International Journal of Engineering and Techniques - Volume 12 Issue 2, March-April - 2026

RESEARCH ARTICLE OPEN ACCESS

An Optimized Ensemble Learning Framework for Network Intrusion Detection using Random Forest & XGBoost

Winson Aravinth Raj C[1], Shenany B[2], Vaishnavi J[3], Arunadevi R[4]

[1] *Department of Computer Science and Engineering, Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: winsonaravinthraj@gmail.com

[2] *Department of Computer Science and Engineering, Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: shenanyshenu.ss@gmail.com

[3] *Department of Computer Science and Engineering, Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu 613006, India*
Email: vaishnavishankar1055@gmail.com

[4] *Department of Computer Science and Engineering, Parisutham Institute of Technology and Science, Thanjavur, Tamil Nadu – 613006, India*
Email: aruna.ap.cse.pits@gmail.com

Abstract:

This project presents a Machine Learning-based Ensemble Intrusion Detection System (IDS) designed to enhance security in IoT networks through intelligent threat detection and real-time monitoring. With the rapid growth of IoT devices, networks have become increasingly vulnerable to various cyberattacks, making efficient intrusion detection mechanisms essential. The proposed system utilizes a labelled network traffic dataset, where the data is pre-processed and used to train machine learning model capable of identifying different types of malicious activities. To improve detection performance, the system employs an ensemble learning approach that combines Bagging (Random Forest) and Boosting (XGBoost) algorithms. For real-time evaluation, the dataset records are replayed in a time-based manner to simulate continuous traffic flow. The data is streamed through Apache Kafka, enabling real-time data ingestion and processing. The incoming traffic is analyzed by the trained model, and the system determines whether the traffic is normal or malicious. A real-time monitoring dashboard displays detection results and generates alerts for suspicious activities, allowing administrators to monitor network behaviour effectively. By integrating time-based dataset streaming, Kafka-based real-time processing, and ensemble machine learning techniques, the proposed IDS improves detection accuracy, reduces false positives, and provides a scalable and reliable solution for securing modern IoT networks.

Keywords — Internet of Things (IoT), Intrusion Detection System (IDS), Ensemble Learning, Real-Time Network Security, Lightweight IDS.

I. INTRODUCTION

Intrusion detection systems (IDS) for Internet of Things (IoT) networks have matured significantly with the adoption of machine learning and deep learning techniques, achieving high detection accuracy across a wide range of attack scenarios. However, in practical deployments, accuracy alone is no longer the primary bottleneck. Modern IoT environments demand intrusion detection solutions that operate in real time, scale with increasing device counts, and remain computationally efficient under constrained network and hardware conditions.

In many existing ML-based IDS solutions, performance gains are achieved at the cost of increased model complexity, high-dimensional feature sets, and offline batch processing

pipelines. Such designs limit their applicability in real-world IoT networks, where traffic is continuous, resources are limited, and security responses must be timely [10][3]. As a result, there is a growing need for IDS frameworks that emphasize efficiency, low latency, and system-level integration rather than purely algorithmic novelty.

Tree-based ensemble learning methods such as Random Forest and XGBoost [2][9] have emerged as reliable and efficient classifiers for network intrusion detection due to their strong generalization capability, robustness to noisy data, and interpretability. While these models are widely adopted, their effectiveness in real-time IoT security depends heavily on how they are integrated within the detection pipeline. Without careful feature handling, streaming support, and deployment-aware design, even state-of-the-art classifiers may fail to meet operational requirements.

This work proposes a lightweight, real-time ensemble ML-based Intrusion Detection System tailored for IoT networks. Instead of introducing new detection algorithms, the proposed approach focuses on optimizing the detection pipeline using a hybrid ensemble of Random Forest and XGBoost, combined with embedded feature selection. Feature importance derived from impurity-based measures in Random Forest and Information Gain in XGBoost is utilized to reduce feature dimensionality during training, thereby lowering complexity and improving response time.

II. RELATED WORK

Intrusion Research on Intrusion Detection for Internet of Things (IoT) networks [10] has evolved rapidly in response to the growing scale, heterogeneity, and exposure of connected devices. Early intrusion detection systems primarily relied on predefined signatures and static rule sets, which were effective for identifying known attack patterns but lacked adaptability to emerging and previously unseen threats. As IoT traffic characteristics differ substantially from those of traditional enterprise networks [4], such limitations accelerated the transition toward learning-based detection approaches.

Machine learning-driven intrusion detection frameworks have since become widely adopted, enabling systems to infer normal and malicious behaviours directly from network traffic data. Supervised learning techniques employing classifiers such as decision trees, support vector machines, and ensemble models have demonstrated strong capability in distinguishing benign and malicious traffic within IoT environments. Among these approaches, ensemble learning has received particular attention due to its ability to improve robustness and generalization by combining multiple learners.

Tree-based ensemble methods, including Random Forest and gradient boosting variants such as XGBoost, are especially well suited for network intrusion detection. Random Forest utilizes bagging to reduce variance and achieve stable performance across diverse traffic patterns,

To enable real-time traffic analysis, the system integrates Apache Kafka as a streaming platform for continuous network data ingestion at the IoT gateway level. This design allows scalable, low-latency processing of network flows and supports immediate detection of malicious behaviours such as lateral data access and anomalous communication attempts by compromised IoT devices. Detected intrusions and system status are visualized through an interactive dashboard that provides actionable alerts and network-level insights.

The main contributions of this work are summarized as follows:

a) A practical and lightweight IDS architecture that integrates ensemble learning with real-time streaming for IoT networks. Reference is mentioned in [8][9].

b) An embedded feature selection strategy leveraging feature importance from Random Forest and XGBoost to reduce computational overhead. Refer [9].

c) A real-time monitoring and alerting framework that enhances situational awareness and enables timely mitigation of IoT intrusions.

Experimental evaluation demonstrates that the proposed system achieves competitive detection performance while significantly improving efficiency and latency, making it suitable for deployment in resource-constrained IoT environments [9].

while boosting-based models enhance sensitivity to subtle and low-frequency attack behaviours. Numerous studies report competitive detection accuracy using these models on benchmark intrusion datasets, confirming their effectiveness for security analytics in IoT contexts [9].

Deep learning-based Intrusion Detection System approaches have also been explored for IoT security, employing architectures such as convolutional and recurrent neural networks to model complex spatial and temporal traffic patterns. While these methods often demonstrate strong detection capability in controlled or offline settings, their reliance on computationally intensive architectures, large labelled datasets, and offline or semi-offline processing pipelines limits their feasibility for deployment at IoT gateways or edge nodes. In latency-sensitive and resource-constrained environments, such requirements can hinder real-time detection and timely response [1].

Another common characteristic of existing intrusion detection systems is the reliance on large, high-dimensional feature sets. Although rich feature representations can improve classification performance, they also increase computational cost and inference latency. Feature selection techniques are sometimes applied as a separate preprocessing step; however, decoupling feature reduction from model training introduces additional overhead and reduces adaptability in dynamic network conditions.

Furthermore, many proposed IDS frameworks are designed around batch-processing workflows and lack integration with real-time data streaming infrastructures. This limits their ability to process continuous traffic flows

and respond promptly to ongoing attacks. Operational aspects such as live monitoring, alert generation, and system-level visualization are also frequently underemphasized, reducing the practical usability of these systems in real-world IoT deployments [5].

In contrast, the proposed work emphasizes deployment-oriented efficiency by integrating ensemble learning with embedded feature selection and real-time data streaming. Feature relevance is inferred directly from Random Forest and XGBoost models during training, enabling dimensionality reduction without introducing separate selection stages. The integration of Apache Kafka facilitates scalable and low-latency ingestion of network traffic, while a real-time dashboard provides actionable insights and alerts. This approach bridges the gap between high-performing machine learning models and the operational requirements of practical IoT intrusion detection systems.

III. PROPOSED METHODOLOGY

The proposed Intrusion Detection System is designed as a modular and scalable real-time security framework for detecting malicious network activities using simulated traffic data. The methodology integrates traffic simulation, Kafka-based data streaming and an optimized ensemble learning approach combining Random Forest and XGBoost to enable accurate and timely intrusion detection while minimizing computational overhead.

a. *System Overview:*

The proposed Intrusion Detection System (IDS) is designed as a modular and scalable framework for detecting malicious network activities using an optimized ensemble learning approach. The system integrates traffic simulation, machine learning, and real-time processing components to ensure efficient and accurate intrusion detection.

The architecture consists of six key modules: Authentication (RBAC), Model Training, Traffic Simulation, Kafka-based Data Streaming, Detection Engine, and Alert Mechanism. Each module performs a specific function, contributing to the overall effectiveness of the system.

b. *Authentication Block (RBAC):*

The system incorporates a Role-Based Access Control (Role-Based Access Control) mechanism to ensure secure access to system functionalities. Users are assigned roles such as administrator or analyst, each with specific permissions.

This module restricts unauthorized access to sensitive operations such as model configuration, system monitoring, and alert management. By enforcing RBAC, the system enhances security and ensures controlled interaction with the IDS.

c. *Model Training Module:*

The model training module is responsible for building the machine learning models used for intrusion detection. The system utilizes an ensemble approach combining:

- Random Forest
- XGBoost

The dataset is pre-processed and split into training and testing sets. Feature selection is performed during training to eliminate irrelevant attributes. Hyperparameter tuning is applied to optimize model performance.

The trained models are stored and later used by the detection engine for classification.

d. *Traffic Simulation Module:*

Instead of capturing live network traffic, the system simulates network behaviour using benchmark datasets such as NSL-KDD or CICIDS2017.

The simulation module generates realistic traffic patterns, including both normal and malicious activities. This enables controlled experimentation and reproducibility while covering diverse attack scenarios.

The simulated traffic is treated as a continuous input stream for the detection pipeline.

e. *Kafka-Based Data Streaming:*

To emulate real-time data flow, the system uses Apache Kafka for streaming simulated traffic data.

Traffic features generated by the simulation module are published to Kafka topics. The detection engine consumes these streams asynchronously, enabling near real-time processing.

This design ensures scalability, fault tolerance, and efficient handling of high-volume data streams.

f. *Detection Engine*

The detection engine is the core component of the system, responsible for classifying incoming traffic as normal or malicious.

It applies the trained ensemble model, where:

- Random Forest provides robust baseline predictions
- XGBoost enhances detection of complex attack patterns

The outputs of both models are combined using a decision fusion strategy (e.g., weighted voting) to generate the final classification.

This hybrid approach improves detection accuracy and reduces false positives.

g. *Alert and Response*

Once an intrusion is detected, the system generates alerts to notify administrators. Alerts include information about the type of attack and its severity.

The module supports real-time monitoring and enables quick response to potential threats. It can also simulate response actions such as isolating suspicious traffic or flagging compromised entities for further analysis.

IV. SYSTEM ARCHITECTURE

The proposed intrusion detection system follows a gateway-centric architecture to enable efficient and real-time security monitoring in IoT networks. Network traffic generated by connected IoT devices is captured at the gateway level, avoiding computational overhead on individual devices.

Captured traffic is transformed into flow-based features and streamed in real time using Apache Kafka. This streaming layer enables scalable, low-latency data ingestion and decouples traffic collection from intrusion analysis.

The intrusion detection engine processes the streamed features through lightweight preprocessing and embedded feature selection, where feature relevance is derived directly from Random Forest and XGBoost models. A hybrid ensemble of Random Forest and XGBoost is then used to classify traffic as normal or malicious [2].

Detection outcomes are visualized through a real-time dashboard that displays network status and alerts. Upon identifying malicious behaviour, the system can logically block the offending device at the gateway level, preventing further attack propagation.

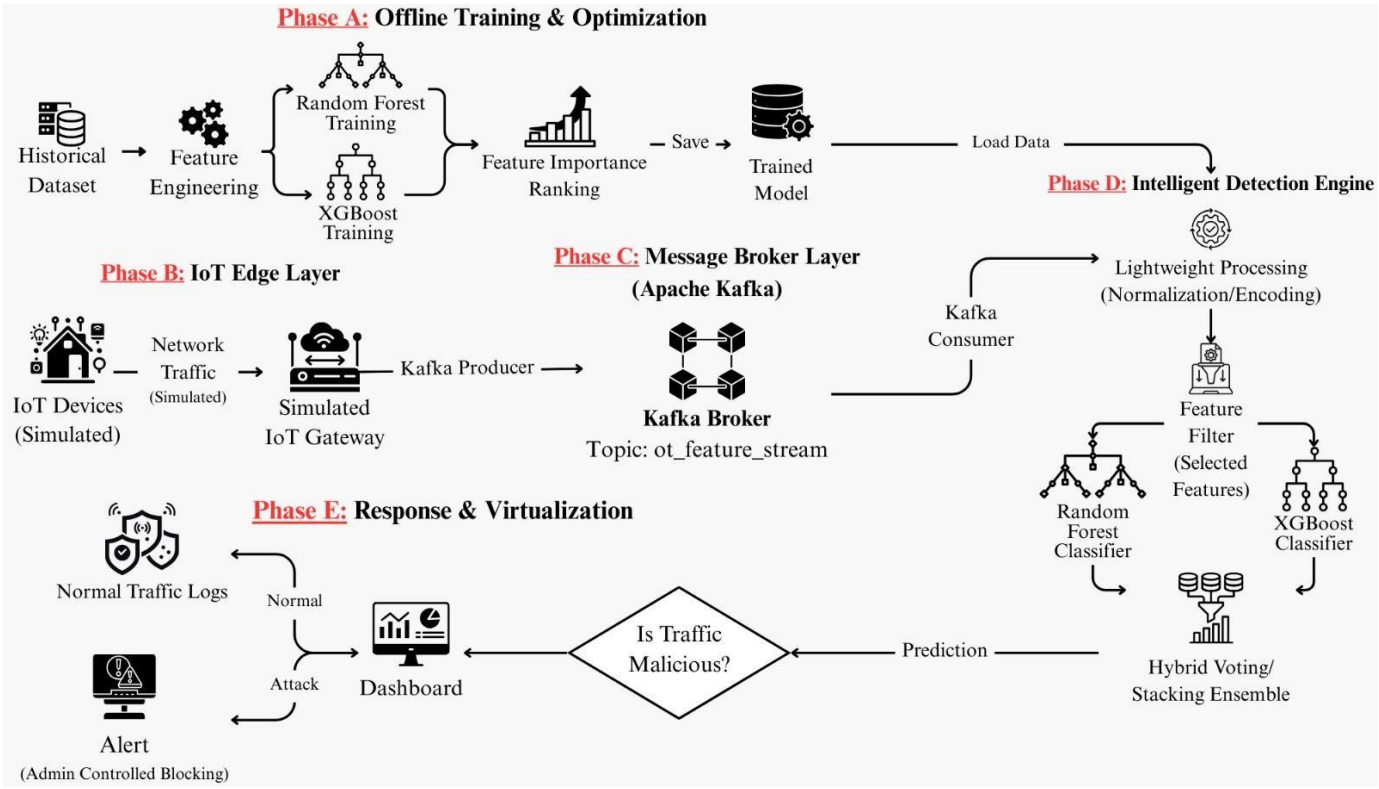

Fig 1: IDS Architecture for IoT Networks integrating Apache Kafka Streaming with Random Forest & XGBoost Ensemble Learning.

V. Experimental Setup

a. Dataset Description

The proposed intrusion detection system is evaluated using a publicly available IoT network intrusion dataset containing both benign and malicious traffic instances. The dataset includes diverse attack scenarios such as scanning, denial-of-service, and unauthorized access behaviours, along with normal IoT communication patterns. Each traffic instance is represented using flow-based network features, making the dataset suitable for gateway-level intrusion detection.

To ensure realistic evaluation, the dataset is pre-processed to handle missing values and normalize numerical features. Categorical attributes, where present, are appropriately encoded. The dataset is divided into training and testing subsets to enable supervised learning and unbiased performance assessment. There are some statistical references on the paper [6].

Dataset from sources like ToN-IoT, UNSW-NB15 are being used here since it has more accuracy than other datasets. Data can be seen in the below table:

Dataset	Accuracy	Precision	Recall	F1-score	False Positive Rate
ToN-IoT	96.6%	95.9%	94.3%	94.6%	0.5%
UNSW-NB15	94.8%	94.4%	92.1%	93.2%	0.7%
Bot-IoT	93.2%	94.4%	91.1%	95.3%	0.4%
NSL-KDD	91.5%	92.1%	93.9%	92.0%	0.8%

b. *Model Training and Configuration*

The Random Forest and XGBoost models are trained independently using the selected feature subset. Hyperparameters are chosen to balance detection performance and computational efficiency, ensuring suitability for real-time deployment. The final classification decision is obtained through ensemble-based fusion of the individual model outputs.

Dataset Composition Used for Experimental Evaluation:

Dataset Source	Traffic Type	No. of Samples
Bot-IoT	Normal	22,18,761
Bot-IoT	Fuzzers	25,246
Bot-IoT	Reconnaissance	49,702
Bot-IoT	Shellcode	2,576
Bot-IoT	Analysis	2,677
Bot-IoT	Backdoors	2,329
Bot-IoT	DoS	20,055
Bot-IoT	Exploits	48,896
Bot-IoT	Worms	174
Bot-IoT	Generic	2,15,481
Total		**25,85,897**

c. *Evaluation Metrics*

The model is trained on approximately 70% of the available data, with the remaining 30% used for testing. The dataset size is chosen to reflect realistic IoT traffic volumes while ensuring efficient real-time processing. The performance of the system is assessed using standard intrusion detection metrics, including accuracy, precision, recall, and F1-score. In addition, the false alarm rate is measured to evaluate the system's reliability in distinguishing benign traffic from malicious activity. Detection latency is also analyzed to assess real-time responsiveness under streaming conditions.

d. *Results and Discussion*

The proposed real-time ensemble-based intrusion detection system is evaluated in terms of detection performance, computational efficiency, and real-time responsiveness. The results demonstrate that the system achieves reliable intrusion detection while maintaining low processing overhead, validating its suitability for deployment in IoT environments.

e. *Detection Performance*

The hybrid ensemble of Random Forest and XGBoost achieves consistently strong classification performance across normal and malicious traffic instances. The ensemble approach improves detection robustness by combining the stable generalization capability of Random Forest with the enhanced sensitivity of XGBoost to complex and low-frequency attack patterns [9].

The system achieves high accuracy and F1-score, indicating an effective balance between precision and recall. Compared to individual classifiers, the ensemble reduces false positives, which is critical in IoT environments where excessive alerts can overwhelm administrators and reduce trust in the detection system.

f. *Real-Time Processing and Latency Analysis*

The integration of Apache Kafka enables the system to process continuous network traffic streams with low latency. Under streaming conditions, the intrusion detection engine maintains stable throughput and processes incoming traffic in near real time [12].

Detection latency remains within acceptable limits for IoT security applications, allowing timely identification of malicious activity and rapid response. The decoupled streaming architecture ensures scalability and resilience to traffic bursts, which are common in dynamic IoT networks [9].

VI. CONCLUSIONS

This paper presented a lightweight, real-time ensemble machine learning–based intrusion detection system designed for practical deployment in IoT networks. Rather than proposing new classification algorithms, the work focused on improving operational efficiency by integrating mature ensemble models with embedded feature selection and real-time data streaming. The combination of Random Forest and XGBoost enabled robust intrusion detection while maintaining low computational overhead, making the system suitable for resource-constrained IoT environments.

The proposed gateway-centric architecture allows continuous monitoring of network traffic without imposing additional burden on individual IoT devices. The integration of Apache Kafka enables scalable and low-latency processing of streaming network data, while embedded feature selection reduces feature dimensionality during training and improves inference efficiency. Experimental results demonstrate that the system achieves reliable detection performance with reduced false alarms and acceptable detection latency, validating its applicability for real-world IoT security monitoring.

In future work, the system will be evaluated across multiple IoT intrusion datasets and extended to support distributed and multi-gateway environments. Additional enhancements may include adaptive model updating to handle evolving attack patterns, incorporation of unsupervised or semi-supervised learning for unknown threat detection, and tighter integration with automated

network-level mitigation mechanisms. These extensions aim to further enhance the robustness and scalability of the proposed intrusion detection framework

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide and faculty members of the Department of Computer Science and Engineering for their continuous support, valuable guidance, and encouragement throughout this research work.

The authors also thank their external guide associated with Monzha Research Lab for providing the necessary guidance and resources to successfully complete this study.

Additionally, we acknowledge the use of publicly available datasets and tools that contributed to the development and evaluation of the proposed model.

REFERENCES

1. R. Varaprasad, A. P. Chakkaravarthy, and M. Veeresha, *"A comprehensive analysis of intrusion detection system using machine learning and deep learning algorithms*" Proc. Int. Conf. Intelligent Algorithms for Computational Intelligence Systems (IACIS), IEEE, 2024.

2. M. Ramaiah, A. Padma, R. Vishnukumar, M. Y. Rahamathulla, and V. Chithanuru, *"A hybrid wrapper technique enabled network intrusion detection system for software-defined networking based IoT networks"* Proc. IEEE Int. Conf. Artificial Intelligence for Internet of Things (AIIoT), IEEE, 2024.

3. S. K. Kodali and C. H. Muntean, *"An investigation into deep learning-based network intrusion detection system for IoT systems"* Proc. IEEE Int. Conf. Data Science and Computer Application (ICDSCA), pp. 374–379, 2021.

4. Z. Liu, K. Roy, N. Thapa, X. Yuan, A. Shaver, and S. Khorsandroo, *"Anomaly detection on IoT network intrusion using machine learning,"* IEEE, 2020.

5. N. Mahamud, M. J. Uddin, and U. Sumaiya, *"Enhancing network security using machine learning for automated anomaly-based intrusion detection systems for IoT environment"* Proc. Int. Research Conf. Smart Computing and Systems Engineering (SCSE), IEEE, 2025.

6. K. Abinaya, T. Lohith, and S. Jayanth Kumar, *"Enhancing network security with intrusion detection systems in IoT devices"* Proc. Int. Conf. Expert Clouds and Applications (ICOECA), IEEE, 2025.

7. S. S. S. Sugi and S. Raja Ratna, *"Investigation of machine learning techniques in intrusion detection system for IoT network*" Proc. Int. Conf. Intelligent Sustainable Systems (ICISS), IEEE, 2020.

8. Z. Alomari, Z. Li, and A. Makanju, *"Lightweight machine learning-based IDS for IoT environments"* Proc. IEEE Cyber Security in Networking Conf. (CSNet), IEEE, 2024.

9. B. Peng, J. Zhao, Y. Sun, and Y. Liu, *"Research and discussion on comparative prediction models based on XGBoost and random forest and clustering analysis"* in Proc. 2024 IEEE 2nd Int. Conf. on Control, Electronics and Computer Technology (ICCECT), 2024.

10. V. Jyothsna, E. Sandhya, R. Roopa, B. Deena Divya Nayomi, D. K. Shareef, and P. Bhasha, *"Intrusion Detection System for IoT networks"* in Proc. 2023 1st Int. Conf. on Optimization Techniques for Learning (ICOTL), 2023.

11. M. Patidar, A. Dave, D. Vekariya, B. Udumula, K. K. Porla, and B. Nidimandi, *"Network Intrusion Detection System using random forest"* in Proc. 2025 12th Int. Conf. on Computing for Sustainable Global Development (INDIACom), 2025.

12. K. M. Kiran Kumar, M. V. Srikar Reddy, K. Ullas, and S. M., *"Distributed Intrusion Detection System using Kafka and Spark streaming"* in Proc. 2025 Int. Conf. on Visual Analytics and Data Visualization (ICVADV), 2025.