A Machine learning-based approach for multi-class intrusion detection and Classification in IoT using CICIoT2023 Dataset

Zohreh Hafezian
Dept. of Computer Engineering
Faculty of Engineering, Shahid Chamran University of Ahvaz
Ahvaz, Iran
z-hafezian@stu.scu.ac.ir

Marjan Naderan
Dept. of Computer Engineering
Faculty of Engineering, Shahid Chamran University of Ahvaz
Ahvaz, Iran
m.naderan@scu.ac.ir

Morteza Jaderyan
Dept. of Computer Engineering
Faculty of Engineering, Shahid Chamran University of Ahvaz
Ahvaz, Iran
m.jaderyan@scu.ac.ir

***Abstract*— The Internet of Things (IoT) is a set of physical devices connected to the Internet that collect information from the operating environment and share it with users or other devices. One of the most important challenges in IoT is the security of connected devices; they are vulnerable to attacks and infiltrations and traditional Intrusion Detection Systems (IDS) often fail to overcome this problem. In this article, six machine learning models, including Logistic Regression, AdaBoost, Perceptron, MLP, Random Forest and Hist-Gradient Boosting, are used to solve this problem. These models are trained on the recently available CICIoT2023 dataset and are used to classify attacks for three different classification modes of binary, eight and 34 classes. The performance of the models is evaluated based on several evaluation criteria such as accuracy, precision, recall, f1-score and training time. Evaluation results suggest that the Random Forest algorithm achieves the best classification results with test accuracy = 0.9955, while the Hist-Gradient Boosting algorithm has the best performance in terms of training time.**

Keywords—Internet of Things, Intrusion Detection System, Machine Learning, Multi-Class Classification, CICIoT2023 dataset

I. INTRODUCTION

The Internet of Things (IoT) network is expanding rapidly in the world and the amount of data transfer in this network is increasing every day. IoT enables users to acquire information about the status of objects and their operating environment and control them through mobile or web softwares. The purpose of the IoT is to communicate and exchange information between objects and devices to improve performance, efficiency and quality. This objective is accomplished by controlling and monitoring these objects remotely. IoT devices include a wide range of devices such as simple sensors, bracelets and smartphones, lamps, home appliances, security and alarm systems, and self-driving cars.

The Internet of Things has various applications in many fields, for example in smart health and medical staff. Doctors will be able to monitor patients' information such as heart rate, breathing status, vital signs and many other things using smart devices in real time and if any anomaly is detected, they can quickly respond. In smart agriculture, the farmer can be aware of the temperature, humidity, and condition of the soil at any moment. Inside a smart home, the lamps can be turned on or off by detecting the presence of people in the house, and the air conditioning system can be adjusted according to the temperature of the house. These situations are just minor examples of how IoT and its applications can impact our future.

IoT devices can communicate independently, without requiring interaction between humans or between humans and computers [1]. Figure 1 shows the general scheme of connecting devices in the Internet of Things.

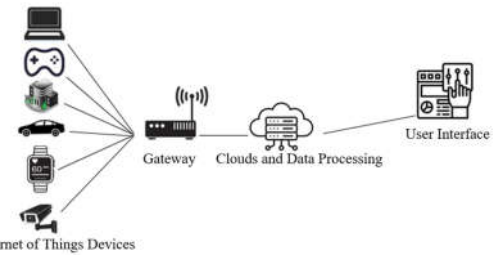

Fig. 1. Schematic view of IoT devices connections

With the dramatic increase in the number of IoT devices with different processing speed, storage capacity, and security mechanisms, IoT networks are being distributed heterogeneously [2]. This makes them vulnerable to physical and cyber-attacks. Since the IoT devices may be controlled by malicious intruders, the security issues, especially information leakage and user privacy caused by these attacks, can be more harmful than we think. Therefore, it is necessary to develop a robust and comprehensive Intrusion Detection System (IDS) to address security issues in the Internet of Things.

In general, IDSs are used to continuously monitor network traffic and issue alerts in case of unusual activities or suspicious attacks. Depending on where the intrusions or the attacks may occur or be detected, intrusion detection systems can be classified into two categories: network-based and host-based. A network-based IDS (NIDS) connects to one or more network segments and monitors network traffic for malicious activities, while a host-based IDS (HIDS) connects to a device and monitors malicious activities on that system.

In the Internet of Things, host-based attack detection approaches are not used often, since most IoT devices have limited computing power and energy which limits the complexity and efficiency of the intrusion detection algorithms. Additionally, these algorithms may affect the performance of the IoT devices by increasing their consumption of processing power and energy [3].

From the point of view of intrusion detection method, IDSs can be classified into two categories: signature-based (knowledge-based) and behavior-based (anomaly-based). Signature-based systems rely on specific characteristics of unusual traffic, while behavior-based systems detect intrusions by identifying deviations from normal traffic

patterns. Signature-based IDSs suffer from serious shortcomings, especially dealing with new and evolving intrusions or attacks. Since the characteristics of these intrusions might not match the characteristics of the unusual traffic patterns of known intrusions, they are less effective against these type of attacks or intrusions. Also, there are numerous issues with maintaining and updating a comprehensive database of intrusion traffic patterns. In contrast, behavior-based detection is more effective against unknown, abnormal, and malicious activities since it identifies any deviation from normal behavior as abnormal [4], [5].

Many studies have proposed the development of network security systems. Machine Learning (ML) methods play an important role in the field of cyber security and designing an intelligent security system for IoT environment. Machine learning algorithms are divided into three categories: Supervised, Unsupervised, and Semi-supervised. Supervised algorithms work with completely labeled data to identify the relationship between the data and its corresponding class. Unsupervised algorithms aim to uncover hidden patterns in unlabeled data. Semi-supervised algorithms leverage both unlabeled data for training and a small quantity of labeled data within a larger set of unlabeled data. [6].

Since specific types of devices are usually configured in IoT environments, there is a need for specific datasets for these devices, making it impossible to use public datasets for intrusion detection. In recent years, a number of new datasets such as BoT-IoT [7], TON-IoT [8], N-BaIoT [3] , and IoTID20 have been presented by researchers, which include normal IoT traffic and several types of attacks. These datasets have been used to train models to detect attacks in the IoT network. However, several possible attacks against IoT are not considered in these datasets. To this end, Neto et al. [9] introduced the newly available dataset CICIoT2023 in 2023.

So far, a number of machine learning algorithms, such as Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN), have been used to detect anomalies in the IoT network [10]. Next, a brief review of the most related body of literature surrounding the works that have used the CICIoT2023 dataset for model training is presented.

Altrad et al. [11] used Naive Bayes, Decision Tree, Random Forest, Quadratic Discriminant Analysis (QDA), AdaBoost, MLP, and KNN algorithms to detect web-based attacks on IoT devices. This approach classifies the network traffic into normal traffic and web attacks.

Ameera et al. [12] used Genetic Algorithm (GA) to optimize problem parameters and feature selection to detect web-based attacks on IoT devices. Also, a variety of learning models such as Gradient Boosting (GB) , MLP, LR, and KNN were used to detect attacks.

Wang et al. [13] proposed a light-weight hybrid model consisting of a deep neural network and a Bidirectional Long Short-Term Memory (Bi-LSTM) network to detect intrusion and integrate it into the infrastructure of IoT. They used the Incremental Principal Component Analysis (IPCA) algorithm to reduce the feature space dimensions and the dynamic quantization technique for the compression of network structure units. They also used the Optuna optimization framework to adjust the network parameters.

Akinul Islam et al. [14] used LSTM. According to the reported results, their proposed method has worked well in detecting some attacks, but it has exhibited poor performance in detecting attacks that have less data than other attacks.

In this paper, a machine learning-based approach for multi-class intrusion detection and Classification in IOT using CICIoT2023 dataset is investigated. A number of the most important machine learning algorithms such as Logistic Regression (LR), AdaBoost, Perceptron, Random Forest (RF), histogram-based Gradient Boosting and Muti-Layer Perceptron (MLP) are trained and evaluated for binary and multi-class intrusion detection and classification (eight and 34 classes). The results of the proposed methods are compared to the results presented in the paper authored by Neto et al. [9]. It is worth noting that, to the best of our knowledge, the histogram-based Gradient Boosting method has not been used for intrusion detection in IoT network.

This paper is structured as follows: in section II, the proposed method is explained in detail. In section III, the performance of the algorithms is evaluated, and finally, in section IV, conclusions and directions for future research are presented.

II. Proposed Method

In this article, the CICIoT2023 dataset is used to detect attacks against IoT devices. Figure 2 shows the flowchart of the proposed method. After preprocessing the network traffic data, the dataset is divided into 80% for training set and 20% for testing set. In the next step, six supervised ML algorithms are trained and evaluated in binary and multi-class classification settings (8 and 34 classes). In the following, each step is described in more detail.

A. CICIoT2023 Dataset

The new CICIoT2023 dataset [9] has a comprehensive set of IoT attack data that includes 33 different types of attacks performed on a network of 105 devices. In this setup, malicious IoT devices launch attacks against other IoT devices. This dataset includes a variety of new attack types not seen in other IoT datasets. The attacks can be categorized in three distinct classification modes:

- Two classes: normal traffic and attack
- Eight classes: normal traffic, DOS, DDOS, Web, Mirai, Spoofing, Recon, and BruteForce

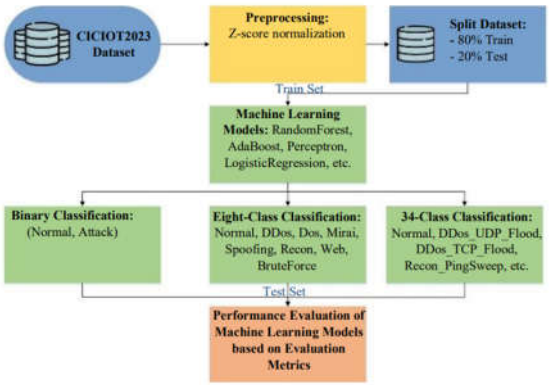

Fig. 2. Flowchart of the proposed method

- 34 classes: normal traffic and 33 types of attacks and sub-attacks. 33 types of attacks and sub-attacks in this data set include 12 types of DDOS attacks, four types of DOS attacks, five types of Recon attacks, two types of Spoofing, six types of Web-base, three types of Mirai, and one type of BruteForce attack.

In total, this dataset contains 47 features and about 46 million data samples.

B. Data Preprocessing

In the proposed methods, the features are scaled by the Z-score normalization method according to (1), which is used to scale the features at μ=0 and σ=1. μ is the mean and σ is the standard deviation.

$$z\text{-}score = \frac{x-\mu}{\sigma} \quad (1)$$

C. Machine Learning Algorithms

To detect and classify intrusions/attacks in IoT network, six conventional supervised machine learning methods are implemented in this paper. A brief description of each learning model is presented as follows.

Logistic Regression (LR): is a linear classifier that models the relationship between a set of input variables and the target variable and gives a probability value in the range between zero and one for each input using the logistic sigmoid function [15].

Perceptron (PER): is a linear classifier and one of the simplest architectures of artificial neural network that learns to separate classes by adjusting the weights of its inputs. It may not converge on data sets that are not linearly separable. In cases where data are not linearly seperable, more complex algorithms such as neural networks should be used.

AdaBoost (ADA): is an ensemble learning technique that combines several weak classifier models to create a strong classifier model. Each classifier focuses on training samples that were misclassified by the previous classifier. Each classifier algorithm is weighted based on the error during training, which has an impact on the final decision.

Random Forest (RF): is an ensemble learning technique which includes several decision trees. The final prediction in this algorithm is done by collecting the predictions from all the trees in the forest using a voting mechanism. The final prediction in classification tasks is the class predicted by the majority of trees as the final prediction [16].

Multi Layer Perceptron (MLP): is an artificial neural network that includes input layer, hidden layers and output layer that can discover complex and non-linear patterns in data [17].

Histogram-based Gradient Boosting (HGBT): is an ensemble learning technique and a type of Gradient Boosting algorithm that uses a histogram-based approach to estimate feature values during the tree building process [18]. Instead of using raw feature values, HGBT creates a histogram with a certain number of bins for each feature, and then uses the bin boundaries as split points for decision trees, which increases the computational efficiency and scalability of the algorithm and reduces the training time.

III. PERFORMANCE EVALUATION AND RESULTS

The six classification models, described in the previous section, have been modeled in Python development environment using the Scikit-Learn library. The hardware system for implementing these models is a server machine with Ubuntu 23.10 OS and 64.0 GB RAM.

In the following subsection, metrics for evaluating the performance of the classifiers are introduced. These metrics form the basis for comparing the proposed methods with related works.

A. Evaluation Metrics

In this article, the metrics of accuracy, precision, recall, F1-score, and training time in minutes are used to evaluate the performance of machine learning algorithms.

In order to understand these evaluation criteria, it is necessary to first introduce the concepts of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These concepts are used to describe the results of the predictions of a classification model compared to the actual labels. Figure 3 provides a clear picture of these concepts. Accordingly the evaluation criteria are defined as follows:

- Accuracy: this metric shows how well a model has correctly recognized samples and is calculated according to:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

- Recall: this metric examines the model's ability to correctly identify examples of a class. It is the ratio of correctly identified instances of a class to all instances of that class and calculated according to:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- Precision: this measure shows how many of the samples that the model has detected as positive were actually positive and is calculated according to:

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

- F1-score: this metric focuses on FN and precision on FP, as a balanced combination of precision and recall. It is a good measure to evaluate the model when the data set is unbalanced and calculated according to:

$$F1-score = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (5)$$

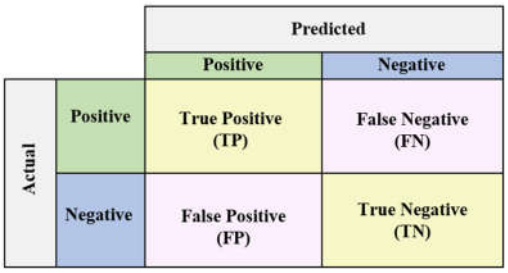

Fig. 3. Confusion Matrix concepts

B. Evaluation Results

Performance evaluation metrics were calculated for each model and compared with the results reported in [9]. Table I shows the results for binary classification mode, Table II shows the results for eight classes classification mode, and in Table III the results are calculated for 34 classes classification mode.

According to the results illustrated in the following tables, all the proposed machine learning algorithms achieved accuracy equal to or higher than 0.98. The evaluation metrics decreased with the increase in problem complexity, specifically in eight and 34 classes classification mode.

According to results illustrated in Table I, for binary classification, the proposed Perceptron model outperforms the Perceptron model presented in [9] in terms of accuracy and recall metric. The proposed AdaBoost outperforms the AdaBoost model presented in [9] in terms of all evaluation metrics. The proposed MLP outperforms the DNN model presented in [9] in terms of accuracy, precision and F1-score. Also, the performance results of the proposed RandomForest algorithm have improved compared to the results of the RandomForest model presented in [9].

According to results illustrated in Table II, for eight classes classification, the proposed Logistic Regression model outperforms the Logistic Regression model presented in [9] in terms of recall metric. The proposed AdaBoost outperforms the AdaBoost model presented in [9] in terms of accuracy, recall, F1-score metrics. The proposed MLP compared outperforms the DNN model presented in [9] in terms of accuracy, precision, F1-score. Also, the performance results of the proposed Random Forest algorithm have improved compared to the results of the Random Forest model presented in [9].

According to results illustrated in Table III, for 34 classes classification, the proposed Logistic Regression model outperforms the Logistic Regression model presented in [9] in terms of recall metric. The proposed MLP outperforms the DNN model presented in [9] in terms of all evaluation metrics.

Table. I. Evaluation results of the proposed machine learning algorithms in binary classification mode, compared with the results in [9]. Improved results of the proposed methods are highlighted.

Machine Learning Algorithms	**Metric**			
	Accuracy	***Recall***	***Precision***	***F1-score***
Logistic Regression [9]	0.98902	0.89040	0.86315	0.87625
Proposed Logistic Regression	0.98896	0.88994	0.86221	0.87554
Perceptron [9]	0.98175	0.79702	0.82543	0.81053
Proposed Perceptron	**0.98529**	**0.90923**	0.73413	0.79670
AdaBoost [9]	0.99589	0.94730	0.96563	0.95627
Proposed AdaBoost	**0.99599**	**0.94746**	**0.96778**	**0.95738**
Deep Neural Network [9]	0.99442	0.93327	0.94757	0.94030
Proposed MLP	**0.99456**	0.93260	**0.95212**	**0.94216**
Random Forest [9]	0.99680	0.96516	0.96539	0.96527
Proposed Random Forest	**0.99783**	**0.97127**	**0.98226**	**0.97670**
Proposed Hist-GradientBoosting	0.99640	0.94964	0.97499	0.96195

Also, the performance results of the proposed Random Forest algorithm have been improved compared to the results of the Random Forest model presented in [9].

As a conclusion, the proposed Random Forest algorithm has improved in all three modes and for all evaluation metrics compared to the previous research and also has the best performance and reliability compared to other algorithms. Histogram-GradientBoosting algorithm in binary and 8 classes classification and MLP algorithm in 34 classes classification after RandonForest have obtained the best results.

Finally, in Table IV, the performance of the proposed models is compared in terms of training time (displayed in minutes) in different classification modes. As mentioned, the Hist-GradientBoosting algorithm has the shortest training time due to the use of bin boundaries as dividing points for decision trees compared to the other algorithms.

Table. II. Evaluation results of the proposed machine learning algorithms in eight classes classification mode, compared with the results in [9]. Improved results of the proposed methods are highlighted.

Machine Learning Algorithms	**Metric**			
	Accuracy	***Recall***	***Precision***	***F1-score***
Logistic Regression [9]	0.83167	0.69605	0.51240	0.53942
Proposed Logistic Regression	0.83117	**0.71392**	0.51123	0.53842
Perceptron [9]	0.86631	0.65913	0.52391	0.55513
Proposed Perceptron	0.83555	0.66257	0.49124	0.51537
AdaBoost [9]	0.35135	0.48778	0.46492	0.36866
Proposed AdaBoost	**0.82430**	**0.52968**	0.43269	**0.41529**
Deep Neural Network [9]	0.99114	0.90664	0.67943	0.69726
Proposed MLP	**0.99194**	0.89378	**0.72529**	**0.76248**
Random Forest [9]	0.99436	0.91001	0.70540	0.71928
Proposed Random Forest	**0.99664**	**0.96548**	**0.84405**	**0.88677**
Proposed Hist-GradientBoosting	0.99320	0.71790	0.71377	0.71552

Table. III. Evaluation results of the proposed machine learning algorithms in 34 classes classification mode, compared with the results in [9]. Improved results of the proposed methods are highlighted.

Machine Learning Algorithms	**Metric**			
	Accuracy	***Recall***	***Precision***	***F1-score***
Logistic Regression [9]	0.80231	0.59520	0.48675	0.49388
Proposed Logistic Regression	0.80149	**0.59668**	0.48479	0.49080
Perceptron [9]	0.81959	0.50750	0.45463	0.44729
Proposed Perceptron	0.74820	0.47696	0.45123	0.43274
AdaBoost[9]	0.60788	0.60767	0.47962	0.47349
Proposed AdaBoost	0.57730	0.58221	0.44518	0.40092
Deep Neural Network[9]	0.98611	0.73186	0.66529	0.67234
Proposed MLP	**0.988761**	**0.78645**	**0.70531**	**0.71994**
Random Forest [9]	0.99164	0.83158	0.70449	0.71402
Proposed Random Forest	**0.99551**	**0.94839**	**0.80360**	**0.84005**
Proposed Hist-GradientBoosting	0.98376	0.66674	0.67592	0.66743

Table. IV. Models' training time in minutes

Machine Learning Algorithms	Classes	Training Time (min)
Proposed Logistic Regression	2-classes	5
	8-classes	24
	34-classes	88
Proposed Perceptron	2-classes	2
	8-classes	16
	34-classes	41
Proposed AdaBoost	2-classes	80
	8-classes	100
	34-classes	165
Proposed MLP	2-classes	180
	8-classes	273
	34-classes	171
Proposed Random Forest	2-classes	90
	8-classes	140
	34-classes	164
Proposed Hist-GradientBoosting	2-classes	5
	8-classes	4
	34-classes	7

IV. Conclusion and Future Works

In this paper, six machine learning methods are utilized to implement an IDS in IoT environments. There are many datasets including normal traffic and attacks for IoT, but due to the large variety of attacks in the CICIoT2023 dataset, it was used in this research. From the obtained results, it was observed that machine learning algorithms have the ability to detect attacks and the Random Forest algorithm had the best results in all three modes and for all evaluation metrics. On the other hand, the Hist-Gradient Boosting method reached the least training time for all three modes of classification.

In future work, considering the variety of network attacks and the complexity of the attack environment, deep learning methods can be used, which have a high ability to solve complex problems. Deep learning can extract complex nonlinear features from network traffic data and be effective in attack detection.

Acknowledgements

This work is supported in part by Shahid Chamran University of Ahvaz.

References

[1] K. Albulayhi, A. A. Smadi, F. T. Sheldon, and R. K. Abercrombie, "IoT intrusion detection taxonomy, reference architecture, and analyses," *Sensors,* vol. 21, no. 19, p. 6432, 2021.

[2] B. A. Alabsi, M. Anbar, and S. D. A. Rihan, "CNN-CNN: dual convolutional neural network approach for feature selection and attack detection on internet of things networks," *Sensors,* vol. 23, no. 14, p. 6507, 2023.

[3] Y. Meidan *et al.*, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing,* vol. 17, no. 3, pp. 12-22, 2018.

[4] T. Zixu, K. S. K. Liyanage, and M. Gurusamy, "Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, 2020: IEEE, pp. 1-7.

[5] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies,* vol. 32, no. 7, p. e4121, 2021.

[6] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science,* vol. 171, pp. 1251-1260, 2020.

[7] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems,* vol. 100, pp. 779-796, 2019.

[8] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustainable Cities and Society,* vol. 72, p. 102994, 2021.

[9] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors,* vol. 23, no. 13, p. 5941, 2023.

[10] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things,* vol. 7, p. 100059, 2019.

[11] A. Altrad, "IOTs Traffics Detection and Analysis Using Machine Learning for Cybersecurity Application," in *2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 2023: IEEE, pp. 78-83.

[12] A. S. Jaradat, A. Nasayreh, Q. Al-Na'amneh, H. Gharaibeh, and R. E. Al Mamlook, "Genetic optimization techniques for enhancing web attacks classification in machine learning," in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2023: IEEE, pp. 0130-0136.

[13] Z. Wang, H. Chen, S. Yang, X. Luo, D. Li, and J. Wang, "A lightweight intrusion detection method for IoT based on deep learning and dynamic quantization," *PeerJ Computer Science,* vol. 9, p. e1569, 2023.

[14] A. I. Jony and A. K. B. Arnob, "A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset," *Journal of Edge Computing,* vol. 3, no. 1, pp. 28-42, 2024.

[15] E. Besharati, M. Naderan, and E. Namjoo, "LR-HIDS: logistic regression host-based intrusion detection system for cloud environments," *Journal of Ambient Intelligence and Humanized Computing,* vol. 10, pp. 3669-3692, 2019.

[16] L. Breiman, "Random forests," *Machine learning,* vol. 45, pp. 5-32, 2001.

[17] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, classifiaction," 1992.

[18] S. Seth, G. Singh, and K. Kaur Chahal, "A novel time efficient learning-based approach for smart intrusion detection system," *Journal of Big Data,* vol. 8, no. 1, p. 111, 2021.