# SecureEdge-TDM: A Threat-to-Defense Mapping Framework for Secure Edge Computing Using Machine Learning-Based IoT Intrusion Detection

*Note: Sub-titles are not captured in Xplore and should not be used


1st Vikas Shukla
*Amity Institute of Information Technology*
*Amity University*
Noida, Uttar Pradesh, India
vikas.shukla8527@gmail.com

2nd Rekha Agarwal
*Amity Institute of Information Technology*
*Amity University*
Noida, Uttar Pradesh, India
ragarwal@amity.edu

3rd Rajesh Kumar Tyagi
*Computer Science*
*Amity University*
Gurgaon, Haryana, India
rktyagi@ggn.amity.edu



***Abstract*—Edge computing enables low-latency IoT services but also increases exposure to threats such as DDoS, DoS, Mirai, spoofing, reconnaissance, brute-force, malware, and web-based attacks. Most existing intrusion detection studies focus mainly on attack classification and do not link detected threats with suitable defense actions. To address this gap, this paper proposes SecureEdge-TDM, a threat-to-defense mapping framework for secure edge computing. The framework evaluates Decision Tree, Random Forest, Extra Trees, XGBoost, and LightGBM on the CICIoT2023 dataset and maps detected threats to countermeasures such as secure offloading, rate limiting, federated threat intelligence, trusted device verification, policy-as-code enforcement, secure aggregation, and blockchain-based audit logging. Experimental results show that Random Forest achieved 99.628% accuracy and 96.098% macro-F1 for binary detection, while XGBoost achieved 99.499% accuracy and 87.819% macro-F1 for multiclass threat detection. The results show that SecureEdge-TDM supports both effective threat detection and actionable defense recommendation.**



***Index Terms*—Edge computing security, IoT intrusion detection, CICIoT2023, machine learning, federated threat intelligence, trusted device verification.**


## I. Introduction

Edge computing supports low-latency and resource-aware IoT applications by processing data closer to devices and users. It reduces delay, bandwidth use, and cloud dependency, making it useful for smart cities, industrial IoT, healthcare, autonomous systems, and surveillance. However, its distributed and heterogeneous nature increases the attack surface, exposing edge nodes to threats such as DDoS, DoS, spoofing, reconnaissance, brute-force attacks, botnets, malicious offloading, poisoning, and privacy leakage.

Recent studies have proposed several IDS solutions for IoT and edge environments. The CICIoT2023 dataset by Neto et al. [1] is widely used for large-scale IoT attack evaluation. Existing works include ensemble IDS [2], ML-based smart-city IDS using CICIoT2023 [3], clustering-based IDS [4], optimized gradient boosting [5], [6], fine-tuned XGBoost for DDoS detection [7], lightweight LightGBM-based IDS [8], feature selection-based IDS [9], [10], deep IDS for IIoT [11], and lightweight ML deployment on smart-home IoT devices [12].

Federated learning has also been explored for distributed IDS because it enables collaborative training without sharing raw data [13]. However, federated IDS remains vulnerable to poisoning, dishonest clients, unreliable aggregation, and privacy leakage. Recent studies address these issues using poisoning-resistant federated learning [14], [15], secure aggregation [16], authenticated federated IDS [17], zero-trust IDS [18], distributed IDS calibration [19], and trust-centric federated edge learning [20]. Despite this progress, most works focus on detection accuracy and rarely connect detected threats with practical defense actions such as secure offloading, trusted verification, access control, secure aggregation, threat intelligence sharing, or audit logging. These challenges motivate the proposed SecureEdge-TDM framework. Instead of developing only another IDS model, this work provides a review-driven and implementation-supported threat-to-defense mapping framework. It evaluates standard and ensemble machine learning models on the CICIoT2023 dataset for binary and multiclass threat detection, and then maps detected threats to suitable countermeasures such as secure offloading, federated threat intelligence, trusted device verification, policy-as-code enforcement, secure aggregation, and blockchain-based audit logging. The main contributions of this paper are as follows:

- A structured review of recent edge and IoT security studies is presented, covering ML-based IDS, lightweight IDS, federated IDS, poisoning-resistant learning, secure aggregation, and zero-trust security.
- A threat-to-defense mapping taxonomy is developed to connect major edge threats with suitable countermeasures.
- A framework supported by implementation, named


Identify applicable funding agency here. If none, delete this.

SecureEdge-TDM, is built using the CICIoT2023 dataset for both binary and multiclass edge/IoT threat detection.
- In the proposed framework, defense strategies are associated with detections such as secure offloading, federated threat intelligence, trusted devices, policy-as-code, secure aggregation, and blockchain-based auditing.

The rest of this paper is organized as follows. Section II provides an overview of existing works on edge/IoT security and intrusion detection. Section III introduces our novel SecureEdge-TDM approach. Section IV discusses the experimental methodology, and Section V elaborates on the results and defense strategies. The paper is concluded in Section VI.

## II. Related Work

Existing studies mainly focus on intrusion detection, lightweight threat classification, federated security intelligence, poisoning-resistant learning, secure aggregation, and trust-aware defense. This section reviews the most relevant studies and identifies the gap addressed by the proposed SecureEdge-TDM framework.

### A. CICIoT2023-Based Intrusion Detection

Reliable datasets are essential for evaluating IDS models in IoT and edge environments. Neto et al. [1] introduced the CICIoT2023 dataset as a large-scale benchmark containing diverse IoT attack traffic, including DDoS, DoS, Mirai, spoofing, reconnaissance, brute-force, and web-based attacks. Several studies have used this dataset for ML-based intrusion detection. Gheni and Al-Yaseen [4] proposed a two-step clustering approach to improve IDS performance, while Houichi et al. [3] applied machine learning methods to the UNB CICIoT2023 dataset for smart-city security. Almahaqeri et al. [5] proposed an optimized gradient boosting framework and provided a detailed evaluation on CICIoT2023. These studies show the usefulness of CICIoT2023 for attack classification; however, they mainly focus on detection accuracy and do not extend the output toward actionable defense recommendation.

### B. Machine Learning, Ensemble, and Lightweight IDS

Machine learning and ensemble models are widely used for IoT intrusion detection because they can identify attack patterns from network traffic. Emanet et al. [2] and Hammood and Sadiq [21] explored ensemble-based IDS, while Adewole et al. [22] emphasized explainable IDS using rule induction. Tree-based and boosting approaches have also shown strong performance, including gradient boosting [6], fine-tuned XGBoost for DDoS detection [7], and Top-K feature selection using XGBoost, LightGBM, and Random Forest [10].

Lightweight IDS is equally important for resource-constrained edge and IoT devices. LightGBM-based and feature-optimized IDS models have been proposed for real-time and industrial IoT security [8], [9], [23], while Javed et al. [12] implemented lightweight ML-based IDS on smart-home IoT devices. However, these studies mainly focus on detection efficiency and rarely map detected attacks to suitable defense actions.

### C. Deep Learning and Federated IDS

Deep learning and federated learning have also been used for IoT/edge IDS. Soliman et al. [11] explored deep learning for industrial IoT security, but such models can be costly for resource-constrained edge devices. Federated IDS enables collaborative learning without sharing raw data [13]; however, it remains vulnerable to poisoning, dishonest clients, unreliable aggregation, and privacy leakage. To address these issues, recent studies have explored poisoning-resistant federated learning [14], [15], secure aggregation [16], authenticated federated IDS [17], zero-trust IDS [18], calibration of distributed IDS [19], and trust-centric federated edge learning [20]. Although these studies improve distributed threat detection, they do not provide a unified mapping between detected threats and suitable defense actions.

Table I shows that most existing works focus on IDS accuracy, lightweight deployment, feature reduction, or federated detection. However, they rarely connect attack identification with practical response. The proposed SecureEdge-TDM framework addresses this gap by integrating ML-based threat detection with a threat-to-defense mapping layer for actionable edge security.

## III. System Model and Problem Formulation

This work considers an edge-enabled IoT security environment consisting of IoT devices, edge nodes, and a security decision layer. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ denote the set of IoT devices that generate network traffic, and let $\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$ denote the edge nodes responsible for traffic monitoring and security analysis. Each traffic instance is represented as a feature vector

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}] \in \mathbb{R}^d, \tag{1}$$

where $d$ is the number of extracted network-flow features. The corresponding threat label is denoted as

$$y_i \in \mathcal{Y}. \tag{2}$$

In this work, the operational threat space is defined as

$$\begin{aligned}\mathcal{Y} = \{&\text{Benign}, \text{DDoS}, \text{DoS}, \text{Mirai}, \text{Spoofing},\\ &\text{Recon}, \text{BruteForce}, \text{Web-Based}, \text{Malware},\\ &\text{Other-Attacks}\}.\end{aligned} \tag{3}$$

Given a dataset $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the objective of the detection module is to learn a classifier

$$f_\theta : \mathbb{R}^d \to \mathcal{Y}, \tag{4}$$

where $\theta$ denotes the model parameters. For each input traffic sample $\mathbf{x}_i$, the predicted threat category is obtained as

$$\hat{y}_i = f_\theta(\mathbf{x}_i). \tag{5}$$

TABLE I
COMPARATIVE ANALYSIS OF RELEVANT STUDIES ON EDGE/IOT INTRUSION DETECTION AND SECURITY DEFENSE

| Study | Security Area | Dataset / Context | Method / Model | Strength | Limitation |
|---|---|---|---|---|---|
| Gheni and Al-Yaseen [4] | IoT IDS | CICIoT2023 | Two-step data clustering for IDS | Better preprocessing | No defense mapping |
| Houichi et al. [3] | Smart-city IoT security | UNB CICIoT2023 | ML-based IDS | Smart-city focus | Classification only |
| Almahaqeri et al. [5] | IoT IDS | CICIoT2023 | Optimized gradient boosting framework | Strong boosting evaluation | No defense layer |
| Adewole et al. [22] | Explainable IoT IDS | IoT environment | Rule induction-based IDS explanation | Explainable decisions | Limited response mapping |
| Kouassi et al. [10] | IoT IDS feature selection | IoT IDS context | Top-K feature selection using XGBoost, LightGBM, and RF | Reduces features | No response selection |
| Bhutta et al. [8] | Lightweight real-time IDS | WiFi-based intrusion environment | LightGBM-based IDS | Lightweight detection | Detection only |
| Zhang et al. [9] | Lightweight IoT IDS | IoT intrusion detection | Two-stage feature selection and Bayesian optimization | Efficient tuning | No defense mapping |
| Rashid et al. [13] | Federated IIoT IDS | Industrial IoT networks | Federated learning-based IDS | Preserves data privacy | Poisoning risk |
| Yang et al. [14] | Poisoning-resistant federated IDS | IoT IDS | Dependable federated learning | Handles poisoning | No full TDM |
| Javeed et al. [18] | Zero-trust IoT IDS | IoT environment | Federated learning-based zero-trust IDS | Zero-trust integration | No defense mapping |

The proposed framework further maps each detected threat to suitable defense actions. Let $\mathcal{A}$ denote the set of available defense mechanisms:

$$\begin{aligned}\mathcal{A} = \{&\text{Secure Offloading}, \text{Federated Threat Intelligence},\\ &\text{Trusted Device Verification}, \text{Policy-as-Code Enforcement},\\ &\text{Secure Aggregation}, \text{Blockchain Audit Logging},\\ &\text{Rate Limiting}, \text{Device Quarantine}\}.\end{aligned} \tag{6}$$

The threat-to-defense mapping function is defined as

$$g : \mathcal{Y} \rightarrow 2^{\mathcal{A}}, \tag{7}$$

where $2^{\mathcal{A}}$ represents the set of possible defense combinations. For a detected threat $\hat{y}_i$, the recommended defense set is

$$\hat{\mathcal{A}}_i = g(\hat{y}_i). \tag{8}$$

Thus, the overall problem is to detect the threat category accurately and associate it with an actionable defense response:

$$\mathbf{x}_i \xrightarrow{f_\theta} \hat{y}_i \xrightarrow{g} \hat{\mathcal{A}}_i. \tag{9}$$

The detection performance is evaluated using accuracy, balanced accuracy, macro-F1, weighted-F1, precision, and recall. Since edge/IoT intrusion datasets are often imbalanced, macro-F1 and balanced accuracy are used to assess performance across both majority and minority attack classes. Defense coverage is defined as

$$\mathcal{C}(g) = \frac{|\{y \in \mathcal{Y} : g(y) \neq \emptyset\}|}{|\mathcal{Y}|}. \tag{10}$$

A higher value of $\mathcal{C}(g)$ indicates that more detected threat categories are linked with suitable defense mechanisms.

## IV. PROPOSED METHODOLOGY

The proposed SecureEdge-TDM framework provides both threat detection and defense recommendation for secure edge computing. Unlike conventional IDS models that mainly classify traffic, it maps each detected edge/IoT threat to suitable countermeasures. The methodology includes five stages: preprocessing, operational threat categorization, threat detection, best-detector selection, and threat-to-defense mapping. The dataset is divided into training, validation, and testing sets, where each traffic sample is represented as $(\mathbf{x}_i, y_i)$. Missing and infinite values are handled through imputation and replacement, followed by feature normalization. Let $\mu_j$ and $\sigma_j$ denote the mean and standard deviation of the $j$-th feature. The normalized value of feature $x_{ij}$ is computed as

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}. \tag{11}$$

Second, raw attack labels are converted into operational threat categories. This step is important because datasets such as CICIoT2023 contain several fine-grained attack labels, which may be difficult to directly use for defense decision-making. Therefore, similar attacks are grouped into higher-level threat categories such as DDoS, DoS, Mirai, spoofing, reconnaissance, brute-force, web-based attacks, malware, and other attacks. This produces a more practical label space for edge security management. Third, multiple machine learning models are trained and evaluated for threat detection. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_K\}$ be the set of candidate classifiers, including Decision Tree, Random Forest, Extra Trees, KNN, AdaBoost, Gradient Boosting, HistGradientBoosting, XGBoost, LightGBM, and CatBoost. Each classifier $f_k$ is trained on the training set and evaluated on the validation set. The best detector is selected using validation macro-F1 because macro-F1 gives equal importance to all classes and is therefore more suitable for imbalanced intrusion detection datasets. The selected detector is defined as

$$f^* = \arg\max_{f_k \in \mathcal{F}} \text{Macro-F1}_{val}(f_k). \tag{12}$$

After selecting the best detector, final evaluation is performed on the independent test set. For each test sample $\mathbf{x}_i$, the selected detector predicts the threat category:

$$\hat{y}_i = f^*(\mathbf{x}_i). \tag{13}$$

The predicted threat category is then passed to the threat-to-defense mapping layer. This layer recommends a set of suitable defense mechanisms:

$$\hat{\mathcal{A}}_i = g(\hat{y}_i). \tag{14}$$

The rule-based mapping layer links each detected threat to suitable defenses: DDoS/DoS to secure offloading and rate limiting, spoofing/botnets to trusted verification and audit logging, poisoning to secure aggregation and federated threat intelligence, and web/unauthorized access attacks to policy-as-code enforcement and zero-trust access control.

The proposed framework can therefore be viewed as an integrated function:

$$\Phi(\mathbf{x}_i) = g(f^*(\mathbf{x}_i)), \tag{15}$$

where $f^*(\cdot)$ performs threat detection and $g(\cdot)$ performs defense recommendation. This integration makes SecureEdge-TDM more actionable than a standalone IDS because the framework outputs both the detected threat and the corresponding defense actions.

**Algorithm 1** SecureEdge-TDM: Threat Detection and Defense Mapping

**Input:** Dataset $\mathcal{S}$, classifiers $\mathcal{F}$, mapping function $g(\cdot)$
**Output:** Predicted threats $\hat{y}_i$ and defense actions $\hat{\mathcal{A}}_i$

1) Split $\mathcal{S}$ into training, validation, and testing sets.
2) Preprocess data by handling missing/infinite values and normalizing features.
3) Map raw labels into operational threat categories:

$$\begin{aligned}\mathcal{Y} = \{&\text{Benign}, \text{DDoS}, \text{DoS}, \text{Mirai},\\ &\text{Spoofing}, \text{Recon}, \text{BruteForce},\\ &\text{Web-Based}, \text{Malware}, \text{Other-Attacks}\}.\end{aligned}$$

4) Train each classifier $f_k \in \mathcal{F}$ and compute validation macro-F1.
5) Select the best detector:

$$f^* = \arg\max_{f_k \in \mathcal{F}} \text{Macro-F1}_{val}(f_k).$$

6) For each test sample $\mathbf{x}_i$, predict the threat and map it to defenses:

$$\hat{y}_i = f^*(\mathbf{x}_i), \qquad \hat{\mathcal{A}}_i = g(\hat{y}_i).$$

7) Evaluate detection performance and defense coverage:

$$\mathcal{C}(g) = \frac{|\{y \in \mathcal{Y} : g(y) \neq \emptyset\}|}{|\mathcal{Y}|}.$$

The proposed methodology fulfils the objective by combining dataset-based threat detection with a review-driven mapping layer that links detected edge threats to practical defenses such as secure offloading, federated threat intelligence, trusted verification, policy-as-code enforcement, secure aggregation, and blockchain audit logging.

## V. Experimental Setup

The experiments were conducted on the CICIoT2023 dataset by using predetermined train, validation, and test data. All records included network flow-based features and a class label that was assigned threat types including Benign, DDoS, DoS, Mirai, Spoofing, Recon, BruteForce, Web-Based, Malware, and Other-Attacks. Infrequent classes, i.e., those with less than 500 training instances, were grouped in Other-Attacks category, whereas missing/infinite values were preprocessed by median imputation, replacement, and standardization.

Five classifiers were compared including Decision Tree, Random Forest, Extra Trees, XGBoost, and LightGBM. In both Random Forest and Extra Trees classifier, 100 estimators were used, and in the latter two classifiers, 120 estimators were used having a learning rate of 0.08. To mitigate the effect of class imbalance, balanced classes/samples were employed. For selecting the best detector, validation F1-score was considered, which was independently tested for binary and multi-class detection. Performance evaluation was done based on accuracy, balanced accuracy, macro-F1, weighted-F1, training time, and testing time.

## VI. Results and Discussion

The performance of the introduced SecureEdge-TDM approach was analyzed with regard to two types of threat identification, namely, attack and multiclass operational threat identification. The analysis used 200,000 training samples, validation samples, and test samples, respectively. There were five different machine learning approaches that were considered, which include decision tree classifier, random forest classifier, extra trees classifier, XGBoost classifier, and light gradient boosting machine (LightGBM).

### A. Binary Attack Detection Results

Table II presents the binary detection results. Random Forest achieved the best overall performance with an accuracy of 99.628%, balanced accuracy of 98.351%, macro-F1 of 96.098%, and weighted-F1 of 99.637%. Although XGBoost achieved the highest balanced accuracy of 99.449%, Random Forest provided the best macro-F1 and weighted-F1, making it the selected detector for binary attack detection. The binary confusion matrix in Fig. 1 shows that the model correctly classifies most attack and benign samples. Specifically, the benign class obtained a recall of 97.013% and an F1-score of 92.386%, indicating that the model is effective even under class imbalance.

TABLE II
Binary attack detection performance

| Model | Acc. | Bal. Acc. | Macro-F1 | W-F1 |
|---|---|---|---|---|
| Decision Tree | 0.9957 | 0.9474 | 0.9522 | 0.9957 |
| Random Forest | **0.9963** | 0.9835 | **0.9610** | **0.9964** |
| Extra Trees | 0.9943 | 0.9912 | 0.9435 | 0.9946 |
| XGBoost | 0.9950 | **0.9945** | 0.9497 | 0.9952 |
| LightGBM | 0.9937 | 0.9908 | 0.9380 | 0.9940 |
| **SecureEdge-TDM** | **0.9963** | 0.9835 | **0.9610** | **0.9964** |

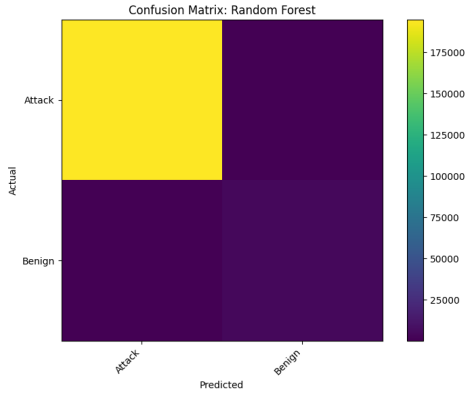


Fig. 1. Confusion matrix for binary attack detection using Random Forest.

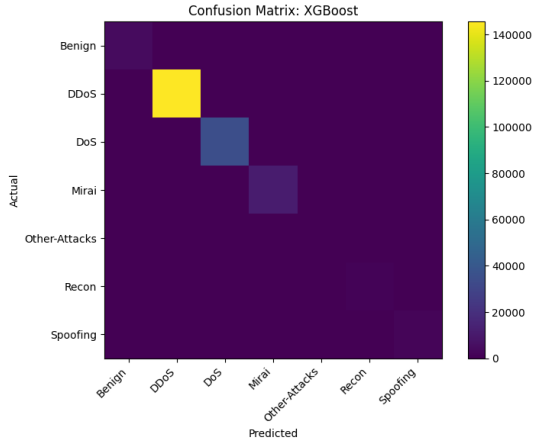


Fig. 2. Confusion matrix for multiclass operational threat detection using XGBoost.

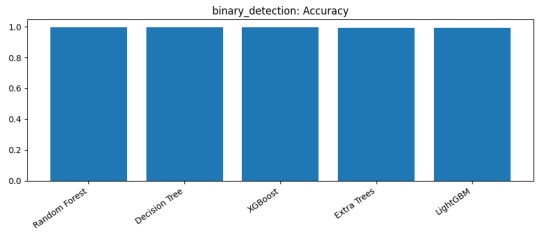


Fig. 3. Binary detection comparison based on accuracy.

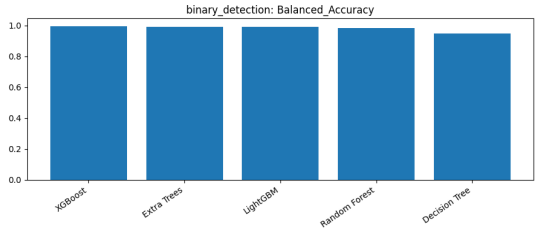


Fig. 4. Binary detection comparison based on balanced accuracy.

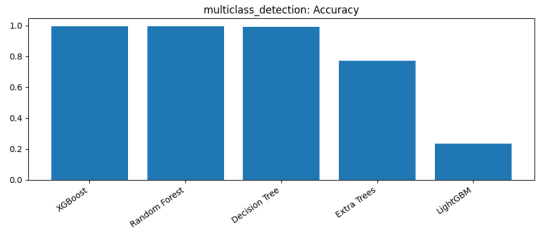


Fig. 5. Multiclass threat detection comparison based on accuracy.

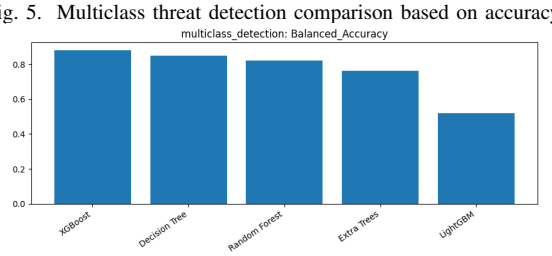


Fig. 6. Multiclass threat detection comparison based on balanced accuracy.

### *B. Multiclass Operational Threat Detection Results*

Table III reports the multiclass threat detection results. XGBoost achieved the best performance with an accuracy of 99.499%, balanced accuracy of 88.021%, macro-F1 of 87.819%, and weighted-F1 of 99.502%. This result is important because macro-F1 and balanced accuracy are more reliable than accuracy for imbalanced intrusion detection datasets. Decision Tree also performed well with a macro-F1 of 85.391%, while Random Forest achieved a higher precision but lower recall across minority classes. LightGBM performed poorly in this setting, suggesting that its current configuration is not suitable for the selected multiclass setup.

TABLE III
MULTICLASS OPERATIONAL THREAT DETECTION PERFORMANCE

| Model | Acc. | Bal. Acc. | Macro-F1 | W-F1 |
|---|---|---|---|---|
| Decision Tree | 0.9927 | 0.8494 | 0.8539 | 0.9927 |
| Random Forest | 0.9939 | 0.8209 | 0.8461 | 0.9937 |
| Extra Trees | 0.7735 | 0.7629 | 0.7622 | 0.7957 |
| XGBoost | **0.9950** | **0.8802** | **0.8782** | **0.9950** |
| LightGBM | 0.2359 | 0.5180 | 0.3333 | 0.3108 |
| **SecureEdge-TDM** | **0.9950** | **0.8802** | **0.8782** | **0.9950** |

The class-wise results show that XGBoost detects major attacks very effectively, with F1-scores of 0.9998, 0.9994, and 0.9998 for DDoS, DoS, and Mirai, respectively. Benign, Recon, and Spoofing also achieve good F1-scores of 0.9305, 0.8348, and 0.8590, while Other-Attacks remains comparatively difficult due to its small and heterogeneous sample distribution.

Fig. 2 shows the multiclass confusion matrix, where the strong diagonal pattern confirms accurate classification across most threat categories. The metric comparisons in Figs. 3–4 and Figs. 5–6 further show that Random Forest is most suitable for binary detection, while XGBoost performs best for multiclass threat detection. SecureEdge-TDM then converts these outputs into actionable defenses, mapping DDoS to

secure offloading and rate limiting, Mirai to federated threat intelligence and trusted verification, and spoofing to device identity validation and trusted verification.

## VII. Conclusion

This paper proposed SecureEdge-TDM, a threat-to-defense mapping framework for secure edge computing that links detected edge/IoT threats with suitable actions such as secure offloading, rate limiting, federated threat intelligence, trusted verification, policy-as-code enforcement, secure aggregation, and blockchain audit logging. The results confirm that SecureEdge-TDM can effectively detect major edge threats and provide actionable defense recommendations. Future work will extend the rule-based mapping layer with adaptive policy learning, real-time edge deployment, federated model updates, and automated defense orchestration.

## References


[1] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[2] S. Emanet, G. K. Baydogmus, and O. Demir, "An ensemble learning based ids using voting rule: Vel-ids," *PeerJ Computer Science*, vol. 9, p. e1553, 2023.

[3] M. Houichi, F. Jaidi, and A. Bouhoula, "Enhancing smart city security: An intrusion detection system using machine learning methods with the unb cic iot 2023 dataset," *IET Smart Cities*, vol. 7, no. 1, p. e70014, 2025.

[4] H. Q. Gheni and W. L. Al-Yaseen, "Two-step data clustering for improved intrusion detection system using ciciot2023 dataset," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 9, p. 100673, 2024.

[5] S. A. Almahaqeri, M. H. Almourish, A. A. Nasser, A. S. A. Alghawli, A. A. Elsayed, and A. N. Alhejoj, "An optimized gradient boosting framework for iot intrusion detection: a comprehensive evaluation on the ciciot2023 dataset," *Scientific Reports*, 2026.

[6] M. A. O. Ahmed, Y. Abdelsatar, R. Alotaibi, and O. Reyad, "Enhancing internet of things security using performance gradient boosting for network intrusion detection systems," *Alexandria Engineering Journal*, vol. 116, pp. 472–482, 2025.

[7] M. Nassef, "Boosting intrusion detection against ddos attacks using a feature engineering-based fine-tuned xgboost model," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 21, no. 1, pp. 1–39, 2025.

[8] A. A. Bhutta, M. u. Nisa, and A. N. Mian, "Lightweight real-time wifi-based intrusion detection system using lightgbm," *Wireless networks*, vol. 30, no. 2, pp. 749–761, 2024.

[9] D. Zhang, D. Huang, Y. Chen, S. Lin, and C. Li, "A lightweight iot intrusion detection method based on two-stage feature selection and bayesian optimization." *AIMS Electronics & Electrical Engineering*, vol. 9, no. 3, 2025.

[10] B. M. Kouassi, A. B. Ballo, K. J. Ayikpa, D. Mamadou, and M. Z. J. Coulibaly, "Top-k feature selection for iot intrusion detection: Contributions of xgboost, lightgbm, and random forest," *Future Internet*, vol. 17, no. 11, p. 529, 2025.

[11] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial internet of things," *Alexandria Engineering Journal*, vol. 81, pp. 371–383, 2023.

[12] A. Javed, A. Ehtsham, M. Jawad, M. N. Awais, A.-u.-H. Qureshi, and H. Larijani, "Implementation of lightweight machine learning-based intrusion detection system on iot devices of smart homes," *Future Internet*, vol. 16, no. 6, p. 200, 2024.

[13] M. M. Rashid, S. U. Khan, F. Eusufzai, M. A. Redwan, S. R. Sabuj, and M. Elsharief, "A federated learning-based approach for improving intrusion detection in industrial internet of things networks," *Network*, vol. 3, no. 1, pp. 158–179, 2023.

[14] R. Yang, H. He, Y. Wang, Y. Qu, and W. Zhang, "Dependable federated learning for iot intrusion detection against poisoning attacks," *Computers & Security*, vol. 132, p. 103381, 2023.

[15] Y. Sanjalawe, T. Alqurashi, Z. H. Alharbi, S. N. Makhadmeh, M. Alsharaiah *et al.*, "Adaptive graph attention-based federated learning for iot intrusion detection: mitigating poisoning attacks," *PeerJ Computer Science*, vol. 11, p. e3281, 2025.

[16] J. Wang, K. Yang, and M. Li, "Nids-fgpa: A federated learning network intrusion detection algorithm based on secure aggregation of gradient similarity models," *PloS one*, vol. 19, no. 10, p. e0308639, 2024.

[17] R. Bensaid, N. Labraoui, A. A. A. Ari, H. Saidi, J. H. M. Emati, and L. Maglaras, "Sa-flids: secure and authenticated federated learning-based intelligent network intrusion detection system for smart healthcare," *PeerJ Computer Science*, vol. 10, p. e2414, 2024.

[18] D. Javeed, M. S. Saeed, M. Adil, P. Kumar, and A. Jolfaei, "A federated learning-based zero trust intrusion detection system for internet of things," *Ad Hoc Networks*, vol. 162, p. 103540, 2024.

[19] J. Talpini, N. Civiero, F. Sartori, and M. Savi, "A federated approach to enhance calibration of distributed ml-based intrusion detection systems." in *ICAART (2)*, 2025, pp. 840–848.

[20] S. T. Ahmed, A. S. Fathima, M. A. Halwani, A. Almusharraf, and A. Albuali, "A trust-centric federated edge learning paradigm in healthcare for decentralized threat intelligence sharing," *Transactions on Emerging Telecommunications Technologies*, vol. 37, no. 4, p. e70401, 2026.

[21] B. A. K. Hammood and A. T. Sadiq, "Ensemble machine learning approach for iot intrusion detection systems," *Iraqi Journal for Computers and Informatics*, vol. 49, no. 2, pp. 93–99, 2023.

[22] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion detection framework for internet of things with rule induction for model explanation," *Sensors*, vol. 25, no. 6, p. 1845, 2025.

[23] Y. Deng, "Design of industrial iot intrusion security detection system based on lightgbm feature algorithm and multi-layer perception network," *Journal of Cyber Security and Mobility*, vol. 13, no. 2, pp. 327–347, 2024.