#ashtag
REVISTA ESPECIALIZADA EN INGENIERÍA
ISSN electrónico: 2346 - 139X
Corporación Unificada Nacional de Educación Superior

Identificación de ataques de Malware móvil en dispositivos Android mediante algoritmos de aprendizaje automático

Identifying Mobile Malware Attacks on Android Devices Using Machine learning Algorithms

Víctor Guzmán-Brand

Psicólogo-Especialista Analítica de datos; investigador;
Instituto Superior de Educación Rural (ISER);
https://orcid.org/0000-0002-6051-3153;
vaguzmanbrand@gmail.com.

Laura Gélvez-García

Licenciada en Lengua Castellana-Magíster en Lingüística Española;
Doctora en Ciencias de la educación; Docente Investigadora;
https://orcid.org/0000-0003-0164-2972;
laura.gelvez@cun.edu.co.

Cómo citar: Guzmán-Brand, V., y Gélvez-García, L. (2025). Identificación de ataques de Malware móvil en dispositivos Android mediante algoritmos de aprendizaje automático. *#ashtag*, 1(26), 39-51. https://doi.org/10.52143/2346139X.1076

Resumen

El presente artículo tiene como objetivo dentificar ataques de Malware móvil en dispositivos Android mediante algoritmos de aprendizaje automático. La metodología empleada se basa en el proceso KDD (*Knowledge Discovery in Databases*), un enfoque estructurado que organiza la minería de datos en etapas claramente definidas. Este modelo garantiza un control preciso en cada fase, permitiendo una extracción, transformación y análisis de la información de manera eficiente. Como resultado se obtuvo que el algoritmo LightGBM demuestra, a través de la matriz de confusión, su capacidad para procesar eficientemente grandes volúmenes de datos y múltiples características, hecho que contribuye a una clasificación más precisa. Además, este modelo sobresale en las métricas de evaluación, logrando un rendimiento óptimo en comparación con otros enfoques de aprendizaje automático. Se abre la discución acerca del aprendizaje automático, clave en ciberseguridad para mejorar la detección de amenazas como malware y ataques DDoS. LightGBM se destaca por su eficiencia logrando la mejor precisión (94.4%), seguido por XGBoost con alto desempeño, pero mayor tiempo de cómputo. Random Forest, aunque más rápido, presenta menor precisión. En conclusión el aprendizaje automático ha revolucionado la ciberseguridad, fortaleciendo la detección de amenazas como malware e intrusiones. En la identificación de malware móvil en Android, LightGBM se destacó por su precisión y eficiencia en el manejo de datos desbalanceados, superando a otros modelos en métricas clave. Sin embargo, el desafío sigue siendo equilibrar precisión y consumo de recursos, especialmente en dispositivos móviles con hardware limitado.

Palabras Clave: Identificar; Malware; dispositivos; Android; algoritmo; aprendizaje automático.

Abstract

Objective: to identify mobile malware attacks on Android devices using machine learning algorithms. Methodology: based on the KDD (Knowledge Discovery in Databases) process, a structured approach that organizes data mining in clearly defined stages. This model ensures precise control at each stage, allowing for efficient information extraction, transformation and analysis. Results: The LightGBM algorithm demonstrates, through the confusion matrix, its ability to efficiently process large volumes of data and multiple features, contributing to more accurate classification. In addition, this model excels in evaluation metrics, achieving optimal performance compared to other machine learning approaches. Discussion: Machine learning is key in cybersecurity, improving the detection of threats such as malware and DDoS attacks. LightGBM stands out for its efficiency achieving the best accuracy (94.4%), followed by XGBoost with high performance, but longer computation time. Random Forest, although faster, presents lower accuracy. Conclusions: Machine learning has revolutionized cybersecurity, strengthening the detection of threats such as malware and intrusions. In identifying mobile malware on Android, LightGBM stood out for its accuracy and efficiency in handling unbalanced data, outperforming other models in key metrics. However, the challenge remains balancing accuracy and resource consumption, especially on mobile devices with limited hardware.

Keywords: Identify; Malware; devices; Android; algorithm; machine learning.

Recibido: 18-02-2025 | Aceptado: 01-06-2025 | Publicado: 29-06-2025

Introducción

El análisis de las tecnologías de telecomunicaciones y sus vulnerabilidades es esencial para prever amenazas frente a la seguridad de los datos. Atacantes con recursos reducidos pueden interceptar tráfico en redes móviles, comprometiendo información crítica en tránsito o almacenada localmente (Álvarez & Montoya, 2020). Los delincuentes interceptan credenciales personales, monitorean correos electrónicos y comprometen datos bancarios, exponiendo a las víctimas a graves riesgos financieros y de privacidad. La falta de percepción sobre estas amenazas agrava su impacto, permitiendo que los ciberdelincuentes operen con impunidad (Cassinda, 2019).

El informe anual de Kaspersky (2024) sobre amenazas móviles evidencia un alarmante incremento en los riesgos de seguridad, impulsado por el desarrollo de herramientas maliciosas cada vez más sofisticadas. Se registró un total de casi 33,8 millones de ataques durante el período analizado, hecho que representa un aumento superior al 50% en comparación con el año anterior. Dentro de este panorama, el adware emergió como la principal amenaza, siendo responsable del 40,8% de los incidentes detectados. Este tipo de software, caracterizado por la visualización no autorizada de anuncios emergentes, afecta significativamente la experiencia del usuario y pone en riesgo la seguridad de los dispositivos móviles.

Ante el creciente desafío de la seguridad informática, los sistemas de detección de malware se desarrollan y perfeccionan de manera continua. Dado que los dispositivos móviles cuentan con recursos computacionales limitados, es fundamental diseñar detectores de malware eficientes que operen con rapidez y mantengan una alta precisión en la identificación de amenazas. El proceso consiste en clasificar aplicaciones móviles desconocidas como benignas o maliciosas; en este contexto, los enfoques basados en aprendizaje automático (ML) han demostrado ser herramientas eficaces para la detección de malware (Mohammed & Awad, 2022).

Al respecto, numerosos estudios han explorado el uso de la inteligencia artificial, combinando técnicas de aprendizaje automático y profundo, para fortalecer la seguridad en redes móviles. Alkahtani & Aldhyani (2022) destacaron la eficacia de técnicas como las máquinas de vectores de soporte (SVM), las redes neuronales recurrentes de largo corto plazo (LSTM) y la combinación de redes convolucionales con LSTM (CNN-LSTM) para la detección de malware en dispositivos móviles. Por otro lado, se encuentra la aplicación de algoritmos como Support Vector Machine, k-Nearest Neighbor, Naïve Bayes para la clasificación (Bashir *et al.*, 2024).

En este mismo sentido, el trabajo de Milosevic *et al.* (2017) presenta dos enfoques de aprendizaje automático, clasificación y agrupamiento, para detectar malware en Android mediante el análisis de permisos y código fuente. Así mismo, en la detección y clasificación de malware en aplicaciones Android, se han desarrollado modelos híbridos que integran análisis estático y dinámico con el propósito de optimizar la precisión y eficiencia del proceso (Z. Liu *et al.*, 2021). Igualmente, experimentos con algoritmos clásicos como Random Forest muestran buenas métricas de precisión del 97.20% (Iqubal & Payal, 2024).

El objetivo de esta investigación es identificar ataques de Malware móvil en dispositivos Android mediante algoritmos de aprendizaje automático. Para el entrenamiento y validación del modelo, se utiliza el conjunto de datos de malware para Android conocido como CICMalDroid-2020, publicado por el Instituto Canadiense de Ciberseguridad.

Metodología

La metodología se fundamenta en la estructura KDD (*Knowledge Discovery in Databases*), un enfoque sistemático que facilita el desarrollo de la minería de datos a través de etapas bien definidas (Ghazal & Hammad, 2022). Este modelo permite llevar un control riguroso en cada fase del proceso, asegurando la correcta extracción, transformación y análisis de la información contenida en grandes volúmenes de datos. Gracias a esta estructuración, se optimiza la identificación de patrones, tendencias y relaciones significativas (Llatas *et al.*, 2024). Este proceso facilita la transformación de datos en conocimiento útil para la toma de decisiones y el desarrollo de estrategias basadas en evidencia (Guzmán-Brand & Gélvez-García, 2024).

Etapa Uno: Selección de los datos

En esta etapa se selecciona el conjunto de datos a trabajar para lo que se debe indagar sobre el conjunto que contiene información rigurosa, estructurada y proveniente de una fuente primaria. Es así como se emplea el conjunto de datos sobre malware para Android (CICMalDroid-2020) desarrollado por el Instituto Canadiense de Ciberseguridad. Este contiene un total de 17,341 muestras recopiladas entre 2017 y 2018 de fuentes especializadas como VirusTotal y Contagio (Mahdavifar *et al.*, 2020). A continuación, en la tabla 1 se muestra el sitio de descarga y composición:

Tabla 1.
Descripción de la base de datos sobre malware para Android

Ubicación	Enlace	Repositorio	Tipo de Licencia	Peso	Origen	Formato
Instituto Canadiense de Ciberseguridad	https://www.unb.ca/cic/datasets/maldroid-2020.html	CICMalDroid-2020	Attribution 4.0 International (CC BY 4.0)	11.428 KB	Canadá	.csv

Este conjunto de datos tiene un gran valor gracias a la diversidad de las muestras, clasificadas en cinco categorías: adware, malware bancario, malware SMS, software de riesgo y aplicaciones benignas. Además, se distingue por la inclusión de características estáticas y dinámicas más completas en comparación con otros conjuntos de datos disponibles públicamente.

Adware móvil: se trata de aplicaciones que despliegan anuncios de manera intrusiva y recopilan datos del usuario con el propósito de personalizar y optimizar la publicidad mostrada (Olguin & Arana, 2024). Los creadores de adware implementan múltiples estrategias para evitar ser detectados, entre ellas, la ofuscación de los nombres de los paquetes y la utilización de técnicas de carga dinámica para desplegar sus cargas útiles de manera encubierta (Wang *et al.*, 2024).

Malware bancario: es una amenaza cibernética especializada que busca acceder de manera fraudulenta a las cuentas bancarias en línea de los usuarios. Para ello, imita la interfaz de las aplicaciones bancarias legítimas o las plataformas web oficiales de las entidades financieras (McElroy, 2024). En su mayoría, estos ataques se ejecutan a través de troyanos, programas maliciosos diseñados para infiltrarse en los dispositivos, capturar credenciales sensibles como nombres de usuario y contraseñas; y transmitir esta información a un servidor de comando y control, facilitando así el acceso no autorizado a los fondos de las víctimas (Mahdavifar *et al.*, 2020).

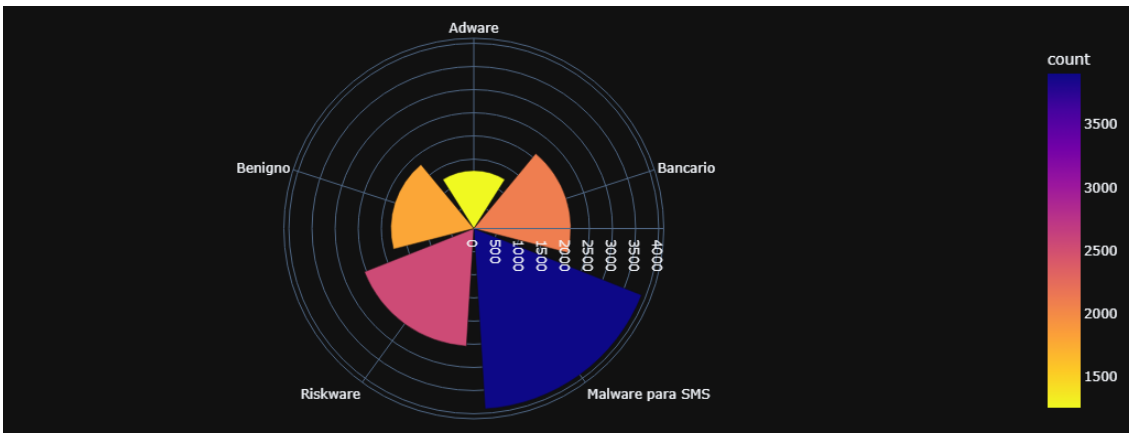
Malware SMS: los ataques de smishing representan una variante del phishing, en la que los ciberdelincuentes utilizan mensajes de texto o aplicaciones de mensajería para engañar a las víctimas (Martínez *et al.*, 2018). A través de enlaces maliciosos o mensajes fraudulentos, buscan inducir a los usuarios a revelar información confidencial, como credenciales bancarias o datos personales, comprometiendo así la seguridad de sus dispositivos (IBM, 2024).

Riskware móvil: el término riskware hace referencia a programas legítimos que, si bien, no fueron concebidos con intenciones maliciosas, pueden representar una amenaza cuando son explotados por ciberdelincuentes (Martínez & Rojas, 2015). Este software puede ser utilizado para eliminar, bloquear, modificar o copiar información, así como para comprometer el rendimiento de sistemas informáticos o redes. Su riesgo radica en la posibilidad de que sus funcionalidades sean manipuladas con fines ilícitos, afectando la seguridad y estabilidad de los entornos digitales (kaspersky, 2017).

Etapa Dos: Preprocesamiento

Durante la fase de preprocesamiento, los datos se someten a un proceso de limpieza y estructuración con el fin de eliminar el ruido y garantizar su calidad antes del análisis. En esta etapa, se separaron las características de las etiquetas y se aplicó una división estratificada en conjuntos de entrenamiento (80%) y prueba (20%), asegurando el mantenimiento de la proporción de clases. Para optimizar el rendimiento del modelo, las características fueron estandarizadas mediante StandardScaler, normalizando los valores a una media de 0 y una desviación estándar de 1, hecho que facilita la convergencia de la red neuronal. Adicionalmente, las etiquetas fueron transformadas al formato one-hot encoding para adaptarlas a la clasificación multiclase.

Figura 1.
Conteo de casos por categoría



Etapa tres: transformación

El proceso de transformación de datos busca reducir la dimensionalidad o modificar la estructura de la información para mejorar su adecuación en tareas de minería de datos. En este caso, se implementó una selección de características mediante *SelectKBest*, utilizando la función de puntuación ANOVA, hecho que permitió reducir el conjunto de características a 150. Esta optimización mejora la eficiencia computacional y también elimina variables irrelevantes o redundantes, asegurando que el modelo se enfoque en los atributos más significativos.

Resultados

Etapa cuatro: minería de datos

Para el entorno de experimentación se escoge la plataforma Google Colaboratory empleando la GPU T4 disponible, lenguaje de programación Python 3.11.11. En cuanto a las librerías para la analítica de datos se utlizaron, por un lado, pandas, numpy, seaborn, matplotlib, sklearn versión 1.6.1, tensorflow 2.18.0, keras 3.8.0. Y por otro lado, también se utilizaron los siguientes algoritmos para observar su comportamiento respecto a los datos:

Red neuronal: una red neuronal artificial está compuesta por un conjunto de unidades interconectadas, denominadas neuronas, que procesan información de manera distribuida. Cada neurona recibe señales de entrada a través de conexiones con otras neuronas, aplica una función matemática específica y genera un valor de salida. Este valor puede, a su vez, servir como entrada para otras neuronas dentro de la red, permitiendo la propagación y transformación de la información a lo largo de las distintas capas del modelo (Gironés *et al.*, 2017).

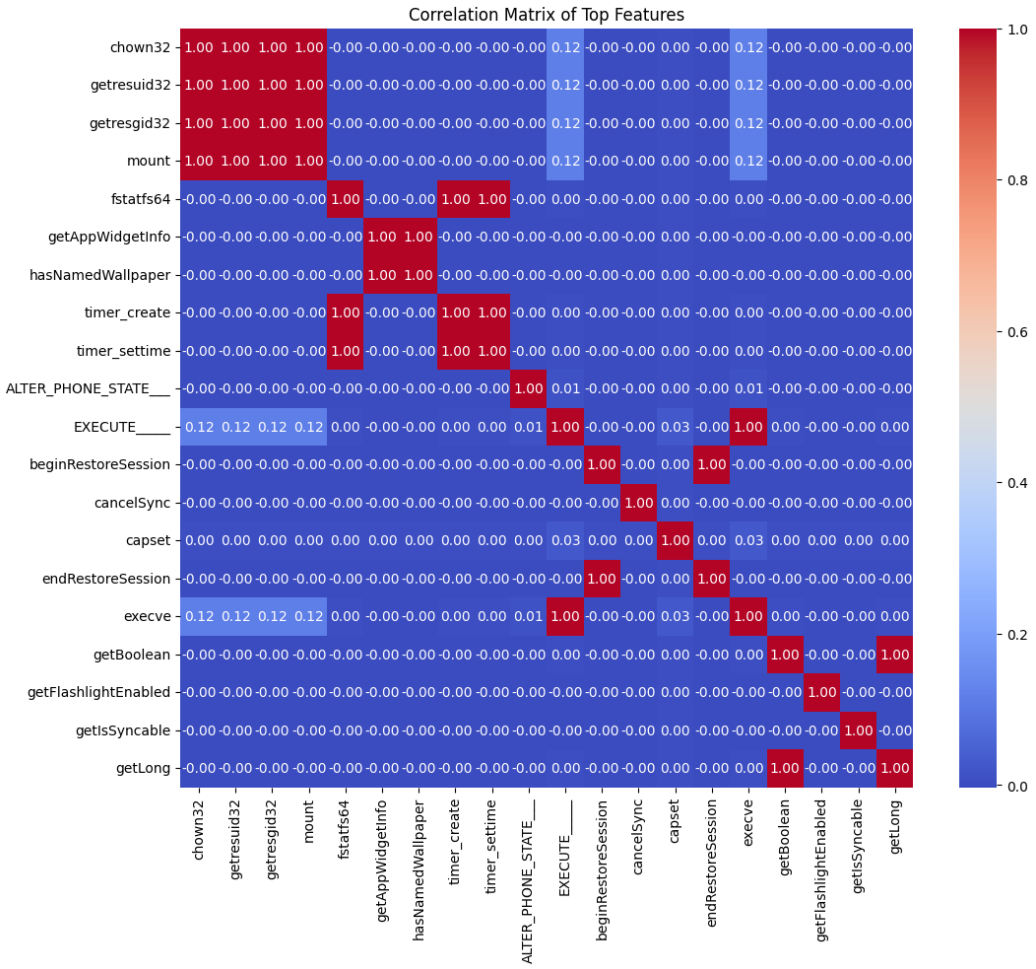
XGBoost: este método se basa en la construcción secuencial de un conjunto de árboles de decisión, conocidos como CART (Classification and Regression Trees). En cada iteración, se incorpora un nuevo árbol con el propósito de mejorar el desempeño del modelo al aprender de los errores cometidos por los árboles anteriores. Este proceso de ajuste continuo permite minimizar progresivamente el margen de error hasta alcanzar un punto en el que las correcciones adicionales resultan insignificantes (Espinosa-Zúñiga, 2020).

CatBoost: es un algoritmo de aprendizaje automático que pertenece a la familia de modelos Boosting, su funcionamiento se basa en el aumento del gradiente y está diseñado específicamente para manejar variables categóricas de manera eficiente. Para lograrlo, utiliza un enfoque basado en permutaciones que permite una asignación imparcial en la construcción de divisiones, dentro de los árboles de decisión CART. Además, en cada iteración, selecciona los valores de las hojas a partir de diferentes subconjuntos de datos, optimizando así su capacidad predictiva y reduciendo el riesgo de sobreajuste (Pincay-Ponce *et al.*, 2024).

LightGBM: su principio fundamental radica en la combinación de múltiples modelos de aprendizaje débil para construir un modelo predictivo más sólido y preciso. La lógica detrás de los algoritmos de potenciación consiste en ajustar dinámicamente la ponderación de los datos: se incrementa el peso de aquellos ejemplos que han sido clasificados erróneamente, mientras que se reduce el de aquellos correctamente identificados. De esta manera, el modelo dirige su atención hacia los casos más desafiantes en las siguientes iteraciones del entrenamiento, optimizando su capacidad de generalización (Li *et al.*, 2024).

Random Forest: los árboles de decisión constituyen un método de clasificación basado en la segmentación de un conjunto de datos de prueba, según sus características distintivas. Su estructura se organiza mediante un proceso de particionamiento iterativo, donde un criterio de división determina la separación de los datos en múltiples subconjuntos. Este procedimiento se desarrolla de manera jerárquica, siguiendo un enfoque descendente que inicia con la muestra de entrenamiento y se refina progresivamente hasta alcanzar una clasificación óptima (Jones, 2019).

Figura 2.
Exposición de las variables con mayor correlación



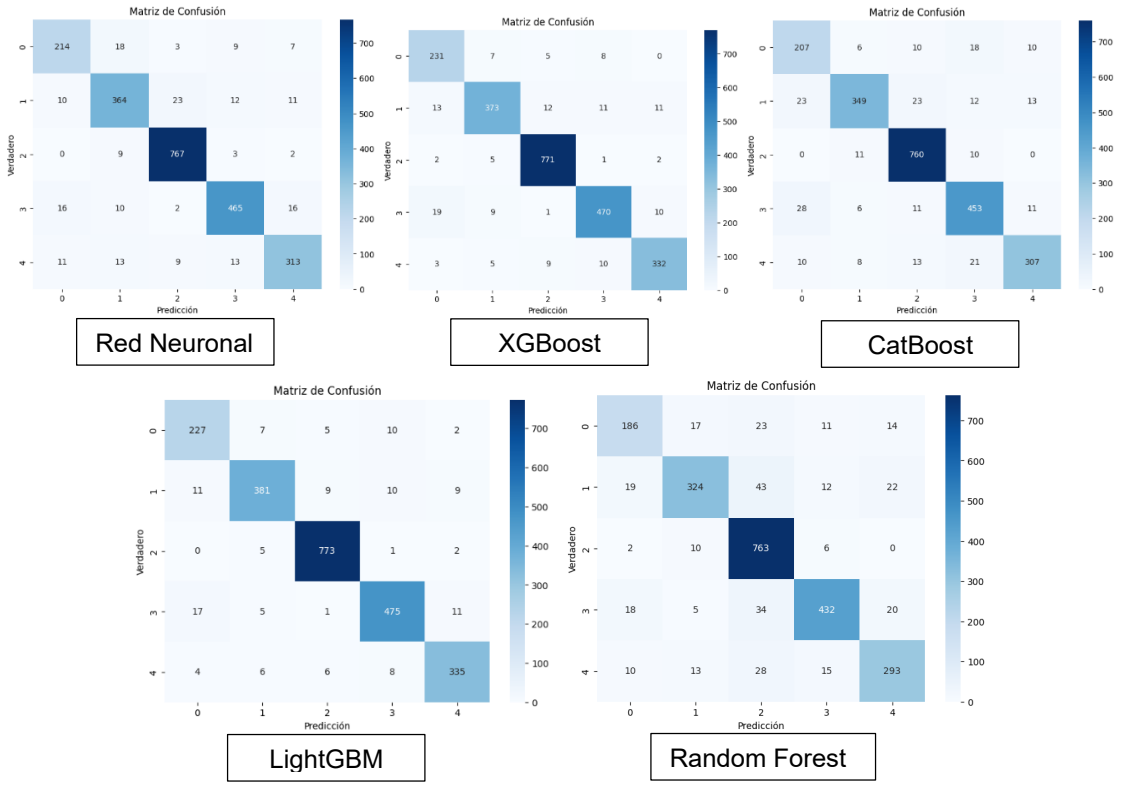
Nota: Una matriz de correlación es una tabla que muestra los coeficientes de correlación entre variables, permitiendo analizar sus relaciones. Se utiliza para sintetizar datos, apoyar estudios avanzados y sirve como herramienta diagnóstica en análisis complejos (León *et al.*, 2008). Elaboración propia.

Las variables con mayor correlación identificadas en la matriz de correlación corresponden a funciones y comportamientos que son típicos de ataques de malware móvil. Estas incluyen la manipulación de permisos y recursos del sistema (chown32, getresuid32, mount), la exploración del sistema de archivos (fstatfs64), el control del estado del dispositivo (ALTER_PHONE_STATE, getFlashlightEnabled), la ejecución de comandos programados (timer_create, EXECUTE_), la persistencia del malware (beginRestoreSession, endRestoreSession) y la interacción con la interfaz del usuario (hasNamedWallpaper, getAppWidgetInfo). Estas actividades son esenciales para el funcionamiento de malware, dado que permiten a los atacantes obtener acceso no autorizado, esconder sus componentes y evitar la detección.

A. Matriz de confusión

La matriz de confusión es una herramienta en la evaluación del desempeño de modelos de clasificación que proporciona una representación clara y estructurada de sus aciertos y errores. Su principal utilidad radica en la medición de la precisión y exactitud del modelo, permitiendo analizar su capacidad para diferenciar entre distintas categorías. Esta matriz se organiza en una tabla de doble entrada con dos dimensiones: valores reales (actuales) y valores predichos. En ella, las filas corresponden a las clases verdaderas observadas en los datos, mientras que las columnas representan las categorías asignadas por el modelo (González, 2019).

Figura 3.
Medición mediante la matriz de confusión



La matriz de confusión es una herramienta en la evaluación del desempeño de modelos de clasificación que proporciona una representación clara y estructurada de sus aciertos y errores. Su principal utilidad radica en la medición de la precisión y exactitud del modelo, permitiendo analizar su capacidad para diferenciar entre distintas categorías. Esta matriz se organiza en una tabla de doble entrada con dos dimensiones: valores reales (actuales) y valores predichos. En ella, las filas corresponden a las clases verdaderas observadas en los datos, mientras que las columnas representan las categorías asignadas por el modelo (González, 2019).

B. Métricas generales de evaluación

A partir de la matriz de confusión, se pueden derivar diversas métricas esenciales para evaluar el rendimiento de un modelo de clasificación. Entre las más relevantes se encuentran:

Tabla 2.
Resultados de evaluación de los algoritmos

Modelo	Accuracy	Precisión	Recall	F1-Score	AUC-ROC	Ratio de error	Tiempo Ejecución
RN + SE	0.9151	0.9146	0.9151	0.9147	0.9878	0.0849	93.21
XGBoost	0.9384	0.9387	0.9384	0.9383	0.9949	0.0616	6.80
CatBoost	0.8948	0.8957	0.8948	0.8945	0.9868	0.1052	7.84
LightGBM	0.9444	0.9445	0.9444	0.9443	0.9951	0.0556	5.24
Random Forest	0.8612	0.8619	0.8612	0.8612	0.9727	0.1388	0.78

En la Tabla 2 se muestran los resultados de evaluación de cinco modelos de aprendizaje automático, destacando que LightGBM obtiene el mejor desempeño general con los valores más altos en Accuracy (0.9444), Precisión (0.9445), Recall (0.9444), F1-Score (0.9443) y AUC-ROC (0.9951), además de un bajo ratio de error (0.0556) y el menor tiempo de ejecución (5.24 segundos). Le sigue XGBoost que también presenta métricas sólidas, pero requiere más tiempo de cómputo (6.80 segundos). Random Forest, aunque es el más rápido (0.78 segundos), tiene las métricas más bajas en términos de precisión y capacidad de clasificación. CatBoost y la Red Neuronal muestran un rendimiento intermedio.

Discusión

Etapa Seis: Interpretación

El aprendizaje automático emerge como un elemento clave para enfrentar el crecimiento de las ciber amenazas, transformando las estrategias de seguridad mediante la combinación adecuada de múltiples técnicas (K. Liu *et al.*, 2020). Su aplicación en áreas como la detección de malware, ataques DDoS, vulnerabilidades de software, intrusiones y análisis de comportamiento en

redes sociales demuestra su capacidad para mejorar la precisión y eficacia de los sistemas de ciberseguridad (Guzman-Brand & Gelvez-Garcia, 2025). Sin embargo, es fundamental reconocer y abordar sus limitaciones y desafíos para garantizar una implementación efectiva. Los resultados actuales sugieren que estas herramientas pueden potenciar significativamente la capacidad de respuesta ante amenazas cibernéticas (Gainza *et al.*, 2023).

Los resultados de esta investigación exponen que al evaluar la matriz de confusión se observa cómo el algoritmo LightGBM tiene una capacidad para manejar grandes volúmenes de datos y características de manera eficiente, hecho que da como resultado una clasificación precisa. Además, este modelo se destaca en las métricas generales de evaluación con el mejor desempeño general. De igual manera, el estudio de Villarroel & Gutiérrez-Cárdenas (2024) experimenta con algoritmos XGBoost, Random Forest y LightGBM, mostrando que este último alcanza mejor desempeño alcanzando la precisión de 94.1%.

Por otro lado, en el segundo lugar, se encuentra XGBoost, un modelo que exhibe métricas de desempeño destacables, aunque su tiempo de cómputo es mayor, alcanzando los 6.80 segundos. Por otro lado, Random Forest se destaca por su rapidez, con un tiempo de procesamiento de 0.78 segundos. Al respecto, la investigación de (Ahmed *et al.*, 2023) evidencia cómo el algoritmo XGBoost ha alcanzado una precisión del 99.5% en la identificación de amenazas. Además, el algoritmo Random Forest ha sido empleado específicamente en la clasificación de malware bancario en estos dispositivos, logrando una precisión del 92.5% (Kumari & Sharma, 2023).

Las técnicas de protección contra malware basadas en inteligencia artificial (IA) han transformado significativamente la seguridad en dispositivos móviles, destacando enfoques como el aprendizaje automático y profundo (Quirumbay *et al.*, 2022). No obstante, su aplicación enfrenta desafíos críticos, particularmente, en cuanto al equilibrio entre el consumo de recursos y la efectividad, especialmente en dispositivos con capacidades limitadas. En el futuro, será esencial optimizar el uso de recursos para garantizar eficiencia energética y adaptabilidad frente a amenazas en constante evolución (Sacramento *et al.*, 2024).

Conclusiones

El campo de la ciberseguridad ha experimentado una transformación significativa gracias a los avances en aprendizaje automático, que han proporcionado herramientas sofisticadas para contrarrestar el incremento exponencial de amenazas digitales. Estas tecnologías han demostrado ser particularmente efectivas en la identificación y mitigación de riesgos como malware, intrusiones malintencionadas y vulnerabilidades estructurales en sistemas informáticos. Como resultado, se ha fortalecido la capacidad de protección de redes y dispositivos frente a incidentes cibernéticos.

La presente investigación logró identificar ataques de malware móvil en dispositivos Android mediante algoritmos de aprendizaje automático, utilizando el conjunto de datos CICMalDroid-2020. Entre los modelos evaluados, LightGBM demostró ser el más eficiente y preciso, destacándose por su capacidad para manejar grandes volúmenes de datos y características de manera óptima.

Este algoritmo mostró un desempeño superior en métricas clave como Accuracy, Precisión, Recall, F1-Score y AUC-ROC, además de minimizar errores y reducir tiempos de ejecución significativamente en comparación con otros modelos.

Además, se observó que LightGBM supera consistentemente a otros modelos en escenarios donde los datos están desbalanceados, una característica común en conjuntos de datos de malware. Su enfoque basado en hojas y las optimizaciones específicas para reducir falsos positivos y falsos negativos lo convierten en una herramienta valiosa para la ciberseguridad móvil. Este resultado sugiere que puede ser particularmente adecuado para aplicaciones en tiempo real, donde la rapidez y la precisión son requisitos indispensables.

En este marco, las estrategias de seguridad impulsadas por inteligencia artificial (IA) han emergido como un componente esencial, especialmente en el ámbito de la protección de dispositivos móviles. Técnicas avanzadas como el aprendizaje automático supervisado y el aprendizaje profundo han permitido mejorar sustancialmente la capacidad de detección de patrones asociados a actividades maliciosas. Sin embargo, uno de los principales obstáculos radica en la necesidad de equilibrar eficiencia analítica y el consumo de recursos, un factor particularmente relevante en dispositivos móviles caracterizados por hardware limitado. En consecuencia, el desarrollo de modelos más eficientes, adaptativos y escalables se presenta como una línea de investigación crucial para el futuro.

Contribuciones de los autores

Fuentes de financiamiento

Esta investigación no ha recibido apoyo financiero por parte de entidades gubernamentales, comerciales o sin fines de lucro.

Conflictos de interés

Los autores declaramos no tener conflictos de interés de carácter financiero, profesional o personal que pudieran influir de manera indebida en los resultados obtenidos, o en su interpretación.

Contribuciones de los autores

Autor 1: responsable del análisis de datos, implementación del software, preprocesamiento de la información, desarrollo metodológico, validación, visualización de resultados e investigación. Además, participó en la redacción del borrador original.

Autor 2: Encargado del análisis formal, desarrollo metodológico, administración del proyecto y gestión de recursos. También supervisó el trabajo, revisó y editó el manuscrito.

Referencias

Ahmed, A., Saeed, M., Hamood, A., Alazab, A., & Ahmed, K. (2023). Comparative Study of Static Analysis and *Machine learning* Approaches for Detecting Android Banking Malware. *2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA), 3*, 01-08. https://doi.org/10.1109/eSmarTA59349.2023.10293602

Alkahtani, H., & Aldhyani, T. (2022). Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices. *Sensors, 22*(6), Article 6. https://doi.org/10.3390/s22062268

Álvarez, M., & Montoya, H. (2020). Ciberseguridad en las redes móviles de telecomunicaciones y su gestión de riesgos. *Ingeniería y Desarrollo, 38*(2), 279-297. https://doi.org/10.14482/inde.38.2.006.31

Bashir, S., Maqbool, F., Khan, F., & Abid, A. (2024). Hybrid *machine learning* model for malware analysis in android apps. *Pervasive and Mobile Computing, 97*(97), 101-121. https://doi.org/10.1016/j.pmcj.2023.101859

Cassinda, F. (2019). Caracterização de sistemas operacionais móveis celulares: Android, Symbian, iphone e Windows phone. *Project Design and Management, 1*(2), Article 2. https://doi.org/10.35992/pdm.v1i2.200

Espinosa-Zúñiga, J. (2020). Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito. *Ingeniería Investigación y Tecnología, 21*(3), 1-16. https://doi.org/10.22201/fi.25940732e.2020.21.3.022

Gainza, D., Reyes, D., Brito, H., Véliz, Y., & Pérez, Y. (2023). Técnicas de Aprendizaje Automático para la detección y prevención de amenazas de ciberseguridad. Proyecciones futuras. *Revista Cubana de Ciencias Informáticas, 10*(10), 5. https://rcci.uci.cu/?journal=rcci&page=article&op=view&path[]=2823

Ghazal, M., & Hammad, A. (2022). Application of knowledge discovery in database (KDD) techniques in cost overrun of construction projects. *International Journal of Construction Management, 22(*9), 1632-1646. https://doi.org/10.1080/15623599.2020.1738205

Gironés, J., Casas, J., Minguillón, J., & Caihuelas, R. (2017). *Minería de datos Modelos y algoritmos*. Editorial UOC (Oberta UOC Publishing, SL).

González, L. (2019). *Machine learning* con Python Aprendizaje Supervisado. Independiente.

Guzmán-Brand, V., & Gélvez-García, L. (2024). Identificación de patrones a través de algoritmos de *machine learning* en los casos registrados de intentos suicidas en una ciudad de Colombia. *Psicoespacios, 18*(32), 50-65. https://doi.org/10.25057/21452776.1634

Guzman-Brand, V., & Gelvez-Garcia, L. (2025). Identificación de ataques de denegación de servicio distribuido (DDoS) mediante la integración de algoritmos de aprendizaje automático y arquitecturas de redes neuronales artificiales. *Revista Ingeniería, Matemáticas y Ciencias de la Información, 12*(23), Article 23. https://doi.org/10.21017/rimci.1116

IBM. (2024, junio 10). *¿Qué es el smishing (phishing por SMS)? | IBM*. https://www.ibm.com/es-es/topics/smishing

Iqubal, A., & Payal, A. (2024). Malware Detection Technique for Android Devices Using *Machine learning* Algorithms. *2024 International Conference on Computing, Sciences and Communications (ICCSC), 2*, 1-6. https://doi.org/10.1109/ICCSC62048.2024.10830310

Jones, H. (2019). *Minería de Datos Guía de Minería de Datos para Principiantes, que Incluye Aplicaciones para Negocios, Técnicas de Minería de Datos, Conceptos y Más*. Editorial Privada.

kaspersky. (2017, noviembre 9). *¿Qué es el riskware? | Amenazas de seguridad en Internet.* https://latam.kaspersky.com/resource-center/threats/riskware

Kaspersky. (2024, febrero 26). *El informe anual de amenazas móviles de Kaspersky destaca la creciente prevalencia de los riesgos de seguridad móvil junto con el avance de herramientas y tecnologías maliciosas*. latam.kaspersky. https://latam.kaspersky.com/about/press-releases/los-ataques-a-dispositivos-moviles-aumentaron-mas-del-50-en-2023

Kumari, A., & Sharma, I. (2023). SafeDroid: Safeguarding Android Mobile Phones from Adware and Banking Maldroid Attacks. *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, 2, 98-103. https://doi.org/10.1109/ICSCNA58489.2023.10370154

León, Á., Llinás, H., & Tilano, J. (2008). Análisis multivariado aplicando componentes principales al caso de los desplazados. *Revista Ingenieria y Desarrollo, 23*(23), 1-20. https://rcientificas.uninorte.edu.co/index.php/ingenieria/article/download/2098/4467?inline=1

Li, S., Jin, N., Dogani, A., Yang, Y., Zhang, M., & Gu, X. (2024). Enhancing LightGBM for Industrial Fault Warning: An Innovative Hybrid Algorithm. *Processes, 12*(1), Article 1. https://doi.org/10.3390/pr12010221

Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., & Liu, H. (2020). A Review of Android Malware Detection Approaches Based on *Machine learning*. *IEEE Access, 8*(8), 124579-124607. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3006143

Liu, Z., Wang, R., Japkowicz, N., Tang, D., Zhang, W., & Zhao, J. (2021). Research on unsupervised feature learning for Android malware detection based on Restricted Boltzmann Machines. *Future Generation Computer Systems, 120*, 91-108. https://doi.org/10.1016/j.future.2021.02.015

Llatas, C., Soust-Verdaguer, B., Castro, L., & Cagigas, D. (2024). Application of *Knowledge Discovery in Databases* (KDD) to environmental, economic, and social indicators used in BIM workflow to support sustainable design. *Journal of Building Engineering, 91*(45), 109546. https://doi.org/10.1016/j.jobe.2024.109546

Mahdavifar, S., Abdul, A. F., Fatemi, R., Alhadid, D., & Ghorbani, A. (2020). *MalDroid 2020 | Conjuntos de datos | Investigación | Instituto Canadiense de Ciberseguridad | UNB* (Versión Primera) [Dataset]. 18.a Conferencia internacional del IEEE sobre computación confiable, autónoma y segura (DASC). https://www.unb.ca/cic/datasets/maldroid-2020.html

Martínez, J., Gavilanes, Y., Gavilanes, T., & Lozano, M. (2018). Seguridad por capas frenar ataques de Smishing. *Dominio de las Ciencias, 4*(1), Article 1. https://doi.org/10.23857/dom.cien.pocaip.2017.4.1.enero.115-130

Martínez, J., & Rojas, L. (2015). Vulnerabilidad en dispositivos móviles con sistema operativo Android. *Cuaderno activa, 7*(7), 55-65. https://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/248

McElroy, S. (2024). Identifying Android Banking Malware Through Measurement of User Interface Complexity. *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2, 348-353. https://doi.org/10.1109/CSR61664.2024.10679403

Milosevic, N., Dehghantanha, A., & Choo, K.-K. (2017). *Machine learning* aided Android malware classification. *Computers & Electrical Engineering, 61*(61), 266-274. https://doi.org/10.1016/j.compeleceng.2017.02.013

Mohammed, A., & Awad, A. I. (2022). AdStop: Efficient flow-based mobile adware detection using *machine learning*. *Computers & Security, 117*(23), 102718. https://doi.org/10.1016/j.cose.2022.102718

Olguin, A., & Arana, J. (2024). Ataques a celulares a través del uso de aplicaciones móviles: Una revisión narrativa. *TECNOCIENCIA Chihuahua, 18*(3), Article 3. https://doi.org/10.54167/tch.v18i3.1584

Pincay-Ponce, J., De Giusti, A., Sánchez-Andrade, D., & Figueroa-Suárez, J. (2024). CatBoost: Aprendizaje automático de conjunto para la analítica de los factores socioeconómicos que inciden en el rendimiento escolar. *TE & ET, 38*(38), 1-20. https://teyet-revista.info.unlp.edu.ar/TEyET/article/view/2492

Quirumbay, D., Castillo, C., & Coronel, I. (2022). Una revisión del aprendizaje profundo aplicado a la ciberseguridad. *Revista Científica y Tecnológica UPSE (RCTU), 9*(1), 57-65. https://doi.org/10.26423/rctu.v9i1.671

Sacramento, L., Salcedo, G., & Mendoza, A. (2024). Técnicas de protección contra malware impulsadas por IA en entorno móviles | Campus. *Revista científica tecnológica Campus, 29*(38), 1-20. https://doi.org/10.24265/campus.2024.v29n38.04

Villarroel, E., & Gutiérrez-Cárdenas, J. (2024). Dynamic Malware Analysis Using *Machine learning*-Based Detection Algorithms. *Interfases, 19*(19), Article 019. https://doi.org/10.26439/interfases2024.n19.7097

Wang, C., Liu, T., Zhao, Y., Zhang, L., Du, X., Li, L., & Wang, H. (2024). Towards Demystifying Android Adware: Dataset and Payload Location. *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, 2, 167-175. https://doi.org/10.1145/3691621.3694948