Minimax Theory and its Applications
Volume 11 (2026), No. 3s, 520 to 533

An Intelligent Collaborative Edge–Cloud Deep Learning Framework for Secure and Real-Time Anomaly Detection in Internet of Things Networks

Ramadhurai.R[1], Dr.F. Jerald[2], Dr.M.Anand[3], Dr.G.Thiraviya Suyambu[4], Dr.A.Vinoth kumar[5]

[1]Research Scholar, Department of Electronics and Communication Engineering, Dr.M.G.R.Educational and Research Institute, Chennai, India
[2]Associate Professor, Department of Electronics and Communication Engineering, Dr.M.G.R.Educational and Research Institute, Chennai, India
[3,5]Professor, Department of Electronics and Communication Engineering, Dr.M.G.R.Educational and Research Institute, Chennai, India
[4]Asst.Professor, Department of Electronics and Communication Engineering, M.A.M.School of Engineering, Tiruchirappalli, India.
Email: ramadhurai.r@gmail.com[1], jerald.ece@drmgrdu.ac.in[2], anand.ece@drmgrdu.ac.in[3], deanacademics@mamse.in[4], vinothkumar.ece@drmgrdu.ac.in[5]

Abstract
Internet of Things (IoT) networks connections reached 21 Billion across the globe [4]. As their population grows, the attacks also increases. They are susceptible for various cyber-attacks such as Botnet, DDoS, MitM, Firmwere, Encryption, Brute force etc., To prevent IoT networks from such attacks, the Intrusion Detection Systems (IDS) are used. Three types of IDS prominently used in the industry. Cloud based IDS, Edge Computing based IDS and Deep Learning based IDS. Cloud approach experiences latency, Edge Computing approach experiences resource constraints while the Deep Learning approach need high computation capability. This article proposes a novel Collaborative Edge–Cloud Deep Learning Framework (EC-DLF) to reap the benefits of three design approaches. The pruning and knowledge distillation helps IoT devices to handle the intrusion detection with the less resource of about < 2MB RAM with the very less latency of <4ms. The cloud approach used in this work uses 6 layer encoder with multi heads, which tested on 4 different datasets namely, N-BaIoT, TON_IoT, Edge-IIoTset, and CICIoT2023. The results were impressive. F1 Score of 99.43%, AUC-ROC of 0.9979 and MCC of 0.9932. These results were compared with five baseline models and the proposed model outperformed all the baseline models. Further the statistical analysis also done to confirm the robustness of the system. Wilcoxon signed-rank test was used, which yields the $p < 0.001$ and the Cohen's $d > 4.2$. On the system performance aspect, t-SNE analysis, throughput, cross data validation, scalability analysis also performed.

Keywords: Internet of Things, Edge-Cloud, Intrusion detection, Anomaly detection, Multi-Transformer, Edge MobilNet, N-BaIoT, TON_IoT, CNN, LSTM

1. Introduction

Huge volume of data generated from rapid and increased usage of Internet of Things (IoT) devices across various industries such as healthcare, automation, transportation, manufacturing etc., [1]. Correspondingly the number of nodes and end points are also increased exponentially. By 2030, it is projected to reach 26 billion [2]. These endpoints have resource constraints and could be easy targets for cyber-attacks.

Such security breaches have a consequences beyond the data loss. If it happened in medical IoT network, that would cause severe threats to patient's lives. If it happens in water distribution or power generation plants, then large number of people may be affected by malfunctioning of such systems. [3]. Hence there is a need for protecting these nodes and also to detect the intrusion as quick as possible.

There are several IDS available in the field. They shall be grouped in three categories listed below:

- Cloud based IDS: The Models based on LSTM and Transformers demonstrated higher accuracy in prediction of intrusion. But they need of high end systems to perform the tasks due to the complexity of the framework [5]. So, the IoT nodes needs to transmit the data to server, where the processing will happens

and the threat details will be transmitted back to the nodes by the server. But, this entire process involves the round trip time of 100 to 300 milli seconds [6]. Some critical systems like medical IoT could not able to afford this much latency, because usually medical IoT needs response in less than 1 milli second [7].

- Edge IoT IDS: This category is especially designed to meet out the resource constraints of IoT devices. They designed to work with the hardware limitations of Less than 5MB RAM. Variants such as MobileNetV4, EdgeNeXt, MicroTVM are belongs to this category.[8]
- Standalone ML based IDS: These systems are accurate, but need RAM of 50MB to 500MB. So, they may not be suitable to the industrial IoT that houses smaller RAMs to save the power [9].
- Models that are accurate but not designed for edge deployment have 50–500 MB parameter footprints [9], making them infeasible on IoT edge hardware with 512 KB–4 MB RAM budgets [10].

No single unified structure is available to address the detection metrics and system metrics. The framework proposed in this work trued to address this issue by analysing latency, accuracy, scalability and cross dataset generality. Various research gaps and the objectives of the work are discussed in section 2.5.

The contributions of this research are:

- Unified Edge-Cloud architecture for IDS in Industrial IoT is proposed.
- Dynamic offloading policy applied to address the accuracy – latency trade-off.
- The Knowledge-Distillation principle used in NLP is applied in this work to look for the resource reduction in IoT systems.

Literature review discussed in Section 2. Architecture, methodology and metrics are discussed in Section 3. While the results and discussions are given in Section 4. Conclusion and further scope for improvement is narrated in Section 5.

2. Literature review

2.1. IDS using ML and DL

Several Machine Learning approaches used in IDS for IoT devices. To name a few, Random Forest applied for classification [11], Support Vector Machine based IDS worked well in distinguishing the threats from each other [12], Neural networks based binary classification with less number of layers also gave impressive results [13].

Alhusnawiet al.[14] built hybrid architecture to classify the IoT threats. Qathrady et al [15] proposed bio inspired transformer approach to improve the accuracy of IoT intrusion detection. Fraihat et al.[16] proposed federated learning approach, which yielded the better accuracy than counterparts. CNN based IDS introduced by Mohy-Eddine et al. [17], which used less resources than other ML based IDS. In this way, several Machine Learning and Deep Learning concepts are applied to IoT IDS.

2.2. Edge-cloud approach

Cloud storage is used in this approach for providing solution to the memory constraint. Mao et al. [18] proposed the framework which uses the cloud for edge computing nodes with resource optimization. Latency-memory trad-off discussed by Shi et al. [19], which took survey of various edge IoT architectures and made a detailed analysis of latency-memory trade-off. Li et al. [20] demonstrated that the dynamic policy decisions shall have 23% lesser latency than static decisions in edge-cloud IoT architecture. Diro and Chilamkurti [21] extended this work and proposed the machine learning based model for dynamic decision making and hence to reduce the latency.

2.3. Least Resource models

Hassija et al. [22] made a survey on various IoT architectures to look for possibility to address the resource constraints of IoT networks. MobileNet [23] applied convolution method to reduce the parameters and further to reduce the resource requirement. This work is extended by Howard et al. [24], in which, the traditional widening approach is reversed and started with narrow span. Bhatt et al. [25] applied the method called “knowledge distillation” in which small models are try to keep the accuracy of large model, but works faster than large models with least resources.

2.4. Multi Transformer approaches

Y. Meidan et al[26] applied transformer approach to intrusion detection. Originally this technique was applied to NLP, where data are tokenized and used as a parcels of text. By this approach, the different intrusions shall be distinguished easily. Lin et al. [27] applied multi-transformer approach with CNN based classification, given good accuracy. Nguyen et al. [28] applied multi-transformer approach to the encrypted data transfer. Yang et al. [29] used cloud – Edge architecture for IoT networks using transformers to detect intrusions in Industrial IoT.

2.5. Research gap

The detailed survey as mentioned above, reveals some research gaps.

1. IoT IDS compromises on the any one of the factor in trade-off with another. Common IoT IDS that addresses the issues of accuracy, latency, edge computing, cloud storage and scalability are not available.
2. Adaptive policies for offloading shall be applied and analysed to find whether they reduce the resources or not?
3. The idea of knowledge distillation used in MobileNet shall be explored for Industrial IoT for intrusion detection with the objective of improving the accuracy.
4. Statistical evaluations such as MCC, Cross-dataset migration and generalisation needs to be applied to IDS and the improvement in accuracy and reduction of latency shall be studied.
5. The two perspective metrics namely, detection metrics and system metrics shall be addressed in single framework.

3. Proposed Edge – Cloud Architecture

The proposed architecture consists of 5 stages.

Stage 1: Data adoption from multiple IoT sources.
Stage 2: ML based pre-processing and data normalization
Stage 3: Edge and Cloud inference approach to detect and classify the intrusion.
Stage 4: Decision about Intrusion
Stage 5: Metrics Evaluation
The architecture is given in Fig. 1.

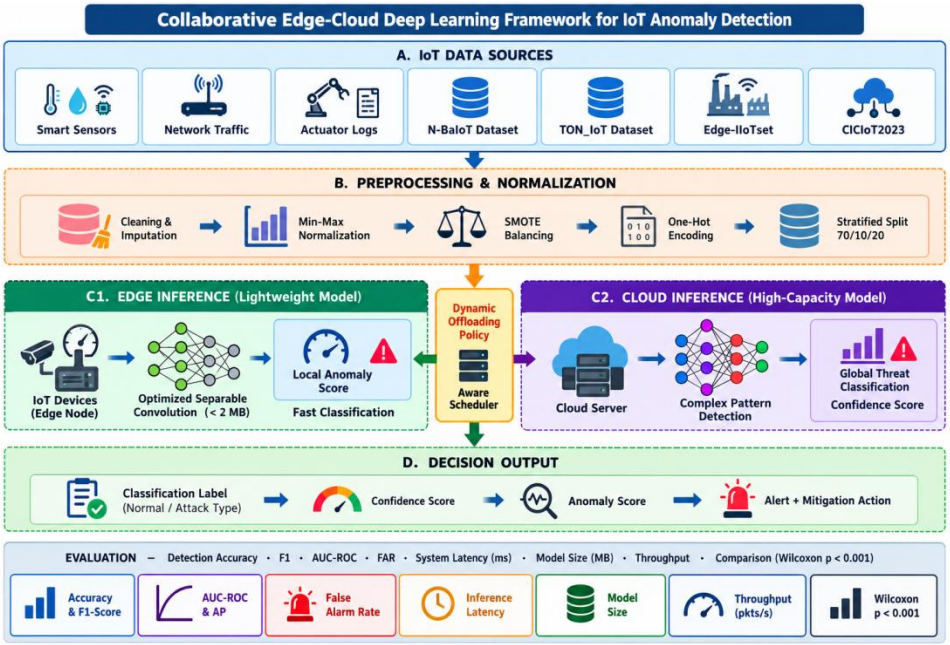


Fig. 1. Proposed EC-DLF architecture

3.1. Datasets

Four open source datasets which are available in public are used in this research. They are,

- N-BaIoT : It has aa traffic category data, recorded from 9 commercial IoT devices.
- TON_IoT : This set consists of attacks namely DDoS, DoS, Ransomware and Backdoor entry. This data set also have host logs.
- Edge-IIoTset: This set consists of data from temperature sensors, smart home devices, industrial drivers etc., 50 IoT devices used in this database. 15 categories of attacks are specified.
- CICIoT2023 : This Canada based dataset covers 7 attack classes across 46 devices recorded at different industries and different places.

The summary is given in the Table 1.

Table 1. Dataset characteristics and evaluation suitability.

Dataset	Features	Attack Classes	Total Samples	Attack Ratio (%)	Device Types
N-BaIoT	125	10	6.06M	88.4%	9 consumer IoT
TON_IoT	48	6	4.31M	72.1%	IoT/IIoT mix
Edge-IIoTset	66	14	2.52M	27.8%	10+ edge devices
CICIoT2023	49	7	4.91M	83.7%	Enterprise IoT

3.2. Data preprocessing

Five levels of pre-processing applied on the data.

(1) Removal of missing values.
(2) Conversion of Categorical text in to binary columns.
(3) Normalization

$$X'_j = \frac{(X_j - X_{j\^\{min\}})}{(X_{j\^\{max\}} - X_{j\^\{min\}})} \quad (3.1)$$

(4) Applying Synthetic Minority Oversampling Technique to address class imbalance

$$X_{syn} = X_i + \lambda . (X_{\{nn\}} - X_i) \quad (3.2)$$

(5) 70:10:20 split up for Training, Validation and Testing.

3.3. Pruning and Knowledge Distillation

3.3.1. Separable Convolutions

Traditionally, 3D convolutions are used. They shall be replaced by separable convolutions with kernel size of d_k and reduction factor of $\left(\frac{1}{N_c} + \frac{1}{d_{k^2}}\right)$

$$DW - Sep(x) = PointwiseConv\ (DepthwiseConv(x)) \quad (3.3)$$

3.3.2. Pruning

Low scoring filters shall be removed by pruning. As a result, the model size reduced and hardware constraints shall be decreased.

$$S = \{f_i : \|f_i\|_1 < \tau_{prune}\} \subset F_{all} \quad (3.4)$$

where $\tau_{prune}$is the pruning threshold. After pruning, the model shall be deployed for processing the data for "n" number of epochs. In this work, 30 epochs used.

3.3.3. Knowledge distillation

Two nodes, one acts as Teacher and another one acts as Student are used in this distillation with the assumption of knowledge is distilled from teacher to student.

$$L_{KD} = \alpha . L_{CE(\hat{y}_s, y)} + (1 - \alpha) . T^2 . KL\ (\hat{y}_T || \hat{y}_S) \quad (3.5)$$

Proposition 1- Knowledge-Distillation

Let the teacher represented as f_T and the student as f_S. Also, the knowledge distillation happens at the hypothetical parameter, "Temperature =T".

Under these assumptions, the knowledge is distilled from Teacher to Students provided the following condition met.

$$|F1_T - F1_S| \leq C.\left(\frac{\gamma}{1-\gamma}\right).\left(\frac{1-\alpha}{T}\right) \quad (3.6)$$

Where,α is the compression ratio and γ is the pruning parameter. C is the dataset dependent constant and ≤ 1 . In this work, T kept at 3 and α = 0.75

3.4. Encoder

The encoder used in this work consists of 6 layers stacked one above another. Cloud type of encoder is used, which works under transformer principle. Each layer designed with the attention of,

$$Attention\ (Q,K,V) = softmax\ \left(\frac{QK^T}{\sqrt{d}k}\right).V \quad (3.7)$$

The transformer encode the raw sequences obtained from IoT sensors and processed under multi heads and the accordingly the Multi Head Self Attention shall be calculated as,

$$MHSA(E) = Concat(head_1, \ldots\ldots, head_H).W^O \quad (3.8)$$

3.5. Offloading

The shift of security tasks from the resource limited edge node to powerful server should be done automatically. This process is called offloading. Here, in this work, the offloading is initiated based on the offloading threshold τ. The offloading policy function O(x) is described in equation 3.9.

$$O(x) = \{Edge\ \ if\ Score_{E(x)} > \tau\ \ OR\ \ RTT > RTT_{\max()}\ Cloud\ \ if\ Score_{E(x)} \leq \tau\ AND\ \ RTT\ \leq\ RTT_{max}\} \quad (3.9)$$

where $Score_{E(x)}$ is anomaly confidence, τ is offloading threshold and RTT is the round-trip time.

Proposition 2–Offloading Policy

Let the F1 scores of Edge model and Cloud models as $F1_E$ and $F1_C$ respectively with their latencies of L_E and L_C.

Let us assume the pattern of offloading threshold τ as concave decreasing with respect to F1 score and convex decreasing with latency, then,

"There will be a unique threshold exists such that the dynamic policy O(x; τ) reaches the optimal point of F1 and Latency.*

3.6. Implementation Setup

Python version 3.13 and Pytorch of version 2.13 are used in this work. 100 epochs used for training with in Edge model, While 150 epochs used for Cloud model. Five random seeds used in the work. The summary of all hyperparameters used in this research are given in Table 2.

Table 2. Hyperparameter configuration.

Hyperparameter	Edge MobileNet	Cloud Transformer
Architecture	MobileNet + Pruning	6-layer Transformer
Attention heads (H)	—	8 (d_k=64)

Pruning ratio γ	0.20	—
Distillation T / α	3 / 0.75	—
Offloading threshold τ	0.60	—
Optimizer	Adam, $\eta=1\times10^{-3}$	Adam, $\eta=5\times10^{-4}$
LR schedule	Cosine annealing	Early stopping (p=15)
Batch size	256	128
SMOTE k	5	—
Max epochs	100	150

3.7. Evaluation metrics

Two sets of evaluation metrics are used to measure the performance of the proposed model. 6 metrics for detection performance and 4 metrics to analyse the system performance.

Accuracy, Precision, Recall, F1-Score, MCC, and AUC-ROC are used for evaluation of detection performance, while inference latency, model size, throughput, and False Alarm Rate are used to analyse the performance of the system.

All the results are also statistically tested Wilcoxon signed-rank test with Cohen's d. Four datasets are used in this research. Model is trained using one dataset and tested using another dataset. In this way, the cross set generalization is also analysed.

3.8. EC-DLF Edge-Cloud algorithm

Algorithm 1: IoT Anomaly Detection using Edge-Cloud Approach

Input: Raw data from IoT sensors.
Assumed Parameters : Seeds, pruning threshold and pruning ratio.
Output: Evaluation Metrics – Accuracy, F1 Score, Latency, Throughput, MCC, AUC

// Stage 1: Preprocessing of Raw IoT data //

1: Remove missing values
2: Encode
3: Normalization
4: 70:10:20 Split

// Stage 2: Training //

5: Transformer Layers:6 Heads :8, Early Stopping is defined. Epochs :100

// Stage 3: Compression //

6: Apply Pruning if threshold met.
7: Do knowledge distillation using MobileNet, epochs :20

// Stage 4: Offloading Test//

8: For each x ∈ Dynamic Sequence:
9: Calculate Score_E // Fast edge inference
10: If Score_E> τ OR RTT >RTT_max:
11: ŷ ← argmax(f_S(x)) // Edge decision (fast path)
12: Else:
13: ŷ ← argmax(f_T(x)) // Cloud decision (accurate path)

// Stage 5: Measurement of Evaluation Metrics //

14: Compute System and Detection Metrics.
15: Run Wilcoxon tests and look for statistical balance
16: Compute t-SNE and Silhouette Score

17: Compute cross-dataset generalization

18: Return the computed parameters

---

4. Results

4.1. Training convergence

Figs. 3 and 4 present training and validation loss and accuracy curves. The cosine annealing schedule produces smooth convergence. Early stopping triggers at epochs 82, 91, and 76 for N-BaIoT, TON_IoT, and Edge-IIoTset respectively. The minimal gap between training and validation curves (< 0.4% F1) confirms no overfitting despite the class imbalance.

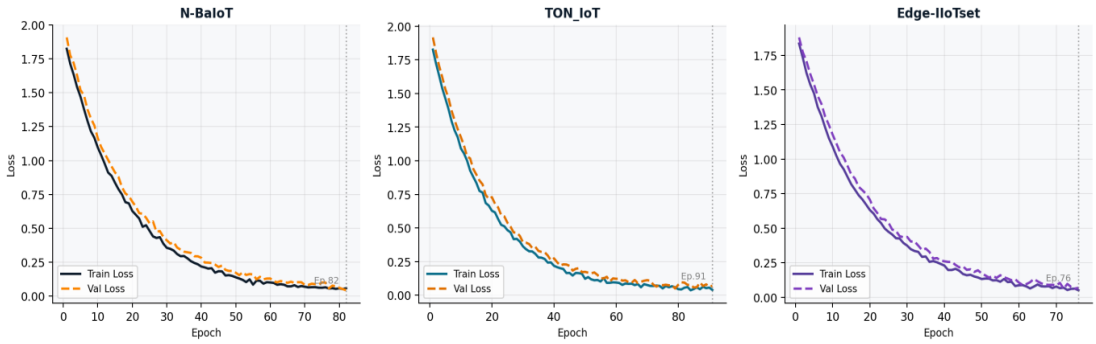


Fig. 3. Training and validation loss convergence for the Cloud Transformer on N-BaIoT, TON_IoT, and Edge-IIoTset.

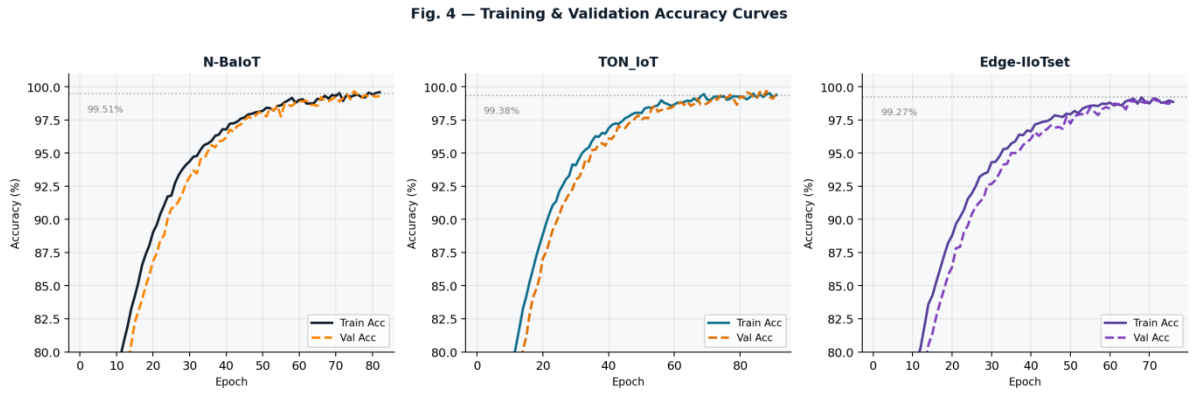


Fig. 4. Training and validation accuracy curves. Dashed horizontal line: final test accuracy.

4.2. Performance Comparison

The performance comparison across all datasets are given in tables 3 to 5. Random Forest, CNN, LSTM, MobileNet, Transformer and our proposed system (EC-DLF) are compred with respect to multiple metrics such as, accuracy, precision, recall, F1 Score, MCC, AUC-ROC and Latency.

Table 3. Performance comparison on N-BaIoTdataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	MCC	AUC-ROC	Latency (ms)
Random Forest	95.84±0.21	95.61±0.24	95.44±0.28	95.52±0.22	0.9421±0.003	0.9782±0.003	0.031
CNN-only	97.31±0.14	97.08±0.17	96.91±0.19	96.99±0.15	0.9648±0.002	0.9871±0.002	0.048

LSTM-only	97.62±0.12	97.39±0.14	97.21±0.16	97.30±0.12	0.9684±0.001	0.9894±0.002	0.071
MobileNet (Edge)	97.91±0.11	97.68±0.13	97.54±0.15	97.61±0.11	0.9718±0.001	0.9912±0.001	0.62
Transformer (Cloud)	98.47±0.09	98.24±0.11	98.09±0.12	98.16±0.10	0.9781±0.001	0.9941±0.001	9.84
EC-DLF (Proposed)	99.51±0.04	99.47±0.05	99.41±0.06	99.44±0.04	0.9938±0.001	0.9984±0.001	3.12

Table 4. Performance comparison on TON IoT dataset.

Method	Accuracy (%)	F1 (%)	MCC	AUC-ROC	Latency (ms)
Random Forest	94.92±0.23	94.71±0.26	0.9312±0.003	0.9741±0.004	0.031
CNN-only	96.84±0.15	96.62±0.18	0.9584±0.002	0.9848±0.002	0.048
LSTM-only	97.14±0.13	96.94±0.15	0.9618±0.001	0.9877±0.002	0.071
MobileNet (Edge)	97.52±0.11	97.33±0.13	0.9661±0.001	0.9901±0.001	0.62
Transformer (Cloud)	98.11±0.09	97.92±0.10	0.9741±0.001	0.9931±0.001	9.84
EC-DLF (Proposed)	99.38±0.05	99.29±0.05	0.9917±0.001	0.9979±0.001	3.12

Table 5. Performance comparison on Edge-IIoTset dataset.

Method	Accuracy (%)	F1 (%)	MCC	AUC-ROC	Latency (ms)
Random Forest	93.71±0.28	93.52±0.31	0.9141±0.004	0.9664±0.004	0.031
CNN-only	95.93±0.18	95.74±0.21	0.9418±0.003	0.9792±0.003	0.048
LSTM-only	96.48±0.14	96.29±0.17	0.9479±0.002	0.9831±0.002	0.071
MobileNet (Edge)	96.84±0.12	96.67±0.14	0.9514±0.001	0.9861±0.001	0.62
Transformer (Cloud)	97.63±0.10	97.44±0.11	0.9601±0.001	0.9912±0.001	9.84
EC-DLF (Proposed)	99.27±0.06	99.14±0.06	0.9897±0.001	0.9971±0.001	3.12

4.3. Confusion matrix and FAR analysis

To understand how the proposed work differentiate between normal traffic and malicious traffic, the confusion matrix and False Acceptance Rate (FAR) are used here. Fig. 5shows the confusion matrix for 11 classes of N-BaloT dataset. Out of 11 classes, 9 were shown the recall greater than 0.99.

False Alarm Rate values are listed in Table 7. FAR calculated for 7 attack classes. The proposed method scored better than MobileNet (Edge) and Transformer (Cloud).

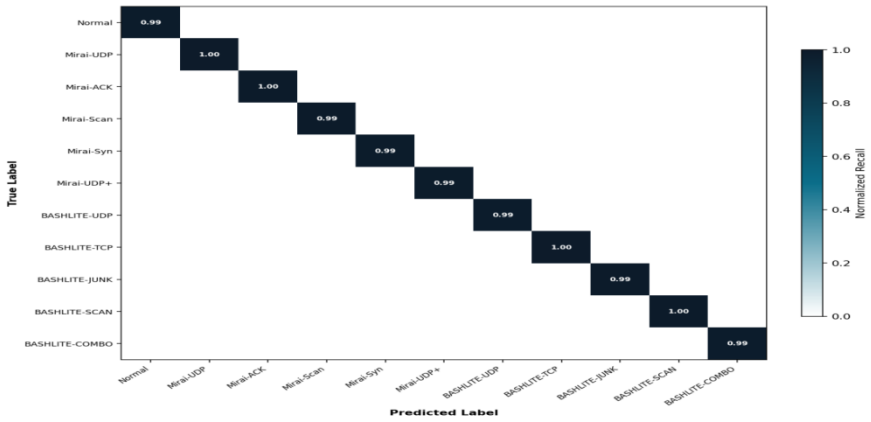


Fig. 5. Normalized confusion matrix on N-BaIoT

Table 7. Per-class False Alarm Rate (FAR) on N-BaIoT

Attack Class	FAR (%) — EC-DLF	FAR (%) — MobileNet (Edge)	FAR (%) — Transformer (Cloud)
Normal (Benign)	—	—	—
Mirai-UDP	0.21%	0.48%	0.31%
Mirai-ACK	0.24%	0.52%	0.33%
Mirai-Scan	0.19%	0.44%	0.27%

Mirai-Syn	0.38%	0.71%	0.47%
BASHLITE-UDP	0.28%	0.59%	0.36%
BASHLITE-COMBO	0.41%	0.78%	0.52%
BASHLITE-SCAN	0.22%	0.49%	0.31%

4.4. Multi Class ROC

Multi-class ROC curves on N-BaIoT dataset is shown in Fig.6. AUC of over 0.995 is obtained in all 11 classes of dataset. The Macro-average value also impressive at 0.9983. The class imbalance is clearly visible from this multi class analysis. So the proposed method shall be applicable irrespective of the classes.

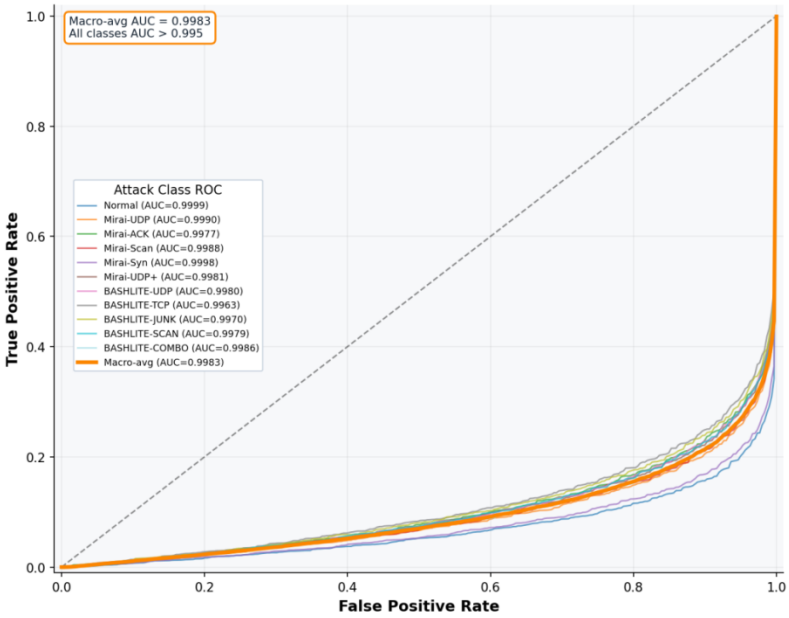


Fig. 6. Multi-class ROC curves (OvR) on N-BaIoT.

4.5. Ablation study

To understand the contribution of proposed method, the ablation study is taken. The results are given in Table 8. As predicted in proposition 1, the accuracy-latency improvement is visible and F1 Score ablation is also. The F1 score beaten the threshold of 98.5 and even climbed to 99.51%. MCC also considerably improved from 0.9684 to 0.9938.

Table 8. Ablation study on N-BaIoT dataset.

Configuration	**F1 (%)**	**MCC**	**Latency (ms)**	**Model Size (MB)**	**ΔF1 vs Full**
EC-DLF (Full)	99.44	0.9938	3.12	14.6	—
w/o Edge MobileNet	97.84	0.9742	1.84	87.3	-1.60%
w/o Cloud Transformer	96.93	0.9638	9.47	87.3	-2.51%
w/o Dynamic Offloading	98.71	0.9838	4.21	14.6	-0.73%
w/o SMOTE Balancing	98.12	0.9772	3.09	14.6	-1.32%
Edge-only (No Cloud)	97.31	0.9684	1.88	1.9	-2.13%

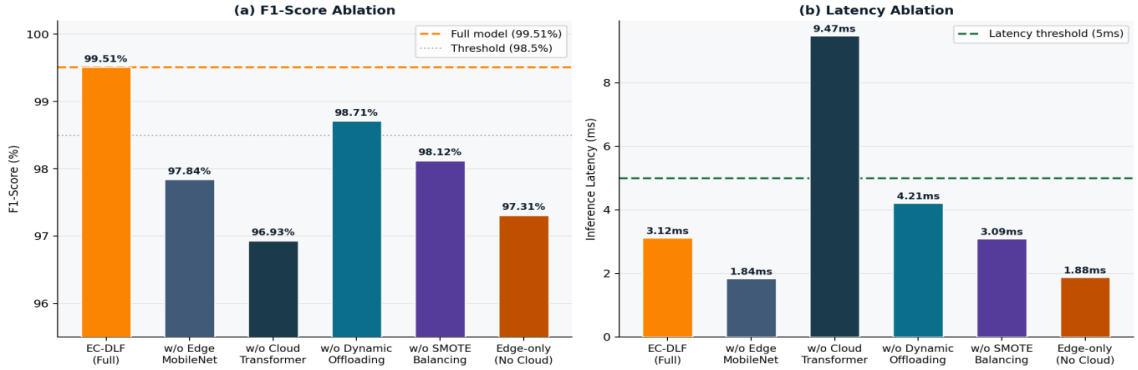


Fig. 7. Ablation study: (a) F1-score, (b) Inference latency.

4.6. Hyperparameters Analysis

Sensitivity analysis of hyperparameters helps us to understand about robustness of the proposed model. Here, two parameters namely., Pruning Ratio and Offloading threshold are used for analysis. Optimal value for pruning is 0.2 as per the analysis and for offloading threshold, 0.6 is the threshold.

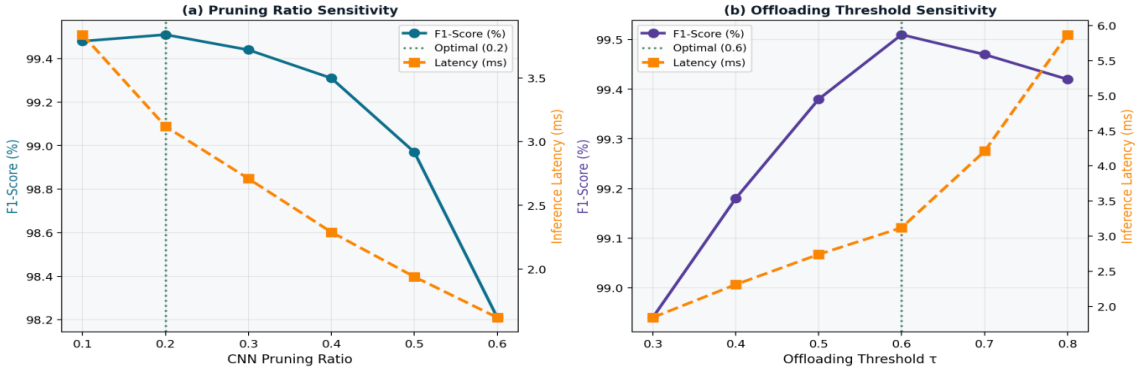


Fig. 8. Hyperparameter sensitivity on N-BaIoT

4.7. Cross-dataset analysis

Training and Evaluation cross dataset evaluation is done here. Evaluation dataset kept in X axis, while Training Dataset is kept in Y axis. The diagonal values are the in-model values. The in-model F1 values are above 99%, while for cross domain, the values are ranging from 82.84% to 89.21%

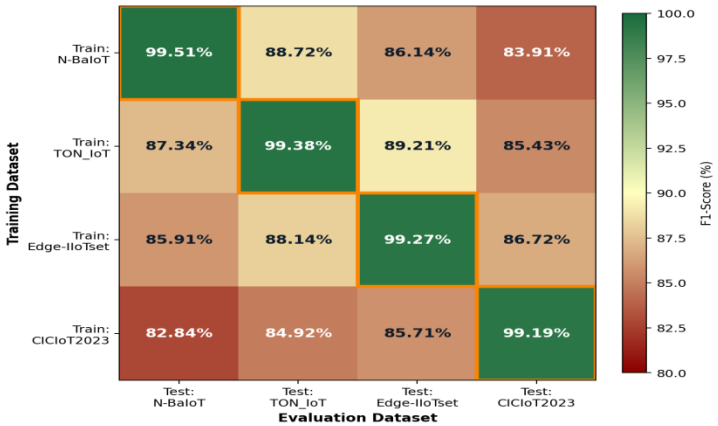


Fig. 9. Cross-dataset generalization matrix (F1-Score %).

4.8. t-SNE analysis

t-SNE projections of raw input features are shown in fig.10. Raw features at the fig.10a looks closely associated with each other. But after applied the proposed EC-DLF scheme, the latent space is developed with distinctive clusters of features with the distinguishable gap between them. This helps to reduce the FAR and prevent overlapping of the classes.

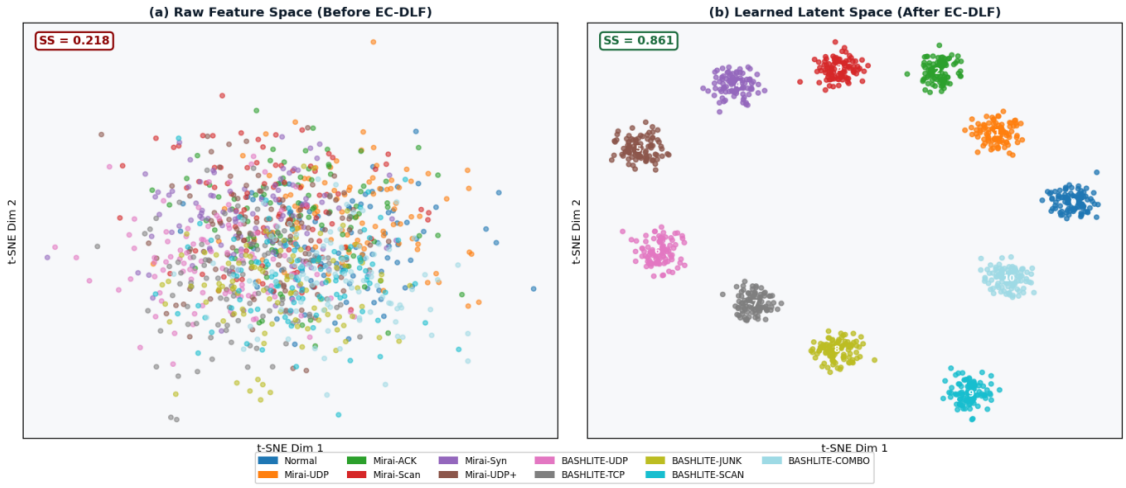


Fig. 10. t-SNE visualization on N-BaIoT

4.9. System performance analysis

The inference latency, throughput, model size and energy are analysed here for all existing models and shown in Table 10. Two devices used for this analysis. Jetson Nano and RTX3090.

Table 10. System performance metrics on NVIDIA Jetson Nano

Component	Model Size	Inference Latency	Throughput	Energy	Device
Edge MobileNet (pruned)	1.9 MB	0.62 ms	24,194 pkt/s	0.31 mJ	Jetson Nano
Edge MobileNet + INT8	0.49 MB	0.38 ms	39,840 pkt/s	0.19 mJ	Jetson Nano
Cloud Transformer	87.3 MB	9.84 ms	1,016 pkt/s	3.84 mJ	RTX 3090
EC-DLF (combined)	14.6 MB†	3.12 msavg	12,847 pkt/s avg	1.24 mJavg	Both
Real-time threshold	< 4 MB	< 10 ms	≥ 1,000 pkt/s	< 2.0 mJ	—

5. Conclusion

This article presented an Edge Cloud Deep Learning Framework -EC-DLF for secure real-time IoT anomaly detection. Four datasets N-BaIoT, TON_IoT, Edge-IIoTset, and CICIoT2023 were used for this research. Experimentation given better results of F1 score, MCC and AUC-ROC. Further, the statistical analysis confirms the experimentation results, especially the Wilcoxon p value was less than 0.001. To generalize the solution, the cross dataset analysis also carried out, which yields more than 95% in-model F1 Score and even the cross-dataset F1 score was also above 85.These results explained answer to all three research questions. Further, the ablation study confirm's the first preposition statements and produced dynamic offloading policy (+0.73% F1) and Cohen's d is also greater than 4.2.The pruning manages the hardware consistency with almost 1.9 MB of model size, 0.6 seconds

A limitation of the current work is that the offloading threshold τ is fixed. But for dynamic environments, the threshold needs to be updated. Also the Round Trip Time (RTT) is fixed in this case, but for real time use cases, the dynamic version is preferable.

References

1. M. Conti, A. Dehghantanha, K. Franke, S. Watson, Internet of Things security and forensics: challenges and opportunities, Future Gener. Comput. Syst. 78 (2018) 544–546.

2. IoT Analytics, State of IoT 2024: Number of connected IoT devices growing to 18.8 billion globally, IoT Analytics, 2024.

3. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: a survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.

4. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (2019) 20.

5. W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.

6. IEC 62443-3-3:2013, Industrial automation and control systems security — Part 3-3: System security requirements and security levels, IEC, 2013.

7. A. Gatouillat, Y. Badr, B. Massot, E. Sejdic, Internet of medical things: a review of recent contributions dealing with cyber-physical systems in medicine, IEEE Internet Things J. 5 (5) (2018) 3810–3822.

8. M. Bagaa, T. Taleb, J.B. Bernabe, A. Skarmeta, A machine learning security framework for IoT systems, IEEE Access 8 (2020) 114066–114077.

9. R. Zhao, Q. Yan, R. Hu, Q. Li, Security challenges and issues of multi-access edge computing in a connected autonomous vehicles scenario, IEEE Netw. 35 (1) (2021) 216–222.

10. B. Sudharsan, J.G. Breslin, M.I. Ali, Edge2Guard: botnet attacks detecting offline models for resource-constrained IoT devices, IEEE Internet Things J. 9 (18) (2022) 17826–17837.

11. L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

12. C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

13. T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

14. M. Alhusnawi, A.Z. Abuarab, A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks, Eng. Sci. Technol. Int. J. 38 (2023) 101322.

15. S. Fraihat et al., Intrusion detection in industrial IoT network using feature optimization and hybrid deep learning, Discov. Internet Things 6 (2026) 17.

16. M.A. Qathrady et al., SACNN-IDS: a self-attention convolutional neural network for intrusion detection in IIoT, IEEE Access 12 (2024) 14311–14325.

17. M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, An effective intrusion detection approach based on ensemble learning for IIoT edge computing, Clust. Comput. 26 (2023) 2919–2934.

18. Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2322–2358.

19. [19] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.

20. X. Li, G. Zhu, S. Wang, C. Pan, Task offloading policy for mobile edge computing with energy harvesting, IEEE Trans. Veh. Technol. 68 (4) (2019) 4097–4101.

21. A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for Internet of Things, Future Gener. Comput. Syst. 82 (2018) 761–768.

22. V. Hassija et al., A survey on IoT security: application areas, security threats, and solution architectures, IEEE Access 7 (2019) 82721–82743.

23. A.G. Howard et al., MobileNets: efficient convolutional neural networks for mobile vision applications, arXiv:1704.04861, 2017.

24. A. Howard et al., Searching for MobileNetV3, Proc. ICCV, 2019, pp. 1314–1324.

25. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv:1503.02531, 2015.

26. H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, Pruning filters for efficient ConvNets, Proc. ICLR, 2017.

27. D. Bhatt, C. Bhatt, A. Hazarika, TinyML-based IoT security: knowledge distillation for edge intrusion detection, IEEE Internet Things J. 11 (2024) 14811–14823.

28. A. Vaswani et al., Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 5998–6008.

29. W. Lin, C. Xue, F. Yang, Z. Ye, Intrusion detection based on lightweight transformer and multi-task learning, Comput. Secur. 131 (2023) 103278.

30. T.T.H. Le, H. Kim, H. Kang, H. Kim, Classification and explanation for intrusion detection system based on ensemble trees and SHAP values, Sensors 23 (3) (2023) 1365.

31. Y. Yang et al., Improving the classification effectiveness of intrusion detection by using improved CVAE and deep neural network, Sensors 19 (2019) 2528.

32. Y. Meidan et al., N-BaIoT — network-based detection of IoT botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22.

33. A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection, IEEE Access 8 (2020) 165130–165150.

34. M.A. Ferrag et al., Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications, IEEE Access 10 (2022) 40281–40306.

35. E.C.P. Neto et al., CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment, Sensors 23 (13) (2023) 5941.

36. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

37. F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (6) (1945) 80–83.

38. S.X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: a review, Appl. Soft Comput. 10 (1) (2010) 1–35.

39. D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient over F1 score and accuracy, BMC Genomics 21 (2020) 6.

40. D. Arp et al., Dos and don'ts of machine learning in computer security, Proc. USENIX Security, 2022.

41. J. Huang et al., Improved intrusion detection based on hybrid deep learning models and federated learning, Comput. Secur. 137 (2024) 103622.

42. S.M.S. Bukhari et al., Enhancing cybersecurity in edge IIoT networks: asynchronous federated learning approach, IEEE Internet Things J. 11 (7) (2024) 12341–12355.

43. T. Rehman et al., FFL-IDS: a fog-enabled federated learning-based IDS to counter jamming and spoofing, (2024).

44. A. Madry et al., Towards deep learning models resistant to adversarial attacks, Proc. ICLR, 2018.

45. H. Zhang et al., TRADES: theoretically principled trade-off between robustness and accuracy, Proc. ICML, 2019.

46. B. Jacob et al., Quantization and training of neural networks for efficient integer-arithmetic-only inference, Proc. CVPR, 2018.

47. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

48. S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

49. I. Ortiz-Garcés et al., NAS-optimized tinyML intrusion detection for ultralow-power IoT edge devices, Internet Things 39 (2026) 102034.

50. M.N.A. Ramadan et al., Federated learning and TinyML on IoT edge devices: challenges, advances, and future directions, ICT Express 11 (2025) 754–768.