
ML-EIDIoT: A Lightweight Machine Learning Framework for Early Intrusion Detection in Resource-Constrained IoT Networks

Samundeeswari¹, A.Ambeth Raja^{2}, V. Devi³ D.R.Ashwin Kumar⁴*

¹Research Scholar ² Associate Professor ³Professor

^{1&2&3} PG & Research Department of Computer Science

⁴Department of Computer Science (Data Science), Manipal Academy of Higher Education, Bengaluru , Thiruthangal Nadar College, Chennai.

¹rsamu165@gmail.com

²arajacs1983@gmail.com³drdevi.venkatesan@gmail.com

⁴ashwinkumardr26@gmail.com

Abstract

The rapid expansion of Internet of Things (IoT) networks has intensified security challenges due to heterogeneous device configurations, dynamic traffic behavior, and limited computational resources. Conventional intrusion detection systems often fail to provide timely and reliable protection in such environments, particularly against evolving and previously unseen attacks. To address these challenges, this paper proposes ML-EIDIoT, a lightweight machine learning framework for early detection of network intrusions in resource-constrained IoT environments. The framework integrates flow-level traffic monitoring, data preprocessing, feature optimization, and supervised machine learning classifiers to accurately distinguish between normal and malicious network activities. Experiments were conducted on the UNSW-NB15 benchmark dataset using Logistic Regression, Support Vector Machine, Random Forest, and Gradient Boosting models. Comprehensive evaluation based on accuracy, precision, recall, F1-score, AUC, and detection latency demonstrates that ensemble-based classifiers outperform linear models. Notably, the Gradient Boosting model achieved an accuracy of 95.72%, an F1-score of 94.54%, and an AUC of **0.98**, while maintaining low inference time suitable for edge-level deployment. The proposed early intrusion detection strategy further enables rapid identification of malicious behavior within limited observation windows, reducing response time and potential attack impact. The results confirm that ML-EIDIoT provides an effective balance between detection performance and computational efficiency, making it a practical solution for securing IoT networks and a strong foundation for future hybrid and deep learning-based intrusion detection systems.

Keywords: Internet of Things , Machine learning, Boosting, Intrusion detection, resources

1. Introduction

The Internet of Things (IoT) paradigm has fundamentally transformed the way digital systems interact with the physical world by enabling seamless communication among billions of interconnected devices. From smart homes and healthcare monitoring systems to

industrial automation and intelligent transportation networks, IoT technologies have become deeply embedded in critical infrastructures and everyday applications. This rapid expansion, however, has also introduced complex security challenges. Unlike traditional enterprise networks, IoT environments are characterized by large-scale device heterogeneity, dynamic network topologies, and constrained computational and energy resources [1-3]. These inherent characteristics make IoT networks particularly vulnerable to a wide spectrum of cyber intrusions, necessitating the development of security mechanisms that are both effective and lightweight.

A key concern in IoT intrusion detection is achieving an optimal balance between detection accuracy and resource efficiency. This necessity has driven recent research toward the development of hybrid and optimized machine learning solutions tailored specifically for IoT networks [4-6].

The contributions of this work are threefold. First, it presents a systematic analysis of machine learning-based intrusion detection techniques in the context of IoT networks, highlighting their strengths and limitations under resource constraints. Second, it introduces an efficient detection framework that integrates lightweight models with carefully selected features to achieve reliable early intrusion detection. Third, it provides an extensive experimental evaluation that demonstrates the effectiveness of the proposed approach compared to baseline classifiers. Through these contributions, the study establishes a solid foundation for subsequent research on advanced deep learning and hybrid intrusion detection systems for IoT security.

Proposed work

The proposed work is motivated by the growing need for intrusion detection mechanisms that are both accurate and computationally efficient for IoT environments. While existing deep learning-based intrusion detection systems demonstrate high detection capability, their deployment in real-world IoT networks is often limited by excessive computational overhead, memory consumption, and latency. This work aims to bridge this gap by designing a machine learning-driven intrusion detection framework that emphasizes early-stage attack identification, reduced feature dimensionality, and lightweight classification models suitable for deployment at IoT gateways or edge nodes.

Methodology

The proposed ML-EIDIoT framework follows a structured multi-stage methodology designed to ensure robustness, scalability, and computational efficiency for intrusion detection in resource-constrained IoT environments. The methodology integrates data preprocessing, feature optimization, supervised learning, and early intrusion detection to enable accurate and timely identification of malicious network activities.

Data Preprocessing

Let the raw network traffic dataset be represented as $D = \{(x_i, y_i)\}_{i=1}^N$,

where $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ denotes the feature vector of the i^{th} network flow, d is the total number of features, and $y_i \in \{0,1\}$ represents the class label corresponding to normal or malicious traffic.

Duplicate records are removed to avoid bias during model training, and missing values are handled using statistical imputation. Numerical attributes are normalized using min–max scaling to ensure uniform feature ranges and improve classifier convergence. The normalized feature x_{ij}' is computed as: $x_{ij}' = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}$ where $\min(x_j)$ and $\max(x_j)$ represent the minimum and maximum values of the j^{th} feature across the dataset.

Categorical features are transformed into numerical representations using label encoding, ensuring compatibility with machine learning algorithms. To mitigate class imbalance, controlled sampling techniques are applied such that the class distribution satisfies: $|D_{normal}| \approx |D_{attack}|$ thereby reducing classifier bias toward majority classes and improving detection sensitivity.

Model Training and Classification

The optimized feature set X^* is used to train multiple supervised machine learning classifiers. Each model learns a mapping function: $f: X^* \rightarrow y$, where y represents the predicted class label.

Logistic Regression (LR) models the posterior probability of an intrusion event as: $P(y = 1 | x) = \frac{1}{1 + e^{-(w^T x + b)}}$ where w and b denote the weight vector and bias term, respectively.

Support Vector Machine (SVM) aims to find an optimal hyperplane by solving: $\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$ subject to $y_i(w^T x_i + b) \geq 1 - \xi_i$, where C is the regularization parameter and ξ_i represents slack variables.

Random Forest (RF) constructs an ensemble of decision trees $\{T_m\}_{m=1}^M$, and the final prediction is obtained through majority voting: $\hat{y} = \text{mode}\{T_1(x), T_2(x), \dots, T_M(x)\}$

Gradient Boosting (GB) builds an additive model by minimizing a loss function $L(y, F(x))$ iteratively: $F_m(x) = F_{m-1}(x) + \eta h_m(x)$ where $h_m(x)$ is the weak learner at iteration m and η is the learning rate.

These models are selected to balance detection accuracy and computational efficiency, with ensemble methods providing improved robustness against noisy traffic patterns. These models are selected to balance detection accuracy and computational efficiency, with ensemble methods providing improved robustness against noisy traffic patterns.

Early Intrusion Detection Strategy

To enable early detection, network traffic is analyzed within sliding observation windows of length w . Let $W_t = \{x_{t-w+1}, \dots, x_t\}$ denote the traffic window at time t . An intrusion alert is generated if the predicted probability of malicious activity exceeds a predefined threshold θ : $\frac{1}{w} \sum_{i \in W_t} P(y_i = 1 | x_i) \geq \theta$. This strategy allows the framework to identify suspicious behavior at an early stage, reducing detection latency and preventing prolonged attack execution. By limiting the observation window, the framework minimizes computational overhead while maintaining timely and accurate intrusion detection.

Proposed Architecture

The ML-EIDIoT framework adopts a modular and layered architecture designed to ensure efficient intrusion detection while accommodating the computational constraints inherent in IoT environments. Each architectural component is responsible for a specific function,

enabling scalability, flexibility, and ease of deployment across heterogeneous IoT infrastructures. The overall data flow proceeds sequentially from traffic acquisition to decision generation, as illustrated in the proposed framework.

Decision and Alert Layer

The final Decision and Alert Layer aggregates classification outcomes and generates intrusion alerts. An alert is triggered when the predicted probability of malicious activity exceeds a predefined threshold: $Alert = \begin{cases} 1, & \text{if } P(y = 1 | x^*) \geq \theta \\ 0 & \text{otherwise} \end{cases}$

where θ denotes the decision threshold. The ML-EIDIoT architecture provides a practical and extensible foundation for early intrusion detection in IoT environments while maintaining a balance between security effectiveness and resource efficiency

Algorithm 1: ML-EIDIoT – Early Intrusion Detection in IoT Networks

Input:

Network traffic flows T , feature set X , class labels y , observation window size w , detection threshold θ

Output:

Intrusion decision $\hat{y}$ and alert generation

Step 1: Initialize traffic monitoring at IoT devices, gateways, or edge nodes.

Step 2: Capture network traffic flows $T \leftarrow \{f_1, f_2, \dots, f_n\}$

Step 3: For each traffic flow $f_i \in T$, perform preprocessing:

3.1 Remove duplicate and inconsistent records

3.2 Handle missing values

3.3 Normalize numerical attributes using min–max scaling $x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}$

3.4 Encode categorical features

Step 4: Construct preprocessed feature matrix $X \in R^{N \times d}$

Step 5: Perform feature optimization

5.1 Compute feature relevance scores S_j

5.2 Select optimized feature subset $X^* = \{x_j | S_j \geq \tau\}$

Step 6: Train supervised machine learning classifiers using X^* and y :

6.1 Logistic Regression

6.2 Support Vector Machine

6.3 Random Forest

6.4 Gradient Boosting

Step 7: For incoming traffic at time t , form observation window $W_t = \{x_{t-w+1}^*, \dots, x_t^*\}$

Step 8: Compute intrusion probability for each instance in W_t : $P(y = 1 | x_i^*) = f(x_i^*)$

Step 9: Generate intrusion alert if $\frac{1}{w} \sum_{i \in W_t} P(y_i = 1 | x_i^*) \geq \theta$

Step 10: Output classification decision and trigger alert for detected intrusion.

End Algorithm

Results

Overall Classification Performance

The overall intrusion detection performance of the evaluated machine learning models is summarized in Table 1

Table 1: Overall Classification Performance of ML-EIDIoT Framework

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC	Detection Time (ms)
Logistic Regression	88.42	86.17	87.05	86.61	0.91	0.41
Support Vector Machine	90.36	89.22	88.94	89.08	0.93	0.78
Random Forest	94.18	93.64	92.91	93.27	0.97	0.63
Gradient Boosting	95.72	94.88	94.21	94.54	0.98	0.71

The Gradient Boosting classifier achieves the highest accuracy and AUC, confirming its effectiveness in capturing complex attack patterns while maintaining acceptable detection latency.

Early Intrusion Detection Analysis

To evaluate early detection capability, intrusion probability aggregation was analyzed over limited observation windows. Table 2 shows detection performance at different window sizes.

Table 2: Early Detection Performance at Different Observation Windows

Window Size	Accuracy (%)	Recall (%)	Detection Delay (ms)
5 flows	90.88	89.74	2.3
10 flows	93.41	92.62	4.1
20 flows	95.72	94.21	7.8

Smaller windows enable faster detection, while slightly larger windows improve accuracy. This trade-off allows flexible configuration based on deployment requirements.

Conclusion

This paper presented ML-EIDIoT, a lightweight and efficient machine learning framework for the early detection of network intrusions in resource-constrained IoT environments. The proposed framework was designed to address the limitations of traditional and deep learning-based intrusion detection systems by balancing detection accuracy with computational efficiency, a critical requirement for practical IoT deployments. In particular, the Gradient Boosting model achieved an accuracy of 95.72%, an F1-score of 94.54%, and an AUC of 0.98, while maintaining low inference latency suitable for edge-level execution. The results further confirmed that feature optimization significantly reduced detection time and computational overhead without compromising classification performance.

References

- [1]. Alsharaiah M, Abualhaj M, Baniata L, Al-Saaidah A, Kharma Q, Al-Zyoued M (2024) An innovative network intrusion detection system (NIDS): hierarchical deep learning model based on the UNSW-NB15. *Int J Data Netw Sci* **8**(2):709–722
- [2]. Li X, Zhang Y, Wang H, Chen J (2024) An adaptive joint optimization framework for pruning and quantization. *Int J Mach Learn Cybern* **15**(11):5199–5215
- [3]. Gou X, Li Y, Zhang J, Wang S (2022) A sketch framework for approximate data stream processing in sliding windows. *IEEE Trans Knowl Data Eng* **35**(5):4411–4424
- [4]. Wu W, Yang P, Zhang W, Zhou C, Shen XS (2023) Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning. *IEEE Trans Ind Inform* **17**(7):4988–4998. <https://doi.org/10.1109/TII.2020.3046128>
- [5]. Lazzarini R, Tianfield H, Charissis V (2023) Federated learning for IoT intrusion detection. *AI* **4**(3):509–530
- [6]. Hua H, Li Y, Wang T, Dong N, Li W, Cao J (2023) Edge computing with artificial intelligence: a machine learning perspective. *ACM Comput Surv* **55**(9):1–35

Biographies



Ms. R. Samundeeswari is an Assistant Professor in the Department of B.Sc. Information Technology at Guru Nanak College, Chennai, with over 20 years of teaching and academic administration experience. She holds a B.Sc. in Computer Science, MCA, MBA (Systems Management), and M.Phil. in Computer Science, and is currently pursuing a Ph.D. in Computer Science at the University of Madras. Her research focuses on Machine Learning-Based Prediction and Detection of Network Intrusions in IoT Environments. Her research interests include Machine Learning, Artificial Intelligence, Cybersecurity, Internet of Things (IoT), Data Analytics, and Educational Technology. She has published research papers in national and international journals and conferences.



Dr. A. Ambeth Raja, M.Sc., M.Phil., Ph.D., is an Associate Professor and Head of the PG & Research Department of Computer Science at Thiruthangal Nadar College with over 18 years of teaching experience. He specializes in Design and Analysis of Algorithms and Data Mining Applications and has presented and published more than 25 papers, including an international presentation in Malaysia. He has served as a resource person, reviewer, and editorial board member for several journals and conferences. His achievements have earned him notable honours such as the International Distinguished Researcher Award 2020 and the Young Scholar Award 2020.