© AUG 2025 | IRE Journals | Volume 9 Issue 2 | ISSN: 2456-8880

Improved Algorithm Technique for Detecting and Eliminating Botnets in Network System

ABU TASIU[1], DR. ABDULLAHI MUSA YOLA[2]
[1]Department of Computer Science, Federal Polytechnic Kaltungo
[2]Department of Computer Science, Federal University, Kashere

Abstract- ***A botnet is a network of infected computers that are used by hackers to launch distributed denial of service (DDoS) attacks, phishing attacks, spambot attacks, chatbot attacks, etc., for the purpose of gaining access to steal confidential information or hook a system for ransom. Concentration of botnet attacks before was on traditional devices like personal computers (PCs), laptops, phones, and so on. The coming of the Internet of Things (IOT) changes the direction of attack and adds strong volume to the attack. When hackers discovered the flaws in IoT, especially in the area of configurations, they took the advantages and regularized the strong volume of attack. Many researchers have studied and contributed to botnet detection methods and techniques, using machine learning models, neural network models, deep learning models, blockchain, or intrusion detection system models. In this study we present an improved technique of detecting, classifying, and eliminating botnets in a network system. We hybridized and modified feature selection algorithms with machine learning to detect botnets in a network system. The accuracy achieved was 99%, precision was 99%, recall was 99.5%, and the F1 score was 99.5%. This means that the classification report shows nearly perfect performance on both normal and botnet traffic.***

Index Terms : Machine Learning (ML), Deep Learning (DL), Internet of things (IoT), Command and Control (C&C), Pair to Pair (P2P), Industrial Internet of Things (IIoT)

I. INTRODUCTION

The terms “bot” and “net” allude to robots and the Internet, respectively. The attacker controls a command-and-control server that is used to operate the bots remotely. The control and command server (one of the bots in the botnet) is used by the attacker to direct and teach the other bots individually and collectively at the same time. By infecting the network server, the botnet’s size can grow. The ability of the botnet to spread allows it to do so across the Internet. The following attacks can be carried out using botnets: phishing, click fraud, password theft, spamming, bit-coin fraud, mass identity theft, traffic sniffing, fresh malware distribution, and key logging.(Obeidat & Yaqbeh, 2022a).

As technology advances, botnet attacks are getting harder to find, especially when they target advanced systems like cloud services and smart devices (IoT). Botnets constitute a major threat to the security and stability of internet and network infrastructures. A botnet is a network of malware-infected hosts, which are typically controlled by a Command and Control (C&C) server. The C&C server architecture allows for distributed malicious attacks on either the infected hosts or other interconnected hosts over LAN or the internet. C&C servers are commonly known as the bot masters, while infected hosts are simply referred to as bots. Botnets are commonly divided into two general architectural structures, centralized and Peer-to-Peer (P2P). These structures are defined by how commands are transmitted throughout the C&C channel (Thanh et al., 2021).

Detecting botnets in network traffic has become more problematic due to the use of command-and-control (C2) servers and peer-to-peer (P2P) systems. Older detection methods like signature-based, anomaly-based, DNS-based, and mining-based techniques used to work well, but now it is easier for attackers to bypass as technology improves. A big reason for this change is the growth of Internet of Things (IoT) devices, which attackers use to hide their actions and carry out progressive attacks. Stressed the importance of quickly detecting IoT botnet attacks to stop

infected devices from spreading and causing more harm to networks (Taylor & Ezekiel, 2022).

This research employed the current use of hybridize ML methods for botnet detection. It focuses on solving the challenges of unbalanced datasets and optimization problem. The review aims to explore various ML techniques used in botnet within network systems. A systematic approach was adopted to collect, filter, and analyze relevant literature. And, we aim to compare with other models utilized to point out what is working, what is not, and how future systems can be improved to better detect and stop botnet attacks.

The remainder of the work is structured as follows: The "Conceptual Review" section outlines the development of a new method based on existing techniques. The "Theoretical Review" analyzes previous research in the field. The "Methodology" section discusses the approach used for detecting botnets in network systems through machine learning and related tools. The "Empirical Review" examines selected datasets, most of which are suitable for network-based detection. The final section presents a Comparative Analysis of Related Works and Discussion.

II. LITERATURE REVIEW

The terms “bot” and “net” allude to robots and the Internet, respectively. The attacker controls a command-and-control server that is used to operate the bots remotely. The control and command server (one of the bots in the botnet) is used by the attacker to direct and teach the other bots individually and collectively at the same time. By infecting the network server, the botnet’s size can grow. The ability of the botnet to spread allows it to do so across the Internet. The following attacks can be carried out using botnets: phishing, click fraud, password theft, spamming, bit-coin fraud, mass identity theft, traffic sniffing, fresh malware distribution, and key logging(Obeidat & Yaqbeh, 2022). Botnets are a dangerous hivemind of fake accounts spun together to create a web, hoping to catch the unsuspecting victim within from a series of attacks which will be categorized within second III. Botnets are comparable to worms, travelling at hi speeds through different computers, the difference being that bots can cooperate and target effectively, keeping in mind they are there for a malicious purpose (Deeks, 2023)
Intruders use different tricky techniques to distribute malicious software that is capable of converting a computer into bot or zombie. When such a situation arises in which a computer is being controlled not by user but by a hacker, it performs several suspicious tasks on internet without the knowledge of the user. In other words, the collections of several computers that are associated to perform suspicious tasks using malicious software are termed as BOTNETs. Attackers usually utilize the bots to infect huge number of computers. These computers form a group known as BOTNET. These zombies can be utilized to spread out spam emails, distribute viruses, attack the servers, and commit various kinds of fraud and cybercrimes. The size of BOTNET is variable that is it can be small or large. The size of BOTNET depends upon the sophistication and complexity of the bots that are used. A large BOTNET consists of tens and hundred thousand zombies. While on the other hand, a smaller BOTNET comprised of a few thousand of zombies (Iftikhar et al., 2020).

Application of Botnets

Botnets severely damage security by taking control of infected computers, which often store sensitive information. If the security of this machine is compromised, the attacker can easily harvest that sensitive and confidential data. Bot herders used to sell or rent their botnets to those who want to perform hacker activities. The strong penetrating capability and strength of botnets, give attacker more and more power on the internet. With the increase in number of botnets, the control over compromised systems of the herder becomes stronger thus performing more complicated, advance and typical activities that internet has never seen before. Some of the severe applications of botnets are discussed below (Iftikhar et al., 2020).

Click Fraud

Botnets can be utilized to engage in click fraud. In this type of scam, the bot software used to navigate different websites on browser and automatically click on advertisements. Now consider about a herder having a bot network of several thousand computers and stealing a large amount of money from online

advertisement organizations that pay small amount on each click. With a large network, each click for few times, returns heavy amount of money. As the clicks are coming from each separate entity distributed across the globe, so investigators cannot find out that this is a scam (Iftikhar et al., 2020).

Distributed Denial of Service (DDoS)

Figure 1 demonstrated that botnets are used to remunerate confrontation on various computers over the network accessing the internet by completely trapping and saturating its bandwidth and various other resources. Such ddos attacks can disable the access the web pages for a long span of time. While considering the financial organizing, this delay of accessibility places a marvelous and enormous burden on financial operators that are unable to service their customers (Syed, 2020).

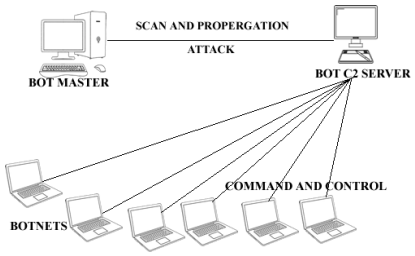

Figure 1 : Distributed Denial of Service (Iftikhar et al., 2020)

Another type of attack comes under the umbrella of DDoS where the attackers demand payment to free the attacked resources and allow the traffic to flow again. This type of attack is known as extortion attacks (Iftikhar et al., 2020).

Keylogging and Mass Identity Theft

Keylogging is the technique that is used to record the sequence of the key pressed. This can be done by installing the key logger software. This is encryption software that is used to gather the sequence of keys pressed by the user. This includes the personal information of the user and passwords. The noticeable expansion in the theft of PayPal accounts in recent years can be attributed to various underlying factors. One primary aspect is the perception of PayPal as a secure and emotionally appealing payment method, which may inadvertently attract malicious actors seeking to exploit its popularity and trustworthiness among users (Casado-Aranda et al., 2018). Furthermore, the rise in online deviance, including the prevalence of malicious software and harassment, has been identified as a critical risk factor contributing to online identity theft, including the theft of PayPal accounts (Guedes et al., 2022).

Traffic Monitoring and Spamming

Botnets are utilized by using the TCP/IP proxy protocol for several applications of network. After the IP of a computer is compromised, bot commander can use this IP to propagate the massive spams, malware, phishing and fraud email to various email address. This is achieved by stealing an IP address of any bot and in conjunction with other bots, the bot commander sends these massive spam emails. Also, a zombie can act as a packet sniffer to monitor the traffic and ongoing activities over the network with the help of infected machines. Typically, these sniffers look for the username and passwords for different accounts which a bot commander can use later for its personal interests (Iftikhar et al., 2020).

Warez

Another application of the botnets is Warez. As shown in figure 2, warez is technique in the world of hacking that is being used for stealing the licenses of the software or applications. botnets possess the tendency to steal, store or propagate Warez. They can do this by scanning the hard drives of the infected machines looking for the software and applications that are licensed. After successful searching, the herder can easily transfer or duplicate that license and can distribute over the internet thus violating the copyrights of the software (Iftikhar et al., 2020).

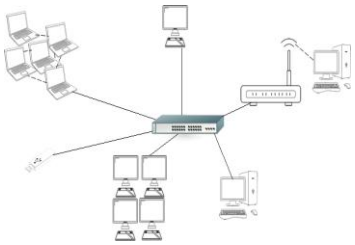

Figure 2: Illustration of Warez (Iftikhar et al., 2020)

There are several datasets created by researchers for testing intrusion detection methods. However, the number of features in these datasets is usually large. Applying intrusion detection methods directly on the original feature set may have some drawbacks, such as the high model complexity and overfitting. Thus, utilizing feature selection techniques to reduce the number of features helps speed up detection model inference. In this work, we propose a feature selection technique based on genetic algorithm for botnet detection in IoT environments (Liu & Du, 2023).

Although many different machine learning algorithms have been put forth in the literature to develop various botnet detection models, almost all of these models and techniques are based on extracting features (or feature development), where various feature sets are derived from the available high-dimensional datasets based on some expertise and skills. On the other hand, it is discovered that the literature on botnet detection pays little attention to feature selection, which plays a crucial role in developing various machine learning models(Obeidat & Yaqbeh, 2022).

Early botnet detection mainly relied on blacklist or whitelist method, although the botnet detection accuracy of this method is high, but the ability to detecting unknown attacks is weak. In recent years, the botnet detection methods based on ML and neural network have become research hotspots. The basic idea is based on network traffic characteristics, and then applies these characteristics to build a model of identifying normal and abnormal traffic. These methods have the advantage of discovering unknown attacks(Duan et al., 2022).

Moreover, various studies have explored innovative approaches to enhance IDS performance. For instance, one study proposed a meta-classification strategy utilizing stacked generalization, achieving an accuracy of 97% on real network traffic datasets (Lo et al., 2022) Another research effort focused on a double Particle Swarm Optimization algorithm to optimize hyperparameters and feature subsets, employing deep learning models such as Deep Belief Networks and Long Short-Term Memory Recurrent Neural Networks (Jain & Waoo, 2023; "Decision Tree: A Machine Learning for Intrusion Detection", 2019). These advancements illustrate the ongoing evolution of intrusion detection methodologies, emphasizing the importance of integrating sophisticated algorithms to address the challenges posed by modern cyber threats.

III. METHODOLOGY

This review explores machine learning approach to detect botnet activity in network traffic. The methodology involves hybridizing algorithms to achieve the following objectives; feature selection using AFSA, class balancing with SMOTE, and classification of normal traffic and malicious traffic using a CS-SVM on linear support vector class.

Machine Learning Technique

Cost-sensitive Support Vector Machines (SVM) assign different penalties for misclassifying instances based on their class labels. This is achieved by introducing cost parameters (C1 and C2) that penalize misclassification of negative and positive instances, respectively (Guido et al., 2022). The objective function in a cost-sensitive SVM aims to maximize the margin while minimizing the weighted sum of misclassification errors (Guido et al., 2022).

1. Penalty Weights (C1, C2): These parameters control the penalty for misclassifying instances of each class. A higher penalty weight means a greater emphasis on correctly classifying instances of that class (Guido et al., 2022).
2. Hinge Loss: The standard SVM hinge loss is modified to incorporate the cost parameters. The loss for misclassifying an instance is then proportional to the cost parameter associated with that instance's class.
3. Dual Formulation: The dual formulation of the cost-sensitive SVM is often used for efficient optimization, especially with kernel methods.
4. Function (Primal Formulation):

Minimize $(1/2)\ ||w||^2 + C1\ \Sigma(\xi i) + C2\ \Sigma(\zeta i)$

Subject to:

$yi\ (wTx + b) >= 1 - \xi i,\ i \in I$ (for positive class)

$-yi(wTx + b) >= 1 - \zeta i,\ i \in I$ (for negative class)

$\xi i, \zeta i >= 0, i \in I$

Where:
w: Weight vector
b: Bias term
xi: Input data point
yi: Class label (+1 or -1)
ξi, ζi: Slack variables (representing misclassification errors)
C1, C2: Cost parameters for positive and negative classes.

Function (Dual Formulation):
Maximize L(α, α*, β) = Σ(αi) - Σ(α*i) - (1/2) Σ Σ αi αj yiyj K(xi, xj)
Subject to:
Σ(αi - α*i) = 0
0 <= αi <= C1
0 <= α*i <= C2

Where:
αi, α*i: Lagrange multipliers (dual variables)
β: Bias term (computed from support vectors)
K(xi, xj): Kernel function (e.g., RBF, linear kernel).
In essence, the cost-sensitive SVM modifies the standard SVM formulation to penalize misclassifications of different classes differently. This allows for fine-tuning the model to prioritize minimizing errors on specific classes, which is particularly useful when the classes are imbalanced or when misclassifying certain classes has a greater cost than others (Guido et al., 2022).

Classification Imbalance Technique
SMOTE is a popular technique used to address the issue of class imbalance in classification tasks, particularly in datasets where one class (e.g., malicious network activity) is significantly underrepresented compared to another (e.g., benign activity). Below is how SMOTE work mathematically:

Step1: Identify minority class

$Let\ D_{min}=\{x1,x2,\ldots\ldots\ldots xn\} CD\ be\ the\ set\ of\ all\ class\ samples$

Step2: For each xi $xi \in D_{min}$
Find K Nearest neighbor Nk(xi)
For each neighbor $x_{zi} \in N_k$(xi) generate:
x_{new}^{j} = xi + π_j (x_{zi} +x_i) for j=1 …… N
Where N is the number of synthetic needed.

Feature Selection Technique
The mathematical implication of (AFSA) lies in its modeling of fish behavior to solve optimization problem e.g foraging, swarming and following. In the context of this study of detecting botnet, AFSA searches for the optimal subset of features that maximizes detection performance.
Max f(xi)
Xi ≤F
Where F is the full feature set
Xi is a subset
F(xi) function of evaluation

Data collection methods
For this research, the data collection process will involve obtaining the KDD CUP 99.csv file dataset from GitHub. The datasets used and analyzed in this study is obtained upon rational request from the corresponding author.

Performance Measurement
The simulation results were evaluated based on the following metrics: accuracy, recall, and precision. These metrics are defined with the botnet class considered as the positive class, where TP (true positives), FP (false positives), FN (false negatives), and TN (true negatives) are measured as follows:

Accuracy: The percentage of events that were successfully predicted compared to the total of all predictions:

$$\frac{TP+TN}{TP+TN+FP+FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Precision: All true positives divided by all positive predictions, presented as follows:

$$\frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Recall: true positives divided by positive results. This pattern indicates that out of all potential positives, how many were found by the model?

$$\frac{TP}{TP+FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

F1-score: The F1-score is the harmonic mean of Precision and Recall. It balances the trade-off between these two metrics, especially when the data is imbalanced.

$$F1-score = 2x\frac{Precision\ x\ Recall}{Precision+Recall} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

IV. DATA ANALYSIS AND RESULTS

Feature Selection and Normalization

In the first stage, we utilized the Artificial Fish Swarm Algorithm (AFSA) to select the most fundamental features from the KDDCup 99 dataset network traffic. The dataset, which contains 41 features, was uploaded and processed. The feature selection process run for sometime to complete. Table 2 below show the result of the selected features. The first row is the selected feature mask, second row is the number of selected features then lastly, accuracy of the fitness score which was at 99.9%.

Table 2: Features Selected

Selected features mask	Number of selected features	Fitness score (accuracy):
0 1 1 1 0 1 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 1 0	17	0.9998733318666373

Table 4 below is a list of the selected features along with their descriptions. The new dataset was saved as a CSV file and will be apply for balancing utilizing the Synthetic Minority Oversampling Technique (SMOTE).

Table 3: Features Selected and Description

Feature	**Description**
protocol_type	Type of protocol used (e.g., TCP, UDP, ICMP).
service	Network service on the destination (e.g., HTTP, FTP, Telnet).
flag	Status of the connection (e.g., normal or error state).
dst_bytes	Number of bytes sent from the source to the destination.
hot	Number of "hot" indicators (suspicious actions) in the connection.
num_compromised	Number of compromised conditions in the connection.
root_shell	1 if root shell is obtained; 0 otherwise.
num_root	Number of root accesses or attempts.
num_file_creations	Number of file creation operations.
srv_count	Number of connections to the same service as the current one in the past 2s.
srv_rerror_rate	% of connections with “REJ” errors to the same service.
same_srv_rate	% of connections to the same service.
diff_srv_rate	% of connections to different services.
srv_diff_host_rate	% of connections to different hosts using the same service.
dst_host_count	Number of connections to the same destination host.
dst_host_same_src_port_rate	% of connections to the same destination host using the same source port.
dst_host_rerror_rate	% of connections to the destination host that had “REJ” errors.

Balancing Dataset with SMOTE

At this stage, (SMOTE) was applied to the reduced Kddcup 99 dataset. The process required uploading the reduced dataset, then generating synthetic

samples to balance the classes, and converting the final balanced data into a DataFrame for saving. Before applying SMOTE, the dataset had 99% normal traffic and only 1% botnet traffic. After using SMOTE, the dataset was balanced to have 50% normal traffic and 50% botnet traffic.

Table 4: Distribution of Traffic Before and After SMOTE

Before SMOTE	
Normal Traffic	99%
Botnet Traffic	1 %

After SMOTE	
Normal Traffic	50%
Botnet Traffic	50%

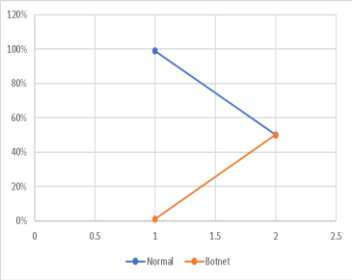

Figure 5: Distribution of Traffic Before and After SMOTE

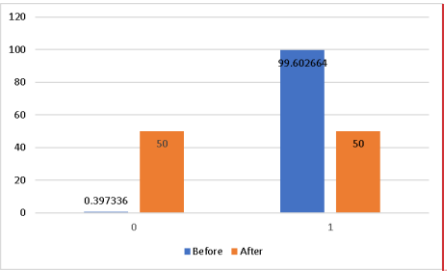

Figure 6: Distribution of Traffic Before and After SMOTE

Application of CS - SVM

The cost-sensitive Support Vector Machine (SVM) has two types of classifiers: SVC with an RBF kernel and Linear SVC. First, we tested the SVC with the RBF kernel and achieved 90% accuracy, which was lower than the accuracy reported in the base paper. Next, we tried the Linear SVC, which is known to perform better and run faster on large datasets. This kernel gave us a much higher accuracy of 99%. This result shows an improvement over the accuracy reported in the base paper. Table 6 shows a comparison between the SVC with RBF kernel and the linear SVC.

Table 5: Comparison of Non-Linear SVC vs Linear SVC

	SVC	Linear SVC
Accuracy	90 %	99 %

Performance Evaluation

Table 6 presents a summary of the evaluation metrics, indicating that the model achieved a classification accuracy of 99.5% on the test data.

Table 6: Summary of evaluation matrix

Accuracy	0.9957
Precision	0.9957
Recall	0.9957
F1 Score	0.9957

Table 7 stated that the class of normal traffic has a precision of 99% of instances predicted as normal, a recall of 100% as actual normal traffic without missing any, and an F1 score of 100% perfect balance of precision and recall. Then the class of botnet traffic has the precision of 100% of every instance predicted as a botnet attack being correct, a recall of 99% of the actual attacks, and an F1 score of 100%, which is nearly perfect.

Table 7: Classification Report Analysis

	Precision	Recall	F1-Score	Support
0 (Normal)	0.99	1.00	1.00	44943
1 (Botnet)	1.00	0.99	1.00	44699

Commented [AM1]: put a graph of this results

Accuracy			1.00	89642
Macro Avg	1.00	1.00	1.00	89642
Weighted Avg	1.00	1.00	1.00	89642

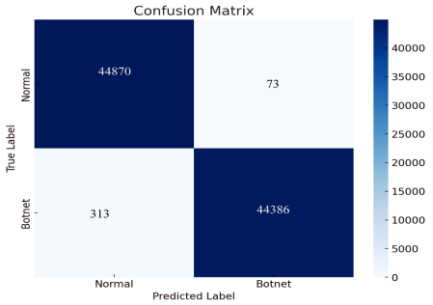

Figure 7: Confusion Matrix

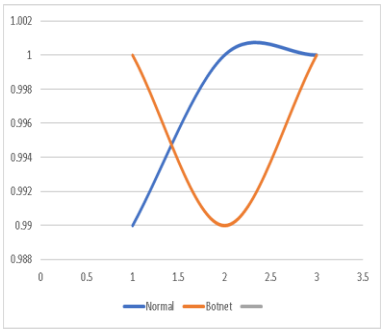

Figure 8 : Model Evaluation Matrix

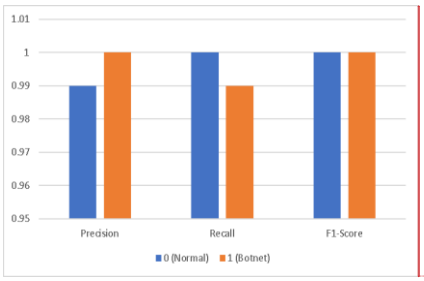

Figure 9 : Model Evaluation Matrix

V. DISCUSSION

The development of this research was to develop an improved algorithm technique for detecting and eliminating botnet in network system using a mixing preprocessing features selection techniques synthetic minority oversampling techniques (SMOTE) for class balancing, and a cost-sensitive support vector machine for classification. our model achieved outstanding performance metrics from the successful implementation of the proposed methodology. The accuracy achieved was 99%, precision of 99%, recall 99.5% and f1- score of 99.5%. In addition, classification report shows nearly perfection performance on both normal and botnet traffic are shown below:

Normal Traffic (0)	Rate	Botnet Traffic (1)	Rate
Precision	0.99	Precision	1.00
Recall	1.00	Recall	0.99
F1-score	1.00	F1-score	1.00

The above classification results demonstrated that the model accurately identifies malicious traffic with minimal false positive and false negatives. The combination of AFSA reduced the feature space, improve speed without sacrificing performance. SMOTE effectively balance the dataset and address the issue of class imbalance. Finally, the CS-SVM classifier illustrated its strength in handling distribution of network classes.

Comparative Analysis

Author	Accuracy	Precision	Recall	F1-Score
Hussain et al. 2021	96.00	95.00		95.40
Catillo et al. 2022	94.50	93.80		94.10

Commented [AM2]: putt a graph also

Li Duaan et al. 2022	96.50	94.90		95.60
Obeidat & Yaqbeh 2022	97.70	97.60	97.40	97.50
Jain & Waoo 2023	91.60	89.70		90.50
Velasco-Mata et al 2023	94.20	93.50		93.90
Nadeem et al. 2023	95.70	95.10		95.40
Wardana et al. 2024	97.60	96.50		96.90
Proposed	99.60	99.60	100.00	100.00

CONCLUSION

In conclusion, the proposed research proved to be highly efficient and effective in botnet detection. Combining AFSA, SMOTE, and CS-SVM yielded a system that is highly accurate, robust and scalable. The confusion matrix verified that confirmed that the objectives of the study were achieved. This study contributes to the field of network security by presenting a practical approach that enhances detection accuracy for imbalance network intrusion data. This model indicates its suitability for real-time network intrusion detection system.

RECOMMENDATIONS AND FUTURE WORK

We recommend that future studies should focus on the real-time deployment of the proposed model to assess its practical effectiveness in live network environments. Additionally, expanding the dataset by incorporating more diverse and up-to-date network traffic data will help improve the model's generalizability and robustness across different types of botnet attacks. Furthermore, the design of a web user interface will enhance accessibility and usability for researchers and cybersecurity practitioners, enabling them to easily interact with the system, upload datasets, visualize results, and monitor network traffic in real time. In the future we intend to implement real-time detection using live network traffic streams. reduce computational overhead through parallelization of AFSA.

REFERENCE

[1] Ali, S., Ghazal, R., Qadeer, N., Saidani, O., Alhayan, F., Masood, A., Saleem, R., Khan, M. A., & Gupta, D. (2024). A novel approach of botnet detection using hybrid deep learning for enhancing security in IoT networks. Alexandria Engineering Journal, 103, 88–97. https://doi.org/10.1016/j.aej.2024.05.113

[2] Alissa, K., Alyas, T., Zafar, K., Abbas, Q., Tabassum, N., & Sakib, S. (2022). Botnet Attack Detection in IoT Using Machine Learning. Computational Intelligence and Neuroscience, 2022. https://doi.org/10.1155/2022/4515642

[3] Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., & Alfaris, R. (2012). Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art. In International Journal of Computer Applications (Vol. 49, Issue 7).

[4] Amin Mohd Yunus, M., Zainulariff Brohan, M., Mohd Nawi, N., Salwana Mat Surin, E., Azwani Md Najib, N., & Wei Liang, C. (n.d.). Review of SQL Injection : Problems and Prevention.

[5] Bandara Mailewa, A. (2019). Security threats/attacks via botnets and botnet detection & prevention techniques in computer networks: AReview.https://www.researchgate.net/publication/334126617

[6] Bederna, Z., & Szadeczky, T. (2020). Cyber espionage through Botnets. Security Journal, 33(1), 43–62. https://doi.org/10.1057/s41284-019-00194-6

[7] Bhattacharya, S., Khanna, A., & Dubey, R. (2024). Botnet Detection and Mitigation: A Comprehensive Literature Review. International Journal of Computer Trends and Technology, 71(1),7782.https://doi.org/10.14445/22312803/ijctt-v72i1p113

[8] Catillo, M., Pecchia, A., & Villano, U. (2022, August 23). Botnet Detection in the Internet of

Things through All-in-one Deep Autoencoding. ACM International Conference Proceeding Series. https://doi.org/10.1145/3538969.3544460

[9] Dagon, D., Zou, C., & Lee, W. (n.d.). Modeling Botnet Propagation Using Time Zones. http://www.cc.gatech.edu/

[10] Gelgi, M., Guan, Y., Arunachala, S., Samba Siva Rao, M., & Dragoni, N. (2024). Systematic Literature Review of IoT Botnet DDOS Attacks and Evaluation of Detection Techniques. In Sensors (Vol. 24, Issue 11). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/s24113571

[11] Gu, G., Yegneswaran, V., Porras, P., Stoll, J., & Lee, W. (2009). Active botnet probing to identify obscure command and control channels. Proceedings - Annual Computer Security Applications Conference, ACSAC, 241–253. https://doi.org/10.1109/ACSAC.2009.30

[12] Haner, J. K., & Knake, R. K. (2021). Breaking botnets: A quantitative analysis of individual, technical, isolationist, and multilateral approaches to cybersecurity. Journal of Cybersecurity, 7(1). https://doi.org/10.1093/cybsec/tyab003

[13] Hoang, X. D., & Nguyen, N. T. (2019). Detectingwebsite defacements based on machine learning techniques and attack signatures. Computers, 8(2). https://doi.org/10.3390/computers8020035

[14] Iftikhar, U., Asrar, K., Waqas, M., & Ali, S. A. (2020). BOTNETs: A Network Security Issue: From Definition to Detection and Prevention. International Journal of Advanced Computer Science and Applications, 11(11), 432–436. https://doi.org/10.14569/IJACSA.2020.0111155

[15] Jaafar, G. A., Abdullah, S. M., & Ismail, S. (2019). Review of Recent Detection Methods for HTTP DDoS Attack. In Journal of Computer Networks and Communications (Vol. 2019). Hindawi Limited. https://doi.org/10.1155/2019/1283472

[16] Karasaridis, A., Rexroad, B., & Hoeflin, D. (n.d.). Wide-scale Botnet Detection and Characterization.

[17] Lin, H. C., Wang, P., Lin, W. H., & Huang, Y. H. (2021). A multiple-swarm particle swarm optimisation scheme for tracing packets back to the attack sources of botnet. Applied Sciences (Switzerland), 11(3), 1–22. https://doi.org/10.3390/app11031139

[18] Lin, K. C., Chen, S. Y., & Hung, J. C. (2014). Botnet detection using support vector machines with artificial fish swarm algorithm. Journal of Applied Mathematics, 2014. https://doi.org/10.1155/2014/986428

[19] Liu, X., & Du, Y. (2023). Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm. Electronics (Switzerland), 12(5). https://doi.org/10.3390/electronics12051260

[20] Lo, W. W., Kulatilleke, G. K., Sarhan, M., Layeghy, S., & Portmann, M. (2022). XG-BoT: An Explainable Deep Graph Neural Network for Botnet Detection and Forensics. http://arxiv.org/abs/2207.09088

[21] Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. International Journal of Distributed Sensor Networks, 13(12). https://doi.org/10.1177/1550147717741463

[22] Moorthy, R. S. S., & Nathiya, N. (2022). Botnet Detection Using Artificial Intelligence. Procedia Computer Science, 218, 1405–1413. https://doi.org/10.1016/j.procs.2023.01.119

[23] Nadeem, M. W., Goh, H. G., Aun, Y., & Ponnusamy, V. (2023). Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning Techniques. IEEE Access, 11, 49153–49171. https://doi.org/10.1109/ACCESS.2023.3277397

[24] Nogueira, A., Salvador, P., & Blessa, F. (2010a). A botnet detection system based on neural networks. 5th International Conference on Digital Telecommunications, ICDT 2010, 57–62. https://doi.org/10.1109/ICDT.2010.19

[25] Nogueira, A., Salvador, P., & Blessa, F. (2010b). A botnet detection system based on neural networks. 5th International Conference on Digital Telecommunications, ICDT 2010, 57–62. https://doi.org/10.1109/ICDT.2010.19

[26] Obeidat, A., & Yaqbeh, R. (2022a). Smart Approach for Botnet Detection Based on Network Traffic Analysis. Journal of Electrical and Computer Engineering, 2022. https://doi.org/10.1155/2022/3073932

[27] Obeidat, A., & Yaqbeh, R. (2022b). Smart Approach for Botnet Detection Based on Network Traffic Analysis. Journal of Electrical and Computer Engineering, 2022. https://doi.org/10.1155/2022/3073932

[28] Owen, H., Zarrin, J., & Pour, S. M. (2022). A Survey on Botnets, Issues, Threats, Methods, Detection and Prevention. Journal of Cybersecurity and Privacy, 2(1), 74–88. https://doi.org/10.3390/jcp2010006

[29] Rajab, M. A., Zarfoss, J., Monrose, F., & Terzis, A. (2006). A Multifaceted Approach to Understanding the Botnet Phenomenon.

[30] Shankar, A., Shetty, R., & Nath, B. (2019). A Review on Phishing Attacks. In International Journal of Applied Engineering Research (Vol. 14, Issue 9). http://www.ripublication.com

[31] Songma, S., Netharn, W., & Lorpunmanee, S. (2024). EXTENDING NETWORK INTRUSION DETECTION WITH ENHANCED PARTICLE SWARM OPTIMIZATION TECHNIQUES. International Journal of Computer Networks and Communications, 16(4), 61–85. https://doi.org/10.5121/ijcnc.2024.16404

[32] Strayer, W. T., Lapsely, D., Walsh, R., & Livadas, C. (2008). Botnet detection based on network behavior. Advances in Information Security, 36, 1–24. https://doi.org/10.1007/978-0-387-68768-1_1

[33] Taylor, O. E., & Ezekiel, P. S. (2022). A Smart System for Detecting Behavioural Botnet Attacks using Random Forest Classifier with Principal Component Analysis. European Journal of Artificial Intelligence and Machine Learning, 1(2), 11–16. https://doi.org/10.24018/ejai.2022.1.2.4

[34] Thanh, S. N., Stege, M., El-Habr, P. I., Bang, J., & Dragoni, N. (2021). Survey on botnets: Incentives, evolution, detection and current trends. Future Internet, 13(8). https://doi.org/10.3390/fi13080198

[35] Vania, J., Meniya, A., & Jethva, H. B. (n.d.). A Review on Botnet and Detection Technique. International Journal of Computer Trends and Technology. http://www.internationaljournalssrg.org

[36] Velasco-Mata, J., González-Castro, V., Fidalgo, E., & Alegre, E. (2023). Real-time botnet detection on large network bandwidths using machine learning. Scientific Reports, 13(1). https://doi.org/10.1038/s41598-023-31260-0

[37] Wardana, A. A., Kołaczek, G., Warzyński, A., & Sukarno, P. (2024). Ensemble averaging deep neural network for botnet detection in heterogeneous Internet of Things devices. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-54438-6

[38] Zeidanloo, H. R., Bt, A., & Manaf, A. (2010). Botnet Detection by Monitoring Similar Communication Patterns. In IJCSIS) International Journal of Computer Science and Information Security (Vol. 7, Issue 3). http://sites.google.com/site/ijcsis/

[39] Zhao, L., Kang, H.-S., & Kim, S.-R. (2013a). Improved Clustering for Intrusion Detection by Princi-pal Component Analysis with Effective Noise Reduction. 490–495. https://doi.org/10.1007/978-3-642-36818-9_55ï

[40] Zhao, L., Kang, H.-S., & Kim, S.-R. (2013b). Improved Clustering for Intrusion Detection by Princi-pal Component Analysis with Effective Noise Reduction. 490–495. https://doi.org/10.1007/978-3-642-36818-9