

Real-Time Wi-Fi Network Traffic Anomaly Detection Using a Supervised Random Forest Model

Manish M. Gowda, BS Abhishek, N. Manjunatha, Gowdra Vijay Kumar, Nandakumar

Department of Computer Science and Engineering, Visvesvaraya Technological University, India

Published October 2026

Abstract

The Wi-Fi Real-Time Anomaly Detection System is a machine-learning-based network security solution designed to monitor network traffic and identify abnormal or potentially malicious activities in real time. The system captures live network packets using Npcap and processes them using Scapy to construct network flows. From each flow, 68 network-flow features are extracted, representing characteristics such as packet counts, packet lengths, flow duration, inter-arrival times, protocol flags, and transmission rates. The extracted features are preprocessed using the same scaling and transformation pipeline applied during model development and are then provided to a Random Forest classifier for traffic classification. The system is developed using the CIC-IDS2017 dataset, which contains both benign and attack traffic. The training and evaluation data represent several attack categories, including PortScan, DDoS, DoS, FTP-Patator, SSH-Patator, Bot activity, Heartbleed, and web attacks. The V6 model classifies observed traffic into BENIGN or ATTACK categories. For real-time monitoring, the prediction results are continuously stored and displayed through a Streamlit dashboard, which provides Wi-Fi information, traffic statistics, source-wise summaries, live network flows, prediction confidence, and anomaly alerts. The proposed system can be applied in companies, colleges, and other organizational networks to continuously monitor traffic and provide an early indication of suspicious network behaviour. The system also supports controlled testing using labelled attack flows to demonstrate its anomaly-detection capability.

Keywords: Wi-Fi security; anomaly detection; Random Forest; network traffic monitoring

1 Introduction

The implementation integrates real-time packet capture, flow construction, 68-feature extraction, preprocessing, Random Forest classification, and Streamlit visualization for continuous Wi-Fi anomaly detection. Our earlier study investigated Wi-Fi traffic anomaly detection using the Isolation Forest approach [1]. Building upon that work, the present study focuses on the implementation and evaluation of the developed network traffic anomaly detection pipeline. The present implementation incorporates packet capture using Npcap and Scapy, flow construction, a fixed 68-feature representation, stored preprocessing, supervised classification, real-time logging, and a Streamlit-based monitoring interface. Isolation Forest is attractive for a lightweight prototype because it is an unsupervised method based on recursive random partitioning. The original method showed that anomalous observations tend to be isolated with shorter path lengths [5].

2 Related work

2.1 Wireless conditions and the base paper

The principal base paper, published in IEEE Access in 2025, investigates dynamic spectrum access in dense Wi-Fi environments using deep reinforcement learning. It models multiple access points and heterogeneous traffic in an NS-3 environment and uses environmental observations such as interference, traffic load, delay, and signal-related information for dynamic channel decisions [2].

2.2 Network anomaly detection

Tao et al. proposed a parallel network-traffic anomaly detection method based on Isolation Forest and Spark for large-scale traffic processing [3].

2.3 Adaptive wireless IDS

IDS-Twin moves wireless intrusion detection toward streaming and self-adaptive anomaly modelling using a digital-twin-based replay environment and contrastive learning [4].

3 Implementation

3.1 Packet capture and traffic collection

The Phase-2 system was implemented in a Windows environment for real-time observation of Wi-Fi network traffic. Npcap was used to capture packets from the selected wireless interface, while Scapy was used for packet processing. The captured traffic was continuously monitored and converted into flow-level information for subsequent analysis.

3.2 Flow construction

Captured packets were grouped into network flows instead of being analysed individually. Each flow records communication characteristics such as source and destination information, ports, transport protocol, duration, packet count, and related traffic statistics. This flow-based representation enables temporal and directional properties of network communication to be considered during detection.

3.3 Feature extraction and preprocessing

Each constructed flow was converted into a fixed **68-feature representation**. The features include packet and byte counts, packet-length statistics, inter-arrival-time characteristics, transport flags, directional rates, subflow statistics, and active/idle measurements. The extracted feature vector was passed through the stored preprocessing pipeline before classification.

3.4 Machine learning classification

The preprocessed flow features were provided to the trained Random Forest classifier, which generated a binary BENIGN or ATTACK decision along with its confidence. The binary classification provides a direct indication of whether the observed flow resembles the traffic patterns learned during model training.

4 Results and performance evaluation

4.1 Experimental setup

The developed Phase-2 system was evaluated using a supervised machine-learning approach for network-flow classification. Unlike an unsupervised anomaly scoring approach, the V6 model was trained to distinguish normal network behaviour from traffic patterns associated with malicious activity. The final V6 feature set consists of 68 numerical network-flow attributes. These attributes describe multiple characteristics of a

communication flow, including packet counts, packet lengths, flow duration, inter-arrival times, transmission rates, TCP flags, header information, window sizes, and active/idle behaviour. Before classification, the selected features are subjected to the same preprocessing pipeline used during model development. Logarithmic transformation is applied to appropriate traffic-rate features, followed by feature scaling. The resulting feature vector is then supplied to the trained Random Forest classifier.

For independent evaluation, a separate Tuesday traffic dataset containing 445,909 records was used. The dataset consisted of 432,074 BENIGN flows, 7,938 FTP-Patator flows, and 5,897 SSH-Patator flows. No feature mismatch occurred during testing, as all 68 features expected by the V6 model were available in the independent dataset.

4.2 Classification performance

The proposed Random Forest classifier achieved the performance shown in Table 1 on the independent Tuesday test data.

Table 1. Performance of the proposed Random Forest classifier on the independent Tuesday test data.

Metric	Proposed system performance
Accuracy	99.99%
Precision	99.65%
Recall	100.00%
F1-Score	99.82%

The high recall indicates that the evaluated attack flows were successfully identified by the classifier. The precision value also indicates that only a small number of normal flows were incorrectly classified as malicious. The overall accuracy was approximately 99.99%, demonstrating strong classification performance on the independent evaluation set.

4.3 Confusion matrix analysis

Table 2 gives the binary confusion matrix obtained in the Tuesday evaluation.

Table 2. Binary confusion matrix for the Tuesday evaluation.

Actual / Predicted	BENIGN	ATTACK
BENIGN	432,025	49
ATTACK	0	13,835

The classifier correctly identified 432,025 benign flows and 13,835 attack flows. Only 49 benign flows were incorrectly classified as attacks, while no attack flows in this test set were classified as benign.

4.4 Attack-wise detection performance

The independent test also enabled evaluation of the two attack classes present in the Tuesday test set (Table 3).

Table 3. Attack-wise detection performance on the Tuesday test set.

Attack type	Total samples	Detected	Detection Rate
FTP-Patator	7,938	7,938	100%
SSH-Patator	5,897	5,897	100%

Both FTP-Patator and SSH-Patator traffic were successfully detected in the independent evaluation. The result demonstrates that the learned network-flow patterns were effective for the attack categories represented in this test set.

4.5 Dashboard output

A real-time monitoring dashboard was developed using Streamlit to display the active Wi-Fi network information, including SSID, signal strength, channel, frequency band, and transmission rates, together with live network traffic statistics and anomaly detection results.

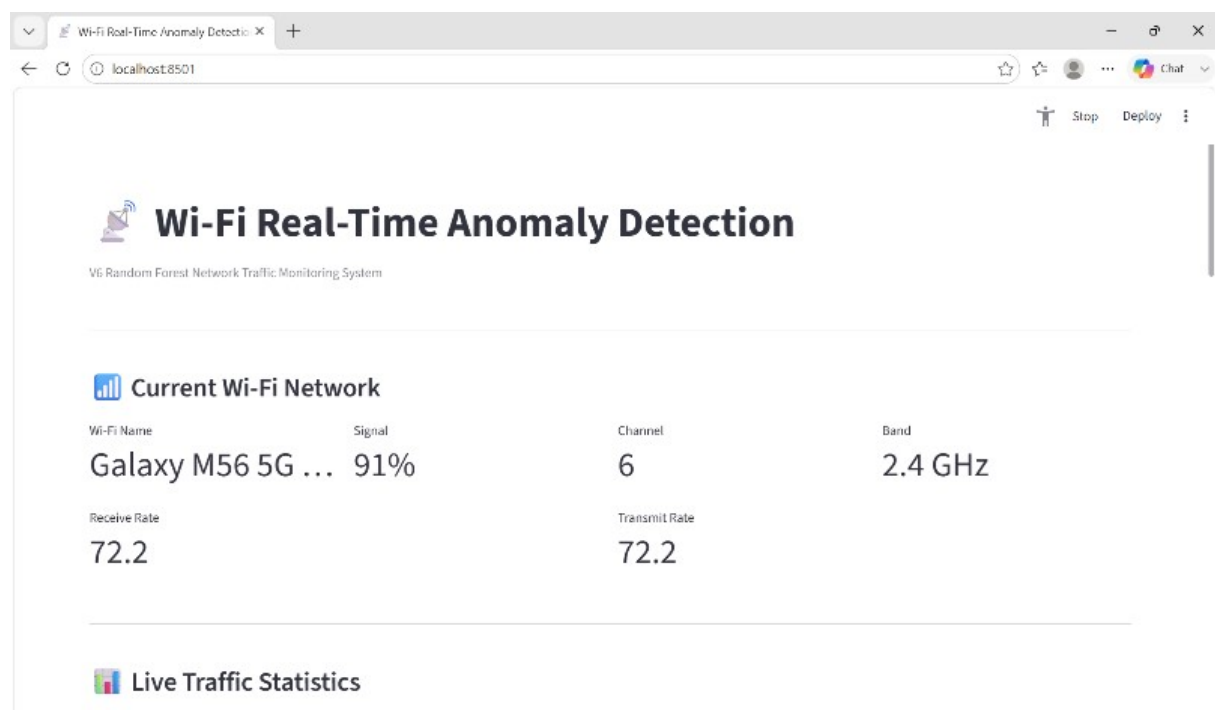


Figure 1. Real-time Wi-Fi network monitoring and anomaly detection dashboard.

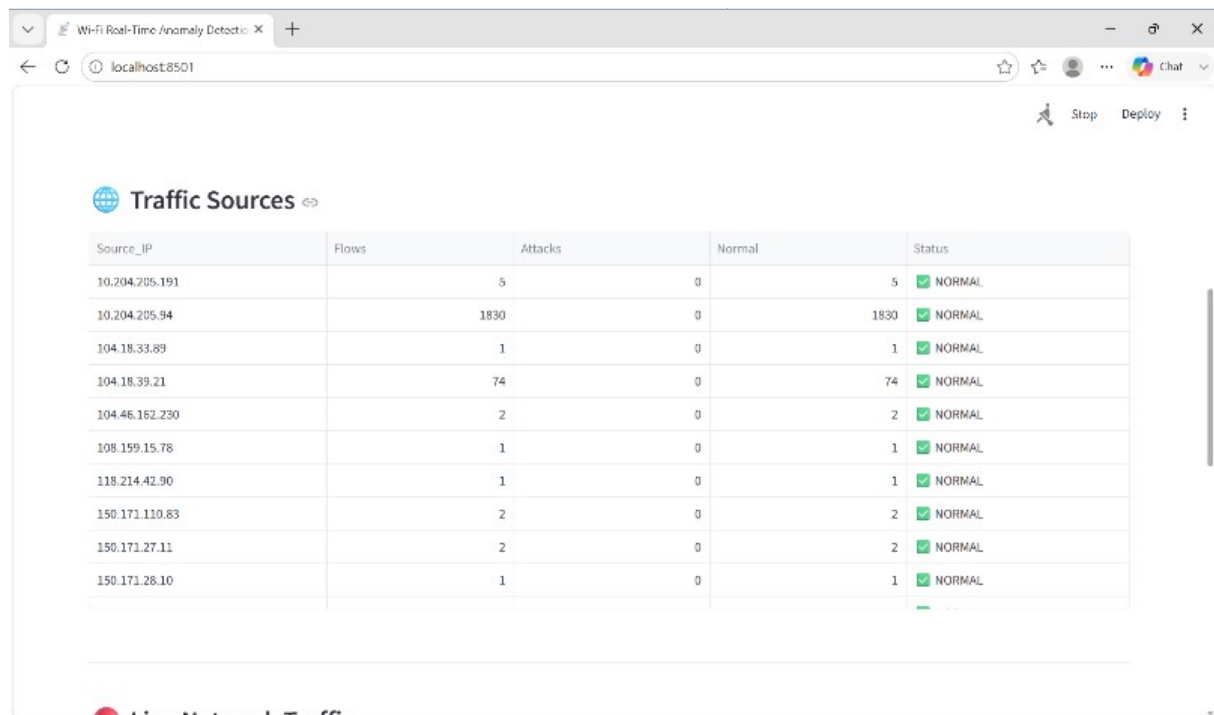


Figure 2. IP addresses communicating with the monitored laptop.

Table 4. Fields of the source-wise summary of IP addresses communicating with the monitored laptop and their observed flows.

Field	Meaning
Source_IP	IP address involved in the observed traffic
Flows	Number of network flows associated with that IP
Attacks	Number of those flows classified as attacks
Normal	Number classified as normal
Status	Overall status based on the detection results

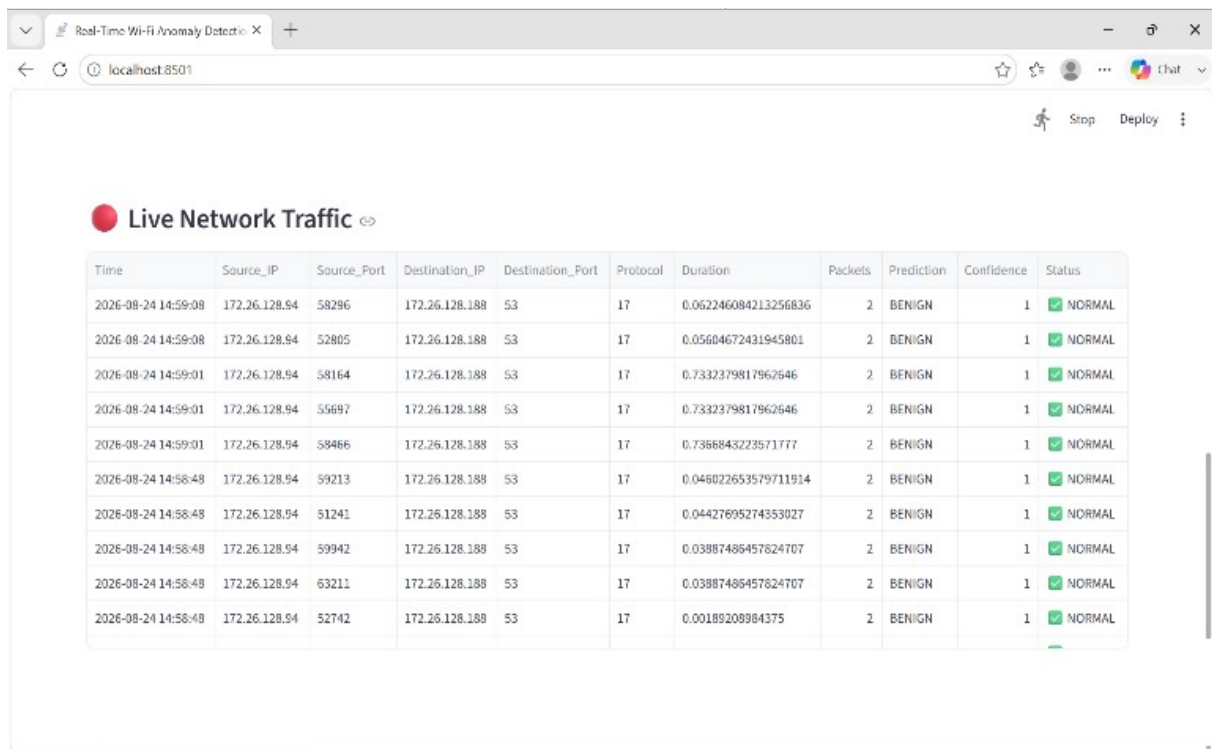


Figure 3. Live network traffic section of the dashboard.

Table 5. Columns of the live network traffic view, showing what the back end detects in real time.

Column	Meaning
Time	When the network flow was observed
Source_IP	Device/source generating the traffic
Source_Port	Port used by the source
Destination_IP	Remote/local destination
Destination_Port	Destination service port
Protocol	17 = UDP, 6 = TCP
Duration	Duration of that flow in seconds
Packets	Number of packets in the flow
Prediction	Our model's classification
Confidence	Model confidence
Status	Human-readable result: NORMAL or ATTACK

4.6 Flow features

The main flow features are computed as follows:

$$\text{Flow Duration} = t_n - t_1 \quad (1)$$

$$\text{Total Fwd Packets} = N_f \quad (2)$$

$$\text{Total Bwd Packets} = N_b \quad (3)$$

where N_f and N_b are the numbers of packets travelling in the forward and backward directions. The total forward and backward bytes are

$$\text{Total Length of Fwd Packets} = \sum_{i=1}^{N_f} L_i \quad (4)$$

$$\text{Total Length of Bwd Packets} = \sum_{j=1}^{N_b} L_j \quad (5)$$

where L_i (L_j) is the length of the i th forward (j th backward) packet, and t_1 and t_n are the times of the first and last packets of the flow.

5 Conclusion

The proposed system provides an effective approach for real-time network anomaly detection using 68 flow-level features and a Random Forest classifier. The system successfully combines live traffic monitoring with machine-learning-based classification and a user-friendly dashboard. The results demonstrate its potential for practical network-security monitoring. Future work can focus on detecting additional attack types and improving real-time deployment across larger networks.

References

- [1] Gowda M, Abhishek BS, Kumar V, Manjunatha N, Nandakumar. Wi-Fi traffic anomaly detection using Isolation Forest algorithm. *Iconic Research and Engineering Journals*. 2026;9(10):452–453.
- [2] Alauthman A, Shraa T. Deep reinforcement learning-driven dynamic spectrum access in dense Wi-Fi environments. *IEEE Access*. 2025;13:178663–178690. <https://doi.org/10.1109/ACCESS.2025.3621489>
- [3] Tao X, Peng Y, Zhao F, Zhao P, Wang Y. A parallel algorithm for network traffic anomaly detection based on Isolation Forest. *International Journal of Distributed Sensor Networks*. 2018;14(11). <https://doi.org/10.1177/1550147718814471>
- [4] El Alami H, Rawat DB. IDS-Twin: Digital twin-based intrusion detection systems for wireless networks. *IEEE Networking Letters*. 2025;7(4):255–259. <https://doi.org/10.1109/LNET.2025.3606789>
- [5] Liu FT, Ting KM, Zhou ZH. Isolation forest. In: *Proceedings of the 8th IEEE International Conference on Data Mining*; 2008; Pisa, Italy. p. 413–422. <https://doi.org/10.1109/ICDM.2008.17>
- [6] Xu H, Pang G, Wang Y, Wang Y. Deep isolation forest for anomaly detection. *arXiv:2206.06602 [Preprint]*. 2022.