriormente, se aplicó un pipeline de preprocesamiento compuesto por imputaciones de valores faltantes mediante estrategia constante, codificación ordinal para variables categóricas, estandarización de características y reducción de dimensionalidad mediante Análisis de Componentes Principales (PCA), fijando el número de componentes en seis con el objetivo de reducir la complejidad computacional, tal como se recomienda en escenarios con limitaciones de recursos [5], [13].

Adicionalmente, se realizó una eliminación manual de atributos de baja relevancia, particularmente banderas TCP e indicadores redundantes, con el fin de mejorar la eficiencia del sistema en entornos con recursos limitados, una práctica común en sistemas de detección aplicados a IoT [2], [4]. El conjunto final de datos se compone de variables numéricas con el comportamiento del tráfico (por ejemplo, tamaño de paquetes, puertos, tiempo entre paquetes) y variables categóricas asociadas al protocolo de red.

En la fase de modelado se implementaron múltiples algoritmos de detección de anomalías no supervisados, incluyendo Isolation Forest, One-Class SVM y K-Means, los cuales han demostrado ser efectivos para la identificación de patrones anómalos en tráfico de red sin necesidad de etiquetas [1], [8], [11]. Los modelos fueron configurados con hiperparametros específicos, tales como 300 estimadores y una tasa de contaminación del 10% para Isolation Forest, kernel RBF y v=0.05 para One-Class SVM, y 29 clusters para k-means. Cabe destacar que algunos modelos fueron entrenados exclusivamente con datos normales, siguiendo el enfoque clásico de detección de anomalías, mientras que otros utilizaron particiones estándar de entrenamiento y prueba (80/20), lo cual permite comparar distintos enfoques dentro de un mismo entorno experimental.

La evaluación de los modelos se realizó mediante métricas de agrupamiento como Silhouette Score, Calinski-Harabasz y Davies-Bouldin, ampliamente utilizadas para evaluar la calidad de clustering en contextos no supervisados [14]. Estas métricas fueron complementadas con matrices de confusión para analizar la capacidad de detección por tipo de ataque, siguiendo enfoques similares en estudios recientes de detección de anomalías en redes IoT [1], [15].

Los modelos se integraron en una aplicación web con Streamlit, facilitando la captura, análisis y visualización de resultados en tiempo real. Este tipo de implementación

facilita la transición de modelos teóricos a sistemas operativos, alineándose con propuestas recientes de IDS basadas en web para monitoreo en tiempo real [7], [10], [12].

A. Comprensión del negocio

Para comprender e identificar el problema y los requerimientos del negocio, se realizó una revisión sistemática de literatura en donde se evaluaron las limitaciones de los sistemas tradicionales de detección de intrusiones (IDS) específicamente en entornos IoT de baja potencia, los objetivos del sistema se definieron teniendo en cuenta aspectos como las restricciones computacionales, la necesidad de detección en tiempo real, escalabilidad y mantenibilidad del modelo, con ello se establecieron criterios de éxito basados en rendimiento de detección y eficiencia computacional.

B. Comprensión de los datos

(i) El entrenamiento para la validación fuera de línea se utilizó el conjunto de datos BoT-IoT, (ii) que contiene tráfico benigno y malicioso representativo de servicios de IoT; (iii) tipos de ataques como DoS/DDoS, reconocimiento y robo de información [5]. (i) La ingeniería de datos incluye la eliminación de duplicados, la gestión de valores faltantes, la estandarización de características y la selección de atributos informativos para facilitar una inferencia eficiente en dispositivos con restricciones[6].

C. Modelado

Se entrenaron cuatro modelos de detección de anomalías: ISolation Forest (IForest), K-means, One-Class SVM (OCSVM) y Autoencoder [3], se realizó ajuste de hiperparametros como n_estimators, n_neighbors, kernel y número de clústeres, con el fin de optimizar el rendimiento del modelo en condiciones de recursos limitados [1].

D. Evaluación

Cada uno de los modelos fue evaluado con un conjunto de pruebas independiente. Las métricas de rendimiento incluyeron precisión, recall, exactitud, F1-Score, AUC y tiempo de inferencia, los modelos IForest y Autoencoder ofrecen un equilibrio entre rendimiento de detección y eficiencia computacional[5].

E. Implementación y despliegue

La implementación de los modelos y el sistema se integró en una aplicación web desarrollada con Streamlit, la cual permitió la detección de anomalías en tiempo real junto con una visualización interactiva de usar en las capturas, predicciones, alertas, las métricas de rendimiento y el comportamiento de la red en IoT. El enfoque está alineado para implementar soluciones IDS en tiempo real basadas en la web, como el estudio de Tamuka y Sibanda [4].

La metodología no solo aporta que el IDS sea eficiente en la detección de anomalías, sino que también escalable y práctico para aplicaciones en tiempo real en entornos IoT con recursos limitados.

III. RESULTADOS EXPERIMENTALES

Los Modelos Isolation Forest (IForest), K-means, One-Class SVM y Autoencoder fueron entrenados, luego evaluados con conjuntos de datos de tráfico de red en formato CSV, con el fin de analizar su estabilidad y tasa de falsas alarmas. Dado a que los modelos son NO supervisados orientados a entornos IoT con recursos limitados, se priorizó la sensibilidad ante patrones atípicos, lo que en ocasiones puede generar cierto margen de error.

El análisis de desempeño se realizó bajo dos escenarios experimentales diferentes: el primero tiene que ver con las condiciones de tráfico normal (previas a la ejecución de los ataques), el segundo bajo situaciones de ataque **TCP RST Flood** simuladas, lo que permitió ver la capacidad de los modelos para diferenciar entre patrones legítimos y actividades maliciosas.

A. Modelo Isolation Forest (IForest):

En la figura 1.0 para la prueba de predicción con el modelo IForest se aprecia que clasificó un total de 470 paquetes entre ellos resultaron 313 paquetes como normales y 157 como anómalas. Aunque representa una proporción de detección superior a la que se esperaba en tráfico normal, su tiempo de inferencia fue bajo (0.0707 s), lo que lo hace una alternativa eficiente computacionalmente. Aquel tipo de comportamiento nos da un modelo sensible a variaciones en el tráfico, adecuado para la detección temprana en entornos donde no se pueden omitir posibles amenazas.

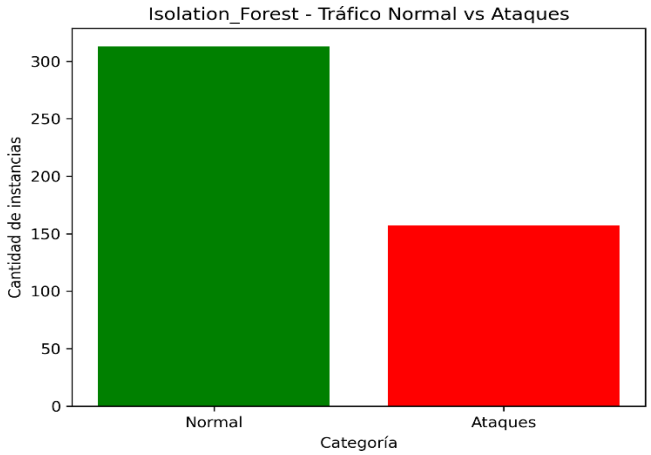


Figura 1.0: IForest gráfica Normal vs Ataque PRE-ATTACK

Después del ataque

El ataque ejecutado fue "TCP RST Flood" con el objetivo de evaluar la capacidad de respuesta de los modelos ante un caso real de intrusión. El IForest tomo un total de 5621 paquetes e identificó 269 paquetes como normales y 5352 como anómalos durante el ataque. Se evidencia una activación clara del mecanismo de detección, con una proporción mayoritaria de tráfico de red detectado como anómalo. Porcentualmente el volumen de detecciones supera el umbral del 80% establecido en la interfaz, lo cual activa el estado crítico del sistema.

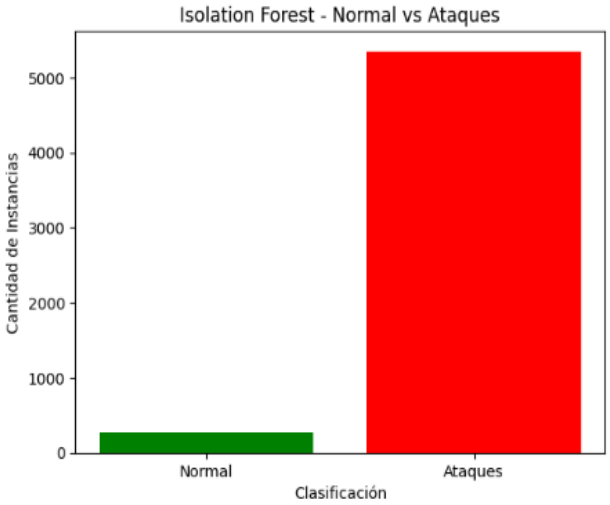


Figura 1.1: IForest gráfica Normal vs Ataque POST-ATTACK

Como se observó en la figura 1.1. La distribución de instancias detectadas, donde el tráfico normal predomina con aproximadamente 269 registros (barra verde), frente a 5352 instancias identificadas como ataques (barra roja).

B. Modelo One-Class SVM (OCSVM)

El modelo OCSVM muestra las siguientes métricas de clúster para los datos de tráfico de red normales [14], El gráfico muestra un total de 70 paquetes detectados como normales y 30 paquetes como anomalías, tiene un leve porcentaje de error, pero aun así está bien dentro de lo normal como se observa en la figura 2.0

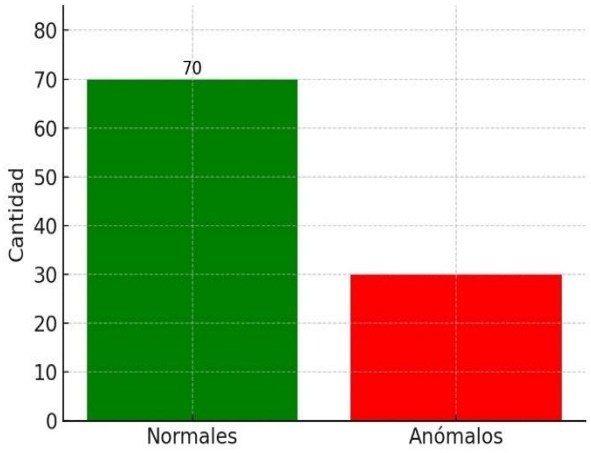


Figura 2.0: One-Class SVM (OCSVM) Normal vs Ataque PRE-ATTACK

Después del ataque

Los valores revelan una separación regular de los clústeres en el tráfico normal como se observa en la figura 2.0. Sin embargo, al ejecutar el ataque "TCP RST Flood", el modelo identificó todos los datos de tráfico como anomalías tal como se ve en la figura 2.1, En el caso posterior al ataque, no fue posible calcular las métricas de los clústeres dado a que solo había un único clúster anómalo [14], lo cual indica que el modelo clasificó correctamente el tráfico del ataque como no perteneciente al patrón normal.

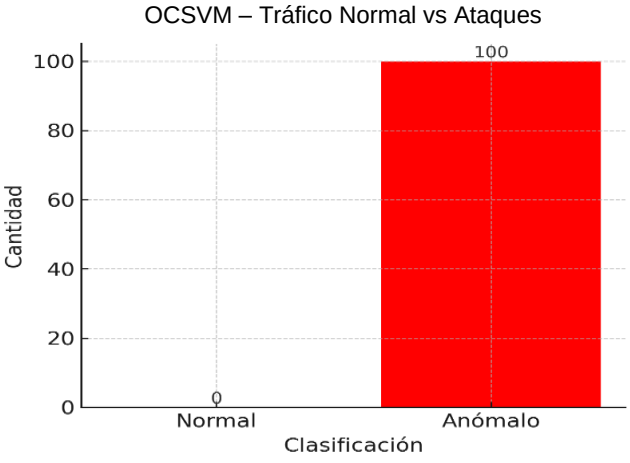


Figura 2.1: One-Class SVM (OCSVM) Normal vr Ataque POST-ATTACK

C. Modelo K-Means

En la figura 3.0 K-Means muestra una segmentación estructural sólida, evidenciada por los índices internos de clustering, clasificó 334 instancias como normal, pero identificó dos subconjuntos dentro del tráfico: 99 instancias de

DDOS_UDP y 37 de **Reconocimiento (Reconnaissance).** Dicho comportamiento refleja que el modelo detecta micro patrones diferenciados dentro del tráfico aparentemente normal.

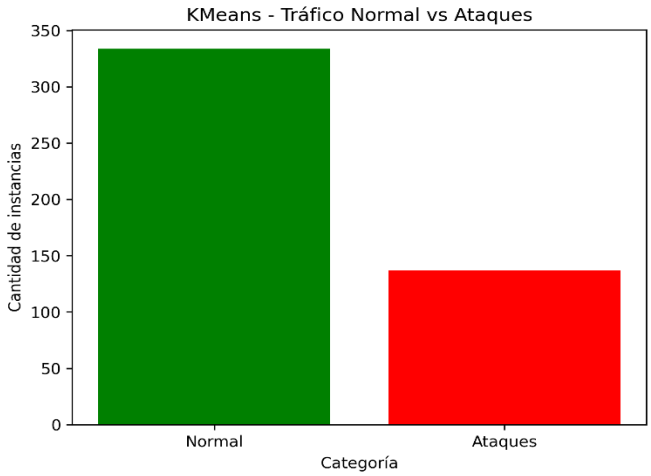


Figura 3.0: K-means – Tráfico Normal vs Ataques PRE-ATTACK

Después del ataque

Como se observa en la figura 3.0 K-Means clasificó 574 paquetes como normales y 5043 como Reconnaissance, además cuatro de los paquetes se identifican como DDoS_UDP. Aunque el modelo no usa etiquetas supervisadas tradicionales, la segmentación estructural ayudó a identificar un clúster claramente asociado al comportamiento anómalo inducido por el ataque TCP RST Flood.

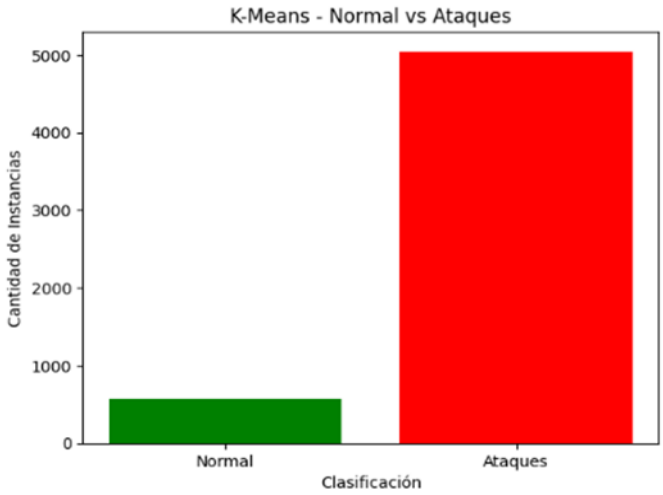


Figura 3.1: K-Means – Normal vs Ataque POST-ATTACK

D. Modelo Autoencoder

El modelo Autoencoder sugiere una alta capacidad de discriminación, aunque requiere validación adicional para descartar falsos positivos, clasificando 398 instancias como normales y únicamente 72 como Reconnaissance tal como se observa en la figura 4.0.

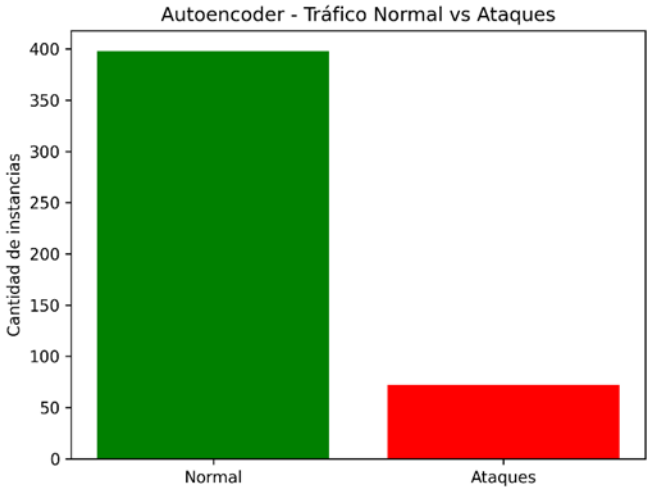
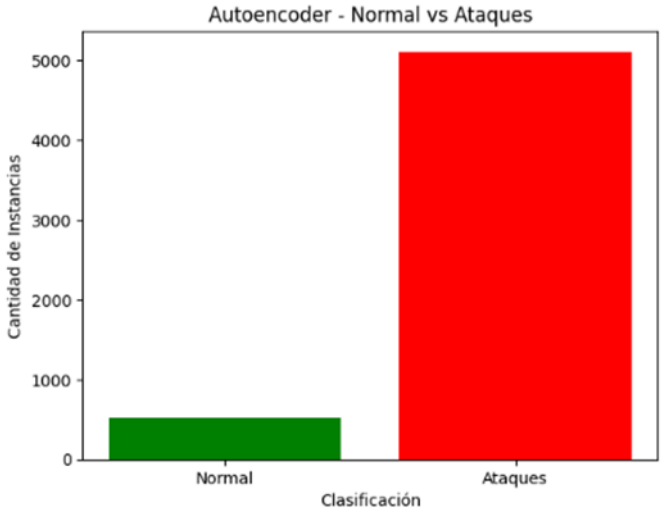


Figura 4.0: Autoencoder Tráfico Normal vs Ataque PRE-ATTACK

Después del ataque

En la figura 4.1 se observa que Autoencoder clasificó 514 paquetes como normales y 5107 como Reconnaissance en la primera prueba. dicho comportamiento nos demuestra una clara diferenciación entre el patrón de tráfico normal y el tráfico inyectado por el ataque “TCP RST Flood”.



Segunda prueba de ataque

En una segunda prueba se realizó el mismo ataque TCP RST Flood, pero generando la gráfica de distribución de detecciones con las etiquetas DDOS_TCP y Reconnaissance.

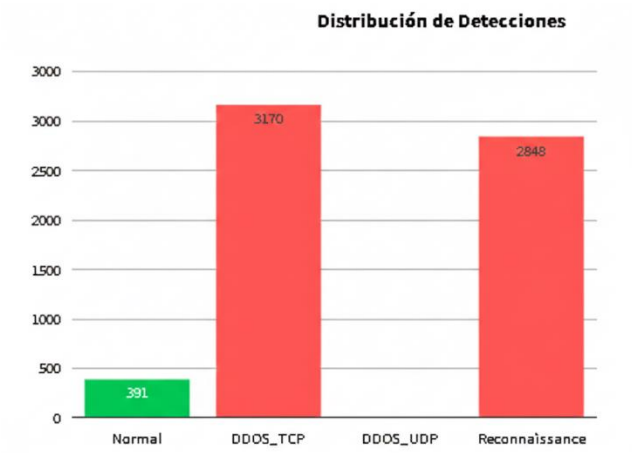


Figura 4.2: Autoencoder Normal vs Ataque POST-ATTACK 2

Autoencoder tomó 391 paquetes como normal, 3170 como denegación de servicios TCP y 2848 como Reconocimiento. Desde el punto de vista, el sistema de alertas implementado en Streamlit, el porcentaje de anomalías supera el 80%, lo cual activa el estado crítico de forma consistente, los resultados confirman que el modelo no solo detecta el ataque, sino que lo realiza con alta contundencia.

El umbral del 80% fue definido de manera empírica a partir del comportamiento observado durante las pruebas experimentales, buscando un equilibrio entre sensibilidad y reducción de falsas alarmas.

III.I. Implementación del sistema y experiencia del usuario

El sistema de detección de anomalías se ha implementado a través de una interfaz web desarrollada en Framework Streamlit, lo que facilita la interacción del usuario con los modelos de aprendizaje automático. En la página principal de "Inicio" como se observa en la figura 5, se muestra información relevante sobre la supervisión y detección de anomalías en tiempo real en redes IoT, aprovechando los modelos de Machine Learning (ML) para garantizar seguridad y fiabilidad [15].

La sección "**Panel de control de detección**" de la interfaz web que se puede ver en la figura 6, muestra la selección del modelo y métricas, un saludo de "Bienvenida" con los pasos para usar el aplicativo, también están los dos tipos de captura (Automática y manual), junto con la opción de cargar CSV y el botón para realizar predicción.

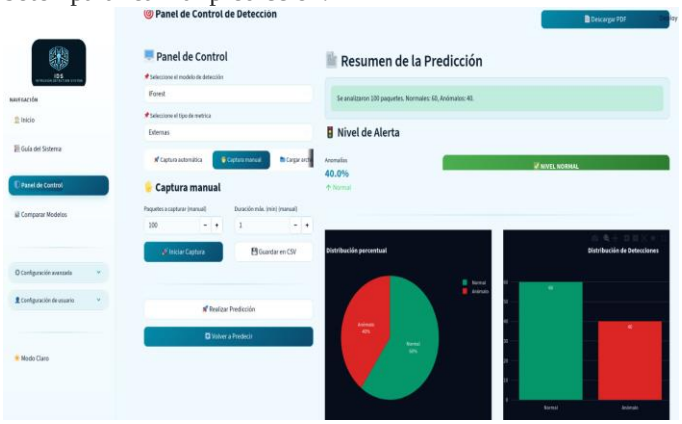


Figura 6: Panel de control de detección

En la sección "comparar **modelos**" como se puede ver en la figura 7, el usuario puede comparar el desempeño de los 4 **modelos,** junto con las métricas de agrupamiento (Silhouette Score, Calinski-Harabasz Score, Davies-Bouldin Score) y las gráficas de barras, el usuario puede ver que modelo le conviene más [14].

Figura 7: Comparación de Modelos

III.II. Comportamiento del sistema implementado

Capturar o cargar datos para la predicción: Como se observa en la figura 9, el usuario puede cargar archivos CSV localmente o también puede capturarlos en tiempo real ya sea usando la "Captura manual" por cantidad de paquetes o con la "Captura Automática" por tiempo de captura e intervalo de tiempo entre capturas (Conteo regresivo para que se ejecute otra captura) la captura automática funcionará continuamente hasta que el usuario decida detenerla como se observa en la

figura 10[12], con el fin de que el IDS pueda notificarle al usuario sobre el estado de la red IoT mediante correo electrónico tal cual como se aprecia en la figura 8.

Figura 8 notificaciones por correo

Figura 9 Opciones de panel de control (Captura automática/manual y carga de CSV.

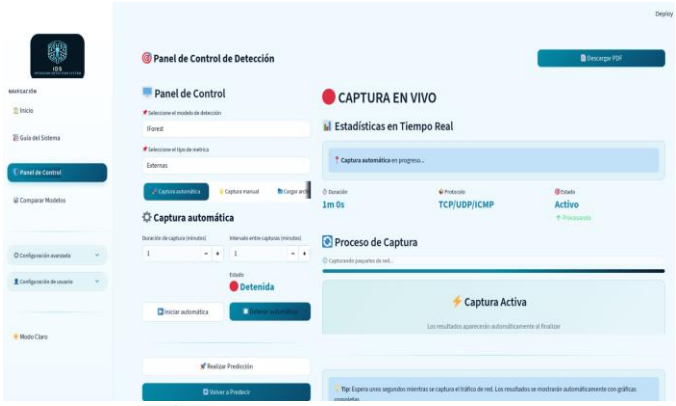


Figura 10 Predicción en tiempo real

Seleccionar modelos: Le permite al usuario seleccionar el algoritmo de detección a utilizar (IForest, Autoencoder, K-means y OCSVM) a partir de modelos cargados de forma dinámica como se muestra en la figura 9.

Visualizar métricas de clústeres en tiempo real: Como se observa en la figura 11, después de realizarse una predicción, la interfaz muestra las métricas de agrupación (Silhouette Score, Calinski Score y Davies Score) lo cual nos ofrece una evaluación rápida de la calidad de los clústeres detectados [14].

Figura 11 Métricas de agrupamiento y rendimiento computacional.

El proceso de preprocesamiento de los datos implementado ayuda a ajustar los datos de tráfico de red capturado para su posterior procesamiento con el modelo seleccionado, lo que facilita una integración perfecta entre la captura de datos y la inferencia del modelo. Permitiendo así el correcto funcionamiento del sistema sin necesidad de conocimientos técnicos avanzados de ML.

III.III Análisis de rendimiento computacional

TABLA I. Rendimiento computacional de los modelos en escenarios pre y post.

Modelo	Tiempo Pre (s)	Tiempo Post (s)	CPU (%)	RAM (%)	Comportamiento
IForest	0.09	1.88	Bajo	Medio	Estable
OCSVM	3.73	19.24	Medio	Alto	No escalable
K-Means	1.58	0.07	Bajo	Medio	Mejora en ataque
Autoencoder	1.16	3.47	Bajo	Alto	Sensible

Con el fin de evaluar la viabilidad del sistema en dispositivos con recursos limitados, se realizó un análisis de rendimiento considerando métricas de uso de CPU, memoria, almacenamiento y tiempo de inferencia en escenarios pre-ataque y post-ataque como se observa en la TABLA I. Los

resultados evidencian diferencias significativas entre los modelos. En condiciones normales, Isolation Forest presentó el menor tiempo de inferencia (0.09 s), seguido por Autoencoder (1.16 s) y K-Means (1.58 s), mientras que One-Class SVM mostró un tiempo considerablemente mayor (3.73 s), lo que limita su aplicabilidad en entornos de tiempo real.

Bajo condiciones de ataque (TCP RST Flood), se observa un incremento general en el consumo de recursos y en los tiempos de inferencia. En particular, One-Class SVM alcanzó un tiempo de 19.24 s, evidenciando una baja escalabilidad frente a tráfico anómalo. Por su parte, Isolation Forest mantuvo tiempos moderados (1.88 s), mientras que K-Means mostró una reducción significativa en el tiempo de procesamiento (0.07 s), lo cual sugiere una mayor facilidad en la separación de patrones bajo condiciones de ataque. El Autoencoder incrementó su tiempo a 3.47 s, reflejando una mayor carga computacional en presencia de anomalías.

Estos resultados evidencian un compromiso entre capacidad de detección y eficiencia computacional. Aspecto crítico en sistemas IDS para IoT. En este sentido, Isolation Forest se posiciona como la alternativa más balanceada para despliegues en edge computing, mientras que modelos como One-Class SVM, aunque sensibles, presentan limitaciones en términos de latencia.

III.IV Análisis de falsos positivos y estrategia de control

Dado que los modelos implementados corresponden a técnicas de aprendizaje no supervisado, es esperable que presenten una alta sensibilidad frente a patrones atípicos, incluso en tráfico considerado legítimo, característica ampliamente documentada en sistemas de detección de anomalías basados en machine learning [1], [8]. Con el objetivo de cuantificar este comportamiento, se evaluó la tasa de falsos positivos (FP) en el escenario pre-ataque, paquetes normales clasificados erróneamente como anomalías y la tasa de falsos negativos (FN) en el escenario post ataque, paquetes de ataque clasificados erróneamente como normales. Los resultados se presentan en la TABLA II.

TABLA II. Análisis cuantitativo de falsos positivos y falsos negativos por modelo.

Modelo	PRE-ATTACK: FP/ Total	Tasa FP	POST-ATTACK: FN/ Total	Tasa FN	Tasa de detección
IForest	157 / 470	33.4%	269 / 5621	4.8%	95.2%
OCSVM	30 / 100	30.0%	0 / 100	0.0%	100%
K-Means	136 / 470	28.9%	574 / 5617	10.2%	89.8%
Autoencoder	72 / 470	15.3%	514 / 5107	10.1%	89.9%

Los resultados evidencian un patrón consistente: todos los modelos presentan tasas de FP moderadas en condiciones de tráfico normal, siendo el Autoencoder el más conservador con un 15.3%, seguido de K-Means (28.9%), OCSVM (30.0%) e IForest (33.4%). Este comportamiento es inherente al diseño de los algoritmos no supervisados, que detectan desviaciones estadísticas sin conocimiento previo de clases [1], [5].

En contraste, durante el escenario de ataque TCP RST Flood, los modelos muestran tasas de FN significativamente menores, lo que indica una alta capacidad de detección ante perturbaciones reales. OCSVM alcanzó una tasa de detección del 100%, clasificando la totalidad del tráfico de ataque como anómalo. IForest presentó la segunda tasa más alta (95.2%), con solo 269 paquetes de ataque no detectados sobre 5621 analizados. K-Means y Autoencoder mostraron tasas equivalentes (~89.9%), con un mayor volumen de FN en valor absoluto.

Este análisis revela un compromiso fundamental entre sensibilidad (baja tasa de FN en ataque) y precisión en tráfico normal (baja tasa de FP): los modelos más agresivos en detección tienden a generar más falsas alarmas en condiciones normales. Para mitigar este efecto y mejorar la utilidad operativa del sistema, se implementó un mecanismo de umbrales basado en la proporción de instancias clasificadas como anomalías: estado normal cuando el porcentaje de anomalías es inferior al 45%, alerta media entre 45% y 70%, y estado crítico cuando supera el 80%. Este enfoque filtra fluctuaciones menores y reduce la sobrecarga de alertas, en línea con propuestas recientes orientadas a mejorar la usabilidad de sistemas IDS en entornos operativos [7], [10].

IV. DISCUSIÓN

A diferencia de estudios que se limitan a la evaluación aislada de modelos, la presente investigación integra de manera completa las etapas de captura de tráfico, inferencia y visualización dentro de una plataforma web interactiva, lo que favorece su aplicabilidad en entornos IoT con restricciones de hardware y necesidad de monitoreo continuo [5], [15]. Esta integración no solo aporta valor técnico, sino que también mejora la interpretabilidad y usabilidad del sistema en escenarios operativos reales.

Los resultados obtenidos evidencian que los modelos de aprendizaje no supervisado son capaces de detectar cambios significativos en el comportamiento del tráfico de red, particularmente en condiciones de ataque. Durante la

ejecución del ataque TCP RST Flood, se observó una separación más clara entre el tráfico normal y anómalo, reflejada en la mejora de métricas de agrupamiento en el escenario posterior al ataque. Este comportamiento sugiere que los modelos responden adecuadamente ante perturbaciones fuertes del sistema, lo cual es consistente con su capacidad para identificar desviaciones respecto a patrones predominantes.

No obstante, en escenarios previos al ataque se identificó la presencia de clasificaciones erróneas, principalmente en forma de falsos positivos. Este fenómeno está asociado a la alta sensibilidad de los modelos no supervisados frente a variaciones del tráfico, incluso cuando estas no corresponden a actividades maliciosas [1], [8]. En este sentido, el sistema no debe interpretarse como un clasificador exacto de eventos individuales, sino como una herramienta de análisis de comportamiento global. Bajo esta perspectiva, la incorporación de un esquema de umbrales basado en la proporción de anomalías detectadas permite filtrar fluctuaciones menores y reducir alertas innecesarias, favoreciendo su viabilidad en contextos operativos.

Desde el punto de vista del despliegue, la implementación mediante Streamlit permitió transformar los modelos en una herramienta accesible e interactiva, facilitando la captura de tráfico en tiempo real, la carga de datos externos, la selección dinámica de modelos y la visualización de métricas [7]. Esta integración representa un avance relevante frente a enfoques tradicionales que permanecen en entornos de laboratorio, al proporcionar un entorno funcional orientado al usuario final. En consecuencia, se establece un puente efectivo entre el desarrollo de modelos de aprendizaje automático y su aplicación práctica en sistemas de monitoreo de red.

Los resultados obtenidos son coherentes con investigaciones previas que destacan la utilidad de técnicas no supervisadas para la detección de anomalías en redes IoT, especialmente en contextos donde no se dispone de firmas de ataque previamente definidas [3], [4]. Asimismo, estudios recientes han resaltado la importancia de integrar capacidades de detección con herramientas de visualización y monitoreo en tiempo real, lo cual refuerza la relevancia del enfoque propuesto [10].

A pesar de estos aportes, es importante señalar que el desempeño observado depende en gran medida del entorno experimental y del tipo de ataque evaluado. Si bien los modelos muestran una respuesta clara ante ataques intensos, su comportamiento en condiciones de tráfico más complejo o en redes reales altamente variables puede diferir. En este sentido, los resultados deben interpretarse como una validación de viabilidad más que como una solución definitiva, lo que abre la puerta futuras mejoras en términos de robustez, reducción de falsos positivos y adaptación a entornos dinámicos.

TABLA III. Métricas de agrupamiento de los modelos de detección de anomalías por escenario.

Modelos	**Métricas de agrupamiento**		
	Silhouette Score	***Calinski-Harabasz Score***	***Davies-Bouldin Score***
OCSVM Pre-Attack (Normal)	0.5 (Regular)	80.8 (Regular)	0.548 (Regular)
OCSVM Post-Attack (TCP Flood)	1.0 (Bueno)	399910.4 (Bueno)	0.006 (Bueno)
IForest Pre-Attack (Normal)	0.1828 (Regular)	163.335 (Regular)	1.0926 (Regular)
IForest post-Attack (TCP Flood)	0.9216 (Bueno)	4655.0205 (Regular)	0.4578 (Bueno)
K-Means Pre-Attack (Normal)	0.6976 (Bueno)	695.7112 (Bueno)	0.4393 (Regular)
K-Means Post-Attack (Any Attack)	0.9345 (Bueno)	16736.3356 (Bueno)	0.6472 (Bueno)
Autoencoder Pre-Attack (Normal)	-0.0241 (Regular)	67.5984 (Regular)	1.0824 (Regular)
Autoencoder Post-Attack (Any Attack)	0.9436 (Bueno)	12587.538 (Bueno)	0.4309 (Bueno)

La TABLA III demuestra que los modelos de detección de anomalías (IForest, OCSVM, K-Means y Autoencoder) muestran un rendimiento significativamente superior en el escenario **Post-Attack** en comparación al estado **Pre-Attack**. Mientras que las métricas iniciales se clasifican mayormente como “regulares”, los resultados después de un ataque (como TCP RST Flood) alcanzan niveles “buenos”, sobresaliendo el modelo **OCSVM Post-Attack** con un Silhouette Score perfecto de 1.0 y un Davies-Bouldin Score cercano a cero, lo que indica una separación entre clústeres casi optima al identificar anomalías.

En términos comparativos, los resultados evidencian que no existe un modelo universal óptimo, sino que cada enfoque presenta ventajas dependiendo del contexto operativo. Isolation Forest ofrece el mejor balance entré precisión y eficiencia computacional, lo que lo hace adecuado para entornos IoT con restricciones. K-Means destaca por su rapidez en escenarios de alta anomalía, mientras que Autoencoder presenta mayor sensibilidad, útil para detección temprana. En contraste, One-Class SVM, aunque efectivo en la separación de patrones,

presenta limitaciones significativas en términos de latencia, lo que restringe su aplicabilidad en sistemas en tiempo real.

V. CONCLUSIONES

El propósito del presente estudio fue desarrollar una plataforma web orientada al despliegue de algoritmos de detección de anomalías en entornos IoT con limitaciones computacionales, particularmente en dispositivos como Raspberry Pi. Para ello, se empleó la metodología CRISP-ML como marco estructurado que permitió integrar de manera coherente las fases de análisis, modelado y despliegue, con el objetivo de construir una solución ligera, accesible y funcional para el monitoreo de tráfico de red en tiempo real [9].

Los resultados obtenidos evidencian que los modelos de aprendizaje automático no supervisados evaluados (Isolation Forest, K-Means, One-Class SVM y Autoencoder) son capaces de identificar cambios significativos en el comportamiento del tráfico durante escenarios de ataque, como el TCP RST Flood. Este comportamiento es consistente con estudios previos que destacan la efectividad de técnicas no supervisadas para la detección de anomalías en redes IoT, especialmente en contextos donde no se dispone de firmas de ataque previamente definidas [1], [3], [8]. En particular, se observó una mejora en las métricas de agrupamiento en condiciones posteriores al ataque, lo que confirma la capacidad de estos enfoques para detectar desviaciones respecto al comportamiento de la red [14].

Desde una perspectiva comparativa, el modelo Isolation Forest mostró el mejor equilibrio entre capacidad de detección, estabilidad en escenarios pre y post-ataque y eficiencia computacional, lo que lo posiciona como una alternativa adecuada para su implementación en entornos IoT con recursos limitados, en línea con investigaciones que resaltan la necesidad de modelos ligeros y eficientes en este tipo de contextos [4], [5]. El análisis cuantitativo de falsos positivos y falsos negativos (TABLA II) refuerza esta conclusión: IForest alcanzó una tasa de detección del 95.2% con solo un 33.4% de falsas alarmas en tráfico normal, mientras que OCSVM, pese a lograr una detección perfecta del 100%, presenta limitaciones críticas de latencia que restringen su uso en tiempo real. Por su parte, K-Means y Autoencoder mostraron tasas de detección equivalentes (~89.9%), siendo el Autoencoder el más conservador en falsos positivos (15.3%), lo que lo hace especialmente útil en escenarios donde se prioriza la detección temprana con menor sobrecarga de alertas [1].

Uno de los principales aportes del estudio radica en la integración de los modelos de detección de anomalías dentro de una plataforma web interactiva desarrollada en Streamlit, que permite la captura de tráfico en tiempo real, la carga de datos externos, la selección dinámica de modelos y la visualización de métricas desde una interfaz accesible. El elemento diferenciador central de este trabajo es precisamente la combinación de estas capacidades (captura en tiempo real, inferencia de modelos no supervisados y visualización interactiva) dentro de una única plataforma web ligera, un enfoque integrado que no ha sido reportado de forma conjunta en estudios previos sobre IDS para IoT. Este enfoque contribuye a reducir la brecha entre el desarrollo experimental de modelos de aprendizaje automático y su aplicación práctica en sistemas IDS para IoT, en concordancia con propuestas recientes orientadas a sistemas de detección en tiempo real basados en web [7], [10], [12].

Asimismo, la adopción de CRISP-ML permitió establecer un flujo de trabajo reproducible y orientado al despliegue, lo que fortalece la trazabilidad del proceso y facilita la transferencia del sistema a contextos operativos [9]. En este sentido, el estudio demuestra que los sistemas de detección de intrusiones basados en aprendizaje automático pueden trascender el ámbito experimental y consolidarse como herramientas funcionales para la supervisión de infraestructura IoT en escenarios con restricciones de recursos [2], [5].

No obstante, es importante considerar ciertas limitaciones. En primer lugar, la evaluación experimental se realizó principalmente utilizando el conjunto de datos BoT-IoT y escenarios controlados de ataque, lo que podría no reflejar completamente la complejidad del tráfico en entornos reales [6]. En segundo lugar, la implementación del sistema depende de herramientas de captura de paquetes como TShark, cuya operación está sujeta a las condiciones del sistema operativo y a los permisos disponibles en el dispositivo [12]. Adicionalmente, los umbrales utilizados para la generación de alertas fueron definidos de manera empírica a partir del comportamiento observado en los modelos, lo que sugiere la necesidad de validarlos en escenarios más diversos.

Como líneas de trabajo futuro, se propone evaluar el sistema en redes IoT reales con mayor variabilidad de tráfico, así como explorar la integración de modelos supervisados para mejorar la precisión en la clasificación de eventos. De igual manera, se plantea la implementación de una arquitectura basada en servicios (API) que permita delegar el procesamiento a servidores dedicados, reduciendo la carga en dispositivos IoT. Finalmente, se considera el uso de datos capturados en tiempo real para el reentrenamiento periódico de los modelos, con el fin de mejorar la adaptabilidad del sistema frente a cambios dinámicos en el comportamiento de la red [13], [15].

AGRADECIMIENTO

Queremos expresar nuestro más sincero agradecimiento a la Universidad Cooperativa de Colombia por su apoyo institucional hacia la presente investigación. El estudio se ha financiado parcialmente por el proyecto n. ° INV3356. Agradecemos de manera especial al Semillero de investigación INNOVASIC y al Grupo de investigación AQUA por sus recursos y colaboración, adicionalmente agradecemos la orientación del PhD. Fernando Gutiérrez Portela y el Mag. Oscar Augusto Díaz Triana. Por último, agradecemos al UNSW Cyber Range Lab por concedernos el acceso al conjunto de datos Bot-IoT para su validación.

REFERENCIAS

[1] E. Ozdogan, "A Comprehensive Analysis of the Machine Learning Algorithms in IoT IDS Systems," *IEEE Access*, vol. 12, pp. 46785–46811, 2024, doi: 10.1109/ACCESS.2024.3382539.

[2] V. A. Devi, E. Bhuvaneswari, and R. K. Tummala, "Decentralized Hybrid Intrusion Detection System for Cyber Attack Identification using Machine Learning," in *2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, 2023, pp. 1–5. doi: 10.1109/ICDSAAI59313.2023.10452439.

[3] A. S. Gomez, F. Gutierrez Portela, and O. A. Diaz Triana, "Intrusion Detection System for IoT using Anomaly Detection Techniques," in *2024 IEEE VII Congreso Internacional en Inteligencia Ambiental, Ingeniería de Software y Salud Electrónica y Móvil (AmITIC)*, 2024, pp. 1–6. doi: 10.1109/AmITIC62658.2024.10747648.

[4] M. Kumar and K. Sharma, "Enhanced Security In Matter-Enabled Iot Networks Through Anomaly Detection," in *2025 International Conference on Pervasive Computational Technologies (ICPCT)*, 2025, pp. 496–500. doi: 10.1109/ICPCT64145.2025.10940459.

[5] A. Verma and V. Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications," *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020, doi: 10.1007/s11277-019-06986-8.

[6] J. L. Leevy, J. Hancock, T. M. Khoshgoftaar, and J. Peterson, "Detecting Information Theft Attacks in the Bot-IoT Dataset," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2021, pp. 807–812. doi: 10.1109/ICMLA52953.2021.00133.

[7] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[8] Z. Liu, N. Thapa, A. Shaver, K. Roy, X. Yuan, and S. Khorsandroo, "Anomaly Detection on IoT Network Intrusion Using Machine Learning," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2020, pp. 1–5. doi: 10.1109/icABCD49160.2020.9183842.

[9] S. Studer *et al.*, "Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology," *Mach. Learn. Knowl. Extr.*, vol. 3, no. 2, pp. 392–413, 2021, doi: 10.3390/make3020020.

[10] N. Tamuka and K. Sibanda, "A Novel Real-Time Web Based Intrusion Detection System (IDS)," in *2024 4th International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2024, pp. 343–350. doi: 10.1109/IMITEC60221.2024.10850945.

[11] C. Jin and J.-A. Liu, "Applications of Support Vector Mathine and Unsupervised Learning for Predicting Maintainability Using Object-Oriented Metrics," in *2010 Second International Conference on Multimedia and Information Technology*, 2010, pp. 24–27. doi: 10.1109/MMIT.2010.10.

[12] M. O. Kaya, M. Ozdem, and R. Das, "A novel approach for graph-based real-time anomaly detection from dynamic network data listened by Wireshark," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 12, no. 2, Jan. 2025, doi: 10.4108/eetinis.v12i2.7616.

[13] M. Ayyannan, "Accuracy Enhancement of Machine Learning Model by Handling Imbalance Data," in *2024 International Conference on Expert Clouds and Applications (ICOECA)*, 2024, pp. 593–599. doi: 10.1109/ICOECA62351.2024.00109.

[14] D. Chicco, A. Campagner, A. Spagnolo, D. Ciucci, and G. Jurman, "The Silhouette coefficient and the Davies-Bouldin index are more informative than Dunn index, Calinski-Harabasz index, Shannon entropy, and Gap statistic for unsupervised clustering internal evaluation of two convex clusters," *PeerJ Comput. Sci.*, vol. 11, p. e3309, 2025, doi: 10.7717/peerj-cs.3309.

[15] E. Muhati and D. Rawat, "Data-Driven Network Anomaly Detection with Cyber Attack and Defense Visualization," *Journal of Cybersecurity and Privacy*, vol. 4, no. 2, pp. 241–263, 2024, doi: 10.3390/jcp4020012.