Hybrid Machine Learning Technique to Detect Active Botnet Attacks for Network Security and Privacy

[1]Venkatesan C, [2]Thamaraimanalan T, [3]Balamurugan D, [4]Gowrishankar J, [5]Manjunath R and [6]Sivaramakrishnan A

[1]Department of Electronics and Communication Engineering, HKBK College of Engineering, Karnataka, India.
[2]Department of Electronics and Communication Engineering, Sri Eshwar College of Engineering, TamilNadu, India.
[3]Department of Computer Science and Engineering, Sona College of Technology, Salem, TamilNadu, India.
[4]Department of Computer Science and Engineering, Jain (Deemed-to-be University), Bangalore, Karnataka, India.
[5]Department of Computer Science and Engineering, R R Institute of Technology, Bangalore, Karnataka, India.
[6]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India.
[1]venkatesanc.ec@hkbk.edu.in, [2]thamaraimanalan.t@sece.ac.in, [3]balamurugand@sonatech.ac.in,
[4]gowrishankar.j@jainuniversity.ac.in, [5]drmanjunath.raj@gmail.com, [6]arulsivaram@kluniversity.in

Correspondence should be addressed to Venkatesan C : venkatesanc.ec@hkbk.edu.in.

Article Info
Journal of Machine and Comp uting (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202303044
Received 15 April 2023; Revised from 25 July 2023; Accepted 25 August 2023.
Available online 05 October 2023.

Abstract – A botnet is a malware application controlled from a distance by a programmer with the assistance of a botmaster. Botnets can launch enormous cyber-attacks like Denial-of-Service (DOS), phishing, spam, data stealing, and identity theft. The botnet can also affect the security and privacy of the systems. The conventional approach to detecting botnets is made by signature-based analysis, which cannot discover botnets that are not visible. The behavior-based analysis appears to be an appropriate solution to the current botnet characteristics that are constantly developing. This paper aims to develop an efficient botnet detection algorithm using machine learning with traffic reduction to increase accuracy. Based on behavioural analysis, a traffic reduction strategy is applied to reduce network traffic to improve overall system performance. Several network devices are typically used to retrieve network traffic information. With a detection accuracy of 98.4%, the known and unknown botnet activities are measured using the supplied datasets. The machine learning-based traffic reduction system has achieved a high rate of traffic reduction, about 82%, and false-positive rates range between 0% to 2%. Both findings demonstrate that the suggested technique is efficient and accurate.

Keywords – Network Security; Botnet Attacks; Denial Of Service; Traffic Reduction; Machine Learning.

I. INTRODUCTION

In recent days, Botnets have recently emerged as a severe danger to secure communication. A botnet network is a pretended system that a Botmaster governs. A botnet is a network of bots and is a significant hazard to network-related operations and applications. When personal computers (PCs) get infected while browsing the Internet or downloading software with malicious code, the system becomes a bot and executes commands directed by the botmaster. These commands are used to steal the secret information of the users. The conventional methods of infecting a computer using a bot are Drive-by Download, Email, and Pirated software. For example, while downloading a software application, the malicious code developed by malware developers may be installed when the executable file is opened. Support vector machine and Grey Wolf Optimization algorithms were used to identify the bot's existence [1].

A botnet's life span is divided into four phases: creation, command and control (C&C), attack, and post-attack. The creation phase occurs when some attacker activities a known susceptibility in a target to infect the target machine, then uses newly attained access to run different programs that obtain a malicious binary from a recognized location. After the malware is mounted, the victim's PC accomplishes the malicious code and transforms it into a bot. Now, the bot will connect to the C&C server using various methods, and once this link is formed, it will formally join the botnet [2]. Botnet traffic is quite unusual from the other malware traffic since it possesses C&C communication channels. When the botmaster

acquires these channels, it sends commands to the botnet's members to carry out suspicious actions. Furthermore, the botmaster obtains the necessary information for attacks through these routes. C&C communication channel traffic happens before the execution of attack actions and can be considered intelligence communication among the botnet participants. Botnet execution is grouped into two phases such as the infection phase and the attack phase. During the infection phase, a botmaster increases the bot group to increase the number of infected computers. The botmaster attacks the identified group of bots during the attack phase. An infected machine sends essential information to the botmaster, considering the infected system's IP address, login name, and operating system [3]. The organization of the C&C phase and Attack phase is shown in **Fig 1**.

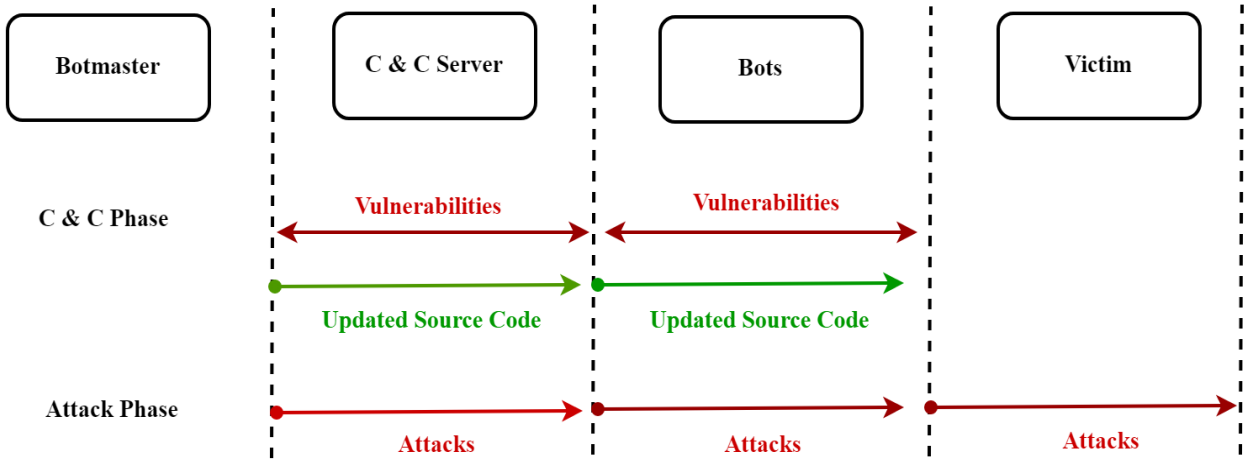

Fig 1. Organization of Command & Control (C&C) phase and Attack phase

This research paper presents a hybrid machine-learning algorithm for botnet detection with a reduced traffic rate. Literature related to botnet attacks and detection is discussed in section 2. Machine learning and traffic reduction-based botnet detection are discussed in section 3. Section 4 discusses the proposed hybrid machine-learning algorithm for active and inactive botnet detection. Section 5 covers the simulation outcomes and discussion. Finally, section 6 delivers the conclusion of the proposed methodology.

II. RELATED WORKS

Complex Event Processing (CEP) is another popular methodology similar to the correlation technique. It compares to correlate the measures in real-time to identify a complex targeted event. The target event may consist of many simple events or complex events. CEP mechanism is applied in information security and network security. The two essential categories of correlation-based detection are vertical and horizontal correlation. Horizontal correlation techniques help observe similarities and differences in host behavior and communication in a network [4].

Botnet detection is accomplished via a Peer to Peer (P2P) method that interprets botnet transmission as stream data. Because history data has an infinite period and drift, a novel multi-chunk ensemble classification technique is proposed, in which trained classifiers are stored instead of old data. When evaluated with botnet traffic data, the proposed approach beats traditional classification techniques for detection accuracy. Building and managing botnets have two significant motivations: money gained from botnets for hire and political benefits from cyber terrorism or nation-states. Botnets offer extensive services, from crypto-mining to information collection to anonymous large-scale cyber-attacks [5].

The bot catcher approach concentrates on malicious activity patterns and their communication in the network. Based on the obtained information, it performs a cross-cluster correlation that can be used to record the bots in the monitored network [6]. Bot Sniffer is another similar approach based on the bot families to detect bots. This approach utilizes spatial and temporal characteristics to identify the similarities among bot families. Bot families and their details are not mandatory in botnet detection. Users and network administrators may not concern about bot family details. The main duty is to protect networks from malicious activities without concern about bot families.

Conversely, security researchers pay more attention to bot family detection. The vulnerability of the host machine or network to botnet infection can be identified based on bot family detection. Remedial strategies can be initiated once the bot family and bot details are identified. These strategies aim to recover the bots from infection and protect bots from malicious activities and infection in the future [7, 8]. An investigation is necessary to evaluate the hidden bot, payload data, and deep packet inspection, which cannot be adequately performed. The behavior-based analysis looks to be a viable option for distinguishing malware's present trends because it just needs the packet header. Using behavior-based analysis, connection, the pattern, and action derived from the message between the botmaster and the bots are monitored. Despite the benefits of malware activity-based analysis over signature-based analysis, most behavior detection methods are restricted to specific rules and botnet arrangements [9-10].

The authors proposed a deep learning-based approach for detecting botnets using DNS traffic analysis. A convolutional neural network (CNN) extracts features from DNS queries and classifies them as a botnet or non-botnet traffic. Domain Name System-based Blackhole List (DNSBL) is a technique used to block spam and other malicious traffic by maintaining a database of IP addresses or domain names associated with spam, malware, or botnets. The proposed method is evaluated using a large-scale dataset and achieves high accuracy in detecting botnets [11]. Botnet discovery has increased attention in present years due to the growing impact of botnets in spreading malware activities. Several research issues have been identified from the review in botnet detection. Signature-based detection methods are usually precise, but it requires a large amount of signature database in 5G networks [12]. The existing training techniques are not efficient, so there is a requirement for intelligent training algorithms and techniques. Artificial intelligent techniques enrich the detection quality and minimize the false-positive rate, but they are inefficient due to the training time and the number of nodes. So, traffic reduction algorithms must be incorporated to increase efficiency [13, 14].

In a zombie-infested network, detecting bots is a complicated process. Botnet identification is a significant worry, even if only one bot is identified in the same network, and the complexity of botnet detection increases when bots measure IoT devices. Botnet detection methods based on a supervised approach (classification) and an unsupervised approach (clustering) are essential in detecting this botnet. Data is frequently learned by machine learning algorithms, which then deliver a data-based prediction. The appropriate feature selection and machine learning algorithm method are the two most essential elements in any machine learning assignment [15-17]. Malware converts compromised computer systems into robots (bots) remotely controlled by the botmaster without the end-user's awareness. It isn't easy to manage the routing protocols [18].

The significant contributions of this paper include,

- A hybrid machine learning approach is used to provide better accuracy compared to conventional techniques.
- The botnet detection scheme focuses on detecting active botnet attacks to provide more network security and privacy.

III. MACHINE LEARNING AND TRAFFIC REDUCTION BASED BOTNET DETECTION

Due to the bot's ability to hide, deep packet inspection (DPI) and P2P necessitate an investigation, which can't be done quickly. Because it only needs the packet header, the behaviour-based analysis appears to be a potential approach for detecting malware's current tendencies. Machine learning (ML) in malware detection is necessitated by complicated processes involving time-consuming human monitoring techniques [19, 20]. Machine learning can learn the model data pattern and recognize a similar configuration despite its complexity. Classification is a supervised process that involves training an algorithm and predicting a class using labelled data. Clustering is an unsupervised approach that plots a similar pattern into clusters using unlabelled data and an algorithm.

Traffic behaviour analysis can be utilized to reduce the amount of traffic and be applied in encrypted network communication protocols. Network traffic information is typically retrieved with the help of numerous network devices. The block diagram of the traffic reduction-based botnet detection system is given in **Fig 2.** Botmaster communicates with the infected host through the C&C server. Hence the bot commands and reports could be received through this server. Before executing commands for botnet detection, a traffic reduction algorithm is used for efficient detection performance.

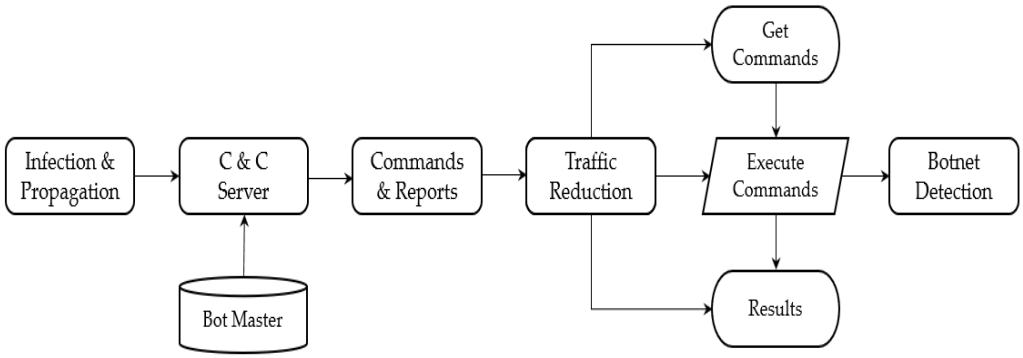

Fig 2. Traffic Reduction-Based Botnet Detection System

This botnet detection system (BDS) looks at the spatial-temporal correlation of botnet command and control activities. A traffic reduction algorithm has been incorporated to decrease network traffic volume so that the proposed system's bot detection performance can be further improved. This approach is based on the perception that the bots of the same botnet typically respond to the botmaster's commands to conduct fraudulent activities [21].

IV. PROPOSED BOTNET DETECTION SYSTEM (BDS) ALGORITHM

The bot detection algorithm steps in a botnet detection environment can be summarized as follows.

Step 1: The user's PCs are infected by propagating malware through the C&C server. First, the botnet operators create or obtain malware they want to spread. The bots then propagate the malware to other computers by various means, such as email spam, social engineering, or exploiting vulnerabilities in software.
Step 2: C&C server is controlled by the botmaster for propagation. The botnet operator uses the C&C server to distribute the malware to the bots.
Step 3: Initial commands and reports are obtained to identify the bot. When a bot is infected with malware and joins a botnet, it is typically controlled by the botnet operators through a C&C server. The botnet operators issue commands to the bot and receives reports from it to monitor its activities.
Step 4: Apply a traffic reduction algorithm to decrease the bot traffic and generates commands. Botnets can generate significant traffic volumes as bots communicate with the C&C server and propagate malware to other computers. This results in network congestion, reduced performance, and increased cyber-attack risk.
Step 5: Get the commands and apply the machine learning for botnet detection.

Using this approach, the botnet activity is monitored on the complete traffic flow behavior of the system. The classification is done based on the time intervals as follows:

- Traffic reduction: Analyzing the botnet activity with traffic reduction is necessary to save time. A botnet detection system can detect bots more efficiently with the help of an efficient traffic reduction algorithm.
- Attribute extraction: Botnet behavior is different from regular computers. The specific features of the bot have to be extracted in the bot detection methodology. An ideal set of features can be applied to bots.
- Pattern recognition: Pattern recognition is another essential criterion similar to feature extraction. The specific botnet features need to be understood clearly to classify input traffic. It should consume less time for recognition. A good recognition technique produces high detection rates and low false-positive rates.

Network Scenario

For a given network topology, the following assumptions are made regarding nodes. N is the nodes count in the chosen network and N_i are the neighbors count at a node x_i at a particular instant of time. The average number of neighbor nodes $\bar{n}$ in that time instant is given by Equation (1).

$$\overline{n} = \frac{\sum_{i=1}^{N} N_i}{N} \tag{1}$$

Now the average number of neighbors can be used to determine the maximum number of neighbors n_{max} and minimum number of neighbors, n_{min}. Let $N_1, N_2, \ldots, N_k$ be the number of neighbors at nodes $x_1, x_2, \ldots, x_k$ respectively, then the expectation for a maximum number of neighbors is given by Equation (2).

$$\overline{n_{\max}} = \frac{\sum_{i=1}^{N} N_i}{k} \tag{2}$$

Also, if $N_1, N_2, \ldots, N_r$ are the number of neighbors at nodes $y_1, y_2, \ldots, y_k$ respectively, such that $N_i < \mathrm{r}$, then the expectation for a minimum number of neighbours is given by Equation (3).

$$\overline{n_{\min}} = \frac{\sum_{i=1}^{r} N_i}{r} \tag{3}$$

Therefore, the expectation for minimum and maximum neighbours and the average number of neighbours are related by $\overline{n_{min}} < n < \overline{n_{max}}$. In computer networks, efficiency is decided based on reaching the correct destination. When a path has a failure node, the following arrival packets of that path are refused. If any of the paths are short and efficient, then it gives the first preference to that path. When the packets are received from that path, it chooses the shortest efficient path. This process is repeated until the entire transmission is completed. Finally, the entire network throughput is increased.

Execution of Bot

Fig 3 depicts the process of identifying the active and inactive bots by executing the bots. The bot is inactive if the DNS server does not detect the domain name and cannot be resolved or if it is impossible to reach the resolved IP addresses. A bot is active if one of the determined IP addresses is valid and can connect to a DNS server for further network connections. Each host has a host analyzer consisting of an in-host monitor and a suspicion-level generator. Two independent datasets are employed containing malicious activity from the honeynet project, including the Storm and Waledac botnets. These datasets can be accessed using the following **link: https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/botnet-and-ransomware-detection-datasets/**.

The in-host monitor monitors system-wide performance in the registry and on a host network in real time. The suspicious level generator uses a machine-learning algorithm to construct a suspicious level based on the performance recorded at each time window. Using an average moving technique, it calculates the overall suspicious level. The host analyser may

supply the network with the suspicious threshold level and rare network feature data to the correspondence machine. The infected machine is identified through IP address mapping, as depicted in **Fig 3.**

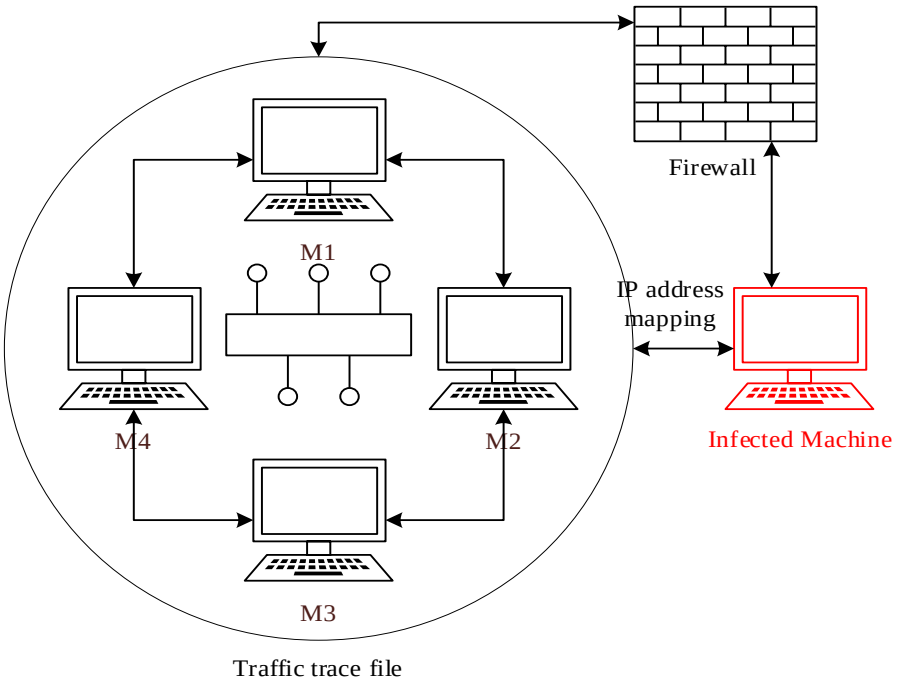

Fig 3. Dataset Merging Process

The clustering module comprises the network analyser, and the flow analyser gathers data after a router and looks for botnet-like trigger-action flow patterns across multiple hosts. The feature vectors are created by converting properties from the linked flows. The clustering module uses these vectors to group hosts that behave similarly and claim they're all part of the same botnet. The host analysers must report the mistrust level and network data to the correspondence machine if the network analyser identifies a doubtful group of hosts. By comparing network data between the network and the host, the integrity of the host information is verified in the correlation engine.

Dataflow Diagram

The loop is programmed to apply a hybrid strategy of classification and clustering to the dividers that comprise the whole dataset until the if statement is true. An evaluation of accuracies for classification, clustering, and hybrid techniques is shown at the end.

Botnet Detection Algorithm

Begin
The dataset must be identified.
The data is pre-processed for feature selection.
Run a clustering algorithm on the dataset and save the results in R11.
Sort the data into categories and save the results in R22.
Change n to 30 and n to 1.
Repeat steps 1–7 if 'i' is greater than 70 and less than 30, V1 = I and V2 = 100 – i
R11= clustering of V1, and the result is stored in R11.
R22 = classification of V2, and the results are stored in R22.
mean of [n] = Percentage of improper occurrences.
i = i+10, n++;
end the loop
Relate the mean of [n], R11, and R22 for further analysis
Stop

The rationale behind designing and operating a hybrid machine learning technique to detect active botnet attacks is to improve the accuracy and effectiveness of botnet detection while minimizing false positives. Botnets are networks of infected computers that are controlled by a C&C server. They are used for DDoS attacks, spamming, phishing, and malware

distribution. Botnets are challenging to detect and mitigate because they often use sophisticated techniques to avoid detection, such as changing their IP addresses, using encryption, and mimicking legitimate traffic.

The hybrid machine learning technique is used to analyse network traffic and identify patterns that indicate botnet activity. This technique involves selecting and extracting relevant features from the network traffic data and packet size. The hybrid technique also includes a feedback loop, where the results of the proposed algorithm is used to improve the accuracy of future detections. For example, if the algorithm detects a false positive, the feedback loop can retrain the algorithm and improve its accuracy.

The following stages outline the botnet detection algorithm utilizing the suggested hybrid technique. The data flow chart for the suggested model is illustrated in **Fig 4.**

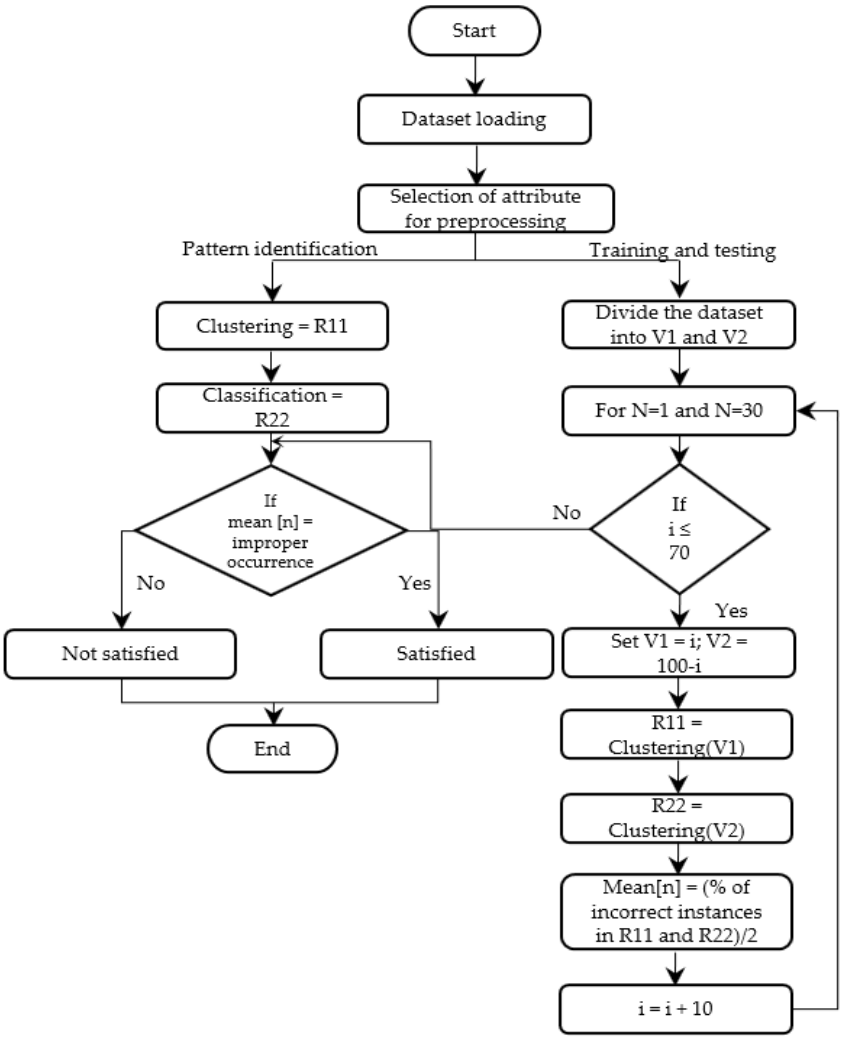

Fig 4. Flowchart of the Proposed Hybrid Approach

V. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

The proposed method is simulated in different configurations to assess the performance. The robustness of this method is tested in typical network traffic situations and abnormal scenarios, such as the presence of uncooperative routers and the network with configuration errors. The Windows operating system uses the honey trap to generate real botnet traces. Even though obtaining real router traces is possible in most environments, obtaining detailed topology is tricky for various reasons, such as security, privacy, and legal reasons. Hence a simulation-based method has been broadly used to analyse the performance of the proposed system.

A variety of topologies with varying numbers of rings have been investigated. The fan-out effect has been measured with the number of IPs between rings. It also establishes good routers to draw attention to the collaboration. The results remain unaffected by the number of routers used, with a slight delay in the time taken to identify an attack due to a more significant number of collaborating routers. Table 1 shows various simulation parameters used for the simulation.

The NS-2 simulator is used for simulation with a transmission range of 200 meters. The size of the network comprises 50 host machines.

The True Positive Rate (TPR) identifies the number of attacks correctly, and the number of False Positives (FPs) refers to the amount of harmless traffic wrongly classified as harmful. By measuring actual packet speeds, horizontal communication is employed to discard them. However, keeping the number of rules as small as possible is vital. Only a few rules are marked as false positives. The TPR and the FPs work on a per-rule and per-time-window basis, and an alert will be generated for true or false-positive conditions.

Table 1. Network Parameters used in NS-2 to Detect Botnets.

Simulation Parameters	Parameter Value
Botnet size	50 hosts
Transmission range	200 meters
Bot Name	Virut
Packet size	1024 bytes
Packet rate	12 packets/sec
Topology size	600 × 600 m2
Traffic type	Constant Bit Rate (CBR)
Bandwidth	2 Mbps
Number of nodes	20, 30, 40 and 50
Number of trials	40
Simulation time	1200 ms
Maximum speed	25 m/s
Maximum storage locations	5

The threshold value of each point determines the true positive rate. Each point is chosen using an average of the 30 simulation runs. When this number rises, a high-risk attack is suspected. The number of false positives and the true positive rates are reduced when the threshold value is reduced. When the value is 0.75 in a five-ring topology, the TPR is close to 100%. Different threshold settings are used to test the proposed bot detection system. **Fig 5** shows malicious IP addresses' false-negative rate (FNR) using various threshold levels based on time intervals. The number of time intervals affects the FNR. The value of FNR is computed using Equation (4).

$$FNR = \frac{FN}{FN+TP} \tag{4}$$

The false-negative rate rises when the threshold and time interval is increased. Some packets and malicious IP addresses may go undetected if the threshold value is too high. Botnet constraints and equivalent detection rates are shown in **Table 2.** From the table, it is detected that the number of malicious addresses detected is 85, and the detection rate of malicious IP addresses is 98.21%. In this scenario, the total DNS and TCP packets used are 312, 516, and 697, 362, respectively.

Table 2. Botnet Parameters and Corresponding Detection Rates

Botnet Parameters	Values
Number of Bots	50
Malicious IP addresses	85
DNS packets	312, 516
TCP packets	697,362
Malicious IP addresses detection rate	98.4%

The application of traces, including Internet relay chat (IRC), P2P traffic, and Hypertext Transfer Protocol (HTTP) generated by different users, have been used to evaluate traffic reduction rate and FPR. **Table 3** lists the total number of active and inactive attributes detected by malevolent IP addresses. The number of malicious IP addresses identified is 85. Out of the identified malicious IP addresses, the active and inactive malicious IP addresses are 45 and 40, respectively.

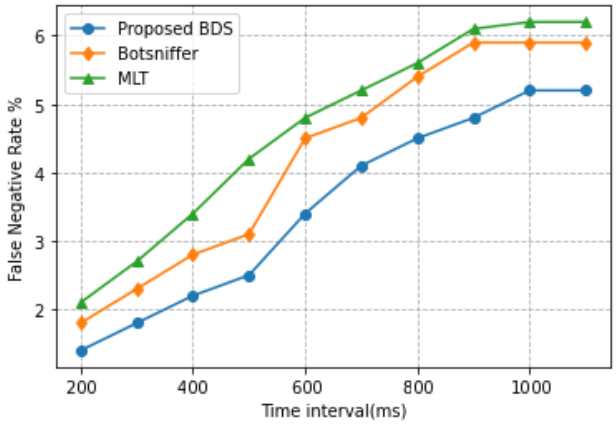

Fig 5. False-Negative rate for Malicious IP Addresses

Table 3. Active and Inactive Malicious IP Addresses

Attributes	Number of IP addresses
Active	45
Inactive	40
Total	**85**

In **Fig 6**, False Positive Rate (FPR) values are plotted based on the number of TCP connection requests for different botnet traces T1, T2, and T3. The FPR value decreases for all the traces while the number of TCP connection requests increases. The reason behind the decrease is the increase in the number of packets examined. The equivalent IP addresses can be identified as malevolent or not.

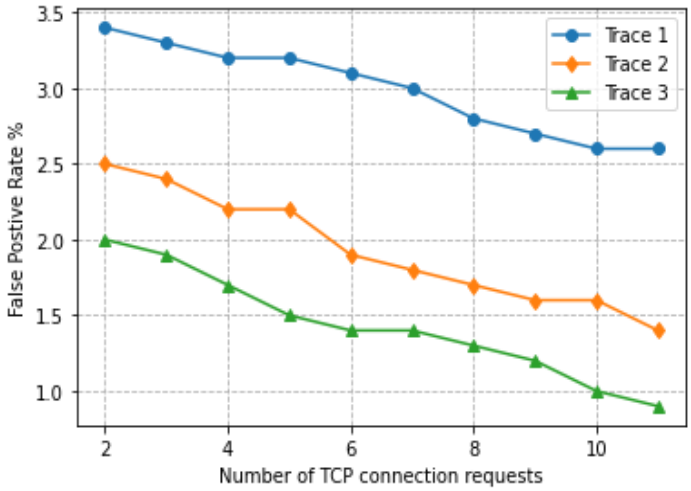

Fig 6. The False Positive Rate for TCP Requests

In **Table 4,** the proposed BDS algorithm attains high reduction rates and low FPR of malicious IP addresses. The number of DNS packets for the T1, T2, and T3 traces is 1232, 456, and 574, respectively. The TCP packets for the T1, T2, and T3 traces are 54367, 45632, and 44657, respectively. The trace T3 using the proposed BDS achieves a low FPR of 1.97 and has a high detection rate of 97.85% for malicious IP addresses. Note that the false-positive rates range from 1.97% to 2.14% for IP addresses, and the detection rate ranges from 96.15% to 97.85%. The threshold value is finalized based on the observation captured over malicious and regular traces. If the quantity of experimental data is adequate, the preferred rate can be extensively realistic, unlike networks.

Table 4. Statistics of Typical and Malicious Traces

Main features	Trace 1	Trace 2	Trace 3
DNS packets	1467	467	596
TCP packets	56185	46851	45716
False Positive Rate of malicious IP addresses	3.37%	2.16%	1.76%
Malicious IP addresses detection rate	92.37%	94.57%	96.71%
Reduction rate	75.6%	94.8%	96.2%

Fig 7 and **Fig 8** shows the simulation result evaluation of the proposed BDS algorithm's false-positive rate and detection rate with the machine learning technique (MLT) and Bot sniffer. The results show that the suggested algorithm has better false-positive and detection rates.

Table 5. Evaluation Analysis of Botnet Detection System (BDS)

Main features	MLT [16]	Botsniffer [5]	Proposed BDS
Adopted technique	Machine learning	Statistics based	Machine learning with traffic reduction
Traffic reduction	Not applicable	Not applicable	More than 82%
Detection of Inactive Bots	No	No	Yes
TPR	90.48%	86.81%	94.62%
FPR	0-13%	0- 5%	0-2%

Table 6. Comparison Table of the Proposed Hybrid Approach

Number of instances	% of accuracy		
	K-means Cluster	J48 Tree Classifier	Hybrid Approach
34223	49.58	52.54	61.74
45630	54.27	58.34	70.67
57038	61.49	67.84	78.45
68447	74.55	79.26	84.59
79854	84.20	88.24	92.14

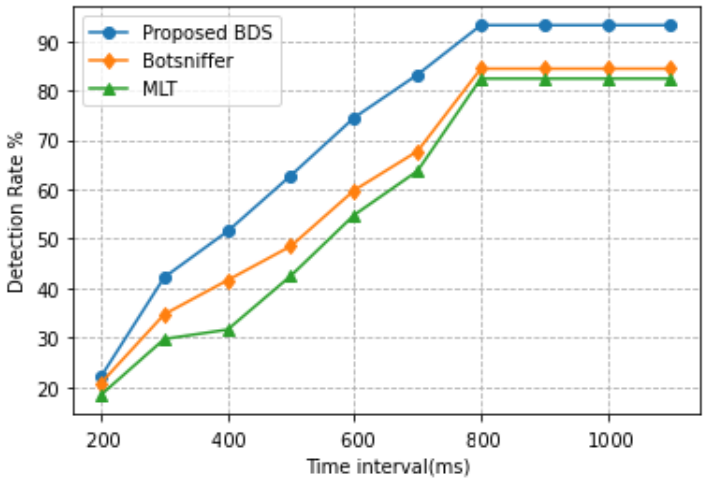

Fig 7. Comparative Analysis of Detection Rate (%)

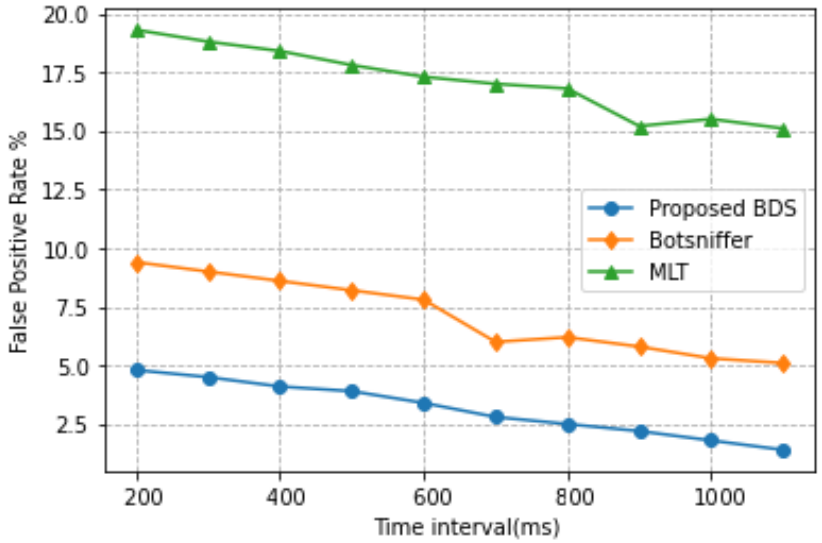

Fig 8. Comparative Analysis of False Positive Rate (%)

Fig 9 shows the percentage of loss for 1000 iterations. By comparing the three lines, it is observed that each technique performs over multiple iterations. The proposed botnet detection scheme appears to have the lowest loss percentage, while the machine learning technique has a higher loss percentage. The bot-sniffer technique has a higher loss percentage than the proposed botnet detection scheme but performs better than the machine-learning technique.

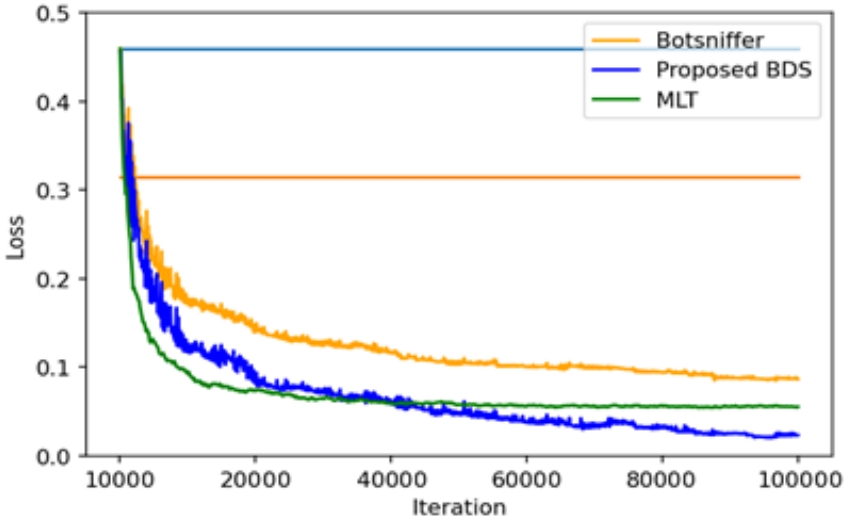

Fig 9. Percentage of Loss for Different Iterations

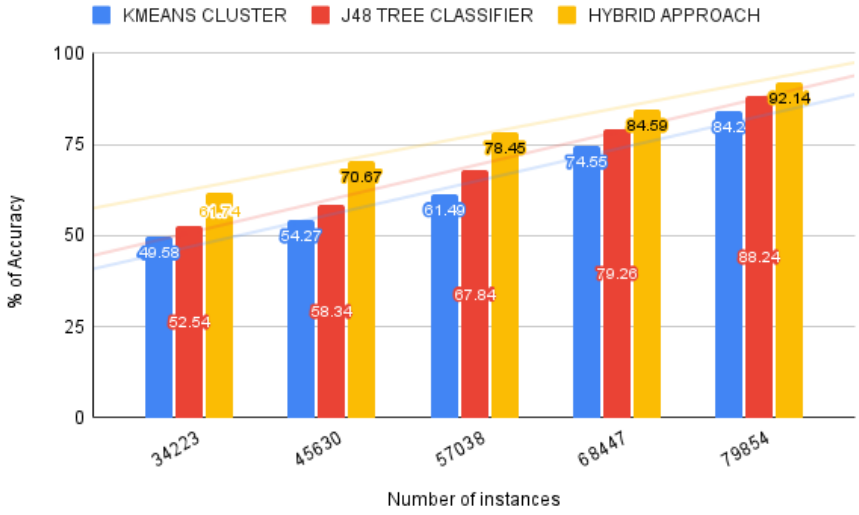

Fig 10. Comparison of the Proposed Hybrid Classifier for Different Instances

From **Fig 10**, the accuracy of the hybrid classifier is higher than the K-means and J48 tree classifier for most instances. At the 79854th iteration, the proposed classifier provides an accuracy of 92.14%, whereas the K-means classifier and J48 tree classifier deliver 84.2% and 88.24%, respectively.

VI. CONCLUSION

A machine learning-based traffic reduction technique is proposed to reduce network traffic and improve overall system performance. Tables 5 and 6 compare the proposed BDS with existing approaches in various ways. When compared to bot sniffers and machine learning techniques, the TPR and FPR are determined to be better with the extra traffic reduction. The results indicate a significant traffic reduction rate of more than 82 per cent and low false-positive rates of 0 to 2% based on typical traces. Both results show that the suggested method is both practical and precise. The suggested algorithm may also be used to detect slothful botnets that can be used to find potentially susceptible hosts. According to experimental data, the proposed hybrid technique offers a high detection rate of 98.4 per cent for bogus IP addresses. Novel hybrid classifiers will be created to test and assess new botnets based on performance and time. The results are compared against various machine learning techniques and botnets, which are tested using a variety of benchmark botnet datasets.

Data Availability
No data was used to support this study.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research

Ethics Approval and Consent to Participate
The research has consent for Ethical Approval and Consent to participate.

Competing Interests
There are no competing interests.

References

[1]. A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 7, pp. 2809–2825, Jul. 2019, doi: 10.1007/s12652-019-01387-y.
[2]. B. AsSadhan, A. Bashaiwth, J. Al-Muhtadi, and S. Alshebeili, "Analysis of P2P, IRC and HTTP traffic for botnets detection," Peer-to-Peer Networking and Applications, vol. 11, no. 5, pp. 848–861, Jul. 2017, doi: 10.1007/s12083-017-0586-0.
[3]. D. Santana, S. Suthaharan, and S. Mohanty, "What we learn from learning-Understanding capabilities and limitations of machine learning in botnet attacks," 2018, doi: 10.48550/arXiv.1805.01333.
[4]. G. Cugola and A. Margara, "Processing flows of information," ACM Computing Surveys, vol. 44, no. 3, pp. 1–62, Jun. 2012, doi: 10.1145/2187671.2187677.
[5]. S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid Botnet Detection Based on Host and Network Analysis," Journal of Computer Networks and Communications, vol. 2020, pp. 1–16, Jan. 2020, doi: 10.1155/2020/9024726.
[6]. G. Sagirlar, B. Carminati, and E. Ferrari, "AutoBotCatcher: Blockchain-Based P2P Botnet Detection for the Internet of Things," 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Oct. 2018, doi: 10.1109/cic.2018.00-46.
[7]. I. Sreeram and V. P. K. Vuppala, "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm," Applied Computing and Informatics, vol. 15, no. 1, pp. 59–66, Jan. 2019, doi: 10.1016/j.aci.2017.10.003.
[8]. J. Wang and I. Ch. Paschalidis, "Botnet Detection Based on Anomaly and Community Detection," IEEE Transactions on Control of Network Systems, vol. 4, no. 2, pp. 392–404, Jun. 2017, doi: 10.1109/tcns.2016.2532804.
[9]. C. Venkatesan, D. Balamurugan, T. Thamaraimanalan, and M. Ramkumar, "Efficient Machine Learning Technique for Tumor Classification Based on Gene Expression Data," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Mar. 2022, doi: 10.1109/icaccs54159.2022.9785294.
[10]. K. Alieyan, A. ALmomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," Neural Computing and Applications, vol. 28, no. 7, pp. 1541–1558, Dec. 2015, doi: 10.1007/s00521-015-2128-0.
[11]. R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers," Applied Sciences, vol. 9, no. 11, p. 2375, Jun. 2019, doi: 10.3390/app9112375.
[12]. A. Ahammadi, W. H. Hassan, and Z. A. Shamsan, "An Overview of Artificial Intelligence for 5G/6G Wireless Networks Security," 2022 International Conference on Cyber Resilience (ICCR), Oct. 2022, doi: 10.1109/iccr56254.2022.10024692.
[13]. M. Debashi and P. Vickers, "Sonification of Network Traffic for Detecting and Learning About Botnet Behavior," IEEE Access, vol. 6, pp. 33826–33839, 2018, doi: 10.1109/access.2018.2847349.
[14]. V. Cherappa, T. Thangarajan, S. S. Meenakshi Sundaram, F. Hajjej, A. K. Munusamy, and R. Shanmugam, "Energy-Efficient Clustering and Routing Using ASFO and a Cross-Layer-Based Expedient Routing Protocol for Wireless Sensor Networks," Sensors, vol. 23, no. 5, p. 2788, Mar. 2023, doi: 10.3390/s23052788.
[15]. R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers," Applied Sciences, vol. 9, no. 11, p. 2375, Jun. 2019, doi: 10.3390/app9112375.
[16]. X. Hoang and Q. Nguyen, "Botnet Detection Based On Machine Learning Techniques Using DNS Query Data," Future Internet, vol. 10, no. 5, p. 43, May 2018, doi: 10.3390/fi10050043.
[17]. S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Computers & Security, vol. 45, pp. 100–123, Sep. 2014, doi: 10.1016/j.cose.2014.05.011.
[18]. M. Asadi, M. A. Jabraeil Jamali, S. Parsa, and V. Majidnezhad, "Detecting botnet by using particle swarm optimization algorithm based on voting system," Future Generation Computer Systems, vol. 107, pp. 95–111, Jun. 2020, doi: 10.1016/j.future.2020.01.055.
[19]. S. Garg, S. K. Peddoju, and A. K. Sarje, "Scalable P2P bot detection system based on network data stream," Peer-to-Peer Networking and Applications, vol. 9, no. 6, pp. 1209–1225, Feb. 2016, doi: 10.1007/s12083-016-0440-9.
[20]. V. H. Bezerra, V. G. T. da Costa, S. Barbon Junior, R. S. Miani, and B. B. Zarpelão, "IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices," Sensors, vol. 19, no. 14, p. 3188, Jul. 2019, doi: 10.3390/s19143188.
[21]. W. N. H. Ibrahim et al., "Multilayer Framework for Botnet Detection Using Machine Learning Algorithms," IEEE Access, vol. 9, pp. 48753–48768, 2021, doi: 10.1109/access.2021.3060778.