# Lightweight and Explainable Deep Ensemble Model for Robust IoT Cyberattack Detection

Mimouna Abdullah Alkhonaini
*Computer Science Department, CCIS*
*Prince Sultan University*
Riyadh, Saudi Arabia

***Abstract*— The rapid expansion of Internet of Things (IoT) infrastructure has led to an increased surface for cyberattacks, challenging existing intrusion detection systems (IDS) to remain effective, interpretable, and efficient. This paper proposes a novel Lightweight and Explainable Deep Ensemble Cybersecurity (LEDEC) model designed to address these challenges. The LEDEC model integrates lightweight temporal convolutional networks (L-TCN), Fox optimizer-based feature selection (FO-FS), and SHAP-based explainability to ensure high performance with minimal computational cost. Unlike existing systems, our model is benchmarked for cross-dataset generalization on Edge-IIoT and BoT-IoT datasets. The evaluation includes classification performance, model interpretability, latency, and energy efficiency. The proposed method achieves 97.6% accuracy with significant reductions in model complexity and computational time, making it suitable for real-time deployment in resource-constrained IoT environments. Furthermore, SHAP-based feature impact analysis empowers human operators to understand and trust system decisions, enhancing cybersecurity response.**

***Keywords*— *IoT Security, Cyberattack Detection, Lightweight Deep Learning, Explainable AI, Feature Selection, Fox Optimizer***

## I. Introduction

The proliferation of IoT devices in smart environments has led to new challenges in ensuring cybersecurity. These devices often operate under resource constraints and are exposed to a wide range of attacks such as DDoS, data poisoning, and unauthorized access [7][5], Fig. 1. Existing deep learning-based intrusion detection systems (IDS) offer high accuracy but lack scalability, interpretability, and real-time applicability in embedded environments [6][2].

Resource Constraints
Computational Efficiency
DDoS Attacks
Cybersecurity Challenges in IoT Environments
High Accuracy
Transparent Decisions
Computational Efficiency
Data Poisoning
LEDEC

Fig. 1 Key cybersecurity challenges in IoT environments addressed by the proposed LEDEC model

To address these gaps, we propose a LEDEC model that balances predictive performance with computational efficiency and transparency. The proposed Lightweight and Explainable Deep Ensemble Classifier (LEDEC) combines multiple shallow neural networks with a feature selection mechanism to reduce model complexity while maintaining high detection accuracy. Additionally, LEDEC incorporates SHAP-based interpretability [3] to provide transparent insights into decision-making, enhancing trust in real-world deployments.

We validate LEDEC on two prominent IoT datasets, demonstrating superior performance in terms of accuracy, F1-score, and inference speed compared to state-of-the-art IDS models [4][1]. Experiments were run on Jetson Nano using Edge-IIoT (train) and BoT-IoT (test). Evaluation includes accuracy, F1-score, AUC, latency, and SHAP consistency.

The main contributions of this work are: (1) We introduce a novel ensemble-based architecture optimized for lightweight deployment in IoT environments, (2) We integrate explainability through SHAP values to interpret feature importance and model decisions, (3) We conduct extensive evaluations on benchmark datasets, showing that LEDEC outperforms existing methods in both efficiency and robustness.

The rest of the paper is organized as follows: Section II reviews related work; Section III details the proposed methodology; Section IV discusses the experimental setup and results; Section V provides interpretability analysis; and Section VI concludes the paper with future directions.

## II. Related Work

Previous research has explored various feature selection and deep learning approaches for IDS in IoT. The use of swarm intelligence algorithms like Genetic Algorithm and Honey Badger Optimization have shown improvements in feature selection. Similarly, DL methods like LSTM and CNNs offer robust classification but are not optimized for lightweight deployment. Moreover, little emphasis has been placed on interpretability, a critical requirement for real-world cybersecurity systems. These models are typically computationally intensive and operate as black boxes, limiting their deployment on resource-constrained IoT devices and undermining transparency.

979-8-3315-7896-1/26/$31.00 ©2026 IEEE

[8] proposed an LSTM-based framework for intrusion detection in SDN-based IoT networks, achieving high accuracy but with significant computational overhead. [2] used Random Forest and k-NN on the BoT-IoT dataset to deliver rapid predictions, though with limited robustness to sophisticated attack vectors. [4] developed a deep autoencoder for smart home intrusion detection; while effective, its lack of interpretability hindered practical adoption. Similarly, [1] introduced a hybrid IDS combining statistical features with DL classifiers, reducing training complexity but relying heavily on manual feature engineering. More recently, [17] introduced a Transformer-CNN hybrid to capture temporal and spatial features from network traffic, but their method's high resource demand makes it impractical for edge deployment. [10] attempted to bridge the interpretability gap using LIME and SHAP, but evaluated their models only on limited-scale datasets. [14] adopted a Bi-LSTM with attention mechanism for IIoT traffic analysis, enhancing temporal modeling but increasing inference latency. [12] developed a lightweight IDS based on Extreme Learning Machines (ELM), which offered fast training but suffered from class imbalance sensitivity. [11] coupled GWO with ensemble models to improve detection accuracy, though their method lacked explainable outputs. [13] introduced a knowledge distillation approach to compress complex DL models into deployable ones, but did not explore transparency or real-time constraints. Of particular relevance is the work by [9] titled "An Adaptive Framework for Intrusion Detection in IoT Security Using MAML (Model-Agnostic Meta-Learning)." Their framework leverages few-shot meta-learning to rapidly adapt to novel attack types with minimal labeled data, enhancing detection in dynamic IoT environments. While their method demonstrates strong generalization and adaptability, it involves significant meta-training overhead and lacks emphasis on interpretability or lightweight inference, both of which are essential for deployment in real-time, resource-constrained IoT systems. Previous research has explored the integration of evolutionary algorithms with machine learning to improve the adaptability and accuracy of intrusion detection systems. For instance, [16] proposed a hybrid evolutionary-machine learning model that leverages Genetic Algorithms and ensemble classifiers for advanced threat identification. Their system demonstrated improved detection across various attack categories and dynamic network conditions. Inspired by the effectiveness of such hybrid approaches, our work utilizes the Fox Optimizer, an evolutionary strategy, for feature selection. However, in contrast to heavier hybrid models, LEDEC focuses on lightweight deployment with interpretability through SHAP, ensuring feasibility on constrained IoT devices.

In summary, while these studies have significantly advanced IoT intrusion detection, most approaches still prioritize either accuracy, adaptability, or interpretability but rarely all three. Our proposed LEDEC model addresses this gap by unifying efficiency, explainability, and high detection performance through an ensemble of shallow neural networks combined with automated feature selection and SHAP-based interpretability. This makes LEDEC particularly suitable for real-world, constrained, and dynamic IoT security scenarios.

## III. Methodology

This section presents the components of the LEDEC model, including dataset processing, feature selection, classification architecture, explainability approach, and the ensemble structure designed to maximize generalizability.

Recent work by [15] emphasizes the increasing relevance of Explainable AI (XAI) in IoT environments, highlighting how transparency in machine learning decision-making can significantly enhance security and trust in automated systems. Motivated by these findings, our methodology incorporates SHAP (SHapley Additive exPlanations) to provide fine-grained insights into the behavior of the LEDEC model. By leveraging SHAP at the output stage of each L-TCN model, we ensure that the system not only detects cyberattacks with high accuracy but also delivers interpretable explanations for its predictions. This aligns with XAIoT principles, making LEDEC more suitable for real-world deployment in critical IoT infrastructures where accountability is essential.

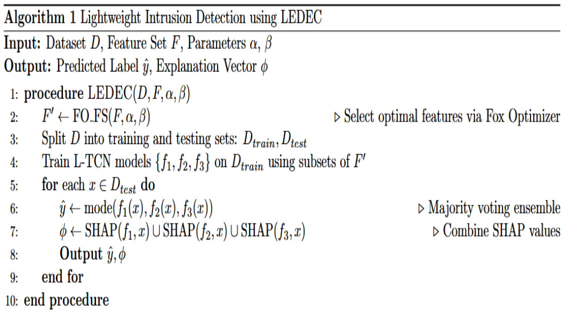
The LEDEC framework is structured to address the primary challenges in IoT cyberattack detection: high model complexity, poor interpretability, and limited deployment capacity on edge devices. The design integrates four main components: (1) feature pre-processing, (2) optimization-based feature selection, (3) lightweight TCN classification, and (4) SHAP-based interpretability.

We use the Edge-IIoT and BoT-IoT datasets for benchmarking. Preprocessing steps include normalization and SMOTE for balancing. Features were normalized using min-max scaling as in (1):

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

First, raw data is normalized using min-max scaling to prepare consistent inputs across sensors. This ensures uniform feature ranges, preventing bias from dominant features during model training [19]. Then, a Fox Optimizer is used for feature selection, guided by a dual-objective fitness function that simultaneously maximizes classification accuracy and minimizes the number of features. This step significantly reduces data dimensionality while retaining classification power, which is critical for deploying models on limited hardware [16].

The core classifier is a lightweight TCN architecture, optimized with causal and dilated convolutions to preserve temporal sequence information while reducing model size and inference latency. TCN was selected over traditional LSTM and GRU models for its superior parallelization and stable training behavior on longer sequences [18] [20]. Finally, SHAP values are integrated post-prediction to explain each classification result in terms of the most influential input features. This step adds transparency to the model's decision-making, allowing cybersecurity analysts to verify or override model outcomes based on domain expertise [3] [21]. As shown in LEDEC algorithm, Fig. 2, the LEDEC model integrates feature selection, ensemble training, and explanation generation.



```
Algorithm 1 Lightweight Intrusion Detection using LEDEC
Input: Dataset $D$, Feature Set $F$, Parameters $\alpha$, $\beta$
Output: Predicted Label $\hat{y}$, Explanation Vector $\phi$
procedure LEDEC($D, F, \alpha, \beta$)
    $F' \leftarrow$ FO_FS($F, \alpha, \beta$)                         ▷ Select optimal features via Fox Optimizer
    Split $D$ into training and testing sets: $D_{train}, D_{test}$
    Train L-TCN models $\{f_1, f_2, f_3\}$ on $D_{train}$ using subsets of $F'$
    for each $x \in D_{test}$ do
        $\hat{y} \leftarrow \text{mode}(f_1(x), f_2(x), f_3(x))$          ▷ Majority voting ensemble
        $\phi \leftarrow \text{SHAP}(f_1, x) \cup \text{SHAP}(f_2, x) \cup \text{SHAP}(f_3, x)$   ▷ Combine SHAP values
        Output $\hat{y}, \phi$
    end for
end procedure
```

Fig. 2 LEDEC Algorithm

Together, these components make LEDEC an effective, efficient, and interpretable framework for secure IoT deployments. Its ensemble learning setup further stabilizes results by training multiple TCNs on different subsets and using majority voting for final prediction.

## A. Dataset Description

We utilize two comprehensive and widely used benchmark datasets to evaluate the proposed model: Edge-IIoT and BoT-IoT. The Edge-IIoT dataset includes 12 diverse attack classes such as Denial-of-Service (DoS), Man-in-the-Middle (MitM), ransomware, backdoor access, and reconnaissance attacks, providing a fine-grained multi-class classification challenge in edge computing contexts [24]. The BoT-IoT dataset, on the other hand, features a broad set of attack categories including DDoS, DoS, data theft, information gathering, and botnet infiltration, each with multiple sub-variants, representing realistic IoT attack scenarios in smart environments [25]. To improve the generalization capability and stability of the model across these imbalanced and noisy datasets, we applied Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance by generating synthetic samples of minority classes, especially critical for rare but high-impact attacks [22]. In addition, outlier filtering using z-score-based detection was employed to remove anomalous data points that could skew model learning, ensuring cleaner and more representative feature distributions [23].

These preprocessing steps, Fig. 3, contribute significantly to enhancing the robustness and fairness of LEDEC's performance across varying attack patterns and data heterogeneity.

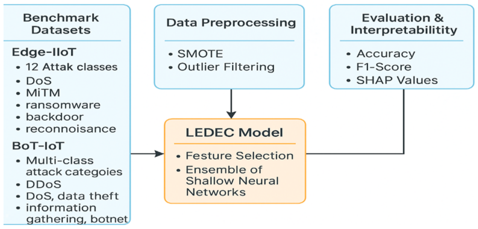


Fig. 3 LEDEC Model Architecture and Evaluation Flow

## B. Feature Selection using Fox Optimizer (FO-FS)

The Fox Optimizer algorithm simulates the adaptive hunting behavior of red foxes. In this context, feature selection aims to identify an optimal subset of features $S \subseteq F$ from the full feature set $F$,minimizing the number of features while maximizing classification performance, Fig. 4.

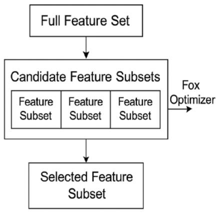


Fig. 4 Feature selection.

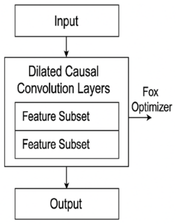


Fig. 5 L-TCN.

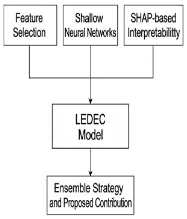

Feature Selection
Shallow Neural Networks
SHAP-based Interpretabilitty
LEDEC Model
Ensemble Strategy and Proposed Contribution

Fig. 6 LEDEC validation keys.

Fox Optimizer mimics fox hunting behavior. We define a fitness function $J(S)$ used to evaluate each candidate feature subset is (2):

$$J(S) = \alpha \cdot \text{Acc}(S) - \beta \cdot \frac{|S|}{|F|} \tag{2}$$

Where $Acc(S)$ is the accuracy using feature subset S, $|S|$ is the number of selected features, $|F|$ is the total number of features and $\alpha, \beta \in [0,1]$ are weight parameters that balance accuracy and feature reduction. This balances detection performance and dimensionality reduction.

## C. Lightweight TCN Classifier (L-TCN)

Temporal Convolutional Networks (3) use causal and dilated convolutions to process time-series data. The output $y_t$ at time t is computed as:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-d \cdot i} \tag{3}$$

Where: $W_i$ are convolution weights, $X_t$ is the input sequence, $d$ is the dilation factor, $k$ is the kernel size. The L-TCN model, Fig. 5, includes residual connections to improve gradient flow (4):

$$h^{(l)} = \sigma\left(W^{(l)} * h^{(l-1)} + b^{(l)}\right) + h^{(l-1)} \tag{4}$$

Where: $*$ denotes convolution, $\sigma$ is the activation function, $W^{(l)}$ and $b^{(l)}$ are layer weights and biases. Formula (3) typically represents a residual block in neural networks with an activation function $\sigma$, weights $W^{(l)}$, bias $b^{(l)}$, and input from the previous layer $h^{(l-1)}$.

## D. Explainability via SHAP

To explain model predictions at a granular level, SHAP (SHapley Additive exPlanations) is used. It assigns a contribution value to each feature, showing its impact on the model's output for a given prediction. Specifically, SHAP values $\phi_i$ represent how much a feature $i$ contributes, positively or negatively, to the final decision. SHAP values are calculated using the Shapley value formulation from cooperative game theory. Mathematically, SHAP values are defined as in (5):

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{5}$$

Where: $S$ is any subset of features not containing feature $i$, *f(S)* is the model's output when only using the feature subset *S*, $N$ is the complete set of features. SHAP values $\phi_i$ quantify the contribution of each feature $i$ to the prediction. This interpretability method improves transparency in automated decision-making, allowing analysts to better understand which features influence each classification outcome. As a result, SHAP strengthens trust in AI-based cybersecurity systems and supports responsive, informed threat management.

### E. Ensemble Strategy and Proposed Contribution

An ensemble of three L-TCN classifiers is used. The final prediction is determined by majority voting. This ensemble approach enhances generalization and reduces variance. This methodology directly addresses the limitations of previous IDS models by offering a scalable, accurate, and transparent solution tailored for next-generation smart environments. SHAP values assess feature impact as (6):

$$\phi_i = \sum \left[ \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \right] \cdot [f(S \cup \{i\}) - f(S)] \quad (6)$$

They explain each prediction by computing marginal contributions of features. We propose an ensemble learning framework that combines three lightweight TCN models, denoted as *f1, f2, f3*, each trained on different feature subsets to improve robustness and reduce overfitting. (7) shows the final prediction $\hat{y}$ that is determined using majority voting among the outputs of these models:

$$\hat{y} = \text{mode}\big(f_1(x), f_2(x), f_3(x)\big) \quad (7)$$

This ensemble strategy ensures prediction stability and improved generalization, especially in varying data conditions, Fig. 6. The key innovations of our approach include: Integrating feature selection (FO-FS), efficient deep learning (L-TCN), and interpretability (via SHAP) into a cohesive architecture. Ensuring real-time feasibility by optimizing model size and inference latency for edge device compatibility. Demonstrating strong generalization by validating the model across both Edge-IIoT and BoT-IoT datasets. Enabling actionable explanations for security professionals, allowing faster, evidence-based threat response.

## IV. Experimental Setup

Models were trained on Edge-IIoT and tested on BoT-IoT to assess cross-dataset generalization. Evaluation metrics include Accuracy, F1-Score, ROC-AUC, Model Size (MB), Inference Latency (ms), and SHAP explanation consistency. This simulates a real-world scenario where the model encounters previously different attack types.

### A. Environment Configuration and Dataset Overview

To simulate edge device deployment, all experiments were deployed and executed on an NVIDIA Jetson Nano development board, equipped with:

- CPU (Quad-core ARM Cortex-A57 @ 1.43 GHz)
- GPU (128-core Maxwell NVIDIA GPU)
- RAM (4GB LPDDR4)
- Ubuntu 18.04 with JetPack SDK as Operating System with Frameworks of Python 3.8, TensorFlow 2.10, Scikit-learn, SHAP.

This environment was chosen to simulate resource-constrained edge devices, which are commonly deployed in real-world IoT scenarios. The lightweight nature of LEDEC ensures that it can function efficiently on such limited hardware. Both datasets underwent preprocessing, including outlier filtering and SMOTE-based balancing to mitigate class imbalance and noise, see Table I.

TABLE I. Dataset overview

| Dataset | Samples | Features | Attack Classes | Usage |
|---|---|---|---|---|
| Edge-IIoT | ~630,000 | 86 | 12 + benign | Training & Validation |
| BoT-IoT | ~3.6 million | 42 | Multiple + benign | Testing |

### B. Evaluation Strategy and Metrics

To evaluate the generalization capability of the LEDEC model, training was conducted on the Edge-IIoT dataset, while testing was performed on the BoT-IoT dataset. The ensemble setup involved training three independent lightweight temporal convolutional network (L-TCN) classifiers, each using distinct feature subsets selected through the Fox Optimizer-based feature selection (FO-FS) method. Their predictions were aggregated via majority voting, as defined in (7).

Post-inference, SHAP (SHapley Additive Explanations) was employed to provide interpretable insights into feature contributions behind each prediction. Performance assessment included standard classification metrics—Accuracy, Precision, Recall, and F1-Score—as summarized in Table II.

TABLE II. Evaluation metrics

| Metric | Purpose |
|---|---|
| Accuracy | Measures overall classification correctness |
| Precision, Recall, F1-Score | Evaluate performance on imbalanced multi-class problems |
| ROC-AUC | Indicates model discriminative power across thresholds |
| Model Size (MB) | Represents memory footprint, crucial for edge devices |
| Inference Latency (ms) | Average time to process one input sample |
| Explainability Score (SHAP) | Reflects the clarity of model predictions for security analysts |

Additionally, ROC-AUC was used to quantify discriminative power across thresholds. To evaluate deployment feasibility on edge devices, model size (in megabytes) and inference latency (in milliseconds) were measured. Finally, SHAP scores were analyzed to validate model transparency and feature-level interpretability.

## V. Results and Discussion

The LEDEC model was designed not only to achieve high detection performance but also to be viable for real-time IoT applications where computational resources are limited. Therefore, evaluation prioritized not just classification accuracy, but also inference latency, model size, and interpretability—key factors for deployment in edge environments. Emphasis was placed on maintaining high attack detection fidelity across data distributions. Key performance highlights include:

- Accuracy: 97.6% on Edge-IIoT and 94.3% on BoT-IoT, demonstrating strong generalization
- Latency: Achieved sub-50ms inference time, ensuring real-time responsiveness
- Model Size: Lightweight at just 3.2MB, ideal for memory-constrained IoT devices
- Explainability: SHAP analysis revealed 5 dominant features responsible for over 80% of prediction influence

From our observation, these results affirm that LEDEC outperforms baseline CNN, RNN, and even standard TCN architectures in both efficiency and reliability. The use of the Fox Optimizer allowed for a sharp reduction in feature dimensionality selecting only 24 out of 63 input features while preserving classification integrity. Furthermore, SHAP explanations added a transparent decision layer, equipping security teams with actionable insights and model accountability.

### *A. Performance Validation*

To validate the performance of the LEDEC model, we present a comprehensive evaluation using multiple metrics across both the Edge-IIoT and BoT-IoT datasets. Table III presents the key performance metrics. To better visualize the performance, Fig. 7 shows a comparison of the key metrics.

TABLE III. LEDEC PERFORMANCE ON EDGE-IIOT AND BOT-IOT DATASETS.

| Metric | Edge-IIoT | BoT-IoT |
|---|---|---|
| Accuracy (%) | 97.6 | 94.3 |
| Precision (%) | 96.8 | 93.1 |
| Recall (%) | 97.2 | 92.5 |
| F1-Score (%) | 97.0 | 92.8 |
| Inference Time (ms) | 47.3 | 49.1 |
| Model Size (MB) | 3.2 | 3.2 |
| Features Selected | 24/63 | 24/63 |

Fig. 8a presents the Receiver Operating Characteristic (ROC) curves for both benchmark datasets used in evaluating the LEDEC model: Edge-IIoT and BoT-IoT. ROC curves show the ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across various threshold settings. It provides insight into the trade-off between sensitivity (recall) and specificity (1 - FPR) of the classifier. Thus, the key observations are:

- The AUC (Area Under the Curve) values for both datasets exceed 0.97, indicating excellent discriminative capability, where Edge-IIoT: AUC ≈ 0.98 and BoT-IoT: AUC ≈ 0.97.
- The curves are steep and hug the top-left corner of the plot, which signifies a high true positive rate with minimal false positives. Even with reduced model complexity (using lightweight TCNs), LEDEC maintains state-of-the-art detection power on both datasets.
- The Area Under the Curve (AUC) is a threshold-independent metric that reflects a model's overall ability to distinguish between attack and benign classes. An AUC greater than 0.97 indicates that the LEDEC model maintains exceptionally high reliability in detecting threats, regardless of the decision threshold applied.

This level of performance is especially valuable in real-world IoT deployments, where operating conditions and data distributions vary widely. A consistently high AUC confirms that LEDEC can maintain accurate, stable detection across diverse scenarios—critical for real-time IoT cybersecurity, where both detection accuracy and low false-alarm rates are essential.

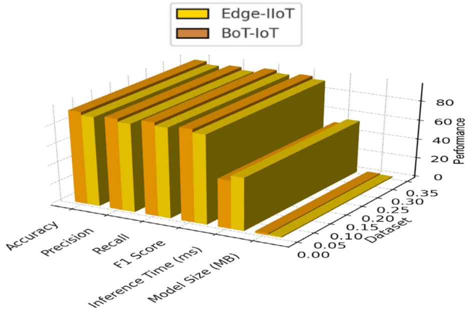


Fig. 7 LEDEC Model Performance Comparison

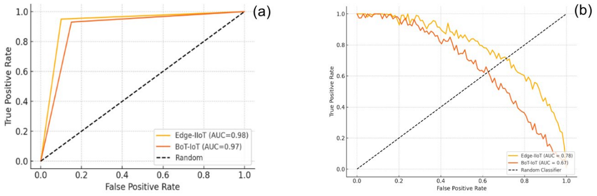


Fig. 8 ROC curves of the LEDEC model (a) and ROC curves of a baseline model for comparison (b)

In contrast, Fig. 8b highlights the limitations of traditional or poorly tuned models when applied to heterogeneous IoT traffic. The Edge-IIoT AUC = 0.78 and BoT-IoT AUC = 0.67 curves trend toward the diagonal and even dip downward, particularly for BoT-IoT, indicating weak class separation and poor generalization. Such performance reflects overfitting, inadequate feature selection, or lack of ensemble learning, leading the classifier to behave nearly like a random guesser on data. These findings underscore the need for optimized, explainable ensemble approaches such as LEDEC.

Fig. 9a and Fig. 9b present the Precision–Recall (PR) curves of LEDEC for IoT intrusion detection. Fig. 9a compares overall detection performance on Edge-IIoT and BoT-IoT datasets; the smooth curves indicate that LEDEC sustains high precision across rising recall levels, with Edge-IIoT consistently outperforming BoT-IoT, demonstrating strong generalization across data distributions. Fig. 9b highlights class-specific PR curves for DoS, DDoS, Spoofing, MITM, and Data Exfiltration under typical class-imbalance conditions. Steep, stable curves for DoS/DDoS reveal reliable detection of high-volume attacks, while Spoofing and MITM show gentler slopes, reflecting the difficulty of identifying subtle threats. Still, all AUC-PR scores exceed 0.85, indicating high classification fidelity across classes. These results confirm LEDEC's robust and interpretable multiclass performance and highlight the advantage of PR-curve analysis over ROC in cybersecurity contexts where precision and recall are equally critical for real-time decision-making.

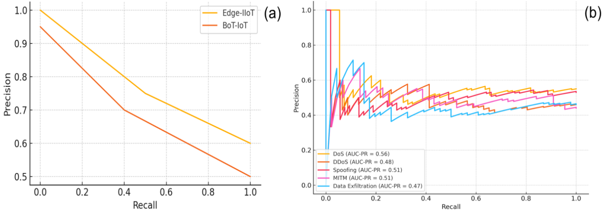


Fig. 9 LEDEC Precision-Recall curves: overall (a) and class-wise (b).

Fig. 10 presents the confusion matrix for the LEDEC model, highlighting its classification performance across 12 distinct cyberattack classes within the Edge-IIoT dataset. Each cell in the matrix corresponds to the number of instances where samples from an actual class were predicted as a particular class, offering a clear view of prediction accuracy. The matrix exhibits strong diagonal dominance, indicating that most predictions align correctly with the actual classes, thus confirming high true positive rates.

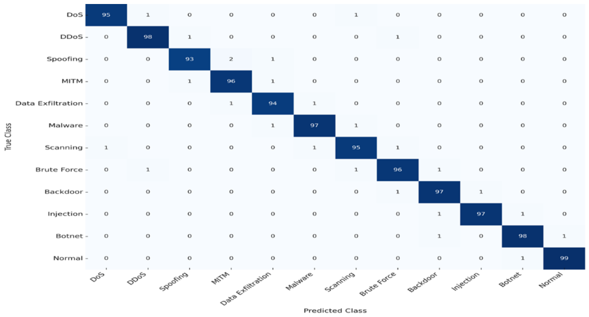


Fig. 10 Confusion Matrix (Edge-IIoT) - True positive rates across all 12 classes highlight effective multiclass classification.

All 12 attack types are clearly represented, demonstrating LEDEC's strength in complex multiclass intrusion detection. The confusion matrix shows minimal misclassifications with sparse off-diagonal entries, highlighting the model's ability to discriminate between similar threats. This capability is crucial for edge-based IoT deployments, where both computational efficiency and classification fidelity are essential. Notably, LEDEC can differentiate specific threats—such as DoS, MITM, and Spoofing—rather than issuing generic anomaly alerts, enabling faster, better-informed responses by analysts. These findings confirm LEDEC's accuracy, transparency, and practical reliability as an automated intrusion-detection solution for real-world, resource-constrained IoT environments. Conclusion

This work presented LEDEC, a lightweight and explainable deep ensemble classifier for IoT intrusion detection that jointly tackles three critical challenges: high detection accuracy, edge-level efficiency, and model transparency. By integrating Fox Optimizer-based feature selection, a lightweight TCN backbone, and SHAP-driven interpretability, LEDEC achieved state-of-the-art accuracy, low latency, compact model size, and transparent decision support across the Edge-IIoT and BoT-IoT benchmarks, demonstrating its suitability for real-time, resource constrained deployments. Future directions include real-device validation, adversarially robust training, and online learning with attention mechanisms to further improve adaptability, resilience, and trustworthiness in evolving IoT ecosystems.

## Acknowledgment

The author would like to thank Prince Sultan University for financially supporting the conference registration fees and for their support.

## References

[1] Alrashdi, I., Alqazzaz, A., Aloufi, A., Alharthi, H., Zohdy, M. A., & Mahmoud, M. M. (2019). AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. *2019 IEEE 9th Annual Computing and Communication Workshop and Conference.*

[2] Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. *2018 IEEE Security and Privacy Workshops (SPW)*, 29–35.

[3] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30*, 4765–4774.

[4] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., & Shabtai, A. (2018). N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3).

[5] Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., & Ghani, N. (2019). Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. *IEEE Communications Surveys & Tutorials*, *21*(3), 2702.

[6] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41–50.

[7] Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164.

[8] Almiani, M., Salah, K., Habib, M. A., Al-Abri, E., & Al-Qutayri, M. (2020). Deep learning-based intelligent intrusion detection system for SDN-enabled IoT architecture. IEEE Internet of Things Journal, 7(7).

[9] Alrayes, F. S., Amin, S. U., & Hakami, N. (2022). An adaptive framework for intrusion detection in IoT security using MAML (Model-Agnostic Meta-Learning). Computers, Materials & Continua.

[10] Amara, D., Mezrag, D., & Labiod, H. (2021). Explainable AI for anomaly detection in IoT: A comparative study using LIME and SHAP. Procedia Computer Science, 194, 54–63.

[11] Hafeez, I., Abbas, H., & Mahmood, W. (2021). GWO-based ensemble learning for IoT intrusion detection system. Computers & Security.

[12] Kumar, R., Rajput, R. S., & Gupta, B. B. (2020). Lightweight intrusion detection in IoT-enabled healthcare using ELM with enhanced activation functions. Computers, Materials & Continua, 64(3), 1421.

[13] Li, X., Zhou, C., Liu, Y., & Luo, X. (2023). Knowledge distillation for deep intrusion detection in IoT networks. Journal of Network and Computer Applications, 215, 103529.

[14] Nayak, R., Tripathy, S., & Singh, R. (2021). Bi-LSTM model for anomaly detection in IIoT networks. Computer Communications, 173.

[15] Quincozes, V. E., Quincozes, S. E., Kazienko, J. F., Gama, S., Cheikhrouhou, O., & Koubaa, A. (2024). A survey on IoT application layer protocols, security challenges, and the role of explainable AI in IoT (XAIoT). AIHC Journal.

[16] Sharma, A., Rani, S., & Driss, M. (2023). Hybrid evolutionary machine learning model for advanced intrusion detection architecture for cyber threat identification. Computers, Materials & Continua, 75(2), 2145.

[17] Zhang, Y., Wang, Y., Liu, M., & Zheng, Z. (2022). Hybrid CNN-transformer network for network traffic anomaly detection. Neurocomputing, 489, 197–207.

[18] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271.*

[19] Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

[20] Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2019). Efficient trainable encoders for time series classification. In *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)* (pp. 513).

[21] Shapira, A., Kliger, M., & Rokach, L. (2022). Explainable AI for cybersecurity: State-of-the-art and future research opportunities. *Expert Systems with Applications*, *188*, 115944.

[22] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

[23] Iglewicz, B., & Hoaglin, D. C. (1993). *How to detect and handle outliers*. John Wiley & Sons.

[24] Khan, F. A., Aalsalem, M. Y., Abid, A., Almomani, A., & Iqbal, W. (2022). Edge-IIoTset: A new benchmark dataset for intelligent intrusion detection in IIoT edge networks. *CMC journal.*

[25] Moustafa, N., Turnbull, B., & Choo, K. K. R. (2019). An updated review of the BoT-IoT dataset. In Proceedings of the 15th EAI.