

Sistema embebido autónomo para auditoría de seguridad inalámbrica mediante TinyML en el borde: clasificador de árbol de decisión sobre ESP32-S3

Autonomous embedded system for wireless security auditing using TinyML at the edge: decision tree classifier on ESP32-S3

AUTORES

Jefferson Wladimir Valdez Coello
<https://orcid.org/0009-0009-7132-3644>
Instituto Internacional de Estudios en Seguridad Global (INISEG)
aldezcoello@yahoo.es

Jussen Paúl Facuy Delgado
<https://orcid.org/0000-0003-1138-4823>
Universidad de Guayaquil, Ecuador
jussen.facuyd@ug.edu.ec
Universidad Agraria del Ecuador
jfacuy@uagraria.edu.ec

Frederick Alexander Farfán Jijón
<https://orcid.org/0009-0007-8249-4575>
frederick.farfán@gmail.com

Como citar:

Valdez Coello, J. W., Facuy Delgado, J. P., & Farfán Jijón, F. A. (2026). Sistema embebido autónomo para auditoría de seguridad inalámbrica mediante TinyML en el borde: clasificador de árbol de decisión sobre ESP32-S3. *Revista Internacional de Investigación y Desarrollo Global (RIIDG)*, 5(4), 26-49. <https://doi.org/10.64041/riidg.v5i4.76>

V. 5, N. 4, Año (2026), Pág. 26-49

Fecha de recepción: 20 de agosto de 2026.

Fecha de aceptación: 15 de octubre de 2026

Fecha de publicación: 22 de octubre de 2026

Resumen

La expansión del Internet de las Cosas (IoT) y de las redes de área amplia y baja potencia (LPWAN) amplía la superficie de ataque, mientras que las herramientas de auditoría inalámbrica de bolsillo (Bruce, Marauder, NEMO) siguen siendo manuales y reactivas. **Objetivo.** Evaluar la viabilidad de ejecutar un modelo de inteligencia artificial en el borde, sobre un microcontrolador de bajo costo, para clasificar de forma autónoma la configuración de seguridad de redes inalámbricas. **Métodos.** Se construyó un prototipo sobre el M5Stack Cardputer-Adv (ESP32-S3) con módulo LoRa SX1262. Se recolectaron 697 registros (333 Wi-Fi, 301 BLE, 63 LoRa) en Guayaquil, Ecuador, se derivaron 17 características y se compararon un árbol de decisión, un bosque aleatorio y una red neuronal pequeña. El modelo seleccionado se transcribió a C++ nativo y se validó contra la versión en Python. **Resultados.** El árbol de decisión (profundidad máxima 5) obtuvo exactitud de 99,0 %, precisión de 87,5 %, exhaustividad de 100 % y F1 de 93,3 %, con latencia de 0,07 ms y 2,3 KB de tamaño. La coincidencia Python-C++ fue del 100 % en 25 casos. El firmware completo usó 16,1 % de la RAM y 27,6 % de la Flash. En 25 redes reales, 12 fueron catalogadas como amenaza, pero solo una correspondía a una red abierta. **Conclusiones.** El enfoque es técnicamente viable en hardware de bolsillo; la validación de campo muestra que la regla de «señal sospechosamente fuerte» genera falsos positivos y que la evaluación de LoRa/BLE en tiempo real queda como trabajo futuro.

Palabras clave: TinyML, IA en el borde, auditoría Wi-Fi, IoT, LPWAN, LoRaWAN, ESP32-S3, árbol de decisión, ciberseguridad.

Abstract

The expansion of the Internet of Things (IoT) and low-power wide-area networks (LPWAN) broadens the attack surface, while pocket-sized wireless auditing tools (Bruce, Marauder, NEMO) remain manual and reactive. Objective. To assess the feasibility of running an artificial intelligence model at the edge, on a low-cost microcontroller, to autonomously classify the security configuration of wireless networks. Methods. A prototype was built on the M5Stack Cardputer-Adv (ESP32-S3) with an SX1262 LoRa module. A total of 697 records (333 Wi-Fi, 301 BLE, 63 LoRa) were collected in Guayaquil, Ecuador; 17 features were derived; and a decision tree, a random forest, and a small neural network were compared. The selected model was transcribed into native C++ and validated against the Python version. Results. The decision tree (maximum depth 5) achieved an accuracy of 99.0%, precision of 87.5%, recall of 100%, and F1 of 93.3%, with a latency of 0.07 ms and a size of 2.3 KB. The Python-C++ agreement was 100% across 25 cases. The complete firmware used 16.1% of RAM and 27.6% of Flash. Out of 25 real networks, 12 were classified as a threat, but only one corresponded to an open network. Conclusions. The approach is technically feasible on pocket-sized hardware; field validation shows that the "suspiciously strong signal" rule generates false positives, and that real-time evaluation of LoRa/BLE remains future work.

Keywords: TinyML, edge AI, Wi-Fi auditing, IoT, LPWAN, ESP32-S3, decision tree, cybersecurity.

1. Introducción

El Internet de las Cosas (IoT) designa el ecosistema de dispositivos físicos interconectados que recopilan, procesan e intercambian datos a través de redes de comunicación, y ha permeado sectores tan diversos como la salud, la industria, el transporte y la agricultura (Atzori et al., 2010; Al-Fuqaha et al., 2015). Las proyecciones del sector estiman que en 2030 habrá más de 29.000 millones de dispositivos IoT conectados (Vailshery, 2023); aunque las cifras varían según la fuente, todas apuntan a una superficie de exposición sin precedentes para la ciberseguridad. En América Latina, y en particular en Ecuador, las redes de área amplia y baja potencia (LPWAN), con LoRaWAN a la cabeza, han mostrado un avance sostenido en aplicaciones de ciudades inteligentes, medición remota de servicios públicos y agricultura de precisión.

Este crecimiento no ha sido acompañado por una evolución equivalente de la seguridad. Butun et al. (2020) y Roman et al. (2013) coinciden en que la heterogeneidad de los dispositivos, la escasez de recursos computacionales y la ausencia frecuente de mecanismos de actualización de firmware dificultan la aplicación de defensas convencionales. Los nodos IoT, concebidos para operar durante años con una única fuente de energía, no pueden incorporar esquemas criptográficos pesados ni protocolos de autenticación mutua complejos, lo que origina una tensión estructural entre seguridad y eficiencia energética (Sicari et al., 2015; Khan y Salah, 2018). Las consecuencias son tangibles: la botnet Mirai mostró cómo dispositivos con credenciales por defecto pueden convertirse en una plataforma de ataques de denegación de servicio a gran escala (Antonakakis et al., 2017; Kolias et al., 2017), y la cadena de suministro de los dispositivos añade riesgos de falsificación y de trazabilidad (Yang et al., 2017).

Las LPWAN surgieron para conectar grandes cantidades de sensores distribuidos con un consumo energético mínimo, y priorizan el alcance kilométrico y la vida útil de la batería por encima del ancho de banda (Raza et al., 2017; Centenaro et al., 2016), lo que condiciona tanto su rendimiento como su postura de seguridad. LoRaWAN opera en bandas sub-GHz no licenciadas y se basa en la modulación de espectro ensanchado por chirp (CSS), que aporta robustez frente al ruido (Augustin et al., 2016); sin embargo, sus limitaciones de diseño (ciclo de trabajo restringido, baja tasa de transmisión y acceso al medio de tipo ALOHA) abren superficies de ataque poco contempladas (Adelantado et al.,

2017). La literatura documenta ataques de repetición asociados a la reutilización de nonces y a contadores de trama mal gestionados tras un reinicio del nodo (Eldefrawy et al., 2019; Yang et al., 2018; Butun et al., 2019), interferencia selectiva y reactiva con hardware de bajo costo, como los transceptores de la familia SX1262 (Aras et al., 2017; Semtech Corporation, 2023), y escucha pasiva, ya que, aunque la carga útil se cifra con AES-128, la cabecera viaja sin protección y habilita el análisis de tráfico (LoRa Alliance, 2020). Banerjee et al. (2012) advierten que, en sistemas ciberfísicos críticos, la seguridad no debe añadirse a posteriori sino integrarse desde el diseño, un principio especialmente pertinente en ciudades inteligentes donde conviven miles de nodos LoRa.

En las redes Wi-Fi, los mecanismos de protección evolucionaron desde esquemas débiles como WEP hacia WPA2 y WPA3, pero la coexistencia de dispositivos heredados y de redes abiertas sigue generando puntos de exposición, y los propios estándares modernos han mostrado debilidades de implementación, como los ataques de reinstalación de claves contra WPA2 (Vanhoeft y Piessens, 2017) y las fallas en el handshake de WPA3 (Vanhoeft y Ronen, 2020). La identificación automatizada de configuraciones inseguras es, por tanto, una necesidad práctica en entornos donde coexisten Wi-Fi, Bluetooth de baja energía, LoRa, Zigbee y NB-IoT.

Pese a estos riesgos, la auditoría de seguridad inalámbrica sigue dependiendo de operadores humanos especializados, y las herramientas disponibles presentan tres limitaciones estructurales. Primero, exigen que el especialista decida qué redes evaluar, qué técnicas aplicar y en qué orden, lo que introduce sesgos de criterio, fatiga y lentitud de respuesta en entornos dinámicos. Segundo, su portabilidad es insuficiente: las evaluaciones convencionales requieren portátiles, adaptadores especializados y antenas externas, lo que resulta impráctico en campo y en infraestructuras distribuidas como redes de sensores agrícolas o sistemas de medición urbana. Tercero, la mayoría se especializa en un único protocolo, habitualmente Wi-Fi de 2,4 GHz, y desatiende la heterogeneidad de los entornos IoT actuales. En los últimos años, firmwares comunitarios de código abierto para microcontroladores ESP32, como Bruce (pr3y, s. f.), ESP32 Marauder (justcallmekoko, s. f.) y NEMO (n0xa, s. f.), democratizaron el acceso a técnicas antes reservadas a equipos costosos y voluminosos, pero comparten una limitación común: operan bajo un paradigma estrictamente manual y reactivo, ejecutan la orden recibida sin

analizar el entorno, priorizar objetivos ni adaptar su comportamiento, y no incorporan ningún modelo de aprendizaje automático que interprete las métricas de radiofrecuencia. Automatizar esa decisión no equivale a prescindir del criterio humano: la adopción de la IA en otras prácticas profesionales, como el diseño gráfico, muestra que su mayor valor aparece cuando actúa como herramienta de apoyo que amplía la exploración de alternativas y reduce el tiempo en tareas repetitivas, sin sustituir el juicio del profesional (Facuy Delgado et al., 2025).

El aprendizaje automático se ha convertido en un instrumento esencial de la ciberseguridad, con capacidades de detección que superan a los enfoques basados en reglas estáticas (Sarker et al., 2020; Buczak y Guven, 2016). Se ha aplicado a la detección de intrusiones y de tráfico anómalo en redes cableadas e inalámbricas, incluidas las redes IoT, mediante modelos profundos (Shone et al., 2018; Diro y Chilamkurti, 2018), autoencoders (Meidan et al., 2018; Mirsky et al., 2018) y conjuntos de datos específicos para redes 802.11 (Kolias et al., 2016). Sin embargo, estos trabajos se ejecutan típicamente en servidores o pasarelas, y la literatura previene sobre los riesgos de usar aprendizaje automático en seguridad sin una evaluación realista de sus supuestos, sus conjuntos de datos y sus tasas de error (Sommer y Paxson, 2010; Arp et al., 2022). Una preocupación similar se ha planteado en los marcos de predicción basados en redes neuronales, que suelen priorizar la exactitud y descuidar la seguridad, la transparencia y la trazabilidad de los datos, y que a menudo dependen de infraestructuras centralizadas (Facuy et al., 2026).

La inteligencia artificial en el borde propone trasladar la inferencia desde la nube hacia los dispositivos que generan los datos, con ventajas de latencia, ancho de banda y privacidad, y con autonomía operativa donde la conectividad es intermitente (Shi et al., 2016; Zhou et al., 2019; Deng et al., 2020). TinyML es su expresión más extrema: lleva modelos a microcontroladores con presupuestos energéticos del orden de los milivatios y memorias medidas en kilobytes (Warden y Situnayake, 2019), apoyándose en marcos como TensorFlow Lite Micro (David et al., 2021), en referencias de evaluación como MLPerf Tiny (Banbury et al., 2021) y en técnicas de compresión y cuantización (Han et al., 2016; Jacob et al., 2018). Las redes neuronales profundas resultan inviables en entornos con restricciones severas (Abiodun et al., 2018), por lo que el diseño favorece modelos compactos como los árboles de decisión. Esta orientación al uso eficiente de recursos

dialoga con la preocupación creciente por el consumo de energía y agua de los sistemas de IA, para los que se ha propuesto, por ejemplo, optimizar la comunicación entre agentes (Facuy Delgado y Arcos-Jacome, 2026). Pese a ello, Ray (2022) señala que la aplicación de TinyML a la ciberseguridad, y en particular a la auditoría activa de redes, sigue siendo un terreno escasamente explorado, un vacío especialmente notable en el contexto latinoamericano.

La plataforma M5Stack Cardputer-Adv, basada en el ESP32-S3, integra un procesador de doble núcleo con instrucciones vectoriales orientadas al aprendizaje automático, teclado físico, pantalla, ranura MicroSD y conectividad inalámbrica en un formato de bolsillo (M5Stack Technology Co., 2024), y ofrece un margen de cómputo y memoria superior al de las placas ESP32 convencionales sobre las que operan los firmwares citados. En este contexto, la pregunta que orienta el trabajo es si resulta viable implementar un modelo de inteligencia artificial en el borde, dentro de un dispositivo embebido de bajo costo, que permita automatizar la auditoría de seguridad inalámbrica con una efectividad comparable a la de las herramientas manuales convencionales.

El objetivo general es desarrollar e implementar un prototipo de auditoría de seguridad portátil sobre el ESP32-S3 que ejecute auditorías automatizadas, utilice TinyML para tomar decisiones tácticas autónomas en tiempo real y permita proponer contramedidas frente a las vulnerabilidades identificadas. Para lograrlo, se diseñó una arquitectura de hardware con el Cardputer-Adv y el módulo Cap LoRa-1262 para capturar señales Wi-Fi, Bluetooth y LoRa; se construyó un dataset de radiofrecuencia con RSSI, tipos de cifrado, densidad de tráfico y canales; se entrenó, optimizó y convirtió a C++ nativo un clasificador ligero que predice el nivel de riesgo de cada red; se desarrolló el firmware en C++ con PlatformIO y el framework Arduino; se evaluó el agente en tasa de acierto, tiempo de respuesta, precisión y consumo de recursos; y se formularon lineamientos de defensa frente a dispositivos de ataque portátiles impulsados por IA.

El trabajo se justifica en tres dimensiones. En lo teórico, aporta evidencia empírica sobre la viabilidad de ejecutar clasificadores de vulnerabilidades en microcontroladores, en la intersección entre inteligencia artificial y ciberseguridad. En lo práctico, responde a la necesidad de herramientas portátiles, autónomas y multiprotocolo para ciudades inteligentes e infraestructuras críticas. En lo tecnológico, añade una capa de inteligencia a

una plataforma de bajo costo y transforma un dispositivo de ejecución manual en un agente adaptable al entorno de radiofrecuencia. El alcance se circunscribe a la fase de detección y clasificación, en entornos controlados y autorizados, con fines académicos y sin ejecutar acciones sobre redes de terceros: el agente implementa en tiempo real el ciclo de escaneo, inferencia, visualización y registro sobre redes Wi-Fi, mientras que BLE y LoRa se incorporaron a la construcción del dataset (697 registros) y su integración en tiempo real, junto con la geolocalización GNSS, queda como trabajo futuro. Las secciones siguientes describen los métodos, presentan los resultados y discuten sus alcances y limitaciones.

2. Metodos

2.1. Diseño de la investigación

Se adoptó una investigación aplicada de carácter experimental, que combina elementos descriptivos (estado del arte y caracterización del problema) con elementos de investigación tecnológica (diseño, desarrollo y evaluación del prototipo). El diseño es cuasi-experimental: se manipulan variables independientes (tipo de red, nivel de cifrado, protocolo) y se miden variables dependientes (tasa de acierto, tiempo de respuesta, consumo de recursos). El desarrollo siguió el modelo de construcción de prototipos: análisis de requisitos, diseño de arquitectura, implementación, pruebas y evaluación en iteraciones sucesivas. Las técnicas de auditoría se abordaron exclusivamente desde una perspectiva defensiva y de evaluación.

2.2. Arquitectura del sistema

El sistema consta de tres capas:

- **Hardware:** M5Stack Cardputer-Adv (ESP32-S3), módulo LoRa SX1262 con antena de 868 MHz y tarjeta MicroSD para el registro permanente de resultados.
- **Software:** firmware en C++ (PlatformIO, framework Arduino) que orquesta el ciclo de escaneo, inferencia, presentación y registro, con módulos separados de configuración, extracción de características, motor de inferencia, interfaz de usuario y logging.
- **Inteligencia:** modelo de aprendizaje automático entrenado y optimizado para ejecución embebida.

Flujo operativo: el dispositivo escanea el entorno de radiofrecuencia, extrae las características de cada red, las normaliza con un escalador previamente ajustado, ejecuta la inferencia para clasificar cada red como normal o anómala, muestra el resultado en pantalla y lo registra en la MicroSD. Todo el proceso es local. El agente ofrece tres modos: manual (escaneo a demanda con la tecla ENTER), automático (escaneo cada 10 s con light-sleep entre ciclos) y continuo (monitorización permanente).

2.3. Recolección y preparación de datos

La captura se realizó en un entorno urbano controlado de Guayaquil, Ecuador, con el propio Cardputer. Se generaron 12 archivos CSV: 5 de escaneos exclusivos de Wi-Fi y 7 multiespectrales (Wi-Fi, BLE y LoRa). El conjunto consolidado reúne 697 registros: 333 redes Wi-Fi (47,8 %), 301 dispositivos BLE (43,2 %) y 63 registros LoRa (9,0 %) (Figura 1). La preparación incluyó la unificación de los esquemas de captura, el tratamiento de valores ausentes y la normalización de las métricas de intensidad de señal.

Figura 1. Composición del dataset consolidado por protocolo.

El dataset es 48 % Wi-Fi, 43 % BLE y solo 9 % LoRa
697 registros de 12 sesiones de escaneo en Guayaquil



2.4. Ingeniería de características y etiquetado

Se construyeron 17 características normalizadas al rango [0, 1] y convertidas a punto flotante de precisión simple, agrupadas en nueve categorías:

Categoría	Características
Protocolo	protocol_enc (Wi-Fi=0, BLE=1, LoRa=2)
Intensidad de señal	rssi_norm (Min-Max sobre [-95, -24] dBm), rssi_cat (muy débil a fuerte)
Cifrado	encryption_risk (Open=1,0; WEP=0,8; WPA=0,1; WPA2/WPA3=0,0)

Canal	is_standard_channel (1, 6 u 11), channel_occ_norm (congestión), channel_norm (canal/14)
Identificador	ssid_entropy_norm (entropía de Shannon), ssid_length_norm, is_hidden
Persistencia	persistence_norm (fracción de escaneos recientes en que aparece)
Fabricante	oui_enc_norm (primeros 3 bytes de la MAC)
BLE	is_ble_identifiable, ble_density_norm
LoRa y metadatos	lora_present, lora_signal (señal real detectada con el SX1262), is_multi_scan

El etiquetado se basó en criterios de seguridad fundamentados en la literatura sobre vulnerabilidades Wi-Fi (Koliass et al., 2016; Vanhoef y Piessens, 2017; Vanhoef y Ronen, 2020), y la entropía del SSID se calculó con la medida de Shannon (1948) e identificó 47 casos anómalos, equivalentes al 6,7 % del conjunto. El conjunto se dividió de forma estratificada en entrenamiento (70 %), validación (15 %) y prueba (15 %).

2.5. Modelos candidatos y selección

Se evaluaron tres arquitecturas: un árbol de decisión (profundidad máxima de cinco niveles; Quinlan, 1986; Breiman et al., 1984), un bosque aleatorio (Breiman, 2001) y una red neuronal de tamaño reducido. La selección consideró el equilibrio entre precisión de clasificación, tamaño en memoria, latencia de inferencia y facilidad de implementación en el microcontrolador. El entrenamiento se realizó en Python con Scikit-learn (Pedregosa et al., 2011), Pandas y NumPy.

2.6. Implementación embebida

El motor de inferencia se implementó como código nativo en C++, reproduciendo exactamente la lógica del árbol entrenado (un enfoque coherente con el despliegue de modelos compactos sin intérprete propio del TinyML; Warden y Situnayake, 2019; Banbury et al., 2021), lo que elimina la necesidad de un intérprete y reduce drásticamente el consumo de memoria. Los parámetros del escalador se incorporaron al firmware como constantes. Además de la clase, el motor calcula una puntuación de confianza heurística a partir de encryption_risk y rssi_norm (no es una probabilidad calibrada) y asigna un motivo legible a cada alerta (cifrado abierto o débil, cifrado desconocido, señal sospechosamente fuerte, patrón anómalo). El umbral de alta prioridad se fijó en 0,80. Los resultados se registran en CSV en la MicroSD con marca de tiempo, SSID, BSSID, RSSI,

canal, cifrado, predicción, confianza y motivo. Entorno: Visual Studio Code con PlatformIO, plataforma espressif32 6.7.0, placa m5stack-stamps3 y librería M5Cardputer 1.1.1.

2.7. Protocolo de evaluación

La evaluación tuvo dos dimensiones: (i) métricas estándar de clasificación (exactitud, precisión, exhaustividad y F1; Powers, 2011) sobre los conjuntos de entrenamiento, validación y prueba; (ii) métricas del sistema embebido (tamaño del modelo, latencia de inferencia y consumo de RAM y Flash). La equivalencia Python-C++ se verificó comparando las predicciones sobre 25 casos de prueba comunes, con criterio de éxito de coincidencia total. Por último, se realizó una validación de campo con 25 redes Wi-Fi reales capturadas en Guayaquil. Las pruebas se hicieron en ambientes controlados y autorizados, sin ejecutar acciones sobre redes de terceros.

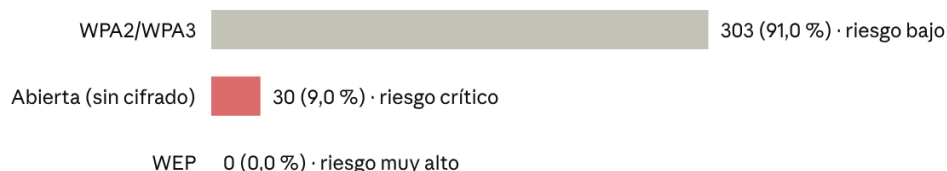
3. Resultados

3.1. Descripción del dataset

El dataset se construyó a partir de doce sesiones de escaneo que generaron 697 observaciones (Figura 1). En las redes Wi-Fi, la intensidad media de señal fue de aproximadamente -79,8 dBm, lo que indica predominio de redes de intensidad moderada a baja. En cuanto al cifrado, el 91 % de las redes Wi-Fi (303 de 333) usa WPA2/WPA3 y cerca del 9 % (30 redes) opera abierta, sin cifrado, vinculadas en su mayoría a infraestructura pública. No se detectaron redes WEP (Figura 2).

Figura 2. Distribución de las configuraciones de cifrado en las redes Wi-Fi detectadas.

Una de cada diez redes Wi-Fi (30 de 333) opera sin cifrado
333 redes Wi-Fi detectadas, por tipo de cifrado y nivel de riesgo

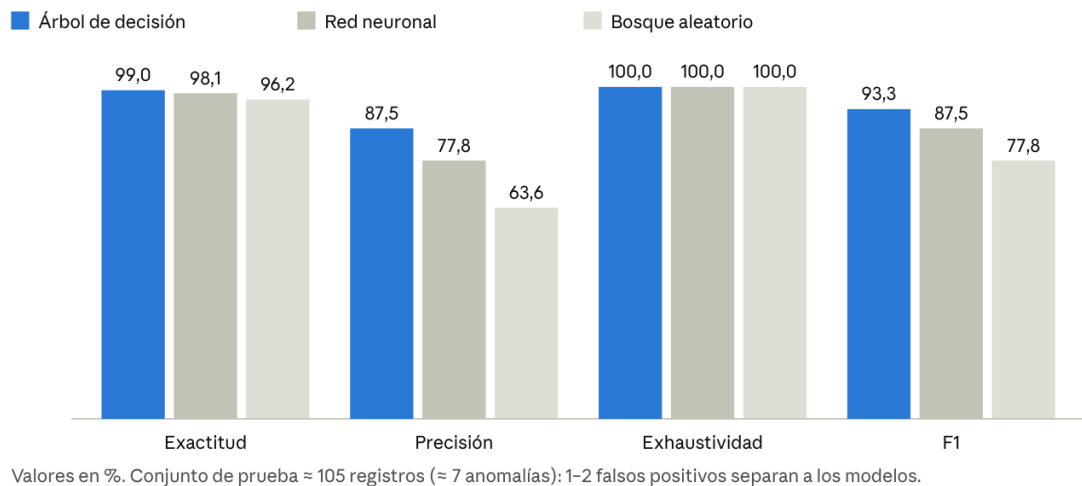


3.2. Comparación de modelos

El árbol de decisión obtuvo la mejor exactitud (99,0 %), precisión (87,5 %) y F1 (93,3 %), la menor latencia (0,07 ms) y el menor tamaño (2.377 bytes $\approx$ 2,3 KB). Los tres modelos alcanzaron una exhaustividad del 100 % (Figura 3 y Figura 4). El análisis de importancia de características situó al riesgo del cifrado como el factor más determinante, seguido por la intensidad de señal normalizada y la entropía del SSID.

Figura 3. Métricas de clasificación de los tres modelos candidatos.

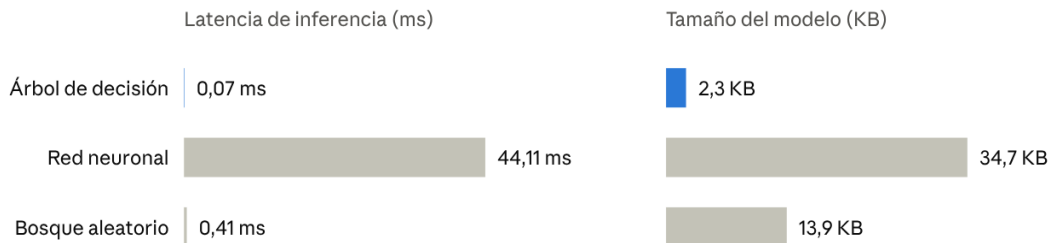
El árbol de decisión lidera exactitud, precisión y F1; los tres tienen 100 % de exhaustividad



Como referencia, con un conjunto de prueba de aproximadamente 105 registros (15 % de 697) y 47 anomalías en total ($\approx$ 7 en la partición de prueba), las métricas reportadas son consistentes con los siguientes conteos derivados (no reportados directamente en el estudio): árbol de decisión, 7 verdaderos positivos y 1 falso positivo; red neuronal, 7 y 2; bosque aleatorio, 7 y 4; sin falsos negativos en ningún caso. Es decir, la diferencia entre modelos depende de muy pocos casos.

Figura 4. Latencia de inferencia y tamaño del modelo por arquitectura.

El árbol de decisión es el más rápido (0,07 ms) y el más pequeño (2,3 KB)



3.3 Equivalencia Python–C++ y consumo de recursos

La coincidencia entre las predicciones del modelo en Python y las del firmware fue del 100 % en los 25 casos de prueba. La compilación con PlatformIO arrojó un uso de 52.720 bytes de RAM sobre 327.680 (16,1 %) y de 922.745 bytes de Flash sobre 3.342.336 (27,6 %), con un tiempo de compilación de 23,63 s. El modelo en C++ nativo ocupa aproximadamente 240 bytes (< 0,1 % de la RAM) y la latencia de inferencia es inferior a 1 ms (Figura 5). El mecanismo de light-sleep entre escaneos en modo automático contribuye a optimizar el consumo energético; no se reportaron mediciones de autonomía de batería.

Figura 5. Consumo de recursos del firmware en el ESP32-S3.

El firmware usa 16,1 % de la RAM y 27,6 % de la Flash; el modelo, menos del 0,1 %
Compilación con PlatformIO sobre ESP32-S3 (la barra completa es el 100 % del recurso)



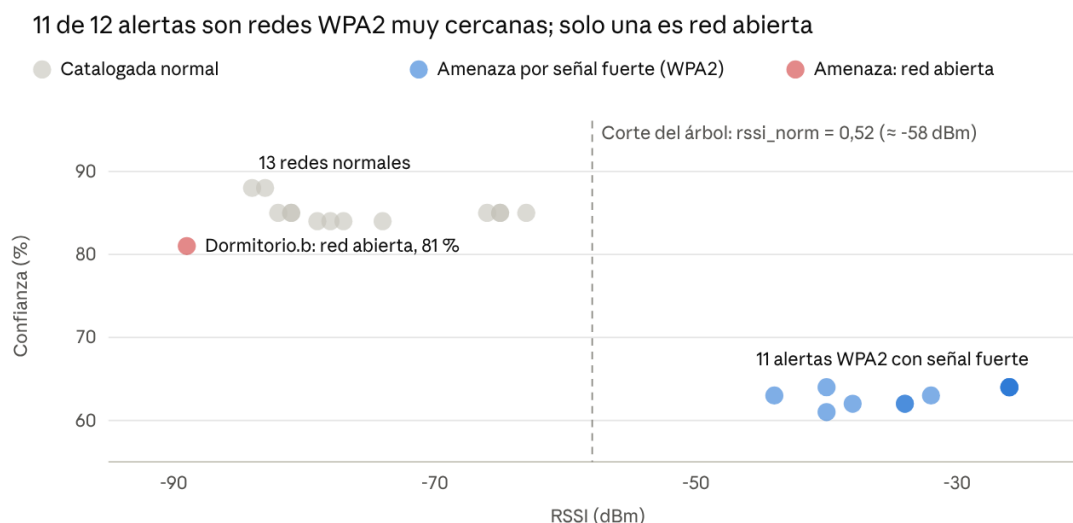
3.4 Evaluación del agente en condiciones reales

El dispositivo completó de forma independiente el ciclo de escaneo, clasificación y presentación de resultados, sin conectividad externa ni intervención humana. La interfaz del Cardputer-Adv muestra una pantalla de escaneo, una lista de redes con SSID, RSSI y

canal codificada por color (verde: segura; rojo: amenaza) y una vista de detalle con el nivel de confianza y el motivo. Ejemplos: «GoolWi-5» (-63 dBm, canal 6) fue clasificada como segura con 86 % de confianza y motivo «Normal»; «CLARO1_VALDEZ» (-37 dBm, canal 11) fue clasificada como amenaza con 64 % de confianza y motivo «Suspiciously strong signal».

En la validación sobre 25 redes reales de Guayaquil, el agente catalogó 12 como amenaza y 13 como normales, con confianzas de 61 a 64 % en las amenazas por señal fuerte, 81 % en la única red abierta y 84 a 88 % en las normales (Figura 6). Todas las amenazas por señal fuerte tienen un RSSI entre -26 y -44 dBm y cifrado WPA2 o WPA/WPA2, mientras que todas las redes catalogadas como normales están entre -63 y -84 dBm. La única red con cifrado abierto («Dormitorio.b», -89 dBm) fue catalogada como amenaza con 81 % de confianza.

Figura 6. Validación de campo: RSSI y confianza reportada por el agente en 25 redes reales.



3.5 Comparativa con herramientas manuales

Característica	Agente autónomo (TinyML)	Herramientas manuales (Bruce/Marauder)
Decisión de auditoría	Automática (modelo IA)	Manual (operador humano)
Velocidad de clasificación	< 1 ms por red	Variable (depende del operador)
Autonomía operativa	Completa	Requiere operador constante
Portabilidad	Dispositivo de bolsillo	Laptop + adaptadores

Evaluación desatendida	Sí (modos No AUTO/CONTINUOUS)
Escalabilidad	Alta (múltiples dispositivos) Limitada (1 operador/equipo)

La comparación es cualitativa: el estudio no midió el desempeño de las herramientas manuales sobre el mismo conjunto de redes.

4. Discusión

Los hallazgos de esta revisión confirman y amplían la literatura previa sobre inteligencia artificial en educación superior, al mostrar que la IAG no sustituye, sino que profundiza y diversifica, las estrategias de personalización ya documentadas en el marco de los sistemas de tutoría inteligente clásicos (VanLehn, 2011) y de las plataformas de aprendizaje adaptativo analizadas por Zawacki-Richter et al. (2019). La diferencia sustantiva radica en la capacidad generativa de estas herramientas, que permite pasar de una personalización basada en la selección entre recursos predefinidos a una personalización basada en la creación dinámica de contenido ajustado al perfil de cada estudiante, tal como sugieren Kasneci et al. (2023). Esta transición tiene implicaciones teóricas relevantes, en la medida en que obliga a repensar los modelos de enseñanza-aprendizaje personalizados no ya como sistemas cerrados de adaptación basados en reglas, sino como ecosistemas híbridos de colaboración entre el estudiante, el docente y un agente generativo que participa activamente en la construcción del contenido educativo.

Viabilidad técnica. Los resultados responden afirmativamente a la dimensión técnica de la pregunta de investigación: un clasificador de árbol de decisión de 2,3 KB, con latencia de 0,07 ms, se despliega en un ESP32-S3 con un uso moderado de recursos (16,1 % de RAM, 27,6 % de Flash) y una equivalencia exacta con el modelo de referencia. Esto es coherente con la advertencia de Abiodun et al. (2018) sobre las redes profundas en entornos restringidos y con la recomendación de modelos compactos en TinyML. La elección del árbol sobre la red neuronal se sustenta en tres ventajas: latencia (0,07 frente a 44,11 ms), tamaño (2,3 frente a 34,7 KB) e interpretabilidad. Además, el árbol permite explicar cada alerta, lo que es valioso en auditoría.

Aporte frente al estado del arte. A diferencia de Bruce (pr3y, s. f.), Marauder (justcallmekoko, s. f.) y NEMO (n0xa, s. f.), el agente incorpora un modelo embebido que

percibe el entorno, clasifica y registra sin intervención del operador. Esto responde al vacío señalado por Ray (2022) sobre TinyML aplicado a la auditoría de redes y a la dependencia del factor humano descrita en la introducción. Además, ejecutar la inferencia localmente es coherente con la preocupación por el consumo de energía y agua de los sistemas de IA centralizados que motiva trabajos recientes sobre agentes eficientes (Facuy Delgado y Arcos-Jacome, 2026). Asimismo, la automatización de la priorización no sustituye el criterio del auditor: los falsos positivos de la validación de campo exigen interpretación humana, en línea con la visión de la IA como herramienta de apoyo y no como sustituto del criterio profesional que se ha planteado en otros campos técnicos y creativos (Facuy Delgado et al., 2025).

Lectura crítica de las métricas. Las altas métricas deben interpretarse con cautela. Primero, el etiquetado se basa en criterios de seguridad derivados de la literatura (un riesgo de circularidad señalado en los trabajos sobre aprendizaje automático en seguridad; Sommer y Paxson, 2010; Arp et al., 2022), y el análisis de importancia sitúa al riesgo del cifrado como el factor dominante: el árbol puede estar reproduciendo en gran medida las reglas de etiquetado, más que descubriendo patrones nuevos. Segundo, con ≈ 7 anomalías en la partición de prueba, un solo falso positivo cambia la precisión varios puntos porcentuales, y los intervalos de confianza serían amplios; además, con clases tan desbalanceadas (6,7 % de anomalías), la precisión y la exhaustividad son más informativas que la exactitud global (He y Garcia, 2009; Saito y Rehmsmeier, 2015). Tercero, el dataset proviene de una sola ciudad y condiciones de captura, lo que limita la generalización a entornos con mayor diversidad de configuraciones. Cuarto, la exhaustividad del 100 % es especialmente relevante en auditoría, donde omitir una amenaza real pesa más que una alerta infundada, aunque la evidencia disponible es limitada.

Validación de campo y falsos positivos. El resultado de campo es el hallazgo más instructivo. De las 12 redes catalogadas como amenaza, 11 usan WPA2 o WPA/WPA2 y fueron señaladas por una señal intensa (-26 a -44 dBm), incluidas redes con nombres que sugieren pertenencia a el propio entorno de captura (por ejemplo, NETLIFE-VALDEZ y su red de invitados) y una impresora con Wi-Fi Direct. Una señal fuerte indica cercanía, no necesariamente malicia, por lo que el criterio de «señal sospechosamente potente» genera alertas sobre infraestructura legítima próxima al dispositivo. Esto sugiere que la

clase anómala del modelo mezcla dos fenómenos distintos (configuración insegura y proximidad atípica) y que conviene separarlos, incorporar reglas de contexto (listas de redes autorizadas conocidas, comparación de BSSID/OUI con baselines) o más características de comportamiento (persistencia, aparición repentina). Además, las confianzas reportadas provienen de una función heurística y no de probabilidades calibradas; los valores de 61 a 64 % indican, en la práctica, baja certidumbre y podrían mapearse a una prioridad «baja» frente al umbral de 0,80 definido para alta prioridad.

Alcance real frente al objetivo planteado. El título original del trabajo y el objetivo general mencionan Wi-Fi y LoRa, pero el agente autónomo funciona en tiempo real únicamente sobre Wi-Fi. BLE y LoRa se usaron en el dataset, y no se contó con infraestructura LPWAN activa en el entorno de recolección, por lo que no se audió ninguna red LoRaWAN. Asimismo, el agente actual cubre la fase de detección y clasificación; las acciones de auditoría activa guiadas por la inferencia no se implementaron. Por ello, la afirmación de una efectividad «comparable o superior» a las herramientas manuales se sustenta en ventajas operativas (autonomía, velocidad, portabilidad), no en una comparación empírica directa de detección.

Implicaciones defensivas. Las configuraciones abiertas o con cifrado obsoleto constituyen los principales indicadores de riesgo identificados por el modelo. Se recomienda priorizar su eliminación, adoptar sistemáticamente los estándares de cifrado más robustos (WPA3 cuando sea posible, manteniendo los equipos actualizados frente a vulnerabilidades de implementación conocidas; Vanhoef y Piessens, 2017; Vanhoef y Ronen, 2020) y establecer programas periódicos de evaluación de la postura de seguridad inalámbrica. Para LPWAN, resulta crucial la correcta gestión de los contadores de trama y la aleatoriedad en los procedimientos de activación (Eldefrawy et al., 2019). Finalmente, la existencia de herramientas de auditoría portátiles y autónomas debe motivar a las organizaciones a fortalecer proactivamente sus defensas, en línea con el principio de seguridad desde el diseño (Banerjee et al., 2012).

4.1 Limitaciones

- Dataset generado en una sola ciudad y en condiciones acotadas, con solo 47 casos anómalos (6,7 %), lo que limita la generalización y la potencia estadística.

- Etiquetado basado en reglas de seguridad, con posible circularidad entre etiquetas y características dominantes.
- Integración en tiempo real limitada a Wi-Fi; BLE y LoRa solo en la fase de dataset; sin GNSS.
- Sin infraestructura LPWAN activa en el entorno, por lo que no se auditaron redes LoRaWAN.
- Restricción de la complejidad del modelo por los recursos del ESP32-S3 (se descartan arquitecturas profundas).
- Alcance limitado a detección y clasificación, sin auditoría activa.
- Ausencia de comparación empírica directa con Bruce/Marauder y de mediciones de autonomía energética.

4.2 Líneas futuras

- Ampliar el dataset a múltiples entornos geográficos y temporales para mejorar robustez y generalización.
- Integrar efectivamente el módulo LoRa para validar la auditoría sobre redes LPWAN con infraestructura activa.
- Explorar aprendizaje incremental para que el dispositivo adapte su modelo a las observaciones acumuladas en campo.
- Extender el sistema hacia acciones de auditoría activa guiadas por la inferencia, cerrando el ciclo de percepción, decisión y acción.
- Calibrar la confianza, separar configuración insegura de proximidad atípica y medir el consumo energético y la autonomía; además, proteger la integridad y la trazabilidad de los registros de auditoría, por ejemplo con arquitecturas que combinen redes neuronales y cadena de bloques (Facuy et al., 2026).

5. Conclusiones

Este trabajo demuestra que es viable ejecutar un modelo de IA en el borde, basado en TinyML, dentro de un dispositivo embebido de bajo costo (M5Stack Cardputer-Adv) para clasificar de forma autónoma la configuración de seguridad de redes Wi-Fi. Un árbol de decisión de 2,3 KB alcanzó 99,0 % de exactitud, 93,3 % de F1 y 100 % de exhaustividad con 0,07 ms de latencia, se reprodujo con coincidencia total en C++ nativo y dejó un amplio margen de recursos para futuras extensiones. La validación de campo mostró que



el agente detecta la configuración más crítica (red abierta), aunque la regla de señal fuerte genera falsos positivos sobre infraestructura legítima. La extensión a BLE y LoRa en tiempo real, la ampliación del dataset y la auditoría activa constituyen los siguientes pasos. Más que sustituir al auditor, el agente se plantea como una herramienta de apoyo que libera tiempo en tareas repetitivas y deja la interpretación final de las alertas al criterio humano (Facuy Delgado et al., 2025).

Declaración Ética

Todas las pruebas se realizaron en ambientes controlados y autorizados, con fines estrictamente académicos y experimentales, sin ejecutar acciones sobre redes de terceros. Las técnicas se abordan desde una perspectiva defensiva y de evaluación.

Referencias

- Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. y Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), e00938. <https://doi.org/10.1016/j.heliyon.2018.e00938>
- Adelantado, F., Vilajosana, X., Tuset-Peiró, P., Martínez, B., Melià-Seguí, J. y Watteyne, T. (2017). Understanding the limits of LoRaWAN. *IEEE Communications Magazine*, 55(9), 34-40. <https://doi.org/10.1109/MCOM.2017.1600613>
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. y Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376. <https://doi.org/10.1109/COMST.2015.2444095>
- Antonakakis, M., April, T., Bailey, M., et al. (2017). Understanding the Mirai botnet. En *Proceedings of the 26th USENIX Security Symposium* (pp. 1093-1110). USENIX Association.
- Aras, E., Ramachandran, G. S., Lawrence, P. y Hughes, D. (2017). Selective jamming of LoRaWAN using commodity hardware. En *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2017)* (pp. 363-372). ACM. <https://doi.org/10.1145/3144457.3144478>
- Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L. y Rieck, K. (2022). Dos and don'ts of machine learning in computer

- security. En Proceedings of the 31st USENIX Security Symposium (pp. 3971-3988). USENIX Association.
- Atzori, L., Iera, A. y Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787-2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Augustin, A., Yi, J., Clausen, T. y Townsley, W. M. (2016). A study of LoRa: Long range & low power networks for the Internet of Things. *Sensors*, 16(9), 1466. <https://doi.org/10.3390/s16091466>
- Banbury, C., Reddi, V. J., Torelli, P., et al. (2021). MLPerf Tiny benchmark. En Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks. <https://arxiv.org/abs/2106.07597>
- Banerjee, A., Venkatasubramanian, K. K., Mukherjee, T. y Gupta, S. K. S. (2012). Ensuring safety, security, and sustainability of mission-critical cyber-physical systems. *Proceedings of the IEEE*, 100(1), 283-299. <https://doi.org/10.1109/JPROC.2011.2165689>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324>
- Breiman, L., Friedman, J. H., Olshen, R. A. y Stone, C. J. (1984). Classification and regression trees. Wadsworth.
- Buczak, A. L. y Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176. <https://doi.org/10.1109/COMST.2015.2494502>
- Butun, I., Österberg, P. y Song, H. (2020). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. *IEEE Communications Surveys & Tutorials*, 22(1), 616-644. <https://doi.org/10.1109/COMST.2019.2953364>
- Butun, I., Pereira, N. y Gidlund, M. (2019). Security risk analysis of LoRaWAN and future directions. *Future Internet*, 11(1), 3. <https://doi.org/10.3390/fi11010003>
- Centenaro, M., Vangelista, L., Zanella, A. y Zorzi, M. (2016). Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications*, 23(5), 60-67. <https://doi.org/10.1109/MWC.2016.7721743>
- David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Wang, T., Warden, P. y Rhodes, R. (2021). TensorFlow Lite Micro:

- Embedded machine learning for TinyML systems. *Proceedings of Machine Learning and Systems*, 3, 800-811.
- Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S. y Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8), 7457-7469. <https://doi.org/10.1109/JIOT.2020.2984887>
- Diro, A. A. y Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, 82, 761-768. <https://doi.org/10.1016/j.future.2017.08.043>
- Eldefrawy, M. H., Butun, I., Pereira, N. y Gidlund, M. (2019). Formal security analysis of LoRaWAN. *Computer Networks*, 148, 328-339. <https://doi.org/10.1016/j.comnet.2018.11.017>
- Facuy, J., Pasini, A., Estévez, E., Moran, C., Illescas, W. y Veloz, V. (2026). A blockchain-enhanced neural network framework for secure e-waste forecasting in smart cities. *Frontiers in Public Health*, 14, 1779806. <https://doi.org/10.3389/fpubh.2026.1779806>
- Facuy Delgado, J. y Arcos-Jacome, D. (2026). Emergent language among AI agents: A path toward energy efficiency and water conservation. *Frontiers in Sustainability*, 6, 1717425. <https://doi.org/10.3389/frsus.2025.1717425>
- Facuy Delgado, J. P., Santillán Indacochea, S. D., Soria Vera, R. I. y Toledo Navarrete, A. E. (2025). La mente creativa del diseñador vs el impacto de la inteligencia artificial en el diseño gráfico: nuevas herramientas y métodos creativos. *Ethos Scientific Journal*, 3(1), 89-102. <https://doi.org/10.63380/esj.v3n1.2025.90>
- Han, S., Mao, H. y Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. En *International Conference on Learning Representations (ICLR 2016)*. <https://arxiv.org/abs/1510.00149>
- He, H. y Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284. <https://doi.org/10.1109/TKDE.2008.239>
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. y Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-

- arithmetic-only inference. En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2704-2713). IEEE. <https://doi.org/10.1109/CVPR.2018.00286>
- justcallmekoko. (s. f.). ESP32 Marauder: A suite of WiFi/Bluetooth offensive and defensive tools for the ESP32 [Software]. GitHub. <https://github.com/justcallmekoko/ESP32Marauder>
- Khan, M. A. y Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 82, 395-411. <https://doi.org/10.1016/j.future.2017.11.022>
- Kolias, C., Kambourakis, G., Stavrou, A. y Gritzalis, S. (2016). Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials, 18(1), 184-208. <https://doi.org/10.1109/COMST.2015.2402161>
- Kolias, C., Kambourakis, G., Stavrou, A. y Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80-84. <https://doi.org/10.1109/MC.2017.201>
- LoRa Alliance. (2020). LoRaWAN L2 1.0.4 specification (TS001-1.0.4). LoRa Alliance. <https://lora-alliance.org>
- M5Stack Technology Co. (2024). M5Stack Cardputer-Adv: Technical documentation and user guide. <https://docs.m5stack.com/en/core/Cardputer-Adv>
- Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D. y 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>
- Mirsky, Y., Doitshman, T., Elovici, Y. y Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. En Proceedings of the Network and Distributed System Security Symposium (NDSS 2018). <https://doi.org/10.14722/ndss.2018.23204>
- n0xa. (s. f.). M5Stick-NEMO: Firmware for M5Stick and Cardputer devices [Software]. GitHub. <https://github.com/n0xa/m5stick-nemo>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, É. (2011). Scikit-learn:

- Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.
- pr3y. (s. f.). Bruce: Versatile ESP32 firmware for red team operations [Software]. GitHub. <https://github.com/pr3y/Bruce>
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106. <https://doi.org/10.1007/BF00116251>
- Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences*, 34(4), 1595-1623. <https://doi.org/10.1016/j.jksuci.2021.11.019>
- Raza, U., Kulkarni, P. y Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2), 855-873. <https://doi.org/10.1109/COMST.2017.2652320>
- Roman, R., Zhou, J. y Lopez, J. (2013). On the features and challenges of security and privacy in distributed Internet of Things. *Computer Networks*, 57(10), 2266-2279. <https://doi.org/10.1016/j.comnet.2012.12.018>
- Saito, T. y Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10(3), e0118432. <https://doi.org/10.1371/journal.pone.0118432>
- Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P. y Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data*, 7, 41. <https://doi.org/10.1186/s40537-020-00318-5>
- Semtech Corporation. (2023). SX1262: Long range, low power, sub-GHz RF transceiver — Datasheet. <https://www.semtech.com/products/wireless-rf/lora-connect/sx1262>
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379-423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- Shi, W., Cao, J., Zhang, Q., Li, Y. y Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637-646. <https://doi.org/10.1109/JIOT.2016.2579198>
- Shone, N., Ngoc, T. N., Phai, V. D. y Shi, Q. (2018). A deep learning approach to network

- intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50. <https://doi.org/10.1109/TETCI.2017.2772792>
- Sicari, S., Rizzardi, A., Grieco, L. A. y Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146-164. <https://doi.org/10.1016/j.comnet.2014.11.008>
- Sommer, R. y Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. En 2010 IEEE Symposium on Security and Privacy (pp. 305-316). IEEE. <https://doi.org/10.1109/SP.2010.25>
- Vailshery, L. S. (2023). Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030. Statista. <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- Vanhoef, M. y Piessens, F. (2017). Key reinstallation attacks: Forcing nonce reuse in WPA2. En Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1313-1328). ACM. <https://doi.org/10.1145/3133956.3134027>
- Vanhoef, M. y Ronen, E. (2020). Dragonblood: Analyzing the Dragonfly handshake of WPA3 and EAP-pwd. En 2020 IEEE Symposium on Security and Privacy (pp. 517-533). IEEE. <https://doi.org/10.1109/SP40000.2020.00031>
- Warden, P. y Situnayake, D. (2019). TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers. O'Reilly Media.
- Yang, K., Forte, D. y Tehranipoor, M. M. (2017). CDTA: A comprehensive solution for counterfeit detection, traceability, and authentication in the IoT supply chain. ACM Transactions on Design Automation of Electronic Systems, 22(3), Artículo 42. <https://doi.org/10.1145/3005346>
- Yang, X., Karampatzakis, E., Doerr, C. y Kuipers, F. (2018). Security vulnerabilities in LoRaWAN. En 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI). IEEE. <https://doi.org/10.1109/IoTDI.2018.00022>
- Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K. y Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738-1762. <https://doi.org/10.1109/JPROC.2019.2918951>