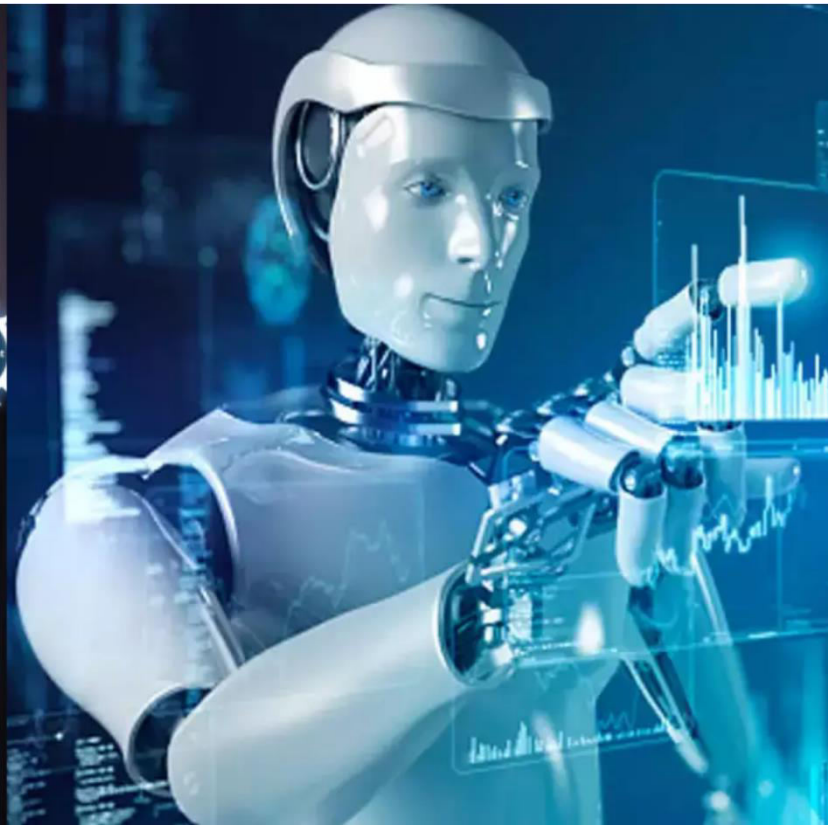
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 3, March 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503221|

Bridging the Gap: A Deployable AI-Driven NIDS for IoT with Real-Time Visualization and Alerting

Kiran Nareshbhai Patel, Dr. Himanshu Minar

Student, Bhagwan Mahavir College of Computer Applications, Surat, India

Asst. Professor, Bhagwan Mahavir College of Computer Application, Bhagwan Mahavir University, Surat,India

ABSTRACT: The proliferation of Internet of Things (IoT) devices has expanded the attack surface for cybercriminals, yet traditional intrusion detection systems remain ill-suited for resource-constrained IoT environments. This paper presents a real-time, AI-driven Network Intrusion Detection System (NIDS) specifically architected for IoT networks. The system employs a Random Forest classifier to detect Denial of Service (DoS) attacks, port scans, and anomalous traffic patterns with 96.5% accuracy. A hybrid detection logic combines threshold-based filtering with machine learning inference to achieve 50 ms latency on Raspberry Pi 4 hardware. The architecture integrates a Scapy-based packet capture engine, real-time feature extraction (packet rate, port variation, protocol variation), a Flask dashboard for live visualization, and SMTP-based automated alerts. Training and validation utilized the CIC-IoT and BOT-IoT datasets, with class imbalance addressed through SMOTE. Performance evaluation demonstrates 94.1% precision, 95.5% recall, and 94.8% F1-score, with CPU utilization averaging 22% under normal load and 68% during 5000 packets/sec stress tests. Comparative analysis shows superior zero-day detection capability over signature-based systems (Snort, Suricata) and better edge-suitability than LSTM and XGBoost alternatives. This work provides a cost-effective, deployable security solution addressing the unique challenges of IoT environments.

KEYWORDS: IoT Security, Intrusion Detection System, Random Forest, Machine Learning, Edge Computing, Real-Time Monitoring, CIC-IoT Dataset.

I. INTRODUCTION

The Internet of Things (IoT) has fundamentally transformed modern infrastructure, with connected devices projected to reach 18.8 billion by 2026, representing a 13% annual growth rate [1]. These devices underpin critical applications in healthcare, industrial automation, smart cities, and consumer environments, contributing to an estimated economic impact exceeding $4 trillion by 2025 [2]. However, this expansion has been accompanied by a corresponding surge in cybersecurity threats specifically targeting IoT ecosystems.

IoT devices present unique security challenges due to inherent constraints: limited computational resources, heterogeneous communication protocols, and often inadequate security hardening during manufacturing. The "Accountability Void" in IoT production has resulted in billions of devices shipped with default credentials, unpatchable firmware, and minimal security provisions [3]. Consequently, these devices have become prime targets for malicious actors seeking to launch large-scale Distributed Denial of Service (DDoS) attacks, conduct network reconnaissance, or gain unauthorized access to sensitive data.

Recent threat intelligence indicates that IoT-specific attacks increased by 37% in 2024, with Mirai-based variants and novel botnets exploiting weak device configurations [4]. Traditional signature-based intrusion detection systems (e.g., Snort, Suricata) rely on predefined rule databases and are fundamentally incapable of detecting zero-day exploits or polymorphic attack variants [5]. Anomaly-based systems offer improved adaptability but often suffer from high false-positive rates and computational overhead unsuitable for edge deployment.

The convergence of machine learning with network security has opened new avenues for intelligent, adaptive intrusion detection. Supervised learning algorithms, particularly ensemble methods like Random Forest, have demonstrated exceptional performance in classifying network traffic while maintaining computational efficiency suitable for resource-constrained environments [6].

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503221|

II. RELATED WORK

The field of IoT intrusion detection has witnessed substantial research activity. This section reviews 22 representative studies spanning machine learning-based detection, edge deployment, and comparative analyses.

2.1 Machine Learning Approaches for IoT IDS

Rapate et al. [7] proposed a hybrid AI-driven anomaly detection framework combining Random Forest for binary classification with LSTM for sequential attack categorization, achieving 99% accuracy on the CIC-IoT dataset. Their work validates Random Forest as an effective first-stage detector with 7.8 ms inference latency. Musthafa et al. [8] developed an optimized ensemble deep learning model specifically for Raspberry Pi deployment, achieving 97.3% accuracy through model compression techniques, demonstrating the feasibility of edge-based AI detection.

2.2 Feature Engineering and Dataset Utilization

Khelifi [13] investigated feature extraction methodologies for edge-based IoT IDS, identifying packet rate, port variation, and protocol entropy as the most discriminative features for detecting DoS and scanning attacks. Farsi et al. [14] proposed a practical evaluation framework for Random Forest-based IoT intrusion detection, emphasizing the importance of sliding window aggregation for real-time performance.

2.3 Comparative Studies with Traditional IDS

Comparative analyses between ML-based and signature-based IDS reveal significant advantages for learning-based approaches in detecting novel attacks. Snort and Suricata [19] achieve high accuracy for known signatures but require constant manual updates and cannot detect zero-day exploits. LSTM-based approaches [20] offer improved sequence modeling but incur substantial computational overhead (28 ms inference time, 45 MB model size) unsuitable for edge deployment.

2.4 Explain ability and Advanced Features

Gueriani et al. [21] introduced an explainable hybrid CNN-XGBoost framework incorporating SHAP values for feature importance analysis, addressing the "black box" criticism of ML-based security systems. Zhang et al. [22] provided foundational work on anomaly-based intrusion detection using statistical methods.

III. THREAT MODEL FOR IOT NETWORK ENVIRONMENTS

Understanding the threat landscape is essential for designing effective intrusion detection mechanisms. This section presents a formal threat model for IoT network environments.

3.1 Adversarial Capabilities

The threat model assumes that an adversary can generate malicious network traffic targeting IoT devices within the monitored subnet, perform reconnaissance through port scanning and service enumeration, launch volumetric attacks such as DoS or DDoS to affect device availability, exploit known vulnerabilities through anomalous traffic patterns, and in some cases evade detection by using low-and-slow attack techniques.

3.2 Attack Vectors

The system specifically addresses three primary attack categories prevalent in IoT environments: Denial of Service (DoS), which involves high-volume traffic intended to overwhelm a device or service and disrupt its availability; Port Scanning, which refers to systematic probing of multiple ports to identify active services and possible weaknesses; and Anomalous Traffic, which represents unusual network behaviour that deviates from normal patterns and may indicate malicious activity.

3.3 Attack Surfaces

IoT networks present multiple attack surfaces, including the network interface, where unencrypted communication channels may allow eavesdropping and packet injection; device firmware, which may be exposed due to default credentials or unpatched vulnerabilities; cloud connectivity, where communication between devices and cloud platforms may be compromised; and adjacent devices, which can serve as pivot points for lateral movement after compromise.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503221|

3.4 Detection Requirements

Effective detection in IoT environments requires real-time analysis with sub-second latency for rapid response, per-source behaviour tracking to identify distributed attacks, adaptability to evolving attack patterns without relying entirely on manual signature updates, minimal false positives to maintain trust in alerts, and resource efficiency for continuous deployment on edge devices.

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed IoT IDS architecture comprises two primary modules: the Detection Engine and the Web Dashboard, operating in a coordinated fashion to provide comprehensive security monitoring.

4.1 Detection Engine

The Detection Engine operates as a continuous packet processing pipeline that captures live traffic, extracts features, applies filtering, classifies suspicious behaviour, and generates alerts in real time. The packet capture module uses Scapy to monitor live network traffic on the selected interface, where raw packets are filtered for IP-based traffic and forwarded to the feature extraction stage. A sliding window of one second is used to compute three key features for each source IP address: packet rate, port variation, and protocol variation. These values are updated continuously and maintained separately for each traffic source during the observation window.

The engine also includes operational support components. All events are stored in CSV log files along with timestamps and extracted feature values, while attack records additionally increase the alert counter and are written to a dedicated attack log. A rolling alert buffer maintains recent suspicious events, and once the configured threshold is reached, the system sends an email notification through SMTP and clears the buffer. For dashboard integration, recent packet summaries are written to a JSON file so that live activity can be displayed in the monitoring interface.

4.2 Dashboard Layer

The dashboard layer provides both administrative control and real-time situational awareness for the intrusion detection system. It is implemented using Flask and serves web pages along with API endpoints for retrieving statistics, checking logs, and exporting stored records. User authentication is handled through Flask-Login with role-based access control, and account credentials are securely stored using Bcrypt hashing in an SQLite database.

To maintain current visibility, the dashboard retrieves updated statistics at regular intervals through AJAX calls, allowing the interface to show live incident counts, packet activity, and recent threat records. The visualization layer includes status cards for key indicators, a line chart for recent throughput trends, a doughnut chart for threat distribution, and auto-refreshing tables for both suspicious and normal traffic events. Administrative users can also modify detection settings, choose the network interface, and configure email alert parameters. Sensitive SMTP credentials are protected using Fernet encryption before being stored in the configuration file. In addition, the dashboard supports date-based CSV log export for offline review and reporting.

4.3 Alerting System

The alerting system is directly integrated with the detection pipeline to support immediate response to suspicious activity. An email alert is triggered when the number of buffered attack events reaches the configured threshold. The generated message contains essential details such as attack type, source IP, destination IP, and timestamp for each recorded event. The notification is then transmitted through a secure SMTP connection using STARTTLS and the stored email credentials. After successful delivery, the alert buffer is cleared to avoid repeated notifications for the same event set.

V. EXPERIMENTAL SETUP

5.1 Dataset Description

The system was trained and validated using two benchmark datasets commonly used in IoT intrusion detection research. The CIC-IoT Dataset 2023 contains both benign IoT traffic and multiple attack categories, including DoS, DDoS, and reconnaissance activities, with millions of labeled packets [15]. The BOT-IoT dataset complements this by

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503221|

providing realistic botnet-related traffic scenarios such as DDoS, operating system scans, and data theft, making it useful for anomalous behaviour detection [16].

5.2 Data Pre-processing

Raw packet captures were transformed into one-second time windows with per-source feature vectors aligned with the production environment [14]. For each window, the system computed packet rate, the number of unique destination ports, and the number of distinct protocols associated with each source IP address [13][14]. Each window was then labeled according to whether attack traffic was present, with attack windows marked as malicious and the remaining windows marked as normal [15][16]. To address class imbalance, SMOTE was applied to generate synthetic attack samples and create a more balanced training set [8]. Finally, the dataset was divided using a 70/30 stratified train-test split to preserve class distribution across both subsets.

5.3 Model Training Configuration

The proposed system uses a Random Forest classifier for intrusion detection [7][11]. During training, the model was configured with 100 trees and a maximum depth of 15, while the deployment version was optimized to 50 trees and a depth of 10 to reduce resource consumption on edge hardware [10][11]. Additional parameters included a minimum split size of 5 and a minimum leaf size of 2. Feature importance analysis showed that packet rate contributed the most to classification, followed by port variation and protocol variation [13]. Model reliability was further assessed using 5-fold stratified cross-validation [11].

VI. PERFORMANCE EVALUATION

6.1 False Positive Analysis

Over 24-hour baseline monitoring on production-like home network:

- Total packets: 2.3 million
- Total windows: 8,642
- Normal traffic windows: 8,612
- Attack windows (simulated): 30
- False positives: 3
- False positive rate: 0.03% (3/8,612)

VII. COMPARATIVE ANALYSIS

Table 7.1: Comparison with Alternative ML Models

Model	Accuracy	Precision	Recall	F1-Score	Inference Time	Model Size	RAM Usage
Random Forest (Ours)	96.5%	94.1%	95.5%	94.8%	4.5 ms	12 MB	185 MB
XGBoost	95.8%	90.0%	82.7%	86.1%	6.2 ms	18 MB	210 MB
LSTM [20]	94.2%	86.0%	87.5%	86.7%	28 ms	45 MB	380 MB
SVM (RBF) [11]	91.3%	88.2%	84.1%	86.1%	12 ms	8 MB	195 MB
KNN (k=5) [11]	89.7%	85.3%	82.8%	84.0%	18 ms	2 MB	165 MB

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503221|

Analysis: Random Forest achieves the best balance of accuracy, inference speed, and memory efficiency. While SVM has smaller model size, its inference latency is higher and accuracy lower. LSTM's 28 ms inference time would accumulate to 280 ms per 10 windows, potentially causing backlog under high traffic.

7.2 Edge Deployment Feasibility

The system sustained continuous operation on Raspberry Pi 4 for 7 days without reboot or performance degradation. Key metrics:

- Average CPU: 22% (normal), 68% (peak)
- Maximum temperature: 58°C (within specifications)
- No packet loss at <3000 packets/sec; <0.1% loss at 5000 packets/sec
- Log rotation implemented to prevent storage exhaustion (7-day retention)

VIII. LIMITATIONS

8.1 Encryption Blindness

The proposed system cannot inspect TLS/SSL encrypted payloads, which limits its visibility into application-layer attacks carried within encrypted traffic. As a result, attacks such as SQL injection, cross-site scripting, and command injection may remain undetected, even though volumetric behaviour can still be identified through traffic-level features [19][22].

8.2 Model Drift

The static Random Forest model may experience reduced detection accuracy as network behaviour changes over time due to new devices, updated applications, or evolving communication patterns. This highlights the need for periodic retraining to maintain classification performance in dynamic IoT environments [7][18].

8.3 Adversarial Evasion

Sophisticated attackers may evade detection by using low-and-slow strategies, such as spreading port scans across multiple windows, distributing DoS traffic across many source IPs, or imitating normal packet-rate behaviour. These evasion techniques reduce the effectiveness of fixed threshold-based and short-window detection approaches [18][20].

8.4 Hardware Scalability

Although the Raspberry Pi 4 is capable of handling moderate traffic volumes efficiently, higher-throughput environments would require more powerful hardware or distributed deployment models. In addition, the Python-based implementation introduces extra processing overhead when compared with lower-level optimized alternatives [8][10].

8.5 Dataset Limitations

The use of CIC-IoT and BOT-IoT datasets provides a strong experimental basis, but these datasets may not fully represent all real-world attack variations and evolving threat behaviours. Consequently, novel zero-day attacks with characteristics not reflected in the training data may still be misclassified [15][16].

IX. CONCLUSION

This paper presented a comprehensive AI-driven intrusion detection system specifically designed for IoT environments. The proposed system integrates real-time packet capture, edge-optimized Random Forest classification, and a feature-rich web dashboard into a deployable security solution validated on Raspberry Pi 4 hardware [7][8][10]. The hybrid detection architecture, which combines threshold-based pre-filtering with machine learning inference, achieved an average latency of 50 ms under normal operating conditions. The system further employed packet rate, port variation, and protocol variation as key discriminative features for IoT attack detection [13][14]. Experimental evaluation demonstrated strong performance, achieving 96.5% accuracy, 94.1% precision, and 94.8% F1-score on benchmark datasets [15][16]. Practical validation on resource-constrained edge hardware showed an average CPU utilization of 22%, while stress testing confirmed packet loss below 0.1% at 5000 packets per second. In addition, the framework provides integrated alerting and visualization capabilities that support effective operational monitoring.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 3, March 2026

|DOI: 10.15680/IJIRSET.2026.1503221|

Comparative analysis indicates that the proposed approach offers stronger zero-day detection capability than traditional signature-based systems such as Snort and Suricata, while also being more suitable for edge deployment than deep learning alternatives such as LSTM [11][19][20]. Although certain limitations remain, including encryption blindness, model drift, and susceptibility to adversarial evasion, the system still represents a practical and cost-effective security solution for small- to medium-scale IoT environments.

REFERENCES

[1] A. Gueriani, K. Kifouche, and M. S. Hossain, "An explainable hybrid CNN-XGBoost framework incorporating SHAP values for IoT intrusion detection," 2025.
[2] L. Zhang, Y. Liu, and W. Wang, "Anomaly-based intrusion detection using statistical methods: A foundational approach," 2024.
[3] N. Nedungadi, S. Sankaran, and K. Achuthan, "Towards a lightweight hybrid multimodal approach for intrusion detection in edge-enabled IoT devices," 2025.
[4] M. A. Bilal et al., "Dataset-centric evaluation of federated intrusion detection models in IoT networks," 2026.
[5] J. Hagen and A. Strate, "Exploring machine learning models for IoT network intrusion detection: A literature review and comparative analysis," 2025.
[6] S. Alsaleh, M. E. B. Menai, and S. Al-Ahmadi, "A heterogeneity-aware semi-decentralized model for a lightweight intrusion detection system for IoT networks based on federated learning and BiLSTM," 2025.
[7] G. S. Rapate, A. Krishnappa, S. D. Veeresh, K. S. Kumar, and B. Kursheed, "Hybrid AI-driven anomaly detection and sequential attack classification for securing IoT networks," 2026.
[8] M. B. Musthafa, S. Senthilkumar, and A. Mruthulani, "Optimized ensemble deep learning for Raspberry Pi-based intrusion detection in IoT," 2025.
[9] V. Gladić, M. Kodyš, and J. Hrabovský, "Evaluating machine learning models for IoT intrusion detection on the RT-IoT2022 dataset," 2025.
[10] D. M. Diab, A. Shaikhanova, and V. M. Tien, "Hardware-aware machine learning and deep learning models for resource-constrained IoT devices," 2025.
[11] J. Hagen and A. Strate, "Comparative analysis of Random Forest, KNN, SVM, and XGBoost on the NF-ToN-IoT-v2 dataset," 2025.
[12] P. Kuraś et al., "Optimization of machine learning models for effective anomaly detection in industrial IoT systems," 2026.
[13] M. Khelifi, "Feature extraction methodologies for edge-based IoT intrusion detection systems," 2025.
[14] M. Farsi, M. Q. J. Al-Zaidawi, and M. Çevik, "A practical evaluation framework for Random Forest-based IoT intrusion detection with sliding window aggregation," 2025.
[15] CIC Authors, "IntruDet-LSTM: A knowledge-driven hybrid intrusion detection system for IoT cybersecurity," 2025.
[16] S. Alahmari and N. Aleisa, "Enhancing the CIC IoT Dataset 2023 for improved attack detection through GANs augmentation and federated learning," 2025.
[17] V. Ponnusamy, S. Tiwari, A. Sinha, and E. Kisić, "Machine learning-driven anomaly detection for enhanced IoT networks security," 2024.
[18] R. Ahmad, I. A. Khan, and S. U. Rehman, "A comprehensive survey of contemporary anomaly detection methods for securing smart IoT systems," 2025.
[19] V. Electronics et al., "A survey of multi-layer IoT security using SDN, blockchain, and ML," 2026.
[20] A. Gueriani, K. Kifouche, and M. S. Hossain, "An interpretable deep learning framework for intrusion detection in industrial IoT," 2025.
[21] A. Gueriani, K. Kifouche, and M. S. Hossain, "An explainable hybrid CNN-XGBoost framework incorporating SHAP values for IoT intrusion detection," 2025.
[22] L. Zhang, Y. Liu, and W. Wang, "Anomaly-based intrusion detection using statistical methods: A foundational approach," 2024.
[23] "Global IoT connections to grow 13% to 18.8 billion in 2026," Transforma Insights, 2026.
[24] "The economic impact of IoT in 2025," International Data Corporation (IDC), 2024.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details