Advances in Artificial Intelligence and Machine Learning; Research 6 (5) 6433-6450 Received 30-06-2026; Accepted 18-10-2026; Published online 25-10-2026

A Framework for Effective Intrusion Detection and Mitigation at the Edge of Things Using Deep Learning Architectures

Vasavi Sravanthi Balusa vasavibvs@klh.edu.in
Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation,
Hyderabad-500075, Telangana, India.

K Srinivas srirecw9@klh.edu.in
Department of Computer Science and Engineering,
Koneru Lakshmaiah Education Foundation,
Hyderabad-500075, Telangana, India.

Corresponding Author: Vasavi Sravanthi Balusa

Copyright © 2026 Vasavi Sravanthi Balusa and K Srinivas. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Edge computing lets data and tasks run close to where people use them. That usually speeds up response and can cut down on bandwidth. Still, the Edge of Things raises privacy and security issues. These issues show up across the layers of an edge setup. As a result, it becomes harder to do intrusion detection in systems that are not centralized. Current intrusion detection systems (IDS) face issues with complexity and accuracy, rendering them unsuitable for resource-limited edge devices. To address this, an efficient lightweight deep learning-based IDS is proposed for EC to safeguard EoT devices. The framework extracts key network features using MNasNet and the elastic collision seeker optimization algorithm (ESOA), followed by SqueezeNet for intrusion detection and classification. A defensive module restores network connectivity post-attack, while the Auxiliary Classifier Generative Adversarial Network (ACGAN) approach effectively mitigates data imbalance. Validation with Edge-IIoT and IoT-23 datasets shows the system excels in identifying diverse attack types in the EC environment.

Keywords: Deep learning, Edge computing, Edge of Things, MNasNet, SqueezeNet

1. INTRODUCTION

Although the cloud is an advanced data storage and processing system, it can sometimes fail to meet the performance requirements for new applications and IoT integration [1, 2]. Edge Computing has entered the picture to assist cloud and IoT [3, 4]. It stores some data locally where it is generated. This can help to reduce the time lag and make the response seem more immediate. It also indicates a reduced dependence on the primary cloud [5–9]. Furthermore, it could be used to secure user sharing of content [10–15]. There are some areas subject to weaknesses in edge systems, despite these advantages [16, 17]. Relationships can be volatile. Many sites also have more difficult security

Citation: Vasavi Sravanthi Balusa and K Srinivas. A Framework for Effective Intrusion Detection and Mitigation at the Edge of Things Using Deep Learning Architectures. **Advances in Artificial Intelligence and Machine Learning**. 2026;6(5):355. https://dx.doi.org/10.54364/AAIML.2026.65355

to manage [18]. That's why attackers might resort to attacks such as Denial of Service [19]. Others may also look for ways to move from user access up to higher privileges [20]. Safeguarding these systems with effective Intrusion Detection Systems (IDS) is critical, but achieving this is hindered by the absence of international standards and the computational intensity of current anomaly detection methods.

The important contributions of the paper are:

An Edge-Intelligent IDS Framework: We present a novel IDS that combines class-balancing generative models (ACGAN), metaheuristic feature selection (ESOA), and lightweight deep learning (MnasNet) that is designed for edge computing scenarios. In contrast to previous IDS methods, our framework maintains computational efficiency on edge devices with limited resources while achieving high detection performance.

Validated Impact of Feature Selection: The ESOA-based feature selection method demonstrated its efficacy in improving the speed and accuracy of IDS systems by reducing input dimensionality and producing significant improvements in performance, increasing detection accuracy by 1.49% on Edge-IIoT and 2.56% on IoT-23. It demonstrates the crucial role of ESOA in scalable, effective IDS design.

Integrated Attack Mitigation Mechanism: In addition to detection, our proposed framework includes a proactive attack mitigation module that automatically restores the network to normal after an attack. It addresses the difficulty of guaranteeing uninterrupted functioning in real-time edge situations, offering a novel viewpoint in IDS research.

Improved Generalization via Synthetic Data: We addressed the persistent issue of class imbalance in intrusion detection by implementing ACGAN-based synthetic data generation. This technique significantly increased the model's resistance to uncommon and new attacks, enhancing its generalizability in various IoT contexts.

Empirical Validation and Benchmark Superiority: The Edge-IIoT and IoT-23 real-world intrusion datasets were used to test the suggested framework and produced state-of-the-art results, with accuracy rates of 99.44% and 99.40%, respectively. These findings demonstrate the framework's superiority over current IDS methodologies and highlight its applicability in edge security research from a practical and scientific perspective.

Remaining of this paper is structured as follows. Summary of the current methods for intrusion detection is provided in Section 2. Our suggested intrusion detection technique is presented in Section 3 and 4 mentioned the system configuration, the efficacy of the suggested work, the outcomes of the experiment, and an analysis of our findings against those of other competing techniques. Section 5 concludes with our findings and our plans for further research.

2. LITERATURE REVIEW:

Using ML approaches, Guezzaz et al. (2022) [21] described a hybrid intrusion detection approach for Edge-Based IIoT Security which combines K-NN, PCA methods for identifying misuse and

anomalies. PCA extracts key features, enhancing detection accuracy via K-NN, assessed with Bot-IoT and NSL-KDD datasets for performance metrics.

Singh et al. (2022) [22] offered a hybrid IDS using Machine Learning was developed for mobile edge computing, incorporating signature, anomaly, and hybrid components with different classifiers. It scored 90.25% accuracy and a false alarm rate of 1.1%.

Bangui and Buhnova (2022) [23] provided a hybrid methodology for attack identification in Edge Computing was presented, utilizing Fuzzy C-Means for ambiguous data and imbalanced datasets, supplemented by ant colony and ant lion optimization to enhance classification performance, with Deep Forest used to minimize false positives.

Hoang et al. (2023) [24] introduced a distributed ML model utilizing a modified voting method was created to identify abnormalities in Edge networks. It featured a partition learning model for intrusion detection systems (IDS) on edge devices, allowing autonomous decisions. Outputs were relayed to the server, while MultiLayer Perceptron, SVM, and Random Forest algorithms operated on the devices. A refined voting algorithm on the server improved attack detection accuracy, evaluated using the UNSW-NB15 database.

Fenanir and Semchedine (2023) [25] utilized Three deep learning algorithms—LSTM, CNN, and DNN—were applied in federated learning for IoT networks, using N-BaIoT, IoT-23, and IoTID20 datasets, with LSTM outperforming the others in accuracy.

Offering an enhanced IDS model for categorizing attacks on EoT networks was the goal of Kumar et al. (2023) [26]. The combination-based grouping method was employed in this study to pick class features optimally. The filter-based feature selection method was then used for the best-reduced features. Ultimately, the random forest classifier was used to detect intrusions based on the attributes that have been chosen. The UNSW-NB15 dataset was used for the investigation.

3. PROPOSED METHODOLOGY

The presented edge computing Deep Learning-based IDS consists of pre-processing, data augmentation, feature extraction, and intrusion categorization. It uses ACGAN for data balance, MNasNet for feature extraction, and ESOA for selection, with SqueezeNet classifying intrusions. A mitigation module enhances defense against network threats. The overall system architecture of the proposed framework is given in FIGURE 1, and each component of the proposed framework is briefly explained in the following subsections.

3.1 Data Pre-Processing:

Data preparation involved replacing missing values with the column mean, one-hot encoding categorical data, and applying Min-Max normalization for scaling features. Equation (1) illustrates how this method scales each feature in our dataset between 1 and 0, converting each feature's between 1

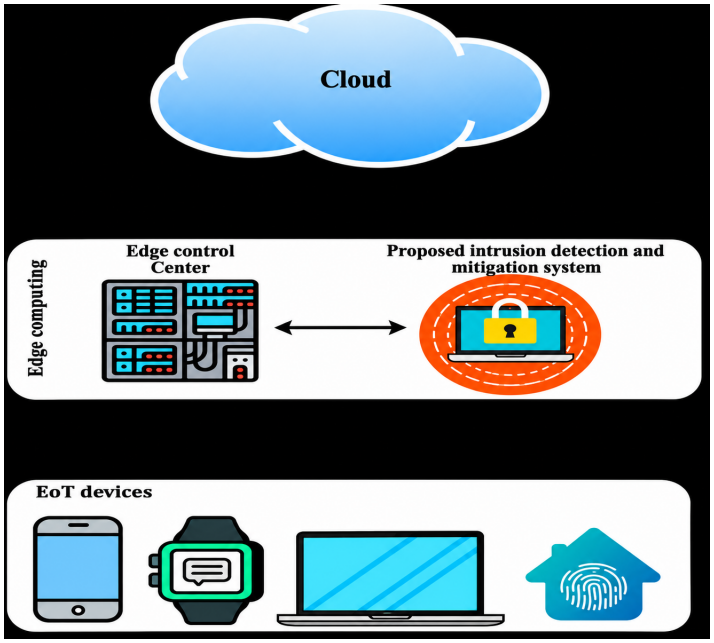

Figure 1: System architecture

and 0.

$$S' = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \tag{1}$$

In this case, S denotes the feature that requires normalization, while Smax and Smin represent the highest and least values, respectively.

3.2 Data Augmentation Using ACGAN:

We employed ACGAN for synthetic data augmentation, enhancing underrepresented classes like theft and reconnaissance, while balancing data distribution by removing records from larger categories.

The discriminator D in ACGAN generates a range of probabilities over class labels and sources, P(SR|X), P(CS|X)=D(X), based on the false data generated by the generator Xfake=G(c,z). Equations (2) and (3) represent the functioning of the ACGAN as the LCS and LSE).

$$LSE = E\left[\log\ P(SR = real\,|X_{real})\right] + E\left[\log\ P(SR = fake\,|X_{fake})\right] \tag{2}$$

$$LCS = E\left[\log\ P(CS = cs\,|X_{real})\right] + E\left[\log\ P(CS = cs\,|X_{fake})\right] \tag{3}$$

Training has been done on the discriminator and generator to maximize LSE + LCS and LCS-LSE, respectively.

3.3 MNasNet-Based Feature Extraction:

Feature extraction is vital for intrusion classification, utilizing an MNasNet-based network suitable for resource-limited devices. The main goal is to cut down inference time, but keep both speed and accuracy. MNasNet uses a small number of parameters and still performs well on recognition tasks. It relies on SepConv and MBConv blocks to pull out detailed features without much extra cost.

Equations (4) and (5) illustrate the MnasNet objective function, where the weight factor is represented by y; the inference time of the target is denoted by T; target model's real inference time on the embedded device is denoted by Lat(n); and target model's accuracy is denoted by Acc(n).In this case, $y = a$ indicates that the goal inference time T is longer than the real inference time (Lat(n)); otherwise, it is y = b. When $a = b = -0.07$, MnasNet can achieve a respectable trade-off between speed and accuracy.

$$\max_{n} Acc(n) \times \left[\frac{\text{Lat (n)}}{\text{T}}\right]^{y} \tag{4}$$

$$y = \begin{cases} \text{a , if Lat(n)} < \text{ T} \\ \text{b , otherwise} \end{cases} \tag{5}$$

Finally, the output of the last MBconv block was fed to the global average pooling layer, and the final layer of the network is a fully connected layer that produces the output.

3.4 ESOA-Based Feature Selection:

ESOA is an algorithm based on seeker optimization (SOA) designed to emulate the thorough search capabilities observed in human search behaviour. This optimization method has been fine-tuned to mimic the process of searching for the best answer, much like a group of explorers travelling through and investigating space. The ESOA comprises three distinct stages, each contributing to its efficacy in identifying and isolating the dataset's most relevant and valuable features.

Direction of the Search

The exploration direction is shaped by the experiential insights gained from an individual's efforts and the assessments made by another individual who has previously explored a similar area. From some dimensions, the individual's preemptive $\overrightarrow{f}_{t.p}(t)$, altruistic $\overrightarrow{f}_{t.a}(t)$, and egoistic paths $\overrightarrow{f}_{t.e}(t)$ are realized.

$$\overrightarrow{f}_{t.e}(t) = \overrightarrow{p}_{t,best} - \overrightarrow{x}_{t}(t) \tag{6}$$

$$\overrightarrow{f}_{t.a}(t) = \overrightarrow{g}_{t,best} - \overrightarrow{x}_{t}(t) \tag{7}$$

$$\overrightarrow{f}_{t.p}(t) = \overrightarrow{x}_{t1} - \overrightarrow{x}_{t2} \tag{8}$$

The seeker obtains the search orientation by applying the approach of an arbitrary weighted average.

$$\overrightarrow{f}_{i}(t) = sign(\omega \overrightarrow{f}_{i,p}(t) + \varphi_1 \overrightarrow{f}_{i,e}(t) + \varphi_2 \overrightarrow{f}_{i,a}(t)) \tag{9}$$

Here, $g_{i,best}$ denotes the neighbourhood's past ideal location, where the i^{th} seeking element was positioned; φ is denoted the arbitrary integer, which ranges from one to zero; ω denotes the inertia weight; $p_{i,best}$ indicates the best location in the i^{th} search factor for the current locality.

In those equations, $\overrightarrow{x}_i(t_2)$ and $\overrightarrow{x}_i(t_1)$ are the optimum advantages of $\overrightarrow{x}_i(t-2)$, $\overrightarrow{x}_i(t)$, $\overrightarrow{x}_i(t-1)$ each separately where $t_1, t_2 \in \{\mathrm{t},\ \mathrm{t}-1, \mathrm{t}-2\}$.

Step Size of the Search

The ESO is a representation of fuzzy estimate reasoning's capacity. When the method states a basic fuzzy rule, it adjusts to the most accurate estimation of the objectively optimized issue. Longer searching stages are linked to greater relevance, while shorter searching stages are related to lesser fitness. The Gaussian distribution function was modified to define the measurements of the search step.

$$\mu(a) = \mathrm{e}^{\frac{a^2}{2\delta^2}} \tag{10}$$

Here, the membership function parameters are represented by α and δ. The likelihood of a consequent variable above $[-3\delta, 3\delta]$ is less than 0.0111, according to Equation (10). Hence, $\mu\ min = 0.0111$. However, $\mu\ max$ is fixed at 0.9 in order to accelerate the speed of convergence and achieve an optimal person to take an undefined step size.

$$\mu(i) = \mu_{max} - \left(\frac{s - I_i}{s - I_{min}}\right)(\mu_{max} - \mu_{min}),\ i = 1, 2, \ldots s \tag{11}$$

$$\mu_{ij} = \mathrm{rand}(\mu_i), j = 1, 2, ..., D \tag{12}$$

In the above equations, μ_{ij} is defined. Here, Ii represents the sequence's number (t) in the current set of individuals, sorted from lower to higher function values. It denotes a real number within a specific range [μ,1]. It is understood that Equation (11) represents the haphazard way in which humans do searches. The following formula defines the j dimension step measurement when searching the interspace:

$$\alpha_{ij} = \delta_{ij} - \sqrt{-\ln(\mu_{\mathrm{ij}})} \tag{13}$$

Here, ln() denotes the natural logarithm, and the Gaussian distribution function's parameter is denoted as δ_{ij}, which is defined in the following Equation.

$$\delta_{ij} = \omega * abs\ (\overrightarrow{x}_{min} - \overrightarrow{x}_{max}) \tag{14}$$

$$\omega = \frac{\mathrm{iter}_{\max} - t}{\mathrm{iter}_{\max}} \tag{15}$$

Here, inertia weight is denoted as ω. As the evolutionary algebra gets better, ω goes from 0.9 to 0.1 in a linear fashion. $\overrightarrow{x}_{max}$ and $\overrightarrow{x}_{min}$ indicate the ranges of the function's maximum and minimum values.

Updates on each individual position

Equation (16) represents the revised location after the explorer's path and foot measurement have been determined.

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) f_{ij}(t),\ \ j = 1, 2, ..., D; ; i = 1, 2, ...s \tag{16}$$

Here, i denotes the i^{th} individual searching; j denotes the dimensional; searching step size and the searchers' path at time t is denoted as $a_{ij}(t)$ and $f_{ij}(t)$, respectively. Then, the searchers' site at the time (t + 1) and t are denoted as $x_{ij}(t+1)$ and $x_{ij}(t)$ respectively. The ESO-FS approach's mathematical model was created. Typically, classification tasks (like supervised learning) call for

datasets of size $NS \times NS$, where NS represents the number of samples and NF denotes the number of features. Selecting a subset of features S from the total number of features (NF) is the primary goal of the FS problem, where the size S is smaller than NF. It is obtained by reducing the following primary function:

$$Fit = \lambda \times \gamma_s + (1 - \lambda) \times (\frac{|S|}{N_F}) \tag{17}$$

Here, the classifier error using S and the total of selected features $|S|$, represented by γ_s. The purpose of γ is to maintain equilibrium between γ_s and $\left(\frac{|S|}{N_F}\right)$.

3.5 SqueezeNet Based Classifier:

SqueezeNet, a compact architecture, classifies intrusions efficiently using fire modules, batch normalization, and ReLu units for enhanced performance and speed.

$$f\{y\} = \sum_{fm1=1}^{FM} \sum_{c=1}^{c} w_c^f x_c^{fm1} \tag{18}$$

In Equation (18) Here, the letters FM stand for feature maps, c for the channel, w for the kernel, f{y} for squeeze process output, and x for the input value. The feature map of the class is reduced to a single value in this network via the global average pool. Ultimately, the multiclass outcome obtained from the softmax activation procedure depends on the features' probability value.

3.6 Attack Mitigation

The suggested framework starts a violation mitigation process after finding an attack. In response, the congestion control mechanism unit is turned on by the Edge Control Center. When a network device detects an attack, it triggers the termination of the connection and reallocates network resources to allowed hosts. As a result, authorized users can reaccess network resources, guaranteeing regular network operations will continue.

4. RESULT AND DISCUSSION

The result section discusses the experimental results of anomaly detection at edge computing layer using the proposed framework, execution scenarios, outcomes, and efficacy. Experiments utilized an i7-4702MQ PC, Python, TensorFlow, and datasets Edge-IIoT and IoT-23 with specific training settings.

4.1 Dataset Description:

Edge-IIoT dataset:

Edge-IIoT is a dataset of traffic from a range of IoT devices, such as sensors, for cybersecurity research. It comes with 2,219,201 traffic instances, of which 603,558 are malicious, in five threat categories, such as malware and DoS attacks.

IoT-23 dataset:

The IoT-23 dataset contains 20 network models (three benign, 20 malicious), 9 attack categories (three benign, 17 malicious) and 325,307,990 data records for research.

The proposed model used an **80:20 train and test split with random state 42 and** Adam optimizer with a learning rate of 0.0002 for 5 epochs and a latent dimension of 100 used for training ACGAN. MNASNet-based feature extraction used 96 convolutional filters with a batch size of 64. ESOA was configured with a population size of 50 and one optimization iteration. The SqueezeNet classifier was trained for 100 epochs with a batch size of 64 using the Adam optimizer (learning rate = 0.001) and Cross-Entropy loss. A StepLR scheduler with a step size of 10 and $\gamma = 0.1$ was employed for learning-rate adjustment and various hyper parameters were used like learning rate, batch size, population size, adam optimization, loss function, Standard and minmax scalar.

4.2 Assessment of Performance on Edge-IIoT dataset:

The proposed approach demonstrated its effectiveness for classification of particular attacks through tests on the dataset. There is a noted imbalance in the data set, resulting in a difference in detection rates. An ACGAN-based data augmentation technique proposed to increase the classifier performance, resulting in higher detection rate for both majority and minority incursions. The model performance is good with high F1 scores, sensitivities and accuracy scores for categories. The proposed algorithm can improve the detection rate of most of the incursions and significantly improves the accuracy of minority intrusion detection by using this method, as presented in TABLE 1.

TABLE 2 analyzes the Edge-IIoT dataset, revealing that RNNs perform poorly with 76.13% accuracy due to the vanishing gradient problem. Feed forward neural networks attain 88.26% accuracy but struggle with variable-length sequences, risking information loss. The proposed method excels in intrusion classification using MNasNet for efficient feature extraction in Edge Computing environments.

The integration of the hybrid approaches leads to efficient results in terms of all performance metrics and outperforms existing methods shown in FIGURE 2, and the classification results are displayed in the confusion matrix (FIGURE 3), highlighting the suggested framework's superiority. The actual and anticipated classes are shown in the confusion matrix's rows and columns, accordingly. The diagonal elements indicate the number of correctly identified samples when the estimated category matches the actual class. The data that is off-diagonal has been incorrectly classified. The diagonal elements are precious when compared to the off-diagonal elements. It implies that the model generates more accurate predictions and effectively predicts examples for their respective classes.

The ROC curve is an important indicator to assess the model's capacity for intrusion detection. The performance of the classifiers' classification in terms of true positive and false positive rates is a crucial criterion for choosing them. A threshold that steadily drops from one to zero is computed. The true positive and false positive rates for each threshold are equated during this process. The

Table 1: Multiclass classification on Edge-IIoT dataset

Classes	f1-score	Precision	sensitivity	specificity	Accuracy
SQL_injection	99.05	99.07	99.04	99.06	99.1
DDoS_HTTP	99.42	99.44	99.41	99.43	99.47
Ransomware	99.59	99.61	99.58	99.6	99.64
DDoS_ICMP	99.81	99.83	99.8	99.82	99.86
DDoS_UDP	99.85	99.87	99.84	99.86	99.9
Uploading	99.1	99.12	99.09	99.11	99.15
DDoS_TCP	99.76	99.78	99.75	99.77	99.81
Backdoor	98.86	98.88	98.85	98.87	98.91
Vulnerability_scanner	99.3	99.32	99.29	99.31	99.35
Port_Scanning	99.22	99.24	99.21	99.23	99.27
XSS	98.94	98.96	98.93	98.95	98.99
Password	99.1	99.1	99.07	99.09	99.13
MITM	99.75	99.77	99.74	99.76	99.8
Fingerprinting	99.27	99.29	99.26	99.28	99.32
Normal	99.9	99.92	99.89	99.91	99.95

Table 2: Performance comparison on Edge-IIoT dataset

sMethods	F1-score	precision	Sensitivity	specificity	accuracy
RNN [28]	75.86	76.26	76	-	76.13
EfficientNet [29]	98.21	98.22	98.21		98.21
CNN-LSTM-GRU [30]	-	92.21	-	-	98.70
Feed forward neural net [31]	86.12	86.20	85.53	-	88.26
ELM [32]	-	96.03	-	-	90.14
Proposed	99.39	99.41	99.38	99.40	99.44

suggested model's ROC curve for the dataset is shown in FIGURE 4. The graphic shows that the attack and normal classes have AUC values of more than 0.99, indicating that our proposed methodology is successful in identifying attacks in edge computing environments.

4.3 Assessment of Performance on IoT-23 Dataset:

This dataset demonstrates the strong performance of the proposed multiclass classification technique, achieving 99% accuracy in most classes, with slightly lower efficiency for "Heart beat" and "Part of Horizontal Port Scan" described in TABLE 3.

TABLE 4 checks how well the new and the earlier categorization methods work on the current dataset. It looks at both accuracy and precision. It also demonstrates the best overall performance of the classifier. The table shows a number of metrics. The proposed method performs better than the existing methods for intrusion detection in the EoT networks. It reaches 99.40% accuracy.

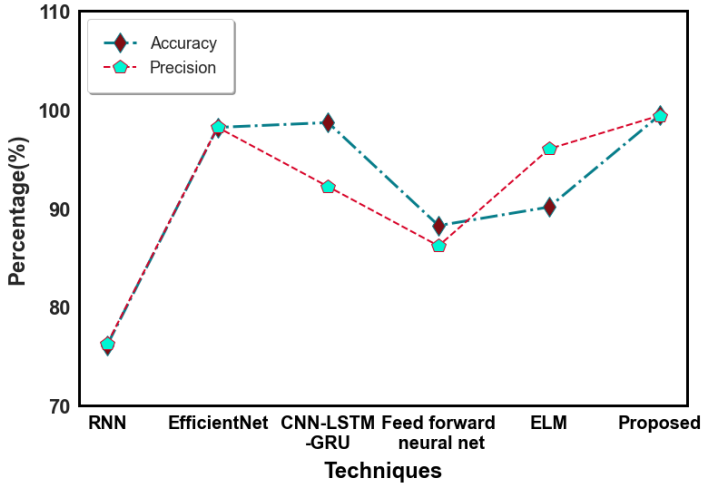

Figure 2: Accuracy and precision comparison on Edge-IIoT dataset

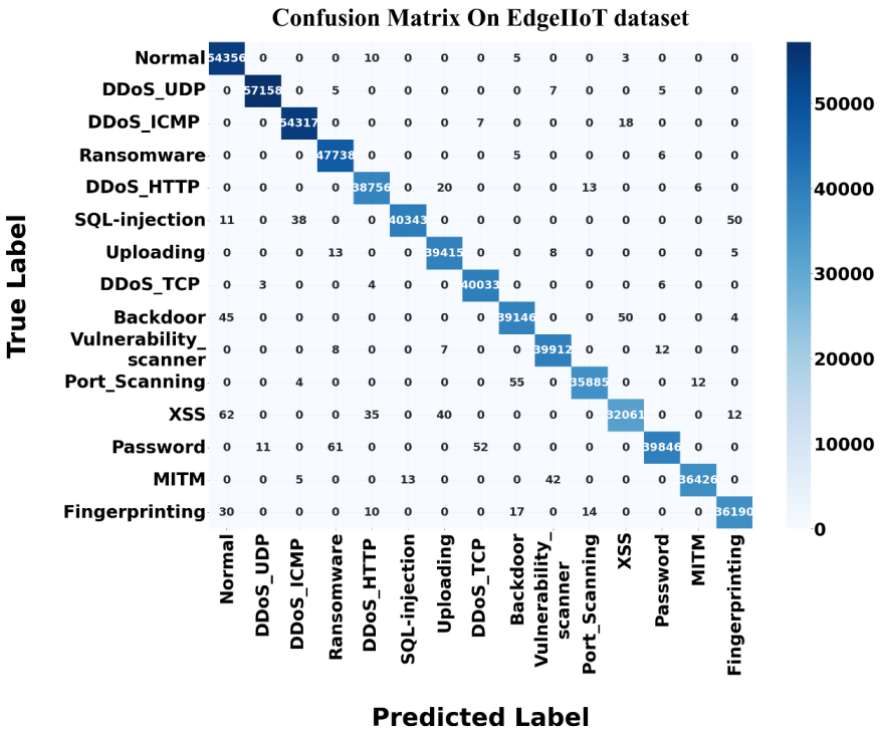

Figure 3: Confusion matrix for Edge IIoT dataset

The F1 score of the proposed method is the best because of using the Elastic Collision Seeker Optimization Algorithm (ESOA) for feature selection, which is used to improve intrusion detection performance. Further, the SqueezeNet-based classifier delivers an efficient and compact deep learning architecture that is well suited for Edge Computing. The performance metrics precision, sensitivity, F1-score are contrasted in FIGURE 5 using IoT-23 data set.

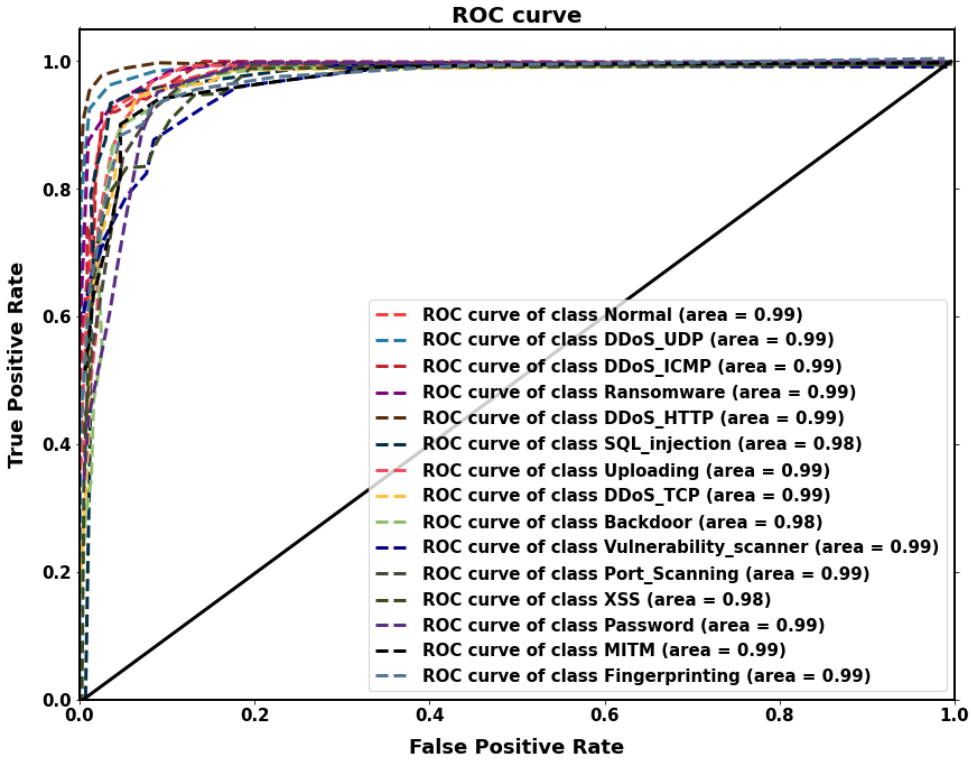

Figure 4: ROC graph on Edge-IIoT dataset

Table 3: Multiclass classification on IoT-23 dataset [27]

Classes	F1-score (%)	Precision (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)
Benign	99.6	99.6	99.5	99.6	99.6
DDoS	99.5	99.5	99.5	99.5	99.6
Mirai	99.4	99.5	99.4	99.4	99.5
Okiru	99.4	99.4	99.4	99.4	99.5
Torii	99.3	99.3	99.2	99.3	99.3
PartofHorizontalPortScan	99.1	99.1	99.1	99.1	99.2
Heart beat	99	99	99	99	99.1
FileDownload	99.2	99.2	99.2	99.2	99.3
Command and control	99.1	99.1	99.1	99.1	99.2

To check the efficacy of a classification model, one needs a confusion matrix, FIGURE 6 presents the confusion matrix from the IoT-23 dataset. The matrix demonstrates that only a tiny percentage of attack classes are incorrectly classified by the suggested framework, with the majority being correctly identified, which can be used to compare ground truth and model predictions. Despite the method's ability to effectively extract features, in the IoT-23 dataset, it can be seen that most classes

Table 4: Performance comparison of the IoT-23 dataset

techniques	F1-score	precision	sensitivity	specificity	accuracy
DNN [33]	94	94.72	93.56	-	95.29
MM-WMVEDL [34]	98.18	98.06	98.31	-	98.12
Adaboost [35]	78.8	81.3	76.3	-	-
KNN [36]	97.44	97.66	97.22	-	97
CNN-LSTM [37]	82.04	80.67	83.84	80.80	82.17
Proposed	99.34	99.36	99.32	99.35	99.40

of attacks are correctly identified, with few misclassifications, mainly between the similar Torii and Mirai packets.

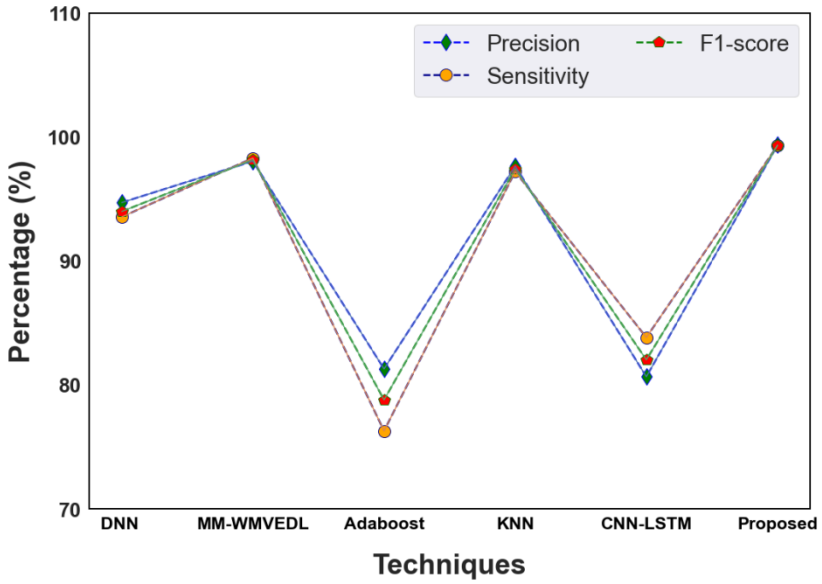

Figure 5: Comparison of precision, sensitivity, and f1-score on IoT-23 dataset

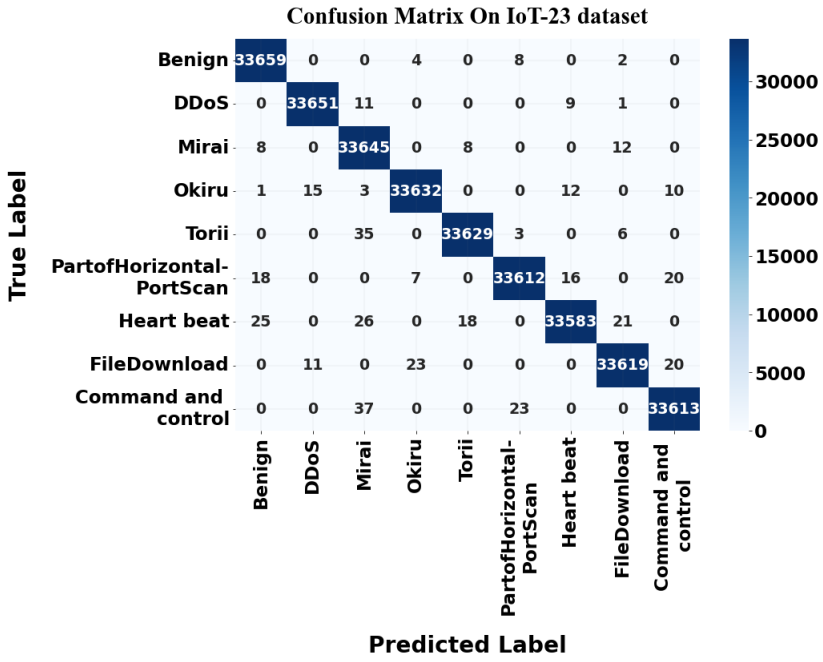

Figure 6: Confusion Matrix on IoT-23 dataset

In any IDS, ROC is an important parameter that can be used to visualize and compare the performance by plotting the FPR versus the TPR. FIGURE 7 shows the ROC curve of the proposed model for each type of attack and it is monitored that the model performs better than remaining models for all the attacks for the dataset IoT-23. The higher ROC(0.99) for all attack types further supports the network's ability to effectively distinguish the various attack traffic scenarios.

Overall, the results of the experiment demonstrate the high efficiency of the proposed system, which makes it a powerful solution for real-time IDS and response in the context of Edge of Things (EoT) security. This efficiency is evident across diverse datasets, showcasing the system's prowess in processing, learning, and detecting threats promptly.

4.4 Mitigation Results:

We tested the attacks in the Edge-IIoT and IoT-23 datasets to see how well they can be mitigated. FIGURE 8 shows the traffic pattern from the test day, with the traffic pattern with the mitigation module turned off and the traffic pattern with the mitigation module turned on.

The blue line shows the amount of traffic that occurred after deploying mitigation methods to attack, while the orange area stands for the part of the traffic without any mitigation policies. In this graphic, we can see the traffic before and after the proposed mitigation process, which demonstrates how the attacks impact network traffic. However, when the process of attack mitigation begins, the network traffic gets normalized.

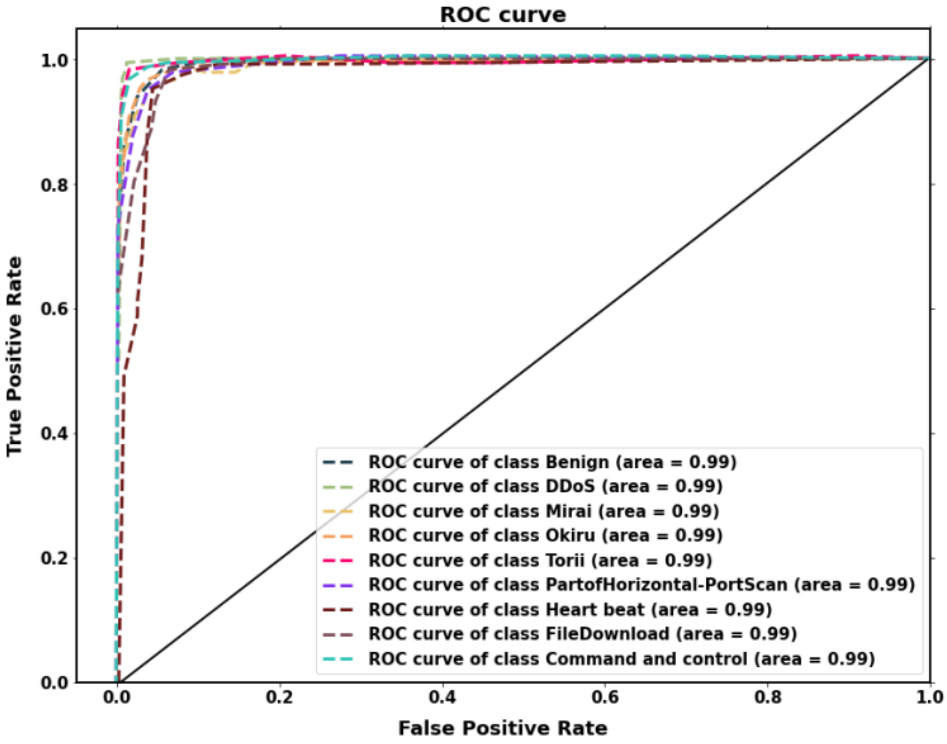

Figure 7: ROC graph on IoT-23 dataset

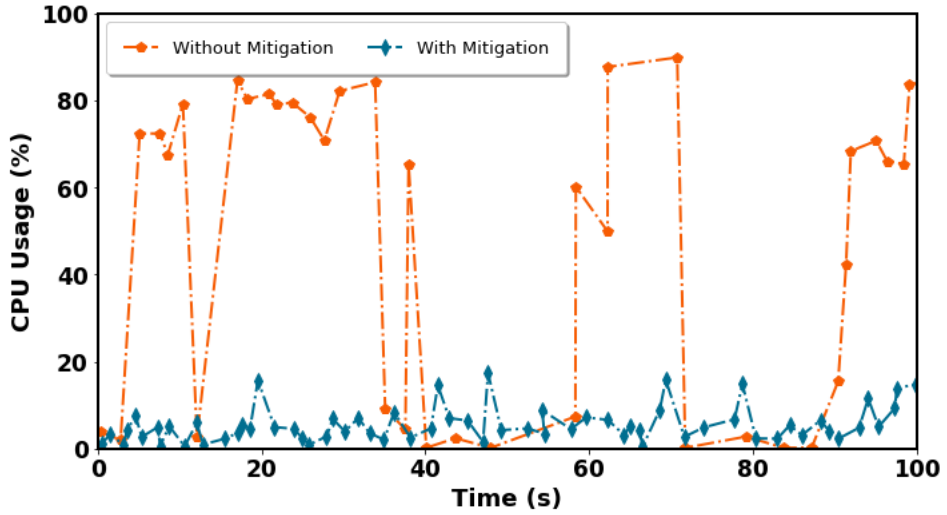

Figure 8: Mitigation Process Evaluation

4.5 Ablation Study:

The suggested IDS based on the MNasNet-SqueezeNet model will perform better than the state-of-the-art IDS techniques in terms of detection accuracy and other performance indicators. Two well-known datasets, Edge-IIoT and IoT-23, were used in the trials. The suggested approach outperformed current methods by a considerable margin on the Edge-IIoT dataset, with an accuracy of 99.44%, precision of 99.41%, sensitivity of 99.38%, specificity of 99.40%, and F1-score of 99.39%. Likewise, the suggested framework obtained a 99.34% F1 score, 99.36% accuracy, 99.32% sensitivity, 99.35% specificity, and 99.40% F1 score on the IoT-23 dataset.

we evaluated our IDS's performance both with and without the ESOA-based feature selection method. TABLE 5 shows the outcomes. The proposed framework with feature selection only employs the best subset of characteristics chosen by the ESOA algorithm. In contrast, the framework without feature selection makes use of the entire collection of features from the Edge-IIoT and IoT-23 datasets. By removing unnecessary or redundant features and choosing the most pertinent ones, the ESOA algorithm was used to decrease the feature space. By simplifying the data, this reduction in dimensionality enhances the proposed framework's efficacy and reliability. It is confirmed by the findings of the proposed framework on the Edge-IIoT dataset, which indicates that the proposed feature selection had an accuracy of 99.44%.

Table 5: Evaluation of with and without feature selection

	Edge-IIoT dataset		**IoT-23 dataset**	
Performance Metrics	**With Feature selection**	**Without feature Selection**	**With Feature selection**	**Without feature Selection**
Accuracy	99.44	98.95	99.40	98.84
Precision	99.41	98.93	99.36	98.79
Sensitivity	99.38	98.87	99.32	98.76
Specificity	99.40	98.92	99.35	98.80
F1-score	99.39	98.90	99.34	98.78

We examine the performance of the suggested framework with and without data augmentation, which claims that using ACGAN-based data augmentation approaches enhances the suggested IDS's performance on imbalanced datasets. The suggested framework with ACGAN-based data augmentation performs better than the model without augmentation in a number of important measures, as indicated in TABLE 6.

5. CONCLUSION

The increasing frequency of attacks in Edge computing can considerably damage information security. The research uses the deep learning technique of MNasNet-SqueezeNet for detecting and mitigating network attacks, which is utilized to address the IoT security issue. The proposed solution undergoes a benchmark testing procedure to determine the performance of intrusion detection using Edge-IIoT and IoT-23 benchmark datasets. The outcome indicate that the approach outperforms

Table 6: Evaluation of With and Without Augmentation

	Edge-IIoT dataset		IoT-23 dataset	
Performance Metrics	**With Augmentation**	**Without Augmentation**	**With augmentation**	**Without Augmentation**
Accuracy	99.44	99.17	99.40	99.08
Precision	99.41	99.13	99.36	99.03
Sensitivity	99.38	99.11	99.32	98.98
Specificity	99.40	99.14	99.35	99.02
F1-score	99.39	99.12	99.34	99.00

existing intrusion detection solutions for IoT networks with an accuracy of 99.44% for Edge-IIoT and 99.40% for IoT-23 datasets. In addition, an impressive area under the ROC curve of 0.99 was recorded, which demonstrates the ability of the algorithm to differentiate attack categories. Blockchain technology will be incorporated into the IDS for Edge-of-Things networks, along with other deep learning algorithms to improve performance further.

References

[1] Mohy-eddine M, Guezzaz A, Benkirane S, Azrour M. An Effective Intrusion Detection Approach Based on Ensemble Learning for Iiot Edge Computing. J Comput Virol Hacking Tech. 2023;19:469-481.

[2] Yu X, Yang X, Tan Q, Shan C, Lv Z. An Edge Computing Based Anomaly Detection Method in IoT Industrial Sustainability. Appl Soft Comput. 2022;128:109486.

[3] Belhadi A, Djenouri Y, Djenouri D, Srivastava G, Lin JC. Group intrusion detection in the Internet of Things using a hybrid recurrent neural network. Clust Comput. 2023;26:1147-1158.

[4] Yang R, He H, Xu Y, Xin B, Wang Y, et al. Efficient Intrusion Detection Toward IoT Networks Using Cloud–Edge Collaboration. Computer Networks. 2023;228:109724.

[5] Farooq DM, Khan MH. Artificial Intelligence-Based Approach on Cybersecurity Challenges and Opportunities in the Internet of Things & Edge Computing Devices. Int J Eng Comput Sci. 2023;12:25763-25768.

[6] Aldribi A, Singh A, Breñosa J. Edge of Things Inspired Robust Intrusion Detection Framework for Scalable and Decentralized Applications. Comput Syst Sci Eng. 2023;46:3865-3881.

[7] Myneni S, Chowdhary A, Huang D, Alshamrani A. Smartdefense: A Distributed Deep Defense Against DDoS Attacks With Edge Computing. Computer Networks. 2022;209:108874.

[8] Karthikeyan K, Madhavan P. Building a Trust Model for Secure Data Sharing (TM-SDS) in Edge Computing Using HMAC Techniques. Comput Mater Contin. 2022;71:4183-4197.

[9] Lin H, Xue Q, Feng J, Bai D. Internet of Things Intrusion Detection Model and Algorithm Based on Cloud Computing and Multi-Feature Extraction Extreme Learning Machine. Digit Commun Netw. 2023;9:111-124.

[10] Do XT, Le KH. Towards Remote Deployment for Intrusion Detection System to IoT Edge Devices. In Recent Advances in Internet of Things and Machine Learning: Real-World Applications. Cham: Springer International Publishing. 2022:301-316.

[11] Cho J. Efficient Autonomous Defense System Using Machine Learning on Edge Device. Comput Mater Contin. 2022;70:3565-3588.

[12] Shen S, Wu X, Sun P, Zhou H, Wu Z, et al. Optimal Privacy Preservation Strategies With Signaling Q-Learning for Edge-Computing-Based IoT Resource Grant Systems. Expert Syst Appl. 2023;225:120192.

[13] Bukhari O, Agarwal P, Koundal D, Zafar S. Anomaly Detection Using Ensemble Techniques for Boosting the Security of Intrusion Detection System. Procedia Comput Sci. 2023;218:1003-1013.

[14] Sharadqh AA, Hatamleh HA, Alnaser AM, Saloum SS, Alawneh TA. Hybrid Chain: Blockchain-Enabled Framework for Bi-Level Intrusion Detection and Graph-Based Mitigation for Security Provisioning in Edge-Assisted IoT Environment. IEEE Access. 2023;11:27433-27449.

[15] Altaf T, Wang X, Ni W, Liu RP, Braun R. Ne-Gconv: A Lightweight Node Edge Graph Convolutional Network for Intrusion Detection. Comput Secur. 2023;130:103285.

[16] Sedjelmaci H, Ansari N. On Cooperative Federated Defense to Secure Multi-Access Edge Computing. IEEE Consum Electron Mag. 2022;13:24-31.

[17] Liang P, Liu G, Xiong Z, Fan H, Zhu H, et al. A Fault Detection Model for Edge Computing Security Using Imbalanced Classification. J Syst Archit. 2022;133:102779.

[18] Qiu L, Wang L. Abnormal Traffic Detection Method of Internet of Things Based on Deep Learning in Edge Computing Environment. J Circuits Syst Comput. 2023;32: 2350283.

[19] Deng X, Chen B, Chen X, Pei X, Wan S, et al. A Trusted Edge Computing System Based on Intelligent Risk Detection for Smart IoT. IEEE Trans Ind Inform. 2024;20:1445-1454.

[20] Dao TN, Van Le DV, Tran XN. Optimal Network Intrusion Detection Assignment in Multi-Level IoT Systems. Computer Networks. 2023;232:109846.

[21] Guezzaz A, Azrour M, Benkirane S, Mohy-Eddine M, Attou H, et al. A Lightweight Hybrid Intrusion Detection Framework Using Machine Learning for Edge-Based IIoT Security. Int Arab J Inf Technol. 2022;19:822-830.

[22] Singh A, Chatterjee K, Satapathy SC. An Edge-Based Hybrid Intrusion Detection Framework for Mobile Edge Computing. Complex Intell Syst. 2022;8:3719-3746.

[23] Bangui H, Buhnova B. Lightweight Intrusion Detection for Edge Computing Networks Using Deep Forest and Bio-Inspired Algorithms. Comput Electr Eng. 2022;100:107901.

[24] Hoang TM, Le Thi TL, Quy NM. A Novel Distributed Machine Learning Model to Detect Attacks on Edge Computing Network. J Adv Inf Technol. 2023;14:153-159.

[25] Fenanir S, Semchedine F. Smart Intrusion Detection in IoT Edge Computing Using Federated Learning. Rev Intell Artif. 2023;37:1133-1145.

[26] Kumar V, Kumar V, Singh N, Kumar R. Enhancing Intrusion Detection System Performance to Detect Attacks on Edge of Things. SN Comput Sci. 2023;4:802.

[27] Maddu M, Rao YN. Res2Net-ERNN: Deep Learning Based Cyberattack Classification in Software Defined Network. Clust Comput. 2024;27:12821-12839.

[28] Rashid MM, Khan SU, Eusufzai F, Redwan MA, Sabuj SR, et al. A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks. Network. 2023;3:158-179.

[29] Latif S, Boulila W, Koubaa A, Zou Z, Ahmad J. DTL-IDS: An Optimized Intrusion Detection Framework Using Deep Transfer Learning and Genetic Algorithm. J Netw Comput Appl. 2024;221:103784.

[30] Khacha A, Saadouni R, Harbi Y, Gherbi C, Harous S, et al. Robust intrusion detection for IoT networks: an integrated CNN-LSTM-GRU approach. In 2023 International Conference on Networking and Advanced Systems (ICNAS). IEEE. 2023:1-6.

[31] Javed SH, Ahmad MB, Asif M, Akram W, Mahmood K, et al. APT Adversarial Defence Mechanism for Industrial IoT-Enabled Cyber-Physical System. IEEE Access. 2023;11:74000-74020.

[32] Dini P, Begni A, Ciavarella S, De Paoli E, Fiorelli G, et al. Design and Testing Novel One-Class Classifier Based on Polynomial Interpolation With Application to Networking Security. IEEE Access. 2022;10:67910-67924.

[33] Mahalingam A, Perumal G, Subburayalu G, Albathan M, Altameem A, et al. ROAST-IoT: A Novel Range-Optimized Attention Convolutional Scattered Technique for Intrusion Detection in IoT Networks. Sensors. 2023;23:8044.

[34] Sanju P. Enhancing Intrusion Detection in IoT Systems: A Hybrid Metaheuristics-Deep Learning Approach With Ensemble of Recurrent Neural Networks. J Eng Res. 2023;11:356-361.

[35] Kozik R, Pawlicki M, Choraś M. A New Method of Hybrid Time Window Embedding With Transformer-Based Traffic Data Classification in IoT-Networked Environment. Pattern Anal Appl. 2021;24:1441-1449.

[36] Abdalgawad N, Sajun A, Kaddoura Y, Zualkernan IA, Aloul F. Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. IEEE Access. 2021;10:6430-6441.

[37] Sahu AK, Sharma S, Tanveer M, Raja R. Internet of Things Attack Detection Using Hybrid Deep Learning Model. Comput Commun. 2021;176:146-154.