International Journal of Mathematical, Engineering and Management Sciences
Vol. 10, No. 2, 537-553, 2025
https://doi.org/10.33889/IJMEMS.2025.10.2.027

Ram Arti Publishers

Developing an Intelligent System for Efficient Botnet Detection in IoT Environment

Ramesh Singh Rawat
Department of Computer Science and Engineering,
Graphic Era (Deemed to be University), Dehradun, Uttarakhand, India.
E-mail: rsrawat@geu.ac.in

Manoj Diwakar
Department of Computer Science and Engineering,
Graphic Era (Deemed to be University), Dehradun, Uttarakhand, India.
Corresponding author: manoj.diwakar@gmail.com

Umang Garg
Computer Science and Engineering,
MIT Art Design and Technology University, Pune, India.
E-mail: umangarg@gmail.com

Prakash Srivastava
Department of Computer Science and Engineering,
Graphic Era (Deemed to be University), Dehradun, Uttarakhand, India.
E-mail: prakash2418@gmail.com

(Received on October 1, 2024; Revised on November 26, 2024 & December 16, 2024; Accepted on December 18, 2024)

Abstract
Smart technological instruments and Internet of Things (IoT) systems are now targeted by network attacks because of their widespread rising use. Attackers can take over IoT devices via botnets, and pre-configured attack vectors, and use them to do harmful actions. Thus, effective machine learning is required to solve these security issues. Additionally, deep learning with the necessary elements is advised to defend the network from these threats. In order to achieve proper detection of hacks in the future, relevant datasets must be used. The device's operation could occasionally be delayed. The sample dataset must be well structured for training the model and validating the suggested model to create the best protection system model feasible for detecting cyber risks. This paper focused on analyzing botnet traffic in an IoT environment using machine learning and deep learning classifiers: Decision tree classifier, Naïve Bayes, K nearest neighbor, Convolution neural network, Recurrent neural network, and Random Forest. We calculated each algorithm's Accuracy, True Positive, False Positive, False Negative, True Negative, Precision, and Recall. We obtained impressive results using these CNN, and LSTM RNN classifiers. We have also achieved a high attack detection rate.

Keywords- IoT, IoT botnet, Secured system, Machine learning.

1. Introduction

The demand for cybersecurity has increased due to the fast rise of Internet of Things (IoT) devices, computer network development, and various applications. Malicious software attacks are increasing, leading to the creation of botnets, which can disrupt daily activities. According to a report in 2021, about 350,000 new malware and potentially unwanted software are registered daily by the AV-TEST institute. The extensive availability of malware can create botnets, enabling cybercriminals to exploit the collective bandwidth of thousands, if not millions, of infected devices to disrupt the day-to-day operations of governments and businesses.

Botnets are networks of infected computers managed by botmasters, used for intrusion attacks like DDoS, click fraud, flooding, or spamming. Detecting botnets is a critical challenge due to the global scale of botnet-assisted attacks. Many approaches have been put forth, such as machine learning techniques, which necessitate feature extraction before the training or learning of ML models.

Three structural types of botnets exist: mixed, distributed, and central. Distributed structures employ non-central P2P mode, while central structures use client-server (C/S) mode. By combining dispersed and central structures, mixed structures improve the robustness and efficiency of communication. To control botnets, several structures are necessary (Wei et al., 2024). The article offers a thorough framework for anomaly detection methods utilizing SVM and random forest algorithms for botnet attack analysis and detection. By utilizing communication patterns, packet flow, and traffic volume, this technique enhances network security (Al-Fawa'reh et al., 2024).

Network traffic analysis is used in both supervised and unsupervised methods to find anomalies. This is a novel method that combines several Decision Trees that have already been processed, along with supervised machine learning algorithms and a random forest. Accuracy and computation time are assessed and tested for the Anomalies Detection System (ADS). Due to the increase in network attacks, which will highlight the vulnerability of all the attacked devices, to combat this problem an exploring machine learning (ML) as a potential solution has been developed. This will be proposing an optimized ML-based framework, combining the Bayesian optimization Gaussian process and decision tree classification models, to detect botnet attacks on IoT devices (Gurevin et al., 2024).

Botnets are a serious security risk as they use sophisticated control methods and are constantly evolving on a worldwide scale. Hacking groups have developed bots that can remotely take over compromised networks, spread malware, send unsolicited emails, and steal confidential data. Approximately 80% of all Internet traffic is associated with botnet activity. Bots in botnets access server IP addresses using Domain Name Service (DNS). The efficiency of a bonnet detection approach utilizing DNS query data is evaluated in this study using machine learning algorithms. Monitoring DNS query data can uncover harmful activity.

Figure 1 shows the IoT botnet scenario with infected devices, C&C server, scanner, and attack scenario. It presents the workflow of an IoT botnet attack. A client identified by IP address runs under proxy to mask its identity and then connects to the C&C server. The client connected to the C&C server works as scanner, and search for vulnerable devices- like cameras, routers, sensors, etc. Further, these devices are infected with malicious software and controlled by the C&C server once compromised. The infected devices are then used to launch a DDoS attack against the chosen target by sending enormous traffic to it. These interconnected devices and components highlight botnet's layered structure and hierarchical design.

Botnets are complicated malware networks that use command-and-control servers to function under the supervision of botmasters, resulting in substantial financial losses. Because of their versatility, detection measures like masking and community-based procedures are useless. ML classification techniques are effective in network security and intrusion detection, making them ideal for detecting botnets. However, optimizing these models is crucial to ensure their maximum capacity, as default versions may not be sufficient for optimal performance (Alshamkhany et al., 2020). Internet security is crucial as cyberattacks can lead to data loss and unauthorized access to computer systems, networks, and devices. Botnets, which are computer networks infected with malware, are becoming a significant risk against cybersecurity. Botmasters, who control these networks, use C&C channels to create and manage botnets, which are an army of bots (Kalakoti et al., 2022). Botnets are compromised and controlled by a hacker, acting as a foot soldier for their botmaster. They pose a significant threat to information security, with botmasters constantly improving their skills.

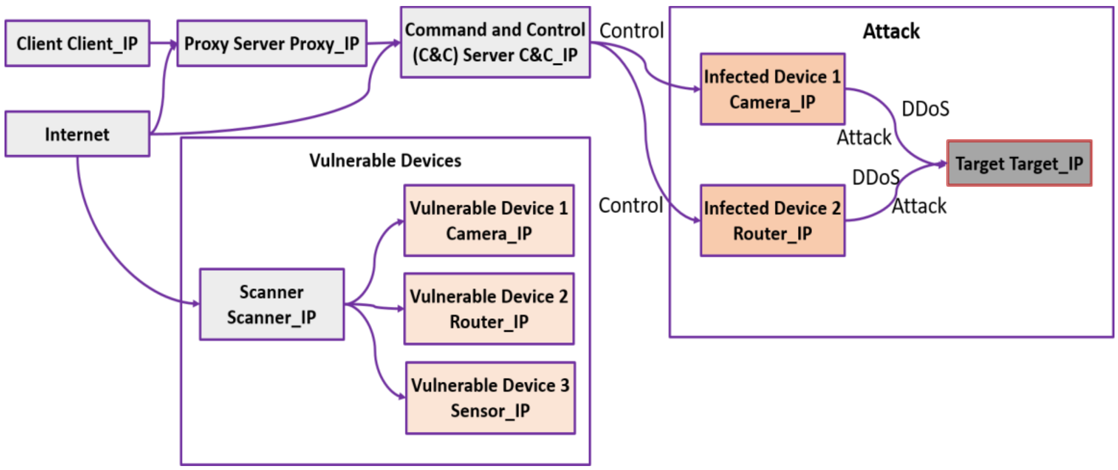

Figure 1. IoT botnet scenario.

The present detection methods face challenges due to the continuous evolution of botnets (Rawat et al., 2018). In this paper, malware-infected IoT devices connected to the Internet and leading to botnet attacks are detected using Machine Learning techniques. The proposed techniques allow for more accurate detection and classification of botnets, which is significant in computer security with the rise of botnets among devices. The increasing number of devices within the IoT structure leave them susceptible to cyberattacks. Machine learning (ML) can identify these threats, but high-dimensionality data can affect performance. So, this paper investigates the effect left by wrapper and hybrid based feature selection techniques on Machine Learning models for Internet of Things (IoT) botnet detection.

Many IoT devices are vulnerable to cyber-attacks and some traditional intrusion detection systems are not fully reliable (Chen et al., 2014). A machine learning-based botnet attack detection architecture with several algorithms for botnet attack detection has been proposed, but due to its complexity and difficulty to collect common normal data, it becomes troublesome to implement. One aim of this study is to find a solution of serious issue of protection against botnets, using an anomaly-based botnet detection system. The method uses IP headers to distinguish between botnet DNS requests (Ibrahim et al., 2021). The machine learning-based botnet detection techniques aim to alleviate the impact of malicious acts. Further, it is essential to maintain confidentialities, integrity, and availability as Internet services become more and more dependent on them.

However, IoT botnet is one of the critical attacks in the IoT environment. So, it is essential to implement the best practices and add security features for the improvement of overall security. Additionally, an efficient and intelligent mechanism is required to analyze the IoT network traffic for better security aspects. The major contributions included in the current article are as follows:
- To design and develop an efficient system for the detection of IoT botnet traffic.
- To apply a feature selection mechanism utilized in model building.
- To optimize and validate the results by using the distinct performance parameters.

The remaining paper is organized in the following sections such as Section 2 focuses on the recent contribution in the field of IoT botnet detection. Section 3 proposes a secured methodology with the implementation approach and configuration. Section 4 discusses about the experimental testbed details and results achieved from the experimentation. Finally, Section 5 concludes the research with significant contributions.

2. Related Work

Botnet architectures can be categorized into three structures: central, distributed, and hybrid. Central structures use a client-server model, while distributed structures use a non-central P2P model. Hybrid architectures combine central and distributed structures, enhancing communication efficiency and robustness. These structures are essential for controlling botnets (Rawat et al., 2018). This study presents a machine learning-based approach for detecting botnet attacks; demonstrating the effectiveness of support vector machines (SVM) and random forests (RF) in accurately identifying anomalies. The approach includes data pre-processing, feature extraction, and model training (He and Sayadi, 2023).

Venkatasubramanian et al. (2023) proposed a system that is based on an efficient botnet detection system using network traffic flow analysis and machine learning, focusing on early detection and optimization using the Weka Toolbox and CTU-13 dataset. Hasan et al. (2022) proposed a system that discusses the framework having an enhanced machine learning framework for identifying botnet assaults on IoT, that will beat the default decision tree and support vector machine models in terms of accuracy, precision, recall, and F-score. It integrates Bayesian optimization with Gaussian processes and decision tree classification models. Its ability to discriminate between legitimate and malicious occurrences demonstrates its efficacy and resilience in IoT settings.

Kalakoti et al. (2022) proposed a system that discusses the ongoing development of botnets that represent a danger to information security. The Signature-based and IDS-based solutions do not work in some cases. By using the DNS query data, a machine learning-based model for botnet identification is created. There are two stages to the model's implementation i.e. training and detection. Machine learning methods are used to extract and preprocess domain names during the training phase. DNS requests are tracked throughout the detection phase and categorized by the classifier.

Memos et al. (2022) proposed a system that assesses machine learning techniques for identifying peer-to-peer botnets. The system/model uses CTU-13 dataset. It highlights the importance of a number of attributes and elements and underlines the strength of ensemble methods. Güven and Gürkas-Aydin (2023) proposed a robust early-stage botnet detection method for combating cybersecurity threats, related to Distributed Denial of Service (DDoS) attacks, malware spreading, phishing and identity theft. The approach employs PCA and IG to extract the optimal features, which are subsequently inputted into several machine learning classifiers: Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), and Multilayer Perceptron (MLP).

Salim et al. (2022) proposed a blockchain based framework in which they describe the employment of machine learning for botnet attack detection using four classifiers Naïve Bayes, K-Nearest neighbor, Support vector machine, and Decision trees. Among various models, Decision trees performed the best with 99.89% accuracy, precision, recall and F1-score. Benlloch-Caballero et al. (2023) proposed a cognitive close loop system that battles against the DDoS attacks. The system uses machine learning for detecting botnet attacks in IoT networks, that focus on timely intrusion detection due to large-scale attacks. It uses wrapper techniques and filter methods to optimize performance. The system discovers that the system is effective against different topologies and finds that hybrid feature selection, combining Fisher's score and wrapper methods, achieves high detection rates while reducing computational demands.

Ngo and Nguyen (2022) introduced a system that describe an increasing concern of cyber-attacks targeting IoT equipment. They authors highlighted the need for network intrusion detection as a preventative step. Authors first discussed the problems of existing detection systems, especially those based on attack signatures, and used a detection architecture to demonstrate the practical issues. Wang et al. (2023) introduced a new botnet detection system that detects botnet behavior by analyzing the DNS traffic. It captures the botnet behavior with real-time network environment and therefore the anomaly based system accurately segregates the botnet and genuine traffic which optimizes the network security. Schumacher et al. (2023) applied machines learning techniques in botnet detection to identify botnet behaviors. The authors analyzed their pros and cons, performance in identifying botnet behavior, and clustering methods. Garg et al. (2023) proposed INFRDET a traffic analysis and regulators based system for botnet infection detection in the IoT network.

3. Proposed Methodology

By integrating state-of-the-art analytics with robust security protocols, the secure predictive demand forecasting system marks a major milestone in supply chain management. The system not only addresses critical data privacy and cybersecurity challenges but also enhances data accuracy and processing speed, enabling businesses to make informed choices, streamline resource allocation, and ultimately improve the overall efficiency of their operations (Hasan et al., 2022; Garg et al., 2024). The proposed framework contributes to the fields of demand forecasting, cybersecurity, and predictive analytics, offering a robust solution to the complex challenges faced by modern enterprises. The proposed methodology, along with the steps used in the current research, is illustrated in **Figure 2**. For the conversion of data into standard format, pre-processing steps need to be applied after selection of the botnet dataset. The process of preparing raw data for the machine learning model which includes the cleaning, transforming, and organizing raw data for the modeling and analysis is called data preprocessing (Bhale et al., 2023). This includes addressing missing values, normalizing features, encoding categorical variables, feature selection, and splitting data into training, validation, and test sets. We split the complete dataset in the ratio of 80:20 i.e 80% data for training and 20% data for test. We trained the models with three supervised machine learning algorithms such as logistic regression, decision tree classifier, LGBM, and neural network. Then the neural processing mechanism can be understood as a hybrid mechanism which gives specific the integration of CNN and LSTM for model building. After fitting the model, the algorithm uses the training and testing dataset, further analyzes it based on the parameters, and then evaluates the performance of the model accordingly. Finally, the test data is applied to the model to evaluate the predicted value and segregation of the benign and attacked traffic.

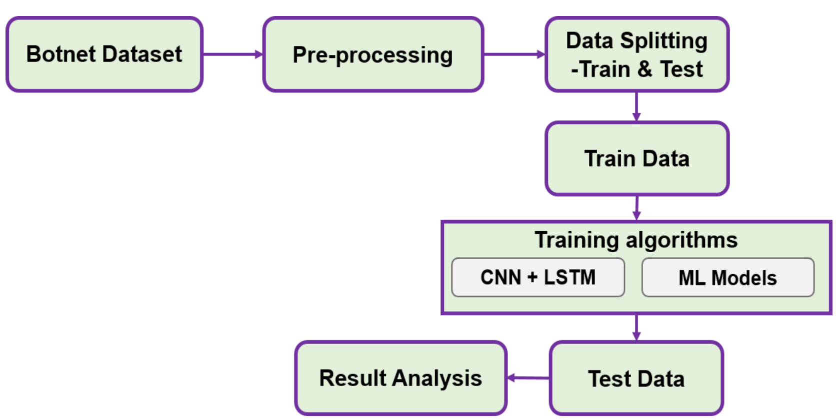

Figure 2. Proposed methodology.

The hybrid model is shown in **Figure 3**. It is a combination of CNN and LSTM that combines the spatial pattern recognition abilities of CNN with the temporal pattern recognition abilities of LSTM to provide a powerful unified model for processing sequential and spatial data. The local patterns and spatial relationship of images are captured using CNN, which is further used to process these patterns in feature detection like edges, textures, and other spatial information. Further, LSTM networks are a kind of RNN and are used to model sequential data. In fact, it works well at memorizing long-term dependencies and patterns in sequences and decreases the problem of vanishing gradient that you see with regular RNNs.

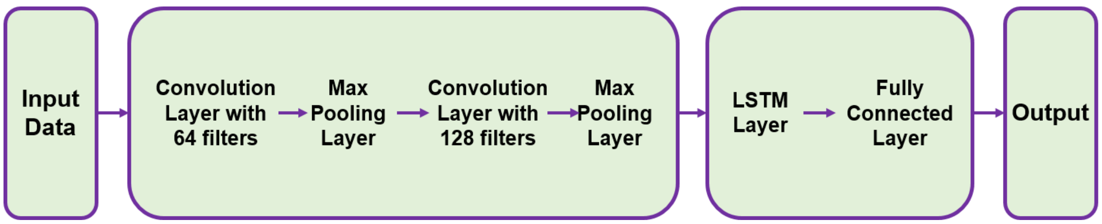

Figure 3. Hybrid model.

4. Experimental Testbed and Results

The setup of N-BaIoT model aims to evaluate the performance of anomaly detection models in the context of IoT network security (Meidan et al., 2018; Kalakoti et al., 2022). The experimental setup and configuration for the N-BaIoT dataset follows the basic steps such as:

- ***Data collection and preparation* (*IoT Devices*):** Identify the IoT devices used in the N-BaIoT dataset. A network environment is established that includes these IoT devices and a range of network traffic types. Collect both normal and malicious traffic data. The malicious traffic includes various types of attacks such as DoS and DDoS.

- ***Data preprocessing*:** Extract the features from the raw IoT network traffic. These features can include packet size, rate, inter-arrival times, source/destination IP addresses, protocols, etc. Normalize or standardize the features to ensure uniformity and reduce the influence of outliers. Finally, assign labels to the data indicating whether it is normal or an attack, and categorize the type of attack (e.g., DDoS, port scan, etc.). The normal instances mark as 0 and each attack type assigned with a unique integer as 1, 2, etc. Feature selection is done by using recursive feature elimination and select top 10 features out of all features. After feature selection, class balancing is an important step that used to prevent the model from being biased towards the frequent labels.

- ***Experimental environment setup*:** Set up the hardware infrastructure (e.g., servers, network equipment) and software (e.g., virtualization tools, network monitoring software). Design a network topology that simulates real-world IoT environments, potentially including switches, routers, and IoT devices. Simulate various attack scenarios as per the dataset to mimic real-world threats.

- ***Model development*:** Choose the type of models to use, like Support Vector Machines (SVM), Random Forest, Neural Networks, or Deep Learning models. Train the models using the preprocessed dataset, applying techniques such as cross-validation to ensure robustness. The proposed model is built by using 19 deep layers consisting of 16 Conv layers with 3x3 filters, 5 MaxPool layers, 3 fully connected layers, and 1 SoftMax layer. To fine-tune the hyper-parameters and maximize performance, modify the model's parameters.

• ***Evaluation of Model and Testing*:** Reserve some data for testing. This set is used to evaluate the generalizability of the model and as such must not be seen during training. For evaluating your models, use standard metrics like ROC-AUC, F1-score, accuracy, precision, and recall. Investigate the model in instances it misclassifies to identify trends or similarities. Repeat steps and check the results, looking for patterns or discrepancies to understand model performances.

4.1 Testbed Details

These experimental testbeds represent a typical setup and configuration for analysis with the N-BaIoT dataset. The N-BaIoT dataset is used for developing a method for detecting abnormal behavior for IoT devices. The dataset is commonly used in the domain of intrusion detection and malicious behavior analysis in the IoT networks are illustrated in **Figure 4**. The N-BaIoT contains network traffic data collected from different IoT devices. It features data from different IoT devices, including smart cameras, smart appliances, and other smart devices. The dataset consists of both normal (legitimate) and malicious traffic, and it is useful for researchers to study the characteristics of different types of attacks on IoT devices. The dataset contains data of various types of attacks, such as Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), and others, offering a comprehensive view of common attack vectors against IoT devices. It contains feature-rich network traffic data with corresponding labels defining whether the traffic is normal or malicious, along with the type of attack.

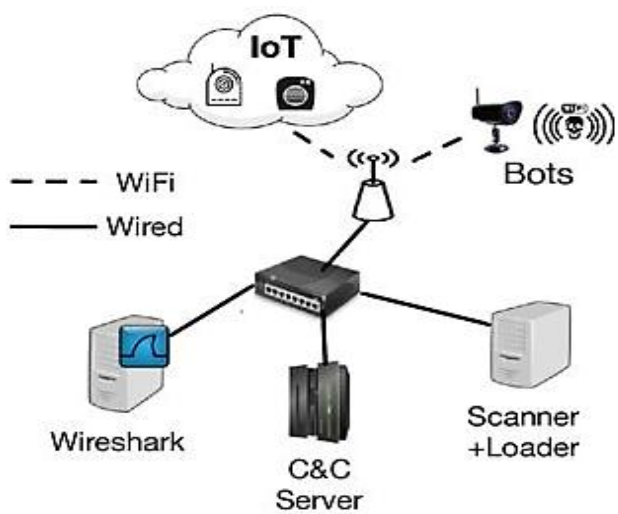

Figure 4. N-BaIoT dataset configuration.

4.2 Results Analysis

This section summarizes the unique outcomes accompanied by their analysis yielded from applying the proposed method on the N-BaIoT dataset. The analysis of different performance metrics like confusion matrix, ROC curve, precision-recall curve, and learning curve provides valuable insights for evaluating and enhancing the model's prediction accuracy and generalization ability. The classification model is evaluated using the performance confusion matrix, which gives a detailed comparison of actual vs predicted outcomes. So, it is being primarily used to check the performance of the model based on accuracy, precision, recall and the F1-score. ROC curve is a graphical representation of a binary classification model performance at different threshold values. The True Positive Rate (TP) vs False Positive Rate (FP) curve was generated to give insights into the trade-offs between sensitivity and specificity. The precision-recall curve is a graphical representation that allows for the evaluation of a binary classification model's performance. A learning curve is a plot that shows how a model's performance improves with more training data or experience. It is to describe the model skill in terms of underfitting, overfitting, and optimal learning, and to illustrate the relationship between model learning and model skill. **Figure 5** displays the class distribution of the dataset, which is divided into three separate classes 0, 1, and 2.

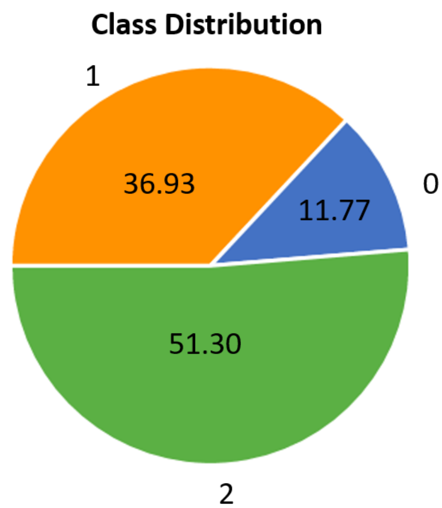

Figure 5. Benign and malware traffic class distribution.

A decision tree classifier operates by recursively partitioning the dataset into subsets based on feature values, leading to a tree-like structure where each node represents a decision based on a specific feature, and each leaf node represents a class label. There are distinct outcomes generates with decision tree classifier such as accuracy: 0.99, precision: 0.985, recall: 0.988, f1 score: 0.989. **Table 1** shows the classification report for decision tree classifier.

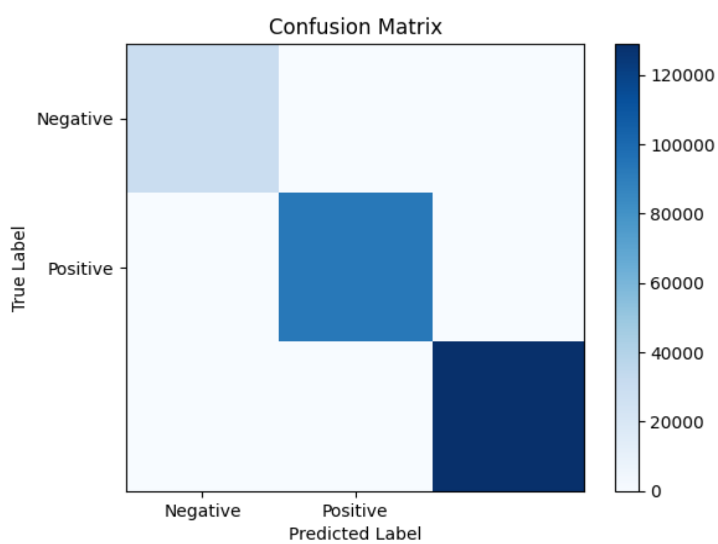

Figure 6. Confusion matrix for decision tree classifier.

Table 1. Classification report for decision tree classifier.

Parameters/ classes	Precision	Recall	F1-score	Support
Class 0	0.985	0.982	0.986	29455
Class 1	0.982	0.981	0.987	92752
Class 2	0.983	0.988	0.988	128860
Accuracy			0.989	251068
Macro average	0.986	0.983	0.982	251068
Weighted average	0.984	0.987	0.983	251068

Figure 6 shows the confusion matrix with true and predicted labels for decision tree. The result shows that good number of true values are achieved with higher accuracy. **Figure 7** shows the ROC curve that indicates the area under curve is 1 for distinct class classification.

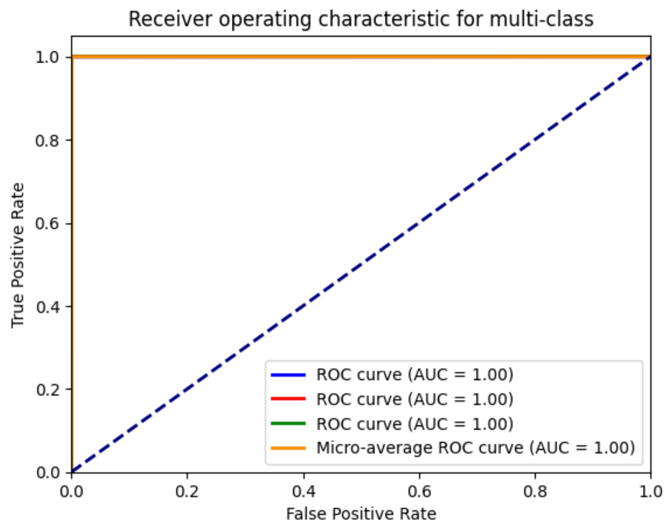

Figure 7. ROC curve for decision tree classifier.

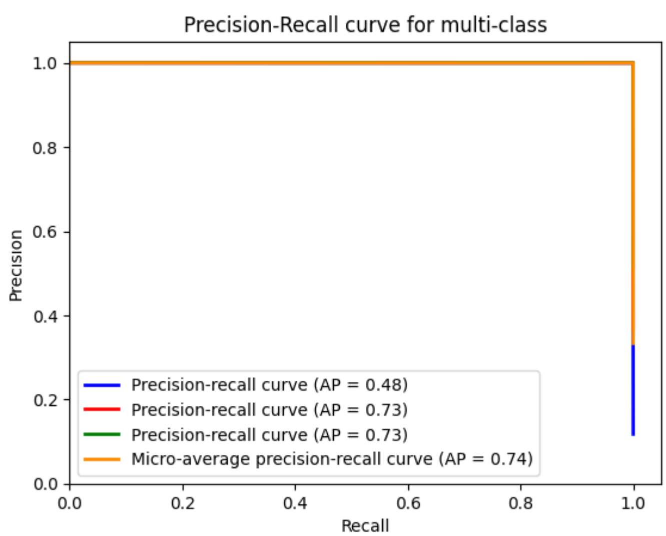

Figure 8. Precision-recall curve for multiclass classification.

Figure 8 shows the precision-recall curve that indicates the average precision values for distinct classes.

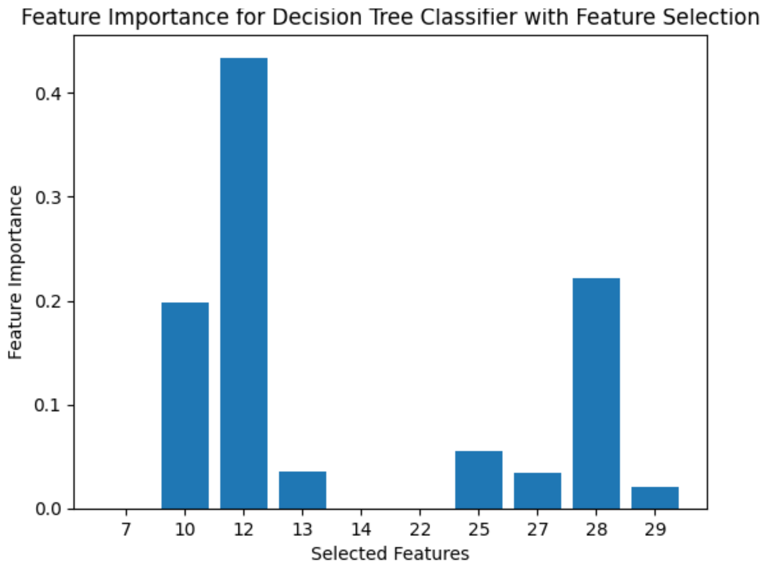

Figure 9. Selected features for decision tree classifier.

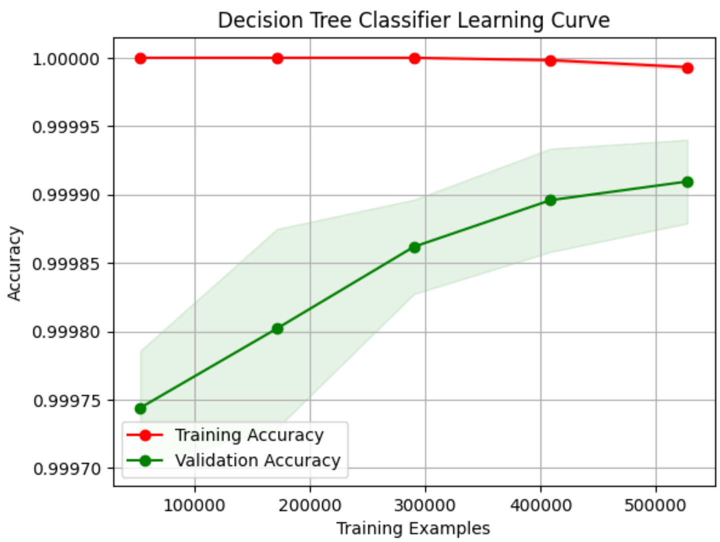

Figure 10. Decision tree classifier learning curve.

The **Figure 9** shows the feature importance values for a Decision Tree Classifier after feature selection. Each bar represents a specific feature and its importance score on the y-axis, indicating how much it contributes to the model's decision-making process. Feature 12 has the highest importance at over 0.4, suggesting it plays a significant role in classification. Features 10 and 28 also have notable contributions, while the other selected features (7, 13, 24, 25, 27, 29) have lower importance. **Figure 10** shows the learning curve for training and validation accuracy. It indicates that validation accuracy is little bit down as compared to the training accuracy.

Logistic Regression is utilized for binary classification tasks. The model provides the probability that a given input belongs to a specific class, typically using a logistic function to map linear combinations of features to probabilities between 0 and 1. **Table 2** shows the classification report for logistic regression with distinct performance parameters. There is distinct performance parameters retrieved from the logistic regression such as accuracy: 0.978, precision: 0.910, recall: 0.978, f1 score: 0.977.

Table 2. Logistic regression classification report.

Parameters/ classes	Precision	Recall	F1-Score	Support
Class 0	0.910	0.972	0.971	29455
Class 1	0.900	0.973	0.972	92753
Class 2	0.910	0.978	0.977	128860
Accuracy			0.978	251068
Macro average	0.910	0.975	0.973	251068
Weighted average	0.900	0.976	0.973	251068

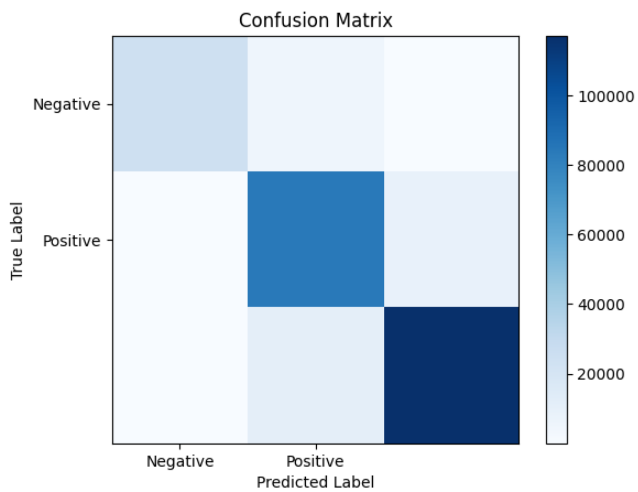

Figure 11. Logistic regression confusion matrix.

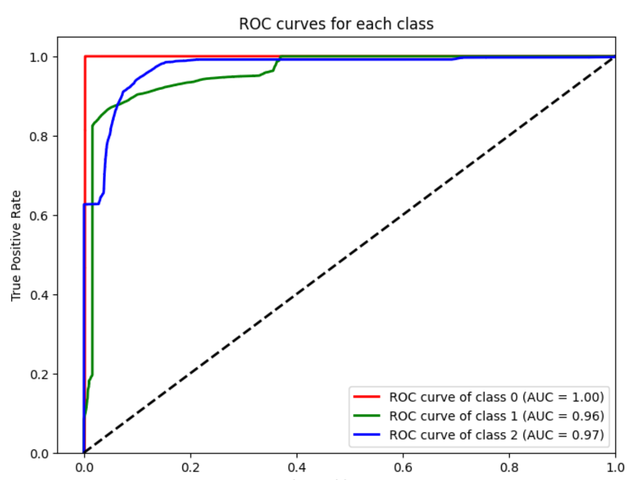

Figure 12. ROC curve for class 0, 1, and 2.

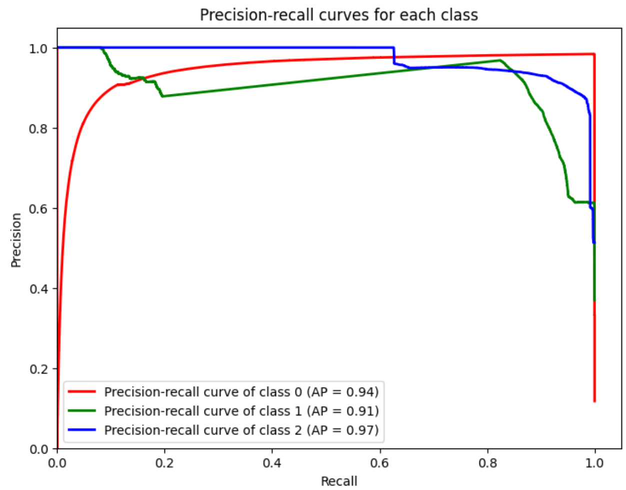

Figure 13. Precision-Recall curve of logistic regression.

Figure 11 shows the confusion matrix for logistic regression with better true positives. **Figure 12** shows the ROC curve for three classes 0, 1, and 2. The area under curve is near to 1 which indicates the better prediction of model. **Figure 13** shows the precision-recall curve with average precision 0.94. **Figure 14** shows the learning curve for training score and cross-validation score for logistic regression. The graph is indicating the overlapping of the line.

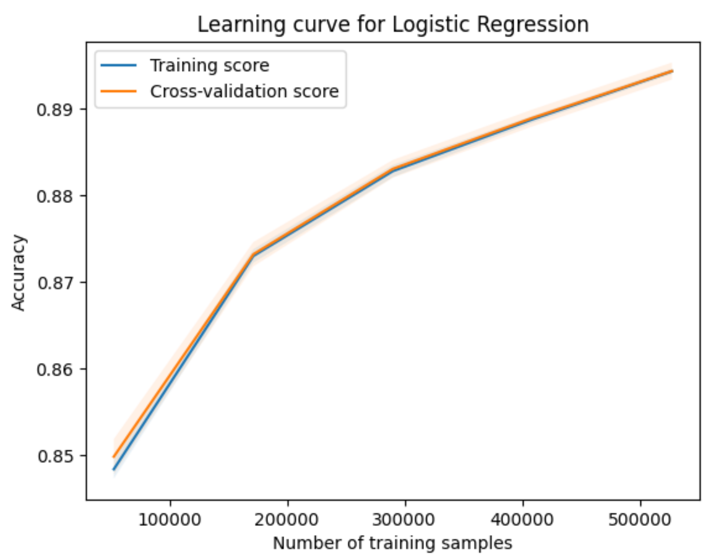

Figure 14. Learning curve for logistic regression.

Light Gradient Boosting Machine (LGBM) is a high-performance gradient boosting framework designed for efficient training and prediction. **Table 3** shows the classification report for LGBM with distinct performance parameters. There is distinct performance parameters retrieved from the logistic regression such as accuracy: 0.987, precision: 0.982, recall: 0.981, and f1 score: 0.982.

Table 3. Classification report for LGBM.

Parameters/ classes	Precision	Recall	F1-Score	Support
Class 0	0.981	0.980	0.980	29455
Class 1	0.982	0.978	0.982	92753
Class 2	0.981	0.981	0.981	128860
Accuracy			0.987	251068
Macro average	0.992	0.980	0.980	251068
Weighted average	0.991	0.981	0.981	251068

Figure 15 shows the ROC curve that indicate the area under curve which is nearly 1 for all classes. It shows the perfect true prediction corresponds to the distinct classes. **Figure 16** shows the precision-recall curve that indicates the average precision 1. **Figure 17** shows the learning curve for training and cross-validation score. **Figure 18** indicates the higher training accuracy and less cross validation score.

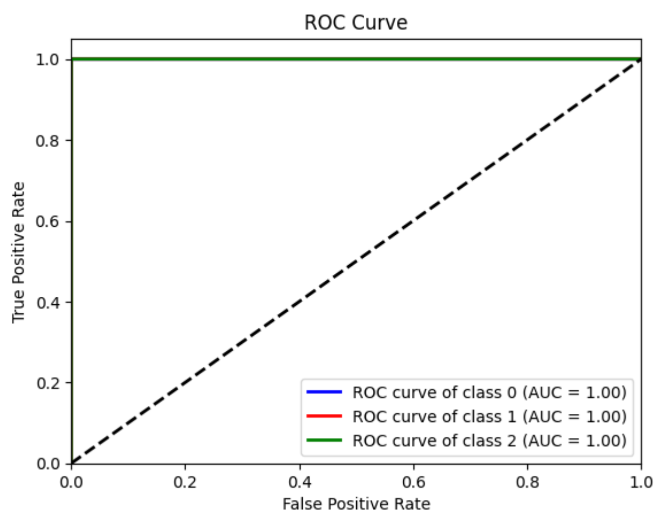

Figure 15. ROC curve LGBM model.

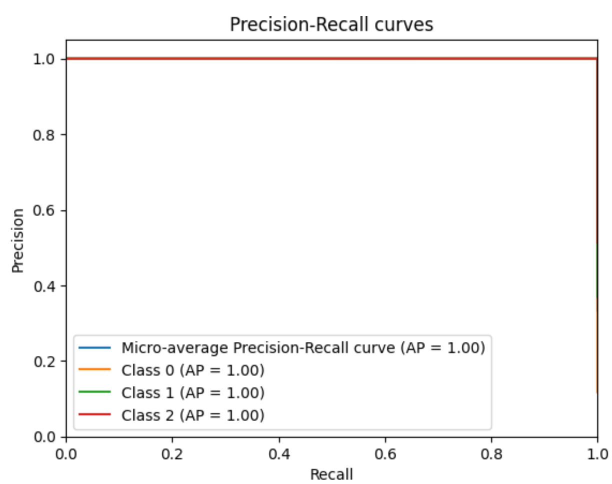

Figure 16. Precision-recall curve for LGBM.

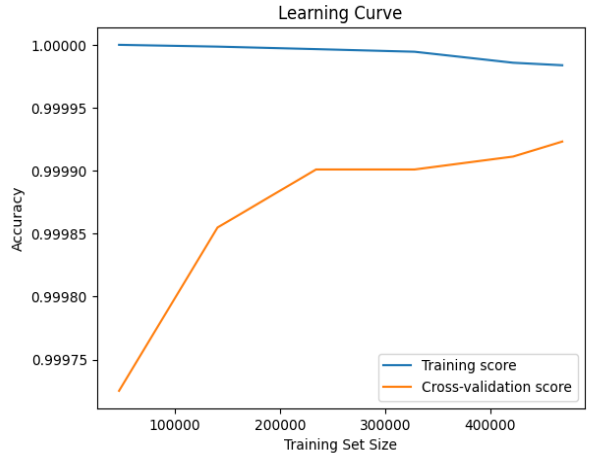

Figure 17. Learning curve for LGBM.

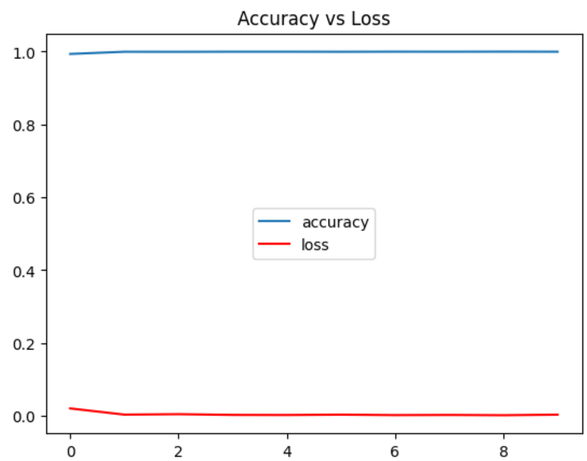

Figure 18. Accuracy vs losses hybrid model.

The proposed hybrid model which is the integration of CNN and LSTM networks generate the results that combines the strengths of both architectures. The CNN extracts the features from data and LSTM model process the sequential model for finding out the temporal patterns. **Table 4** presents the values of distinct performance parameters such as accuracy: 0.985, precision: 0.981, recall: 0.979, and f1 score: 0.983.

Table 4. Classification report for hybrid model.

Parameters/ classes	Precision	Recall	F1-Score	Support
Class 0	0.981	0.978	0.971	29455
Class 1	0.982	0.978	0.971	92753
Class 2	0.981	0.979	0.972	128860
Accuracy			0.985	251068
Macro average	0.982	0.980	0.979	251068
Weighted average	0.981	0.981	0.972	251068

Figure 19 shows the ROC curve that indicate the area under curve which is nearly 1. Since all ROC curves are horizontal at a TPR of 1.0 and a corresponding AUC of 1.00, this indicates that for all classes, the model

has achieved good true positive rates without any false positives. **Figure 20** shows the precision-recall curve that indicates the average precision 1.

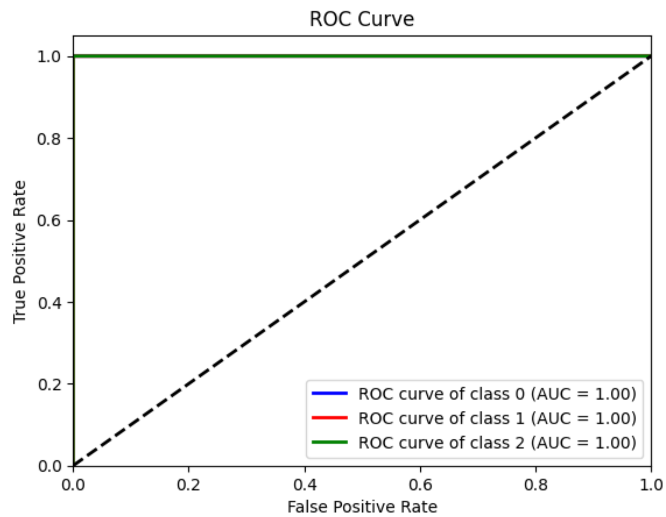

Figure 19. ROC curve for hybrid model.

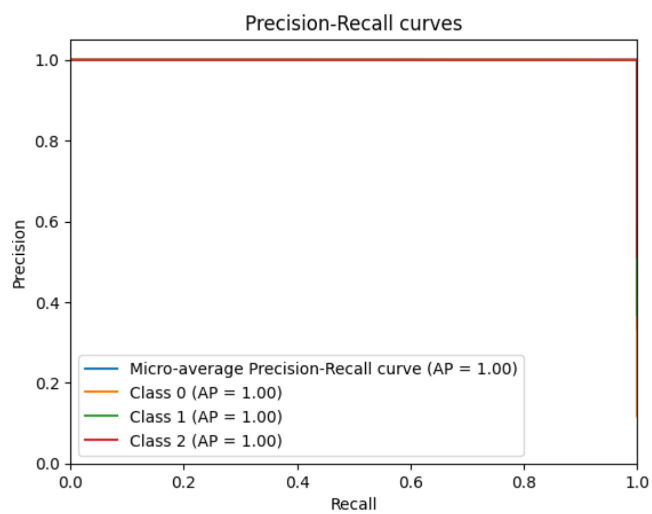

Figure 20. Precision-recall curve for hybrid model.

5. Conclusion

The proliferation of IoT devices has brought unprecedented convenience and connectivity, but it has also introduced new vectors for cyber threats, particularly through botnets that can compromise vast networks of devices. An IoT botnet is a network of IoT devices infected by malicious commands and controlled by a C&C server by several channels including DNS queries. The bot is a program that perform tasks as commanded by the bot master. The design and development of an efficient and intelligent IoT botnet detection model represent a vital step in enhancing the security and reliability of IoT systems. The proposed IoT botnet detection model offers a robust and intelligent solution to the growing challenge of IoT security.

It balances efficiency and accuracy, ensuring the timely detection of botnet activities while minimizing disruption to normal IoT operations. By integrating advanced deep learning techniques such as CNN and LSTM to provide a strong defense against emerging IoT threats. Results have been evaluated in terms of distinct performance parameters such as accuracy, precision, recall, and other parameters. In future, the implementation of the work can be conducted on distinct datasets with ensemble learning classifiers.

Conflict of Interest

The authors declare that there is no conflict of interest.

AI Disclosure

The author(s) declare that no assistance is taken from generative AI to write this article.

Acknowledgments

The authors declare that, there is no financial support from any of the institutions or personal relationship to affect the quality of the paper.

References

Al-Fawa'reh, M., Abu-Khalaf, J., Szewczyk, P., & Kang, J.J. (2024). MalBoT-DRL: malware botnet detection using deep reinforcement learning in IoT networks. *IEEE Internet of Things Journal*, *11*(6), 9610-9629. https://doi.org/10.1109/jiot.2023.3324053.

Alshamkhany, M., Alshamkhany, W., Mansour, M., Khan, M., Dhou, S., & Aloul, F. (2020). Botnet attack detection using machine learning. In *2020 14th International Conference on Innovations in Information Technology* (pp. 203-208). IEEE. Al Ain, United Arab Emirates. https://doi.org/10.1109/iit50501.2020.9299061.

Benlloch-Caballero, P., Wang, Q., & Alcaraz Calero, J.M. (2023). Distributed dual-layer autonomous closed loops for self-protection of 5G/6G IoT networks from distributed denial of service attacks. *Computer Networks, 222*, 109526. https://doi.org/10.1016/j.comnet.2022.109526.

Bhale, P., Biswas, S., & Nandi, S. (2023). Effective injection of adversarial botnet attacks in <scp>IoT</scp> ecosystem using evolutionary computing. *Internet Technology Letters*, *6*(4), e433. https://doi.org/10.1002/itl2.433.

Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: applications, challenges, and opportunities with China perspective. *IEEE Internet of Things Journal*, *1*(4), 349-359. https://doi.org/10.1109/jiot.2014.2337336.

Garg, U., Kumar, S., & Kumar, M. (2023). INFRDET: IoT network flow regulariser-based detection and classification of IoT botnet. *International Journal of Grid and Utility Computing*, *14*(6), 606-616.

Garg, U., Kumar, S., & Mahanti, A. (2024). IMTIBOT: An intelligent mitigation technique for IoT botnets. *Future Internet*, *16*(6), 212. https://doi.org/10.3390/fi16060212.

Gurevin, D., Shan, M., Huang, S., Hasan, M.A., Ding, C., & Khan, O. (2024). PruneGNN: algorithm-architecture pruning framework for graph neural network acceleration. In *2024 IEEE International Symposium on High-Performance Computer Architecture* (pp. 108-123). IEEE. Edinburgh, United Kingdom. https://doi.org/10.1109/hpca57654.2024.00019.

Güven, E.Y., & Gürkaş-Aydin, Z. (2023). Mirai botnet attack detection in low-scale network traffic. *Intelligent Automation & Soft Computing*, *37*(1), 419-437. https://doi.org/10.32604/iasc.2023.038043.

Hasan, T., Malik, J., Bibi, I., Khan, W.U., Al-Wesabi, F.N., Dev, K., & Huang, G. (2022). securing industrial internet of things against botnet attacks using hybrid deep learning approach. *IEEE Transactions on Network Science and Engineering*, *9*(0), 1-11. https://doi.org/10.36227/techrxiv.19313318.v3.

He, Z., & Sayadi, H. (2023). Image-based zero-day malware detection in IoMT devices: a hybrid ai-enabled method. In *2023 24th International Symposium on Quality Electronic Design* (pp. 1-8). IEEE. San Francisco, CA, USA. https://doi.org/10.1109/isqed57927.2023.10129348.

Ibrahim, W.N.H., Anuar, S., Selamat, A., Krejcar, O., Crespo, R.G., Herrera-Viedma, E., & Fujita, H. (2021). Multilayer framework for botnet detection using machine learning algorithms. *IEEE Access*, *9*, 48753-48768. https://doi.org/10.1109/access.2021.3060778.

Kalakoti, R., Nomm, S., & Bahsi, H. (2022). In-depth feature selection for the statistical machine learning-based botnet detection in IoT networks. *IEEE Access, 10*, 94518-94535. https://doi.org/10.1109/access.2022.3204001.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing, 17*(3), 12-22. https://doi.org/10.1109/mprv.2018.03367731.

Memos, V.A., Psannis, K.E., & Lv, Z. (2022). A secure network model against bot attacks in edge-enabled industrial internet of things. *IEEE Transactions on Industrial Informatics, 18*(11), 7998-8006. https://doi.org/10.1109/tii.2022.3162837.

Ngo, Q.D., & Nguyen, Q.H. (2022). A reinforcement learning-based approach for detection zero-day malware attacks on IoT system. In: Silhavy, R. (ed) *Artificial Intelligence Trends in Systems*. Springer International Publishing, Cham, pp. 381-394. https://doi.org/10.1007/978-3-031-09076-9_34.

Rawat, R.S., Pilli, E.S., & Joshi, R.C. (2018). Survey of peer-to-peer botnets and detection frameworks. *International Journal of Network Security, 20*(3), 547-557. https://doi.org/10.6633/ijns.201805.20(3).18.

Salim, M.M., Comivi, A.K., Nurbek, T., Park, H., & Park, J.H. (2022). A blockchain-enabled secure digital twin framework for early botnet detection in IIoT environment. *Sensors, 22*(16), 6133. https://doi.org/10.3390/s22166133.

Schumacher, N., Santos, P.M., Souto, P.F., Martins, N., Sousa, J., Ferreira, J.M., & Almeida, L. (2023). One-class models for intrusion detection at ISP customer networks. In: Maglogiannis, I., Iliadis, L., MacIntyre, J., Dominguez, M. (eds) *Artificial Intelligence Applications and Innovations* (pp. 28-41). Springer Nature, Switzerland. ISBN: 978-3-031-34107-6. https://doi.org/10.1007/978-3-031-34107-6_3.

Venkatasubramanian, M., Lashkari, A.H., & Hakak, S. (2023). IoT malware analysis using federated learning: a comprehensive survey. *IEEE Access, 11*, 5004-5018. https://doi.org/10.1109/access.2023.3235389.

Wang, Z., Huang, H., Du, R., Li, X., & Yuan, G. (2023). IoT intrusion detection model based on CNN-GRU. *Frontiers in Computing and Intelligent Systems, 4*(2), 90-95. https://doi.org/10.54097/fcis.v4i2.10302.

Wei, N., Yin, L., Tan, J., Ruan, C., Yin, C., Sun, Z., & Luo, X. (2024). An autoencoder-based hybrid detection model for intrusion detection with small-sample problem. *IEEE Transactions on Network and Service Management, 21*(2), 2402-2412. https://doi.org/10.1109/tnsm.2023.3334028.

Publisher's Note- Ram Arti Publishers remains neutral regarding jurisdictional claims in published maps and institutional affiliations.