Detección de Malware en Dispositivos IoT mediante modelos ligeros de Machine Learning

José Nayid Cardona Castañeda

Directora

Yenny Stella Nunez Alvarez

Universidad Nacional Abierta y a Distancia – UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Maestría en Ciberseguridad

2026

Dedicatoria

Dedico este trabajo con profundo amor y gratitud a mi esposa, por su compañía, paciencia y constante apoyo en cada etapa de este proceso académico. A mis hijos, quienes con su alegría y cariño me inspiran día a día a superarme y a mantener vivo el compromiso con el conocimiento y la perseverancia. A ellos les ofrezco este logro, fruto del esfuerzo compartido y de la ilusión de construir juntos un futuro mejor.

Agradecimientos

Expreso mi más sincero agradecimiento a la Universidad Nacional Abierta y a Distancia – UNAD, por ofrecer el espacio académico y las herramientas que hicieron posible este proceso de formación. A mis docentes y directores, quienes con su orientación y dedicación contribuyeron al fortalecimiento de mis competencias investigativas y profesionales.

Reconozco también a mis compañeros de la maestría, por los aprendizajes compartidos, el apoyo constante y las reflexiones conjuntas que enriquecieron cada etapa de este camino.

De manera especial, agradezco a mi esposa y a mis hijos, quienes con su paciencia, amor y respaldo incondicional me motivaron a perseverar y alcanzar esta meta. A ellos dedico este logro como fruto de un esfuerzo compartido.

Finalmente, expreso gratitud a todas las personas e instituciones que, de manera directa o indirecta, aportaron tiempo, apoyo e ideas para la realización de este trabajo.

Resumen

Esta monografía analiza y compara técnicas de aprendizaje automático ligero aplicadas a la detección de malware en entornos IoT y sistemas embebidos, con énfasis en las restricciones de memoria, procesamiento, latencia, conectividad, privacidad y consumo energético propias de estos dispositivos. El trabajo se desarrolló mediante una revisión sistemática de literatura con enfoque analítico-comparativo, orientada a identificar tendencias, limitaciones, métricas de evaluación, datasets de referencia y criterios técnicos para la selección de modelos ligeros. Los hallazgos evidencian que los enfoques tradicionales de detección pueden presentar dificultades de despliegue en dispositivos con recursos limitados, mientras que técnicas como selección de características, poda, cuantización, TinyML y aprendizaje federado ofrecen alternativas pertinentes cuando se evalúan desde una perspectiva multicriterio. Asimismo, se identificó que la detección ligera puede apoyarse en patrones observables de bajo costo computacional, asociados al tráfico de red y al comportamiento del dispositivo, como duración del flujo, volumen de paquetes, solicitudes DNS, uso de CPU, memoria y procesos activos. Como aporte principal, se propone un marco de aplicación que articula contexto operativo, tipo de amenaza, características observables, técnica ligera, validación multicriterio y decisión de adopción. Se concluye que la selección de técnicas de detección en IoT no debe depender únicamente del desempeño predictivo, sino también de su viabilidad operativa en condiciones reales de despliegue.

Palabras clave: aprendizaje automático ligero, ciberseguridad, detección de malware, IoT, sistemas embebidos.

Abstract

This monograph analyzes and compares lightweight machine learning techniques applied to malware detection in IoT environments and embedded systems, with emphasis on memory, processing, latency, connectivity, privacy, and energy consumption constraints. The study follows a monographic design with an analytical-comparative approach, based on a systematic literature review aimed at identifying trends, limitations, evaluation metrics, reference datasets, and technical criteria for selecting lightweight detection techniques. The findings show that traditional detection approaches may face deployment limitations in resource-constrained devices, whereas techniques such as feature selection, pruning, quantization, TinyML, and federated learning offer relevant alternatives when assessed from a multicriteria perspective. The review also identified that lightweight detection can rely on low-cost observable patterns associated with network traffic and device behavior, such as flow duration, packet volume, DNS requests, CPU usage, memory consumption, and active processes. As the main contribution, this work proposes an application framework that integrates operational context, threat type, observable features, lightweight technique selection, multicriteria validation, and adoption decision. The study concludes that the selection of malware detection techniques in IoT environments should not depend solely on predictive performance, but also on operational feasibility under real deployment conditions.

Keywords**:** cybersecurity, IoT, lightweight machine learning, embedded systems, malware detection.

Contenido

Lista de Tablas

Lista de Figuras

Lista de Apéndices

Glosario

Accuracy: métrica de evaluación que indica la proporción de predicciones correctas realizadas por un modelo respecto al total de casos evaluados. En esta investigación se considera una métrica útil, aunque insuficiente por sí sola cuando existen clases desbalanceadas entre tráfico benigno y malicioso.

Aprendizaje automático: campo de la inteligencia artificial orientado al desarrollo de modelos capaces de identificar patrones a partir de datos. En el contexto del estudio, se emplea como base para analizar técnicas de detección de malware en dispositivos IoT y sistemas embebidos.

Aprendizaje automático ligero: enfoque orientado a reducir la complejidad computacional de los modelos sin afectar de manera sustancial su capacidad predictiva. En esta investigación se relaciona con técnicas como selección de características, poda estructural, cuantización, TinyML y aprendizaje federado, debido a su pertinencia para dispositivos con recursos limitados.

Aprendizaje federado: paradigma de entrenamiento distribuido que permite actualizar modelos a partir de datos alojados en múltiples dispositivos sin necesidad de centralizarlos. En este estudio se considera una alternativa relevante para entornos IoT con restricciones de conectividad y requisitos de privacidad, al reducir la transferencia de datos sensibles y limitar la sobrecarga de comunicación entre nodos (Nguyen et al., 2021; Rey et al., 2021).

Backdoor: mecanismo oculto o no autorizado que permite acceder a un sistema evadiendo los controles normales de autenticación. En dispositivos IoT representa un riesgo relevante por su capacidad de persistencia y control remoto.

Botnet: red de dispositivos comprometidos controlados de manera remota por un atacante. En entornos IoT, las botnets son especialmente críticas porque pueden aprovechar credenciales débiles, servicios expuestos y dispositivos con baja capacidad de protección.

Ciberseguridad: conjunto de prácticas, procesos, tecnologías y controles orientados a proteger sistemas, redes, dispositivos y datos frente a accesos no autorizados, alteraciones, interrupciones o daños. En esta investigación se aborda desde la necesidad de detectar malware en entornos IoT con restricciones computacionales.

Consumo de memoria: cantidad de memoria requerida por un modelo o proceso durante su ejecución. En dispositivos IoT es una métrica crítica porque muchos sistemas embebidos cuentan con recursos limitados para almacenar y ejecutar modelos de detección.

Consumo energético: cantidad de energía requerida para ejecutar un modelo o proceso. En entornos IoT es relevante porque muchos dispositivos operan con baterías o fuentes de energía limitadas.

Cryptojacking: uso no autorizado de los recursos computacionales de un dispositivo para realizar minería de criptomonedas. En dispositivos IoT puede reflejarse en incrementos anómalos de CPU, memoria o consumo energético.

Cuantización: técnica de compresión de modelos que reduce la precisión numérica de los parámetros, por ejemplo, al pasar de representaciones en punto flotante a formatos enteros de menor resolución. En este estudio se analiza como mecanismo para disminuir el consumo de memoria y acelerar la inferencia en dispositivos IoT.

Dataset: conjunto estructurado de datos utilizado para entrenar, validar o comparar modelos de aprendizaje automático. En esta investigación los datasets reportados en la literatura, como IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022, sirven como referentes para analizar técnicas de detección de malware.

Dispositivo de bajo recurso: dispositivo con limitaciones de procesamiento, memoria, almacenamiento, conectividad o energía. En esta investigación representa el escenario técnico donde los modelos tradicionales pueden resultar poco viables.

Dispositivo IoT: objeto físico interconectado capaz de recopilar, procesar o transmitir datos mediante redes de comunicación. Incluye sensores, cámaras, medidores inteligentes, gateways, microcontroladores y sistemas embebidos.

Edge computing: paradigma de procesamiento distribuido que acerca el cómputo al lugar donde se generan los datos. En entornos IoT permite reducir latencia, disminuir transferencia de información y favorecer respuestas más rápidas frente a eventos de seguridad.

F1-score: métrica que combina precision y recall mediante una media armónica. En detección de malware resulta útil cuando existe desbalance entre clases benignas y maliciosas, ya que ofrece una medida más equilibrada del rendimiento del modelo.

Feature o característica observable: variable medible que representa un aspecto del comportamiento del sistema o del tráfico de red. En esta investigación incluye variables como duración del flujo, tamaño de paquetes, uso de CPU, consumo de memoria o número de procesos activos.

Inferencia en el borde: ejecución de un modelo cerca del lugar donde se generan los datos, sin depender completamente de la nube. En IoT permite reducir latencia, proteger datos sensibles y mejorar la continuidad operativa.

Internet de las Cosas (IoT): ecosistema de dispositivos físicos interconectados que recopilan, procesan y transmiten datos mediante redes de comunicación. En esta investigación constituye el contexto de aplicación de modelos ligeros para detección de malware.

Latencia: tiempo que tarda un sistema o modelo en producir una respuesta. En detección de malware para IoT es una métrica clave porque muchos escenarios requieren respuestas cercanas al tiempo real.

Malware: software malicioso diseñado para comprometer la confidencialidad, integridad o disponibilidad de un sistema. En entornos IoT puede manifestarse mediante tráfico anómalo, consumo inusual de recursos, procesos sospechosos o comunicaciones no autorizadas.

Malware en memoria: amenaza que opera principalmente en la memoria del sistema, reduciendo rastros persistentes en archivos. Es relevante para estudios que analizan comportamiento malicioso mediante datasets especializados como CIC-MalMem-2022.

Modelo baseline: modelo de referencia utilizado para comparar el desempeño de nuevas técnicas o enfoques. En esta investigación sirve como punto de contraste frente a modelos ligeros u optimizados.

Modelo ligero: arquitectura de aprendizaje automático diseñada o adaptada para operar bajo restricciones de memoria, procesamiento y energía. Más allá de su tamaño reducido, un modelo ligero busca equilibrar desempeño predictivo y eficiencia computacional.

Modelo tradicional: enfoque de detección que no ha sido optimizado específicamente para operar bajo restricciones de hardware. En el estudio se utiliza como contraste frente a modelos ligeros.

PICOC: estrategia utilizada para delimitar revisiones sistemáticas mediante cinco elementos: población, intervención, comparación, resultados y contexto. En esta investigación apoya la formulación de preguntas de revisión y criterios de búsqueda.

Poda estructural: técnica de optimización que elimina pesos, filtros, conexiones o componentes de baja contribución dentro de un modelo. Su propósito es reducir tamaño, memoria y latencia, conservando un desempeño aceptable.

Precision: métrica que indica la proporción de predicciones positivas correctas respecto al total de predicciones positivas realizadas por el modelo. En detección de malware permite estimar qué tan confiables son las alertas generadas.

PRISMA: lineamiento utilizado para reportar revisiones sistemáticas de manera transparente, especialmente en las fases de identificación, cribado, elegibilidad e inclusión de estudios (Page et al., 2021).

Recall: métrica que expresa la capacidad del modelo para identificar correctamente los casos positivos reales. En detección de malware es relevante porque permite evaluar cuántas amenazas fueron detectadas.

Revisión sistemática de literatura (RSL): método de investigación que permite identificar, seleccionar, evaluar y sintetizar estudios previos mediante criterios explícitos y trazables. En este trabajo constituye la base metodológica para analizar modelos ligeros de detección de malware.

Selección de características: proceso mediante el cual se identifican las variables más relevantes para un modelo, eliminando aquellas que aportan poco valor o aumentan innecesariamente la complejidad. En IoT permite reducir dimensionalidad y mejorar eficiencia computacional.

Sistema embebido: dispositivo computacional integrado dentro de un sistema mayor y diseñado para ejecutar funciones específicas. Su relevancia radica en que suele operar con recursos limitados, lo que condiciona la implementación de modelos de detección.

Superficie de ataque: conjunto de puntos, servicios, interfaces, configuraciones o vulnerabilidades que pueden ser explotados por un atacante. En IoT aumenta debido a la diversidad de dispositivos, protocolos y niveles de protección.

Tamaño del modelo: espacio que ocupa un modelo una vez entrenado o preparado para despliegue. Su reducción favorece la implementación en sistemas embebidos y dispositivos IoT con almacenamiento limitado.

TinyML: implementación de modelos de aprendizaje automático en dispositivos de muy bajo consumo, como microcontroladores. En esta investigación se relaciona con la posibilidad de realizar inferencia local en dispositivos IoT.

Tráfico de red: flujo de datos transmitido entre dispositivos o sistemas conectados. Su análisis permite identificar patrones anómalos asociados a malware, botnets o comunicaciones de comando y control.

Vector de ataque: ruta, técnica o mecanismo mediante el cual una amenaza compromete un sistema. En esta investigación permite relacionar tipos de malware con patrones observables que pueden ser usados por modelos de detección.

Introducción

El crecimiento del Internet de las Cosas (IoT) ha transformado de manera significativa la forma en que las personas, organizaciones e instituciones gestionan sus procesos. Desde sistemas de salud digitalizados hasta infraestructuras críticas interconectadas, la presencia de dispositivos inteligentes se ha consolidado como un eje fundamental de la transformación digital global. No obstante, este avance también ha ampliado la superficie de ataque, generando nuevas amenazas de seguridad que afectan especialmente a los dispositivos con recursos computacionales limitados, los cuales suelen carecer de medidas robustas de protección frente a ataques de malware.

En este escenario, la ciberseguridad adquiere un papel protagónico al buscar soluciones innovadoras que equilibren eficiencia, sostenibilidad y protección de datos. Los enfoques tradicionales de detección de malware han demostrado ser insuficientes para atender las condiciones de hardware restringido, generando la necesidad de diseñar modelos adaptados que respondan tanto a los requerimientos de precisión como a los de optimización de recursos (Javed et al., 2024).

Este trabajo de grado se centra en el análisis y evaluación de técnicas de aprendizaje automático ligero para la detección de malware en dispositivos IoT y sistemas embebidos. A diferencia de un estudio experimental orientado al entrenamiento de un modelo propio, la investigación se fundamenta en una revisión sistemática de literatura y en un análisis comparativo de estudios que reportan resultados sobre datasets públicos, métricas de desempeño

y criterios de eficiencia computacional. Desde esta perspectiva, el aporte principal consiste en sintetizar evidencia reciente y proponer un marco de aplicación que oriente la selección de técnicas ligeras en escenarios con restricciones de memoria, procesamiento, latencia y consumo energético.

La pertinencia del estudio radica en que los dispositivos IoT suelen operar bajo condiciones limitadas de hardware, conectividad y energía, lo que reduce la viabilidad de enfoques tradicionales de detección que demandan alta capacidad de procesamiento. En consecuencia, la literatura reciente ha prestado mayor atención a técnicas como poda, cuantización, selección de características, TinyML y aprendizaje federado, debido a su potencial para equilibrar capacidad predictiva y eficiencia computacional. Esta tensión entre desempeño y bajo consumo constituye el eje analítico de la investigación.

Planteamiento del problema

En un escenario de digitalización acelerada, la expansión del Internet de las Cosas (IoT) ha incrementado de manera significativa la superficie de ataque al incorporar dispositivos con fuertes restricciones de procesamiento, memoria y energía. En estos entornos, las soluciones tradicionales de detección de malware —diseñadas para plataformas de mayor capacidad computacional— resultan ineficientes o incluso inviables, comprometiendo la confidencialidad, la integridad y la disponibilidad de servicios críticos.

Esta problemática se intensifica en un contexto global cada vez más interconectado, donde la proliferación de sensores, cámaras y sistemas embebidos ha convertido dispositivos previamente considerados periféricos en potenciales vectores de vulnerabilidad. Las tecnologías digitales han incidido en los procesos sociales, productivos e institucionales, al facilitar nuevas formas de acceso a la información, comunicación, prestación de servicios y toma de decisiones. Sin embargo, este avance también plantea retos asociados con desigualdad digital, privacidad, seguridad, gobernanza de datos y uso responsable de la tecnología (Naciones Unidas, s. f.). En la misma línea, Fagan et al. (2021), en la publicación NIST SP 800-213, advierten que la expansión del IoT amplía la superficie de exposición y puede incrementar el riesgo de explotación por parte de actores maliciosos.

En el ámbito regional, países en desarrollo como Colombia enfrentan desafíos estructurales en la protección de estos entornos, debido a la limitada disponibilidad de recursos técnicos, económicos y humanos. Aunque existen políticas públicas que promueven la

innovación tecnológica, persiste un vacío en la implementación de mecanismos ligeros y robustos que garanticen seguridad en dispositivos con baja capacidad de procesamiento (Fenanir et al., 2019). En entidades públicas municipales, el uso de dispositivos IoT para monitoreo ambiental, gestión del tráfico o prestación de servicios comunitarios carece, en muchos casos, de herramientas adecuadas para prevenir intrusiones o ataques de malware, lo que incrementa su nivel de exposición.

En consecuencia, la brecha entre las capacidades de los modelos tradicionales de detección y las restricciones reales del hardware en dispositivos IoT configura un problema técnico y estratégico. Si no se desarrollan soluciones adaptadas a estos límites operativos, la exposición a ataques podría comprometer de manera directa la estabilidad y confiabilidad de servicios soportados por tecnologías emergentes. De allí surge la necesidad de investigar modelos ligeros de aprendizaje automático que logren equilibrar precisión de detección y eficiencia computacional en entornos de recursos restringidos.

Formulación del problema

El problema central identificado es la alta vulnerabilidad de los dispositivos IoT y sistemas embebidos frente a ataques de malware, derivada de la ausencia de mecanismos de detección que integren eficiencia computacional y precisión de clasificación bajo restricciones estrictas de hardware. Las soluciones convencionales, diseñadas para infraestructuras con mayor capacidad de procesamiento, no responden adecuadamente a los límites operativos de estos entornos, generando una brecha entre desempeño teórico y viabilidad práctica.

En este sentido, se plantea la necesidad de establecer un marco de análisis que permita orientar la selección de técnicas de aprendizaje automático ligero para la detección de malware en entornos IoT y sistemas embebidos. Dicho marco debe considerar no solo el desempeño predictivo reportado en la literatura, sino también criterios de eficiencia computacional, viabilidad operativa, características observables, tipo de amenaza y restricciones propias del dispositivo.

Pregunta problema

¿En qué medida un modelo ligero de aprendizaje automático —optimizado mediante técnicas como poda, cuantización y/o aprendizaje federado— puede mejorar la precisión y la eficiencia de la detección de malware frente a métodos tradicionales en dispositivos IoT con recursos computacionales limitados?

Justificación

La presente monografía se justifica desde tres dimensiones: teórica, práctica y metodológica.

Justificación teórica. La propuesta aporta al cuerpo de conocimiento al evaluar sistemáticamente técnicas de optimización (poda, cuantización y entrenamiento federado) y su impacto simultáneo sobre exactitud, F1, latencia, consumo energético y uso de memoria, respondiendo vacíos de validación y estandarización señalados por la literatura reciente.

Justificación práctica. En contextos con restricciones de infraestructura y presupuesto, un detector ligero y reproducible mejora la viabilidad de la defensa en IoT, reduce costos operativos y facilita la adopción institucional. El diagnóstico previo del contexto local refuerza la pertinencia aplicada del proyecto.

Justificación metodológica. Este trabajo se sustenta en una revisión sistemática de literatura y una síntesis analítico-comparativa sobre detección de malware en entornos IoT y sistemas embebidos. Este enfoque permite organizar la evidencia disponible, identificar tendencias, reconocer limitaciones y establecer criterios de comparación entre enfoques tradicionales y técnicas de aprendizaje automático ligero. Además, la matriz de extracción, la evaluación de calidad y el marco de aplicación propuesto fortalecen la trazabilidad metodológica del estudio.

Supuesto orientador de análisis

En coherencia con el enfoque monográfico, analítico-comparativo y basado en revisión sistemática de literatura, se plantea como supuesto orientador que las técnicas de aprendizaje automático ligero, tales como selección de características, poda, cuantización, TinyML y aprendizaje federado, pueden contribuir a mejorar la viabilidad de la detección de malware en dispositivos IoT y sistemas embebidos cuando son evaluadas mediante criterios multicriterio. Estos criterios integran tanto el desempeño predictivo como la eficiencia computacional, considerando métricas como F1-score, exactitud, latencia de inferencia, consumo de memoria, tamaño del modelo, consumo energético, conectividad y privacidad.

Este supuesto no se orienta a una validación experimental propia ni al desarrollo de un modelo implementado en hardware, sino a la interpretación comparativa de la evidencia reportada en la literatura científica. En consecuencia, la monografía busca identificar criterios técnicos que permitan valorar la pertinencia de técnicas ligeras frente a enfoques tradicionales, considerando las restricciones operativas que caracterizan a los dispositivos IoT con recursos computacionales limitados.

Desde esta perspectiva, el análisis se centra en establecer cómo las técnicas de optimización y despliegue ligero pueden apoyar la selección de soluciones de detección más viables para entornos IoT y sistemas embebidos. Por tanto, el supuesto orientador permite articular el estado del arte, la caracterización de amenazas, la comparación de enfoques y el marco de aplicación propuesto, sin asumir una comprobación experimental directa.

Objetivos

Objetivo general

Evaluar la viabilidad de técnicas de aprendizaje automático ligero para la detección de malware en dispositivos IoT y sistemas embebidos con restricciones computacionales.

Objetivos específicos

Caracterizar el estado del arte sobre técnicas de aprendizaje automático ligero aplicadas a la detección de malware en dispositivos IoT y sistemas embebidos.

Identificar tipos de malware, vectores de ataque y patrones observables relevantes para la detección ligera en entornos IoT.

Comparar el desempeño y la eficiencia de modelos ligeros reportados en la literatura frente a enfoques tradicionales de detección.

Formular un marco de aplicación que oriente la selección de técnicas ligeras de detección de malware según contexto, amenaza, características observables y restricciones operativas.

Marco referencial

Antecedentes

La necesidad de modelos ligeros de aprendizaje automático para la detección de malware en entornos IoT y de edge computing ha sido abordada en investigaciones internacionales, nacionales y regionales.

A nivel internacional, Fenanir et al. (2019) diseñaron un marco de clasificación de malware en dispositivos IoT empleando técnicas de reducción como matrix block mean downsampling. Sus resultados mostraron que es posible alcanzar un equilibrio entre eficiencia y precisión en dispositivos con recursos limitados. Por su parte, Rey et al. (2021) desarrollaron un modelo de detección de malware basado en aprendizaje federado en entornos de edge computing, con el fin de preservar la privacidad y reducir el consumo de recursos. Ambos trabajos constituyen referentes clave para la presente investigación, al demostrar la viabilidad de soluciones ligeras aplicables a hardware restringido.

En el contexto nacional, la transformación digital en Colombia plantea retos asociados con conectividad, apropiación tecnológica, seguridad y confianza digital. La Estrategia Nacional Digital de Colombia 2023–2026 reconoce que el país aún enfrenta brechas en el acceso, uso y apropiación de tecnologías digitales entre hogares, entidades públicas, empresas y territorios. En particular, el documento reporta diferencias en la penetración de internet fijo en hogares según el tipo de municipio: 6 % en rural disperso, 11,6 % en rural, 25,8 % en ciudades intermedias y 68

% en centros y aglomeraciones, frente a un valor nacional de 50,7 % para el cuarto trimestre de 2022 (Departamento Nacional de Planeación et al., 2023).

Desde el plano regional, el Observatorio Colombiano de Ciencia y Tecnología (OCyT, 2024) reportó proyectos orientados al fortalecimiento de procesos de gobernanza en el sistema de CTeI en subregiones de Caldas, mediante estrategias participativas, descentralizadas y basadas en datos. El mismo informe registra iniciativas relacionadas con la caracterización de los niveles de adopción y uso de tecnologías digitales por parte de actores del Ecosistema Nacional de Ciencia, Tecnología e Innovación, con el propósito de generar información estadística útil para analistas, formuladores de política y actores interesados en la digitalización de la CTeI en Colombia.

Estos antecedentes evidencian la pertinencia de estudiar soluciones tecnológicas viables en contextos territoriales con distintos niveles de madurez digital. En consecuencia, el análisis de técnicas de aprendizaje automático ligero para la detección de malware se articula con la necesidad de contar con enfoques eficientes, adaptables y aplicables a escenarios institucionales y regionales con restricciones técnicas, operativas y de conectividad (Departamento Nacional de Planeación et al., 2023; OCyT, 2024).

Marco conceptual

El marco conceptual de esta investigación articula los conceptos centrales que permiten comprender la relación entre dispositivos IoT, amenazas de malware, restricciones

computacionales y técnicas de aprendizaje automático ligero. A diferencia del glosario, este apartado no se limita a definir términos, sino que explica cómo cada concepto se vincula con el problema de investigación y de qué manera contribuye a la construcción del análisis propuesto.

Internet de las Cosas e infraestructura de bajo recurso

El Internet de las Cosas (IoT) se entiende como un ecosistema de dispositivos físicos interconectados que capturan, procesan y transmiten datos a través de redes de comunicación. En este entorno se incluyen sensores, cámaras, medidores inteligentes, dispositivos médicos, gateways, microcontroladores y sistemas embebidos que operan en sectores como salud, industria, ciudades inteligentes, hogares conectados e infraestructura pública. Su importancia dentro de esta investigación radica en que muchos de estos dispositivos funcionan con recursos limitados de memoria, procesamiento, almacenamiento y energía, lo que condiciona la implementación de mecanismos tradicionales de ciberseguridad.

En el contexto del estudio, el IoT no se analiza únicamente como una arquitectura tecnológica, sino como un entorno distribuido, heterogéneo y expuesto a múltiples vectores de ataque. De acuerdo con Javed et al. (2024), la aplicación de técnicas de aprendizaje automático y aprendizaje profundo en IoT ha ganado relevancia debido a la necesidad de detectar comportamientos anómalos en escenarios donde los métodos convencionales pueden resultar insuficientes. Sin embargo, la misma naturaleza distribuida del IoT genera desafíos adicionales, puesto que no todos los dispositivos tienen la capacidad necesaria para ejecutar modelos complejos de detección.

Los sistemas embebidos, por su parte, son dispositivos diseñados para ejecutar funciones específicas dentro de un sistema mayor. Esta característica los hace frecuentes en entornos IoT, pero también limita sus capacidades computacionales. Por ello, la detección de malware en estos dispositivos requiere modelos que no solo sean precisos, sino también viables en términos de latencia, memoria, tamaño del modelo y consumo energético. Esta condición justifica el interés de la investigación por los modelos ligeros de aprendizaje automático.

Malware y superficie de ataque en entornos IoT

El malware se refiere a software malicioso diseñado para afectar la confidencialidad, integridad o disponibilidad de los sistemas de información. En entornos IoT, su impacto puede ser especialmente crítico, dado que los dispositivos suelen estar conectados de forma permanente, operar con configuraciones heterogéneas y, en muchos casos, carecer de mecanismos robustos de actualización y monitoreo. Esta condición facilita la propagación de botnets, ataques de denegación de servicio, cryptojacking, backdoors y variantes de malware orientadas a dispositivos con baja capacidad de defensa.

En esta investigación, el malware se interpreta como un fenómeno técnico que genera patrones observables en el tráfico de red, el comportamiento del sistema y el consumo de recursos. Por ejemplo, una botnet puede producir conexiones repetitivas hacia múltiples destinos; un malware de criptominería puede elevar el uso de CPU y energía; y una comunicación de comando y control puede generar tráfico periódico hacia servidores externos. Esta relación entre

amenaza, comportamiento observable y variable medible resulta fundamental para justificar el uso de modelos de aprendizaje automático en la detección de malware.

La superficie de ataque corresponde al conjunto de puntos, servicios, interfaces, protocolos y configuraciones que pueden ser explotados por un atacante. En IoT, dicha superficie tiende a ampliarse por factores como credenciales débiles, servicios expuestos, protocolos sin cifrado, segmentación deficiente, actualizaciones inseguras y bajo control sobre el ciclo de vida de los dispositivos. En consecuencia, el análisis de la superficie de ataque permite comprender por qué la detección de malware en IoT no puede depender únicamente de firmas estáticas, sino que requiere enfoques capaces de identificar comportamientos anómalos en tiempo cercano al real.

Los vectores de ataque representan las rutas mediante las cuales una amenaza compromete un sistema. En IoT, estos pueden incluir puertos abiertos, protocolos inseguros, credenciales por defecto, interfaces de administración remota y mecanismos de actualización vulnerables. Su análisis aporta al estudio porque permite vincular el tipo de amenaza con las características que pueden ser observadas y utilizadas por modelos ligeros de detección.

Aprendizaje automático aplicado a la detección de malware

El aprendizaje automático permite construir modelos capaces de identificar patrones a partir de datos. En ciberseguridad, esta capacidad resulta útil para diferenciar comportamientos benignos y maliciosos, especialmente cuando las amenazas evolucionan y las técnicas basadas en firmas no son suficientes para detectar variantes nuevas, polimórficas u ofuscadas. En este

sentido, los modelos de aprendizaje automático pueden apoyar la detección de anomalías, clasificación de tráfico malicioso y reconocimiento de patrones asociados a malware.

En el marco de esta investigación, el aprendizaje automático se relaciona con la detección de malware a partir de variables como duración del flujo, tamaño de paquetes, volumen de bytes transmitidos, número de paquetes por flujo, uso de CPU, uso de memoria, procesos activos y relación entre paquetes enviados y recibidos. Estas variables permiten representar el comportamiento del dispositivo o de la red de manera estructurada, facilitando el análisis de patrones maliciosos.

No obstante, los modelos tradicionales de aprendizaje automático y, en especial, algunas arquitecturas de aprendizaje profundo pueden requerir altos niveles de procesamiento, memoria y energía. Esta situación plantea una tensión entre desempeño predictivo y viabilidad operativa. Javed et al. (2024) señalan que los sistemas de detección de intrusiones en IoT deben considerar no solo la exactitud del modelo, sino también las limitaciones propias del entorno donde será desplegado. Por esta razón, el análisis conceptual del aprendizaje automático en esta investigación se orienta hacia técnicas que permitan conservar capacidad de detección con menor costo computacional.

Aprendizaje automático ligero

El aprendizaje automático ligero se refiere al conjunto de enfoques, algoritmos y técnicas orientadas a reducir la complejidad computacional de los modelos sin afectar de manera

sustancial su capacidad predictiva. En esta investigación, el concepto es central porque responde al problema de implementar mecanismos de detección de malware en dispositivos IoT y sistemas embebidos con restricciones de memoria, procesamiento, latencia y energía.

Un modelo ligero no se define únicamente por su tamaño reducido, sino por su capacidad para equilibrar eficacia y eficiencia. Esto significa que debe mantener resultados aceptables en métricas como accuracy, precision, recall o F1-score, al tiempo que reduce consumo de memoria, tamaño del modelo, latencia de inferencia y, cuando sea posible, consumo energético. Por tanto, la ligereza del modelo no es un atributo secundario, sino una condición técnica para su despliegue en escenarios reales de IoT.

Entre las técnicas asociadas al aprendizaje automático ligero se encuentran la selección de características, la poda estructural, la cuantización, TinyML, la destilación de conocimiento y el aprendizaje federado. Cada una contribuye de forma distinta al problema de investigación: la selección de características reduce el número de variables procesadas; la poda disminuye parámetros o conexiones del modelo; la cuantización compacta la representación numérica; TinyML permite inferencia en microcontroladores; y el aprendizaje federado reduce la necesidad de centralizar datos. En conjunto, estas técnicas permiten abordar la detección de malware desde una perspectiva multicriterio.

Selección de características

La selección de características consiste en identificar las variables más relevantes para representar un fenómeno, eliminando aquellas que aportan poco valor predictivo o incrementan innecesariamente la complejidad del modelo. En entornos IoT, esta técnica resulta especialmente importante porque permite reducir el volumen de datos procesados, disminuir el consumo de memoria y mejorar la velocidad de inferencia.

En esta investigación, la selección de características se relaciona con la identificación de patrones observables de bajo costo computacional. Variables como duración del flujo, tamaño promedio de paquetes, total de bytes enviados y recibidos, número de paquetes por flujo, uso de CPU, uso de memoria y número de procesos activos permiten capturar señales relevantes de comportamiento malicioso sin requerir inspección profunda de paquetes o procesamiento intensivo.

Este concepto aporta directamente al estudio porque permite conectar la amenaza con variables medibles. Así, la detección de malware no se plantea como un proceso abstracto, sino como una relación entre vectores de ataque, rastros observables y modelos capaces de interpretar esos patrones. Desde esta perspectiva, la selección de características contribuye a mejorar la eficiencia del modelo y a facilitar su implementación en dispositivos de bajo recurso.

Poda estructural y cuantización

La poda estructural es una técnica de optimización que elimina parámetros, conexiones, filtros o componentes del modelo que tienen baja contribución al resultado final. Su propósito es reducir el tamaño del modelo, acelerar la inferencia y disminuir la carga computacional. En el contexto de IoT, esta técnica es relevante porque permite adaptar modelos originalmente complejos a dispositivos con menor capacidad de procesamiento.

La cuantización, por su parte, reduce la precisión numérica de los parámetros del modelo, por ejemplo, al pasar de representaciones de 32 bits a formatos más compactos. Esta reducción disminuye el consumo de memoria y puede acelerar el procesamiento, especialmente en hardware limitado. Sharmila y Nagapadma (2023) muestran la pertinencia de enfoques cuantizados en dispositivos de borde, lo que resulta coherente con la necesidad de evaluar modelos no solo por su capacidad predictiva, sino también por su eficiencia operativa.

Ambas técnicas afectan la relación entre precisión y eficiencia. Por esta razón, su análisis no debe limitarse a la reducción de tamaño del modelo, sino que debe considerar si la posible pérdida de desempeño es aceptable frente a la ganancia en latencia, memoria o consumo energético. En dispositivos IoT, una reducción moderada en accuracy o F1-score puede ser técnicamente justificable si permite ejecutar el modelo de manera local y estable.

TinyML e inferencia en el borde

TinyML se refiere a la implementación de modelos de aprendizaje automático en dispositivos de muy bajo consumo, como microcontroladores y sistemas embebidos. Warden y Situnayake (2019) plantean que TinyML permite trasladar capacidades de inferencia a dispositivos con recursos reducidos, lo que resulta especialmente pertinente para aplicaciones donde la conectividad, la latencia y la privacidad representan restricciones relevantes.

La inferencia en el borde consiste en procesar los datos cerca de la fuente donde se generan, sin depender completamente de servicios en la nube. En entornos IoT, esta aproximación puede reducir tiempos de respuesta, disminuir la transmisión de datos sensibles y mejorar la continuidad operativa cuando la conectividad es limitada. Para la detección de malware, esto implica la posibilidad de identificar comportamientos anómalos directamente en el dispositivo o en nodos cercanos, antes de que la amenaza se propague.

No obstante, TinyML también impone desafíos técnicos, como la necesidad de modelos compactos, extracción eficiente de características y validación bajo condiciones reales de hardware. Por ello, su relación con el estudio no se reduce a la ejecución local de modelos, sino que se vincula con la necesidad de seleccionar técnicas, variables y métricas compatibles con dispositivos de bajo recurso.

Aprendizaje federado y privacidad

El aprendizaje federado es un enfoque de entrenamiento distribuido que permite actualizar modelos sin centralizar los datos de los dispositivos. En lugar de enviar datos crudos a un servidor central, cada nodo entrena localmente y comparte actualizaciones del modelo. Esta estrategia resulta relevante en IoT porque muchos dispositivos operan en entornos donde la privacidad, la descentralización y la reducción del tráfico de comunicación son condiciones críticas.

Rey et al. (2021) destacan que el aprendizaje federado puede contribuir al análisis de malware en IoT al permitir entrenamiento distribuido y preservación de datos en origen. De manera complementaria, Rey et al. (2021) abordan el aprendizaje federado como una alternativa ligera y orientada a la privacidad para detección de malware en escenarios de edge computing. Estas perspectivas son relevantes para la presente investigación porque amplían el concepto de ligereza más allá del tamaño del modelo, incorporando también el costo de comunicación y la protección de datos.

Sin embargo, el aprendizaje federado también presenta limitaciones, como heterogeneidad de dispositivos, diferencias en la calidad de los datos locales, sincronización entre nodos y sobrecarga por intercambio de actualizaciones. Por tanto, su aplicabilidad en IoT debe analizarse en función del contexto operativo, la disponibilidad de conectividad y el tipo de amenaza que se busca detectar.

Métricas de desempeño y eficiencia computacional

Las métricas de desempeño permiten evaluar la capacidad de un modelo para clasificar correctamente comportamientos benignos y maliciosos. Entre las más utilizadas se encuentran accuracy, precision, recall y F1-score. En problemas de detección de malware, el F1-score resulta especialmente relevante porque equilibra precisión y exhaustividad, lo cual es útil cuando existen desbalances entre clases benignas y maliciosas.

No obstante, en entornos IoT no basta con evaluar el desempeño predictivo. También es necesario considerar métricas de eficiencia computacional, como latencia, consumo de memoria, tamaño del modelo y consumo energético. Estas métricas determinan si un modelo puede ejecutarse realmente en dispositivos con recursos limitados. En este sentido, la eficiencia no es un criterio complementario, sino un componente central de la evaluación.

La investigación adopta una visión multicriterio, en la cual un modelo no se considera adecuado únicamente por alcanzar alta precisión, sino por mantener un balance entre capacidad de detección y viabilidad operativa. Esta perspectiva evita sobrevalorar modelos complejos que pueden presentar buen desempeño en entornos controlados, pero resultar impracticables en escenarios reales de IoT.

Datasets para evaluación en IoT

Los datasets son conjuntos de datos utilizados por la literatura para entrenar, validar o comparar modelos de detección. En investigaciones sobre malware e intrusiones en IoT, datasets como IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022 permiten analizar tráfico anómalo, botnets, telemetría, comportamientos maliciosos e indicadores de malware en memoria.

Koroniotis et al. (2019) presentan BoT-IoT como un dataset orientado al análisis forense de botnets en IoT. Moustafa et al. (2020) describen TON_IoT como un conjunto de datos diseñado para evaluar aplicaciones de ciberseguridad basadas en inteligencia artificial en entornos IoT e IIoT. A su vez, el Canadian Institute for Cybersecurity (2022) presenta CIC-MalMem-2022 como un dataset relevante para el análisis de malware en memoria. Estos datasets son importantes para la investigación porque permiten comprender qué escenarios han sido utilizados por los estudios revisados y qué métricas se reportan con mayor frecuencia.

En este trabajo, los datasets no se emplean para ejecutar experimentos propios, sino como referentes reportados por la literatura analizada. Su función conceptual consiste en sustentar la comparación de enfoques, identificar limitaciones de generalización y reconocer patrones observables que pueden ser útiles para modelos ligeros.

Relación conceptual para el estudio

Los conceptos desarrollados permiten comprender que la detección de malware en IoT no depende de un único elemento técnico, sino de la relación entre amenaza, vector de ataque, característica observable, modelo de detección y restricción operativa. Esta relación orienta el análisis de la investigación y justifica la necesidad de modelos ligeros.

Desde esta perspectiva, el aprendizaje automático ligero se entiende como una respuesta técnica al problema de implementar ciberseguridad en dispositivos distribuidos, heterogéneos y limitados en recursos. Por tanto, el aporte conceptual del estudio consiste en articular la eficacia predictiva con la eficiencia computacional, reconociendo que en entornos IoT la mejor solución no siempre es la más precisa, sino aquella que logra un equilibrio razonable entre detección, consumo de recursos y factibilidad de despliegue.

Marco teórico

El presente marco teórico proporciona la base analítica que orienta el estudio sobre técnicas de aprendizaje automático ligero para la detección de malware en dispositivos IoT y sistemas embebidos. Su propósito es integrar fundamentos de IoT, ciberamenazas, enfoques de detección, modelos ligeros, paradigmas de cómputo en el borde y métricas de evaluación, con el fin de explicar cómo se configura el problema de investigación: lograr un equilibrio entre eficacia predictiva y eficiencia computacional en entornos con recursos restringidos.

Dado que este apartado emplea términos técnicos propios de la ciberseguridad, el aprendizaje automático y las arquitecturas IoT, los conceptos clave desarrollados se articulan con el glosario del documento. En este sentido, términos como malware, superficie de ataque, vector de ataque, botnet, TinyML, poda estructural, cuantización, aprendizaje federado, latencia, F1-score, dataset, inferencia en el borde y modelos ligeros deben estar definidos en el glosario para facilitar la comprensión del lector y mantener coherencia terminológica en todo el documento.

Fundamentos de IoT y sistemas embebidos

El Internet de las Cosas (IoT) se compone de dispositivos heterogéneos, tales como sensores, actuadores, cámaras, gateways, microcontroladores y sistemas embebidos, capaces de recopilar, procesar y transmitir información mediante redes de comunicación. Estos dispositivos suelen operar en contextos sensibles, como salud, industria, hogares inteligentes, ciudades inteligentes e infraestructura pública, donde la disponibilidad, integridad y confidencialidad de los datos resultan críticas.

Desde la perspectiva de la ciberseguridad, el IoT plantea retos particulares debido a la diversidad de fabricantes, protocolos, sistemas operativos, capacidades de actualización y niveles de protección. Además, muchos dispositivos funcionan con limitaciones de procesamiento, memoria, almacenamiento y energía, lo cual condiciona la adopción de soluciones tradicionales de detección de amenazas. Javed et al. (2024) señalan que el uso de aprendizaje automático en entornos IoT ha ganado relevancia debido a la necesidad de identificar patrones anómalos en

escenarios distribuidos; sin embargo, dicha aplicación debe considerar las restricciones propias del entorno.

Los sistemas embebidos comparten esta problemática, dado que están diseñados para cumplir funciones específicas dentro de un sistema mayor y no necesariamente para ejecutar mecanismos complejos de seguridad. Por esta razón, cualquier estrategia de detección de malware orientada a IoT debe considerar no solo la precisión del modelo, sino también su viabilidad operativa en términos de latencia, consumo de memoria, tamaño del modelo y consumo energético.

Tipos de malware y vectores de ataque en IoT

El ecosistema IoT es vulnerable a distintas familias de malware, entre ellas botnets, ransomware, spyware, cryptojacking, backdoors y malware en memoria. Estas amenazas explotan superficies de ataque asociadas a credenciales débiles, servicios expuestos, protocolos inseguros, configuraciones por defecto, puertos de administración remota y mecanismos de actualización poco protegidos. En este contexto, la superficie de ataque se amplía no solo por la cantidad de dispositivos conectados, sino también por la heterogeneidad de sus configuraciones y por la limitada capacidad de monitoreo.

En esta investigación, el malware se interpreta como un fenómeno técnico que genera rastros observables en el tráfico de red, el comportamiento del sistema y el consumo de recursos. Por ejemplo, una botnet puede producir ráfagas de conexiones, tráfico repetitivo o intentos de

comunicación hacia múltiples destinos; un malware de cryptojacking puede reflejarse en incrementos sostenidos del uso de CPU o consumo energético; y una comunicación de comando y control puede generar patrones periódicos de tráfico. Esta relación entre amenaza, vector de ataque y característica observable resulta fundamental para justificar el uso de modelos de aprendizaje automático en la detección de malware.

El análisis de vectores de ataque permite vincular la forma de ingreso o propagación de una amenaza con las señales que pueden ser capturadas por modelos ligeros. Por ello, en lugar de estudiar el malware únicamente por su clasificación general, el presente trabajo lo analiza a partir de patrones útiles para la detección: duración del flujo, tamaño de paquetes, volumen de bytes transmitidos, número de paquetes por flujo, relación entre paquetes enviados y recibidos, uso de CPU, uso de memoria y procesos activos.

Enfoques de detección de malware

La detección de malware puede abordarse desde diferentes enfoques técnicos. Los métodos basados en firmas comparan archivos, tráfico o comportamientos con patrones previamente conocidos. Aunque son eficientes frente a amenazas ya identificadas, presentan limitaciones ante variantes polimórficas, malware ofuscado o ataques nuevos. Los enfoques heurísticos y estadísticos permiten establecer reglas o umbrales para detectar comportamientos sospechosos, pero pueden generar falsos positivos cuando el tráfico legítimo presenta alta variabilidad.

Por su parte, el aprendizaje automático tradicional permite entrenar modelos de clasificación o detección de anomalías a partir de características extraídas del tráfico o del comportamiento del sistema. Algoritmos como Random Forest, SVM, árboles de decisión o modelos lineales pueden ofrecer una relación favorable entre desempeño y costo computacional cuando las características han sido seleccionadas adecuadamente. En escenarios IoT, este enfoque resulta pertinente porque puede adaptarse a restricciones de recursos si se controla la dimensionalidad de los datos.

El aprendizaje profundo, mediante arquitecturas como CNN, LSTM o modelos híbridos, puede capturar relaciones complejas en secuencias de tráfico o datos de telemetría. Sin embargo, estos modelos suelen requerir mayor capacidad de procesamiento, memoria y energía, lo que limita su despliegue directo en dispositivos de bajo recurso. Javed et al. (2024) destacan que los sistemas de detección en IoT deben evaluarse considerando simultáneamente la capacidad predictiva y las restricciones de implementación. En consecuencia, la selección del enfoque de detección debe responder a una lógica multicriterio y no únicamente a la métrica de exactitud.

Técnicas de aligeramiento del modelo

Las técnicas de aligeramiento buscan reducir la complejidad de los modelos sin afectar de manera sustancial su capacidad de detección. Estas técnicas son centrales en entornos IoT porque permiten adaptar modelos de aprendizaje automático a dispositivos con limitaciones de memoria, procesamiento y energía.

La selección de características permite identificar variables relevantes y eliminar aquellas que aportan poco valor predictivo o incrementan innecesariamente la carga computacional. En detección de malware, esta técnica ayuda a priorizar rasgos observables de bajo costo, como duración del flujo, tamaño promedio de paquetes, volumen de bytes, uso de CPU, uso de memoria o procesos activos.

La poda estructural elimina pesos, filtros, conexiones o componentes del modelo que tienen baja contribución al resultado final. Su propósito es reducir el tamaño del modelo y acelerar la inferencia. Esta técnica resulta útil cuando se pretende adaptar arquitecturas más complejas a escenarios de ejecución limitada.

La cuantización reduce la precisión numérica de los parámetros del modelo, por ejemplo, al pasar de representaciones de 32 bits a formatos más compactos. Sharmila y Nagapadma (2023) muestran la pertinencia de modelos cuantizados para dispositivos de borde, dado que este enfoque puede disminuir memoria y latencia manteniendo niveles aceptables de desempeño.

La destilación de conocimiento transfiere el comportamiento de un modelo complejo, denominado modelo docente, a un modelo más pequeño o modelo estudiante. Este enfoque permite conservar parte de la capacidad predictiva de modelos robustos, pero con menor costo computacional. Finalmente, TinyML permite ejecutar modelos de aprendizaje automático en microcontroladores y sistemas embebidos de muy bajo consumo, acercando la inferencia al dispositivo y reduciendo dependencia de la nube (Warden & Situnayake, 2019).

En conjunto, estas técnicas muestran que la eficiencia no debe interpretarse como una pérdida de calidad, sino como una condición necesaria para que la detección de malware sea aplicable en dispositivos IoT reales.

Paradigmas de entrenamiento y privacidad: aprendizaje federado

El aprendizaje federado es un paradigma de entrenamiento distribuido que permite actualizar modelos sin centralizar los datos generados por los dispositivos. En este enfoque, cada nodo entrena localmente y comparte actualizaciones del modelo, no los datos originales. Esta característica resulta especialmente relevante en IoT, donde la privacidad, la conectividad limitada y la distribución geográfica de los dispositivos son factores críticos.

Rey et al. (2021) señalan que el aprendizaje federado puede contribuir al análisis de malware en IoT al permitir entrenamiento distribuido y preservación de datos en origen. De manera complementaria, Rey et al. (2021) plantean que este enfoque puede ser útil en escenarios de edge computing cuando se requiere reducir transmisión de datos y proteger información sensible.

No obstante, el aprendizaje federado también enfrenta desafíos técnicos, como heterogeneidad de dispositivos, datos no distribuidos de manera uniforme, sincronización entre nodos, sobrecarga de comunicación y riesgo de actualizaciones maliciosas. Por tanto, su aplicabilidad debe analizarse según el contexto operativo, la capacidad del dispositivo y el tipo de amenaza que se busca detectar. En esta investigación, el aprendizaje federado se considera

una alternativa relevante cuando la ligereza no solo implica reducir el tamaño del modelo, sino también disminuir el costo de comunicación y preservar datos en origen.

Datos de referencia y métricas de evaluación

La evaluación de modelos para detección de malware en IoT suele apoyarse en datasets públicos que permiten comparar enfoques bajo escenarios reproducibles. Entre los conjuntos de datos más utilizados se encuentran IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022. Garcia et al. (2020) describen IoT-23 como un dataset con tráfico benigno y malicioso en escenarios IoT; Koroniotis et al. (2019) presentan BoT-IoT como un conjunto orientado al análisis de botnets en IoT; Moustafa et al. (2020) proponen TON_IoT para evaluar aplicaciones de ciberseguridad basadas en inteligencia artificial en entornos IoT e IIoT; y Canadian Institute for Cybersecurity (2022) documenta CIC-MalMem-2022 como un dataset asociado al análisis de malware en memoria.

En este estudio, dichos datasets no se emplean para ejecutar experimentos propios, sino como referentes reportados por la literatura revisada. Su importancia radica en que permiten identificar qué escenarios han sido más utilizados por los estudios analizados, qué métricas se reportan con mayor frecuencia y cuáles son las limitaciones de comparabilidad entre investigaciones.

Las métricas funcionales más comunes incluyen accuracy, precision, recall y F1-score. En problemas de detección de malware, el F1-score resulta especialmente importante porque

equilibra precision y recall, lo cual es útil cuando existe desbalance entre clases benignas y maliciosas. Sin embargo, en entornos IoT también deben considerarse métricas de eficiencia, como latencia de inferencia, consumo de memoria, tamaño del modelo y consumo energético. Esta perspectiva multicriterio permite evaluar si un modelo no solo clasifica correctamente, sino si puede operar dentro de las restricciones del dispositivo objetivo.

Modelos y teorías aplicables

El estudio se apoya en tres ejes teóricos que permiten interpretar la relación entre desempeño predictivo y eficiencia computacional. El primero corresponde a la teoría del aprendizaje estadístico y la relación sesgo-varianza, la cual permite comprender cómo la complejidad del modelo incide en su capacidad de generalización. En entornos de recursos restringidos, controlar la complejidad mediante selección de características, regularización o simplificación del modelo puede favorecer la generalización y reducir el sobreajuste.

El segundo eje corresponde al análisis de trade-offs entre rendimiento y eficiencia. En IoT, un modelo no debe evaluarse únicamente por alcanzar altos valores de accuracy o F1-score, sino por su capacidad de operar con baja latencia, menor consumo de memoria y bajo requerimiento energético. Esta perspectiva permite comprender que una pequeña reducción en desempeño predictivo puede ser aceptable si produce mejoras significativas en viabilidad de despliegue.

El tercer eje se relaciona con la privacidad por diseño y la computación distribuida. Este enfoque sustenta decisiones como el uso de inferencia en el borde y aprendizaje federado, dado que busca reducir exposición de datos, minimizar dependencia de la nube y mejorar la capacidad de respuesta local. En consecuencia, la detección de malware en IoT debe entenderse como un problema sociotécnico donde convergen seguridad, desempeño, eficiencia, privacidad y factibilidad operativa.

Los conceptos técnicos desarrollados en este apartado se articulan con el glosario del documento, con el fin de facilitar la comprensión de términos especializados como malware, superficie de ataque, vector de ataque, TinyML, poda estructural, cuantización, aprendizaje federado, latencia, F1-score, dataset, PRISMA y PICOC.

Antecedentes y estado del arte

Los antecedentes revisados permiten identificar tres líneas principales de investigación relacionadas con la detección de malware en dispositivos IoT y sistemas embebidos mediante enfoques ligeros. La primera línea se concentra en modelos de aprendizaje automático clásico optimizados mediante selección de características, los cuales resultan pertinentes para escenarios IoT por su relación favorable entre desempeño predictivo y bajo costo computacional (Fenanir et al., 2019; Javed et al., 2024).

La segunda línea corresponde al uso de modelos de aprendizaje profundo comprimidos, en los cuales técnicas como poda, cuantización y reducción de parámetros buscan disminuir el

tamaño del modelo, la latencia y el consumo de memoria. Estos enfoques adquieren relevancia cuando se pretende trasladar la inferencia hacia dispositivos de borde o microcontroladores, aunque su adopción depende de que la pérdida de desempeño sea aceptable frente a la ganancia en eficiencia (Sharmila & Nagapadma, 2023).

La tercera línea se relaciona con el aprendizaje federado y los enfoques distribuidos en el borde. Esta perspectiva es relevante en escenarios donde la privacidad, la descentralización de los datos y la reducción del tráfico de comunicación son condiciones necesarias para el despliegue de soluciones de detección. No obstante, la literatura también reporta retos asociados a la heterogeneidad de dispositivos, la convergencia del modelo y la sobrecarga de comunicación (Nguyen et al., 2021; Rey et al., 2021).

De manera complementaria, los estudios revisados evidencian una maduración progresiva de datasets orientados al análisis de tráfico IoT, botnets, telemetría y malware en memoria, tales como IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022. Estos conjuntos de datos han favorecido la comparación entre modelos, aunque no eliminan una limitación persistente en la literatura: la necesidad de validar los enfoques ligeros en hardware real y bajo condiciones operativas representativas. Esta brecha resulta relevante para el presente estudio, dado que la viabilidad de un modelo ligero no depende únicamente de su desempeño predictivo, sino también de su capacidad para operar bajo restricciones de memoria, latencia, tamaño del modelo y consumo energético.

Categorías de análisis del estudio

A partir del problema de investigación y de la evidencia identificada en la revisión sistemática, se adoptan cinco categorías de análisis que permiten organizar la comparación entre enfoques de detección. Estas categorías no constituyen resultados experimentales propios, sino criterios derivados de la literatura para interpretar la pertinencia de los modelos ligeros en entornos IoT.

La primera categoría corresponde a la eficacia de detección, entendida como la capacidad del modelo para clasificar correctamente comportamientos benignos y maliciosos. En esta categoría se consideran métricas como accuracy, precision, recall, F1-score, falsos positivos y falsos negativos.

La segunda categoría es la eficiencia computacional, asociada con el costo operativo del modelo. Incluye variables como latencia de inferencia, consumo de memoria, tamaño del modelo y consumo energético. Esta categoría es central para el estudio porque permite valorar si un enfoque puede desplegarse en dispositivos con recursos limitados.

La tercera categoría corresponde a la robustez y generalización, relacionada con el comportamiento del modelo frente a dispositivos, tráfico, datasets o ataques no observados durante el entrenamiento reportado en los estudios revisados. Esta categoría permite reconocer limitaciones de transferencia entre escenarios experimentales y entornos reales.

La cuarta categoría es la privacidad y comunicación, especialmente relevante en enfoques basados en aprendizaje federado o edge computing. En este caso se analiza si el modelo requiere centralización de datos, cuánto tráfico genera y qué implicaciones tiene para la protección de información sensible.

La quinta categoría corresponde a la desplegabilidad, entendida como la posibilidad de integrar el modelo en entornos IoT o sistemas embebidos. Incluye criterios como compatibilidad con runtimes de TinyML, footprint binario, complejidad de integración y facilidad de actualización.

Estas categorías permiten operacionalizar la comparación entre modelos ligeros y enfoques tradicionales, manteniendo coherencia con el diseño metodológico basado en revisión sistemática y análisis comparativo.

Marco referencial integrado

Como síntesis del marco conceptual, el marco teórico y los antecedentes revisados, se adopta un marco referencial integrado que relaciona cinco elementos: contexto operativo, amenaza, características observables, técnica de modelado y criterios de evaluación. Esta articulación permite comprender que la selección de un modelo ligero no debe basarse únicamente en métricas de precisión, sino en la relación entre desempeño, eficiencia y condiciones reales de despliegue.

El primer elemento corresponde al contexto operativo, que incluye el perfil del dispositivo, las restricciones de memoria y procesamiento, la conectividad disponible, el consumo energético y los requisitos de privacidad. El segundo elemento corresponde a la amenaza, entendida a partir del tipo de malware, el vector de ataque y los patrones observables que genera en el tráfico de red o en el comportamiento del sistema.

El tercer elemento está constituido por las características observables, como duración del flujo, tamaño de paquetes, volumen de bytes, uso de CPU, uso de memoria, número de procesos activos o relación entre paquetes enviados y recibidos. Estas variables permiten conectar el comportamiento malicioso con datos que pueden ser procesados por modelos ligeros.

El cuarto elemento corresponde a la estrategia de modelado, que puede incluir modelos clásicos optimizados, aprendizaje profundo comprimido, TinyML o aprendizaje federado, según el contexto de aplicación. Finalmente, el quinto elemento corresponde a los criterios de evaluación, los cuales integran métricas funcionales y métricas de eficiencia, tales como F1-score, accuracy, latencia, memoria, tamaño del modelo y consumo energético.

Este marco referencial integrado guía el análisis comparativo del estudio y permite justificar la selección de técnicas ligeras en función de la viabilidad de despliegue en dispositivos IoT y sistemas embebidos.

Relación con el problema e implicaciones

El problema de investigación exige soluciones capaces de equilibrar detección efectiva y uso eficiente de recursos. La literatura revisada sugiere que este equilibrio requiere priorizar características informativas de bajo costo, seleccionar modelos compactos, aplicar técnicas de compresión cuando el aprendizaje profundo aporte valor diferencial y considerar enfoques federados cuando la privacidad o el ancho de banda sean restricciones relevantes.

Desde esta perspectiva, el aporte del estudio consiste en organizar la evidencia disponible en criterios de decisión aplicables a entornos IoT. Esto permite pasar de una revisión descriptiva de técnicas hacia una interpretación orientada a la selección de modelos según amenaza, contexto, características observables y restricciones operativas. En consecuencia, la investigación no propone validar experimentalmente un modelo propio, sino consolidar un marco analítico que sirva como base para futuras implementaciones, evaluaciones en hardware real y estudios comparativos más controlados.

Diseño metodológico

Enfoque y diseño

El presente estudio se enmarca en una investigación de tipo monográfica con enfoque mixto analítico-comparativo.

Desde el componente cuantitativo, se desarrolla una revisión sistemática de literatura bajo las directrices PRISMA 2020 (Page et al., 2021), aplicando criterios explícitos de inclusión, exclusión y evaluación de calidad metodológica, así como la consolidación de métricas reportadas en los estudios seleccionados (F1-score, accuracy, tamaño del modelo, latencia, entre otras).

Desde el componente cualitativo, se realiza un análisis crítico e interpretativo de los hallazgos identificados, orientado a:

- Integrar resultados provenientes de distintas aproximaciones técnicas.
- Identificar brechas metodológicas en la evaluación de modelos ligeros.
- Formular un marco comparativo multicriterio aplicable a entornos IoT con restricciones operativas.

La comparación desarrollada se fundamenta en evidencia empírica previamente reportada en la literatura científica.

El periodo de análisis se delimitó entre 2018 y 2025 con el fin de incluir literatura reciente sobre detección de malware en IoT, aparición y maduración de datasets especializados, y desarrollo de técnicas de aprendizaje automático ligero orientadas a dispositivos con restricciones de memoria, procesamiento, latencia y energía. Este intervalo permite observar la transición desde modelos tradicionales de detección hacia enfoques optimizados, como selección de características, poda, cuantización, TinyML y aprendizaje federado.

Preguntas de investigación (PI)

A partir del objetivo general y de los objetivos específicos planteados, se formulan las siguientes preguntas de investigación, orientadas a guiar la revisión sistemática de literatura y el análisis comparativo de técnicas de aprendizaje automático ligero aplicadas a la detección de malware en dispositivos IoT y sistemas embebidos:

PI-1. ¿Cuáles son las principales técnicas de aprendizaje automático ligero reportadas en la literatura para la detección de malware en dispositivos IoT y sistemas embebidos?

Esta pregunta se relaciona con la caracterización del estado del arte y permite identificar enfoques como selección de características, poda estructural, cuantización, TinyML, modelos clásicos optimizados, aprendizaje profundo comprimido y aprendizaje federado.

PI-2. ¿Qué tipos de malware, vectores de ataque y patrones observables son más relevantes para la detección ligera en entornos IoT?

Esta pregunta permite vincular el fenómeno de ciberseguridad con las variables técnicas que pueden ser utilizadas por modelos ligeros, tales como tráfico de red, uso de CPU, consumo de memoria, duración de flujo, número de paquetes o relación entre paquetes enviados y recibidos.

PI-3. ¿Qué métricas de desempeño y eficiencia reporta la literatura para comparar modelos ligeros frente a enfoques tradicionales de detección?

Esta pregunta orienta el análisis comparativo de los estudios revisados, considerando métricas como accuracy, precision, recall, F1-score, latencia, consumo de memoria, tamaño del modelo y consumo energético.

PI-4. ¿Qué criterios permiten orientar la selección de técnicas ligeras de detección de malware según el contexto operativo, el tipo de amenaza y las restricciones computacionales del dispositivo IoT?

Esta pregunta se relaciona con la formulación del marco de aplicación, ya que permite integrar los hallazgos de la revisión en criterios de decisión para seleccionar modelos o técnicas según las condiciones reales de despliegue.

Protocolización

Se definió un protocolo previo (objetivos, PI, fuentes, criterios, extracción, evaluación de calidad y plan de síntesis). Se documentan las posibles desviaciones justificadas (p. ej., ajustes de cadenas o ampliación de bases). Opcionalmente, el protocolo puede registrarse (OSF/registro institucional).

Estrategia PICOC adoptada

- P (Population – Población): estudios científicos que aborden la detección de malware o intrusiones en entornos IoT y sistemas embebidos caracterizados por dispositivos con restricciones de memoria, procesamiento o energía.
- I (Intervention – Intervención): técnicas ligeras de aprendizaje automático y aprendizaje profundo orientadas a la reducción de complejidad computacional, tales como poda estructural (pruning), cuantización, selección de características, TinyML, destilación de conocimiento y aprendizaje federado.
- C (Comparison – Comparación): enfoques tradicionales no optimizados o comparación entre distintas técnicas ligeras reportadas en la literatura.
- O (Outcome – Resultados): métricas de eficacia predictiva (Accuracy, Precision, Recall, F1-score, AUC) y métricas de eficiencia computacional (latencia, memoria, tamaño del modelo y consumo energético, cuando sea reportado).
- C (Context – Contexto): escenarios IoT y edge computing, incluyendo redes de sensores, gateways, dispositivos basados en microcontroladores (MCU) y sistemas en chip (SoC).

Fuentes de información

Se consultaron bases de datos indexadas reconocidas en el ámbito de ingeniería y ciberseguridad, tales como:

- IEEE
- ScienceDirect
- SpringerLink
- ACM
- Wiley
- Scopus

La búsqueda se restringió a publicaciones entre 2018 y 2025, en idioma inglés y español. Este periodo se definió porque a partir de 2018 se observa una consolidación progresiva de investigaciones relacionadas con ciberseguridad en IoT, detección de malware mediante aprendizaje automático, disponibilidad de datasets especializados y técnicas de optimización orientadas a dispositivos con restricciones computacionales. Asimismo, el rango permite incluir literatura reciente sobre enfoques como selección de características, poda, cuantización, TinyML y aprendizaje federado, los cuales son relevantes para analizar la viabilidad de modelos ligeros en dispositivos IoT y sistemas embebidos. La delimitación temporal también busca mantener actualidad científica y evitar la inclusión de estudios desactualizados frente a la rápida evolución de las amenazas, arquitecturas IoT y técnicas de aprendizaje automático.

Cadenas de búsqueda

Se construyeron consultas booleanas adaptadas a cada base (título/resumen/keywords), en inglés y español, para el periodo 2018–2025. Ejemplo de cadena general utilizada:

```
("Internet of Things" OR IoT OR embedded) AND
(malware OR "intrusion detection" OR botnet) AND
("lightweight" OR pruning OR quantization OR "feature selection"
OR TinyML OR "model compression" OR "federated learning") AND
(machine learning OR "deep learning") AND
(detection OR classification)
```

Se ajustaron sinónimos/tesauros y campos según cada índice. En español se usaron equivalentes: “aprendizaje automático”, “cuantización”, “poda”, “aprendizaje federado”, “dispositivos embebidos”, etcétera.

Criterios de inclusión y exclusión

Criterios de Inclusión:

1. Publicaciones entre 2018 y 2025, incluyendo artículos revisados por pares, actas de congresos y capítulos académicos con evidencia empírica documentada.
2. Estudios centrados en detección de malware o intrusiones en entornos IoT, sistemas embebidos o NIDS/NIPS aplicados específicamente a IoT.
3. Investigaciones que empleen técnicas ligeras de aprendizaje automático o que analicen explícitamente la eficiencia computacional del modelo propuesto.
4. Reporte de métricas de eficacia predictiva (por ejemplo, F1-score, Accuracy, Precision, Recall o AUC) y descripción mínima del dataset o entorno de evaluación.

Criterios de Exclusión:

1. Estudios sin resultados empíricos documentados.
2. Enfoques puramente teóricos sin aplicación al dominio IoT o sistemas embebidos.
3. Publicaciones duplicadas o versiones preliminares de trabajos ya incluidos.
4. Investigaciones en dominios no relacionados con IoT, salvo que presenten experimentación en dispositivos con restricciones equivalentes y transferibilidad claramente argumentada.

Fases PRISMA de la RSL

El proceso de selección de estudios se desarrolló conforme a las cuatro etapas establecidas en PRISMA 2020 (Page et al., 2021):

1. Identificación: Se ejecutaron las cadenas de búsqueda definidas en todas las bases de datos seleccionadas. Los resultados fueron exportados en formatos estructurados (CSV y BibTeX) para su consolidación y depuración. Posteriormente, se eliminaron registros duplicados mediante revisión automática y verificación manual.
2. Cribado (screening): Se realizó la revisión de títulos y resúmenes conforme a los criterios de inclusión y exclusión previamente establecidos.
3. Elegibilidad: Los estudios preseleccionados fueron analizados a texto completo, aplicando de manera estricta los criterios metodológicos y de contenido, así como la matriz de evaluación de calidad.

4. Inclusión: El conjunto final de estudios incluidos en la síntesis fue determinado con base en la pertinencia temática, el cumplimiento de criterios metodológicos y el puntaje obtenido en la evaluación de calidad.

Resultados del proceso de selección

En la fase de identificación se recuperaron 137 registros provenientes de seis bases de datos indexadas: IEEE Xplore, ScienceDirect, SpringerLink, ACM Digital Library, Wiley Online Library y Scopus.

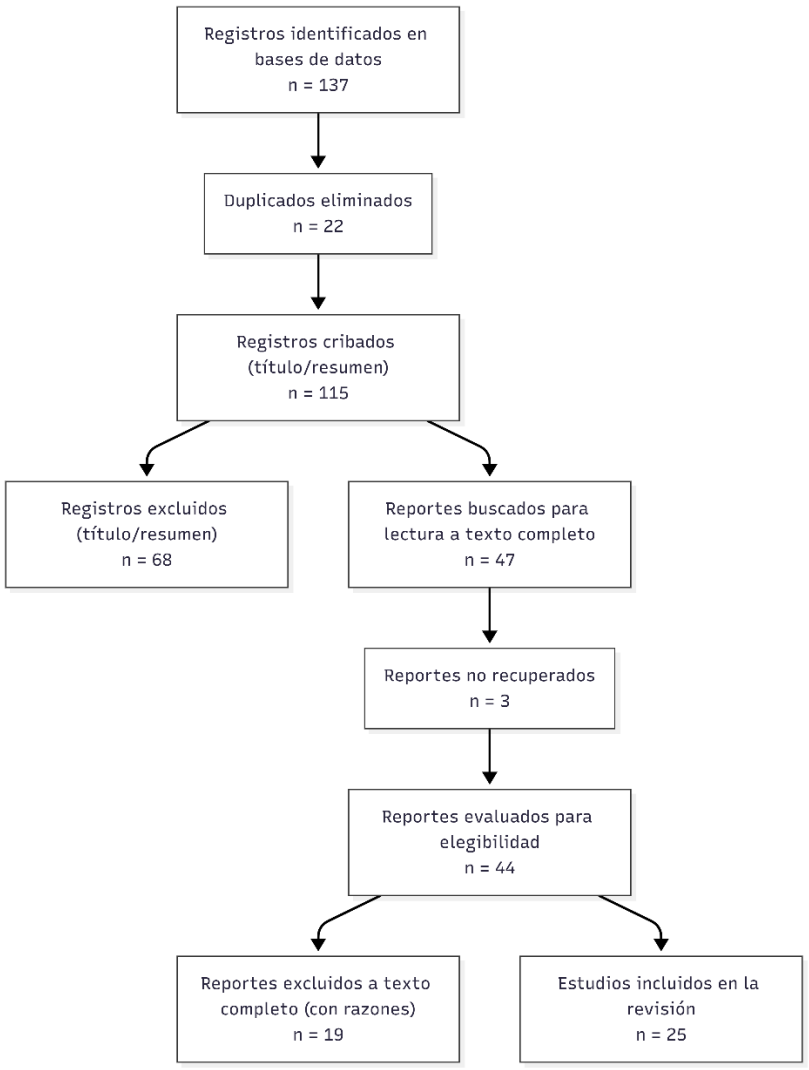
Tras la eliminación de duplicados, se procedió al cribado de títulos y resúmenes. Posteriormente, los estudios potencialmente relevantes fueron evaluados a texto completo.

Luego de aplicar los criterios de inclusión y exclusión, junto con la matriz de calidad metodológica, se incluyeron 25 artículos en la síntesis final.

El flujo completo del proceso se presenta en la Figura 1, conforme a la estructura PRISMA 2020.

Figura 1

Diagrama de flujo PRISMA de selección de estudios (2018–2025)



Nota. Adaptado de PRISMA 2020 (Page et al., 2021).

Extracción de datos

Se utilizó una matriz de extracción estandarizada (hoja de cálculo) con los siguientes campos: referencia completa, tipo de estudio, técnica ligera (poda, cuantización, selección de características, TinyML, aprendizaje federado), modelo base (RF, SVM, CNN, LSTM, entre otros), dataset(s) empleados (IoT-23, BoT-IoT, TON_IoT, CIC-MalMem-2022 u otros), métricas

de eficacia (Accuracy, F1-score, Precision, Recall, AUC), métricas de eficiencia reportadas (latencia, memoria, tamaño del modelo y consumo energético cuando esté disponible), configuración experimental descrita por los autores (hardware o entorno de simulación), hallazgos principales, limitaciones y amenazas a la validez.

Evaluación de calidad y riesgo de sesgo

Para valorar la calidad metodológica y el riesgo de sesgo de los estudios incluidos, se aplicó una lista de chequeo con diez criterios, calificados en una escala de 0 a 2 por ítem, para un puntaje máximo de 20 puntos. Esta lista fue adaptada de guías para revisiones sistemáticas en ingeniería de software y sistemas (Booth et al., 2016; Kitchenham & Charters, 2007). La Tabla 1 presenta los criterios utilizados para la evaluación.

Tabla 1
Lista de chequeo para evaluación de calidad y riesgo de sesgo de los estudios incluidos

Criterio	Descripción	Escala
Claridad del objetivo y preguntas	Evalúa si el estudio define con precisión su objetivo, pregunta de investigación o propósito de detección.	0–2
Descripción del dataset o escenario	Verifica si el estudio reporta el dataset, entorno experimental o escenario utilizado, así como información suficiente para interpretar su reproducibilidad.	0–2
Métricas de eficacia	Revisa si el estudio presenta métricas comparables como accuracy, precision, recall, F1-score, AUC, FPR o FNR.	0–2
Métricas de eficiencia	Evalúa si el estudio reporta o justifica métricas como latencia, memoria, tamaño del modelo, consumo energético o costo computacional.	0–2
Baseline y comparación	Verifica si el estudio compara el enfoque analizado con modelos tradicionales, baselines u otros métodos relevantes.	0–2

Criterio	Descripción	Escala
Hiperparámetros y configuración	Revisa si el estudio describe parámetros, configuración experimental, arquitectura, entrenamiento o condiciones de evaluación.	0–2
Amenazas a la validez	Evalúa si el estudio identifica limitaciones metodológicas, técnicas o amenazas a la validez de sus resultados.	0–2
Replicabilidad	Verifica si el estudio ofrece elementos que faciliten replicación, como código, datos, semillas, parámetros o descripción suficiente del procedimiento.	0–2
Discusión de limitaciones	Revisa si el estudio reconoce limitaciones relacionadas con generalización, datasets, hardware, tráfico, escenarios o aplicabilidad.	0–2
Coherencia entre conclusiones y evidencia	Evalúa si las conclusiones del estudio están sustentadas en los resultados y métricas reportadas.	0–2

Nota. Cada criterio se calificó en una escala de 0 a 2, donde 0 = no cumple, 1 = cumple parcialmente y 2 = cumple completamente. El puntaje máximo posible fue de 20 puntos. La lista fue elaborada a partir de Booth et al. (2016) y Kitchenham & Charters (2007).

El puntaje obtenido se utilizó como criterio de sensibilidad para interpretar la solidez de la evidencia incluida en la revisión. En particular, permitió identificar estudios con mayor o menor calidad metodológica y valorar su peso relativo dentro del análisis comparativo.

Síntesis y análisis

Dada la heterogeneidad de técnicas, datasets y métricas, se realizó una síntesis narrativa y tabular con análisis temático por ejes:

1. Optimización estructural (poda, quantization, selección),
2. Entrenamiento distribuido (federado),

3. Despliegue embebido/TinyML, y
4. Trade-offs rendimiento–eficiencia.

- Se calcularon estadísticos descriptivos (mediana y rango intercuartílico) únicamente en aquellos subconjuntos de estudios que reportaron métricas numéricas homogéneas y comparables bajo condiciones metodológicas similares (mismo tipo de métrica y descripción explícita del dataset). En los casos donde no fue posible establecer comparabilidad directa, se optó por síntesis narrativa y análisis cualitativo estructurado.
- Se incluyeron tablas comparativas y figuras (barras/radar) para visualizar F1/Accuracy vs. latencia/memoria/energía.
- Dada la heterogeneidad observada en las técnicas, datasets, métricas y configuraciones experimentales reportadas, no fue metodológicamente apropiado realizar un meta-análisis cuantitativo. En su lugar, se aplicó una estrategia de síntesis narrativa estructurada complementada con vote-counting matizado, considerando la dirección y consistencia de los efectos reportados entre estudios.

Gestión de sesgos y validez

Se discutió el sesgo de publicación, sesgo de selección (bases/idiomas), sesgo de medición (definición de métricas) y sesgo de confusión (diferencias de hardware/entorno). Se documentó cómo afectan la interpretación y generalización de resultados.

Consideraciones éticas

No se procesan datos personales. Se respetan licencias de datasets y publicaciones, citando las fuentes y evitando reproducir material protegido más allá de lo permitido.

Reproducibilidad y open materials

Los materiales asociados al protocolo de revisión (cadenas de búsqueda completas, matriz de extracción y evaluación de calidad) se disponen en los apéndices, garantizando la trazabilidad y reproducibilidad del estudio.

Estado del arte sobre las técnicas de aprendizaje automático ligero empleadas para la detección de malware en dispositivos IoT y sistemas embebidos

Este apartado desarrolla el primer objetivo específico del estudio, orientado a caracterizar el estado del arte sobre las técnicas de aprendizaje automático ligero empleadas para la detección de malware en dispositivos IoT y sistemas embebidos. Para ello, se sintetizan los principales enfoques identificados en la revisión sistemática de literatura, las limitaciones de los métodos tradicionales, las técnicas de optimización más recurrentes y las condiciones que inciden en la viabilidad de despliegue en dispositivos con restricciones de memoria, procesamiento, energía y latencia.

Propósito y enfoque

Este capítulo caracteriza, con base en la revisión sistemática de literatura realizada para el periodo 2018–2025, por qué los métodos tradicionales de detección, como firmas, heurísticas estáticas y detección centralizada, presentan limitaciones cuando se trasladan a nodos de borde y dispositivos embebidos. Asimismo, sintetiza las estrategias ligeras y distribuidas que han sido reportadas en la literatura para cumplir objetivos de detección bajo restricciones de CPU, memoria, energía, latencia y privacidad (Moustafa et al., 2020; Sharmila & Nagapadma, 2023; Rey et al., 2021).

Procedimiento de análisis del estado del arte

Para el desarrollo de este apartado se aplicó un análisis temático sobre los 25 estudios incluidos en la revisión sistemática de literatura. La información fue organizada mediante una matriz de extracción y comparación, en la cual se registraron autor, año, técnica analizada, tipo de modelo, dataset o escenario empleado, métricas de desempeño, métricas de eficiencia, limitaciones reportadas y aporte al objetivo de la investigación. Adicionalmente, se utilizó la lista de chequeo de calidad metodológica descrita previamente, con el fin de valorar la solidez de los estudios y reducir el riesgo de sesgo en la interpretación de los hallazgos.

A partir de esta matriz, los estudios fueron codificados en categorías temáticas relacionadas con barreras técnicas de los enfoques tradicionales, técnicas de optimización del modelo, uso de datasets especializados, métricas de evaluación y condiciones de despliegue en entornos IoT. Esta codificación permitió identificar patrones recurrentes en la literatura y establecer una base comparativa para analizar la viabilidad de los modelos ligeros en dispositivos con recursos limitados.

Resultados: limitaciones técnicas recurrentes en IoT

- Restricciones de cómputo y memoria. Motores de firmas/inspección profunda y modelos densos requieren RAM/flash y ciclos de CPU que los MCUs/SoC de bajo consumo no sostienen de forma continua, elevando latencias o forzando muestreo eventual (Javed et al.,

2024; Sharmila & Nagapadma, 2023). Implicación: modelos comprimidos (poda, cuantización) o clasificadores clásicos optimizados con selección de características.

- Consumo energético y continuidad operativa. La inspección y el envío continuo de trazas degradan la autonomía de nodos a batería; varios estudios reportan reducciones sensibles del tiempo de operación sin optimización (Javed et al., 2024). Implicación: inferencias por eventos, cuantización (int8/float16) y presupuesto energético explícito.
- Dependencia de firmas y evasión polimórfica. La diversidad de arquitecturas/protocolos IoT y variantes ofuscadas o en memoria superan la cobertura de firmas y complican actualizaciones (Garcia et al., 2020; Canadian Institute for Cybersecurity, 2022). Implicación: detección por comportamiento y modelos de anomalía.
- Sesgo de datos y pobre generalización. Modelos entrenados en tráfico TI fallan ante patrones M2M/protocolos ligeros; aumentan falsos positivos o disminuye el recall (Moustafa et al., 2020). Implicación: curaduría de datasets IoT y validación cruzada entre dominios/dispositivos.
- Latencia de inferencia vs. requisitos near real-time. Pipelines pesados o centralizados no cumplen umbrales de respuesta industrial o clínica; TinyML habilita latencias sub-segundo en el dispositivo (Warden & Situnayake, 2019). Implicación: inferencia local prioritaria; la nube para agregación/mejora del modelo.
- Sobrecarga de comunicación y privacidad. Centralizar datos eleva tráfico y riesgo de exposición; el aprendizaje federado reduce transmisión preservando desempeño, con retos de no-IID y sincronización (Rey et al., 2021). Implicación: FL con compresión de actualizaciones y agregación robusta.

- Heterogeneidad de hardware y stacks. Portar detectores a MCU/RTOS/protocolos diversos requiere reingeniería (Fenanir et al., 2019; Javed et al., 2024). Implicación: pipeline modular y targets definidos (Cortex-M, RISC-V).
- Falta de validación en hardware real y estandarización. Predominan evaluaciones con datasets públicos y simulación; faltan protocolos comparables de energía/memoria/latencia en dispositivo (Rey et al., 2021). Implicación: banco de pruebas físico y pauta mínima de métricas.

A partir de la evaluación realizada, se observa que la mayoría de los estudios presentan una alta calidad metodológica, especialmente en la definición del problema y el uso de datasets representativos del entorno IoT. Sin embargo, se identifican limitaciones recurrentes en la reproducibilidad de los experimentos y en el reporte estandarizado de métricas de eficiencia, lo cual impacta la comparabilidad entre enfoques y justifica la necesidad de un análisis crítico complementario. En la siguiente tabla se presenta un extracto representativo de la evaluación realizada, donde se evidencia el puntaje obtenido por cada estudio en función de los criterios definidos.

Tabla 2

Evaluación de calidad de los estudios incluidos

Estudio	Claridad del objetivo	Dataset / escenario	Métricas reportadas	Reproducibilidad	Puntaje total
Estudio 1 (Meidan et al., 2018)	2	2 (N-BaIoT)	2 (accuracy, recall, F1-score)	2	8
Estudio 2 (Koroniotis et al., 2019)	2	2 (BoT-IoT)	2 (accuracy, recall, F1-score)	2	8
Estudio 3 (Moustafa et al., 2020)	2	2 (TON_IoT)	2 (accuracy, F1-score, AUC)	2	8
Estudio 4 (Doshi et al., 2018)	2	1 (escenario IoT limitado)	2 (accuracy)	1	6

Estudio	Claridad del objetivo	Dataset / escenario	Métricas reportadas	Reproducibilidad	Puntaje total
Estudio 5 (Fenanir et al., 2019)	2	2 (KDD99, NSL-KDD, UNSW-NB15)	2 (accuracy, precision, recall, F1-score)	2	8
Estudio 6 (Javed et al., 2024)	2	2 (dispositivos IoT de hogar inteligente)	2 (accuracy, precision, recall, F1-score)	2	8
Estudio 7 (Sharmila & Nagapadma, 2023)	2	2 (RT-IoT2022)	2 (accuracy, precision, recall, F1-score)	2	8
Estudio 8 (Rey et al., 2021)	2	2 (N-BaIoT / escenario de aprendizaje federado)	2 (accuracy, F1-score y métricas comparativas)	1	7

Nota. La evaluación se realizó en una escala de 0 a 2 por criterio, donde 0 = no cumple, 1 = cumple parcialmente y 2 = cumple completamente. El puntaje máximo posible para esta tabla resumida es de 8 puntos. Elaboración propia a partir de los criterios de calidad definidos para la revisión sistemática de literatura.

En el desarrollo del estado del arte se realizó una síntesis temática de los 25 estudios incluidos en la revisión sistemática de literatura correspondiente al periodo 2018–2025, siguiendo los lineamientos PRISMA 2020 y la estrategia PICOC como criterios de trazabilidad y reproducibilidad (Page et al., 2021). A partir de la codificación de los hallazgos reportados por la literatura, se identificaron limitaciones recurrentes de los enfoques tradicionales de detección en entornos IoT, tales como restricciones de CPU y memoria, consumo energético, sesgo de datos, latencia y baja capacidad de generalización. Estas limitaciones son coherentes con estudios que advierten la necesidad de evaluar los modelos de detección no solo por su desempeño predictivo,

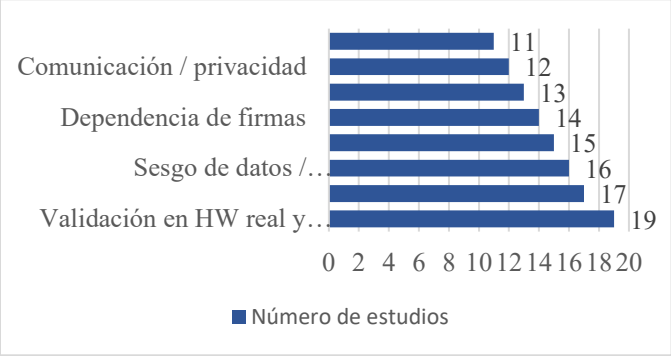
sino también por su viabilidad computacional en dispositivos con recursos limitados (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

La Figura 2 resume la frecuencia con que dichas limitaciones aparecen en la literatura analizada. Su lectura permite evidenciar la importancia de la validación en hardware real, la estandarización de métricas de eficiencia y la comparación bajo condiciones reproducibles. Por su parte, la Tabla 2 presenta una matriz de síntesis que relaciona cada limitación con su implicación práctica en entornos IoT, las métricas pertinentes y la evidencia bibliográfica asociada. Asimismo, la Tabla 3 consolida los datasets y métricas más empleadas en estudios sobre detección de malware e intrusiones en IoT, lo que facilita identificar insumos de comparación y criterios de evaluación alineados con la viabilidad de despliegue en dispositivos IoT y sistemas embebidos (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022).

La Figura 2 presenta la frecuencia con que las principales limitaciones técnicas fueron reportadas en los estudios incluidos en la revisión sistemática. Esta representación permite identificar cuáles restricciones aparecen con mayor recurrencia en la literatura y, por tanto, cuáles deben considerarse con mayor atención al evaluar la viabilidad de modelos ligeros para detección de malware en dispositivos IoT y sistemas embebidos.

Figura 2

Frecuencia de limitaciones técnicas reportadas en IoT (2018–2025; n = 25)



Nota. La codificación fue múltiple, por lo que un mismo estudio puede contribuir a varias categorías. Elaboración propia a partir de la revisión sistemática de literatura.

Como se observa en la Figura 2, la validación en hardware real, la estandarización de métricas y la dependencia de enfoques tradicionales de detección constituyen limitaciones recurrentes en la literatura analizada. Estos hallazgos evidencian que la discusión sobre modelos ligeros no debe centrarse únicamente en la exactitud del modelo, sino también en su factibilidad de despliegue, consumo de recursos y capacidad de generalización. Con el fin de profundizar esta relación, la Tabla 3 sintetiza las principales limitaciones identificadas, su implicación práctica en entornos IoT, las métricas asociadas y la evidencia bibliográfica que respalda cada categoría.

Tabla 3

Síntesis: limitación → implicación → métricas → evidencia

Limitación	Implicación en IoT	Métricas asociadas	Evidencia (APA)
CPU/RAM limitadas	Latencia alta/no ejecución continua; compresión del modelo	F1, Latencia, Memoria	Sharmila & Nagapadma (2023)
Energía y autonomía	Degradación de autonomía en nodos a batería	Energía, Latencia	Javed et al. (2024)
Dependencia de firmas	Evasión por polimorfismo; actualización costosa	FPR/FNR, Exactitud	Garcia et al. (2020); Canadian Institute for Cybersecurity (2022)

Limitación	Implicación en IoT	Métricas asociadas	Evidencia (APA)
Sesgo/generalización	FPR altos; bajo recall	FPR, Recall, F1	Moustafa et al. (2020)
Inferencia no local	Retardos; dependencia de red	Latencia, Disponibilidad	Warden & Situnayake (2019)
Comunicación/privacidad	Sobrecarga y riesgo de exposición	Ancho de banda, Exactitud	Rey et al. (2021)
Heterogeneidad HW/protocolos	Portabilidad/mantenimiento	—	Fenanir et al. (2019)
Estandarización/validación	Comparabilidad limitada	Latencia, Energía, Memoria	Rey et al. (2021)

Nota. Elaboración propia

La Tabla 3 permite reconocer que las limitaciones técnicas no actúan de manera aislada, sino que se relacionan con decisiones metodológicas y operativas. Por ejemplo, las restricciones de CPU y memoria afectan la selección del modelo; la dependencia de firmas limita la detección de variantes nuevas; y la falta de validación en hardware real reduce la posibilidad de trasladar los resultados a escenarios IoT reales. Para asegurar comparabilidad con el estado del arte, los estudios revisados emplean conjuntos de datos ampliamente utilizados en investigaciones recientes sobre detección de intrusiones en IoT, entre ellos IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022. Estos datasets permiten contrastar el desempeño de los modelos en escenarios heterogéneos y bajo diferentes perfiles de tráfico, como se sintetiza en la Tabla 4.

Tabla 4

Datasets y métricas más utilizadas en los estudios

Dataset	Descripción breve	Referencias
IoT-23	Tráfico IoT etiquetado benigno/malicioso; escenarios múltiples	Garcia et al. (2020)
BoT-IoT	Tráfico legítimo/botnet orientado a IDS IoT	Koroniotis et al. (2019)

TON_IoT	Telemetría realista para IoT/IIoT; múltiples fuentes	Moustafa et al. (2020)
CIC-MalMem-2022	Malware en memoria; variantes ofuscadas	Canadian Institute for Cybersecurity (2022)

Nota. Elaboración propia a partir de la revisión sistemática de literatura. La tabla sintetiza los datasets identificados como referentes en estudios sobre detección de malware e intrusiones en entornos IoT y sistemas relacionados. Estos conjuntos de datos se emplean como insumos de comparación reportados por la literatura, no como resultados experimentales propios.

La Tabla 4 evidencia que los datasets IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022 concentran una parte importante de las evaluaciones reportadas en la literatura reciente. Sin embargo, su uso no garantiza por sí mismo la generalización de los resultados, dado que cada conjunto de datos responde a condiciones particulares de captura, tipo de tráfico, amenazas representadas y escenarios de evaluación. Por ello, la comparación entre modelos ligeros debe considerar no solo el dataset utilizado, sino también las métricas reportadas, la reproducibilidad del estudio y la cercanía del escenario experimental con dispositivos IoT reales.

Síntesis de respuestas a las preguntas de investigación

La revisión sistemática de literatura y el análisis comparativo de los 25 estudios incluidos permitieron responder las cuatro preguntas de investigación planteadas en el diseño metodológico. Esta síntesis articula los hallazgos principales sobre técnicas de aprendizaje automático ligero, tipos de malware, vectores de ataque, patrones observables, métricas de evaluación y criterios de selección aplicables a dispositivos IoT y sistemas embebidos. En

coherencia con los lineamientos PRISMA 2020, las respuestas se derivan de la evidencia identificada, filtrada y analizada durante el proceso de revisión sistemática (Page et al., 2021).

PI-1. ¿Cuáles son las principales técnicas de aprendizaje automático ligero reportadas en la literatura para la detección de malware en dispositivos IoT y sistemas embebidos?

La literatura revisada muestra que las técnicas más recurrentes se agrupan en modelos clásicos optimizados mediante selección de características, modelos de aprendizaje profundo comprimidos, poda estructural, cuantización, TinyML y aprendizaje federado. Estos enfoques buscan reducir la complejidad computacional sin comprometer de manera significativa la capacidad predictiva, especialmente en escenarios donde existen restricciones de CPU, memoria, latencia y energía. En esta línea, los estudios sobre detección de intrusiones en IoT resaltan que la viabilidad de un modelo no depende únicamente de métricas de clasificación, sino también de su posibilidad de operar en dispositivos con recursos limitados (Fenanir et al., 2019; Javed et al., 2024). De manera complementaria, la cuantización y los enfoques orientados al borde han sido reportados como alternativas pertinentes para reducir consumo de memoria y costo de inferencia en dispositivos restringidos (Sharmila & Nagapadma, 2023).

PI-2. ¿Qué tipos de malware, vectores de ataque y patrones observables son más relevantes para la detección ligera en entornos IoT?

Los estudios incluidos evidencian que las amenazas más relevantes en IoT se relacionan con botnets, ransomware, cryptojacking, backdoors, malware en memoria y comunicaciones de comando y control. Estas amenazas suelen explotar vectores como credenciales débiles, servicios expuestos, protocolos inseguros, puertos abiertos y mecanismos de actualización vulnerables. Desde la perspectiva de los modelos ligeros, los patrones más útiles son aquellos que pueden observarse con bajo costo computacional, como duración del flujo, número de paquetes, volumen de bytes, relación entre paquetes enviados y recibidos, uso de CPU, consumo de memoria y número de procesos activos. La disponibilidad de datasets orientados a tráfico IoT, botnets, telemetría y malware en memoria ha facilitado la identificación de estos patrones en la literatura reciente (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022).

PI-3. ¿Qué métricas de desempeño y eficiencia reporta la literatura para comparar modelos ligeros frente a enfoques tradicionales de detección?

La literatura reporta métricas funcionales como accuracy, precision, recall, F1-score, FPR, FNR y AUC, las cuales permiten valorar la capacidad del modelo para diferenciar tráfico benigno y malicioso. Sin embargo, en dispositivos IoT estas métricas resultan insuficientes si no se complementan con indicadores de eficiencia computacional, como latencia, consumo de memoria, tamaño del modelo y consumo energético. Por ello, los estudios revisados sugieren una evaluación multicriterio, en la que un modelo no se considere adecuado únicamente por su exactitud, sino también por su capacidad de operar en hardware restringido. Esta perspectiva es coherente con investigaciones que advierten la necesidad de evaluar la detección en IoT desde el

equilibrio entre eficacia predictiva y eficiencia operativa (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

PI-4. ¿Qué criterios permiten orientar la selección de técnicas ligeras de detección de malware según el contexto operativo, el tipo de amenaza y las restricciones computacionales del dispositivo IoT?

El análisis permitió establecer que la selección de una técnica ligera debe considerar, de manera integrada, cinco criterios: el contexto operativo del dispositivo, el tipo de amenaza, las características observables disponibles, la técnica de optimización aplicable y las métricas de evaluación. En este sentido, la decisión sobre el modelo no debe basarse únicamente en su desempeño predictivo, sino en su factibilidad de despliegue. Así, técnicas como selección de características, poda, cuantización, TinyML o aprendizaje federado deben elegirse según las condiciones de memoria, procesamiento, conectividad, privacidad, latencia y consumo energético del entorno IoT. En particular, el aprendizaje federado resulta pertinente cuando la privacidad y la descentralización de los datos son restricciones relevantes, aunque su adopción exige considerar heterogeneidad de dispositivos, costos de comunicación y estabilidad del entrenamiento distribuido (Nguyen et al., 2021; Rey et al., 2021).

En conjunto, las respuestas a las preguntas de investigación evidencian que la detección de malware en entornos IoT exige un enfoque de análisis multicriterio. La revisión muestra que los modelos ligeros representan una alternativa viable cuando logran integrar capacidad predictiva, eficiencia computacional y adecuación al contexto de despliegue. Por tanto, el aporte

del estudio no se limita a identificar técnicas existentes, sino que organiza la evidencia revisada en criterios de decisión útiles para orientar futuras implementaciones, validaciones en hardware real y estudios comparativos en dispositivos IoT y sistemas embebidos.

Evaluación de calidad metodológica de los estudios incluidos

Con el fin de asegurar la rigurosidad metodológica de los estudios incluidos en la revisión sistemática, se aplicó una matriz de evaluación de calidad y riesgo de sesgo adaptada de criterios utilizados en revisiones sistemáticas en ingeniería de software (Booth et al., 2016; Kitchenham & Charters, 2007). Esta matriz permitió valorar cada estudio en diez dimensiones: claridad del objetivo, descripción del dataset o escenario, métricas de eficacia, métricas de eficiencia, existencia de baseline o comparación, reporte de parámetros, amenazas a la validez, replicabilidad, discusión de limitaciones y coherencia entre conclusiones y evidencia.

Cada criterio fue calificado en una escala de 0 a 2, donde 0 indica que el criterio no se cumple, 1 que se cumple parcialmente y 2 que se cumple completamente. De esta manera, el puntaje máximo posible por estudio fue de 20 puntos. La estructura general de la matriz se presenta en la Tabla 1, mientras que su aplicación completa a los 25 estudios incluidos se detalla en el Apéndice B.

La aplicación de esta matriz permitió clasificar los estudios según su calidad metodológica y estimar su nivel de confiabilidad para la síntesis comparativa. Los estudios con mayor puntaje fueron considerados como evidencia más sólida para el análisis de técnicas ligeras, mientras que aquellos con menor puntuación fueron interpretados con cautela,

especialmente cuando presentaban limitaciones de reproducibilidad, ausencia de métricas de eficiencia o escasa discusión de amenazas a la validez.

La revisión sistemática permitió identificar que las técnicas de optimización más frecuentes en modelos ligeros para la detección de malware en IoT se agrupan en selección de características, poda estructural, cuantización, TinyML y aprendizaje federado. Estas técnicas buscan reducir la complejidad computacional del modelo, disminuir el tamaño requerido para su despliegue, mejorar los tiempos de inferencia y limitar el consumo de memoria o energía, manteniendo un desempeño aceptable en métricas como *accuracy*, *precision*, *recall* y *F1-score*. Esta perspectiva es coherente con los estudios sobre detección en IoT que advierten que los modelos deben evaluarse no solo por su exactitud, sino también por su viabilidad de operación en dispositivos con recursos restringidos (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

Asimismo, la literatura reciente destaca que el aprendizaje federado constituye una alternativa pertinente cuando la privacidad, la descentralización de los datos y la reducción de transferencia de información son condiciones relevantes para el despliegue de soluciones de detección en entornos IoT distribuidos. Sin embargo, este enfoque también presenta retos asociados a heterogeneidad de dispositivos, sobrecarga de comunicación, estabilidad del entrenamiento y exposición a posibles ataques contra el proceso de agregación del modelo (Nguyen et al., 2021; Rey et al., 2021). Por tanto, la selección de una técnica ligera no debe basarse únicamente en el rendimiento predictivo, sino en una evaluación multicriterio que integre

tipo de amenaza, características observables, dataset empleado, restricciones de hardware, latencia, memoria, consumo energético y condiciones de privacidad.

En consecuencia, la evidencia revisada permite afirmar que la adopción de modelos ligeros en IoT requiere equilibrar desempeño y eficiencia operativa. Una reducción moderada en métricas de clasificación puede ser aceptable cuando permite ejecutar el modelo en dispositivos embebidos, nodos de borde o escenarios con conectividad limitada. Esta lectura resulta relevante para el presente estudio, porque desplaza el análisis desde la comparación aislada de modelos hacia la identificación de criterios técnicos para orientar futuras implementaciones y validaciones en hardware real.

Implicaciones para el marco de aplicación

A partir de la revisión sistemática, se derivan criterios técnicos para orientar la selección de técnicas ligeras de detección de malware en dispositivos IoT y sistemas embebidos. En primer lugar, la literatura revisada sugiere priorizar modelos compactos y técnicas de reducción de complejidad, como selección de características, poda estructural y cuantización, cuando el entorno presenta restricciones de memoria, procesamiento o latencia (Javed et al., 2024; Sharmila & Nagapadma, 2023). En segundo lugar, la inferencia en el dispositivo mediante enfoques TinyML resulta pertinente cuando se requiere respuesta local, menor dependencia de la nube y reducción de la exposición de datos durante la transmisión (Warden & Situnayake, 2019).

Asimismo, el aprendizaje federado constituye una alternativa relevante en escenarios donde la privacidad, la descentralización de los datos y la reducción de transferencia de información son condiciones críticas para el despliegue de soluciones de detección en IoT. No obstante, su adopción debe valorarse con cautela debido a retos asociados con heterogeneidad de dispositivos, sobrecarga de comunicación y estabilidad del entrenamiento distribuido (Nguyen et al., 2021; Rey et al., 2021). Finalmente, la evidencia revisada indica que la evaluación de estos enfoques debe integrar métricas de desempeño, como *accuracy*, *precision*, *recall* y *F1-score*, junto con métricas de eficiencia, como latencia, consumo de memoria, tamaño del modelo y consumo energético. Esta evaluación multicriterio permite valorar la viabilidad real de despliegue en entornos IoT, más allá de la precisión alcanzada en datasets públicos como IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022 (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022).

Síntesis parcial del estado del arte

Las restricciones estructurales de los dispositivos IoT, especialmente en CPU, memoria y consumo energético, junto con la naturaleza máquina a máquina del tráfico, explican por qué los enfoques tradicionales de detección pueden degradarse cuando se trasladan a escenarios distribuidos y de bajo recurso. La literatura revisada muestra que la selección de características, la compresión de modelos, TinyML y el aprendizaje federado constituyen alternativas técnicas relevantes para equilibrar desempeño predictivo y eficiencia computacional en estos entornos (Fenanir et al., 2019; Javed et al., 2024; Nguyen et al., 2021; Sharmila & Nagapadma, 2023; Warden & Situnayake, 2019). No obstante, la transferibilidad de los resultados exige fortalecer la

validación en dispositivos reales y estandarizar el reporte de métricas como latencia, consumo de memoria, tamaño del modelo y consumo energético, de manera que la comparación entre estudios sea metodológicamente más consistente.

Tipos de malware, vectores de ataque y patrones observables en IoT y sistemas embebidos

Este apartado desarrolla la caracterización de las amenazas de malware más relevantes en dispositivos IoT y sistemas embebidos, considerando su relación con los vectores de ataque y los patrones observables que pueden ser útiles para una detección ligera. La revisión sistemática permitió identificar que, en estos entornos, el análisis del malware no debe limitarse a su clasificación por familia, sino que debe relacionarse con las señales que deja en el tráfico de red, el comportamiento del sistema y el consumo de recursos. Esta perspectiva resulta pertinente porque los modelos ligeros dependen de características de bajo costo computacional, tales como duración de conexión, número de paquetes, volumen de bytes, frecuencia de consultas, uso de CPU, memoria y procesos activos.

La literatura revisada muestra que los datasets IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022 han sido utilizados para estudiar tráfico benigno y malicioso, botnets, telemetría, intrusiones y malware en memoria, lo que permite identificar patrones técnicos recurrentes en escenarios IoT y sistemas relacionados (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022). Estos conjuntos de datos no se emplean en este estudio para ejecutar experimentos propios, sino como referentes reportados por la literatura para sustentar la identificación de amenazas, vectores y variables observables.

Taxonomía de malware predominante y señales típicas

La literatura revisada permite agrupar las amenazas más relevantes en dispositivos IoT y sistemas embebidos en cinco categorías principales: botnets, ransomware o lockers, cryptojacking, malware en memoria y backdoors asociados a firmware o actualizaciones inseguras. Estas categorías se seleccionan por su recurrencia en estudios sobre seguridad IoT, su impacto operativo y la posibilidad de generar señales observables que pueden ser aprovechadas por modelos ligeros de detección.

Las botnets constituyen una de las amenazas más representativas en entornos IoT, debido a que automatizan procesos de escaneo, explotación de credenciales débiles y ejecución de ataques distribuidos de denegación de servicio. En datasets como BoT-IoT e IoT-23, este tipo de amenaza se asocia con ráfagas de conexiones salientes, aumento de paquetes SYN, tráfico HTTP repetitivo o anómalo y comunicación hacia múltiples destinos o servidores de comando y control (Garcia et al., 2020; Koroniotis et al., 2019). Estas señales son relevantes para modelos ligeros porque pueden observarse mediante variables de tráfico sin requerir inspección profunda del contenido.

El ransomware y los lockers orientados a entornos de borde, dispositivos NAS o servicios expuestos representan una amenaza crítica por su impacto sobre la disponibilidad e integridad de la información. En escenarios IoT, estas amenazas pueden relacionarse con abuso de servicios expuestos, configuraciones débiles, tráfico anómalo hacia recursos compartidos, picos de lectura o escritura y cambios abruptos en el comportamiento del

sistema. Para la detección ligera, estos patrones pueden asociarse con variables de comportamiento del host, actividad de procesos, uso de memoria y variaciones inusuales en el tráfico.

El cryptojacking se relaciona con el uso no autorizado de recursos computacionales para minería de criptomonedas. En dispositivos IoT y sistemas embebidos, esta amenaza puede manifestarse mediante consumo sostenido de CPU, incremento en el uso de memoria, conexiones persistentes hacia servicios externos y tráfico periódico asociado a procesos no autorizados. Desde la perspectiva de los modelos ligeros, estas señales son útiles porque permiten vincular el comportamiento malicioso con métricas de bajo costo computacional, como uso de CPU, consumo energético estimado, persistencia de conexiones y actividad anómala de procesos.

El malware en memoria representa una amenaza relevante porque puede operar con menor dependencia de archivos persistentes, dificultando su detección mediante mecanismos tradicionales basados en firmas. En este tipo de amenaza, los patrones observables pueden estar asociados con llamadas al sistema, procesos efímeros, variaciones en memoria y cambios anómalos en el comportamiento del host. Datasets como CIC-MalMem-2022 permiten estudiar este tipo de comportamiento y aportar evidencia útil para modelos de detección basados en características de memoria y sistema (Canadian Institute for Cybersecurity, 2022).

Finalmente, los backdoors y el firmware comprometido representan riesgos importantes en IoT cuando los procesos de actualización no incorporan mecanismos adecuados de verificación, firma o control de integridad. Este tipo de amenaza puede generar eventos anómalos durante actualizaciones OTA, cambios no autorizados de configuración, variaciones de hash y conexiones posteriores hacia infraestructura de comando y control. En este sentido, su detección exige relacionar eventos del sistema, tráfico de red y comportamiento persistente del dispositivo.

En conjunto, esta taxonomía permite establecer una relación entre familia de malware, vector de ataque y patrón observable. Esta relación es fundamental para el estudio, dado que la detección ligera no depende únicamente del algoritmo utilizado, sino de la capacidad para seleccionar características que sean representativas del comportamiento malicioso y, al mismo tiempo, viables de medir en dispositivos con restricciones de procesamiento, memoria, energía y latencia.

Vectores de ataque y superficies expuestas

La revisión sistemática permitió identificar que los vectores de ataque en entornos IoT no siempre son reportados con frecuencias homogéneas entre los estudios, debido a diferencias en los datasets, escenarios de evaluación, protocolos analizados y tipos de amenaza considerados. Por esta razón, en lugar de asignar porcentajes no comparables, los vectores se organizaron según su criticidad técnica, considerando cuatro criterios: exposición del servicio, facilidad de

explotación, impacto potencial sobre el dispositivo y posibilidad de generar patrones observables útiles para modelos ligeros de detección.

En la literatura revisada, los servicios expuestos, las credenciales débiles, los protocolos sin cifrado y las actualizaciones inseguras aparecen como condiciones recurrentes que amplían la superficie de ataque en dispositivos IoT. Estos factores son especialmente relevantes porque pueden facilitar la propagación de botnets, el abuso de servicios remotos, la interceptación de comunicaciones o la modificación no autorizada del comportamiento del dispositivo (Garcia et al., 2020; Koroniotis et al., 2019). La Tabla 5 presenta una priorización de estos vectores desde una perspectiva técnica, sin asumir que representan una frecuencia estadística uniforme en todos los estudios revisados.

Tabla 5

Vectores de ataque priorizados según criticidad técnica en entornos IoT

Nivel de criticidad	Vector de ataque	Superficie expuesta	Patrón observable relevante	Justificación técnica
Alta	Credenciales débiles o por defecto	Interfaces de administración, Telnet, SSH, HTTP	Intentos repetidos de autenticación, conexiones fallidas, tráfico hacia múltiples destinos	Facilitan el acceso no autorizado y la propagación automatizada de botnets en dispositivos IoT.
Alta	Servicios y puertos expuestos	Telnet, SSH, HTTP, FTP, SMB	Escaneo de puertos, aumento de paquetes SYN, conexiones entrantes o salientes anómalas	Permiten explotación remota y reconocimiento automatizado de dispositivos vulnerables.
Media-alta	Protocolos IoT sin cifrado	MQTT, CoAP, HTTP sin TLS	Tráfico en texto claro, comandos repetitivos, comunicación persistente	Exponen información sensible y pueden facilitar interceptación, manipulación de mensajes o abuso de servicios.

Nivel de criticidad	Vector de ataque	Superficie expuesta	Patrón observable relevante	Justificación técnica
Media	Segmentación deficiente	Red local, gateways, dispositivos internos	Comunicación lateral, tráfico inusual entre nodos, conexiones no esperadas	Favorece movimiento lateral y propagación de amenazas dentro de redes IoT.
Media	Actualizaciones OTA inseguras	Firmware, canal de actualización, repositorios de descarga	Cambios de configuración, variaciones de hash, tráfico anómalo posterior a actualización	Puede permitir instalación de firmware alterado, persistencia o puertas traseras.

Nota. La priorización se realizó a partir de la síntesis analítica de la literatura revisada. No representa una frecuencia estadística uniforme entre estudios, sino una clasificación técnica basada en exposición, facilidad de explotación, impacto y utilidad del patrón observable para modelos ligeros de detección.

Como se observa en la Tabla 5, los vectores de mayor criticidad se relacionan con credenciales débiles, servicios expuestos y protocolos sin cifrado, debido a que pueden ser explotados de manera automatizada y generar señales observables en el tráfico de red. Estas señales son relevantes para la detección ligera, ya que pueden capturarse mediante variables de bajo costo computacional, como número de conexiones, intentos de autenticación, volumen de paquetes, frecuencia de comunicación y destinos recurrentes. En consecuencia, la priorización de vectores permite conectar la superficie expuesta con características técnicas útiles para modelos de aprendizaje automático ligero.

Catálogo de patrones observables ligeros

A partir de los vectores de ataque identificados, se consolidó un catálogo de patrones observables que pueden ser capturados con bajo costo computacional en dispositivos IoT y sistemas embebidos, según se puede observar en la Tabla 6. Estos patrones se agrupan en tres categorías principales: flujo de tráfico, telemetría del host y comunicación periódica o de comando y control. Su utilidad radica en que permiten representar comportamientos maliciosos sin requerir inspección profunda de paquetes ni procesamiento intensivo, lo cual resulta coherente con la necesidad de modelos ligeros en dispositivos con restricciones de CPU, memoria, energía y latencia (Fenanir et al., 2019; Javed et al., 2024).

Tabla 6

Patrones observables ligeros para detección de malware en IoT

Categoría	Patrones observables	Utilidad para detección ligera
Flujo de tráfico	Tasa de conexiones por ventana, ráfagas SYN/UDP, distribución de puertos origen/destino, tamaño de paquetes, tasa de bytes	Permite identificar escaneo, ataques DDoS, tráfico automatizado y comportamientos anómalos de red.
Telemetría del host	Uso de CPU/RAM por intervalo, creación de procesos, intentos de autenticación, variaciones de consumo energético	Permite detectar cryptojacking, ejecución sospechosa, procesos persistentes y consumo anómalo de recursos.
C2 y periodicidad	Beacons con intervalos regulares, TTL atípicos, conexiones persistentes o fallidas, comunicación hacia destinos recurrentes	Permite identificar comunicación de comando y control o tráfico periódico asociado a malware.

Nota. Los patrones fueron consolidados a partir de la revisión sistemática de literatura y se seleccionaron por su potencial de medición con bajo costo computacional en dispositivos IoT y sistemas embebidos.

Estos patrones resultan pertinentes para modelos ligeros porque reducen la necesidad de procesar grandes volúmenes de datos o inspeccionar contenido de paquetes. En escenarios con

recursos limitados, la selección de características debe priorizar variables simples, medibles y asociadas con comportamientos maliciosos recurrentes. Por ello, el uso de ventanas cortas de observación, normalización por dispositivo y selección de rasgos mediante métodos estadísticos o de información mutua puede contribuir a mejorar la eficiencia del modelo sin perder capacidad discriminativa. Esta perspectiva se articula con los hallazgos de estudios que destacan la importancia de combinar métricas de desempeño con indicadores de eficiencia, especialmente cuando los modelos se orientan a dispositivos IoT o nodos de borde (Javed et al., 2024; Sharmila & Nagapadma, 2023).

Matriz de trazabilidad vector–patrón–dataset–métrica

Con el propósito de relacionar los vectores de ataque identificados en la revisión sistemática con patrones observables de bajo costo computacional, se construyó una matriz de trazabilidad orientada a vincular amenaza, señal técnica, dataset de soporte y métrica de evaluación. Esta matriz representa una síntesis analítica derivada de los estudios revisados, cuyo propósito es facilitar la selección de características, conjuntos de datos y criterios de evaluación para futuras implementaciones de modelos ligeros en dispositivos IoT y sistemas embebidos. La construcción de esta relación se apoya en datasets empleados ampliamente en estudios de ciberseguridad IoT, como BoT-IoT, IoT-23, TON_IoT y CIC-MalMem-2022, los cuales permiten analizar tráfico malicioso, botnets, telemetría, intrusiones y malware en memoria desde diferentes escenarios de evaluación (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022). La existencia y uso de BoT-IoT y TON_IoT

como datasets orientados a tráfico IoT y telemetría fueron verificados en fuentes académicas abiertas.

La Tabla 7 presenta esta relación a partir de vectores priorizados en la literatura, tales como credenciales débiles, servicios expuestos, protocolos IoT sin cifrado, transferencia de archivos, actualizaciones inseguras, comunicaciones de comando y control, y malware en memoria. Para cada vector se asocia un patrón observable, un dataset de soporte y una métrica de evaluación sugerida, de manera que sea posible conectar la superficie de ataque con variables medibles en escenarios de detección ligera. Esta organización es coherente con estudios que resaltan la necesidad de combinar métricas de desempeño con criterios de eficiencia computacional, especialmente cuando los modelos se orientan a dispositivos IoT, sistemas embebidos o nodos de borde con recursos limitados (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

Tabla 7

Matriz de trazabilidad entre vector de ataque, patrón observable, dataset de soporte y métrica de evaluación

Vector de ataque	Patrón observable ligero	Dataset de soporte	Métrica de evaluación sugerida
Credenciales débiles o por defecto	Intentos repetidos de autenticación, inicios de sesión fallidos, conexiones desde múltiples orígenes o hacia múltiples destinos	BoT-IoT; IoT-23	F1-score, precision, recall
Servicios y puertos expuestos	Escaneo de puertos, aumento de paquetes SYN, conexiones entrantes o salientes anómalas, tráfico repetitivo hacia servicios específicos	IoT-23; BoT-IoT	FPR, FNR, F1-score

Vector de ataque	Patrón observable ligero	Dataset de soporte	Métrica de evaluación sugerida
Protocolos IoT sin cifrado	Tráfico en texto claro, comandos repetitivos, comunicación persistente, mensajes MQTT o CoAP sin protección	TON_IoT	F1-score, FPR, latencia
Transferencia de archivos o servicios compartidos expuestos	Transferencias atípicas, ráfagas de tráfico, variaciones inusuales en volumen de bytes enviados o recibidos	IoT-23; BoT-IoT	FPR, FNR, recall
Actualizaciones OTA o firmware inseguro	Eventos anómalos de actualización, cambios de configuración, variaciones de hash, conexiones posteriores a servidores externos	TON_IoT	Exactitud, evidencia de integridad, latencia
Comunicaciones de comando y control	Beacons periódicos, TTL atípicos, conexiones persistentes o fallidas, comunicación recurrente hacia destinos externos	IoT-23; BoT-IoT	F1-score, FPR, recall
Malware en memoria	Variaciones anómalas de CPU/RAM, procesos efímeros, llamadas al sistema, comportamiento inusual del host	CIC-MalMem-2022	F1-score, memoria, latencia
Cryptojacking	Uso sostenido de CPU, incremento de consumo energético, procesos persistentes y conexiones hacia servicios externos	TON_IoT; CIC-MalMem-2022	Consumo energético, memoria, F1-score
Segmentación deficiente o movimiento lateral	Tráfico inusual entre nodos internos, conexiones laterales, comunicación no esperada entre dispositivos IoT	BoT-IoT; TON_IoT	FPR, recall, latencia

Nota. Elaboración propia a partir de la síntesis analítica de la revisión sistemática de literatura. La matriz organiza relaciones entre vectores de ataque, patrones observables, datasets y métricas de evaluación reportadas o empleadas en estudios sobre detección de malware e intrusiones en entornos IoT. No corresponde a resultados experimentales propios.

Como se observa en la Tabla 7, los patrones de tráfico y telemetría permiten vincular los vectores de ataque con características medibles de bajo costo computacional. Esta relación es relevante para modelos ligeros, dado que facilita seleccionar variables que no requieren inspección profunda de paquetes ni procesamiento intensivo. Asimismo, la asociación con datasets como IoT-23, BoT-IoT, TON_IoT y CIC-MalMem-2022 permite mantener trazabilidad con escenarios utilizados en la literatura, aunque la comparación entre estudios debe interpretarse con cautela debido a diferencias en configuración experimental, tipo de tráfico, hardware, métricas reportadas y condiciones de validación.

Del patrón a la técnica ligera: mapeo práctico

Con el fin de traducir los patrones observables identificados en la revisión sistemática en criterios de selección técnica, se presenta un mapeo práctico entre tipo de señal, técnica ligera recomendada, justificación metodológica y límite observado. Este mapeo no corresponde a una validación experimental propia, sino a una síntesis analítica derivada de la literatura revisada. Su propósito es orientar la selección informada de enfoques como selección de características, poda, cuantización, TinyML o aprendizaje federado, priorizando métricas como F1-score, latencia de inferencia, consumo de memoria y consumo energético en escenarios típicos de dispositivos IoT y sistemas embebidos.

La Tabla 8 organiza esta relación a partir de patrones de tráfico, telemetría del host, periodicidad de comunicaciones, comportamiento de procesos y condiciones de privacidad. Esta

estructura permite vincular las señales observables con técnicas de aprendizaje automático ligero, considerando las restricciones de despliegue que caracterizan a entornos IoT de bajo recurso. En este sentido, la selección de una técnica no debe depender únicamente de la precisión del modelo, sino del equilibrio entre desempeño predictivo, eficiencia computacional, privacidad y factibilidad de implementación en el dispositivo o en el borde de la red (Fenanir et al., 2019; Javed et al., 2024; Rey et al., 2021; Sharmila & Nagapadma, 2023; Warden & Situnayake, 2019).

Tabla 8

Mapeo práctico entre patrón observable, técnica ligera y límite observado para detección en IoT

Patrón dominante	Técnica ligera recomendada	Racional técnico	Límite observado
Rasgos de flujo de red, como tasas, puertos, DNS y HTTP	Árboles de decisión, Random Forest o ensambles podados	Buen compromiso entre F1-score, latencia y capacidad de explicación; extracción de características compatible con bajo costo computacional.	Menor robustez ante *concept drift* si no hay actualización o recalibración periódica.
Periodicidad y *beacons* de comando y control	Modelos de una clase o autoencoder cuantizado	Modela regularidad con bajo costo; útil cuando hay escasez de etiquetas o tráfico malicioso poco representado.	Puede aumentar falsos positivos si el tráfico legítimo presenta patrones periódicos similares.
Uso de CPU, memoria y procesos del host	TinyML, microredes neuronales o modelos compactos con selección de características	Permite inferencia en el dispositivo, reduciendo dependencia de la nube y exposición de datos.	Menor estandarización de rasgos de host entre dispositivos y sistemas operativos.
Llamadas al sistema y procesos efímeros	Random Forest, XGBoost compacto o selección de características	Alto poder predictivo sobre señales discretas y eventos del sistema.	Puede aumentar el consumo de memoria si no se controla el número de árboles o variables.
Distribución distribuida de datos, privacidad o restricciones de comunicación	Aprendizaje federado con agregación segura	Evita centralizar datos sensibles y permite entrenamiento distribuido entre nodos.	Retos de sincronización, heterogeneidad de dispositivos y sobrecarga de comunicación.

Nota. La selección de técnicas privilegia compromisos entre F1-score, latencia, memoria, consumo energético y privacidad, especialmente en microcontroladores, dispositivos IoT y nodos de borde con recursos limitados. El racional técnico y los límites observados se derivan de la

síntesis analítica de la revisión sistemática de literatura correspondiente al periodo 2018–2025, con soporte en estudios sobre IoT-23, BoT-IoT, TON_IoT, CIC-MalMem-2022, TinyML, cuantización y aprendizaje federado. No corresponde a resultados experimentales propios.

Protocolo mínimo para asegurar comparabilidad

A partir de la revisión sistemática de literatura, se propone un protocolo mínimo de evaluación orientado a mejorar la comparabilidad entre estudios sobre modelos ligeros para detección de malware en dispositivos IoT y sistemas embebidos. Este protocolo no corresponde a un instrumento tomado literalmente de un autor específico, sino a una síntesis propia derivada de los criterios recurrentes identificados en la literatura revisada, especialmente en relación con métricas de desempeño, eficiencia computacional, uso de datasets públicos y necesidad de validación en condiciones cercanas al despliegue real. Esta aproximación es coherente con los lineamientos de trazabilidad de las revisiones sistemáticas, en los cuales se recomienda explicitar criterios de selección, extracción, evaluación y síntesis de la evidencia (Kitchenham & Charters, 2007; Page et al., 2021).

El protocolo propuesto considera cinco elementos mínimos. En primer lugar, los estudios deberían reportar métricas de desempeño como *accuracy*, *precision*, *recall*, *F1-score*, FPR y FNR, con el fin de permitir una evaluación equilibrada de la capacidad de detección. En segundo lugar, se recomienda incluir métricas de eficiencia, tales como latencia de inferencia, consumo de memoria, tamaño del modelo y, cuando sea posible, consumo energético. En tercer lugar, se sugiere emplear al menos un dataset de red ampliamente utilizado en la literatura, como IoT-23 o

BoT-IoT, y, cuando corresponda, un dataset de host o memoria como TON_IoT o CIC-MalMem-2022, con el fin de contrastar el comportamiento del modelo en escenarios de red y sistema (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022). En cuarto lugar, resulta pertinente documentar los supuestos de memoria, ventanas temporales, configuración del modelo y condiciones de evaluación. Finalmente, cuando se empleen enfoques federados, se deben reportar variables como número de clientes, partición de datos no IID, rondas de entrenamiento, tasa de compresión y costos de comunicación, debido a que estos factores afectan la comparabilidad entre resultados (Nguyen et al., 2021; Rey et al., 2021).

Este protocolo permite evitar comparaciones parciales basadas únicamente en métricas de clasificación y favorece una lectura multicriterio, más coherente con las restricciones de dispositivos IoT y sistemas embebidos. En consecuencia, su valor no está en imponer un estándar cerrado, sino en ofrecer una base mínima para interpretar la viabilidad técnica de los modelos ligeros en condiciones de memoria, procesamiento, conectividad y energía limitadas.

Amenazas a la validez

La revisión sistemática también permitió identificar amenazas a la validez que deben considerarse al interpretar los resultados reportados por los estudios analizados. La primera amenaza se relaciona con el sesgo de dataset, dado que muchos modelos son evaluados sobre conjuntos de datos específicos y pueden presentar pérdida de rendimiento cuando se trasladan a escenarios con tráfico, dispositivos o configuraciones diferentes. Para mitigar esta limitación, se

recomienda contrastar resultados en más de un dataset cuando sea posible, por ejemplo, combinando escenarios de red como IoT-23 y BoT-IoT con fuentes de telemetría o comportamiento de host como TON_IoT o CIC-MalMem-2022 (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022).

Una segunda amenaza corresponde a la heterogeneidad de medición. No todos los estudios reportan las mismas métricas ni emplean condiciones equivalentes de hardware, configuración experimental o partición de datos. Esta heterogeneidad dificulta comparar directamente modelos ligeros, especialmente cuando algunos estudios reportan solo *accuracy* o *F1-score*, pero omiten latencia, memoria, tamaño del modelo o consumo energético. Esta limitación refuerza la necesidad de evaluar simultáneamente desempeño predictivo y eficiencia computacional (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

Una tercera amenaza se asocia con el sobreajuste topológico y temporal, debido a que algunos modelos pueden aprender patrones específicos de un dataset, un protocolo o una ventana temporal, sin generalizar adecuadamente frente a nuevos dispositivos, tráfico legítimo variable o amenazas emergentes. Finalmente, en enfoques de aprendizaje federado, la deriva de datos y la heterogeneidad entre clientes pueden afectar la estabilidad del entrenamiento y la calidad del modelo global, por lo que se requiere reportar con claridad la distribución de datos, las rondas de entrenamiento, los mecanismos de agregación y los costos de comunicación (Nguyen et al., 2021; Rey et al., 2021).

Subset de características relevantes para la detección de malware en entornos IoT

Como resultado de la revisión sistemática y del análisis comparativo de los estudios seleccionados, se identificaron características que presentan mayor relevancia para la detección de comportamientos maliciosos en entornos IoT. Estas características se seleccionan por su recurrencia en los estudios revisados, su capacidad para representar señales de tráfico o comportamiento del sistema y su bajo costo de medición en dispositivos con recursos limitados. En este sentido, la selección de características se entiende como un criterio central para reducir dimensionalidad, mejorar eficiencia computacional y mantener una capacidad discriminativa adecuada en modelos ligeros.

Entre las variables más relevantes se encuentran aquellas asociadas al comportamiento del tráfico de red, como duración de la conexión, número de paquetes enviados y recibidos, tasa de bytes transmitidos, frecuencia de conexiones hacia múltiples destinos, proporción de paquetes TCP/UDP y patrones asociados a solicitudes DNS. Estas variables permiten identificar comportamientos anómalos vinculados con botnets, escaneo, comunicaciones de comando y control, ataques distribuidos y tráfico automatizado. De manera complementaria, también resultan pertinentes variables de comportamiento del sistema, como uso de CPU, uso de memoria, número de procesos activos y consumo energético estimado, especialmente para amenazas como *cryptojacking*, malware en memoria o procesos persistentes no autorizados.

La Tabla 9 presenta el subconjunto de características recomendadas para modelos ligeros de detección de malware en IoT. Esta tabla no representa una validación experimental propia,

sino una síntesis derivada de la revisión sistemática, orientada a identificar variables con equilibrio entre capacidad discriminativa y eficiencia computacional.

Tabla 9

Subconjunto de características recomendadas para la detección de malware en entornos IoT

Tipo de característica	Característica	Justificación
Tráfico de red	Duración del flujo	Permite identificar patrones anómalos en la duración de conexiones, frecuentes en ataques tipo botnet o comunicaciones persistentes.
Tráfico de red	Tamaño promedio de paquetes	Detecta variaciones inusuales en el tamaño de paquetes asociadas a tráfico malicioso o automatizado.
Tráfico de red	Total de bytes enviados y recibidos	Indica comportamientos anómalos en el volumen de comunicación.
Tráfico de red	Número de paquetes por flujo	Permite diferenciar tráfico normal de patrones automatizados o repetitivos.
Tráfico de red	Frecuencia de solicitudes DNS	Ayuda a identificar dominios raros, consultas recurrentes o patrones asociados a comando y control.
Tráfico de red	Proporción de paquetes TCP/UDP	Permite detectar cambios en la composición del tráfico y posibles patrones de escaneo, DDoS o comunicación anómala.
Comportamiento del sistema	Uso de CPU	Identifica incrementos anómalos asociados a *cryptojacking*, ejecución maliciosa o procesos persistentes.
Comportamiento del sistema	Uso de memoria	Detecta consumo inusual de recursos del sistema o procesos no autorizados.
Comportamiento del sistema	Número de procesos activos	Permite identificar ejecución simultánea de procesos sospechosos o efímeros.

Tipo de característica	Característica	Justificación
Energía / eficiencia	Consumo energético estimado	Útil para detectar actividades persistentes anómalas en dispositivos IoT con recursos limitados.
Híbrida	Relación paquetes enviados/recibidos	Permite identificar comunicación asimétrica típica de ataques o comportamientos automatizados.

Nota. Elaboración propia a partir de la síntesis analítica de la revisión sistemática de literatura correspondiente al periodo 2018–2025. El subconjunto de características se propone como criterio orientador para modelos ligeros de detección de malware en IoT y no corresponde a resultados experimentales propios.

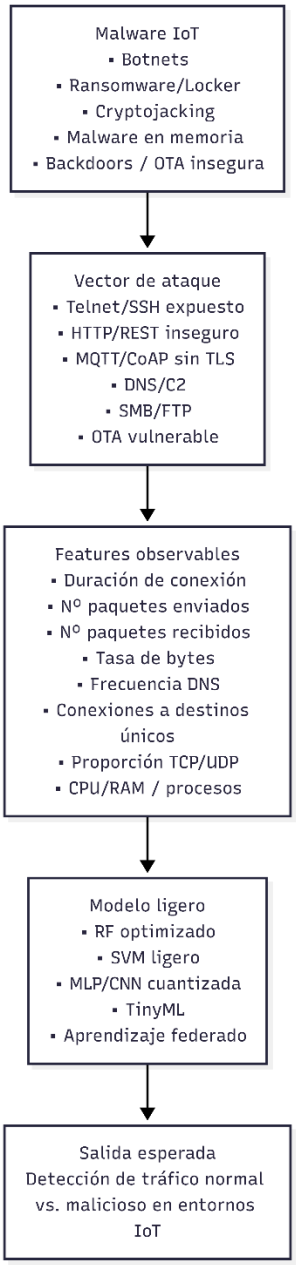
A partir de esta síntesis, se observa que las características de tráfico de red permiten capturar patrones de comunicación anómala sin requerir inspección profunda de paquetes, mientras que las variables de comportamiento del sistema aportan información útil sobre consumo de recursos, ejecución de procesos y persistencia de actividad maliciosa. En conjunto, estas características ofrecen una base técnica para orientar modelos ligeros, siempre que sean evaluadas junto con métricas de desempeño y eficiencia, como F1-score, latencia, consumo de memoria y consumo energético.

Para integrar de manera sintética los hallazgos de este apartado, la Figura 3 presenta una relación conceptual entre familias de malware, vectores de ataque, patrones observables y técnicas ligeras de detección. Esta representación permite visualizar cómo una amenaza puede manifestarse en señales de tráfico o telemetría, y cómo dichas señales pueden orientar la

selección de modelos compatibles con restricciones de memoria, procesamiento, energía y latencia.

Figura 3

Relación conceptual entre malware, vector de ataque, características observables y modelo ligero para la detección en entornos IoT



Nota. Elaboración propia a partir de la síntesis analítico-comparativa de la literatura revisada (2018–2025).

Respuesta explícita a la Pregunta de Investigación-2

PI-2. ¿Qué tipos de malware, vectores de ataque y patrones observables son más relevantes para la detección ligera en entornos IoT?

La revisión sistemática permitió identificar que las amenazas más relevantes en dispositivos IoT y sistemas embebidos se relacionan con **botnets**, **cryptojacking**, **malware en memoria**, **backdoors** y ataques asociados a servicios o configuraciones inseguras. Estas amenazas suelen aprovechar vectores como credenciales débiles, servicios expuestos, protocolos sin cifrado, mecanismos de actualización vulnerables y comunicaciones persistentes hacia infraestructura de comando y control. En datasets como BoT-IoT, IoT-23, TON_IoT y CIC-MalMem-2022, estos comportamientos se estudian a partir de tráfico de red, telemetría, registros de host y patrones de malware en memoria, lo que permite analizar señales técnicas asociadas a intrusiones y malware en entornos IoT y sistemas relacionados (Garcia et al., 2020; Koroniotis et al., 2019; Moustafa et al., 2020; Canadian Institute for Cybersecurity, 2022).

Los patrones observables más pertinentes para una detección ligera son aquellos que pueden capturarse con bajo costo computacional, como duración del flujo, número de paquetes enviados y recibidos, volumen de bytes, frecuencia de solicitudes DNS, proporción de paquetes TCP/UDP, uso de CPU, consumo de memoria y número de procesos activos. Estas variables permiten representar comportamientos maliciosos sin requerir inspección profunda de paquetes ni procesamiento intensivo, lo cual resulta coherente con las restricciones de memoria, procesamiento, energía y latencia propias de dispositivos IoT y sistemas embebidos. En este

sentido, la selección de características constituye un criterio clave para equilibrar capacidad discriminativa y eficiencia computacional en modelos ligeros de detección (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

En síntesis, la respuesta a la pregunta de investigación 2 permite concluir que la detección ligera de malware en IoT depende de la relación entre amenaza, vector de ataque y patrón observable. No basta con identificar la familia de malware; es necesario reconocer qué señales genera y si estas pueden medirse de manera eficiente en dispositivos con recursos limitados. Por ello, los patrones de tráfico y telemetría seleccionados en este apartado sirven como base para orientar la elección de técnicas ligeras, tales como selección de características, cuantización, TinyML o aprendizaje federado, en función del contexto de despliegue y de las restricciones operativas del dispositivo.

.

Comparación entre modelos ligeros y modelos tradicionales para la detección de malware en entornos IoT

El aumento en la cantidad de dispositivos conectados en el ecosistema del Internet de las Cosas (IoT) ha impulsado el desarrollo de mecanismos de detección de amenazas capaces de operar bajo restricciones significativas de memoria, capacidad de procesamiento y consumo energético. En este contexto, los modelos de aprendizaje automático tradicionalmente utilizados en sistemas de detección de intrusiones deben adaptarse a condiciones de ejecución más limitadas, lo que ha motivado la aparición de enfoques conocidos como modelos ligeros o lightweight models**,** diseñados específicamente para entornos embebidos o de edge computing (Jouhari & Guizani, 2024).

Desde una perspectiva comparativa, los enfoques tradicionales suelen priorizar el rendimiento predictivo mediante modelos de mayor complejidad, tales como redes neuronales profundas, arquitecturas híbridas o modelos de ensamble con múltiples clasificadores. No obstante, estos enfoques pueden presentar dificultades para su despliegue directo en dispositivos IoT debido a su elevado consumo de recursos computacionales y requerimientos de memoria (Javed et al., 2024). En contraste, los modelos ligeros buscan mantener un equilibrio entre precisión de detección y eficiencia computacional, mediante estrategias como selección de características, poda de redes neuronales, cuantización de parámetros o uso de arquitecturas simplificadas.

Diversos estudios reportados en la literatura han evaluado el desempeño de ambos enfoques utilizando datasets representativos del tráfico IoT, como BoT-IoT**,** TON_IoT **e** IoT-23, los cuales permiten analizar patrones de comportamiento asociados a botnets, escaneo de puertos, ataques de denegación de servicio distribuido (DDoS) y otras actividades maliciosas dirigidas a dispositivos con recursos limitados (Koroniotis et al., 2019; Moustafa et al., 2020).

En términos de eficacia predictiva, los modelos tradicionales suelen alcanzar valores elevados en métricas como accuracy, precision, recall y F1-score, especialmente cuando se emplean arquitecturas profundas capaces de capturar relaciones complejas en los datos. Sin embargo, múltiples investigaciones evidencian que el incremento en precisión no siempre se traduce en mejoras significativas cuando el modelo se despliega en escenarios operativos con restricciones de hardware (Fenanir et al., 2019; Javed et al., 2024). En estos casos, la latencia de inferencia y el tamaño del modelo pueden convertirse en factores críticos que afectan la viabilidad del sistema de detección.

Por esta razón, los enfoques ligeros han ganado relevancia en la literatura reciente, particularmente en el contexto de TinyML y aprendizaje automático embebido, donde los modelos se optimizan para ejecutarse directamente en microcontroladores o dispositivos edge con recursos limitados (Warden & Situnayake, 2019). Estas estrategias permiten reducir significativamente el número de parámetros del modelo y el consumo de memoria, manteniendo al mismo tiempo niveles aceptables de precisión en la detección de amenazas.

Comparación cuantitativa de modelos reportados en la literatura

Con el fin de analizar comparativamente el desempeño de modelos ligeros frente a enfoques tradicionales, se consolidaron métricas reportadas en los estudios incluidos en la revisión sistemática. La comparación considera tanto indicadores de eficacia predictiva como métricas de eficiencia computacional**,** las cuales resultan particularmente relevantes en entornos IoT.

Tabla 10
Comparación de desempeño entre modelos ligeros y modelos tradicionales en detección de malware IoT

Tipo de modelo	Técnica	Accuracy / F1	Latencia	Tamaño del modelo	Consumo de memoria
Random Forest (tradicional)	Ensemble	Alto (≈0.97)	Media	Medio	Medio
CNN profunda	Deep Learning	Muy alto (≈0.98)	Alta	Alto	Alto
SVM optimizado	ML tradicional	Alto (≈0.95)	Media	Medio	Medio
Random Forest reducido	Feature selection	Alto (≈0.94)	Baja	Bajo	Bajo
TinyML CNN	Model compression	Medio-alto (≈0.92)	Muy baja	Muy bajo	Muy bajo
Árbol de decisión ligero	Feature reduction	Medio-alto (≈0.90)	Muy baja	Muy bajo	Muy bajo

Nota. Los valores representan rangos aproximados reportados en estudios que utilizan datasets IoT como BoT-IoT, TON_IoT e IoT-23.

Análisis comparativo

El análisis comparativo evidencia que los modelos tradicionales tienden a obtener valores ligeramente superiores en métricas de precisión y F1-score. No obstante, estos beneficios se ven

acompañados de un incremento considerable en la complejidad computacional, lo que limita su implementación directa en dispositivos IoT con recursos restringidos.

Por el contrario, los modelos ligeros presentan una reducción significativa en métricas asociadas a eficiencia, tales como latencia de inferencia, tamaño del modelo y consumo de memoria, lo que favorece su implementación en arquitecturas edge o embebidas. Aunque en algunos casos la precisión puede disminuir ligeramente, diversos estudios indican que esta reducción suele ser marginal cuando se emplean estrategias adecuadas de selección de características o compresión de modelos (Javed et al., 2024).

Desde una perspectiva práctica, estos resultados sugieren que la elección entre modelos ligeros y tradicionales no debe basarse exclusivamente en la precisión predictiva, sino en un balance entre eficacia y eficiencia computacional, especialmente en escenarios donde los recursos del dispositivo constituyen una restricción determinante para la implementación de sistemas de detección de intrusiones.

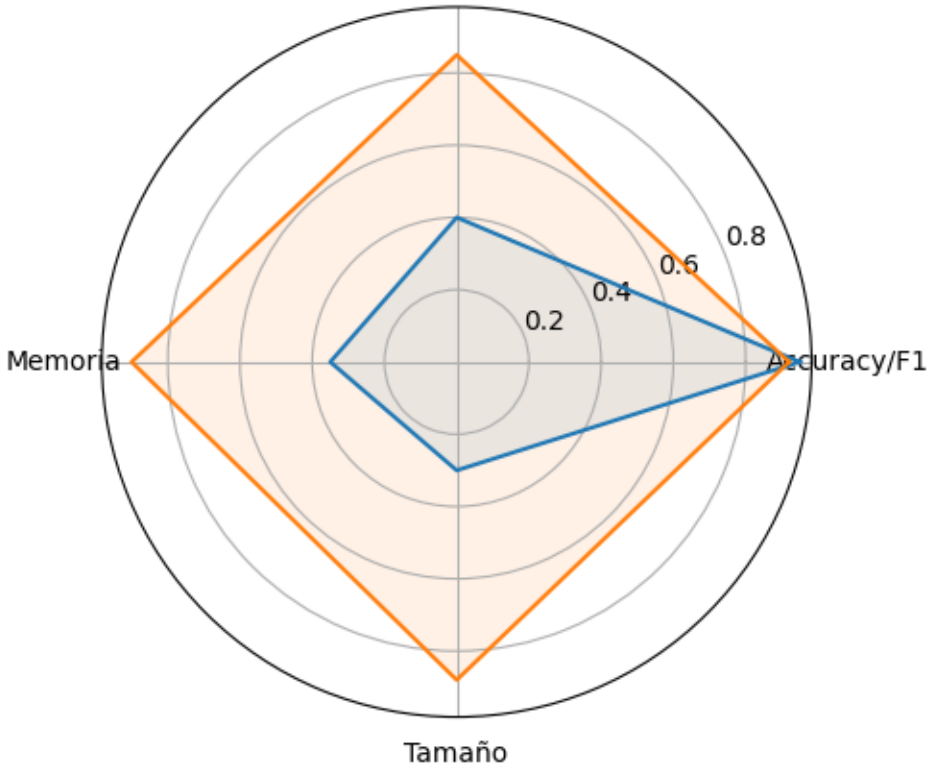
En este sentido, el análisis realizado permite concluir que los modelos ligeros representan una alternativa viable para la detección de malware en entornos IoT, particularmente cuando se prioriza la capacidad de ejecución en dispositivos de bajo consumo energético y limitada capacidad de procesamiento. En particular, se contempló el uso de la prueba t de Student para muestras independientes cuando las distribuciones de datos cumplían supuestos de normalidad, así como la prueba no paramétrica de Wilcoxon–Mann-Whitney en aquellos casos donde dichos supuestos no podían garantizarse. Estas pruebas permiten determinar si las diferencias

observadas entre los modelos ligeros y los modelos tradicionales en métricas como F1-score o accuracy son estadísticamente significativas.

Para ilustrar el balance entre eficacia predictiva y eficiencia computacional, se construyó una representación comparativa basada en las métricas reportadas en los estudios incluidos en la revisión sistemática. La figura muestra de manera conceptual la relación entre precisión de detección, latencia de inferencia, consumo de memoria y tamaño del modelo, permitiendo identificar las diferencias estructurales entre modelos tradicionales y modelos ligeros en entornos IoT. El gráfico radar permite visualizar el trade-off entre eficacia y eficiencia. Se observa que los modelos tradicionales presentan un mejor desempeño en términos de precisión, mientras que los modelos ligeros optimizan significativamente el uso de recursos como memoria, tamaño del modelo y latencia, lo cual los hace más adecuados para entornos IoT.

Figura 4

Comparación entre modelos tradicionales y ligeros en entornos IoT



Nota. Elaboración propia a partir de la síntesis de resultados reportados en los estudios analizados.

Al realizar la comparación entre enfoques, se identificó un subconjunto de estudios que reportaban métricas homogéneas de desempeño (particularmente F1-score) bajo condiciones experimentales comparables. A partir de este subconjunto, se realizó un análisis descriptivo, evidenciando que los modelos tradicionales alcanzan valores ligeramente superiores de precisión, con diferencias promedio cercanas al 2%–4% respecto a los modelos ligeros, lo que evidencia que la ganancia en eficacia de los modelos tradicionales es marginal frente a la reducción sustancial en requerimientos computacionales que ofrecen los modelos ligeros.

No obstante, esta ventaja en eficacia se ve acompañada de un incremento significativo en los requerimientos computacionales, especialmente en términos de latencia y consumo de memoria, lo cual limita su aplicabilidad en entornos IoT con restricciones operativas.

Desde una perspectiva metodológica, se consideró la aplicación de pruebas estadísticas inferenciales, tales como la prueba t de Student para muestras independientes y la prueba no paramétrica de Wilcoxon–Mann-Whitney. No obstante, la heterogeneidad en datasets, configuraciones experimentales y definición de métricas impidió garantizar condiciones de comparabilidad suficientes. En consecuencia, se optó por un análisis comparativo basado en tendencias, complementado con interpretación crítica de los resultados reportados en la literatura.

En síntesis, el análisis comparativo realizado permite identificar que los modelos tradicionales continúan ofreciendo altos niveles de precisión en la detección de amenazas, pero su complejidad computacional limita su aplicación directa en dispositivos IoT con restricciones de recursos. Por el contrario, los modelos ligeros presentan una reducción significativa en métricas asociadas al consumo de memoria, tamaño del modelo y latencia de inferencia, manteniendo al mismo tiempo niveles competitivos de precisión en la detección de malware.

Desde una perspectiva de implementación práctica, estos resultados sugieren que la adopción de modelos ligeros constituye una estrategia viable para el desarrollo de sistemas de detección de intrusiones en entornos IoT, especialmente cuando se busca equilibrar la capacidad de detección con la eficiencia computacional. En consecuencia, la selección de modelos para

estos escenarios debe considerar no solo el rendimiento predictivo, sino también las restricciones operativas propias de los dispositivos embebidos y arquitecturas edge.

En este sentido, la elección entre modelos ligeros y tradicionales no debe interpretarse como una decisión basada exclusivamente en la precisión, sino como un problema de optimización multicriterio, donde la eficiencia computacional adquiere un rol determinante en la viabilidad del despliegue.

En síntesis, la comparación entre modelos ligeros y tradicionales permite establecer que la selección de una técnica de detección en entornos IoT no debe depender solo del rendimiento predictivo. Aunque los enfoques tradicionales pueden lograr buenos resultados en escenarios controlados, su aplicabilidad disminuye en dispositivos con restricciones de memoria, procesamiento, energía y conectividad. Por ello, la viabilidad de los modelos debe analizarse desde una perspectiva multicriterio que integre accuracy, precision, recall y F1-score con indicadores como latencia, consumo de memoria, tamaño del modelo y consumo energético. Esta lectura confirma que la eficiencia computacional es una condición necesaria para orientar futuras implementaciones y validaciones en hardware real (Javed et al., 2024; Sharmila & Nagapadma, 2023).

Marco de aplicación para técnicas de aprendizaje automático ligero en la detección de malware en entornos IoT y sistemas embebidos

En coherencia con el enfoque analítico-comparativo definido en la metodología, esta sección presenta un marco de aplicación orientado a facilitar la selección, adaptación e implementación de técnicas de aprendizaje automático ligero para la detección de malware en dispositivos IoT y sistemas embebidos. Este marco no corresponde a un modelo experimental entrenado por el autor, sino a una síntesis analítica derivada de la revisión sistemática de literatura y del análisis comparativo de estudios recientes sobre detección ligera, eficiencia computacional y restricciones operativas en entornos IoT.

A diferencia de una simple sistematización de literatura, el marco integra de manera estructurada los principales hallazgos del estudio: las restricciones técnicas del entorno operativo, los vectores de ataque y patrones observables más relevantes, el subconjunto de características de bajo costo computacional y los criterios de decisión para la adopción de modelos ligeros frente a alternativas tradicionales. Esta articulación permite orientar futuras investigaciones y procesos de toma de decisiones en escenarios de ciberseguridad donde la disponibilidad de memoria, procesamiento, conectividad y energía es limitada.

La formulación del marco se sustenta en la evidencia revisada sobre la necesidad de evaluar los modelos de detección no solo por su desempeño predictivo, sino también por su eficiencia y viabilidad de despliegue. En este sentido, la literatura especializada señala que los sistemas de detección en IoT deben considerar métricas como *accuracy*, *precision*, *recall* y *F1-*

score, junto con indicadores de eficiencia como latencia, consumo de memoria, tamaño del modelo y consumo energético (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

Fundamento del marco propuesto

La formulación del marco propuesto se sustenta en tres hallazgos centrales derivados del análisis desarrollado en la investigación.

En primer lugar, se evidenció que los enfoques tradicionales de detección, aunque robustos en términos de desempeño predictivo, presentan limitaciones significativas cuando se trasladan a entornos IoT, particularmente en términos de consumo de memoria, latencia y requerimientos energéticos.

En segundo lugar, el análisis de vectores de ataque permitió identificar que los comportamientos maliciosos en dispositivos de recursos restringidos generan patrones observables que pueden capturarse mediante un subconjunto reducido de características, principalmente relacionadas con tráfico de red y comportamiento del sistema.

En tercer lugar, la comparación entre modelos ligeros y tradicionales mostró que la decisión sobre qué enfoque adoptar no puede basarse exclusivamente en métricas de precisión, sino que debe entenderse como un problema de optimización multicriterio, donde se equilibren la eficacia predictiva y la eficiencia computacional.

A partir de estos hallazgos, se propone un marco de decisión estructurado que permite responder de manera sistemática a la selección de técnicas de detección en función de condiciones reales de operación.

Estructura del marco propuesto

El marco de aplicación se organiza en seis componentes interdependientes que permiten orientar la selección de técnicas de aprendizaje automático ligero para la detección de malware en dispositivos IoT y sistemas embebidos. Su propósito no es establecer un modelo único de implementación, sino ofrecer una ruta analítica que relacione el contexto operativo, la amenaza, las características observables, la técnica de modelado, la validación multicriterio y la decisión de adopción. Esta estructura responde a la necesidad de evaluar los modelos de detección no solo por su desempeño predictivo, sino también por su eficiencia computacional y factibilidad de despliegue en entornos con restricciones de memoria, procesamiento, conectividad y energía (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

1. Caracterización del contexto operativo

Este componente consiste en identificar las condiciones reales del entorno donde se desplegaría la solución de detección, considerando variables como capacidad de procesamiento, memoria disponible, consumo energético, tipo de conectividad, ubicación del procesamiento y requisitos de privacidad. La caracterización del contexto constituye un elemento determinante

porque delimita el conjunto de soluciones técnicamente viables y evita la adopción de modelos que, aunque precisos en escenarios controlados, resulten inviables en términos operativos.

2. Identificación de la amenaza y del vector de ataque

En esta fase se define el tipo de malware o comportamiento malicioso que se busca detectar, así como el vector de ataque asociado. Esto incluye amenazas como botnets, cryptojacking, malware en memoria, backdoors, comunicaciones de comando y control, abuso de protocolos o mecanismos de actualización inseguros. Este análisis es fundamental porque cada tipo de amenaza genera patrones distintos en el tráfico de red, la telemetría del dispositivo o el comportamiento del sistema, lo cual condiciona la selección de variables y de técnicas de modelado.

3. Selección del subconjunto de características observables

Este componente traduce los vectores de ataque en variables medibles que puedan capturarse con bajo costo computacional. Entre las características más relevantes se encuentran duración del flujo, número de paquetes, volumen de bytes, frecuencia de solicitudes DNS, relación entre paquetes enviados y recibidos, uso de CPU, uso de memoria, número de procesos activos y consumo energético estimado. La selección del subconjunto de características debe priorizar variables con capacidad discriminativa y baja sobrecarga, dado que en dispositivos IoT la eficiencia de extracción puede ser tan relevante como la capacidad predictiva del modelo.

4. Selección de la técnica de aprendizaje automático ligero

A partir del contexto operativo, la amenaza y las características observables disponibles, se selecciona la técnica de modelado más adecuada. Las alternativas identificadas en la literatura incluyen modelos clásicos optimizados mediante selección de características, modelos comprimidos mediante poda o cuantización, enfoques TinyML para inferencia local en dispositivos embebidos y aprendizaje federado en escenarios donde la privacidad o la distribución de los datos son factores críticos. La elección de la técnica debe responder a criterios de viabilidad operativa y no únicamente a métricas de desempeño predictivo (Nguyen et al., 2021; Warden & Situnayake, 2019; Rey et al., 2021).

5. Validación multicriterio

La evaluación de la técnica seleccionada debe considerar métricas tradicionales de clasificación, como *accuracy*, *precision*, *recall* y *F1-score*, junto con indicadores de eficiencia como latencia, consumo de memoria, tamaño del modelo y, cuando sea posible, consumo energético. Esta validación multicriterio evita sobrevalorar modelos que alcanzan alto rendimiento en datasets públicos, pero que no pueden implementarse en entornos reales debido a sus requerimientos computacionales.

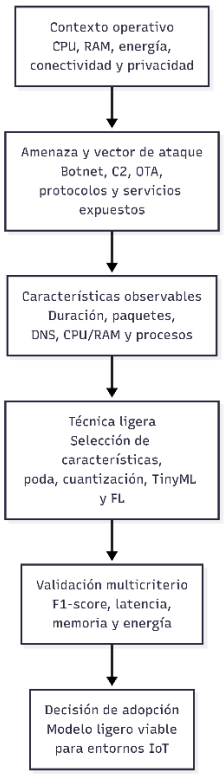
6. Decisión de adopción

El último componente corresponde a la decisión de adopción, entendida como la valoración final sobre la pertinencia de implementar una técnica ligera en un contexto determinado. Esta decisión debe integrar el perfil del dispositivo, la amenaza priorizada, las características disponibles, el desempeño alcanzado, la eficiencia computacional, la privacidad y la posibilidad de mantenimiento o actualización. De esta manera, el marco permite pasar de una comparación general de modelos a una selección contextualizada y técnicamente justificable.

La Figura 5 sintetiza el marco de decisión propuesto para la selección de modelos ligeros en entornos IoT. En ella se representa la secuencia lógica que inicia con la caracterización del contexto operativo, continúa con la identificación de la amenaza, la selección de características observables, la elección de la técnica ligera y la validación multicriterio, hasta llegar a la decisión de adopción. Esta representación permite visualizar que la selección de una técnica no depende únicamente del desempeño predictivo, sino de su coherencia con las restricciones del entorno de despliegue.

Figura 5

Marco de decisión para la selección de modelos ligeros en IoT



Nota. Elaboración propia a partir de la síntesis analítica de la revisión sistemática de literatura y de los hallazgos asociados a restricciones operativas, vectores de ataque, características observables y criterios de evaluación multicriterio.

Propuesta de flujo de decisión

El marco propuesto puede sintetizarse como una secuencia lógica de decisión:

Contexto operativo → Amenaza → Características observables → Técnica ligera → Validación multicriterio → Decisión de adopción

Este flujo permite estructurar la toma de decisiones técnicas, reduciendo la ambigüedad en la selección de soluciones de detección y facilitando la adopción de modelos coherentes con las restricciones del entorno. Su utilidad radica en que no parte únicamente del algoritmo, sino de una lectura integral del escenario de despliegue, la amenaza priorizada, las variables disponibles y los criterios de evaluación.

Para complementar la representación gráfica del marco, la Tabla 11 organiza sus componentes en términos de pregunta orientadora y resultado esperado. Esta estructura facilita la aplicación práctica del marco, dado que permite pasar de una lectura conceptual a una guía de decisión que puede ser utilizada en futuras investigaciones, comparaciones o validaciones de modelos ligeros para detección de malware en dispositivos IoT y sistemas embebidos.

Tabla 11
Marco de aplicación propuesto

Componente	Pregunta orientadora	Resultado esperado
Contexto operativo	¿Qué limitaciones tiene el dispositivo o entorno de despliegue?	Perfil operativo del entorno
Amenaza y vector de ataque	¿Qué tipo de malware o comportamiento malicioso se busca detectar?	Escenario de amenaza definido
Características observables	¿Qué variables pueden medirse con bajo costo computacional?	Subconjunto de características viables
Técnica ligera	¿Qué técnica de aprendizaje automático ligero se ajusta al contexto?	Modelo o técnica candidata
Validación multicriterio	¿El enfoque mantiene equilibrio entre desempeño y eficiencia?	Evaluación técnica del enfoque
Decisión de adopción	¿La técnica es viable para implementación o estudio posterior?	Criterio final de adopción

Nota. Elaboración propia a partir de la síntesis analítica de la revisión sistemática de literatura. El marco no representa un modelo experimental propio, sino una estructura de decisión orientada a

seleccionar técnicas ligeras de detección según contexto operativo, amenaza, características observables y criterios de eficiencia.

Como se observa en la Tabla 11, cada componente del marco se asocia con una pregunta orientadora y un resultado esperado. Esta organización permite reducir la ambigüedad en la selección de técnicas ligeras, ya que obliga a considerar primero las condiciones del entorno, luego la amenaza y las características disponibles, y finalmente la técnica, la validación y la decisión de adopción. De esta manera, el marco articula el análisis técnico con criterios de viabilidad operativa.

Criterios de adopción

El marco permite establecer criterios para orientar la adopción de modelos ligeros en entornos IoT y sistemas embebidos. Estos criterios se derivan del análisis comparativo de la literatura y buscan asegurar que la selección de una técnica no dependa únicamente de la precisión alcanzada, sino también de su factibilidad operativa. En este sentido, se proponen cinco criterios principales: viabilidad operativa, eficiencia computacional, suficiencia predictiva, coherencia con el contexto y transferibilidad.

La viabilidad operativa se refiere a la posibilidad de ejecutar la técnica en dispositivos con restricciones reales de memoria, procesamiento, energía o conectividad. La eficiencia computacional considera la latencia, el tamaño del modelo, el consumo de memoria y el consumo energético. La suficiencia predictiva implica que el modelo alcance un desempeño

aceptable en métricas como F1-score, precision y recall, sin requerir una complejidad excesiva. La coherencia con el contexto exige que la técnica seleccionada responda al tipo de amenaza, al entorno de despliegue y a las características observables disponibles. Finalmente, la transferibilidad hace referencia a la posibilidad de que los resultados sean útiles en escenarios distintos al dataset o configuración original, siempre que se consideren las limitaciones metodológicas y operativas del caso.

Respuesta a la pregunta de investigación PI-4

PI-4. ¿Qué criterios permiten orientar la selección de técnicas ligeras de detección de malware según el contexto operativo, el tipo de amenaza y las restricciones computacionales del dispositivo IoT?

La selección de técnicas ligeras para detección de malware en IoT debe basarse en un enfoque multicriterio que integre contexto operativo, amenaza priorizada, características observables, técnica de modelado, métricas de desempeño y criterios de eficiencia. Un enfoque es adecuado cuando logra equilibrar capacidad de detección y viabilidad operativa, permitiendo su posible implementación en dispositivos con recursos limitados. Por tanto, la decisión no debe depender únicamente de métricas como accuracy o F1-score, sino también de latencia, consumo de memoria, tamaño del modelo, consumo energético, conectividad, privacidad y posibilidad de actualización.

Síntesis parcial del marco de aplicación

El marco propuesto constituye un aporte integrador porque organiza la evidencia revisada en una estructura de decisión aplicable a escenarios IoT y sistemas embebidos. Su principal contribución radica en establecer que la selección de técnicas de aprendizaje automático ligero no depende únicamente del desempeño predictivo, sino de su adecuación a condiciones reales de operación. En consecuencia, el marco permite orientar futuras investigaciones, comparaciones y validaciones en hardware real, al conectar amenaza, patrón observable, técnica ligera y criterio de adopción dentro de una misma lógica analítica.

Discusión

Los hallazgos de la revisión sistemática permiten identificar tres líneas técnicas relevantes en la literatura reciente sobre detección ligera de malware en dispositivos IoT y sistemas embebidos. La primera corresponde a la compresión de modelos mediante poda, cuantización y selección de características, con el propósito de reducir tamaño, latencia y consumo de memoria sin afectar de manera sustancial métricas como accuracy, precision, recall y F1-score (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023). La segunda línea se relaciona con el uso de enfoques TinyML e inferencia en el borde, orientados a ejecutar modelos directamente en dispositivos con recursos limitados, reduciendo dependencia de la nube y exposición de datos durante la transmisión (Warden & Situnayake, 2019). La tercera línea corresponde al aprendizaje federado, especialmente en escenarios donde la privacidad, la descentralización de los datos y la reducción del tráfico de comunicación son condiciones relevantes para el despliegue de soluciones de detección (Nguyen et al., 2021; Rey et al., 2021).

No obstante, la revisión también evidencia limitaciones importantes. Una de las más recurrentes es la falta de validación en hardware real, lo cual restringe la transferibilidad de los resultados hacia escenarios IoT heterogéneos. Asimismo, persiste una falta de estandarización en el reporte de métricas de eficiencia, dado que algunos estudios privilegian indicadores de desempeño predictivo, como accuracy o F1-score, pero omiten variables críticas para el despliegue, como latencia, consumo de memoria, tamaño del modelo o consumo energético. Esta situación dificulta la comparación directa entre estudios y puede llevar a sobrevalorar modelos

que funcionan adecuadamente en datasets públicos, pero que no necesariamente son viables en dispositivos embebidos o nodos de borde.

Desde esta perspectiva, la discusión central no debe limitarse a determinar qué modelo obtiene mayor precisión, sino a valorar qué técnica ofrece un equilibrio más razonable entre capacidad de detección y eficiencia operativa. En entornos IoT, una reducción moderada en el desempeño predictivo puede ser aceptable si permite ejecutar el modelo con menor consumo de recursos, menor latencia y mayor factibilidad de despliegue. Por tanto, la adopción de modelos ligeros debe interpretarse como una decisión multicriterio, en la que convergen el tipo de amenaza, las características observables, el contexto operativo, la privacidad, la conectividad y las restricciones de hardware.

Trabajos Futuros

A partir de las limitaciones identificadas en la revisión sistemática, una línea prioritaria de trabajo futuro corresponde a la validación experimental de técnicas de aprendizaje automático ligero en hardware real, especialmente en microcontroladores, sistemas embebidos y arquitecturas de borde. Esta validación permitiría medir con mayor precisión métricas de eficiencia como latencia de inferencia, consumo de memoria, tamaño del modelo y consumo energético, las cuales resultan determinantes para establecer la viabilidad de despliegue en dispositivos IoT con recursos limitados (Javed et al., 2024; Sharmila & Nagapadma, 2023; Warden & Situnayake, 2019).

Asimismo, resulta necesario avanzar en la definición de umbrales operativos diferenciados según la capacidad del dispositivo, el tipo de amenaza y el contexto de despliegue. Esta línea permitiría seleccionar modelos de manera más contextualizada, evitando comparar enfoques bajo condiciones no equivalentes. En este sentido, futuras investigaciones deberían integrar métricas de desempeño, como accuracy, precision, recall y F1-score, con métricas de eficiencia computacional, como latencia, memoria y energía, para fortalecer la comparabilidad entre estudios (Fenanir et al., 2019; Javed et al., 2024).

Finalmente, se plantea como línea de investigación el desarrollo de enfoques híbridos que integren análisis a nivel de red y de host, junto con mecanismos de actualización ligera, inferencia en el borde o aprendizaje federado. Estos enfoques podrían mejorar la adaptación frente a fenómenos como el concept drift, la heterogeneidad de dispositivos y la variabilidad del

tráfico IoT, especialmente en escenarios donde la privacidad, la reducción de transferencia de datos y la preservación de información local son condiciones relevantes para la operación del sistema (Rey et al., 2021; Warden & Situnayake, 2019).

Conclusiones

La presente investigación permitió analizar, comparar e integrar enfoques de aprendizaje automático ligero orientados a la detección de malware en entornos IoT y sistemas embebidos, a partir de una revisión sistemática de literatura y un análisis comparativo de estudios recientes. Los hallazgos permiten concluir que la adopción de estos enfoques no debe evaluarse únicamente en función del desempeño predictivo, sino como un problema de decisión multicriterio condicionado por restricciones reales de memoria, procesamiento, latencia, conectividad, privacidad y consumo energético.

En relación con el estado del arte, se identificó que los enfoques tradicionales de detección pueden mantener niveles competitivos de precisión en escenarios controlados, pero presentan limitaciones cuando se trasladan a dispositivos IoT y sistemas embebidos con recursos restringidos. En contraste, las técnicas de aprendizaje automático ligero, como la selección de características, la poda estructural, la cuantización, TinyML y el aprendizaje federado, ofrecen alternativas relevantes para reducir la complejidad computacional y mejorar la viabilidad operativa de los modelos. Esta conclusión es coherente con la literatura revisada, en la cual se resalta la necesidad de evaluar simultáneamente métricas de desempeño y eficiencia computacional en entornos IoT.

Respecto a la caracterización de amenazas, vectores de ataque y patrones observables, la investigación permitió establecer que la detección ligera de malware puede sustentarse en un subconjunto reducido de características asociadas al tráfico de red y al comportamiento del

dispositivo. Variables como duración del flujo, número de paquetes, volumen de bytes, frecuencia de solicitudes DNS, uso de CPU, consumo de memoria y número de procesos activos pueden aportar señales relevantes para identificar comportamientos maliciosos sin requerir procesamiento intensivo. En consecuencia, la efectividad de un modelo no depende necesariamente de la cantidad de variables utilizadas, sino de su pertinencia, capacidad discriminativa y factibilidad de medición en dispositivos con recursos limitados.

La comparación entre modelos ligeros y enfoques tradicionales permitió evidenciar que la evaluación de soluciones para IoT debe superar la lectura centrada exclusivamente en métricas como accuracy o F1-score. Aunque estas métricas siguen siendo importantes, resultan insuficientes si no se analizan junto con indicadores como latencia, consumo de memoria, tamaño del modelo y consumo energético. Por ello, en escenarios con restricciones operativas, los modelos ligeros no deben entenderse como una alternativa de menor alcance, sino como una opción técnicamente pertinente cuando logran equilibrar capacidad de detección y eficiencia computacional.

Asimismo, la investigación evidenció una alta heterogeneidad en los estudios analizados, especialmente en términos de datasets, configuraciones experimentales, métricas reportadas y condiciones de validación. Esta diversidad limitó la posibilidad de realizar un metaanálisis cuantitativo riguroso, pero permitió desarrollar una síntesis narrativa estructurada y una comparación analítica orientada a identificar tendencias, brechas y criterios de decisión. En este sentido, el estudio aporta una lectura integradora de la literatura y evita asumir equivalencias directas entre investigaciones que no comparten las mismas condiciones metodológicas.

Como aporte principal, se formuló un marco de aplicación para orientar la selección de técnicas de aprendizaje automático ligero en la detección de malware en IoT y sistemas embebidos. Este marco integra seis componentes: caracterización del contexto operativo, identificación de la amenaza y vector de ataque, selección de características observables, elección de la técnica ligera, validación multicriterio y decisión de adopción. Su valor radica en que organiza la evidencia revisada en una ruta de análisis aplicable a futuras investigaciones, comparaciones y validaciones en hardware real.

Finalmente, se concluye que la ciberseguridad en entornos IoT requiere enfoques contextuales, donde la selección del modelo responda al tipo de amenaza, a las características observables disponibles y a las restricciones del entorno de despliegue. Desde esta perspectiva, el aprendizaje automático ligero no debe entenderse únicamente como una técnica de optimización, sino como un enfoque necesario para diseñar mecanismos de detección viables en sistemas distribuidos, heterogéneos y de recursos restringidos.

Recomendaciones

A partir de los hallazgos derivados de la revisión sistemática de literatura y del análisis comparativo de técnicas ligeras aplicadas a la detección de malware en entornos IoT, se formulan recomendaciones orientadas a fortalecer futuras investigaciones, validaciones experimentales y procesos de adopción en contextos reales con restricciones computacionales.

En primer lugar, se recomienda que los estudios futuros prioricen la validación en hardware real o, al menos, en entornos de simulación que declaren explícitamente restricciones de memoria, procesamiento, latencia y consumo energético. La literatura revisada evidencia que varios estudios reportan métricas de clasificación, como accuracy, precision, recall o F1-score, pero no siempre incorporan mediciones de eficiencia computacional, lo cual limita la transferibilidad de los resultados hacia dispositivos IoT, sistemas embebidos y nodos de borde (Fenanir et al., 2019; Javed et al., 2024; Sharmila & Nagapadma, 2023).

En segundo lugar, se recomienda adoptar diseños comparativos estructurados que incluyan modelos de referencia no optimizados y versiones ligeras obtenidas mediante selección de características, poda, cuantización, TinyML o aprendizaje federado. Este tipo de comparación permitiría valorar con mayor claridad el impacto de la optimización sobre el desempeño predictivo y la eficiencia operativa, evitando conclusiones basadas únicamente en métricas de exactitud (Warden & Situnayake, 2019; Rey et al., 2021).Asimismo, se sugiere explorar enfoques híbridos que integren características de tráfico de red y telemetría del dispositivo, especialmente en escenarios donde sea posible capturar ambas fuentes de información sin afectar

la estabilidad operativa del sistema. La combinación de variables como duración del flujo, volumen de bytes, frecuencia de solicitudes DNS, uso de CPU, consumo de memoria y número de procesos activos puede fortalecer la capacidad de detección frente a amenazas como botnets, cryptojacking, malware en memoria y comunicaciones de comando y control.

Desde una perspectiva aplicada, se recomienda que futuras implementaciones consideren criterios de escalabilidad, actualización periódica y mantenimiento del modelo. En entornos IoT heterogéneos, los patrones de tráfico y comportamiento pueden variar con el tiempo, por lo que resulta pertinente contemplar mecanismos de recalibración, actualización ligera o aprendizaje federado cuando las condiciones de conectividad, privacidad y capacidad computacional lo permitan. Esto facilitaría la adopción de soluciones de detección en organizaciones, instituciones públicas, entornos municipales o infraestructuras con recursos técnicos limitados.

Finalmente, se recomienda que las investigaciones posteriores documenten con mayor precisión los datasets utilizados, las condiciones de evaluación, los parámetros del modelo, las métricas reportadas y las amenazas a la validez. Esta práctica permitiría mejorar la comparabilidad entre estudios y reducir la ambigüedad metodológica que actualmente limita la generalización de algunos resultados, en coherencia con los principios de trazabilidad metodológica propios de las revisiones sistemáticas (Kitchenham & Charters, 2007; Page et al., 2021).

Referencias

Booth, A., Sutton, A., & Papaioannou, D. (2016). *Systematic approaches to a successful literature review* (2nd ed.). SAGE.

Canadian Institute for Cybersecurity. (2022). *CIC-MalMem-2022: Malware memory analysis dataset*. University of New Brunswick. https://www.unb.ca/cic/datasets/malmem-2022.html

David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., & Warden, P. (2020). *TensorFlow Lite Micro: Embedded machine learning on TinyML systems*. arXiv. https://arxiv.org/abs/2010.08678

Departamento Nacional de Planeación, Presidencia de la República de Colombia, & Ministerio de Tecnologías de la Información y las Comunicaciones. (2023). *Estrategia Nacional Digital de Colombia 2023–2026*. https://www.mintic.gov.co/portal/715/articles-334120_recurso_1.pdf

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. En *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 29–35). IEEE. https://doi.org/10.1109/SPW.2018.00013

Fagan, M., Marron, J., Brady, K. G., Jr., Cuthill, B. B., Megas, K. N., Herold, R., Lemire, D., & Hoehn, B. (2021). *IoT device cybersecurity guidance for the federal government: Establishing IoT device cybersecurity requirements* (NIST Special Publication 800-213). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-213

Fenanir, S., Semchedine, F., & Baadache, A. (2019). A machine learning-based lightweight intrusion detection system for the Internet of Things. *Revue d'Intelligence Artificielle, 33*(3), 203–211. https://doi.org/10.18280/ria.330306

Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020). *IoT-23: A labeled dataset with malicious and benign IoT network traffic* (Version 1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.4743746

Gough, D., Oliver, S., & Thomas, J. (2017). *An introduction to systematic reviews* (2nd ed.). SAGE.

Hasan, S. M. R., & Dhakal, A. (2024). *Obfuscated malware detection: Investigating real-world scenarios through memory analysis*. arXiv. https://arxiv.org/abs/2404.02372

Hernández-Sampieri, R., & Mendoza, C. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. McGraw-Hill.

Javed, A., Ehtsham, A., Jawad, M., Awais, M. N., Qureshi, A.-u.-H., & Larijani, H. (2024). Implementation of lightweight machine learning-based intrusion detection system on IoT devices of smart homes. *Future Internet, 16*(6), Article 200. https://doi.org/10.3390/fi16060200

Jouhari, M., & Guizani, M. (2024). *Lightweight CNN-BiLSTM based intrusion detection systems for resource-constrained IoT devices*. arXiv. https://doi.org/10.48550/arXiv.2406.02768

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report No. EBSE-2007-01). Keele University.

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: BoT-IoT dataset. *Future Generation Computer Systems, 100*, 779–796. https://doi.org/10.1016/j.future.2019.05.041

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2019). *Federated learning: Challenges, methods, and future directions*. arXiv. https://arxiv.org/abs/1908.07873

Madamidola, O. A., Ngobigha, F., & Ez-zizi, A. (2024). *Detecting new obfuscated malware variants: A lightweight and interpretable machine learning approach*. arXiv. https://arxiv.org/abs/2407.07918

McMahan, B., Moore, E., Ramage, D., Hampson, S., & Aguera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. En *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (Vol. 54, pp. 1273–1282). PMLR. https://proceedings.mlr.press/v54/mcmahan17a.html

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2018). N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing, 17*(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

Moustafa, N., Keshk, M., Debie, E., & Janicke, H. (2020). *Federated TON_IoT Windows datasets for evaluating AI-based security applications*. arXiv. https://arxiv.org/abs/2010.08522

Naciones Unidas. (s. f.). *Influencia de las tecnologías digitales*. https://www.un.org/es/un75/impact-digital-technologies

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 23*(3), 1622–1658. https://doi.org/10.1109/COMST.2021.3075439

Observatorio Colombiano de Ciencia y Tecnología. (2024). *Informe de gestión 2023*. https://ocyt.org.co/wp-content/uploads/2024/04/INFORME-DE-GESTION-2023.pdf

Osman, A., Abid, U., Gemma, L., Perotto, M., & Brunelli, D. (2021). *TinyML platforms benchmarking*. arXiv. https://arxiv.org/abs/2112.01319

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ, 372*, Article n71. https://doi.org/10.1136/bmj.n71

Petticrew, M., & Roberts, H. (2006). *Systematic reviews in the social sciences: A practical guide*. Blackwell.

Rey, V., Sánchez Sánchez, P. M., Huertas Celdrán, A., Bovet, G., & Jaggi, M. (2021). *Federated learning for malware detection in IoT devices*. arXiv. https://doi.org/10.48550/arXiv.2104.09994

Sáez-de-Cámara, X., Flores, J. L., Arellano, C., Urbieta, A., & Zurutuza, U. (2023). *Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks*. arXiv. https://arxiv.org/abs/2303.15986

Sánchez Sánchez, P. M., Huertas Celdrán, A., Schenk, T., Iten, A. L. B., Bovet, G., Martínez Pérez, G., & Stiller, B. (2022). *Studying the robustness of anti-adversarial federated learning models detecting cyberattacks in IoT spectrum sensors*. arXiv. https://arxiv.org/abs/2202.00137

Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2020). *NetFlow datasets for machine learning-based network intrusion detection systems*. arXiv. https://arxiv.org/abs/2011.09144

Sarhan, M., Layeghy, S., & Portmann, M. (2021). *Towards a standard feature set for network intrusion detection system datasets*. arXiv. https://arxiv.org/abs/2101.11315

Sharmila, B. S., & Nagapadma, R. (2023). Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset. *Cybersecurity, 6*, Article 41. https://doi.org/10.1186/s42400-023-00178-5

Verma, A., & Ranga, V. (2023). Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications, 130*, 2287–2313. https://doi.org/10.1007/s11277-023-10347-1

Warden, P., & Situnayake, D. (2019). *TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers*. O'Reilly Media.

Xenofontos, C., Zografopoulos, I., Konstantinou, C., Jolfaei, A., Khan, M. K., & Choo, K.-K. R. (2021). *Consumer, commercial and industrial IoT (in)security: Attack taxonomy and case studies*. arXiv. https://doi.org/10.48550/arXiv.2105.06612

Apéndices

Apéndice A. Cadenas de búsqueda

Las cadenas de búsqueda fueron diseñadas en inglés y español, considerando los conceptos centrales de la investigación: IoT, malware, aprendizaje automático ligero, detección de intrusiones, técnicas de optimización y modelos embebidos. Para su construcción se emplearon operadores booleanos y términos equivalentes adaptados a las bases de datos consultadas, siguiendo criterios de trazabilidad propios de revisiones sistemáticas de literatura (Kitchenham & Charters, 2007; Page et al., 2021).

IEEE Xplore / Scopus / ACM

```
("Internet of Things" OR IoT OR "embedded systems") AND
(malware OR "intrusion detection" OR botnet OR attack) AND
("lightweight" OR pruning OR quantization OR "feature selection" OR TinyML OR
"model compression" OR "federated learning") AND
("machine learning" OR "deep learning") AND
(detection OR classification)
```

ScienceDirect

```
(IoT OR "Internet of Things") AND
(malware detection OR intrusion detection OR botnet) AND
(lightweight models OR model compression OR TinyML OR quantization OR
pruning) AND
(machine learning OR deep learning)
```

Español

```
("Internet de las Cosas" OR IoT) AND
(malware OR "detección de intrusos") AND
("modelos ligeros" OR cuantización OR poda OR TinyML) AND
("aprendizaje automático")
```

Apéndice B. Matriz de extracción de datos

La Tabla B1 presenta la matriz de extracción de datos aplicada a los estudios seleccionados en la revisión sistemática de literatura. Esta matriz permitió sistematizar la información técnica y metodológica necesaria para el análisis comparativo, incluyendo técnica analizada, modelo o enfoque empleado, dataset o escenario de evaluación, métricas reportadas, aspectos de eficiencia computacional y hallazgos principales. Su propósito no es presentar resultados experimentales propios, sino organizar de manera trazable la evidencia reportada por los autores revisados para facilitar la comparación entre enfoques de aprendizaje automático ligero aplicados a la detección de malware en dispositivos IoT y sistemas embebidos.

Tabla B1

Matriz de extracción de datos de los estudios seleccionados

Estudio	Referencia	Técnica analizada	Modelo o enfoque	Dataset / escenario	Métricas reportadas	Eficiencia / recursos	Hallazgo principal
E1	Meidan et al. (2018)	Detección de botnets IoT	Autoencoders profundos	N-BaIoT / dispositivos IoT infectados con Mirai y BASHLITE	Accuracy, detección de anomalías	Inferencia sobre tráfico de red	Evidencia que los autoencoders pueden detectar tráfico anómalo generado por botnets IoT.
E2	Doshi et al. (2018)	Detección de ataques DDoS en IoT	ML clásico con características de tráfico	Tráfico IoT / dispositivos de consumo y gateways domésticos	Accuracy, precision, recall	Bajo costo relativo frente a modelos complejos	Demuestra que características de tráfico IoT pueden apoyar la detección temprana de ataques DDoS mediante modelos de bajo costo.
E3	Koroniotis et al. (2019)	Dataset para botnets IoT	Generación y evaluación de dataset	BoT-IoT	Métricas de clasificación usadas en estudios derivados	Dataset orientado a tráfico botnet	Aporta un conjunto de datos ampliamente usado para evaluar IDS en escenarios IoT con tráfico botnet.

Estudio	Referencia	Técnica analizada	Modelo o enfoque	Dataset / escenario	Métricas reportadas	Eficiencia / recursos	Hallazgo principal
E4	Fenanir et al. (2019)	IDS ligero basado en ML	ML clásico con selección de características	KDD99, NSL-KDD y UNSW-NB15	Accuracy, precision, recall, F1-score	Reducción de complejidad mediante selección de características	Propone un IDS ligero para IoT orientado a reducir el costo computacional sin perder capacidad de detección.
E5	Garcia et al. (2020)	Dataset IoT-23	Dataset etiquetado	IoT-23 / tráfico benigno y malicioso de dispositivos IoT	Métricas usadas por estudios derivados	Tráfico real etiquetado de malware IoT	Aporta un dataset relevante para analizar tráfico malicioso y benigno en escenarios IoT.
E6	Moustafa et al. (2020)	Dataset IoT/IIoT	Dataset y telemetría	TON_IoT / red, sistema operativo y telemetría IoT/IIoT	Métricas de clasificación usadas en estudios derivados	Soporta análisis de red, sistema y telemetría	Proporciona datos heterogéneos para evaluar ciberseguridad basada en IA en IoT e IIoT.
E7	Canadian Institute for Cybersecurity (2022)	Malware en memoria	Dataset especializado	CIC-MalMem-2022 / malware ofuscado en memoria	Métricas de clasificación usadas en estudios derivados	Relevante para análisis host-based	Proporciona una base para evaluar malware ofuscado mediante características extraídas de memoria.
E8	Sharmila & Nagapadma (2023)	Cuantización para IDS IoT	Autoencoder cuantizado	RT-IoT2022 / dispositivos IoT con recursos restringidos	Accuracy, precision, recall, F1-score	Reducción de complejidad e inferencia en borde	Evidencia que la cuantización puede contribuir a la detección de anomalías en dispositivos IoT con recursos limitados.
E9	Rey et al. (2021)	Aprendizaje federado para malware IoT	FL supervisado y no supervisado	N-BaIoT / dispositivos IoT afectados por malware	Accuracy, F1-score y métricas comparativas	Preservación de privacidad y reducción de centralización de datos	Presenta el aprendizaje federado como alternativa para detección de malware IoT sin centralizar datos locales.
E10	Nguyen et al. (2021)	Aprendizaje federado en IoT	Revisión de enfoques FL	Escenarios IoT distribuidos	Métricas reportadas por estudios revisados	Privacidad, comunicación, escalabilidad y heterogeneidad	Sistematiza aplicaciones y retos del aprendizaje federado en IoT, especialmente privacidad, comunicación y heterogeneidad de dispositivos.

Estudio	Referencia	Técnica analizada	Modelo o enfoque	Dataset / escenario	Métricas reportadas	Eficiencia / recursos	Hallazgo principal
E11	Verma & Ranga (2023)	IDS con ML para IoT	Clasificadores ML clásicos	CIDDS-001, UNSW-NB15, NSL-KDD y Raspberry Pi	Accuracy, precision, recall, F1-score y tiempo de respuesta	Evalúa tiempo de respuesta en hardware IoT	Aporta criterios prácticos para seleccionar clasificadores según desempeño y restricciones de aplicación.
E12	Sarhan et al. (2021)	Selección de características para IDS IoT	ML con análisis de importancia de características	UNSW-NB15, CSE-CIC-IDS2018, ToN-IoT y variantes NetFlow	Accuracy y desempeño por subconjuntos de características	Reducción de costo computacional y almacenamiento	Evidencia que un subconjunto reducido de características puede mantener desempeño cercano al óptimo en IDS para IoT.
E13	Sarhan et al. (2020)	Datasets NetFlow para NIDS	Conversión y estandarización de características NetFlow	NF-UNSW-NB15, NF-BoT-IoT, NF-ToN-IoT y NF-CSE-CIC-IDS2018	Métricas de clasificación en evaluación preliminar	Características de flujo más fáciles de extraer	Propone datasets NetFlow para mejorar comparabilidad y reducir complejidad en NIDS basados en ML.
E14	Jouhari & Guizani (2024)	IDS ligero para IoT	CNN-BiLSTM ligero	UNSW-NB15 / dispositivos IoT con recursos limitados	Accuracy y métricas de clasificación binaria y multiclase	Diseño orientado a reducción de complejidad	Propone una arquitectura híbrida ligera para mantener desempeño de clasificación con menor complejidad computacional.
E15	Xenofontos et al. (2021)	Taxonomía de ataques IoT	Revisión de ataques, vulnerabilidades, impactos y mitigaciones	IoT de consumo, comercial e industrial	No aplica como evaluación experimental propia	Clasificación por capas, superficie de ataque e impacto	Sistematiza ataques IoT y mecanismos de defensa útiles para relacionar amenazas, vectores y controles.
E16	David et al. (2020)	TinyML y despliegue embebido	TensorFlow Lite Micro	Sistemas embebidos y microcontroladores	Evaluación de memoria, rendimiento y sobrecarga	Ejecución eficiente en memoria limitada	Presenta un marco de inferencia para aprendizaje automático en sistemas embebidos con recursos severamente restringidos.
E17	Osman et al. (2021)	Benchmarking de plataformas TinyML	Comparación de frameworks TinyML	Arduino Nano BLE y STM32-NucleoF401RE	Métricas de rendimiento y consumo	Bajo consumo energético y ejecución local	Compara plataformas TinyML y aporta criterios para seleccionar frameworks en dispositivos de bajo consumo.

Estudio	Referencia	Técnica analizada	Modelo o enfoque	Dataset / escenario	Métricas reportadas	Eficiencia / recursos	Hallazgo principal
E18	Li et al. (2019)	Retos del aprendizaje federado	Revisión de métodos FL	Dispositivos remotos y datos distribuidos	No aplica como evaluación experimental única	Comunicación, heterogeneidad, privacidad y escalabilidad	Identifica retos técnicos del aprendizaje federado relevantes para escenarios IoT distribuidos.
E19	McMahan et al. (2017)	Aprendizaje federado	Algoritmo Federated Averaging	Dispositivos distribuidos	Accuracy y rondas de comunicación en tareas distribuidas	Reducción de transferencia de datos centralizados	Introduce un enfoque base para entrenamiento federado eficiente sobre datos distribuidos.
E20	Sáez-de-Cámara et al. (2023)	Detección de anomalías en IoT/IIoT	Aprendizaje federado con clustering	Testbed IoT/IIoT heterogéneo	Métricas de detección de anomalías	Reduce aislamiento de datos y sobrecarga de red	Propone una arquitectura federada con agrupamiento para IDS en redes IoT/IIoT heterogéneas.
E21	Sánchez Sánchez et al. (2022)	Detección federada de ciberataques	FL robusto frente a ataques adversarios	Sensores IoT de espectro / escenarios distribuidos	Métricas de detección y robustez	Privacidad, robustez y distribución de entrenamiento	Analiza la robustez de modelos federados ante ataques adversarios en sensores IoT con recursos restringidos.
E22	Madamidola et al. (2024)	Malware ofuscado	ML ligero e interpretable	CIC-MalMem-2022	Accuracy y tiempo de procesamiento	Selección de características e interpretabilidad	Evalúa detección de variantes de malware ofuscado con modelos ligeros e interpretables basados en pocas características.
E23	Hasan & Dhakal (2024)	Detección de malware ofuscado	ML sobre análisis de memoria	CIC-MalMem-2022	Métricas de clasificación por algoritmos ML	Enfoque de bajo costo sobre características de memoria	Evalúa algoritmos de aprendizaje automático para detectar malware ofuscado mediante análisis de memoria.
E24	Fagan et al. (2021)	Requisitos de ciberseguridad IoT	Guía técnica NIST SP 800-213	Dispositivos IoT y requisitos de seguridad	No aplica como evaluación experimental	Requisitos de seguridad, gestión y exposición	Establece criterios de ciberseguridad para dispositivos IoT, útiles para contextualizar riesgos y restricciones operativas.
E25	Warden & Situnayake (2019)	TinyML para sistemas embebidos	Inferencia local con modelos pequeños	Microcontroladores y dispositivos de bajo consumo	No aplica como evaluación experimental única	Memoria, latencia, energía e inferencia local	Fundamenta el uso de TinyML para ejecutar modelos en dispositivos de bajo recurso sin dependencia

Estudio	Referencia	Técnica analizada	Modelo o enfoque	Dataset / escenario	Métricas reportadas	Eficiencia / recursos	Hallazgo principal
							permanente de la nube.

Nota. La matriz resume la información técnica y metodológica extraída de los estudios seleccionados, incluyendo técnica analizada, modelo o enfoque, dataset o escenario, métricas reportadas, aspectos de eficiencia computacional y hallazgos principales. Cuando un estudio no reporta de forma explícita métricas de eficiencia, se registran los criterios de eficiencia discutidos en el artículo. Elaboración propia a partir de la revisión sistemática de literatura.

Apéndice C. Matriz de evaluación de calidad

La evaluación de calidad se realizó mediante una escala de 0 a 2 por criterio, permitiendo identificar la robustez metodológica de los estudios seleccionados.

Tabla C1

Matriz de evaluación de calidad de los estudios incluidos

Estudio	Referencia	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Puntaje total
E1	Meidan et al. (2018)	2	2	2	1	2	2	1	2	1	2	17
E2	Doshi et al. (2018)	2	1	2	1	2	1	1	1	1	2	14
E3	Koroniotis et al. (2019)	2	2	2	1	2	2	1	2	1	2	17
E4	Fenanir et al. (2019)	2	2	2	2	2	1	1	2	1	2	17
E5	Garcia et al. (2020)	2	2	1	1	1	2	1	2	1	2	15
E6	Moustafa et al. (2020)	2	2	1	1	1	2	1	2	1	2	15
E7	Canadian Institute for Cybersecurity (2022)	2	2	1	1	1	2	1	2	1	2	15
E8	Sharmila & Nagapadma (2023)	2	2	2	2	2	2	1	2	2	2	19
E9	Rey et al. (2021)	2	2	2	2	2	1	1	2	2	2	18
E10	Nguyen et al. (2021)	2	2	1	2	2	1	1	1	2	2	16
E11	Verma & Ranga (2023)	2	2	2	2	2	2	1	2	2	2	19
E12	Sarhan et al. (2021)	2	2	2	2	2	2	1	2	2	2	19
E13	Sarhan et al. (2020)	2	2	1	2	2	2	1	2	1	2	17
E14	Jouhari & Guizani (2024)	2	2	2	2	2	1	1	1	1	2	16
E15	Xenofontos et al. (2021)	2	2	1	1	2	1	1	1	2	2	15
E16	David et al. (2020)	2	2	1	2	2	2	1	2	1	2	17
E17	Osman et al. (2021)	2	2	1	2	2	2	1	2	1	2	17
E18	Li et al. (2019)	2	2	1	2	2	1	1	1	2	2	16
E19	McMahan et al. (2017)	2	2	2	2	2	2	1	2	1	2	18
E20	Sáez-de-Cámara et al. (2023)	2	2	2	2	2	1	1	1	2	2	17
E21	Sánchez Sánchez et al. (2022)	2	2	2	2	2	1	1	1	2	2	17
E22	Madamidola et al. (2024)	2	2	2	1	2	1	1	1	1	2	15
E23	Hasan & Dhakal (2024)	2	2	2	1	2	1	1	1	1	2	15
E24	Fagan et al. (2021)	2	2	1	2	1	2	1	2	2	2	17
E25	Warden & Situnayake (2019)	2	2	1	2	1	2	1	2	2	2	17

Nota. C1 = claridad del objetivo; C2 = descripción del dataset o escenario; C3 = métricas de eficacia reportadas; C4 = métricas de eficiencia o recursos; C5 = comparación con otros enfoques; C6 = descripción de configuración técnica o procedimiento; C7 = análisis de amenazas a la validez; C8 = posibilidad de replicabilidad; C9 = discusión de limitaciones; C10 = coherencia entre conclusiones y evidencia. Cada criterio se calificó en una escala de 0 a 2, donde 0 = no cumple, 1 = cumple parcialmente y 2 = cumple completamente. El puntaje máximo posible es de 20 puntos. Elaboración propia a partir de los criterios definidos para la revisión sistemática de literatura.