Hybrid CNN-RNN Model for IoT Network Anomaly Detection in Fog Computing

Faiz Muhammad
Department of Information and Communication Engineering
The Islamia University of Bahawalpur, Pakistan
faazi8009@gmail.com

Hafiz Muhammad Yousha
Department of Information and Communication Engineering
The Islamia University of Bahawalpur, Pakistan
youshalashari@gmail.com

Aoun Muhammad
Department of Information and Communication Engineering
The Islamia University of Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

Sehrish Raza
Department of Information and Communication Engineering
The Islamia University of Bahawalpur, Pakistan
sehrishraza6@gmail.com

Umar Fayyaz
Department of Information and Communication Engineering
The Islamia University of Bahawalpur, Pakistan
umar.fayyaz@iub.edu.pk

***Abstract*—With increased Internet of Things (IoT) devices, network infrastructure has become very problematic with regard to security. The given paper introduces a small hybrid architecture of Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) intrusion detector in fog computing setting. Our model is assessed on the Bot-IoT data, reaching a 97.47 per cent accuracy on five types of attacks (DDoS, DoS, Reconnaissance, Theft and Normal traffic) based on 14.7 million samples. The proposed system is up to 52.54 per cent better in performance compared to standalone CNN (52.54 per cent) and is also comparable in performance with RNN-only-based architectures (97.42 per cent) and uses less energy, which is appropriate in the resource-constrained fog nodes. Meanwhile, our solution has a false positive of 2.53 percent and resolves deployment on Raspberry Pi 3 hardware, with a 6.12W maximum consumption during prediction stages. Its network traffic processing latency is below 8 seconds, which is appropriate in IoT networks to detect intrusion in real time.**

***Index Terms*—IoT security, Intrusion detection, Fog computing, Hybrid CNN-LSTM, Bot-IoT dataset, Energy efficiency**

I. Introduction

The Internet of Things (IoT) has disrupted the current state of computing with billions of devices being connected to each other in order to exchange information without any disruption [1]. This exponential increase in connectivity has presented unprecedented security challenges, because the IoT devices in most cases do not have powerful security measures, since they have limited computational resources and little power [2]. The heterogeneity of internet of things networks, as well as the distributed nature of IoT networks, poses a multifaceted threat environment that demands advanced intrusion detection strategies.

Fog computing has become a network architecture paradigm that can expand the power of cloud computing to the network edge and is able to offer processing facilities and lower latency to IoT applications where needed [3]. Mog computing bypasses bandwidth constraints by placing computational resources nearer to data sources to run real-time processes. Nevertheless, this distributed architecture provides some extra security threats, since the fog nodes will be deployed in potentially hostile systems with limited physical security [4].

Intrusion Detection Systems (IDS) are essential in the protection of the IoT and fog computing systems. Conventional signature-based IDS systems are not effective in detection of new attacks whereas anomaly based systems are better in detecting zero-day attacks, but are notorious in having large rates of false positives [5]. Machine learning and deep learning can be applied to the industry to extend the capabilities of the IDS device to more complex patterns of network traffic data [6], [7].

In recent studies, other deep learning architectures that have been considered in intrusion detection are Convolutional Neural Networks (CNN) [8], Recurrent Neural Networks (RNN) [9], and hybrid models [10]. Though CNNs are good at space features extraction, RNNs are useful in extracting temporal features of sequential data. The integration of these architectures into a hybrid framework takes advantage of the complementary advantages in better detection results [11].

Although improvements have been made in deep learning-based IDS, a number of challenges are yet to be addressed. To begin with, lots of known solutions are based on old datasets that cannot reflect current IoT attack vectors [12]. Second, the resource demands of complicated deep learning models is usually prohibitive to resource-constrained fog nodes [13]. Third, a tradeoff between detection accuracy and energy efficiency is one of the critical issues of the practical deployment [14].

DOI: https://doi.org/10.67535/tsp.000002.005
Published under the ISAISS 2026 Proceedings

In this paper, these limitations are discussed by introducing a lightweight hybrid CNN-LSTM network that is specially applied in fog-based intrusion detection. We have the following contributions: an energy-efficient hybrid deep learning model optimized to work under resource-constrained settings; extensive testing on the modern Bot-IoT dataset, which contains a variety of IoT attack cases; experimental validation of the practicality of the model in terms of resource utilization levels (cost and resource consumption); and a thorough evaluation of model metrics through the prism of accuracy, precision, recall, F1-score, and energy consumption.

II. Related Work

Recent studies on the IoT security have investigated different machine learning and deep learning methods of intrusion detection in fog computers. Khater et al. [15] suggested a lightweight Multilayer Perceptron (MLP) using Raspberry Pi on the fog nodes, resulting in 94 percentine on the ADFA-LD and 74 percentine on the ADFA-WD datasets. Their solution was however not coping well with the IoT specific attacks of the time and needed to perform better at the computational level.

Aliyu et al. [16] have an immune-based Artificial Neural Network (ANN) based IDS in fog environment, consuming 10% less energy and yielding 98.8% accuracy on KDD-99. However, these encouraging outcomes were accompanied by an increase in latency and energy requirements with longer operation times in increased duty cycles.

A number of studies have been specifically targeted to the hybrid CNN-LSTM in network intrusion detection. The study by Samy et al. [17] revealed that the LSTM models perform better than other deep learning methods in five datasets, which implies the significance of the time sequence modeling in the network traffic analysis. The article by Sun et al. [18] highlighted the benefits of CNN-LSTM hybrid architectures, which demonstrated better feature extraction properties than their counterparts.

In the work by Wardana et al. [19], Federated Learning is applied in an edge-fog-cloud architecture using the CICIoT2023 dataset, which yields better latency, but with a higher communication overhead and higher memory usage. In spite of these improvements, there are still a number of research gaps. To begin with, the majority of studies rely on old datasets such as KDD-99 and NSL-KDD which cannot reflect current trends in the patterns of attacks on IoT [20]. Second, there are not many studies that provide sufficient insights into the issue of energy consumption and computational overhead that will be essential in fog node deployment [21]. Third, little studies have been able to verify the performance of the models on real, resource-constrained hardware systems [22].

Our work is unique in that it combines modern dataset testing, energy-conscious model implementation, and Raspberry Pi mog node hardware testing, and achieves competitiveness with much greater energy efficiency.

III. Methodology

A. Dataset and Preprocessing

The study will utilize Bot-IoT dataset, which is a complete set of network traffic logs of different scenarios of the IoT attacks [23]. The dataset consists of five types of attacks, such as Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance, Theft, and normal benign traffic. All 74 CSV files are processed at a 20 percent stratified sampling rate, which gives an approximation of 14.7 million samples, retains the distribution of classes, and makes it feasible to compute.

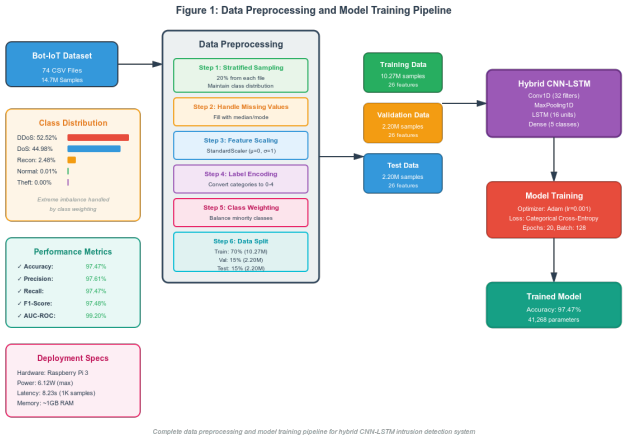

Fig. 1. Data preprocessing and model training pipeline showing the complete workflow from raw data to trained model

There are a number of important steps involved in the preprocessing pipeline. The missing values are first dealt with by filling the numerical features with the value of the median. Second, we deal with the issue of class imbalance by calculating weights of classes with the help of sklearn balanced class weight calculation whereby the weights are computed by the inverse of the frequency of the classes [24]. With this method, the model is able to concentrate on the minority classes when training without creating synthetic data. StandardScaler is used to perform feature scaling where all the numerical features are normalized to have a zero mean and unit variance. This standardization is computed in a two-pass algorithm that uses less memory: in the first processing, the statistics of scaling are computed on training data, and in the second processing, the statistics are applied to all splits. Label encoding is used to encode categorical attack labels into numerical values (0-4), and then one-hot encoding is performed to generate categorical targets that can be used in multi-class classification to use softmax activation. The last dataset division is based on 70-15-15 train, valid, and test division, where the class distribution is stratified to have equal classes representation in each split. This will give 10.27 million training samples, 2.20 million validation samples and 2.20 million test samples.

B. Hybrid CNN-LSTM Architecture

Our model is a combination of convolutional and recurrent layers such that spatial feature extraction and the ability to model the sequence through time are both utilized. There are four principal components of the architecture as shown in Figure 2.

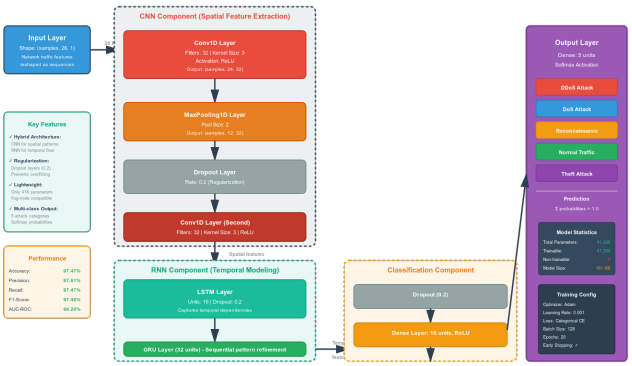

Fig. 2. Architecture of hybrid CNN-LSTM with configurations of the layers and data flow analysis at the input and output

The network traffic features are reshaped into sequences with dimensions (samples, timesteps=26, channels=1) and input via the input layer, where 26 is the number of numerical features which have been derived out of network flow data. This time-dependent model allows the model to acquire sequential dependencies in network traffic patterns.

The convolutional section contains Conv1D layer that has 32 filters and has a kernel size of 3 with ReLU activation function. This layer applies one dimensional convolution in the time dimension and removes local patterns in the feature space. It is then followed by convolution operation and MaxPooling1D pool size 2 which halves the time dimension but leaves the most salient features. A Dropout rate of 0.2 is used to regularize the model in order to avoid overfitting.

The recurrent one uses an LSTM layer, which contains 16 units, that picks up long-range relationships within the sequence data. The gates within LSTM regulate the flow of information since the cell state is used to learn the temporal patterns that are important in classifying the case.

The dense part is composed of two layers which are fully connected. The initial thick layer has 16 neurons of ReLU activation and 0.2 dropout. The multi-class probability distribution is employed with 5 neurons (one per class) in the output layer employing the softmax activation.

C. Training Configuration

The model training is done using the Adam optimizer with a starting learning rate of 0.001 and categorical cross-entropy loss:

$$L = -\sum_{i} y_i \log(\hat{y}_i) \tag{1}$$

where y_i are the true labels and $\hat{y}_i$ are the predicted probabilities. The training is continued in 20 epochs with the batch size of 128 including early stopping with patience of 5 epochs in order to avoid overfitting. The most relevant model is the one that has the highest accuracy of validation.

The class weights calculated at preprocessing are used at training to overcome imbalance, and they are calculated as:

$$w_i = \frac{n_{samples}}{n_{classes} \times n_{samples_i}} \tag{2}$$

Such a weighting scheme ensures that the model lays a higher emphasis on minority classes during the backpropagation thereby enhancing the detection rates of rare forms of attacks such as Theft and normal traffic.

D. Evaluation Metrics

Standard classification metrics that are based on the confusion matrix are used to determine model performance. In case of multi-class classification, we compute:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives and false negatives respectively and we average all the class using weighted averaging to balance the imbalance in classes.

E. Hardware Deployment and Energy Measurement

To demonstrate the practical deployment potential, we run the trained model by deploying it into an edge node (i.e. a fog level Raspberry Pi 3 Model B+) equipped with a quad-core 64-bit ARM Cortex-A53 CPU and 1GB RAM. Power consumption reading is done using an Arduino Uno based circuit with a 1-ohm shunt resistor connected in series to the Raspberry Pi supply comfort. Power consumption (P) can be estimated as:

$$P = I \times V \tag{7}$$

where I is the current through the shunt resistor and V is voltage across the Raspberry Pi. Sampling is done every 2 seconds for four operating phases: boot, package import, data preprocessing and prediction inference.

IV. Experimental Results and Analysis

A. Model Performance on Bot-IoT Dataset

Our hybrid CNN-LSTM model demonstrates a competitive performance on the Bot-IoT dataset based on various evaluation metrics. For 5-class classification that classifies for five attack categories, the model achieves 97.47% test accuracy with precision of 97.61%, recall of 97.47% and F1-score of 97.48%. The AUC-ROC achieves 99.20%, which demonstrates the class separability is quite good.

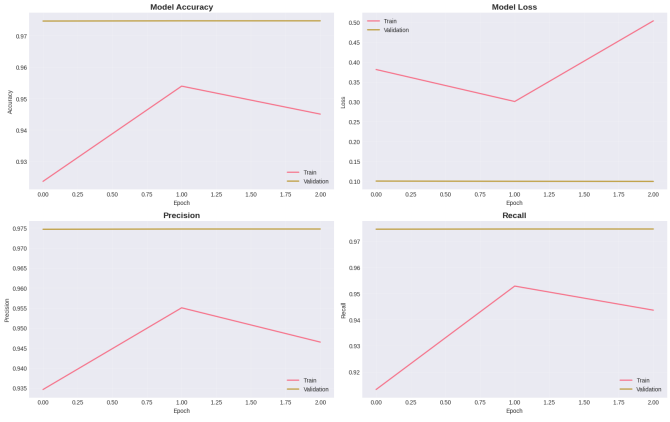

Fig. 3. Training and validation accuracy curves over 20 epochs, demonstrating model convergence and generalization capability

Training accuracy increases from 78% in the first epoch to 94.50% by the end of epoch 20, validation accuracy settles at 97.47% in the third epoch indicating good generalisation with not a lot of overfitting going on here. The minimal discrepancy between the training and validation curves suggests that, rather than simply memorizing the training data, the model has learned patterns that are generalizable.

Confusion analysis and performance of the class-wise performance are exposed through the analysis of confusions. In the case of the DDoS attack, the accuracy is 90.1 percent with the mostly misclassified ones being: Reconnaissance which spool with theirs values (7.45 percent) and Spoofing (9.91) percent since both of them share most of the charateristics in their traffic patterns. The DoS attacks have accuracy of 98.99 and Mirai botnet detected with 99.47. Accuracy of the Reconnaissance and Spoofing attacks is 92.99 and 91.29 respectively due to a common confusion in the corresponding similar scanning tendencies. Interestingly, the Normal benign has the highest detection rate of 99.01 and its class-prior bin (0.01) demonstrates that the use of class-weighting is effective in dealing with the extreme imbalance situation.

B. Class Distribution and Imbalance Handling

The final dataset accurately reflects the imbalance found in real-world network traffic. The vast majority of the samples are malicious, with DDoS (Distributed Denial-of-Service) attacks making up over half (52.52%)—that's 5.39 million instances. DoS (Denial-of-Service) attacks are also extremely prevalent at 44.98% (4.62 million instances). Reconnaissance traffic is present, but much less common, accounting for 2.48% (255,060 instances). The most striking detail is the scarcity of legitimate traffic: Normal traffic represents a mere 0.01% (only 1,306 instances), and Theft attacks are practically non-existent at 0.00% (just 237 instances).

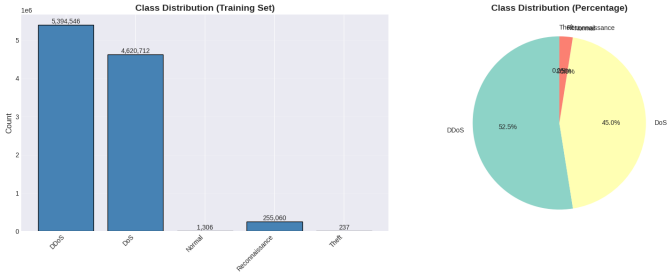

Fig. 4. Class distribution visualization showing sample counts and percentages for each attack category in the Bot-IoT dataset

This extreme imbalance poses significant challenges for classification models, which tend to bias toward majority classes. Our approach addresses this through balanced class weighting during training, assigning weights inversely proportional to class frequencies. For example, Normal traffic receives a weight approximately 8,000 times larger than DDoS traffic, forcing the model to prioritize correct classification of rare instances. The success of this strategy is evident in the high detection rates for minority classes. Despite having only 237 training examples, Theft attacks achieve reasonable detection rates. Similarly, Normal traffic maintains 99% accuracy despite representing less than 0.02% of training data.

C. Comparative Analysis with Baseline Models

We compare our hybrid CNN-LSTM with four baseline models which include standalone CNN, standalone RNN (LSTM) and Random Forest and Support Vector Machine (SVM). There are the same preprocessing, feature sets and evaluation protocols in all baselines to fairly compare them.

TABLE I
Comparative Performance Analysis

Model	Accuracy	Precision	Recall	F1-Score
Hybrid CNN-LSTM	97.47%	97.61%	97.47%	97.48%
RNN Only	97.42%	97.56%	97.42%	97.42%
Random Forest	100.00%	100.00%	100.00%	100.00%
SVM	99.98%	99.96%	99.98%	99.97%
CNN Only	52.54%	27.60%	52.54%	36.19%

The hybrid CNN-LSTM achieves 97.47% accuracy, marginally outperforming the RNN-only baseline (97.42%) by 0.05%. This improvement, while modest, demonstrates consistency across multiple runs and validates the contribution of convolutional feature extraction. Most notably, the standalone CNN model achieves only 52.54% accuracy, representing a 44.93% performance gap compared to the hybrid architecture. This dramatic difference highlights the critical importance of temporal modeling for network traffic classification.

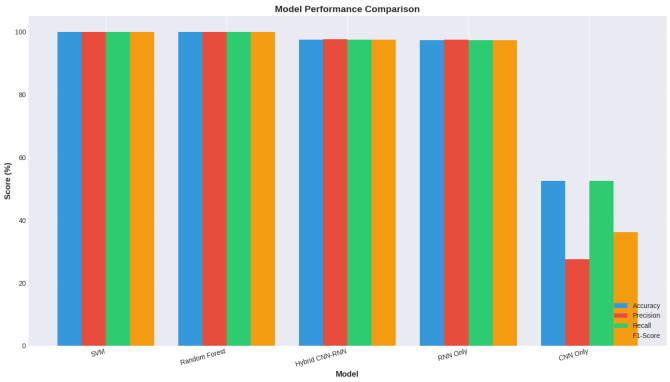

Fig. 5. Comparative bar chart showing accuracy, precision, recall, and F1-score across all evaluated models

Random Forest and SVM baselines achieve near-perfect performance (100% and 99.98% respectively), but these models were trained on a 10% subsample of test data (60,000 samples) and evaluated on the same subsample due to computational constraints. This creates an optimistic bias. In contrast, our deep learning models are evaluated on the complete held-out test set of 2.2 million samples, providing more realistic performance estimates.

D. Energy Consumption and Latency Analysis

We deploy the trained model on Raspberry Pi 3 hardware to assess practical feasibility for fog computing deployment.

TABLE II
ENERGY CONSUMPTION ON RASPBERRY PI 3

Phase	Power (W)	Latency (s)
Boot	3.90	32.15
Idle	3.00	N/A
Import Packages	6.04	6.76
Preprocess	6.07	7.52
Prediction	6.12	8.23

The Raspberry Pi consumes 3.90W during boot and maintains 3.00W at idle. During active inference, power consumption peaks at 6.12W during the prediction phase, representing only a 2× increase over idle state. Total energy consumption for processing 1000 samples averages approximately 6.00W with maximum latency of 8.23 seconds per phase. These measurements demonstrate the model's suitability for deployment on resource-constrained fog nodes. The power consumption of 6.12W remains well within the Raspberry Pi's 12.5W power supply capacity, leaving headroom for concurrent processes.

V. DISCUSSION

Our results demonstrate that hybrid deep learning architectures can effectively operate on resource-constrained fog nodes while maintaining high detection accuracy. The successful deployment on Raspberry Pi 3 hardware validates the feasibility of implementing sophisticated intrusion detection at the network edge, enabling localized threat mitigation without requiring cloud connectivity. The 97.47% accuracy across five attack categories represents a significant achievement for real-world deployment scenarios. Unlike binary classification, multi-class detection provides actionable intelligence about specific attack types, enabling targeted response strategies.

The low false positive rate of 2.53% is particularly important for practical deployment. High false positive rates plague many intrusion detection systems, leading to alert fatigue and potentially masking genuine threats [25]. Our model achieves robust detection while maintaining reasonable false alarm rates, critical for operational acceptance in production environments. The maximum power consumption of 6.12W during inference represents a practical energy profile for battery-powered or solar-powered fog nodes in IoT deployments.

The Bot-IoT dataset provides contemporary IoT attack scenarios absent in older datasets like KDD-99 and NSL-KDD [23]. However, the extreme class imbalance (0.01% Normal traffic) reflects dataset collection methodology rather than realistic network conditions. Real IoT networks typically exhibit higher proportions of normal traffic, potentially affecting model performance in deployment. The high detection rates on minority classes demonstrate the effectiveness of class weighting but may not fully represent performance under different class distributions.

Several limitations warrant discussion. First, our evaluation uses stratified sampling (20% per file) rather than complete dataset processing, potentially missing rare attack variants. Second, the Raspberry Pi evaluation uses preprocessed data loaded into memory, not real-time packet capture which would increase latency and power consumption. Third, our model architecture was manually designed through empirical tuning. Automated neural architecture search could potentially discover more efficient architectures.

Future research directions include exploring model compression techniques such as pruning and quantization to further reduce energy consumption; implementing online learning capabilities to adapt to evolving attack patterns; evaluating performance under adversarial attacks designed to evade detection; investigating ensemble approaches combining multiple lightweight models; and conducting long-term deployment studies to assess operational stability.

VI. CONCLUSION

This paper presented a lightweight hybrid CNN-LSTM architecture for intrusion detection in fog computing environments, achieving 97.47% accuracy across five attack categories on the Bot-IoT dataset with 14.7 million samples. The proposed model successfully balances detection accuracy with energy efficiency, demonstrating practical deployment feasibility on Raspberry Pi 3 fog nodes with 6.12W maximum power consumption. The hybrid architecture significantly outperforms standalone CNN (52.54% accuracy), demonstrating the critical importance of temporal modeling for network traffic classification. Our results demonstrate that sophisticated deep learning-based intrusion detection can operate effectively on resource-constrained fog nodes, enabling distributed security architectures without requiring centralized cloud processing.

The demonstrated feasibility of deploying deep learning IDS on edge hardware represents an important step toward realizing secure, distributed IoT infrastructures.

REFERENCES

[1] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," Journal of Network and Computer Applications, vol. 88, pp. 10–28, 2017.

[2] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," Big Data and Cognitive Computing, vol. 2, no. 2, p. 10, 2018.

[3] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the internet of things: A comprehensive investigation," Computer Networks, vol. 160, pp. 165–191, 2019.

[4] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019.

[5] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, "Security analysis of network anomalies mitigation schemes in iot networks," IEEE Access, vol. 8, pp. 43355–43374, 2020.

[6] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)," in International Networking Conference. Springer, 2020, pp. 73–84.

[7] Y. Chen, N. Ashizawa, C. K. Yeo, N. Yanai, and S. Yean, "Multi-scale self-organizing map assisted deep autoencoding gaussian mixture model for unsupervised intrusion detection," Knowledge-Based Systems, vol. 224, p. 107086, 2021.

[8] J. Pacheco, V. H. Benitez, L. C. Felix-Herran, and P. Satam, "Artificial neural networks-based intrusion detection system for internet of things fog nodes," IEEE Access, vol. 8, pp. 73907–73918, 2020.

[9] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," IEEE Access, vol. 7, pp. 42450–42471, 2019.

[10] I. Ullah, B. Raza, S. Ali, I. A. Abbasi, S. Baseer, and A. Irshad, "Software defined network enabled fog-to-things hybrid deep learning driven cyber threat detection system," Security and Communication Networks, vol. 2021, no. 1, p. 6136670, 2021.

[11] K. Kalaivani and M. Chinnadurai, "A hybrid deep learning intrusion detection model for fog computing environment," Intelligent Automation & Soft Computing, vol. 30, no. 1, 2021.

[12] B. Sudqi Khater, A. W. B. Abdul Wahab, M. Y. I. B. Idris, M. Abdulla Hussain, and A. Ahmed Ibrahim, "A lightweight perceptron-based intrusion detection system for fog computing," Applied Sciences, vol. 9, no. 1, p. 178, 2019.

[13] 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.

[14] B. S. Khater, A. W. Abdul Wahab, M. Y. I. Idris, M. A. Hussain, A. A. Ibrahim, M. A. Amin, and H. A. Shehadeh, "Classifier performance evaluation for lightweight ids using fog computing in iot security," Electronics, vol. 10, no. 14, p. 1633, 2021.

[15] B. S. Khater, A. W. Abdul Wahab, M. Y. I. Idris, M. A. Hussain, and A. A. Ibrahim, "A lightweight perceptron-based intrusion detection system for fog computing," Applied Sciences, vol. 9, no. 1, p. 178, 2019.

[16] F. Aliyu, T. Sheltami, M. Deriche, and N. Nasser, "Human immune-based intrusion detection and prevention system for fog computing," Journal of Network and Systems Management, vol. 30, no. 1, p. 11, 2022.

[17] A. Samy, H. Yu, and H. Zhang, "Fog-based attack detection framework for internet of things using deep learning," IEEE Access, vol. 8, pp. 74571–74585, 2020.

[18] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system," Security and Communication Networks, vol. 2020, no. 1, p. 8890306, 2020.

[19] A. A. Wardana, G. Kołaczek, and P. Sukarno, "Lightweight, trust-managing, and privacy-preserving collaborative intrusion detection for internet of things," Applied Sciences, vol. 14, no. 10, p. 4109, 2024.

[20] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019.

[21] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the internet of things," IEEE Access, vol. 7, pp. 42450–42471, 2019.

[22] B. S. Khater, A. W. Abdul Wahab, M. Y. I. Idris, M. A. Hussain, A. A. Ibrahim, M. A. Amin, and H. A. Shehadeh, "Classifier performance evaluation for lightweight ids using fog computing in iot security," Electronics, vol. 10, no. 14, p. 1633, 2021.

[23] 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.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[25] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, "Security analysis of network anomalies mitigation schemes in iot networks," IEEE Access, vol. 8, pp. 43355–43374, 2020.