International Journal of Artificial Intelligence Trends (IJAIT)
Vol. 5, Issue III; No. 63, March, 2026, pp. 665-676

DEVELOPMENT OF AN INTELLIGENT BOTNET DETECTION AND MITIGATION FRAMEWORK USING MACHINE LEARNING TECHNIQUES

[1]Nkechi Oji, [2]Ogochukwu Okeke C., [3]Ike J. Mgbeafulike
[1]Department of Computer Engineering, Federal University of Technology, Owerri (FUTO), Imo State, Nigeria.
[1,2,3]Department of Computer Science; Chukwuemeka Odumegwu Ojukwu University, Uli, Anambara State, Nigeria
Author Email: [1]nkechi.oji@futo.edu.ng, [2]ogoookeke@yahoo.com

Article Info
Received: 18/02/ 2026
Revised: 2/4/2026
Accepted 13/4/2026
Corresponding Authors
[1*]Email:
[1]nkechi.oji@futo.edu.ng
Corresponding Author's Tel:
[1*]+2348035057652

ABSTRACT

The rapid evolution of cyber threats, particularly botnet attacks, poses significant risks to network security, challenging traditional defence mechanisms. This study presents the development of an intelligent botnet detection and mitigation framework using machine learning techniques. A dataset comprising multiple malware families and botnet captures was collected, pre-processed, and used to train three classifiers such as Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM). These models were integrated into an ensemble classifier to leverage their complementary strengths for enhanced detection accuracy. Performance evaluation revealed that the SVM achieved the highest individual accuracy of 0.97, while the ensemble model demonstrated superior detection capability, attaining an accuracy and recall of 0.99. System integration using a simulated wireless network showed that the ensemble classifier effectively identified and isolated malicious nodes, outperforming traditional firewall solutions. The study concludes that ensemble-based machine learning approaches provide a robust and reliable solution for botnet detection, offering a scalable framework for enhancing cybersecurity in heterogeneous network environments.

Keywords: Botnet Detection, Cybersecurity, Machine Learning, Ensemble Classifier, Network Security

1. INTRODUCTION

In the digital realm, where threats evolve rapidly, only through unwavering vigilance can we fortify our defenses and protect against the ever-changing landscape of cyber threats". What this means is that cybersecurity demands a steadfast commitment to perpetual watchfulness. Today, cyberspace has witnessed a surge in sophisticated attacks, with botnet standing out as one of the popular attack models for cybercrime (Khan and Mailewa, 2023). Botnet is a network of compromised computers that are controlled by a single entity, known as the botmaster. These compromised devices, often referred to as "bots" and "zombies," are infected with malicious software, allowing the attacker to control them remotely (Anwar and Saravanan, 2022).

The primary purpose of a botnet is to perform various coordinated tasks without the knowledge of the network owners (Mousavi et al., 2020). This attack employed various types of malwaresuch as phishing, worms, Trojan horses, ransomware, and spyware to compromise a network's security and exploit vulnerabilities. Phishing attacks within botnets involve deceptive tactics to trick users into revealing sensitive information, while worms and Trojan horses focus on self-replication and delivering malicious payloads. Ransomware, on the other hand, encrypts files, demanding payment for their release, and spyware discreetly monitors and collects user data.

The effects of a botnet are multifaceted and can severely impact a system's integrity, availability, and confidentiality. Signs of a system under a botnet threat include unusual network traffic patterns, increased bandwidth usage, unexplained system slowdowns, unauthorized access to sensitive information, and a rise in the number of compromised devices (Joshi et al., 2022). Additionally, frequent system crashes, unexpected pop-ups, and altered browser settings are indicative of potential botnet activities (Lo et al., 2023). To mitigate the risks associated with botnets, it is essential to implement robust cybersecurity measures, conduct regular security audits, and educate users about recognizing and avoiding phishing attempts and suspicious activities. Early detection and proactive response are crucial in minimizing the potential damage caused by botnet threats.

According to Nasir et al. (2023), traditional security mechanisms such as encryption, access control and botnet intrusion detection systems often struggle to keep pace with the dynamic and evolving nature of these threats. Attackers leverage a variety of tactics, such as social engineering, zero-day exploits, and polymorphic malware, making it challenging to predict and prevent security breaches effectively. Moreover, Ayo et al. (2023) revealed that the sheer volume of devices connected to networks, including IoT devices with varying security postures, introduces new attack vectors. Weaknesses in IoT device security, inadequate authentication mechanisms, and insufficient encryption protocols create opportunities for attackers to infiltrate and compromise systems (Abrantes et al., 2021). The scale and heterogeneity of these environments make it difficult to implement uniform security measures across all devices and platforms. In addition, securing data in transit and at rest, ensuring the integrity of cloud-based applications, and protecting against unauthorized access become paramount concerns (Nazir et al., 2023).

To address these challenges, organizations need a multi-faceted approach to cybersecurity. This includes the integration of advanced threat detection technologies, machine learning algorithms for anomaly detection, and behavior analytics to identify patterns indicative of potential security incidents (Ayo et al., 2023). Continuous monitoring and real-time response capabilities are essential to mitigate the impact of attacks and vulnerabilities swiftly. To study propose the development of an intelligent botnet detection and mitigation framework for enhanced cyber security network using machine learning technique.

2. METHODOLOGY

The methodology used for this work is a combination of agile and method. In realizing this methodology first a data model which captures the dynamic characteristics of Botnet will be developed using data augmentation approach, then the collected data will be used to train selected machine learning algorithms such as artificial neural network, decision tree and support vector machine respectively to generate an ensemble model for the classification of botnet. The model will be evaluated with various metrics which define success of cyber security model and analyze the results for recommendations to facilitate the design of future cyber security frameworks.

2.1 Data Collection

The dataset and structure are critical for efficiently storing, managing, and querying large datasets generated during the botnet detection and mitigation process. In this system, the dataset was structured to handle data related to various malware families, botnet captures, and attack signatures. A clear and organized design ensures that network traffic, system logs, threat classification data, and mitigation actions are stored in a way that supports real-time analysis and decision-making. Table 1 illustrates the structure of the dataset, focusing on the malware classes and the associated botnet captures. This organization will help in categorizing and identifying various botnet activities linked to specific malware families, which is crucial for accurate botnet detection.

Table 1: Botnet Class

Malware Family	CTU Botnet Captures
Dynamer	189-1, 217-1, 229-1
Taobao	232-1, 237-1
OpenCandy	194-1, 195-1, 208-1, 213-1
Cridex	108-1, 109-1
Dridex	113-1, 153-1, 218-1, 227-1, 228-1, 246-1, 248-1, 259-1
Yakes	104-1, 107-1, 108-1, 203-1, 310-1

This Table 1 categorizes the different botnet classes identified in the botnet dataset, with each family associated with a specific set of botnet captures. These captures represent network traffic or event logs tied to botnet activities, providing valuable data for detecting and analysing botnet behaviour. The dataset structure allows for efficient querying of these records, enabling the detection system to quickly identify and classify botnet activities in real time.

2.2 The Proposed System Models

The section presented the various flowcharts of the proposed algorithms starting with the ensemble model classifier, the low chart of the botnet classification model and the flowchart of the botnet detection and management system. The Figure 1 presents the system flowchart of the ensemble classifier.

The flowchart outlines an ensemble model workflow for classification tasks. It begins with input features obtained from a dataset, followed by a feature selection and extraction step to prepare the data. The processed features are passed to three different machine learning modelsANN, DT, and SVM. The predictions from these models are combined in a voting process within the ensemble model. A decision diamond checks if the voting process is complete, and once it is, the ensemble produces the classification output. The flow ends at the stop point. This approach leverages the strengths of multiple models to improve accuracy and robustness in classification tasks. Figure 2 present the flowchart of the botnet classifer.

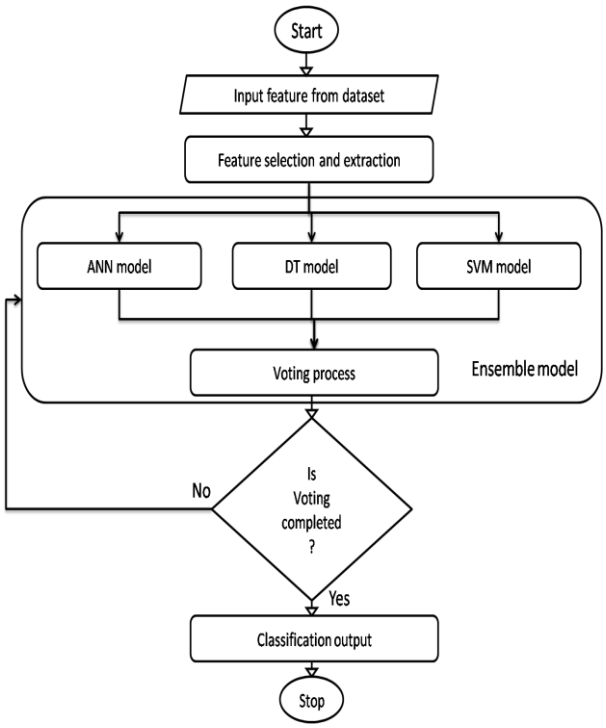

Figure 1: Flow chart of ensemble classifier

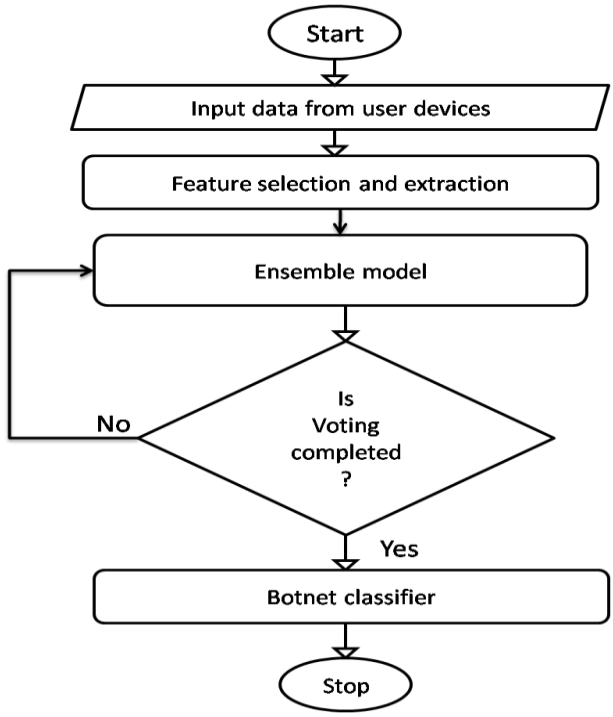

Figure 2: Flow chart of the botnet classifer

The Botnet Classifier flowchart represents a systematic approach for classifying botnet activity. It starts with input data collected from user devices, followed by a feature selection and extraction process to identify relevant data features for analysis. The refined features are passed into an ensemble model, which aggregates predictions from multiple models to enhance accuracy. A decision diamond checks if the voting process within the ensemble is completed. If voting is successful, the botnet classifier generates the final classification output, indicating whether botnet activity has been detected. The process concludes at the Stop point, ensuring an organized detection pipeline. Figure 3 presents the flow chart of the botnet detection and management system.

The Botnet Detection and Management System flowchart outlines a systematic process for identifying and mitigating botnet activity in a network environment. The process begins with input data collection from user devices, which involves gathering network traffic data or user activity logs. These datasets could belong to either normal users or systems infected with botnets. The collected data serves as the foundation for further analysis. Once the data is gathered, the system proceeds to feature selection and extraction. In this step, the system identifies and extracts relevant features that represent network behaviour or user activity patterns. This process is crucial for eliminating irrelevant information and optimizing the performance of the botnet detection model. Features such as abnormal traffic rates, unusual connection patterns, or suspicious resource usage are commonly extracted. The extracted features are then passed through the botnet classifier, which leverages machine learning techniques to analyze the data and detect botnet activities. The classifier could utilize ensemble methods, combining multiple models to improve detection accuracy. At this stage, the

process checks whether the voting process of the ensemble model is completed. If voting is incomplete, the system loops back to ensure that sufficient predictions are obtained for a reliable classification. Once the voting process is finalized, the system verifies whether a botnet is classified based on the analysis results. If the system does not detect botnet activity, the process concludes. However, if a botnet is successfully classified, the system triggers a decision-based algorithm to take appropriate actions. These actions may include blocking malicious traffic, isolating compromised devices, notifying administrators, or applying mitigation strategies to contain the threat.The process ultimately ends with the stop stage, signifying the completion of botnet detection and management. By combining feature extraction, ensemble classification, and decision-based actions, this flowchart provides a robust framework for early detection and response to botnet threats, enhancing overall cybersecurity in the connected environment.

2.3 Botnet Detection and Management

Botnet detection and management systems use the ensemble machine learning classifiers to identify infected devices or malicious activities in real-time. Once a botnet is detected, the system initiates isolation actions, such as blocking the infected device's IP address or disconnecting it from the network to prevent further damage. Simultaneously, an alert system is triggered, notifying network administrators of the detected botnet threat. The alert contains information such as the affected device's identity, the type of botnet activity, and any actions taken. This approach allows for a swift response to mitigate the threat and prevents the botnet from spreading further across the network, ensuring ongoing protection and network stability. The Figure 4 presented the new ensemble model block diagram proposed.

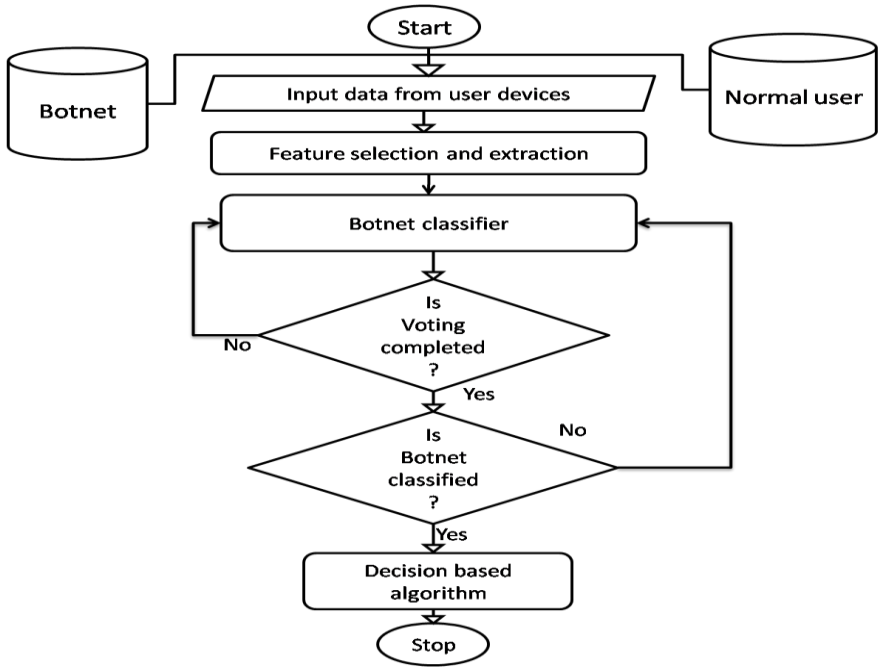

Figure 3: Flow chart of the botnet detection and management system

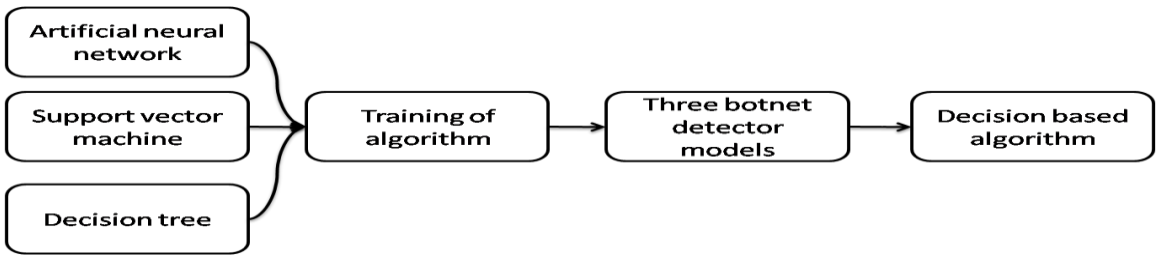

Figure 4: Proposed New ensemble botnet classifier bock diagram

The Figure 4 presented the proposed ensemble model for the development of the botnet classifier. The classifier is made of three algorithms which are neural network, support vector machine and decision tree which are trained respectively to generate a model for the detection of botnet. These three modes form the base of the decision-based algorithm which computed the output of the three model to determine botnet in a network infrastructure. The complete block diagram of the proposed system was presented in the Figure 5

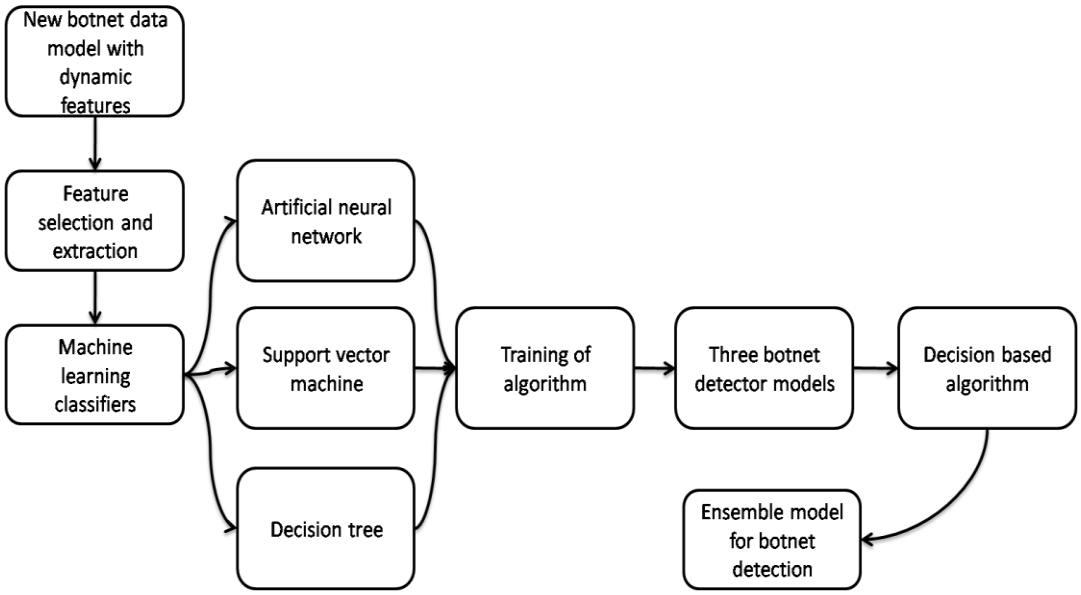

Figure 5: The complete system block diagram for the proposed botnet detection

The Figure 5 presented the block diagram of the botnet detection model using the three-machine learning algorithm, decision-based algorithm and data process approach. First the data model of botnet will be collected and extracted considering key selected botnet features to train machine learning classifiers as an ensemble mode integrating decision-based algorithm. This produced the ensemble model for the botnet detection process.

2.4 System Implementation

The implementation of the botnet classifier system in PYTHON involves designing a robust framework for detecting and classifying botnet activities using machine learning techniques. The system begins by importing input data collected from user devices, such as network traffic logs or behavioral metrics. The feature selection and extraction process uses PYTHON's built-in tools like `pca()` for dimensionality reduction and custom scripts for extracting statistical features, including packet rate, connection duration, or unusual resource usage. Pre-processing steps, such as normalization, are performed to standardize the data, ensuring efficient model training. PYTHON's `fitensemble()` function or the Classification Learner toolbox is utilized to develop an ensemble model comprising machine learning classifiers like DT, SVM, and ANN. A voting mechanism is integrated to combine the outputs of individual classifiers, improving prediction reliability.

In the botnet classification phase, the trained ensemble model processes new input data in real time. The system evaluates the features and categorizes them as either "botnet" or "normal user" based on learned patterns. If botnet activity is detected, a decision-based algorithm is triggered, using conditional structures like `if` and `switch` statements to isolate the compromised nodes or send alerts to system administrators for mitigation. PYTHON's visualization tools, such as `plotconfusion()` for confusion matrix plots and `roc()` for Receiver Operating Characteristic curve analysis, are employed to validate model performance. By leveraging PYTHON’s robust machine learning and analysis toolboxes, the botnet classifier achieves high accuracy, reduced false positives, and reliable detection of malicious network activities.

3. PERFORMANCE EVALUATION

This section assesses the performance of the machine learning algorithms used to develop the ensemble model. Accuracy and loss were the primary metrics applied to evaluate the models throughout the training process. Accuracy measured the

proportion of correctly classified instances, providing insight into each model's overall effectiveness. Meanwhile, loss quantified the error or deviation from the expected results, reflecting the model's ability to minimize incorrect predictions. The analysis of these metrics allowed for a detailed comparison of individual models, highlighting their strengths and weaknesses. This evaluation was critical in identifying the most reliable and efficient models to combine into the ensemble, ensuring optimal botnet detection performance. Figure 6 presents the accuracy and loss result of the DT training. The Figure 7 reported the result of the ANN training performance considering accuracy and loss values. Figure 8 presents the SVM result.

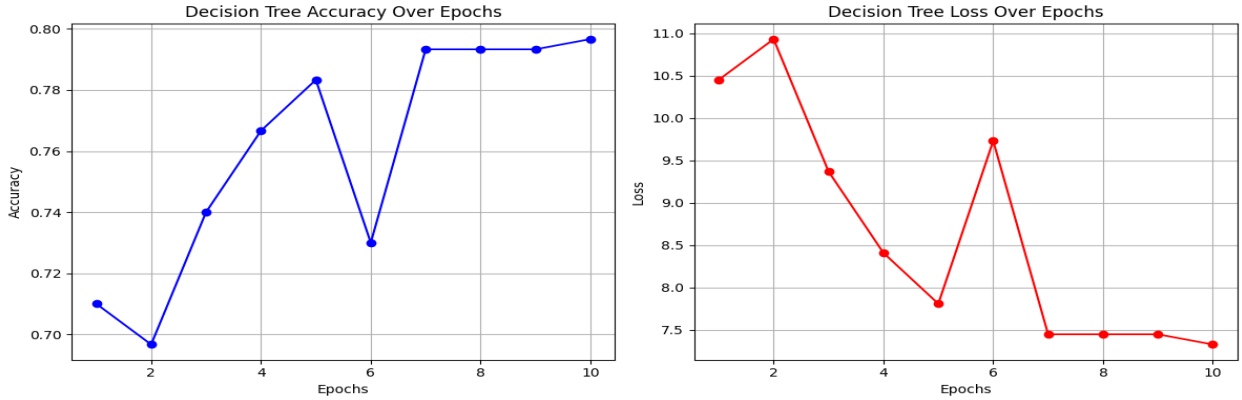

Figure 6: Accuracy and loss result of the DT

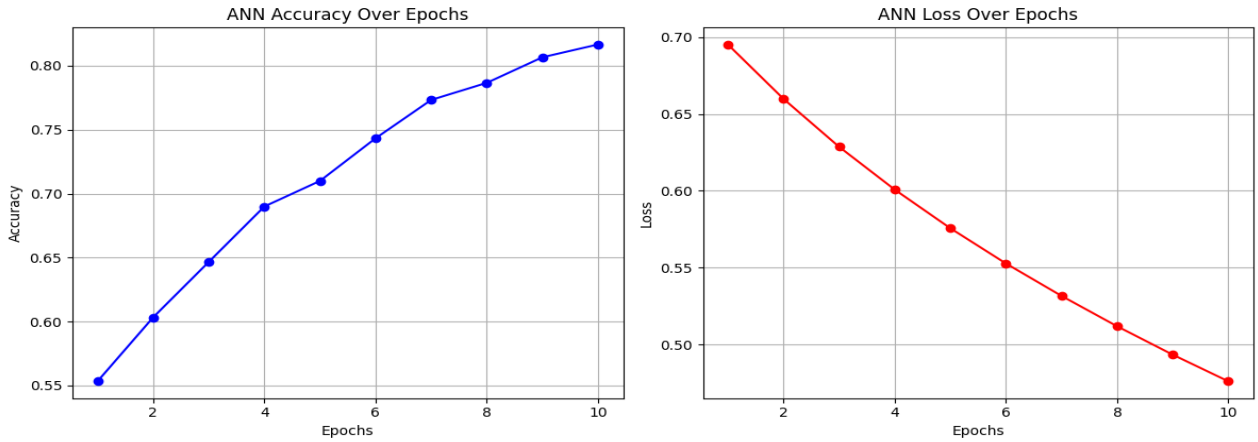

Figure 7: Accuracy and loss result of the ANN based botnet classifier

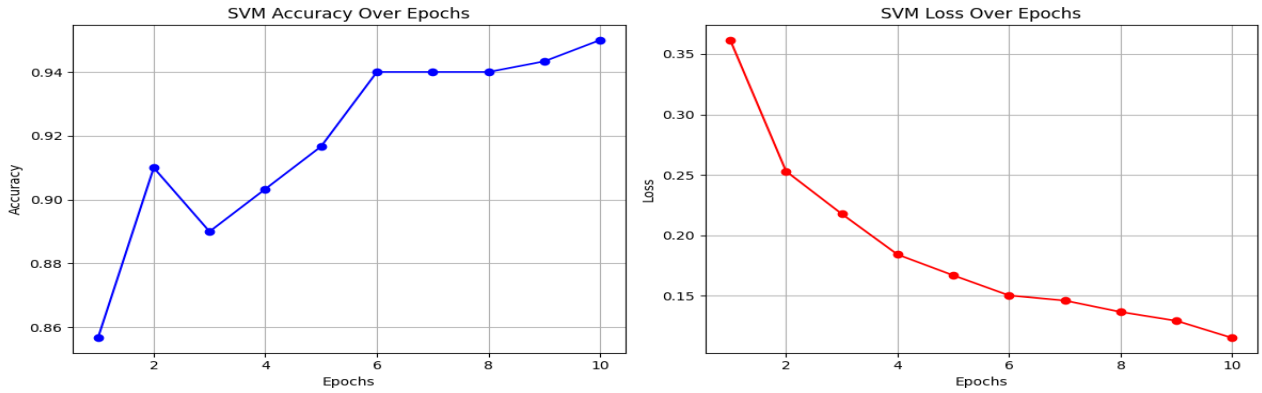

Figure 8: Result of the SVM botnet classifier

Figure 6 illustrates the accuracy and loss metrics obtained during the training of the DT-based botnet classifier. The normalized accuracy of 0.79 indicates that the model correctly identified botnet features 79% of the time, reflecting its moderate efficacy in distinguishing between malicious and benign traffic. Meanwhile, the reported loss value of 7.2 suggests room for improvement in minimizing prediction errors, as it quantifies the extent to which the model's output deviates from the expected results. These metrics collectively indicate that while the DT classifier successfully captures significant patterns for botnet detection, its performance could benefit from further optimization, such as fine-tuning hyperparameters, incorporating additional features, or exploring ensemble techniques to enhance predictive accuracy and reduce the loss.

The results indicate that the ANN botnet classifier achieved a higher accuracy of 0.83, surpassing the performance of the Decision Tree classifier, and demonstrating its superior ability to identify botnet features. The reported loss value of 0.44 is significantly lower, highlighting the ANN's improved efficiency in minimizing prediction errors compared to the DT model. These outcomes suggest that the ANN classifier not only learns the botnet patterns more effectively but also generalizes better, making it a more robust choice for botnet detection tasks. Further analysis of the ANN's architecture and training parameters could shed light on its enhanced performance, offering insights into potential refinements for other models.

Figure 8 highlights the performance of the SVM-based botnet classifier, showcasing exceptional results with an accuracy of 0.97 and a remarkably low loss value of 0.12. These metrics indicate that the SVM classifier demonstrates superior capability in correctly identifying botnet features, with minimal deviation from expected outcomes. The high accuracy reflects the SVM's effectiveness in handling complex decision boundaries, while the low loss value signifies a well-optimized model with minimal error. These results position the SVM-based classifier as a highly reliable solution for botnet detection, outperforming both the ANN and DT classifiers in terms of accuracy and loss, and affirm its robustness in learning intricate patterns within the dataset.

3.1 Result of System Integration

This secton presents the result of system integration. This was achieved through complication of the tonet detection model developed on the routing device of the wireless network aimed to protect against botnet attack. In the system integration process, gns-3 software, Kali-liux and ubuntu server were all utilized to simulate the system integraito of the network with the dynamic botnet classier with ensemble model and also the netwok with the covetioal static firewall. Figure 9 presents the result of the existing network with the traditional firewall (Wiress Access Protocol-Version3), which is desiged to protect the network against man in the middl attack and ensure packetsecurity.

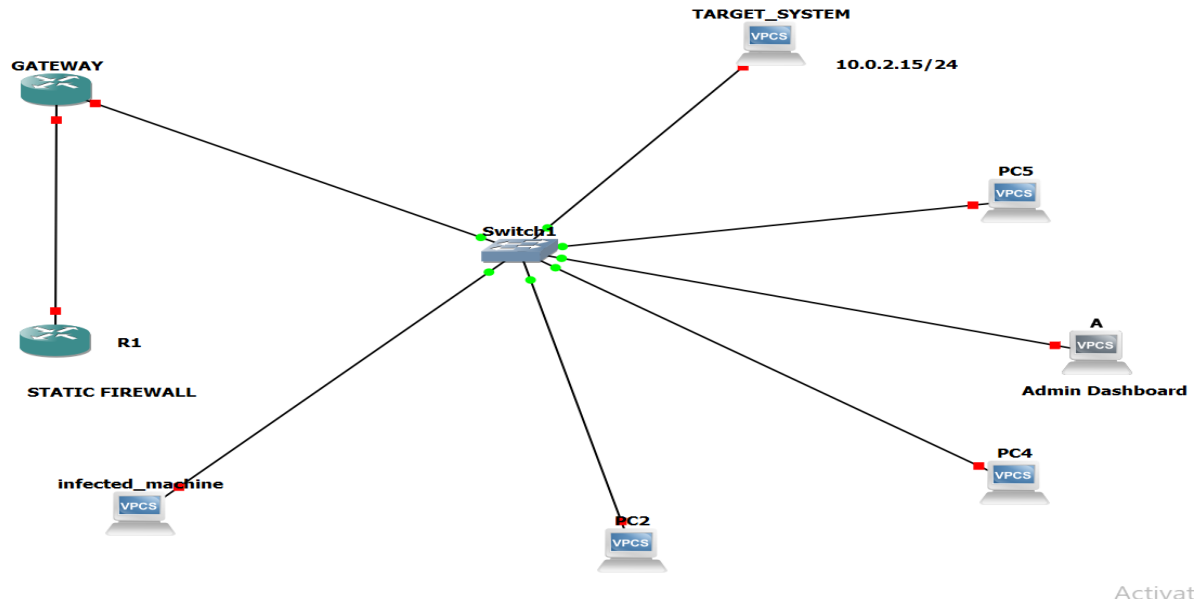

Figure 9: Simulation model of the existing system security with WAP-3 firewall

Figure 9 presents the simulation result of the existing network architecture without the Botnet classification model. The network is made of four PC connected as WLAN and an infected machine used to send botnet to the network. The passed through the gateway to the firewall which is aimed as protected the network from the bonet features. The simulation results were reported in Figure 10. From the results, it was observed that all the IP address of the user equipment were allowed throughput to the server, which is not very good. This implied that the existing firewall while able to secure the packets from accessby intruder due to its encryption features, were vulnerable in detecting botnet features, and hence were all allowed access to the network. The highlighted IP address in the results revealed the attacker IP address as it was also allowed access to the server, which will go on to slow down the server and then eventually shutdown to interrupt services and user experience.

No.	Time	Source	Destination	Protocol	Length	Inf
1	0.000000000	10.0.2.15	185.125.190.97	TCP	74	54
2	0.451078283	185.125.190.97	10.0.2.15	TCP	60	80
3	0.451123602	10.0.2.15	185.125.190.97	TCP	54	54
4	0.453244621	10.0.2.15	185.125.190.97	HTTP	142	GE
5	0.453881866	185.125.190.97	10.0.2.15	TCP	60	80
6	0.552855440	185.125.190.97	10.0.2.15	HTTP	223	HT
7	0.552855681	185.125.190.97	10.0.2.15	TCP	60	80
8	0.552899610	10.0.2.15	185.125.190.97	TCP	54	54
9	0.554386820	10.0.2.15	185.125.190.97	TCP	54	54
10	0.554966602	185.125.190.97	10.0.2.15	TCP	60	80
11	0.934372743	10.0.2.15	185.125.190.97	TCP	74	54
12	1.169777450	185.125.190.97	10.0.2.15	TCP	60	80
13	1.169825240	10.0.2.15	185.125.190.97	TCP	54	54
14	1.170964557	10.0.2.15	185.125.190.97	HTTP	142	GE
15	1.171374369	185.125.190.97	10.0.2.15	TCP	60	80
16	1.273287898	185.125.190.97	10.0.2.15	HTTP	223	HT
17	1.273288292	185.125.190.97	10.0.2.15	TCP	60	80

Figure 10: Result of the existing network without Botnet classifier

The Figure 10 has demonstrated the vulnerabilities of the existing firewall, as it allows eve the attacker packet access to the network, which is not good and affect network integrity. The Figure 11 presented the simulation result of the WLAN with the botnet classifier.

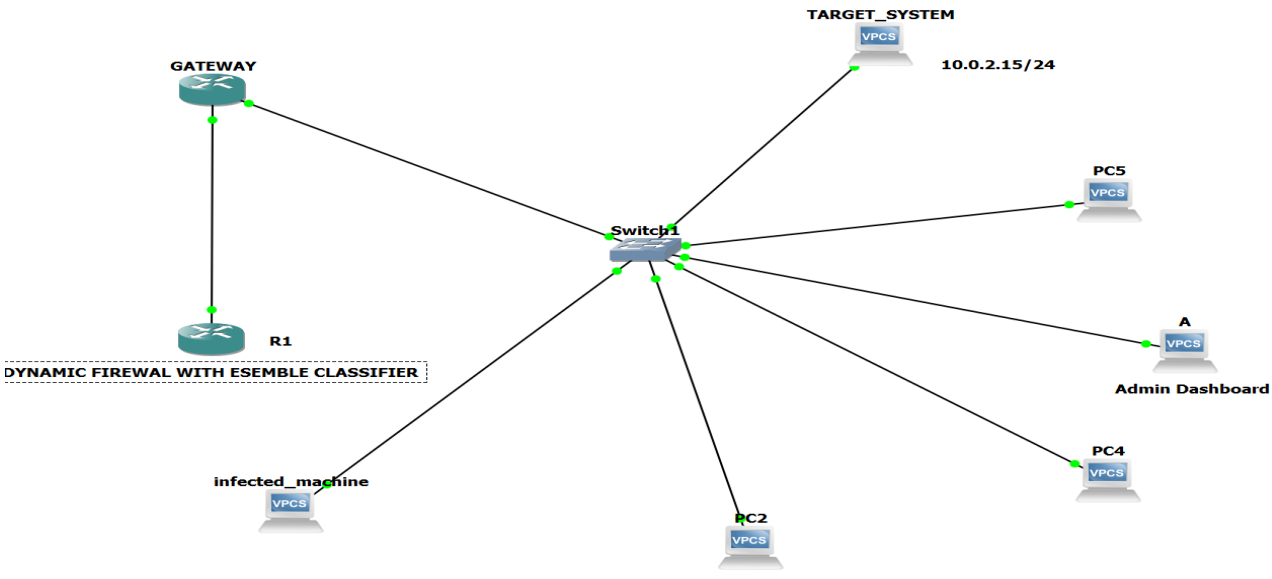

Figure 11: Simulation of WLAN with botnet classifier

Figure 11 presents the simulation of the integrated botnet classifier on the WLAN for network security against botnet. In the simulation, the infected machine was used to transmit botnet to the network, while the performance of the botnet classifier was accessed. The botnet was transmitted by pinging multiple packets the same time to the network, while monitoring the dynamic botnet classification firewall. The results were reported in Figure 12.

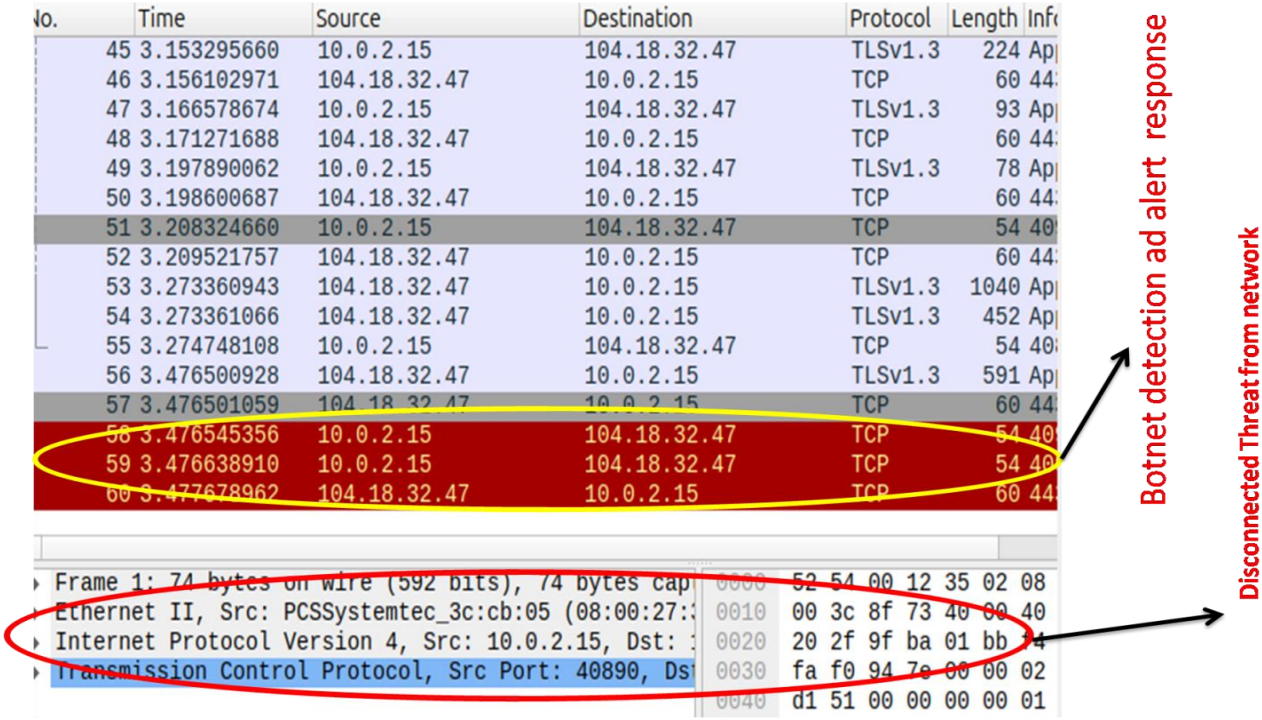

Figure 12: Result of WLAN with botnet classifer

The Figure 12 presents the result of the ensemble botnet classifier on the network during botnet attack. From the results, it was observed that the classifier was able to identify malicious IP address used to send the botnet attack as highlighted in red colour, and then isolate from the server by dropping the packets as shown in the down part of the results. This result has demonstrated the effectiveness of the botnet classifier in successfully detecting and isolating threat from botnet infected nodes, thereby protecting the WLAN against cyber threat, more specifically botnet attack.

4. CONCLUSION

This study focuses on developing a robust botnet detection system using machine learning techniques. By integrating multiple machine learning algorithms such as SVM, DT, and ANN, the study demonstrates the potential of these methods for accurately identifying botnet activities in network traffic. The ensemble model, combining these individual algorithms, further improves the detection performance by leveraging the strengths of each model, resulting in higher accuracy, recall, and precision rates. The comparative evaluation of the models highlights the ensemble approach as the most effective solution, with accuracy reaching 0.99 and recall at 0.99, offering significant improvements over individual models like DT. The findings confirm that ensemble-based techniques can enhance the robustness and reliability of botnet detection systems. The results also reflect the strengths and weaknesses of each model. While SVM and ANN showed high performance in terms of accuracy and recall, the Decision Tree model faced challenges due to its simplicity, with significantly lower performance in comparison. Despite these challenges, the Decision Tree model was still included in the ensemble model, showcasing the advantage of hybrid systems that combine multiple models to reduce individual model biases. The ensemble approach, while introducing slightly higher loss, outperformed individual models in terms of detection capability, demonstrating the importance of combining different machine learning techniques to achieve superior results. However, the study also revealed several limitations of the system. The performance of the models is highly dependent on the quality and diversity of the training data. In conclusion, this research contributes to the field of

network security by demonstrating the potential of machine learning algorithms for botnet detection. The study shows that ensemble models offer substantial improvements in detection performance compared to individual models. Despite some limitations, such as dependency on high-quality data and real-time processing challenges, the system represents a step forward in enhancing network security and provides a foundation for future research and development in botnet detection and cybersecurity applications.

REFERENCES

Abrantes, R., Mestre, P., & Cunha, A. (2021). Exploring dataset manipulation via machine learning for botnet traffic. *Procedia Computer Science, 196,* 133–141. https://doi.org/10.1016/j.procs.2021.11.082

Anwar, F., & Saravanan, S. (2022). Comparison of artificial intelligence algorithms for IoT botnet detection on Apache Spark platform. *Procedia Computer Science, 215,* 499–508. https://doi.org/10.1016/j.procs.2022.12.052

Ayo, F. E., Awotunde, J. B., Folorunso, S. O., Adigun, M. O., & Ajagbe, S. A. (2023). A genomic rule-based KNN model for fast flux botnet detection. *Egyptian Informatics Journal, 24,* 313–325. https://doi.org/10.1016/j.eij.2023.05.002

Joshi, C., Ranjan, R. K., & Bharti, V. (2022). A fuzzy logic based feature engineering approach for botnet detection using ANN. *Journal of King Saud University – Computer and Information Sciences, 34,* 6872–6882. https://doi.org/10.1016/j.jksuci.2021.06.018

Khan, S., &Mailewa, A. B. (2023). Discover botnets in IoT sensor networks: A lightweight deep learning framework with hybrid self-organizing maps. *Microprocessors and Microsystems, 97,* 104753. https://doi.org/10.1016/j.micpro.2022.104753

Lo, W. W., Kulatilleke, G., Sarhan, M., Layeghy, S., & Portmann, M. (2023). XG-BoT: An explainable deep graph neural network for botnet detection and forensics. *Internet of Things, 22,* 100747. https://doi.org/10.1016/j.iot.2023.100747

Mousavi, S. H., Khansari, M., & Rahmani, R. (2020). A fully scalable big data framework for botnet detection based on network traffic analysis. *Information Sciences, 512,* 629–640. https://doi.org/10.1016/j.ins.2019.10.018

Nasir, M. H., Arshad, J., & Khan, M. M. (2023). Collaborative device-level botnet detection for internet of things. *Computers and Security, 129,* 103172. https://doi.org/10.1016/j.cose.2023.103172

Nazir, A., He, J., Zhu, N., Wajahat, A., Ma, X., Ullah, F., Qureshi, S., & Pathan, M. S. (2023). Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets. *Journal of King Saud University – Computer and Information Sciences, 35.*https://doi.org/10.1016/j.jksuci.2023.101820