Articulo de investigación

Algoritmos supervisados para la predicción de fallos en la red LAN de la UPSE

Supervised Algorithms for Failure Prediction in the UPSE LAN network

Ariel Oswaldo Fernández Loor[1] https://orcid.org/0000-0003-0508-6133
Alicia Germania Andrade Vera[1] https://orcid.org/0000-0003-1457-2571

[1]Instituto de Postgrado-Universidad Estatal Península de Santa Elena, La Libertad-Ecuador | CP 240204

afernandez@upse.edu.ec

https://doi.org/10.26423/03cdzf27
Páginas: 90- 107

Resumen

En redes universitarias como la de la Universidad Estatal Península de Santa Elena, la gestión LAN sigue siendo reactiva, sin historial ni alertas tempranas. Este estudio propone aplicar algoritmos supervisados de aprendizaje automático, seleccionados con base en evidencia científica, para construir y evaluar un modelo predictivo de fallos a partir de telemetría SNMP obtenida mediante Zabbix. Se utilizó una metodología combinada de Investigación en Ciencias del Diseño (DSR) y CRISP–DM, con ventanas de 60 minutos sobre 7571 ejemplos (729 fallos y 6 842 normales). Se compararon dos modelos: Random Forest, entrenado con características estadísticas, y una red neuronal convolucional unidimensional, aplicada sobre secuencias multivariadas. Random Forest alcanzó una exactitud del 96,88 %, mientras que la red neuronal logró un recall del 73,10 %. Los resultados demuestran su complementariedad y evidencian que la combinación de ambos modelos favorece una gestión proactiva de la red institucional, reduciendo los tiempos de respuesta ante incidencias.

Palabras clave: Aprendizaje automático, Análisis predictivo, Gestión de redes, Redes universitarias, SNMP

Abstract

In university networks such as the one at the Universidad Estatal Península de Santa Elena, LAN management remains predominantly reactive, lacking historical records and early-warning mechanisms. This study proposes the application of supervised machine-learning algorithms, selected based on scientific evidence, to build and evaluate a predictive failure-detection model using SNMP telemetry collected through Zabbix. A combined Design Science Research (DSR) and CRISP–DM methodology was applied, with 60-minute windows over 7 571 samples (729 failures and 6 842 normal cases). Two approaches were compared: a Random Forest model trained on statistical features, and a one-dimensional convolutional neural network applied to multivariate sequences. Random Forest achieved an accuracy of 96.88 %, while the neural network reached a recall of 73.10 %. The results show the complementary nature of both models and demonstrate that their combined use supports proactive institutional network management, reducing response times to incidents.

Keywords: Machine learning, Predictive analytics, Network management, University networks, SNMP.

Recepción: 17/10/2025 | Aprobación: 26/11/2025 | Publicación: 26/12/2025

1. Introducción

El avance tecnológico y la digitalización han convertido a las redes de datos en un pilar esencial para las funciones académicas y administrativas. Esto exige mecanismos de seguimiento que vayan más allá del monitoreo reactivo [1]. La operación de aulas virtuales, sistemas de matrícula, repositorios académicos y servicios de identidad demanda visibilidad continua del rendimiento de la red. Plataformas que recopilan indicadores a través del Protocolo simple de administración de red (SNMP) son una forma práctica de obtener datos. Esto posibilita integrar registros históricos y facilitar la toma de decisiones [2, 3]. Existen herramientas como PRTG que en conjunto con firewalls avanzados combinan visualización, correlación e identificación de ataques en tiempo real [4]. No obstante, garantizar los niveles adecuados en la calidad del servicio, como latencia, pérdida de paquetes y disponibilidad, resulta ser un reto en la gestión de redes en entornos universitarios [5].

Las universidades enfrentan la presión de garantizar conectividad permanente. Aunque las plataformas de monitoreo institucional recolectan indicadores, su enfoque sigue siendo reactivo y carece de predicción. En la Universidad Estatal Península de Santa Elena (UPSE), la herramienta The Dude permite observación en tiempo real [6], pero no conserva registros históricos ni genera alertas anticipadas. Alternativas como Zabbix, Nagios o Prometheus ofrecen recopilación de datos SNMP y disponibilidad, transformando los datos operativos en conocimiento de gestión [7].

Ante esta limitación, se genera la necesidad de contar con una gestión proactiva. Esto se logra al utilizar algoritmos supervisados capaces de reconocer patrones en registros históricos de la red. Diversos estudios han demostrado que algoritmos como los Árboles de Decisión, Random Forest y Redes Neuronales resultan ser eficaces en la detección y clasificación de anomalías. Así, se confirma su utilidad en entornos universitarios [8]. Estos métodos se incorporan cada vez más a las prácticas de monitoreo avanzado, donde la telemetría y el *Machine Learning* (ML) permiten transformar datos en predicciones y acciones preventivas [9]. El uso de modelos predictivos reduce el tiempo de inactividad y optimiza recursos al priorizar intervenciones según el nivel de riesgo [10], un aspecto clave para mantener la calidad de servicio [11].

Asimismo, la evidencia empírica reporta altos niveles de desempeño en algoritmos supervisados aplicados a bases SNMP–MIB [12], así como experiencias colaborativas orientadas a la detección de fallos y ataques DDoS mediante aprendizaje automático [13]. Estos enfoques se basan en el aprendizaje supervisado, donde los modelos se entrenan con ejemplos etiquetados (normal o fallo) para minimizar el error esperado [14].

La predicción de fallos en redes LAN no solo es un reto técnico, sino también operativo. En entornos académicos, una interrupción prolongada puede afectar la continuidad institucional, retrasar procesos administrativos y generar costos adicionales por la recuperación del servicio. El presente estudio se desarrollará en la red LAN de la UPSE, utilizando el paradigma gestor-agente SNMP. Se recolectarán series temporales de contadores de octetos de entrada y salida y variables de estado, como la latencia ICMP, la pérdida de paquetes y la disponibilidad, mediante herramientas institucionales. Con esta información se construirá un conjunto de datos etiquetado que represente tanto condiciones de fallo como de operación normal [15].

Finalmente, este entorno representa un escenario adecuado para evaluar el impacto de modelos de aprendizaje en métricas reales de producción [16]. La investigación se orienta en responder la siguiente pregunta: ¿Qué algoritmos supervisados, seleccionados con base en evidencia científica, predicen con mayor eficacia los fallos en la red LAN de la UPSE usando indicadores históricos SNMP, y qué desempeño (accuracy, precision, recall, F1, AUC-ROC) alcanzan frente a líneas base y entre sí?

En este contexto y con el fin de responder a la pregunta de investigación se propone un objetivo principal: Aplicar algoritmos supervisados de aprendizaje automático, seleccionados con base en evidencia científica, para evaluar su rendimiento en la predicción de fallos en la red LAN de la Universidad Estatal Península de Santa Elena, utilizando métricas históricas recolectadas del entorno institucional. El estudio se desarrolló a partir de indicadores históricos recolectados mediante telemetría SNMP, incluyendo latencia, disponibilidad, pérdida de paquetes y tráfico de red. Este enfoque permitió identificar los periodos de mayor consumo de ancho de banda y validar el desempeño de los modelos mediante métricas como exactitud, precisión, recall y área bajo la curva ROC, fortaleciendo la gestión proactiva de la infraestructura institucional.

FUNDAMENTACIÓN TEÓRICA

Gestión de redes LAN y monitoreo SNMP

En el modelo gestor–agente de SNMP, cada dispositivo expone objetos MIB como contadores de interfaz y estados operativos. El gestor consulta periódicamente dichos objetos para persistir las lecturas como series temporales en su base de datos. En este estudio, ese rol lo cumple Zabbix, cuya función es consolidar indicadores por dispositivo o interfaz para su análisis posterior, evitando configuraciones redundantes y habilitando la explotación histórica de la telemetría [14] [[4] p. 14].

Variables predictoras de fallos

El conjunto de predictores se fundamenta en objetos MIB de interfaz y sondeo activo que capturan desempeño y disponibilidad [17]:

- **Octetos de entrada / Octetos de salida:** contadores acumulativos por interfaz; sus diferencias por intervalo estiman ancho de banda (bit/s).
- **Paquetes Unicast de entrada y Salida normal / Paquetes Unicast de entrada y Salida con errores:** evidencian degradaciones físicas o de capa 2.
- **Latencia y pérdida ICMP:** indica el tiempo y el porcentaje de respuesta, proporcionando información para evaluar la disponibilidad y la calidad de servicio.
- **Disponibilidad y reinicios:** reinicio del tiempo de actividad puede utilizarse para identificar interrupciones en las series temporales.

Estos indicadores, almacenados como series temporales, facilitan la creación del conjunto de datos monitorizado para la predicción de fallos.

Criterios de selección de los algoritmos supervisados

Para la selección de los algoritmos se definieron criterios objetivos basados en la literatura reciente:

- Evidencia de aplicación en redes académicas o empresariales con telemetría SNMP, MIB o protocolos equivalentes.
- Reporte de métricas comparables entre estudios (accuracy, precision, recall, F1 y AUC-ROC).
- Capacidad de generalización y razonable costo computacional para entornos de producción.

La Tabla 1 presenta los algoritmos identificados entre 2019 y 2025, organizados por tipo de aprendizaje, precisión reportada, interpretabilidad y escalabilidad. Se observa que los algoritmos supervisados, como Random Forest (RF), Máquinas de Vectores de Soporte (SVM), Árboles de decisión, Gradient Boosting y k-vecinos más cercanos (k-NN), dominan la literatura, mientras que los modelos profundos como redes neuronales convolucionales (CNN), redes neuronales recurrentes de tipo LSTM y perceptrones multicapa (MLP) muestran métricas superiores (95–98 %) aunque con mayor demanda computacional.

Los métodos no supervisados e híbridos —por ejemplo, redes generativas antagónicas (GAN), K-means, redes profundas de creencia (DBN) y redes neuronales de grafos con atención multiagente (GNN+MAB)— aparecen en menor proporción y se orientan principalmente a tareas de agrupamiento o generación de datos sintéticos.

Tabla 1: Comparación entre algoritmos para detección de fallos en redes (2019–2025): supuestos, métricas típicas, complejidad y consideraciones de implementación.

Tipo de Aprendizaje	Algoritmo	Precision / Accuracy	Interpretación	Escalabilidad	Fuente
Supervisado	Árbol de decisión	Media-alta (70–90 %)	Alta	Limitada en grandes datasets	[18, 19, 20, 21, 22, 23]
	CNN	Muy alta (>95 %)	Baja	Alta (entrenamiento distribuido)	[20, 24, 25, 26, 27, 28, 29]
	Gradient Boosting	Muy alta (90–98 %)	Media	Alta	[23, 26, 30]
	k-Nearest Neighbors (KNN)	Media (75–90 %)	Alta	Baja en datasets grandes	[20, 21, 22, 23, 31, 32]
	LSTM	Muy alta (>95 %)	Baja	Alta (requiere GPUs)	[24, 27]
	MLP	Alta (85–95 %)	Baja	Alta	[22, 23, 25, 26]
	Naive Bayes	Media (70–85 %)	Alta	Muy alta	[21, 22, 32]
	Random Forest	Alta (90–95 %)	Media	Alta	[19, 20, 22, 23, 26, 27, 31, 32, 33, 34, 35]
	Regresión Logística	Media (70–85 %)	Muy alta	Alta	[19, 22, 23, 35]
	Support Vector Machine	Alta (85–95 %)	Media	Limitada (>10k ejemplos)	[19, 21, 22, 23, 26, 31, 35]
	Transformers autorregresivo	Muy alta (>97 %)	Baja	Muy alta (requiere GPU/TPU)	[36]
No supervisado	GAN	Muy alta (datos sintéticos reales)	Muy baja	Limitada (entrenamiento inestable)	[37, 38]
	K-means	Variable (70–90 %)	Media	Muy alta	[33]
	WSBM	Alta (85–95 %)	Media	Alta en grafos medianos	[39]
Híbrido	DBN	Alta (85–95 %)	Baja	Media	[26]
	GNN + MAB	Muy alta (95–97 %)	Baja-media	Alta (aplicable a IoT/SDN)	[40]

Comparativa y justificación final

La bibliografía reciente destaca dos familias de algoritmos. Los superficiales, basados en árboles [18, 23, 26, 27, 30, 35] o Máquinas de Vectores de Soporte [19, 21, 22, 23, 26, 31, 35]. Por otra parte, los algoritmos profundos basados en redes neuronales [20, 22, 23, 24, 25, 26, 27, 28, 29, 36, 37, 38, 40].

En la Tabla 2 se resumen los algoritmos por tipo (superficial o profundo), junto con su rasgo distintivo y la frecuencia observada en la literatura.

En la revisión PRISMA, el algoritmo Random Forest es

el más aplicado con 11 trabajos. Seguido de SVM con 7 y Árboles de decisión que tiene 6 aplicaciones. Las aplicaciones para los algoritmos profundos fueron de 7 para CNN y 4 para MLP.

Tabla 2: Algoritmos más aplicados por profundidad.

N	Tipo	Algoritmo	Característica diferencial	Prec./Acc.	Fuente
1	Superficial	Random Forest	Robusto contra overfitting, estándar en benchmarks	Alta (90–95 %)	[19, 20, 22, 23, 26, 27, 31, 32, 33, 34, 35]
2		SVM	Datos pequeños con fronteras no lineales	Alta (85–95 %)	[19, 21, 22, 23, 26, 31, 35]
3		Árbol dec.	Base de ensambles (RF, GB, XGBoost)	Media-alta (70–90 %)	[18, 19, 20, 21, 22, 23]
1	Profundo	CNN	Extrae patrones relevantes automáticamente	Muy alta (>95 %)	[20, 24, 25, 26, 27, 28, 29]
2		MLP	Red neuronal densa con capas ocultas	Alta (85–95 %)	[22, 23, 25, 26]

La preselección de Random Forest y CNN-1D se basó en su desempeño superior en la literatura reciente, su equilibrio entre interpretabilidad y capacidad de generalización, y su aplicabilidad a conjuntos de datos multivariados derivados de telemetría SNMP.

Los trabajos previos han demostrado la utilidad de estos modelos en la detección de anomalías, aunque la mayoría se enfoca en redes heterogéneas o entornos controlados. Este estudio aborda un segmento de red universitaria real, basada en telemetría SNMP, aportando evidencia bibliográfica sobre la aplicación de estos métodos en contextos institucionales.

2. Materiales y Métodos

La presente investigación es de tipo aplicado, con un enfoque cuantitativo y predictivo. Está orientada a identificar y evaluar algoritmos de aprendizaje supervisado para predecir fallos en la red LAN de la UPSE a partir de indicadores históricos SNMP.

Se utilizaron dos algoritmos: Random Forest (RF), que utiliza vectores de características tabulares agregados por ventana, y una Red Neuronal Convolucional 1D (CNN-1D), que procesa secuencias multivariadas crudas de 60 pasos por ventana. Esta combinación permitió comparar modelos explicables (RF) con modelos de alto rendimiento (CNN-1D), buscando optimizar la precisión y generalización del sistema de predicción.

El estudio se basó en la metodología de Investigación en Ciencias del Diseño (DSR, por sus siglas en inglés), que incluye tres fases principales: relevancia, diseño y rigor [41]. En la fase de relevancia se definió el problema de la gestión reactiva de fallos en la LAN de la UPSE y se caracterizó el contexto operativo. Se tomaron datos históricos de telemetría vía SNMP durante el horario académico. Este enfoque garantizó un impacto institucional medible en términos de continuidad de servicio y tiempos de respuesta.

En la fase de diseño, se construyó el artefacto experimental. En el que se incluye una secuencia de datos con indicadores SNMP desde Zabbix. Los datos se limpiaron, posteriormente se etiquetaron y segmentaron en ventanas temporales.

Finalmente, se realizó la extracción de características y la implementación de los modelos supervisados seleccionados: Random Forest y CNN-1D. Para el desarrollo del modelo Random Forest se utilizó la biblioteca Scikit-learn.

Para el modelo CNN-1D se aplicó Lightning sobre PyTorch. Para la búsqueda de hiperparámetros y selección de características se empleó la métrica F1-macro como criterio de evaluación. El desempeño de los modelos se evaluó mediante validación cruzada estratificada, garantizando consistencia entre los experimentos.

Entre las métricas reportadas se incluye accuracy, precision, recall, F1-Score, F2-Score, ROC-AUC y PR-AUC, complementadas con análisis comparativos entre los modelos y sus umbrales. Asimismo, se fijaron semillas aleatorias para asegurar la reproducibilidad en los resultados.

Los datos se tomaron de 38 puntos de acceso de la marca Ruckus distribuidos de la siguiente forma: 2 del modelo R320, 4 del modelo T350C y 32 del modelo R650. Los datos brutos y scripts no se publican por motivos de confidencialidad.

Sin embargo, la investigación preservó la trazabilidad y replicabilidad interna mediante documentación técnica detallada. En conjunto, la metodología DSR permitió diseñar, evaluar y transferir una solución predictiva aplicable a entornos reales de redes universitarias como la UPSE, tal como se muestra en la Figura 1.

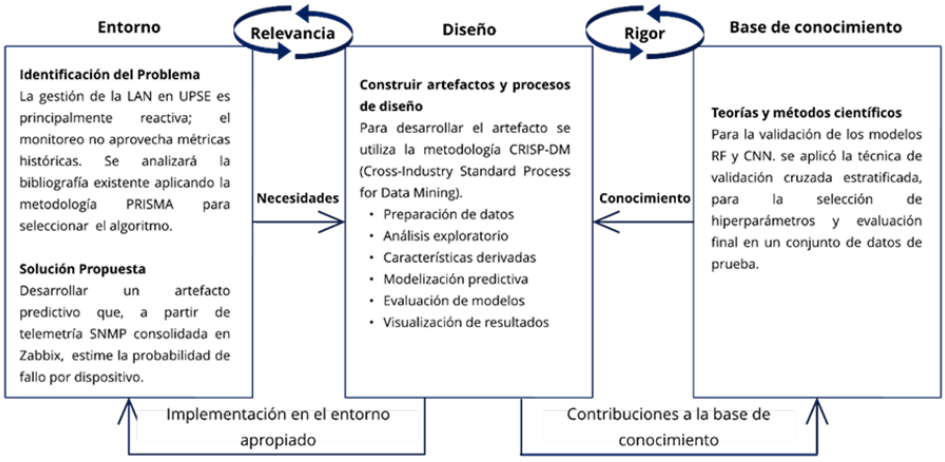

Figura 1: Metodología DSR aplicada al desarrollo del sistema de predicción de fallos en la red LAN de la UPSE.

Ciclo de relevancia La primera fase consistió en una revisión exhaustiva de la literatura científica sobre el uso del aprendizaje automático en Redes LAN enfocada en entornos universitarios y públicos. Esta revisión permitió identificar las principales causas de fallos en la red y los indicadores que pueden anticiparlos. En síntesis, los fallos observados en la red LAN de la UPSE se originan principalmente por (i) la sobrecarga de enlaces y puertos, (ii) la degradación física en las capas 1 y 2, (iii) la saturación y reinicios de equipos, y (iv) la elevada demanda de usuarios durante las horas de mayor actividad académica. Estas condiciones derivan en congestión y degradación del rendimiento, provocando interrupciones en los servicios académicos, deterioro de la calidad de servicio (QoS), aumento de incidentes gestionados por la mesa de ayuda y afectación reputacional para la institución.

Se aplicó una revisión estructurada de la literatura en Scopus y Web of Science para identificar estudios que aplicaran algoritmos supervisados a la predicción de fallos en redes LAN con telemetría SNMP/MIB. Del total de 269 registros iniciales, se eliminaron duplicados y estudios no pertinentes, seleccionándose finalmente 31 trabajos con métricas comparables y contextos afines a redes universitarias. Esta base permitió definir los algoritmos candidatos para el modelado.

Ciclo de diseño

Para el ciclo de diseño se aplicó la metodología CRISP–DM para construir el artefacto predictivo. En la fase de comprensión del negocio se estableció el objetivo de anticipar fallos en la red LAN de un edificio académico de la UPSE. La fase de comprensión y preparación de datos se materializó en un pipeline que recolecta telemetría SNMP desde Zabbix, estandariza y alinea las series temporales. Los datos se agruparon en ventanas de 60 minutos con etiquetado de normal o fallo. Una vez agrupado se extrajeron características estadísticas como la media, desviación estándar, el mínimo, el máximo y el jitter para el indicador tiempo de respuesta.

Posteriormente, en la aplicación de los algoritmos, el modelo Random Forest se entrenó sobre características extraídas, mientras que la CNN-1D se aplicó con los datos crudos. Ambos modelos fueron entrenados y validados mediante estrategias mixtas de particionado, reservando un conjunto independiente para la prueba final. Los detalles de ajuste de hiperparámetros, selección de características y calibración de umbrales se describen en la sección Evaluación del modelo. La arquitectura general del sistema y el flujo metodológico completo, desde la recolección SNMP hasta la validación de los modelos, se detallan en la Figura 2.

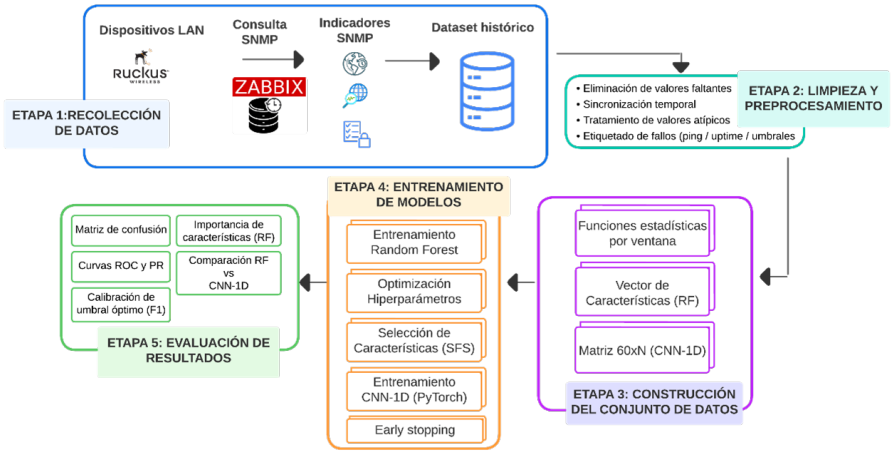

Figura 2: Flujo metodológico de construcción de modelos supervisados según la metodología CRISP–DM.

Conjunto de datos

Los datos se recolectaron mediante SNMP con Zabbix en un edificio académico de la UPSE, ubicado en la zona este del campus. La recolección se hizo durante cuatro semanas, desde el 25 de agosto al 17 de septiembre del 2025, de lunes a viernes entre las 07:00 y las 19:00 horas. Se recolecta los datos en este periodo debido a que es cuando el tráfico de usuarios es más representativo. El total de datos almacenados alcanzó los 547 874 registros. Los datos están distribuidos de la siguiente forma: Semana 1 (136 925), Semana 2 (136 990), Semana 3 (136 991) y Semana 4 (136 968). Los indicadores incluyeron disponibilidad (ping exitoso/fallido), tiempo de respuesta ICMP, octetos transmitidos y recibidos, paquetes unicast enviados y recibidos y uptime para identificar reinicios en los dispositivos. En la Tabla 3 se detalla el total de datos recolectados por dispositivo, así como la cantidad de ocasiones en que se registraron o no incidencias. Las proporciones se mantienen consistentes entre modelos, con aproximadamente un 9,6 % de ventanas con fallos, lo que garantiza una representación adecuada de ambas clases para la validación cruzada y la comparación entre algoritmos supervisados.

Tabla 3: Algoritmos más aplicados por profundidad.

Dispositivo	Cantidad	Normal	Fallo	Total
R320	2	407	55	462
T350C	4	855	77	932
R650	32	5580	597	6177
Total	38	6842	729	7571

Descripción de las variables del conjunto de datos

Las variables consideradas en el modelado corresponden a los indicadores descritos en la subsección “Variables predictoras de fallos”, obtenidos mediante telemetría SNMP desde Zabbix con muestreo de un minuto. Es fundamental aclarar que este estudio no se basa en el análisis de paquetes o tramas de red (frames) a nivel de enlace, sino en series temporales de métricas de gestión agregadas. Por ejemplo, la variable ”tiempo de respuesta ICMP”no implica la captura de la trama del paquete ping, sino el registro del valor escalar (en milisegundos) reportado por el agente SNMP en ese instante.

Para el análisis, los datos se estructuraron en ventanas temporales de 60 minutos. Cada ventana representa una ïnstancia.º ejemplo de entrenamiento (N). Cada una de estas ventanas contiene 60 mediciones consecutivas de cinco indicadores SNMP: latencia ICMP, octetos transmitidos (TX), octetos recibidos (RX), paquetes unicast TX y paquetes unicast RX. La Figura 3 ilustra la representación matricial (5 × 60) correspondiente a una ventana de datos crudos utilizada como entrada base para ambos modelos supervisados.

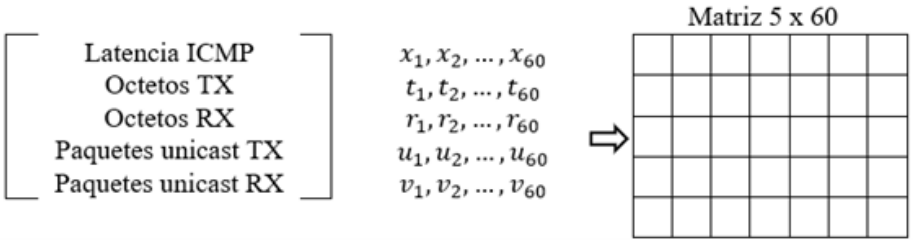

Figura 3: Estructura de la ventana temporal multivariada (5×60) utilizada como entrada para la CNN-1D.

La estructura de datos difiere según el algoritmo utilizado, aprovechando la naturaleza de cada modelo:

- **Para Random Forest (Enfoque tabular de características agregadas):** Se construyó una matriz bidimensional de tamaño (N x M), donde N es el número total de ventanas (7 571) y M es el número de características derivadas. A partir de los 60 valores brutos de cada indicador dentro de una ventana, se calcularon estadísticas descriptivas (Media, Desviación Estándar, Mínimo y Máximo) para conformar el vector de características de entrada.
- **Para CNN-1D (Enfoque de secuencias multivariadas crudas):** Se preservó la secuencialidad temporal de los datos, estructurándolos como tensores tridimensionales de tamaño (N x L x C), donde N es el número de ventanas, L es la longitud de la secuencia (pasos de tiempo) y C corresponde a los canales o indicadores brutos analizados simultáneamente. Esta estructura permite a la red convolucional aplicar filtros a lo largo de la dimensión temporal para extraer patrones secuenciales locales y globales en la evolución de los indicadores.

Procesamiento de datos

El procesamiento de datos se desarrolló en tres etapas principales:

1. **Depuración inicial:** se eliminaron registros duplicados y con valores faltantes en cualquier indicador.
2. **Agrupación temporal:** se agruparon los indicadores en

ventanas de 60 minutos (60 registros por ventana) para generar ejemplos homogéneos.

3. **Tratamiento de valores atípicos:** se descartaron observaciones fueras de rango normal.

La variable de salida se definió a partir del indicador ICMP Ping: Y = 0 para estado normal y Y = 1 para fallo. Adicionalmente, se consideraron como fallos los casos en que se detectó un reinicio del dispositivo (variación del uptime inferior al intervalo de consulta) o el incumplimiento de umbrales operativos, tales como tiempos de respuesta elevados o pérdida de paquetes. Estas condiciones fueron tratadas de manera equivalente dentro del conjunto supervisado. Tras el preprocesamiento se obtuvieron 7 571 ejemplos, de los cuales 729 corresponden a incidencias (Y = 1) y 6 842 indican estado normal (Y = 0). Esta proporción se mantiene en los indicadores analizados, evidenciando un desbalance de clases que fue considerado en el modelado. Las características se generaron mediante funciones estadísticas aplicadas a cada métrica, y su resumen general se presenta en la Tabla 4.

Tabla 4: Funciones estadísticas aplicadas por indicador recolectado.

Indicador	Media	Desv. Est.	Mínimo	Máximo	Muestras	Y=1	Y=0
ICMP response time	Sí	Sí	Sí	Sí	7571	729	6842
Jitter	Sí	No	No	Sí	7571	729	6842
Paquetes unicast enviados ETH	Sí	Sí	Sí	Sí	7571	729	6842
Paquetes unicast recibidos ETH	Sí	Sí	Sí	Sí	7571	729	6842
Trafico LAN Recibido	Sí	Sí	Sí	Sí	7571	729	6842
Trafico LAN Transmitido	Sí	Sí	Sí	Sí	7571	729	6842

División de datos

El conjunto de datos se dividió mediante muestreo aleatorio estratificado por clase en una proporción de 75 % para entrenamiento y 25 % para prueba, preservando la relación entre clases. Se fijó una semilla aleatoria de 1 para garantizar la reproducibilidad en todas las ejecuciones del algoritmo.

Evaluación del modelo

La estrategia de evaluación se diseñó para comparar, de forma justa y reproducible, el desempeño predictivo de ambos modelos: el modelo Random Forest (basado en características estadísticas agregadas) y la CNN-1D (basada en secuencias temporales crudas).

En el caso de la CNN-1D, el análisis del comportamiento del modelo permitió identificar los patrones temporales que contribuyen a la detección de fallos. Las primeras capas convolucionales aprendieron fluctuaciones locales en la latencia ICMP y en el tráfico LAN transmitido y recibido, mientras que las capas más profundas integraron relaciones de mayor alcance entre el jitter, la disponibilidad y los reinicios detectados en el uptime.

Aunque los valores brutos permanecen confidenciales por razones institucionales, se proporciona esta descripción conceptual para garantizar transparencia sobre el proceso de aprendizaje y las variables que más influyen en las predicciones del modelo. Todas las curvas y métricas (ROC, PR, F1–umbral y matrices de confusión) se generaron directamente a partir de las probabilidades estimadas por los modelos sobre el conjunto de prueba; para cada umbral entre 0 y 1 se calcularon sistemáticamente las métricas estándar de *Scikit-learn*, lo que permite reproducir matemáticamente las gráficas a partir de y_{test} y $(Y = 1)$.

Selección de hiperparámetros

Para optimizar el modelo Random Forest se aplicó una búsqueda exhaustiva mediante GridSearchCV evaluando distintas combinaciones de hiperparámetros sobre el conjunto de entrenamiento, mientras que el conjunto de prueba se mantuvo independiente durante todo el proceso. El criterio de selección fue la métrica F1-macro, debido a su capacidad para equilibrar el desempeño en ambas clases en presencia de desbalance. En resumen, se exploraron los siguientes hiperparámetros:

- `n_estimators`: número de árboles en el bosque, explorado en el rango de 10 a 100 en incrementos de 10.
- `criterion`: función de impureza utilizada para dividir los nodos, evaluando las opciones `"gini"`, `.entropy"` y `"log_loss"`.
- `max_features`: proporción de variables en cada división, con valores de sqrt, log2 y None

Los demás hiperparámetros se mantuvieron en sus valores por defecto al no evidenciar mejoras relevantes.

Selección de características

Se implementó selector de características secuencia (SFS) en modo forward, con una tolerancia de 1×10^{-3} y una validación cruzada estratificada (k = 5), utilizando el algoritmo Random Forest como modelo base y la métrica F1-macro como criterio de evaluación. El método permitió identificar un subconjunto reducido de variables sin degradar el rendimiento, destacando aquellas asociadas a la variabilidad del tráfico LAN y dispersión de paquetes unicast.

Entrenamiento profundo

La CNN-1D se implementó en PyTorch Lightning con un máximo de 150 épocas, tamaño de lote de 64 y el optimizador Adam (tasa de aprendizaje 1×10^{-5}). La arquitectura consta de dos capas convolucionales 1D con 64 y 128 filtros (tamaños de kernel 5 y 3, respectivamente), cada una seguida de una activación ReLU y una operación de max pooling tras la primera convolución. Posteriormente se

aplica un módulo de adaptive average pooling que reduce la dimensión temporal a un único valor por filtro y una cabeza densa compuesta por una capa totalmente conectada de 128 unidades con dropout (p = 0,2) y una capa lineal de salida con dos neuronas. La representación esquemática completa de esta arquitectura se muestra en la Figura 4.

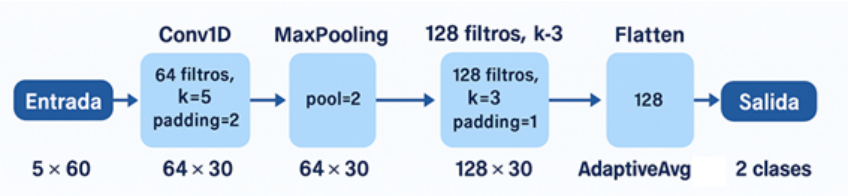

Figura 4: Arquitectura de la red neuronal convolucional 1D.

Visualización de resultados

Para facilitar la interpretación se generaron:

- **Curvas ROC y PR** en validación y prueba, con marcadores del umbral seleccionado.
- **Matriz de confusión** en el conjunto de prueba.
- **Importancias de características** para Random Forest.
- **Curvas de entrenamiento para la CNN** para evidenciar convergencia y descartar sobreajuste.
- **Análisis de umbral:** F1, precision y recall en función del umbral en validación.
- **Comparativa:** Random Forest vs. CNN.

Fase rigor

La validación se realizó mediante validación cruzada estratificada (k-fold) aplicada sobre el conjunto de entrenamiento para optimizar los hiperparámetros y la selección de características. Posteriormente, se evaluó el desempeño final sobre un conjunto de prueba independiente. El umbral de decisión se mantuvo constante respecto al determinado en validación. Las métricas principales fueron F1-score de la clase fallo y AUC-PR, complementadas con accuracy, precision, recall y AUC-ROC para proporcionar una visión integral del comportamiento del modelo.

Consideraciones éticas y replicabilidad

La recolección de indicadores de red se realizó con autorización del área de Tecnologías de la Información y Comunicación de la UPSE. El proceso se limitó a información técnica de infraestructura (SNMP, ICMP, paquetes transmitidos y uptime), sin incluir datos personales ni contenidos de usuario. Por razones de seguridad institucional, el conjunto de datos crudo y los scripts de procesamiento no se publican. Sin embargo, se documentaron íntegramente los procedimientos de preprocesamiento, selección de características y validación. El estudio mantiene trazabilidad completa y puede replicarse bajo convenios autorizados en contextos similares, garantizando la confidencialidad institucional.

3. Resultados

En esta sección se presentan los resultados obtenidos con los algoritmos de aprendizaje automático aplicados a la predicción de fallos en la red LAN de la UPSE. Los modelos se entrenaron a partir de los indicadores históricos capturados mediante SNMP y se evaluaron según las métricas definidas en la metodología. De acuerdo con la revisión sistemática descrita en la Fundamentación Teórica, los algoritmos Random Forest y CNN-1D fueron seleccionados por su desempeño en escenarios similares reportados en la literatura, su capacidad para manejar datos de telemetría y su equilibrio entre interpretabilidad y generalización. En coherencia con los procedimientos detallados en Materiales y Métodos, se presentan primero los resultados del modelo Random Forest y posteriormente los de la CNN-1D.

Modelo Random Forest

Para optimizar el modelo Random Forest se aplicó un proceso conjunto de selección de hiperparámetros y selección de características. En la primera etapa se exploraron distintas combinaciones de criterion, max_features y número de árboles. El mejor desempeño se obtuvo utilizando el criterio entropy, la opción sqrt para la selección aleatoria de características en cada nodo y 20 árboles, alcanzando un F1-macro de 86,83 %. Posteriormente se ejecutó un proceso de selección secuencial de características (SFS) con el fin de identificar un subconjunto más compacto de variables relevantes. Este procedimiento seleccionó cuatro características clave asociadas a la variabilidad y dispersión del tráfico de red: ICMP_response_time_jitter_max, Trafico_LAN_Transmitido_std, Trafico_LAN_Recibido_std y Paquetes_unicast_recibidos_ETH_max.

En la Tabla 5 se compara el rendimiento obtenido en ambos enfoques: el modelo optimizado mediante hiperparámetros y el modelo reducido con las mejor características. En general, ambos alcanzan un desempeño muy similar, con una exactitud cercana al 97 %. El modelo basado en selección de características presenta una ligera mejora en precisión (96,24 % frente a 94,81 %) y en las métricas F1 (81,27 %) y F2 (74,33 %), evidenciando una mayor proporción de predicciones positivas clasificadas correctamente. Por su parte, el modelo completo optimizado por hiperparámetros obtuvo valores superiores en ROC-AUC (94,21 %) y PR-AUC (84,11 %), indicando una mejor capacidad discriminativa en escenarios de desbalance de clases. Estos resultados muestran que la reducción de características no compromete el rendimiento general del modelo y, en algunos casos, mejora su estabilidad sin afectar su capacidad predictiva.

Tabla 5: Desempeño del modelo Random Forest con los mejores Hiperparámetros vs Mejores Características

Métrica	Hiperparámetros	Características
Accuracy	96,78 %	96,88 %
Precision	94,81 %	96,24 %
Recall	70,33 %	70,33 %
Puntuación F1	80,76 %	81,27 %
Puntuación F2	74,16 %	74,33 %
ROC-AUC	94,21 %	91,69 %
PR-AUC	84,11 %	81,21 %

La Figura 5 muestra las matrices de confusión del modelo Random Forest bajo dos configuraciones: (a) utilizando los hiperparámetros optimizados y (b) empleando el conjunto reducido de características seleccionadas. En ambos casos, el modelo mantiene una proporción muy alta de verdaderos negativos, superiores al 99,5 %, lo que evidencia una tasa de falsas alarmas prácticamente nula. En cuanto a la detección de fallos, el desempeño es estable en ambas configuraciones: el modelo identifica correctamente 128 de 182 fallos reales, alcanzando un recall de 70,33 %. Estas matrices confirman que la reducción de características no afecta la capacidad discriminativa del modelo y que conserva un equilibrio adecuado entre precisión y sensibilidad al distinguir entre condiciones normales y de fallo en la red LAN.

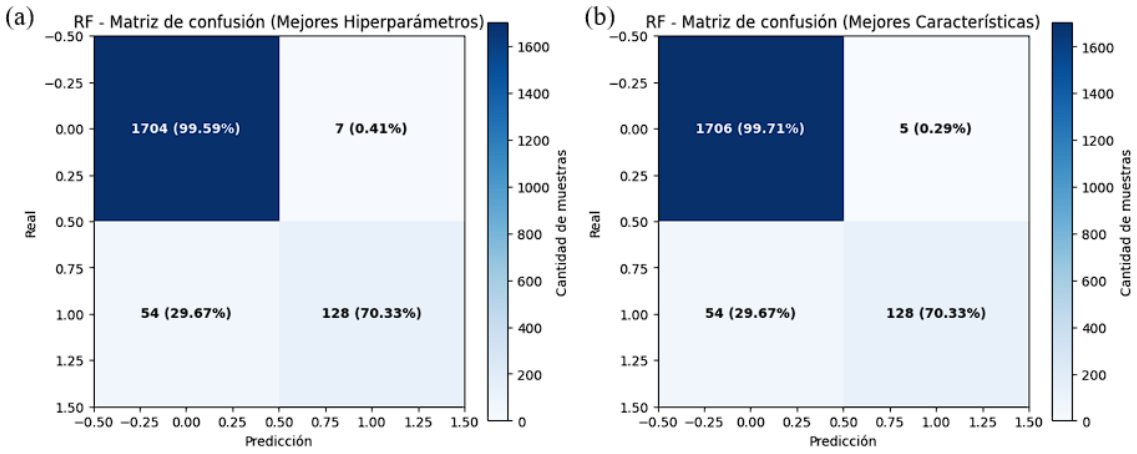

Figura 5: Matrices de confusión del modelo Random Forest: (a) mejores hiperparámetros y (b) mejores características.

La Figura 6 muestra las curvas ROC obtenidas para el modelo Random Forest bajo los dos enfoques evaluados: (a) utilizando los hiperparámetros optimizados y (b) considerando únicamente las características seleccionadas. En ambos casos, las curvas se aproximan al extremo superior izquierdo, lo que evidencia una alta sensibilidad y una baja tasa de falsos positivos. El área bajo la curva confirma este comportamiento: el modelo optimizado alcanzó un AUC de 94,21 %, mientras que el modelo reducido obtuvo un AUC de 91,69 %. Estos valores indican que, aunque la reducción de características disminuye ligeramente la capacidad discriminativa, el desempeño global se mantiene robusto y estable.

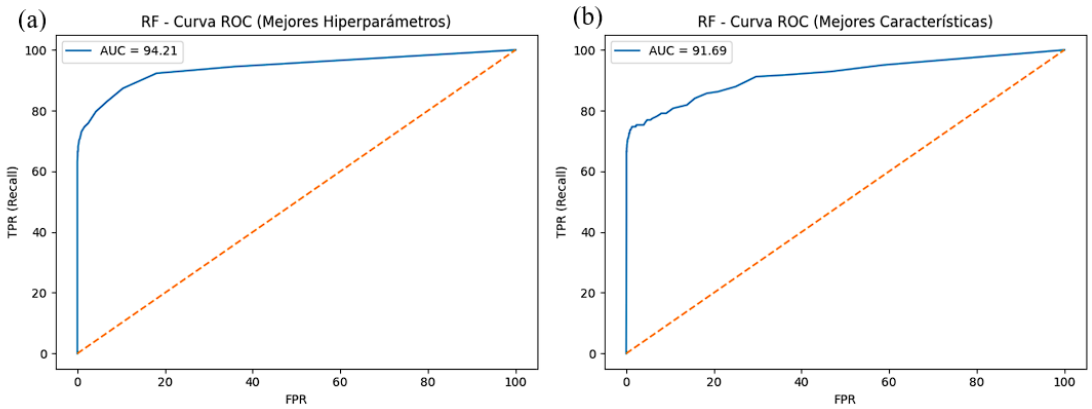

Figura 6: Curvas ROC del modelo Random Forest: (a) mejores hiperparámetros y (b) mejores características.

La Figura 7 presenta las curvas Precision–Recall de los modelos: (a) utilizando los hiperparámetros optimizados y (b) empleando únicamente las características seleccionadas. En ambos casos, las curvas mantienen valores elevados de precisión en los niveles altos de sensibilidad, lo que evidencia una baja proporción de falsos positivos incluso cuando se incrementa el recall. El área bajo la curva confirma este comportamiento: el modelo optimizado alcanzó un PR-AUC de 84,11 %, mientras que el modelo reducido obtuvo un PR-AUC de 81,21 %. Estos resultados indican que, aunque la reducción de características produce una ligera disminución en la capacidad para mantener precisión a medida que aumenta el recall, el desempeño global permanece estable y adecuado para escenarios con clases desbalanceadas.

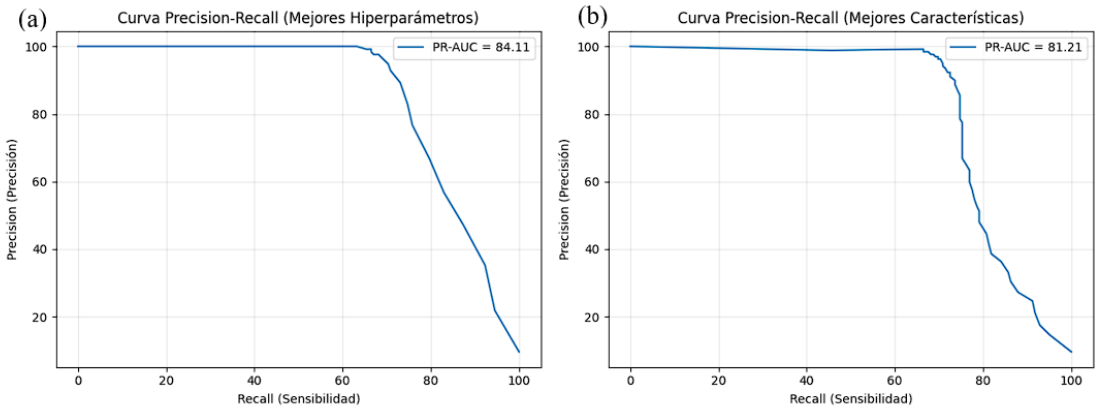

Figura 7: Curvas Precision–Recall del modelo Random Forest: (a) mejores hiperparámetros y (b) mejores características.

La Figura 8 muestra la evolución de las métricas del modelo Random Forest en función del umbral de decisión, considerando los enfoques de (a) mejores hiperparámetros y (b) mejores características. Las curvas corresponden a las métricas de exactitud (azul), precisión (naranja), recall (verde) y F1 (rojo), de acuerdo con la leyenda mostrada en la figura. En ambos casos, la exactitud se mantiene elevada y estable en la mayor parte del rango evaluado, con valores cercanos al 95 %. La precisión aumenta de manera sostenida conforme se incrementa el umbral, superando el 90 % en los niveles más altos. El recall, por el contrario, inicia en valores elevados para umbrales bajos, pero disminuye progresivamente hasta estabilizarse alrededor del 60 %. La puntuación F1 se mantiene cercana al 70 %, mostrando un comportamiento estable en la región óptima y descendiendo conforme el incremento del umbral reduce el número de instancias positivas correctamente identificadas. El umbral óptimo determinado para maximizar la métrica F1 fue de 46 % para el modelo con hiperparámetros optimizados y de 47 % para el modelo basado en características seleccionadas. En ambos casos, estos puntos reflejan un equilibrio adecuado entre precisión y sensibilidad, especialmente relevante en un escenario con clases desbalanceadas.

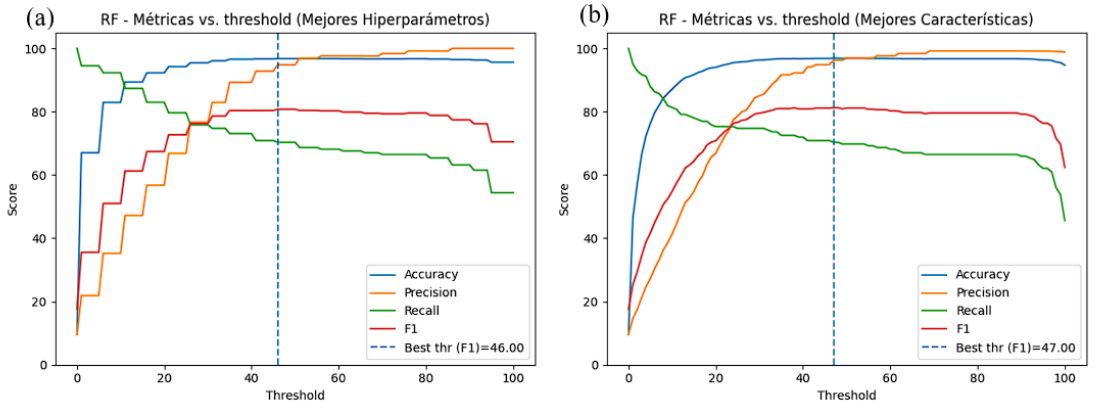

Figura 8: Métricas del modelo Random Forest en función del umbral: (a) mejores hiperparámetros y (b) mejores características.

La caída observada en la métrica F1 ante aumentos del umbral se debe al desbalance del conjunto de datos con aproximadamente el 9,6 % de fallos. En escenarios de este tipo, incluso incrementos pequeños del umbral reducen rápidamente el recall, ya que menos instancias de la clase positiva alcanzan probabilidades superiores al nuevo límite de decisión. Dado que F1 depende del equilibrio entre precisión y recall, su disminución provoca que la curva descienda con rapidez, aun cuando la precisión aumente ligeramente. Este comportamiento es consistente con la teoría de clasificación en datasets desbalanceados y se reproduce también en los experimentos posteriores.

Modelo CNN-1D

Una vez analizado el desempeño del modelo Random Forest, se evaluó el modelo basado en CNN-1D con el fin de comparar su capacidad predictiva bajo la misma estructura del problema. La Tabla 6 resume las métricas de desempeño obtenidas por el modelo CNN-1D. El modelo alcanzó una exactitud del 94,51 %, evidenciando un alto nivel de aciertos globales en la clasificación. La precisión fue del 68,31 %, lo que indica una proporción moderada de predicciones positivas correctas. El modelo detectó adecuadamente fallos reales, alcanzando un recall del 73,10 %, mientras que la puntuación F1 obtuvo 70,62 %, reflejando un equilibrio adecuado entre precisión y sensibilidad. La puntuación F2, más sensible al recall, alcanzó 72,09 %, mostrando un buen desempeño en escenarios donde es prioritario reducir falsos negativos. Finalmente, las métricas ROC-AUC (92,70 %) y PR-AUC (76,64 %) confirmaron una buena capacidad discriminativa incluso bajo condiciones de desbalance de clases, aunque con valores inferiores a los obtenidos con Random Forest.

Tabla 6: Desempeño del modelo CNN-1D

Métrica	CNN-1D (%)
Accuracy	94,51
Precision	68,31
Recall	73,10
Puntuación F1	70,62
Puntuación F2	72,09
ROC-AUC	92,70
PR-AUC	76,64

La Figura 9 muestra la matriz de confusión obtenida por el modelo CNN-1D. El modelo clasificó correctamente el 96,81 % de los estados normales (verdaderos negativos), con una tasa de falsas alarmas del 3,19 %. En la detección de fallos, alcanzó un 71,35 % de aciertos (verdaderos positivos), mientras que el 28,65 % de los casos de fallo no fueron detectados. Estos resultados indican que la CNN-1D logró una buena capacidad para distinguir entre condiciones normales y de fallo, con una ligera tendencia a priorizar la reducción de falsos positivos frente a la sensibilidad de detección.

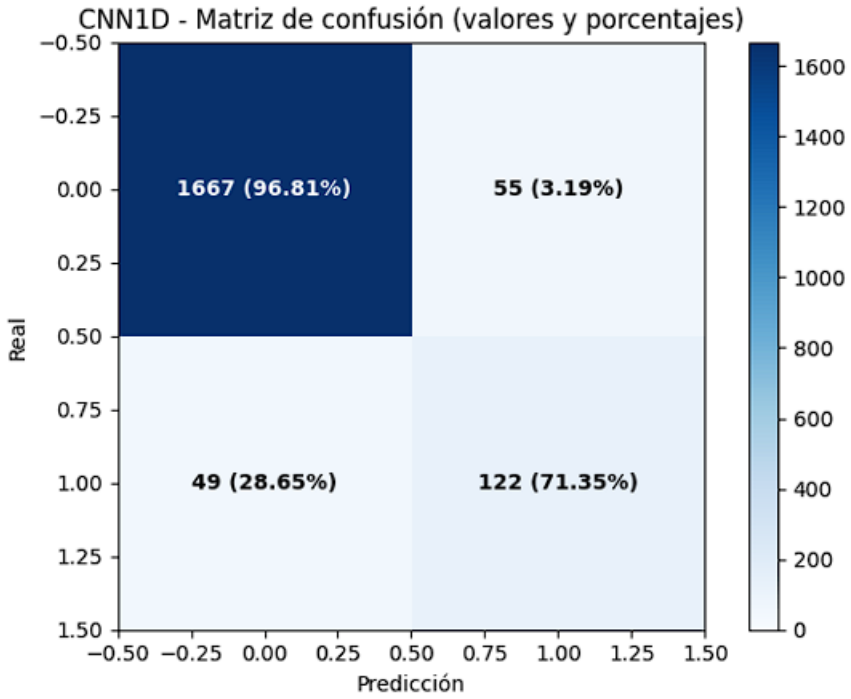

Figura 9: Matriz de confusión del modelo CNN-1D.

La Figura 10 muestra la curva ROC correspondiente al modelo CNN-1D. La curva se aproxima al extremo superior izquierdo, indicando una alta sensibilidad y baja tasa de falsos positivos. El área bajo la curva de 92,70 % demuestra una buena capacidad discriminativa del modelo para separar las clases normales y de fallo. Estos resultados confirman que la CNN-1D mantuvo un rendimiento competitivo, comparable al obtenido con el modelo Random Forest, pese a utilizar una arquitectura de mayor complejidad

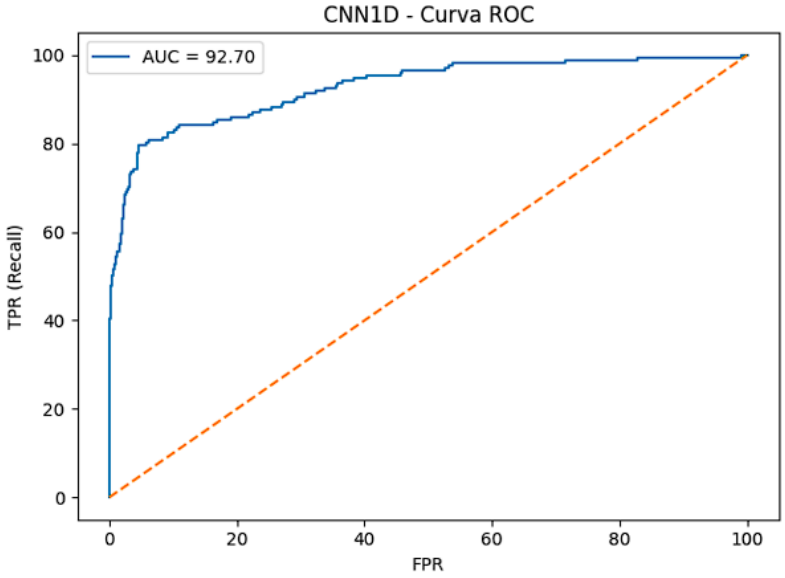

Figura 10: Curva ROC del modelo CNN-1D.

En la Figura 11 se muestra el resultado de la curva Precision–Recall del modelo CNN-1D, obteniendo 76.64 %. La curva indica alta precisión en los niveles iniciales de sensibilidad, haciendo referencia a que el modelo mantiene una baja tasa de falsos positivos en las predicciones más confiables. El área bajo la curva indica un buen equilibrio entre precisión y recall, demostrando que la red neuronal es capaz de identificar fallos de manera efectiva, incluso ante la presencia de desbalance en las clases.

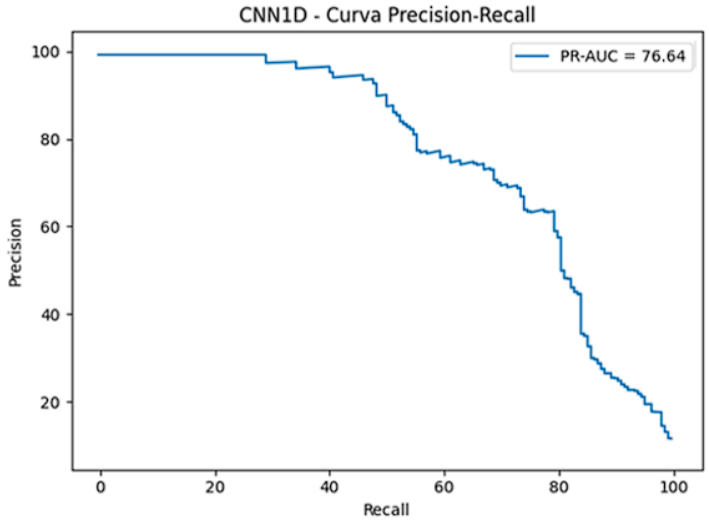

Figura 11: Curva ROC del modelo CNN-1D.

La Figura 12 muestra la variación en las métricas del modelo CNN-1D según el umbral de decisión. La exactitud se mantuvo elevada y estable en todo el rango, con valores cercanos al 95 %. La precisión aumentó progresivamente conforme se elevó el umbral, superando el 90 % en los niveles más altos. El recall mostró el comportamiento opuesto: inició con valores altos en umbrales bajos y disminuyó hasta estabilizarse cerca del 60 %. La puntuación F1 alcanzó su valor máximo alrededor del 46 %, punto identificado como umbral óptimo, donde el modelo logró un equilibrio adecuado entre precisión y sensibilidad.

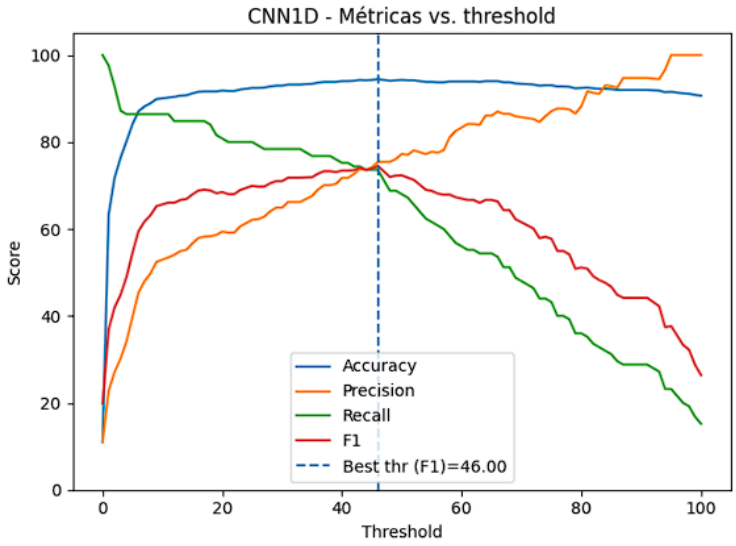

Figura 12: Métricas del modelo CNN-1D en función del umbral.

Finalmente, de acuerdo con la comparación general presentada en la Tabla 7, el modelo Random Forest evidenció el mejor desempeño global entre los modelos evaluados. La versión optimizada mediante hiperparámetros obtuvo una exactitud superior al 96 % y una precisión del 94,81 %, reflejando una alta capacidad para identificar correctamente los estados normales y minimizar las falsas alarmas. La variante con selección de características mantuvo métricas muy similares, lo que confirma que la reducción del número de variables no afectó de manera significativa su capacidad predictiva, preservando su estabilidad y consistencia. Por otra parte, la CNN-1D alcanzó el mayor recall (73,10 %) entre los modelos comparados, lo que indica una mayor sensibilidad para detectar fallos reales, aunque con una precisión menor (68,31 %), lo que incrementa la probabilidad de falsos positivos. En términos de capacidad discriminativa, todos los modelos superaron el 90 % en ROC-AUC, evidenciando un rendimiento sólido incluso en condiciones de desbalance de clases.

Tabla 7: Desempeño comparativo entre los modelos Random Forest y CNN-1D

Métrica	RF Selección Hiperparámetros	RF Selección de características	CNN-1D
Accuracy	96,78 %	96,88 %	94,51 %
Precision	94,81 %	96,24 %	68,31 %
Recall	70,33 %	70,33 %	73,10 %
Puntuación F1	80,76 %	81,27 %	70,62 %
Puntuación F2	74,16 %	74,33 %	72,09 %
ROC-AUC	94,21 %	91,69 %	92,70 %
PR-AUC	84,11 %	81,21 %	76,64 %

En conjunto, los resultados sugieren que Random Forest es el modelo más estable, consistente y explicable, adecuado para su despliegue en entornos operativos de monitoreo. No obstante, la CNN-1D constituye una alternativa más sensible ante eventos de fallo, lo que la convierte en un enfoque complementario para escenarios en los que la prioridad es maximizar la detección temprana de incidencias en redes LAN.

4. Discusión

En el escenario experimental, un fallo en la red LAN se definió como la presencia de condiciones anómalas detectadas en los puntos de acceso, tales como pérdida de conectividad ICMP, incrementos abruptos de tiempo de respuesta, picos de jitter, congestión repentina del tráfico LAN, caída temporal del servicio o ausencia de paquetes unicast. Estos eventos representan las incidencias reales que afectan la continuidad del servicio en la red LAN de la UPSE y constituyen la base sobre la cual los modelos aprendieron patrones para anticipar estados de fallo.

Los resultados demostraron que ambos modelos, Random Forest y la CNN-1D, alcanzaron un desempeño sólido en la predicción de fallos en la red LAN de la UPSE. El modelo Random Forest registró una exactitud superior al 96 % y un equilibrio adecuado entre precisión, aproximadamente 94 % y un recall cercano a 70 %. Esto evidencia su capacidad para detectar incidentes sin incrementar las falsas alarmas. De manera similar, la CNN-1D logró una exactitud del 94 % y un recall ligeramente mayor con 73,10 %, lo que confirma su habilidad para reconocer secuencias anómalas en indicadores temporales.

La diferencia entre ambos modelos se debió en gran medida al fuerte desbalance de clases en el conjunto de datos (729 fallos frente a 6 842 normales). Este desequilibrio influyó directamente en el rendimiento de ambos algoritmos. El algoritmo Random Forest tendió a favorecer la clase mayoritaria y logró una alta precisión, pero menor recall. La CNN-1D mostró una mayor sensibilidad hacia la clase minoritaria y detectó más fallos a costa de un aumento de falsos positivos. Esta relación inversa entre precisión y recall destaca la complementariedad de ambos modelos y justifica el uso de una estrategia híbrida, en la que el algoritmo Random Forest actúa como un filtro de alta confianza y el algoritmo CNN-1D como un detector sensible. Tal combinación permitiría mitigar los efectos del desbalance y mejorar la detección temprana de fallos en entornos reales de red.

Estos hallazgos coinciden parcialmente con lo reportado por Myrzatay et al. [19] [19 p. 9] quienes integraron técnicas de suavizado exponencial con algoritmos supervisados para predecir fallos en switches LAN. Aunque sus resultados mostraron un desempeño aceptable, con precision de 81,2 % y recall de 61,9 %, el presente estudio superó dichas métricas al alcanzar valores de F1 de 80 % y recall de 70 % en la red de la UPSE. Esta mejora se debe al uso de indicadores SNMP más representativos y a un marco de validación más riguroso, que permitió a los modelos capturar patrones más consistentes relacionados con las condiciones de error.

En términos comparativos, Random Forest demostró una estabilidad y generalización comparativamente mayores, mientras que la CNN-1D destacó por su capacidad de adaptación a patrones temporales complejos. Estos resultados concuerdan con los hallazgos de Murphy et al. [42], quienes reportaron un mejor rendimiento de los modelos basados en árboles bajo restricciones de datos en comparación con las redes profundas con mayor sensibilidad a las series temporales. En general, los resultados demuestran que ambos modelos son complementarios y útiles en escenarios operativos con diferente granularidad temporal.

La selección secuencial de características (SFS) confirmó que el subconjunto reducido de variables —ICMP response time jitter max, Trafico LAN Transmitido std, Trafico LAN Recibido std y Paquetes unicast recibidos ETH max— conservó un rendimiento equivalente al modelo completo, alcanzando un F1 de 81 % y un AUC-ROC de 91,69 %. Esto coincide con Edozie et al. [43] quienes resaltan la relevancia de las métricas de variabilidad y dispersión en la predicción de anomalías. La reducción de variables sin pérdida de rendimiento refuerza la eficiencia del enfoque propuesto y su aplicabilidad en entornos con recursos de monitoreo limitados.

Los valores obtenidos de ROC- AUC (94 % en Random Forest y 92 % en CNN-1D) y de PR-AUC (84 % en Random Forest y 77 % en CNN-1D). Estos resultados confirmaron que los indicadores SNMP, procesados y normalizados adecuadamente, pueden servir como base fiable para la predicción de fallos en redes universitarias. Además, destacaron la viabilidad de aplicar algoritmos supervisados como parte de un sistema de gestión proactiva en infraestructuras reales.

Las limitaciones del estudio incluyeron el desbalance de clases, el uso de una única topología (38 puntos de acceso Ruckus) y el tamaño limitado del conjunto de datos. Sin embargo, el diseño experimental —con validación cruzada estratificada y calibración de umbral en validación— minimizó el riesgo de sobreajuste y garantizó la reproducibilidad de los resultados. En trabajos futuros se podría explorar técnicas de balanceo sintético (SMOTE), arquitecturas híbridas basadas en Transformers y evaluaciones en redes heterogéneas para ampliar la generalización del modelo.

A diferencia de los estudios previos centrados en redes corporativas o infraestructuras controladas, el escenario universitario analizado presenta una dinámica temporal más compleja, con variaciones horarias pronunciadas, alta movilidad de usuarios y tráfico heterogéneo asociado a dispositivos personales, laboratorios académicos y actividades de docencia. Este comportamiento explica las diferencias observadas respecto a trabajos como Myrzatay et al [19]. y Murphy et al. [42], donde las redes analizadas exhibían menor variabilidad y condiciones operativas más estables. En este contexto, la mayor sensibilidad de la CNN-1D a patrones transitorios y la mayor estabilidad del modelo Random Forest ante ruido estadístico reflejan características propias de la telemetría SNMP en campus universitarios. Por tanto, los resultados no se limitan a replicar tendencias de la literatura, sino que aportan evidencia específica sobre cómo estos algoritmos responden ante patrones reales de uso académico, fortaleciendo su aplicabilidad en entornos educativos.

En resumen, la evidencia empírica confirmó que la predicción de fallos mediante aprendizaje supervisado es una estrategia viable y eficaz para las redes LAN universitarias. La combinación de modelos explicables y redes profundas representa un avance tangible hacia la automatización del monitoreo y la optimización operativa de la infraestructura de red de la UPSE.

5. Conclusiones

El estudio confirmó que los indicadores obtenidos mediante SNMP —latencia ICMP, tráfico LAN y paquetes unicast— contienen información suficiente para anticipar eventos de fallo en la red universitaria, entendidos operacionalmente como pérdida de respuesta ICMP, reinicios inesperados (cambios en el uptime) o degradación sostenida del rendimiento. Estos eventos permitieron construir un conjunto de datos supervisado adecuado para la predicción en entornos reales.

Los modelos Random Forest y CNN-1D demostraron fortalezas complementarias: Random Forest destaca por su alta precisión y baja generación de falsas alarmas, mientras que la CNN-1D exhibe mayor sensibilidad ante patrones anómalos en secuencias temporales, detectando un número superior de fallos reales. Esta dualidad respalda la adopción de un esquema híbrido, en el que la CNN-1D opere como detector temprano y Random Forest como mecanismo de verificación. De manera operativa, se recomienda mantener los umbrales calibrados mediante validación interna, priorizando aquellos que maximizan la métrica F1 en escenarios con clases desbalanceadas.

La principal limitación del estudio fue el desbalance natural del dataset, típico de redes estables donde los fallos son poco frecuentes. Para investigaciones futuras, se sugiere explorar métodos de balanceo como SMOTE o modelos

generativos únicamente para ampliar la representación de la clase minoritaria, sin sustituir a los modelos predictivos principales. También será pertinente validar el modelo en otras topologías académicas y considerar arquitecturas modernas de series temporales, como Transformers.

En conjunto, los resultados evidencian que la combinación de modelos explicables y redes profundas es una estrategia eficaz para fortalecer la gestión predictiva de la infraestructura de red de la UPSE. Este enfoque sienta las bases para una transición sostenible hacia sistemas de monitoreo proactivo impulsados por inteligencia artificial en entornos universitarios.

Financiamiento:

Los autores expresan que no ha sido necesario financiamiento para realizar esta obra de investigación.

Conflicto de intereses:

Los autores declaran no tener conflicto de intereses.

Contribución de autor/es:

Bajo los lineamientos CRediT (Taxonomía de Roles de Contribuyente), los autores dan fe de las contribuciones realizadas al trabajo de investigación, que se detallan: Autor principal: Ariel Oswaldo Fernández Loor, 70 %, Visualización; Revisión–Edición; Redacción del borrador original; Validación; Recursos y materiales; Software; Análisis de datos; Conducción de la investigación; Curación de datos; Metodología; Conceptualización. Coautora 1: Alicia Germania Andrade Vera, 30 %, Revisión–Edición; Supervisión; Administración del proyecto; Metodología; Conceptualización.

6. Referencias

1. SHIRATSUCHI H., K. HORIUCHI y MATSUZAKI, T. *Studies on development of web-based integrated learning and education support system*. *ICIC Express Letters, Part B: Applications [online]*. 2020, vol. 11, n.º 2, págs. 197-205. Disponible en: https://doi.org/10.24507/icicelb.11.02.197.
2. SILVA ROCHA, Élisson da; SILVA, Leylane G. F. da; SANTOS, Guto L.; BEZERRA, Diego; MOREIRA, André; GONÇALVES, Glauco; MARQUEZINI, Maria Valéria; MEHTA, Amardeep; WILDEMAN, Mattias; KELNER, Judith y ENDO, Pedro T. *Aggregating data center measurements for availability analysis*. *Software: Practice and Experience [online]*. 2021, vol. 51, n.º 5, págs. 868-892. Disponible en: https://doi.org/https://doi.org/10.1002/spe.2934.
3. FATHIMA, A. y DEVI, G.S. *Enhancing university network management and security: a real-time monitoring, visualization & cyber attack detection approach using Paessler PRTG and Sophos Firewall*. *International Journal of System Assurance Engineering and Management [online]*. 2024, pág. . Disponible en: https://doi.org/10.1007/s13198-024-02448-y.
4. ALHARI, M.I. y M. LUBIS. *ALHARI, M.I. y M. LUBIS. 2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. 2023, págs. 345-349. Disponible en: https://doi.org/10.1109/IAICT59002.2023.10205625..
5. ESPINEL VILLALOBOS, R.I.; TRIANA, E. ARDILA; CEBALLOS, H. ZARATE y TRIVIÑO, J.E. ORTIZ. *Diseño e implementación de un sistema de monitoreo de red para infraestructura de campus usando agentes de software*. *Ingenieria e Investigacion [online]*. 2022, vol. 42, n.º 1. Disponible en: https://doi.org/10.15446/ing.investig.v42n1.87564.
6. SETYANTORO, D; AFIFAH, V.; HASIBUAN, R.A.; APRILIA, N. y SARI., N.P. *The Wireless Computer Network Management Security Analysis*. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer) [online]*. 2022, vol. 7, n.º 2, págs. 105-110. Disponible en: https://doi.org/10.33480/jitk.v7i2.2786.
7. HELALI, S. *Monitoring Systems and Networks*. *Systems and Network Infrastructure Integration: Design, Implementation, Safety and Supervision*. 2020, págs. 157-171. Disponible en: https://doi.org/10.1002/9781119779964.ch9.
8. MAU, D.O. *Integrated Intelligent Agent for SNMP-Based Network Management System*. *Industrial Networks and Intelligent Systems*. 2020, págs. 19-33.
9. STOYKOVA, S. y N. SHAKEV. *Artificial Intelligence for Management Information Systems: Opportunities, Challenges, and Future Directions*. *Algorithms [online]*. 2023, vol. 16, n.º 8. Disponible en: https://doi.org/10.3390/a16080357.
10. MOOSA M.A., A.K. VANGUJAR y D.P. MAHAJAN. *Detection and Analysis of DDoS Attack Using a Collaborative Network Monitoring Stack*. *16th International Conference on Security of Information and Networks, SIN 2023*. 2023, pág. . Disponible en: https://doi.org/10.1109/SIN60469.2023.10474700.
11. MOHAMMED B., M. KIRAN y B. ENDERS. *NetGraf: An End-to-End Learning Network Monitoring Service*. *2021 IEEE Workshop on Innovating the Network for Data-Intensive Science (INDIS)*. 2021, págs. 12-22. Disponible en: https://doi.org/10.1109/INDIS54524.2021.00007.

12. AL-NAYMAT G., M. AL-KASASSBEH y E. AL-HAWARI. *Using machine learning methods for detecting network anomalies within SNMP-MIB dataset. International Journal of Wireless and Mobile Computing [online]*. 2018, vol. 15, n.º 1, págs. 67-76. Disponible en: https://doi.org/10.1504/IJWMC.2018.094644.

13. SAYED, M.S. El; LE-KHAC, N.-A.; AZER, M.A. y JURCUT, A.D. *A Flow-Based Anomaly Detection Approach With Feature Selection Method Against DDoS Attacks in SDNs*. *IEEE Transactions on Cognitive Communications and Networking [online]*. 2022, vol. 8, n.º 4, págs. 1862-1880. Disponible en: https://doi.org/10.1109/TCCN.2022.3186331.

14. PAES D.S.F., C.H.V. DE MORAES y B.G. BATISTA. *Analysis of supervised machine-learning techniques in computer networks attack detection*. *Computer Communications [online]*. 2025, vol. 240, pág. 108203. Disponible en: https://doi.org/https://doi.org/10.1016/j.comcom.2025.108203..

15. ZHAO, S.; CHANDRASHEKAR, M.; LEE, Y. y MEDHI, D. *Real-time network anomaly detection system using machine learning*. *2015 11th International Conference on the Design of Reliable Communication Networks, DRCN 2015*. 2015, vol. 2015, págs. 267-270. Disponible en: https://doi.org/10.1109/DRCN.2015.7149025.

16. SAFARI, A.; SOROURI, H.; RAHIMI, A. y OSHNOEI, A. *A Systematic Review of Energy Efficiency Metrics for Optimizing Cloud Data Center Operations and Management*. *Electronics [online]*. 2025, vol. 14, n.º 11. Disponible en: https://doi.org/10.3390/electronics14112214.

17. GUAN J., J. LU y W. WANG. *17. 2023 IEEE 6th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*. 2023, págs. 1055-1059. Disponible en: https://doi.org/10.1109/AUTEEE60196.2023.10407966.

18. AZAM Z., Md.M. ISLAM y M.N. HUDA. *Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree*. *IEEE Access [online]*. 2023, vol. 11, págs. 0348-8039. 1. Disponible en: https://doi.org/10.1109/ACCESS.2023.3296444.

19. MYRZATAY, A.; RZAYEVA, L.; BANDINI, S.; SHAYEA, I.; SAOUD, B.; ÇOLAK, I. y KAYISLI, K. *Predicting LAN switch failures: An integrated approach with DES and machine learning techniques (RF/LR/DT/SVM)*. *Results in Engineering [online]*. 2024, vol. 23, pág. 102356. Disponible en: https://doi.org/10.1016/J.RINENG.2024.102356.

20. MAJUMDER S., M.K. DEB BARMA y A. SAHA. *ARP spoofing detection using machine learning classifiers: an experimental study*. *Knowl. Inf. Syst. [online]*. 2024, vol. 67, n.º 1, págs. 727-766. Disponible en: https://doi.org/10.1007/s10115-024-02219-y.

21. HASSANAT, A.B.; TARAWNEH, A.S.; ABED, S.S.; ALTARAWNEH, G.A.; ALRASHIDI, M. y RDPVR, M. ALGHAMDI. *Random Data Partitioning with Voting Rule for Machine Learning from Class-Imbalanced Datasets*. *Electronics [online]*. 2022, vol. 11, n.º 2. Disponible en: https://doi.org/10.3390/electronics11020228.

22. MISHRA S., A. ALBARAKATI y S.K. SHARMA. *Cyber Threat Intelligence for IoT Using Machine Learning*. *Processes [online]*. 2022, vol. 10, n.º 2. Disponible en: https://doi.org/10.3390/pr10122673.

23. KIRUTHIKA DEVI, B.S.; ARAVINDHAN, K.; KINI, P.S.; REDDY, G.A. y SUBBULAKSHMI, T. *A Prediction Model for Flooded Packets in SNMP Networks*. *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2018, págs. 82-86. Disponible en: https://doi.org/10.1109/ICCONS.2018.8662972.

24. GARG, S.; KAUR, K.; KUMAR, N. y RODRIGUES, J.J.P.C. *Hybrid Deep-Learning-Based Anomaly Detection Scheme for Suspicious Flow Detection in SDN: A Social Multimedia Perspective*. *IEEE Transactions on Multimedia [online]*. 2019, vol. 21, n.º 3, págs. 566-578. Disponible en: https://doi.org/10.1109/TMM.2019.2893549.

25. IZADI S., M. AHMADI y A. RAJABZADEH. *Network Traffic Classification Using Deep Learning Networks and Bayesian Data Fusion*. *Journal of Network and Systems Management [online]*. 2022, vol. 30, n.º 2. Disponible en: https://doi.org/10.1007/s10922-021-09639-z.

26. NOETZOLD, D.; ROSSETTO, A.G.D.M.; LEITHARDT, V.R.Q. y M. COSTA, H.J. de. *Enhancing Infrastructure Observability: Machine Learning for Proactive Monitoring and Anomaly Detection*. *Journal of Internet Services and Applications [online]*. 2024, vol. 15, n.º 1, págs. 508-522. Disponible en: https://doi.org/10.5753/jisa.2024.4509.

27. AHSAN, M.; GOMES, R.; CHOWDHURY, Md.M. y NYGARD, K.E. *Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector*. *Journal of Cybersecurity and Privacy [online]*. 2021, vol. 1, n.º 1, págs. 199-218. Disponible en: https://doi.org/10.3390/jcp1010011.

28. SUN, Y.; OCHIAI, H. y ESAKI., H. *Multi-Type Anomaly Detection Based on Raw Network Traffic*. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. 2021, págs. 1-2. Disponible en: https://doi.org/10.1109/CCNC49032.2021.9369654.

29. REN, L.; JIA, Z.; WANG, T.; MA, Y. y WANG, L. *LM-CNN: A Cloud-Edge Collaborative Method for Adaptive Fault Diagnosis With Label Sampling Space Enlarging*. *IEEE Transactions on Industrial Informatics [online]*. 2022, vol. 18, n.º 12, págs. 9057-9067. Disponible en: https://doi.org/10.1109/TII.2022.3180389.

30. SURYOTRISONGKO, H.; MUSASHI, Y.; TSUNEDA, A. y SUGITANI, K. *Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing*. *IEEE Access [online]*. 2022, vol. 10, págs. 34613-34624. Disponible en: https://doi.org/10.1109/ACCESS.2022.3162588.

31. PORTELA, A.L.; MENEZES, R.A.; COSTA, W.L.; SILVEIRA, M.M.; BITTECNOURT, L.F. y GOMES, R.L. *Detection of IoT Devices and Network Anomalies based on Anonymized Network Traffic*. *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023, pág. 1-6. Disponible en: https://doi.org/10.1109/NOMS56928.2023.10154276.

32. PRIYA, S.S.; SIVARAM, M.; YUVARAJ, D. y JAYANTHILADEVI, A. *Machine Learning based DDOS Detection*. *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*. 2020, págs. 234-237. Disponible en: https://doi.org/10.1109/ESCI48226.2020.9167642.

33. MITROPOULOU, K.; KOKKINOS, P.; SOUMPLIS, P. y VARVARIGOS, E. *Anomaly Detection in Cloud Computing using Knowledge Graph Embedding and Machine Learning Mechanisms*. *Journal of Grid Computing [online]*. 2023, vol. 22, n.º 1, pág. 6. Disponible en: https://doi.org/10.1007/s10723-023-09727-1.

34. FATHI-KAZEROONI, S.; KAYMAK, Y. y ROJAS-CESSA, R. *Tracking User Application Activity by using Machine Learning Techniques on Network Traffic*. *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. 2019, págs. 405-410. Disponible en: https://doi.org/10.1109/ICAIIC.2019.8669040.

35. SCHUMMER, P.; RIO, A. DEL; SERRANO, J.; JIMENEZ, D.; SÁNCHEZ, G. y LLORENTE, Á. *Machine Learning-Based Network Anomaly Detection: Design, Implementation, and Evaluation*. *AI [online]*. 2024, vol. 5, n.º 4, págs. 2967-2983. Disponible en: https://doi.org/10.3390/ai5040143.

36. HOU, Y.; XU, Z.; WANG, L.; WANG, Y. y LI., H. *NadGPT: Semi-Supervised Network Anomaly Detection via Auto-Regressive Auxiliary Prediction*. *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2023, págs. 133-138. Disponible en: https://doi.org/10.1109/SMC53992.2023.10394639.

37. VEMULA, M.B.; KUMAR, K.P.; THIPPARTHI, H.R.S. y JASTI, P.S. *Network Anomaly Detection Using Generative Adversarial Networks. 2024 First International Conference on Software, Systems and Information Technology (SSITCON)*. 2024, págs. 1-6. Disponible en: https://doi.org/10.1109/SSITCON62437.2024.10796515.

38. DUY, P.T.; TIEN, L.K.; KHOA, N.H.; HIEN, D.T.T.; NGUYEN, A.G.-T. y PHAM, V.-H. *DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks*. *Computers & Security [online]*. 2021, vol. 109, pág. 102367. Disponible en: https://doi.org/https://doi.org/10.1016/j.cose.2021.102367.

39. STEPHAN, M.; ZERWAS, J. y KELLERER, W. *T-MAW: Online Network Traffic Monitoring and Analysis using Weighted Stochastic Block Models*. *2024 20th International Conference on Network and Service Management (CNSM)*. 2024, págs. 1-9. Disponible en: https://doi.org/10.23919/CNSM62983.2024.10814420.

40. GUO, Y.; WANG, Y.; KHAN, F.; AL-ATAWI, A.A.; ABDULWAHID, A. Al; LEE, Y. y MARAPELLI, B. *Traffic Management in IoT Backbone Networks Using GNN and MAB with SDN Orchestration*. *Sensors [online]*. 2023, vol. 23, n.º 16. Disponible en: https://doi.org/10.3390/s23167091.

41. CHANGO, W.; J. ERAZO, P. BUÑAY an; AGUILAR, P.; SAYAGO, J.; FLORES, A. y SILVA, G. *Predicting Urban Traffic Congestion with VANET Data*. *Computation*

[online]. 2025, vol. 13, n.º 4. Disponible en: https : / / doi . org / 10 . 3390 / computation13040092.

42. MURPHY, K.; LAVIGNOTTE, A. y LEPERS, C. *Fault Prediction for Heterogeneous Telecommunication Networks Using Machine Learning: A Survey*. *IEEE Transactions on Network and Service Management [online]*. 2024, vol. 21, n.º 2, págs. 2515-2538. Disponible en: https://doi.org/10.1109/TNSM.2023.3340351.

43. EDOZIE, E.; SHUAIBU, A.N.; SADIQ, B.O. y JOHN, U.K. *Artificial intelligence advances in anomaly detection for telecom networks*. *Artificial Intelligence Review [online]*. 2025, vol. 58, n.º 4, pág. 100. Disponible en: https://doi.org/10.1007/s10462-025-11108-x.