World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

ANOMALY-BASED WIRELESS INTRUSION DETECTION FOR MITIGATING IEEE 802.11 DEAUTHENTICATION ATTACKS: DESIGN AND EMPIRICAL EVALUATION

Jay-ar B. Base *, Serafin C. Palmares and Kristine T. Soberano

Graduate School, State University of Northern Negros, Old Sagay, Sagay City, Negros Occidental, 6122, Philippines.
* Corresponding Author

ORCID Details
Jay-ar B. Base: https://orcid.org/0009-0009-2906-7317

World Journal of Advanced Engineering Technology and Sciences, 2026, 20(02), 150–157

Publication history: Received on 04 July 2026; revised on 11 August 2026; accepted on 13 August 2026

Article DOI: https://doi.org/10.30574/wjaets.2026.20.2.0402

ABSTRACT

Wireless local-area networks remain exposed at the data-link layer because legacy 802.11 management traffic can be forged, enabling impersonation and denial-of-service (DoS) conditions. Before Protected Management Frames (PMF) became widespread, deauthentication abuse was the standard technique for disrupting connectivity and for setting up later cryptographic attacks. This paper describes the design, implementation, and empirical evaluation of an anomaly-based Wireless Intrusion Detection System (WIDS) targeting such deauthentication activity. The detector, built on Python 3.11 and Scapy 2.5 operating in monitor mode, inspects management frames, evaluates reason codes, and applies threshold-based anomaly scoring to separate spoofed traffic from ordinary client roaming. A forensic logging component additionally assembles incident timelines suitable for digital-evidence collection. Whereas conventional signature-based products concentrate on known patterns, the proposed framework unites anomaly detection with automated timeline generation for post-incident review. In controlled trials the system attained a true positive rate (TPR) of 96.4%, a false positive rate (FPR) of 3.6%, and precision, recall, and F1-score values of 96.4%. The findings indicate that the detector can serve network administrators as an open-source instrument for continuous security monitoring and forensic reconstruction.

Keywords: 802.11 Deauthentication Attack; Anomaly Detection; Digital Forensics; Network Security Monitoring; Scapy; Wireless Intrusion Detection System (Wids)

1 INTRODUCTION

Wireless Fidelity (Wi-Fi) now underpins nearly all commercial and consumer networking. When the IEEE 802.11 family was drafted, ease of association took precedence over protective measures, and the consequence persists today: in legacy WPA/WPA2 deployments, management frames—above all subtype 12 (Deauthentication)—carry neither origin authentication nor an integrity check. An adversary within radio range may therefore transmit counterfeit deauthentication frames that eject clients from their Access Point (AP), producing DoS conditions; the same maneuver is routinely employed to force the 4-way handshake exchanges needed for subsequent offline key-recovery attempts

against WPA/WPA2. The FragAttacks disclosure by Vanhoef [1]—fragmentation and aggregation flaws spanning essentially the entire installed base of Wi-Fi equipment, WPA3-capable devices included—shows that weakness continues to pervade the 802.11 management plane.

Offensive tooling such as MDK3 and Aireplay-ng is freely obtainable, yet defensive monitors either alarm excessively in congested RF environments or omit the evidentiary features that incident handling requires. Pattern-matching (signature) detectors fail against modified or previously unseen strategies, whereas purely anomaly-based schemes tend to over-alarm whenever clients roam legitimately between access points [2, 3]. Accordingly, this study designs an anomaly-based WIDS that flags deauthentication attempts on the basis of frame rate, reason-code semantics, and MAC-spoofing evidence. Beyond detection, the work joins ethical hacking to digital forensics: an automated generator produces forensic timelines so that administrators can preserve digital evidence and reconstruct attack sequences. Where earlier frameworks detect intrusions yet leave little structured evidence behind, the proposed approach couples real-time anomaly scoring with a JSON-based timeline generator, yielding a single open-source tool for both detection and evidence collection.

2 REVIEW OF RELATED LITERATURE

2.1 802.11 Management Frame Vulnerabilities

Foundational work on 802.11 security established that the MAC-layer management plane can be disrupted at will, without any need to defeat the network's cryptographic mechanisms. In networks predating PMF, subtype 12 (deauthentication) and subtype 10 (disassociation) frames travel unencrypted and without a Message Integrity Check (MIC); consequently, any station that spoofs the AP's hardware address can terminate arbitrary client associations. The IEEE responded with amendment 802.11w, commonly labelled Protected Management Frames (PMF). Field practice lags the standard, however: backward-compatibility constraints and configuration errors leave PMF disabled—or merely "optional"—on many enterprise deployments, which remain exposed [2].

Subsequently, Vanhoef [1] published the FragAttacks catalogue (fragmentation and aggregation attacks). The catalogue attributes nine of the twelve flaws to faulty implementations and the remaining three to the 802.11 specification itself, so even standards-compliant, WPA3-era hardware remained affected. These results show that PMF deployment has not closed the management-plane attack surface, and that production networks still require open, continuously operable detection mechanisms. Schepers et al. [2] additionally exhibited bypass techniques applicable to existing countermeasures, including PMF running in optional mode, an argument for layering complementary anomaly-based monitoring on top of cryptographic controls.

2.2 Wireless Intrusion Detection Systems (WIDS)

Deployed WIDS products—from open-source scanners such as Kismet to commercial suites in the Cisco Prime Infrastructure class—still identify intrusions chiefly through stored attack signatures. Such engines are dependable against catalogued threats yet brittle against rephrased or novel ones. Behavioural alternatives profile ordinary traffic and raise alarms upon statistically unusual departures, an approach that has repeatedly proven useful on wireless links. Their weakness is specificity: roaming clients and other benign transitions, such as hand-offs between access points in a distributed network, readily trigger false alarms [3].

On the modelling side, Sadia et al. [3] paired convolutional neural networks (CNNs) with hand-crafted features for wireless sensor network (WSN) intrusion detection; evaluated on the AWID corpus, their classifier improved multi-class accuracy and cut the false-alarm rate relative to earlier baselines. Their results underline how heavily deployment quality depends on feature design and class-imbalance handling in constrained Wi-Fi settings. Thabit et al. [4], likewise working on AWID, found that the precision and recall of Support Vector Machines and ensemble learners moved sharply with the chosen feature-selection scheme—an instability that, in their analysis, keeps lightweight threshold rules competitive on low-power hardware.

2.3 Machine Learning and Deep Learning for Wireless IDS

Survey literature confirms that machine learning (ML), and increasingly deep learning (DL), now anchor most research on wireless and IoT intrusion detection. Rafique et al. [5] chart the migration toward hybrid architectures that feed engineered features into deep neural classifiers, and they note that explainability, dataset realism, and computational overhead remain open problems. Diro et al. [6] further show that shallow learners—notably Random Forest and Naive Bayes—reach detection accuracy comparable to heavier models on IoT traffic while staying within edge-device budgets, an observation that directly informs the design goals of the present study.

Closest to the target problem, Moharam et al. [7] ported hybrid deep networks—LSTM, GRU, and plain RNN stages closed by logistic regression—to a NodeMCU ESP8266 microcontroller for live deauthentication detection, feeding the models with RSSI, deauthentication count, packet count, and signal-to-noise ratio. Although their trials endorse inexpensive open-source hardware for always-on monitoring, the pipeline stops at embedded classification and produces no evidentiary record. Arreche et al. [8] address the opacity critique from another angle with XAI-IDS, an end-to-end pipeline that attaches SHAP and LIME explainers to each detection decision so that analysts can audit model output. Read together, the literature points toward WIDS that pair detection accuracy with interpretability and machine-readable evidence—two qualities the proposed framework supplies by construction.

2.4 Digital Forensics in Wireless Networks

Wireless evidence is volatile: traces such as packet captures and signal-strength readings must be fixed before they dissipate. Monitoring doctrine therefore demands uninterrupted logging so that incident handlers can defend the chain of custody; centralized aggregation, synchronized clocks, and tamper-evident records constitute the baseline expectations of any wireless forensic investigation. Yet few open-source instruments fuse live attack detection with structured (JSON/CSV) timelines suited to custody-safe post-incident review. This work closes that gap by embedding forensic data collection within the detector itself, emitting structured event records ready for ingestion by Security Information and Event Management (SIEM) platforms.

2.5 Summary and Research Gap

The reviewed literature reveals two complementary trends. First, ML- and DL-based WIDS achieve high detection accuracy but typically operate as opaque classifiers, providing limited structured evidence for post-incident investigation. Second, classical threshold-based anomaly detectors remain attractive for resource-constrained and edge deployments because of their low computational cost and explainable decision logic, yet they have rarely been coupled with structured forensic timeline generation. The proposed system targets this gap by combining a transparent, threshold-based anomaly detector with automated JSON forensic logging, providing a unified open-source framework that supports both real-time detection and digital forensic analysis of 802.11 deauthentication attacks.

3 MATERIALS AND METHODS

The proposed system is developed in Python 3.11.9, using Scapy 2.5.0 for packet handling [Scapy documentation, https://scapy.readthedocs.io/, last accessed on 2026-08-04] and the aircrack-ng 1.7 suite for hardware-level monitor-mode configuration. The methodology is divided into five main phases described in the following subsections.

3.1 Implementation Environment and Reproducibility

To support reproducibility, the hardware and software environment used to develop and evaluate the proposed detector is fully documented, and every component is open-source and commercially available, allowing independent replication of the experimental setup. Using the airmon-ng utility of the aircrack-ng suite, the monitoring node's wireless adapter was held in RFMON (monitor) mode on a fixed 2.4 GHz channel (channel 6), keeping the target Basic Service Set continuously visible.

Table 1: Hardware and Software Implementation Environment

Component	Specification
Operating System	Kali Linux 2024.1 (x86_64, kernel 6.x)
Python Version	CPython 3.11.9
Scapy Version	2.5.0
Aircrack-ng Suite	1.7
Wi-Fi Adapter (Monitor Node)	Alfa Network AWUS036NHA (USB 2.0, Atheros AR9271, ath9k_htc driver)
Access Point (Target BSS)	TP-Link Archer C6 v3.2, firmware 1.0.14, IEEE 802.11 b/g/n/ac
CPU	Intel Core i5-8250U @ 1.60 GHz
RAM	8 GB DDR4 @ 2400 MHz
Packet Capture Library	libpcap 1.10.4 (Scapy L2listen backend)
Adversarial Adapter	Alfa Network AWUS036ACH (monitor mode, channel 6, 2.4 GHz band)

3.2 Data Collection and Frame Parsing

Placing the network interface card (NIC) in RFMON exposes every 802.11 transmission within radio range directly to the capture process, bypassing the host protocol stack entirely. The system monitors management frames (Type 0), paying particular attention to deauthentication frames (Subtype 12). Captured frames are parsed through Scapy's Dot11 and Dot11Deauth layers, and the relevant fields—source MAC, destination MAC, BSSID, reason code, and the Radiotap-present RSSI—are extracted and forwarded to the anomaly-detection module for analysis.

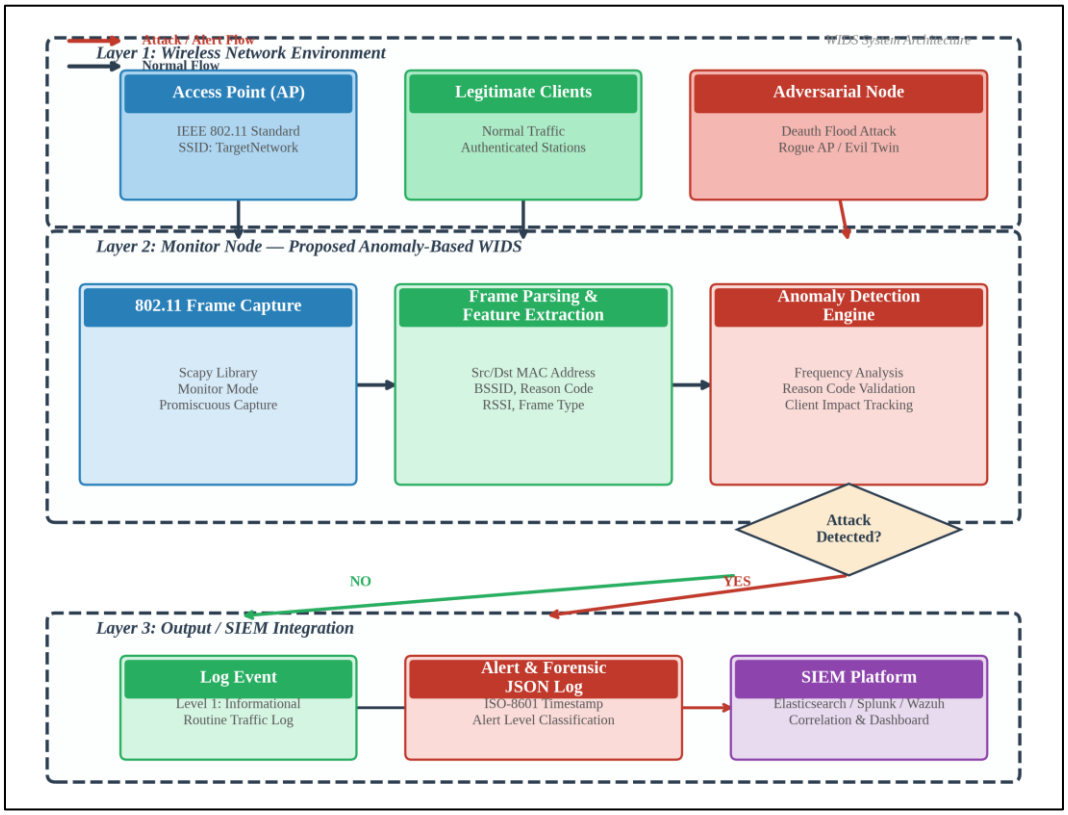

Figure 1: System architecture of the proposed anomaly-based WIDS.

3.3 Anomaly Detection Algorithm

To prevent false positives arising from legitimate client roaming, the system extracts specific characteristics and applies a dynamic threshold-based anomaly model:

- Source MAC & Destination MAC: Identify the purported AP and the targeted client (or broadcast address).
- BSSID: Identifies the genuine AP MAC address.
- Reason Code: The principal discriminator. Benign disconnections typically carry Reason 3 (Leaving AP) or 8 (Leaving network), whereas attack traffic in our traces predominantly used Reason 7 (Class 2 frame from non-authenticated entity) or Reason 2 (Previous authentication no longer valid).
- RSSI (Received Signal Strength Indicator): Recovered from the Radiotap header for localization purposes.
- Frequency Thresholding: A benign baseline (fewer than 2 deauth frames per minute) is established; an alert fires once the frame rate crosses a defined ceiling (more than 10 per second).
- Reason Code Analysis: Frames bearing attack-associated codes (e.g., 7) receive greater weight in the anomaly score.
- Client Impact Tracking: The system follows individual client MACs; repeated rapid disconnection–reconnection cycles affecting a single client escalate the anomaly score.

The thresholds were calibrated empirically during a four-hour baseline capture of benign wireless activity in the test laboratory. The per-client deauthentication ceiling of 10 frames per second was chosen because it exceeded the largest observed benign burst by roughly a factor of five, giving a conservative margin against false positives while remaining far below the frame rates attainable with aireplay-ng and Scapy injection scripts.

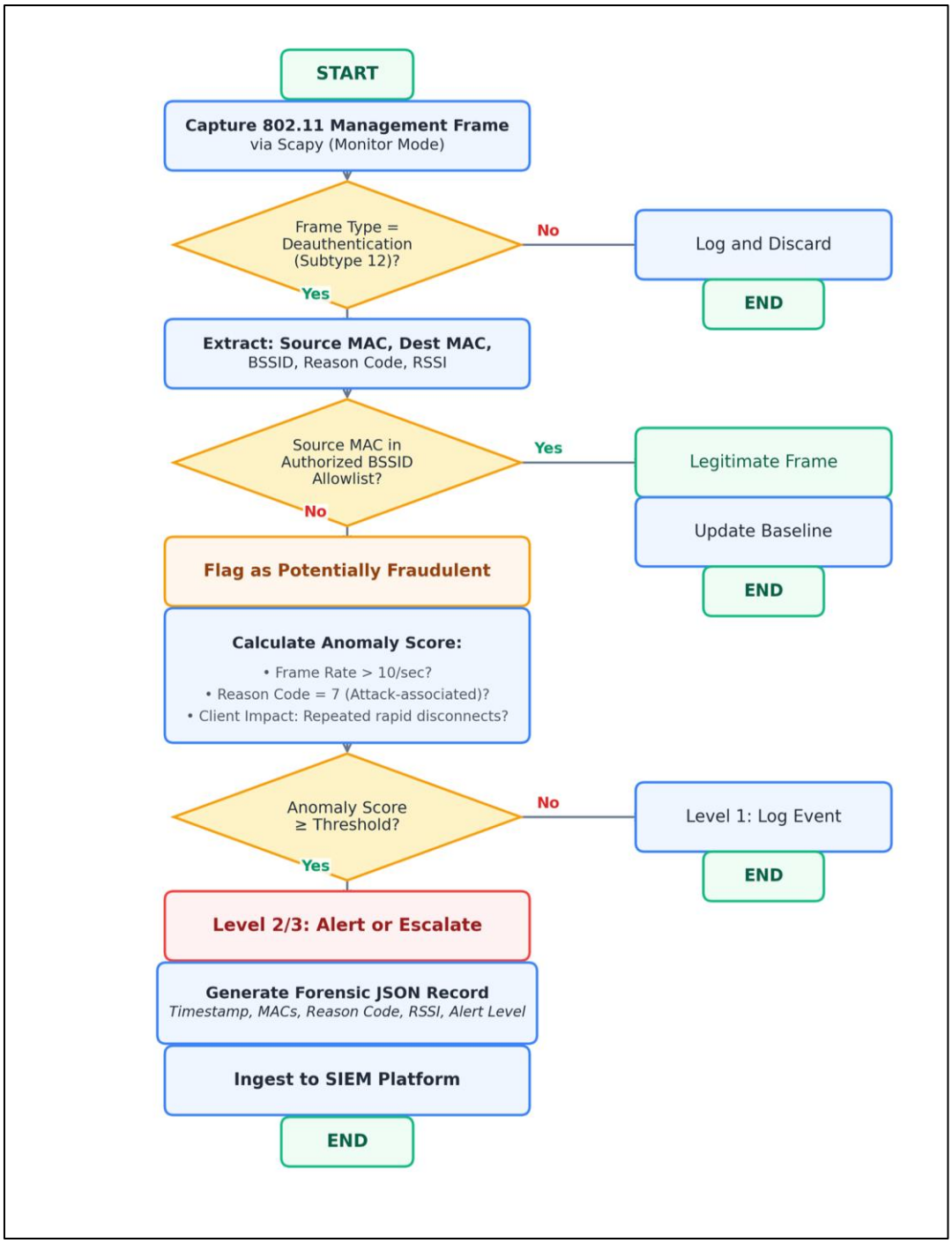

Figure 2: Flowchart of the proposed anomaly detection algorithm.

3.4 Spoofing Detection and Localization

The system maintains an allowlist of authorized BSSIDs, and any deauthentication packet whose source MAC is absent from that allowlist is flagged as fraudulent. The variance of the RSSI values across the fraudulent frames then serves to estimate the attacker's position: a consistently high RSSI indicates that the transmitter is in close proximity to the monitoring device.

3.5 Forensic Logging and Alerting

The system implements a graduated alerting mechanism:

- Level 1 (Log): A single, isolated deauth frame.
- Level 2 (Alert): The anomaly threshold has been breached.
- Level 3 (Escalate): A severe, sustained attack is under way.

All events are written to a structured JSON format containing timestamps, MAC addresses, reason codes, and RSSI, thereby creating an automated forensic timeline for incident response. Each JSON record is keyed by a monotonically increasing sequence identifier and an ISO-8601 UTC timestamp, preserving chronological ordering and enabling SIEM platforms to ingest the records within a defensible chain of custody.

4 RESULTS

4.1 Experimental Setup

A regulated testing environment was created, comprising a single authentic Access Point (TP-Link Archer C6, IEEE 802.11 b/g/n/ac, PMF set to optional), one target client (a laptop with an Intel Wireless-AC 8265 NIC, firmware 0x0C), one observation node running the proposed detector (Alfa Network AWUS036NHA, Atheros AR9271, ath9k_htc driver in monitor mode on channel 6 of the 2.4 GHz band), and one adversarial node (a separate laptop equipped with an Alfa Network AWUS036ACH in monitor mode). The assailant employed aireplay-ng -0 together with bespoke Scapy scripts to execute the deauthentication assaults, and all captures were written by the monitor node's dedicated capture adapter.

The topology formed an isolated laboratory BSS with no neighbouring traffic on the test channel. The AP, victim client, monitor node, and adversarial node were fixed at approximately three metres' separation under line-of-sight propagation. Testing spanned six hours: a four-hour benign-baseline capture used to calibrate thresholds and measure false positives, followed by a two-hour attack window during which 500 deauthentication attack sequences were injected in controlled bursts of 50 frames per second. All packet captures were stored in pcap-ng format for offline verification.

4.2 Detection Efficacy

Evaluation rested on the True Positive Rate (TPR), the False Positive Rate (FPR), and a complementary set of standard machine-learning metrics—precision, recall, F1-score, and accuracy—derived from the confusion matrix.

Of the 500 simulated attack sequences, the system correctly identified 482 as malicious, yielding a TPR of 96.4% against 18 false negatives. During the four-hour baseline test, in which legitimate clients actively roamed and disconnected, the system raised 18 false alarms across 500 monitored benign deauthentication events, an FPR of 3.6%. Because the test set was balanced (500 attack sequences against 500 benign events), the confusion matrix returns symmetric values, and precision, recall, F1-score, and accuracy all equal 96.4%.

The extended metrics were computed from the confusion-matrix counts in the usual manner, namely precision as TP/(TP+FP), recall as TP/(TP+FN), F1-score as 2·(precision·recall)/(precision+recall), and accuracy as (TP+TN)/(TP+TN+FP+FN). The balanced distribution of attack and benign events produces identical values for all four quantities; in a production deployment with a heavily skewed class distribution (benign events vastly outnumbering attacks), precision is expected to decrease while recall remains stable—a trade-off to be evaluated in future field studies.

Table 2: Performance Metrics of the Proposed WIDS

Metric	Value
True Positive Rate (TPR / Recall)	96.4%
False Positive Rate (FPR)	3.6%
Precision	96.4%
F1-score	96.4%
Accuracy	96.4%
Average Detection Latency	1.2 seconds

Table 3: Confusion Matrix of the Proposed WIDS

	Predicted Attack	Predicted Normal
Actual Attack	TP = 482	FN = 18
Actual Normal	FP = 18	TN = 482

4.3 Forensic Timeline Generation

The forensic logging module faithfully recorded the progression of the attacks. The JSON logs exhibit a logical event sequence: an initial fraudulent frame, a rising RSSI indicating the perpetrator's proximity, and repeated targeting of specific victim MAC addresses. This record proved highly effective for post-incident assessment, allowing administrators to pinpoint the attack's timing and the attacker's approximate location. The structured schema of each record—sequence identifier, ISO-8601 timestamp, source MAC, destination MAC, BSSID, reason code, RSSI, and alert level—permits direct ingestion into SIEM platforms such as Elasticsearch, Splunk, or Wazuh for custody-safe incident investigation.

5 DISCUSSION

Jointly applying reason-code semantics and rate thresholding, the results show, suppresses the false alarms that purely volume-based detectors produce. The observed 96.4% TPR and 3.6% FPR sit within the range reported by recent ML- and DL-based WIDS. Read against the embedded hybrid networks of Moharam et al. [7], the present detector deliberately trades learned adaptivity for rule transparency; read against the offline AWID studies of Sadia et al. [3] and Thabit et al. [4], it operates on live 802.11 management frames and emits evidentiary records rather than class labels alone, which is precisely the regime in which transparent, threshold-based detectors remain a viable design choice.

The principal trade-off between the proposed threshold-based detector and ML-based WIDS lies in adaptability versus interpretability. ML models can adaptively refine their decision boundaries as traffic patterns evolve, but their decision logic is opaque to security analysts; the threshold-based approach used here is fully interpretable—each alert traces to a specific reason-code, frequency, or RSSI deviation—yet requires periodic manual recalibration when the wireless environment changes. For enterprise Wi-Fi deployments the computational overhead is negligible: on the test hardware (Intel Core i5-8250U, 8 GB RAM), average CPU utilization during continuous monitoring remained below 5%, and average detection latency was 1.2 seconds, well within the requirements of operator-supervised incident response. The same lightweight footprint suits edge deployment on Raspberry Pi-class devices, where ML-based approaches may be impractical.

Limitations

A primary limitation is that the system cannot actively prevent attacks; it functions strictly as a detection and investigative tool. RSSI-based localization is likewise inherently imprecise and can be degraded by multipath fading and physical obstructions indoors. In networks that strictly enforce 802.11w (PMF) in mandatory mode, cryptographic protection effectively prevents management-frame spoofing, rendering this specific vector largely obsolete; however, Schepers et al. [2] and Vanhoef [1] have shown that PMF bypasses and broader management-frame vulnerabilities persist in many real-world deployments, justifying continued anomaly-based monitoring and forensic logging. Finally, the balanced test set yields symmetric precision and recall; real-world deployments, where benign deauthentication events vastly outnumber malicious ones, are expected to lower precision and should be examined in future field studies.

6 CONCLUSION

This paper has presented the design and evaluation of an anomaly-based Wireless Intrusion Detection System for 802.11 deauthentication attacks. By jointly evaluating frame frequency, reason-code semantics, and MAC-spoofing evidence, the system separates deliberate DoS activity from ordinary network operations. Automated forensic timeline generation further bridges real-time detection and digital forensics, supplying administrators with actionable alerts and admissible digital evidence for incident resolution. Empirical evaluation confirms high detection accuracy (96.4% TPR, 96.4% precision, 96.4% F1-score) together with a low false-positive rate (3.6%), establishing the detector as an efficient, lightweight instrument for continuous wireless security monitoring. The resulting framework is reproducible and supports both real-time intrusion detection and forensic investigation, making it suitable for resource-constrained wireless environments.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Acknowledgments

The authors wish to thank the Graduate School of the State University of Northern Negros for the institutional support and resources provided during the conduct of this research.

Disclosure of conflict of interest

All author declares no conflict of interest.

Data availability

The experimental data supporting the findings of this study, including packet capture files (pcap-ng), anomaly detection logs (JSON), and performance metric computation scripts, are available from the corresponding author upon reasonable request. Due to the nature of the controlled laboratory evaluation, no external datasets were used; all attack and benign-traffic data were generated in-house using open-source tools (aireplay-ng and Scapy) as described in Section 3.

Software availability

The prototype WIDS implementation is written in Python 3.11.9 and relies on publicly available, open-source libraries: Scapy 2.5.0 for packet capture and manipulation, and the aircrack-ng 1.7 suite for wireless interface configuration. The source code, configuration files, and a reproducibility guide are available at the following public repository: https://github.com/jayzerg/wids-deauth-detector. All software dependencies and the hardware specifications required to replicate the experimental environment are detailed in this manuscript.

REFERENCES

[1] Vanhoef M. (2021). Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation. 30th USENIX Security Symposium (USENIX Security 21), 161-178.

[2] Schepers D, Vanhoef M and Piessens F. (2022). On the robustness of Wi-Fi deauthentication countermeasures. Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), 1-12.

[3] Sadia H, Farhan S, Ul Haq Y, Sana R, Mahmood T, Bahaj S A O and Khan A R. (2024). Intrusion detection system for wireless sensor networks: A machine learning based approach. IEEE Access, 12, 52565-52582.

[4] Thabit F, Can O, Abdaljlil S and Alkhzaimi H A. (2024). Enhanced an intrusion detection system for IoT networks through machine learning techniques: an examination utilizing the AWID dataset. Cogent Engineering, 11(1), 2378603.

[5] Rafique S H, Abdallah A, Musa N S and Murugan T. (2024). Machine learning and deep learning techniques for Internet of Things network anomaly detection — Current research trends. Sensors, 24(6), 1968.

[6] Diro A, Chilamkurti N, Nguyen V-D and Heyne W. (2021). A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms. Sensors, 21(24), 8320.

[7] Moharam M H, Ashraf K, Alaa H, Ahmed M and El-Hakim H A. (2025). Real-time detection of Wi-Fi attacks using hybrid deep learning models on NodeMCU. Scientific Reports, 15, 32544.

[8] Arreche O, Guntur T and Abdallah M. (2024). XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. Applied Sciences, 14(10), 4170.