International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 4, July – August 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

AI-Based Intelligent Wi-Fi Security and Adaptive Optimization

[1]Adithya K, [2]Darshan Chandrakanth Kulkarni, [3]Ekanth R, [4]Harsha A, [5]Dr. R. Sivakumar, [6]Dr. Shankara Gowda S R

[1,2,3,4]Department of ISE, Don Bosco Institute of Technology Kumbalagodu, Mysuru Road, Bengaluru, India
[5,6]Associate Professor, Department of ISE, Don Bosco Institute of Technology Kumbalagodu, Mysuru Road, Bengaluru, India
Email: adithya006gowda@gmail.com, ckdarshan255@gmail.com, ekanthgowda1999@gmail.com, harsha1828j@gmail.com, sivakumar.r@dbit.co.in, shankaracademics@gmail.com

Abstract:

Nowadays, almost all the household, corporate, and academic networks are based on wireless infrastructure, and this dependency creates a great number of targets for potential attacks. There are many unauthorized access attempts, MAC address spoofing, packet sniffing, brute force attacks, and denial-of-service floods. The traditional, rule-based security measures of these networks cannot withstand them anymore. Besides, there is another important problem associated with wireless communication - inefficiency of the bandwidth usage. However, this issue becomes obvious only when users start complaining about it. Instead of tackling security and efficiency separately, this work suggests a solution that will tackle both of these problems simultaneously. Captured live traffic is analyzed with Scapy and cross-checked through protocol analysis in Wireshark, while Nmap scans network to discover hosts, identify OS type, and map open ports/services. The behavioral features extracted through the process become the input for five classifiers: Random Forest, Decision Tree, and Support Vector Machine classifiers, as well as XGBoost and Isolation Forest models, which can detect anomalies that are not present in the labeled training data. The deviation from the device-specific baseline behavior can be detected due to the device-specific monitoring that was implemented, and, as a result, there will be the risk assessment for each individual device in the form of a risk score. This risk score serves as the basis for the recommendation engine (e.g. block device, require WPA3, disable WPS, change credentials), and, at the same time, as the input to the adaptive optimization layer for bandwidth allocation, channel selection, and traffic prioritization. Live Flask web application will be the user interface for this system. The majority of the research papers deal with the issues of security or performance, and they rarely solve both problems at once - this paper states that the feedback loop between security and performance should exist. It provides the architecture and a sixteen-week roadmap to develop it, and discusses how such a system can be assessed in the enterprise, healthcare, IoT, and smart home applications.

Keywords— *Wi-Fi Security, Intrusion Detection, Machine Learning, Random Forest, XGBoost, Support Vector Machine, Isolation Forest, Scapy, Wireshark, Nmap, Device Behavior Profiling, Risk Scoring, Adaptive Optimization, Recommendation Engine, Network Traffic Analysis.*

I. INTRODUCTION

The wired Ethernet cable infrastructure is almost completely gone. All the households, offices, hospitals, campuses, and public spaces are switched to wireless networks, and this transformation happened too quickly for the security standards to catch up. Each newly purchased smartphone, computer, or any other device is another possible entry point. As the number of connected devices rises, so does the attack surface - regardless of whether or not it was predicted by anyone.

Currently, the networks are mostly relying on a limited set of defense mechanisms - a password, WPA2/WPA3 encryption, and firewall. While these solutions do pretty good job against the attacks of the known signature, they fail miserably when faced with something new. The attacks like rogue access points, MAC address spoofing, deauthentication flood attacks, brute force attacks, and packet sniffing do not need to be very sophisticated to slip past the network, which recognizes only the threats it already knows.

Another issue exists besides this problem, and it is usually not considered with the security issues - performance. As the number of connected devices increases, the bandwidth allocation mechanism that was efficient before begins to produce congestions, latency, and dropped calls. Typically, the administrator of the network reacts to this problem and manually adjusts the configuration of the router, which works perfectly fine for a household but not scalable to campus with thousands of active connections.

Machine learning is the true way forward here not because it is trendy, but because of the nature of the task - recognizing patterns on a scale that humans simply cannot achieve. A trained model can flag traffic that doesn't match anything it has previously encountered, adjust its own thresholds as usage patterns shift, and make bandwidth decisions continuously instead of reactively. Such a system is discussed in this paper - a pipeline that captures network traffic at the packet level, identifies and fingerprints every connected device on a network, analyzes behavior and risk and adapts the network behavior

accordingly, without waiting for a user to see something is amiss.

However, there is another reason why automated network detection and behavior adjustment is necessary - once a network scales beyond a few hundred connected devices, manual monitoring of network becomes impossible not because of technical difficulties but due to cost and time considerations - it simply consumes too much resources to keep people watching dashboards 24/7. Automated bots that stuff credentials, coordinated DoS attacks happen too fast to allow reacting manually. In this sense, at least one additional layer of continuous automated detection is almost inevitable.

Nevertheless, this does not come without some downsides. Too aggressive approach turns into liability itself - legitimate devices get blocked, administrators do not trust alerts anymore and, thus, lose all the benefits from automation. That is why the design of the system proposed here incorporates both supervised and unsupervised models and uses the rate of false positives as one of the key metrics to measure and report.

The rest of the paper is organized as follows. Section II provides literature survey related to ML-driven wireless intrusion detection and network optimization. Section III identifies the drawbacks of the current solutions. Section IV states the problem that this work tries to solve. Section V presents the high-level architecture overview. Section VI discusses every module of the system in the same order as data is processed: packet capturing, network discovery, feature extraction, ML-driven detection, behavior analysis, risk estimation, network optimization, recommendations and finally, the dashboard. Section VII describes the machine learning techniques used in this work. Section VIII presents the development schedule of sixteen weeks. Section IX outlines testing procedure. Section X talks about potential use cases and limitations of the system. Section XI concludes the paper and offers ideas for future work.

II. LITERATURE SURVEY

1.AI Powered Network Intrusion Detection System using Machine Learning – Venkatesh et al. (ICCCES 2026)

The proposed model detects malicious network traffic (e.g., DDoS attacks) using a lightweight ML-based Network Intrusion Detection System (NIDS).

It extracts: preprocessed network traffic features (via missing-value handling, chi-square feature selection, one-hot encoding, and feature scaling) from a Kaggle traffic dataset, and feeds them into multiple classification models — Logistic Regression, AdaBoost, Random Forest, XGBoost, and SVM — trained to distinguish normal vs. anomalous (attack) traffic.

The model outputs a classification (normal/anomaly) with a confidence-based prediction; the system can then trigger a real-time alert, achieving up to 99.72% accuracy (Random Forest) with detection latency below 55 ms, while SVM performed poorly (~53.6% accuracy) due to the dataset's high dimensionality and class imbalance.

2. Security Analysis of WiFi-Based Sensing Systems: Threats From Perturbation Attacks (WiIntruder) – Cao et al. (IEEE TDSC 2026)

The proposed model (an attack framework, not a defense) generates adversarial perturbations to fool WiFi-based sensing systems (gesture recognition, respiration monitoring, authentication, localization) rather than detecting intrusions.

It extracts: CSI amplitude and phase parameters from wireless channels, and feeds them into an importance-weight-driven optimization (via particle swarm optimization) that maximizes disruption of user-state-specific features while accounting for device desynchronization and wireless propagation distortion, then uses an energy-based GAN to generate diverse perturbation surrogates for stealth.

The model outputs a black-box perturbation signal that an attacker (Eve) emits to contaminate legitimate CSI; this misleads the sensing model's decision (e.g., false respiratory readings, wrong gesture, rejected identity), achieving an average accuracy degradation of 72.9% across four sensing applications — with the paper also proposing an adversarial-training defense that restores >97% detection accuracy against the attack.

3.Anomatrix: WiFi Anomaly Detection – Chandushree & Shetty (AIDE 2026)

The proposed model detects anomalous WiFi traffic (illegal access, DoS-like behavior) using flow-level features.

It extracts: flow-level statistics (packet counts, byte ratios, flow duration, inter-arrival timing) from a synthetic WiFi traffic dataset, after scaling/encoding preprocessing, and feeds them into an XGBoost classifier trained to distinguish benign from anomalous flows.

The model outputs a "Normal"/"Anomalous" label via a real-time Flask API; the system can then alert on suspicious flows (e.g., forward-byte bursts, packet floods), achieving 92.62% accuracy with high precision (94.83%) and recall (95.65%), giving a low false-alarm rate suitable for real-time wireless monitoring.

4. Edge Intelligence-Enabled Network Intrusion Detection in Compute First Networking – Pei et al. (IEEE TCE 2025)

The proposed model detects network intrusions in IoT/edge environments while minimizing detection latency through computation offloading.

It extracts: network traffic converted into graph-structured data (feature matrix + adjacency matrix representing packet relationships) from PCAP files, and feeds them into a Node-Level Attention-Based Auto-Encoder Graph Convolutional Network (NAAE-GCN) — comprising an encoder, graph convolutional block, hierarchical attention block, and decoder — while a deep reinforcement learning-based broker decides whether to run detection locally on the gateway or offload it to a nearby edge server based on latency constraints.

The model outputs an intrusion classification along with the offloading decision; the system can then alert the security gateway, achieving 99.18%, 96.54%, and 98.89% accuracy on the CIC-Darknet, ISCXTor, and CIC-UNSW-NB15 datasets

respectively, outperforming graph-less DL models (CNN, LSTM, GRU) and other GNN baselines, while also reducing latency compared to full/local/random offloading strategies.

5. Intelligent Intrusion Detection in IoT with Explainable AI and Distributed Web Systems – Ileana et al. (Cyber-AI 2025)

The proposed model detects intrusions in IoT networks using a hybrid, explainable edge-cloud architecture.

It extracts: network traffic features (from the CIC-IDS2017 dataset) at the edge, feeding them into a lightweight Random Forest classifier deployed on resource-constrained devices (Raspberry Pi 4B), with SHAP computed in real time to explain each prediction; a distributed middleware (Docker Swarm, RESTful APIs, MQTT) synchronizes logs and correlates anomalies across edge nodes, while a centralized server handles deeper analysis using deep autoencoders, LSTMs, and generative models for retrospective threat detection.

The model outputs a classification (normal/attack) along with a SHAP-based explanation; the system can then flag the threat locally without needing constant cloud access, achieving 95.6% detection accuracy with <50 ms latency and low CPU/memory usage (edge CPU <30%, memory <250MB), and scaling consistently (middleware sync <120 ms) from 5 to 100 nodes with fault recovery via container replication.

6.AI-Powered Intrusion Detection Systems for Cloud Networks – Gurrapu et al. (ICNSoC 2025)

The proposed model is a hybrid AI-IDS that detects both known and novel attacks in dynamic cloud networks while resisting adversarial manipulation.

It extracts: cloud network traffic features, feeding them into a hybrid architecture combining supervised learning (to classify known attack signatures) and unsupervised learning (to flag anomalies), reinforced with adversarial training, poisoning detection, and evasion-detection modules to protect the training pipeline itself.

The model outputs an intrusion classification with certification/verification checks for reliability; the system achieved 98.5% accuracy on NSL-KDD and 97.8% on CICIDS2017, with the hybrid approach cutting false positive rate to 2% and detection latency to 30ms — clearly outperforming standalone supervised (120ms, 5% FPR) or unsupervised (150ms, 7.5% FPR) models.

7.Cybersecurity in Cloud Computing AI-Driven Intrusion Detection and Mitigation Strategies – Wang & Xie (IEEE Access 2025)

The proposed model detects intrusions in cloud environments by jointly modeling the spatial (topology) and temporal (event sequence) structure of cloud traffic.

It extracts: network traffic represented as a graph (nodes = VMs/services, edges = communication links) with feature vectors encoding protocol, IP/port interactions, and session dynamics, feeding them into a Transformer-based Spatio-Temporal Graph Neural Network (ST-GNN) trained with Sharpness-Aware Minimization (SAM) for better generalization and adversarial robustness.

The model outputs a multi-class intrusion classification; the system achieved the best results on CICIDS2017 (93% precision, 91% recall, 92% F1, 0.98 ROC-AUC) with <1 second detection latency, outperforming CNN, LSTM, Isolation Forest, and conventional GNN baselines, though it remains vulnerable to adversarial examples.

8.WiFi-based Intrusion Detection via Distribution Matching – Wang et al. (RoFi, ICASSP 2024)

The proposed model detects human intrusion using commodity WiFi signals instead of cameras or sensors.

It extracts: the autocorrelation function (ACF) of Channel State Information (CSI) power, which captures motion-induced correlation patterns in the wireless signal, and models it as Gaussian distributions — one for intrusion, one for non-intrusion — feeding these into a likelihood-ratio test that distinguishes real indoor intrusion (e.g., surreptitious entry, patrolling near the door) from non-threatening motion (e.g., outdoor passersby, animals).

The model outputs a detection statistic ($-\log \hat{l}$); the system can then trigger an alert or suppress false alarms from outdoor interference, achieving over 97.5% accuracy without requiring per-environment calibration.

9. A Host-based Intrusion Detection: Using Signature-based and AI-driven Anomaly Detection – Rehman et al. (ICoDT2 2024)

The proposed model detects both known and unknown cyber threats on a host machine using a Hybrid Intrusion Detection System (HIDS).

It extracts: preprocessed network/system traffic features (connection duration, protocol type, service, bytes transferred, etc.) from the NSL-KDD dataset, feeding them in parallel into a Gradient Boosting classifier (signature-based, matches known attack patterns) and a KNN classifier (anomaly-based, flags deviations from normal behavior), combining their outputs via a voting mechanism where Gradient Boosting takes precedence on conflicts.

The model outputs a threat classification with an alert; the system then notifies the SOC/NOC with threat details, achieving 90.37% accuracy — outperforming standalone signature-based (90.4%... same) and anomaly-based (87%) models individually, though it remains limited to a single host machine and needs sufficient training data.

10.AI-Based Methods to Detect and Counter Cyber Threats in Cloud Environments – Chaudhary et al. (ICEECT 2024)

The proposed model detects and automatically mitigates cyber threats in cloud infrastructure using anomaly detection and ML.

It extracts: system logs, network traffic logs, and user activity data collected via distributed log-gathering agents, preprocessed and normalized, then fed into deep learning neural networks (CNNs, RNNs) combined with ensemble methods (random forests, gradient boosting) trained via supervised learning.

The model outputs a threat classification; the system then automatically triggers predefined response actions (revoking

access, isolating compromised systems, patching), achieving 98.5% detection accuracy with a 1.2% false positive rate and 2.0% false negative rate — notably higher than the compared existing system (89.7% accuracy).

11.Machine Learning-based Intrusion Detection System using Wireless Sensor Networks – Pande et al. (ICAECT 2024)

This paper is a broad conceptual/survey study of IDS fundamentals (NIDS vs. HIDS, signature/anomaly/behavior-based detection, ML techniques) rather than a single proposed model.

It discusses: IDS architecture components (sensors, analysis engine, rule sets, alerting) and reviews how supervised learning (signature-based), unsupervised learning (anomaly-based), and deep learning (behavior-based, using RNNs/CNNs) are applied to classify network/host activity as normal or malicious, illustrated with a small demonstrative ML pipeline (decision tree, random forest, KNN) on a traffic dataset.

The demonstrative results show a confusion matrix (TP=10818, TN=11259, FP=991, FN=2127) and protocol-based traffic breakdowns (e.g., ~53% of TCP traffic normal, 47% attack); the paper's main contribution is a comprehensive review of IDS design, challenges (evasion, scalability, privacy), and IDS-IPS integration rather than a novel high-accuracy detector.

III. PROPOSED SYSTEM OVERVIEW

As mentioned above, the described system, named the AI-Based Intelligent WiFi Security and Adaptive Optimization framework, solves the problems identified in Section II by providing live network visibility instead of relying on a static dataset. All network traffic is captured, analyzed, and assessed - first for maliciousness, second for consistency with the sending device's own behavior, and third for network-optimization potential if it is deemed legitimate. This is the key difference between the proposed system and most of the existing solutions reviewed in Section II.

A. High-Level Architecture

Pipeline overview. The whole pipeline is shown in Figure 1. Traffic on the WiFi network is captured at the packet level using Scapy and validated through protocol analysis based on Wireshark for cases requiring deeper analysis. Simultaneously, Nmap continuously scans the network and fingerprints all devices - discovering hosts, detecting the operating system, identifying MAC addresses, and checking open ports and running services. Both flows feed into a feature-engineering phase, the output of which is passed to the machine learning layer (Random Forest, XGBoost, Isolation Forest, SVM), where attacks are detected.

The detection result is then used by the device behavior profiling module, which maintains a baseline behavior for each device and flags any deviation regardless of whether it matches a known attack signature. The combination of behavioral and detection results is used to calculate a per-device risk score, which drives both the recommendation engine (block device, enable WPA3, disable WPS, change credentials, restart router) and the adaptive optimization module (bandwidth allocation, channel selection, traffic prioritization). All collected data - packets, device information, prediction results, risk scores, alerts, and recommendations - is logged to the database and displayed on the dashboard.

End-to-End System Architecture

- Wi-Fi Network
- Scapy Packet Capture
- Wireshark Protocol Analysis
- Feature Extraction
- Nmap Device Discovery
- Machine Learning (Random Forest, XGBoost, Isolation Forest, SVM)
- Attack Detection
- Device Behaviour Profiling
- Risk Score Engine
- Adaptive Optimization
- Recommendation Engine
- Dashboard
- Reports

IV. SYSTEM MODULES

This section walks through each module in the order data actually flows through the pipeline, from raw packets to the dashboard a network administrator sees.

A. Packet Capture (Scapy) and Protocol Analysis (Wireshark)

Packet capturing and processing is done using Scapy to capture packets and extract TCP, UDP, DNS, ARP, and ICMP traffic. Wireshark is implemented as a deep protocol-analysis layer above the packet-capturing layer; it is used to validate the distribution of protocols, packet sizes, and packet counts, and to cross-check what has been captured with Scapy. The result of this process is a structured packet log, which is what the downstream modules consume - not the raw pcap file.

B. Network Discovery (Nmap)

Nmap runs on a schedule to provide an updated map of all devices attached to the network: host discovery, OS detection, MAC address resolution, and identification of open ports and running services. For example, scanning the device 192.168.1.25 might reveal that it is a Windows 11 host with ports 22, 80, and 443 open. This information is logged to the database and joined with the packet log via IP/MAC address, giving the downstream modules device-level context - not just "traffic from an IP address," but "traffic from a specific fingerprinted host with a specific service set."

C. Feature Engineering

The combination of captured packets and Nmap device context is converted into a feature vector per flow, aggregated over a sliding time window: source/destination IP, MAC address, protocol, packet size, packet rate, RSSI, DNS query count, throughput, and flow duration. Categorical features are

one-hot-encoded, and numerical features are min-max normalized.

D. Machine-Learning-Based Attack Detection

E. Device Behaviour Profiling

F. Risk Score Engine

G. Adaptive Optimization

H. Recommendation Engine

I. Dashboard and Reporting

D. Dataset Strategy

Training data is drawn from four public benchmark datasets - CICIDS2017, AWID, UNSW-NB15, and NSL-KDD - which together cover deauthentication, spoofing, flooding, and a broad range of intrusion categories. This is supplemented with traffic captured locally in a small testbed consisting of an access point, a handful of client devices, and a dedicated machine used to reproduce brute-force, ARP spoofing, and denial-of-service scenarios under appropriate ethical oversight. Combining public and local data is intended to reduce the risk of a model that performs well only on the exact network it was trained on.

V. ADVANTAGES AND APPLICATIONS

A. Advantages

- Bases detection on live packet capture and active network discovery rather than relying on static datasets alone.
- Identifies unauthorized access attempts, spoofing activity, and intrusions as they occur in real time.
- Builds a behavioural profile for each device, which helps surface compromised devices that stay quiet enough to evade signature-based matching.
- Consolidates multiple scattered signals into one actionable per-device risk score.
- Limits false-positive alerts by combining supervised classification with unsupervised anomaly detection.
- Automatically adjusts bandwidth allocation, channel selection, and traffic prioritization in response to shifting demand.
- Presents concrete recommended actions to the administrator rather than raw, unfiltered alerts.

B. Applications

The system is applicable across a range of network environments that depend on continuous security monitoring and stable performance. In enterprise settings, this includes corporate offices, IT and data-center infrastructure, and banking.

VI. CONCLUSION AND FUTURE WORK

This paper proposes an AI-based Intelligent Wi-Fi Security and Adaptive Optimization System that integrates network security monitoring with performance optimization in a unified framework. The proposed architecture combines packet analysis using Scapy and Wireshark, network discovery through Nmap, supervised machine learning algorithms such as Random Forest, Decision Tree, Support Vector Machine (SVM), and XGBoost for known attack detection, and Isolation Forest for anomaly detection. Behaviour profiling and risk scoring are intended to aggregate multiple security indicators into a unified risk assessment, while a recommendation engine is designed to suggest appropriate security and optimization actions. The proposed framework aims to enhance both network protection and performance by continuously monitoring traffic and providing intelligent recommendations.

Future work will involve implementing the proposed framework and evaluating it using both real-world and simulated network traffic. Performance will be assessed based on metrics such as detection accuracy, precision, recall, F1-score, false positive rate, risk-scoring effectiveness, and overall network performance improvements. Additional research directions include reinforcement learning for dynamic channel and power optimization, federated learning for privacy-preserving model training across multiple access points, deployment on resource-constrained edge devices, and extending the framework to support emerging wireless technologies such as Wi-Fi 7 and 5G/6G networks.

REFERENCES

[1] V. Venkatesh et al., “AI Powered Network Intrusion Detection System Using Machine Learning,” in Proc. Int. Conf. Communication, Computing and Electronics Systems (ICCCES), 2026.

[2] Y. Cao et al., “Security Analysis of WiFi-Based Sensing Systems: Threats From Perturbation Attacks (WiIntruder),” IEEE Transactions on Dependable and Secure Computing, 2026.

[3] C. Chandushree and P. Shetty, “Anomatrix: WiFi Anomaly Detection,” in Artificial Intelligence and Data Engineering (AIDE), 2026.

[4] Y. Pei et al., “Edge Intelligence-Enabled Network Intrusion Detection in Compute First Networking,” IEEE Transactions on Consumer Electronics, vol. 71, no. 1, 2025.

[5] I. Ileana et al., “Intelligent Intrusion Detection in IoT with Explainable AI and Distributed Web Systems,” in Cyber-AI Conference, 2025.

[6] G. Gurrapu et al., “AI-Powered Intrusion Detection Systems for Cloud Networks,” in Proc. Int. Conf. Network Softwarization (ICNSoC), 2025.

[7] Y. Wang and X. Xie, “Cybersecurity in Cloud Computing: AI-Driven Intrusion Detection and Mitigation Strategies,” IEEE Access, vol. 13, 2025.

[8] Y. Wang et al., “WiFi-based Intrusion Detection via Distribution Matching (RoFi),” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2024.

[9] A. Rehman et al., “A Host-based Intrusion Detection: Using Signature-based and AI-driven Anomaly Detection,” in Proc. Int. Conf. on Digital Technologies and Data Science (ICoDT2), 2024.

[10] A. Chaudhary et al., “AI-Based Methods to Detect and Counter Cyber Threats in Cloud Environments,” in Proc. Int. Conf. Electrical, Electronics, Communication and Computing Technologies (ICEECT), 2024.

[11] A. Pande et al., “Machine Learning-based Intrusion Detection System using Wireless Sensor Networks,” in Proc. Int. Conf.

Advances in Electrical, Computing and Communication Technologies (ICAECT), 2024.

