Hybrid Data Augmentation Based Machine Learning Approach for Botnet Attack Detection in IOT Networks

Fatma Zafar
M. Tech Scholar
Department of CSE
Oriental Group of Institutes,
Bhopal, M.P., India
fatmazafar000@gmail.com

Prof. Shivank Soni
Assistant Professor
Department of CSE,
Oriental Group of Institutes
Bhopal, M.P., India

Abstract: This paper presents a comprehensive approach to botnet detection in Internet of Things (IoT) networks through the development and evaluation of a Generative Adversarial Network (GAN) augmented machine learning model. The methodology encompasses a multi-step process, starting with data collection and pre-processing, including feature extraction, normalization, and handling missing values. To address the challenge of data imbalance, a novel application of GANs is proposed. For classification of network traffic into botnet and legitimate traffic is performed using xgboost. The performance of the proposed model is rigorously evaluated using the N-BaIoT dataset, demonstrating its effectiveness through high accuracy, precision, recall, and F1-score metrics. The results indicate significant improvements over existing models, showcasing the potential of the proposed methodology in enhancing IoT network security against botnet threats.

Keywords: - IoT Security, Botnet Detection, Data Augmentation, Generative Adversarial Networks, Machine Learning

I. INTRODUCTION

The widespread adoption of Internet of Things (IoT) devices has significantly improved convenience in our daily lives but also introduced substantial cybersecurity threats, notably botnet attacks. Botnets, networks of hijacked devices orchestrated by attackers, can launch devastating Distributed Denial of Service (DDoS) attacks and data breaches [1]. The continuous expansion of the Internet of Things (IoT) across various sectors has significantly improved efficiency and convenience but has simultaneously increased the vulnerability to cyberattacks, particularly botnet attacks. Botnets, formed by compromised IoT devices, pose a grave threat to the security and stability of IoT ecosystems by enabling cybercriminals to launch distributed denial of service (DDoS) attacks, steal data, and spread malware [2]. The critical importance of detecting IoT botnet attacks in the modern interconnected era is emphasized by the potential for these attacks to cause widespread disruption and significant damage. Botnets, networks of compromised devices orchestrated by malicious actors, pose a severe threat across various sectors including healthcare, transportation, energy, and smart cities, by enabling devastating cyberattacks that can lead to service disruptions, data breaches, and substantial financial losses. The vulnerability of many IoT devices, often lacking robust security features, underscores the necessity for early detection mechanisms to prevent malicious exploitation and the assembly of botnets [3]. Early detection not only allows for swift containment of threats but also facilitates the identification of vulnerabilities, leading to improved security measures. Additionally, the interconnected nature of IoT systems means that botnet attacks can have cascading effects, potentially disrupting critical services and infrastructure on a large scale. Effective detection and mitigation of such attacks are crucial not only for safeguarding infrastructure and preserving user trust in IoT technologies but also for ensuring the continued growth and integration of these technologies into everyday life. Traditional security solutions often fall short against these sophisticated and evolving threats, underscoring the urgent need for more advanced, adaptive detection methods [2].

Machine learning has risen as a key solution, offering the ability to analyze large data volumes, identify intricate patterns, and adjust to new attack strategies [3]. Unlike conventional methods that depend on predefined signatures, machine learning models learn from historical data to recognize new attack patterns, making them especially effective for the dynamic IoT landscape [4]. They can swiftly process and analyze data from IoT devices to spot anomalies signaling botnet attacks, thus enhancing IoT security. Machine learning employs various techniques for botnet detection, including supervised learning, which learns from labeled datasets of past attacks; unsupervised learning, which detects anomalies without pre-labeled examples, hence useful for identifying novel botnet behaviors; and reinforcement learning, which refines its performance by interacting with the environment [5].

However, this approach faces challenges, such as the scarcity of high-quality, real-world datasets for training, the risk of adversarial attacks designed to deceive the models,

and the complexity of interpreting these models' decisions. Addressing these issues is crucial for leveraging machine learning effectively against IoT botnet threats.

II. LITERATURE REVIEW

Z. Alothman et al. [1], investigated the vulnerability of IoT devices to botnet attacks. They developed a machine learning-based method to detect both normal and malicious traffic, specifically targeting IoT botnet attacks. Utilizing the Bot-IoT dataset and employing preprocessing techniques like SMOTE, they experimented with various classifiers. Among these, Random Forest (RF) and J48 classifiers demonstrated superior performance compared to MLP networks, achieving high accuracy. Abraham et al. [3] highlight the significant threat posed by botnets to network security, emphasizing their role in infecting devices with malware and forming networks under the attacker's control, leading to economic and social consequences. Their study focuses on developing an anomaly-based intrusion detection system using network traffic data from the Bro monitoring framework. Various supervised learning methods were compared for anomaly detection, with Random Forest performing best in traditional cross-validation. CD McDermott et al. [4], address the increasing prevalence of IoT-based DDoS attacks, proposing a novel solution for detecting botnet activity in consumer IoT devices and networks using Deep Learning techniques, specifically a Bidirectional Long Short-Term Memory-based Recurrent Neural Network (BLSTM-RNN). Their study compares the BLSTM-RNN model to a conventional LSTM-RNN in detecting Mirai botnet attack vectors, demonstrating that despite additional overhead and processing time, the bidirectional approach is more effective and progressive over time. They also mention the availability of a labeled dataset generated during their research upon request.

Injadat et al. [5], focus on the increasing threat of IoT malware attacks due to the rapid growth of IoT devices. They propose an optimized machine learning framework that combines the Bayesian optimization Gaussian Process (BO-GP) algorithm with a decision tree (DT) classification model to detect attacks on IoT devices efficiently and accurately. Evaluation on the Bot-IoT-2018 dataset demonstrates high detection accuracy, precision, recall, and F-score. TA Tuan et al. [6], emphasize the significant and complex threat of botnets in network traffic, particularly in various internet attacks like DDoS, spam, malware, and phishing. They evaluate the performance of machine learning methods on two well-known datasets, UNBS-NB 15 and KDD99, aiming to assist developers in selecting suitable approaches for computer security studies. T. Hasan et al. [7], address the vulnerability of Industrial Internet of Things (IIoT) infrastructure to sophisticated bot attacks, which can have catastrophic consequences. They propose a hybrid intelligent Deep Learning (DL)-enabled mechanism to secure IIoT infrastructure. AA Hezam et al. [8], address the growing threat of Distributed-Denial-of-Service (DDoS) attacks, particularly targeting Internet-of-Things (IoT) devices, which remain a significant risk despite various solutions. Their research proposes a deep learning (DL) approach utilizing recurrent neural network (RNN), convolutional neural network (CNN), and long short-term memory (LSTM)-RNN algorithms. Evaluation on a real-world N-BaIoT dataset infected with two dangerous DDoS botnets (Mirai and Bashlite) shows that RNN achieves the highest accuracy at 89.75%, highlighting the effectiveness of DL in countering DDoS attacks on IoT networks. Celil et al. [9], focus on enhancing IoT security by employing machine learning methods to accurately detect normal and attack network traffic. They utilize the N-BaIoT Provision 737E security camera dataset for analysis and apply both supervised and unsupervised learning approaches.

Desai et al. [10] highlighted the security concerns associated with IoT technology, especially regarding IoT botnet attacks like the Mirai attack. They propose a hybrid approach to detect botnet attacks, combining both supervised and unsupervised machine learning techniques. The unsupervised method is initially employed to detect attacks, and its outputs are utilized in the supervised machine learning method for classifying data into benign or attack traffic, enhancing the accuracy of botnet attack detection. Lee et al. [11], discuss the security challenges faced by smart factories in the industry 4.0 era, highlighting the threat of cyberattacks like botnets and Distributed Denial of Service (DDoS). They propose a novel approach that combines honeypots with machine learning for botnet attack classification. Experiments conducted in a simulated smart factory environment show high accuracy (above 96%), fast processing time (0.1 ms), and a low false positive rate (0.24127) using the random forest algorithm with the Weka machine learning program, demonstrating the feasibility and effectiveness of this approach for detecting botnet attacks in smart factory networks.

Bahşi et al. [12] addressed security concerns arising from the rapid growth of the Internet of Things (IoT) and the threat of cyber-attacks launched by large IoT botnets. They focus on detecting compromised devices, emphasizing the need for scalability, resource constraints, interpretability, and intrusion detection signature generation in IoT environments. Taher et al. [13] proposed a hybrid filter and wrapper selection approaches to select the most relevant features. The novel approach integrated with clustering rank the features and then applies the Grasshopper algorithm

(GOA) to minimize the top-ranked feature. The author applied NN for classification.

III. PROPOSED METHODOLOGY

The flowchart of the working are presented in figure 1.

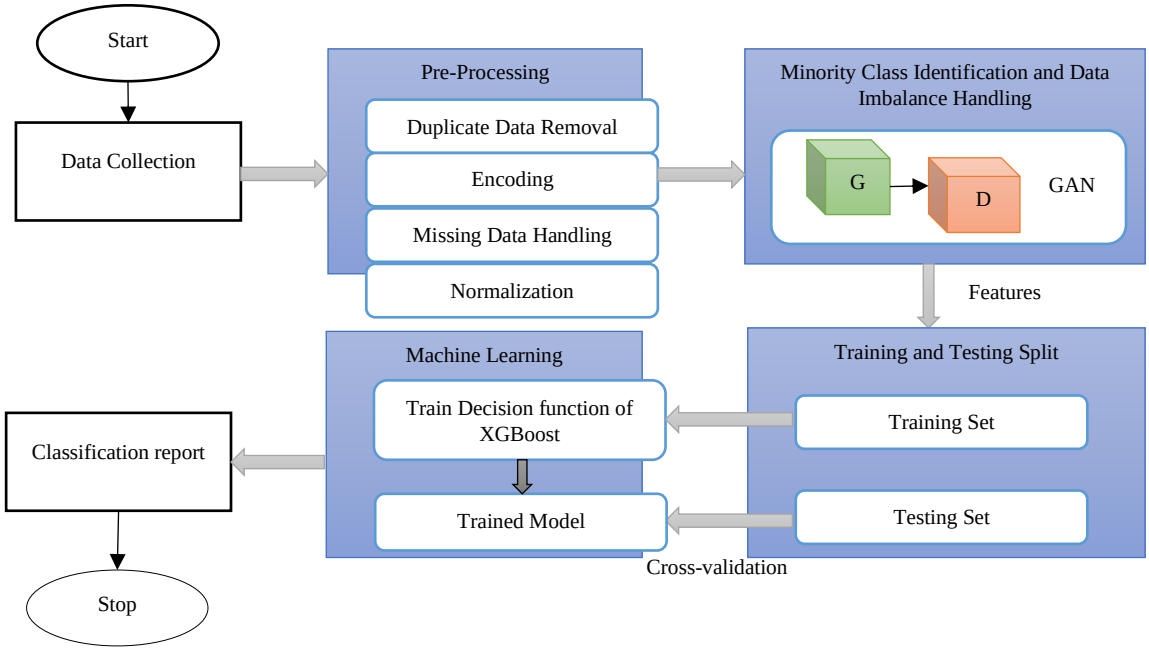

Figure 1. Flowchart of Botnet Detection in IoT Networks

A. *Data Collection*

In this step, dataset is collected from sources [14]. The data is collected after capturing raw network traffic data in pcap format via port mirroring to collect clean IoT

B. *Data Pre-processing*

Data pre-processing is a crucial step in the data analysis and machine learning pipeline. It involves cleaning and transforming raw data into a format that can be easily understood and utilized for analysis or model training. Fig 2 presents the detailed steps of pre-processing.

Figure 2. Pre-processing

Following key steps are involved during pre-processing:

Removing duplicates: Identifying and eliminating duplicate entries to prevent skewed analysis.

Label encoding: Assigns a unique integer to each non-numerical data.

Handling missing values: Depending on the context and significance of missing data, data imputation is applied to handle missing data. Imputation is a method used to handle missing values in a dataset by replacing them with substitute values, thereby allowing for more complete analysis without discarding data. For data imputation, K-Nearest Neighbors (KNN) is used. Missing values are imputed using the values of the k most similar instances (neighbors), based on other, non-missing attributes. The similarity between instances is usually measured using a distance metric such as Euclidean distance. The imputed value is the mean (for numerical variables) or mode (for categorical variables) imputation, K-Nearest Neighbors (KNN) is used. Missing values are imputed using the values of the k most similar instances (neighbors), based on other, non-missing attributes. The similarity between instances is usually measured using a distance metric such as Euclidean distance. The imputed value is the mean (for numerical variables) or mode (for categorical variables) of the k nearest neighbors.

Normalization: Scaling numerical features to a specific range, typically between 0 and 1, helps with model convergence and interpretation. For this z-score normalization is adopted stated as:

$$\text{Data}_{\text{i}} = \frac{x_{\text{i}} - \bar{x}}{std} \quad (1)$$

Where, x_i = The data value at instance i and $\bar{x}$= mean value

$$\bar{x} = \frac{1}{n}\sum_{\text{i}=1}^{n} x_{\text{i}} \quad (2)$$

std=standard deviation

$$std = \sqrt{\frac{1}{n}\sum_{\text{i}=1}^{n}(x_{\text{i}} - \bar{x})} \quad (3)$$

C. Data Imbalance Handling using Augmentation Approach

Generative Adversarial Networks (GANs) offer a novel approach to handling data imbalance, particularly in domains like image classification, where acquiring or generating more data for underrepresented classes can be challenging. GANs, introduced by Ian Goodfellow et al. in 2014, consist of two neural networks—the generator and the discriminator—that are trained simultaneously in a competitive manner [16]. This setup can be leveraged for data augmentation to balance datasets by generating synthetic data for the minority classes. An overview of how GAN-based augmentation works to address data imbalance:

Generator (G): This network learns to generate new data instances that mimic the real data. Initially, it produces data that might not closely resemble the target distribution, but it improves as training progresses.

Discriminator (D): This network learns to distinguish between real data instances and the fake instances produced by the generator. It gets better at telling real from fake as training progresses.

The two networks improve through their competition, with the generator striving to produce increasingly convincing data, and the discriminator getting better at distinguishing real from synthetic data.

Generator (G(z;θ_g) aims to generate data xˆthat resembles the actual data distribution of the minority class. It takes a random noise vector z as input and generates samples that mimic the real data distribution p_data. Here θ_g are the parameters of the generator network. Discriminator (D(x;θ_d)) tries to distinguish between the real data samples x from the minority class and the synthetic samples xˆ produced by the generator. It outputs a probability D(x;θ_d) that represents the likelihood of x being a real rather than a generated sample and θ_d are the parameters of the discriminator network. The training of GANs involves a min-max game objective function. The discriminator D aims to maximize V(D,G) so that it can correctly classify real and generated samples. The generator G aims to minimize V(D,G) so that D mistakenly believes generated samples are real.

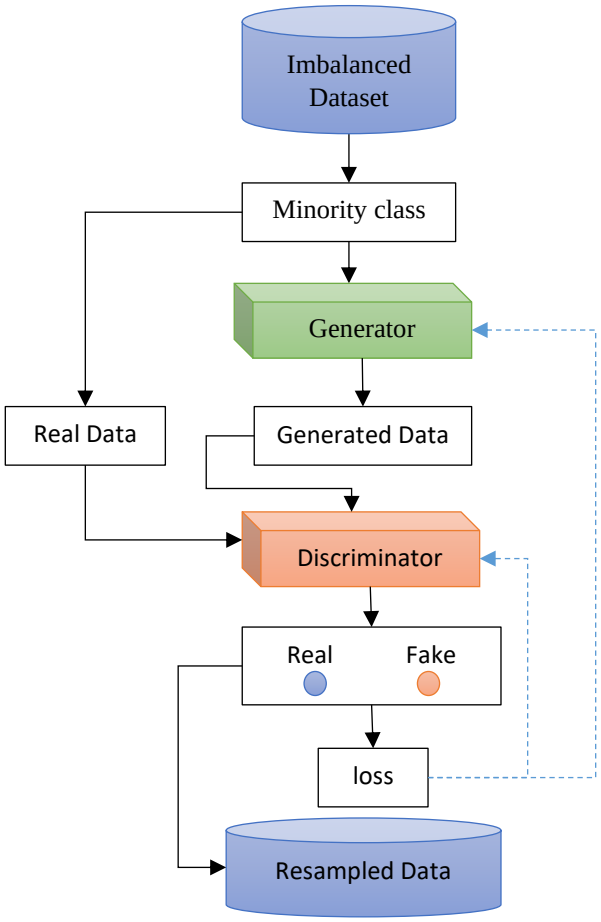

Figure 3: Data Augmentation for Data Imbalance Handling

D. Classification

In this stage, the dataset is split into two sub-sets i.e., training and testing. The selection of testing and training dataset is done on random selection. The testing data is different from training data and cross-validation is used. In this step, XGBoost (ensemble learning) is used to predict the attack on testing data. For a classification problem with K classes, the output of the XGBoost is given by: Y=⟦argmax⟧_(k∈{1,...,K}) ∑_(i=1)^N▒⟦I(h_i (x)=k)⟧ where N is the number of trees, h_i (x) is the prediction of the ith tree, and I is the indicator function. For a regression problem, the output is the average of all the tree outputs: Y=1/N ∑_(i=1)^N▒⟦h_i (x) ⟧ where h_i (x) is the prediction of the ith tree. Training Dataset: D={(X_1,Y_1),(X_1,Y_1),…..(X_n,Y_n)}

Number of Trees to be Created in the Boosted Model: N

Learning Rate (η): A value between 0 and 1 to shrink the feature weights after each boosting step, making the model more robust.

Start with a single model that predicts all the target values. For a classification task, it might involve calculating the log(odds) for the target classes.

For Each Tree Ti, where i = 1 to N:

Calculate the residuals (errors) by comparing the predicted values from the current model with the actual target values.

Use the residuals as the target values for building the next tree.

For each node in the tree, select the best split from all features based on an objective function. The objective function typically involves a combination of the loss function.

Grow the tree to a maximum depth specified as a parameter.

No sub-sampling of features; instead, use all features to determine the best split.

Add the new tree to the model with a weight α determined by the learning rate η and the objective function's optimization.

Remove splits that have a minimal impact on the objective function, using a complexity parameter that penalizes the number of leaves and the splits' similarity.

Repeat the steps for each new tree, each time fitting to the residuals of the previous model's predictions.

For classification tasks, the output is the class label that received the majority vote. For regression tasks, the output is the average of all the tree outputs.

IV. RESULTS AND DISCUSSION

A. Dataset Used

The N-BaIoT dataset [14], which includes data samples with 115 features, has been extensively utilized in previous research for botnet detection within IoT and IIoT environments, as referenced in multiple studies. Its application across these studies has demonstrated high performance in botnet detection. This dataset supports both multiclassification and binary classification tasks, making it a versatile tool for identifying botnet activities. The data within the N-BaIoT dataset was meticulously collected by mirroring the ports of IoT devices and ensuring the network was properly configured to gather benign data accurately. The consistent use and proven performance of the N-BaIoT dataset in related work highlight its significance and suitability for botnet detection research, contributing valuable insights and facilitating advancements in securing IoT ecosystems.

B. Performance Evaluation Measures

To evaluate the proposed algorithm, it is concentrated on three indications of performance:

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)*100 \quad (1)$$

$$\text{Precision} = TP/(TP+FP)*100 \quad (2)$$

$$\text{Recall} = TP/(TP+FN)*100 \quad (3)$$

$$\text{F\_Measure} = 2*\text{Precision}*\text{Recall}/(\text{Precision} + \text{Recall}) \quad (4)$$

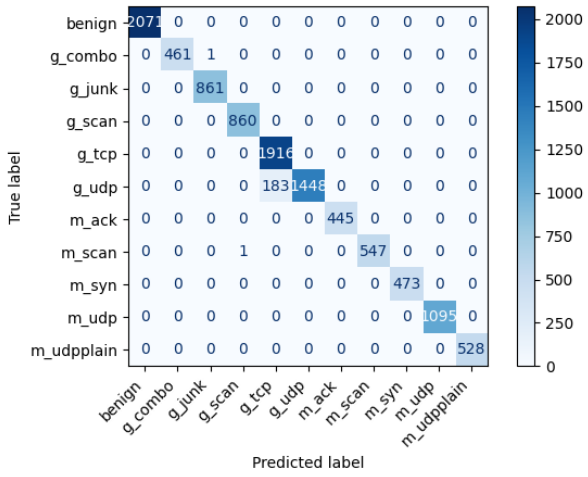

Figure 4. Confusion Matrix for Botnet Detection.

Normalization: Scaling numerical features to a specific range, typically between 0 and 1, helps with model convergence and interpretation. For this z-score normalization is adopted stated as:

The performance evaluation of a proposed hybrid GAN augmented machine learning approach for botnet detection is presented in table 1. With an accuracy of 98.4%, precision at 99.1%, recall at 98.8%, and an F1-score of 98.8%, the system demonstrates an outstanding ability to accurately identify botnet activities with minimal false positives and negatives. These metrics indicate that the model is not only highly effective in distinguishing between botnet and legitimate traffic but also ensures that almost every botnet threat is detected and correctly classified. This balance between precision and recall, as evidenced by the high F1-score, highlights the system's reliability and efficiency in mitigating botnet threats, making it a promising solution for network security. The proposed method demonstrates a comprehensive improvement over the existing method [13] across all key performance metrics: accuracy, precision, recall, and F1-Score. With a increase in accuracy (0.4%) and recall (0.1%), the most notable enhancement is observed in precision, where the proposed method outperforms the existing one by a significant 2.1%. Furthermore, the improvement in the F1-Score by 1.0% suggests a better balance between precision and recall. Fig 4 presents the confusion matrix for 10 classes of attack classification with normal network traffic. Fig 5 presents the comparative analysis of proposed model with existing model [13]. The proposed model shows 0.4% improvement in accuracy and approx. 2% improvement in precision and 1% in f1-score over existing work.

Table 1. Performance Evaluation

Parameters	Existing [13]	Proposed
Accuracy	98.0	98.4
Precision	97.0	99.1
Recall	98.7	98.8
F1-Score	97.8	98.8

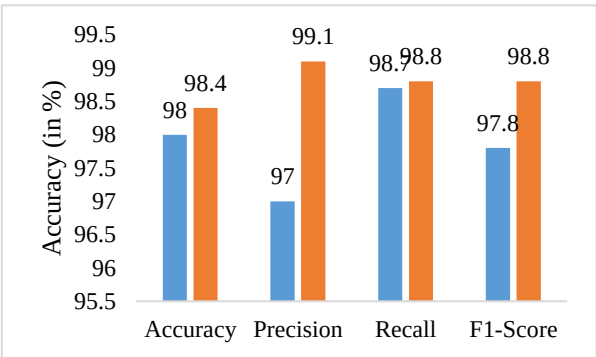

Figure 5. Comparative Analysis for Accuracy

IV. CONCLUSION

The research introduces a novel botnet detection framework for IoT networks, leveraging the strengths of GANs for data augmentation and machine learning for traffic classification. The proposed methodology effectively addresses the critical challenges of data imbalance and feature extraction in the

context of IoT security. By employing a GAN-augmented approach, the model significantly improves the quality and quantity of training data, enabling more accurate and reliable botnet detection. The empirical evaluation using the N-BaIoT dataset highlights the model's superior performance, with notable improvements in accuracy, precision, recall, and F1-score compared to existing approaches. These outcomes affirm the efficacy and efficiency of the proposed model in identifying and mitigating botnet threats within IoT networks. In future this work will be extended on new attacks on distributed IoT networks.

Conflict of Interest: The corresponding author, on behalf of all authors, confirms that there are no conflicts of interest to disclose.

REFERENCES

[1] Alothman, Zainab, Mouhammd Alkasassbeh, and Sherenaz Al-Haj Baddar. "An efficient approach to detect IoT botnet attacks using machine learning." Journal of High Speed Networks 26.3 (2020): 241-254.

[2] Hussain, Faisal, et al. "A two-fold machine learning approach to prevent and detect IoT botnet attacks." Ieee Access 9 (2021): 163412-163430.

[3] Abraham, Brendan, et al. "A comparison of machine learning approaches to detect botnet traffic." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.

[4] McDermott, Christopher D., Farzan Majdani, and Andrei V. Petrovski. "Botnet detection in the internet of things using deep learning approaches." 2018 international joint conference on neural networks (IJCNN). IEEE, 2018.

[5] Injadat, Mohammad Noor, Abdallah Moubayed, and Abdallah Shami. "Detecting botnet attacks in IoT environments: An optimized machine learning approach." 2020 32nd International Conference on Microelectronics (ICM). IEEE, 2020.

[6] Tuan, Tong Anh, et al. "Performance evaluation of Botnet DDoS attack detection using machine learning." Evolutionary Intelligence 13 (2020): 283-294.

[7] Hasan, Tooba, et al. "Securing industrial internet of things against botnet attacks using hybrid deep learning approach." IEEE Transactions on Network Science and Engineering (2022).

[8] Hezam, Abdulkareem A., et al. "Deep learning approach for detecting botnet attacks in IoT environment of multiple and heterogeneous sensors." International Conference on Advances in Cyber Security. Singapore: Springer Singapore, 2021.

[9] Celil, O. K. U. R., and Murat DENER. "Detecting IoT Botnet attacks using machine learning methods." 2020 International Conference on Information Security and Cryptology (ISCTURKEY). IEEE, 2020.

[10] Desai, Madhuri Gurunathrao, Yong Shi, and Kun Suo. "A hybrid approach for IoT botnet attack detection." 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2021.

[11] Lee, Seungjin, et al. "Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning." PeerJ Computer Science 7 (2021): e350.

[12] Bahşi, Hayretdin, Sven Nõmm, and Fabio Benedetto La Torre. "Dimensionality reduction for machine learning based iot botnet detection." 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE, 2018.

[13] Taher, Fatma, et al. "Reliable Machine Learning Model for IIoT Botnet Detection." IEEE Access (2023).

[14] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, ‘‘N—BaIoT-network-based detection of IoT botnet attacks using deep autoencoders,’’ IEEE Pervasive Comput., vol. 17, no. 3, pp. 12–22, Sep. 2018.