Received: October 17, 2023. Revised: December 20, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Botnet Attack Analysis through Graph Visualization

Muhammad Aidiel Rachman Putra[1] Tohari Ahmad[1]* Dandy Pramana Hostiadi[2]
Royyana Muslim Ijtihadie[1] Pascal Maniriho[3]

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
[2]*Department of Magister Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia*
[3]*Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: Botnet attacks on computer networks require proper handling because they can have dangerous consequences. Botnets are dynamic and able to evolve quickly. A botnet can resemble normal activity, making it challenging to detect. Previous research has introduced botnet detection models but has not focused on analyzing intensity behavior based on incoming and outgoing flows in graph visualization. This analysis is needed to get the botnet attack flow. This paper proposes a detection and comprehensive analysis of botnet attack behavior based on a directed graph. The goal is to detect the attacker and extract the behavior from the directed graph. First, all network traffic is grouped based on the time distance between activities. Visualization is carried out by representing the attacker and target as nodes in every activity group and analyzing the direction of communication in the form of in-degree and out-degree. Meanwhile, interactions are represented in edges and weighted edges based on activity intensity. Then, all graph representation is extracted for classification using random forest, decision tree, support vector classification, Naïve bayes, *k* -nearest neighbors, logistic regression, and XGBoost. In the experiment, three different datasets are used, namely CTU-13, NCC-1, and NCC-2. The proposed approaches perform well, with an average of 99.97% accuracy, 46.82% precision, and 83.33% recall. These results can form a knowledge base of botnet attacks that can be used in attack detection models on the network.

Keywords: Botnet detection, Network infrastructure, Network security, Information security, Graph visualization.

1. Introduction

Attacks and threats on computer networks require serious attention in the technological era. Several malicious software known as malware are often found in cybercrime, including botnet attacks. Botnet attacks are very dangerous because they continue to develop rapidly and require special techniques to detect and anticipate [1]. Botnets are said to be dangerous because they are decentralized [1, 2] and have an attack structure consisting of a bot master and bot client [3]. The client bot will be ideal and carry out attacks based on instructions sent by the master bot via the command and control (C&C) service [4, 5]. Decentralized architecture is more difficult to detect in conventional intrusion detection models because they can form new communication networks when the bot master is isolated [1, 4]. Thus, understanding the characteristics of botnet attacks and knowledge of attack mitigation is needed to build a system attack detection model through in-depth analysis [6]. The results of analyzing the characteristics of botnet attacks can be used to develop signature, anomaly, mining, and case-based detection models.

The botnet attack detection model with the concept of signature-based analysis has been introduced in previous research [7–10]. This detection model has been widely developed because it produces optimal reliability, simplicity, accuracy, and time processing [8–10]. Signature-based models extract botnet attack characteristics such as attack communication pattern behavior [11], features [12], attack sequential patterns [6], number of attack stages[13], and attack time gap [14]. However, there

needs to be more focus on analyzing botnet communication intensity using visualization with a graph analysis approach. Visualization of botnet attacks with communication intensity analysis is needed as a knowledge base for the characteristics of botnet attacks and as a basis for developing accurate and precise detection models.

Contribution. This paper proposes a method for analyzing botnet characteristics based on directed graph visualization, constructed from the visualization of network traffic data, where a node is represented as a vertex. The model analyzes the direction of communication by recognizing the incoming direction as in-degree and the outgoing direction as out-degree at each vertex and applying a weighting method based on intensity into weighted-in-degree and weighted-out-degree. The visualization results in the form of in-degree and out-degree graphs are extracted to be used in the classification phase. Then, the classification results from extracting in-degree and out-degree graph parameters are combined to obtain the final detection decision. The proposed model aims to obtain the characteristics of bot attacks based on visualization based on the directed communication graph. Network administrators can use this analysis method to investigate attacks, build and develop intrusion detection models, or optimize current botnet attack detection models.

The structure of this paper is as follows. Part II discusses related work on botnet detection models with graph analysis and analysis of botnet characteristics. The next section (section 3) presents a detailed explanation of the proposed method. Section 4 contains the experimental results of the proposed method and its analysis. Finally, section 5 concludes the research results and discusses future work.

2. Related works

Botnet detection research has been carried out previously by analyzing network traffic using clustering approaches [15–17], machine learning classification [2, 18], deep learning [12, 19], and signature-based techniques [8–10].

The signature-based approach has been widely developed because it is reliable, can detect quickly, and produces good accuracy. However, the signature-based detection model requires proper analysis of botnet characteristics in a knowledge base based on network traffic analysis for accurate and optimal detection performance against botnet [8–10].

Botnet characteristic analysis. Analysis of network traffic and botnet attacks has been widely carried out in research. Daneshgar and Abbaspour [6] analyzed the distribution of packets transmitted during botnet attacks in P2P networks. The analysis used a statistical approach to find differences between packet distribution in botnet activity and normal activity. Chu et al. [20] developed a machine learning-based detection model while analyzing network characteristics often exploited by botnets in launching attacks. Then, the ratio of botnet connectivity with a statistical approach from the model they created to overcome the massive spread of botnets. In Papadogiannaki and Ioannidis study [21], botnet detection was carried out by in-depth analysis of botnet behavior and malware traces in the form of fingerprints. The analysis is carried out by observing the timeframe and comparing the fingerprint with the legitimate servers. Then, an analysis is carried out on server interactions based on the registered C&C services to compile a malware fingerprint database on the TLS server. To prevent misidentification of P2P servers, this detection model relies on updating the malware fingerprint list as a knowledge base for botnet attacks.

Graph Visualization in Cyber Security. Graph visualization-based analysis techniques are often implemented in botnet attack analysis techniques [15, 22], attack event analysis [23], or event forensics [24]. In [15, 22], simple visualization can be done by identifying the direction of the attacker's communication in graph-directed interaction (GDI), representing the attack as a node and communication as an edge. Rabzelj et al. [23] conducted an attack analysis with data distribution originating from an intrusion detection system known as a honeypot. The extent of node distribution in the visualization of attack activity depends on the honeypot as a detection system that identifies the arrangement of nodes and edges. In event correlation model analysis, graph-based analysis can be used optimally as in [24], reconstructing the drone event sequence into a directed graph. This research uses sentiment analysis to investigate events of interest or suspicious events, represented vertices in the graph. Experiments show that the proposed technique can provide reconstruction and detect suspicious events.

Botnet graph-based detection model. Several researchers use graph analysis as a popular approach to detect botnets. Wang et al. [25] combine flow-based analysis and graph-based analysis to detect botnet nodes. The flow-based analysis combines two components, namely similarity-based and stability-based to verify the network flow data. Meanwhile, graph-based analysis is to detect anomalies in environmental traffic graphs. The purpose of this

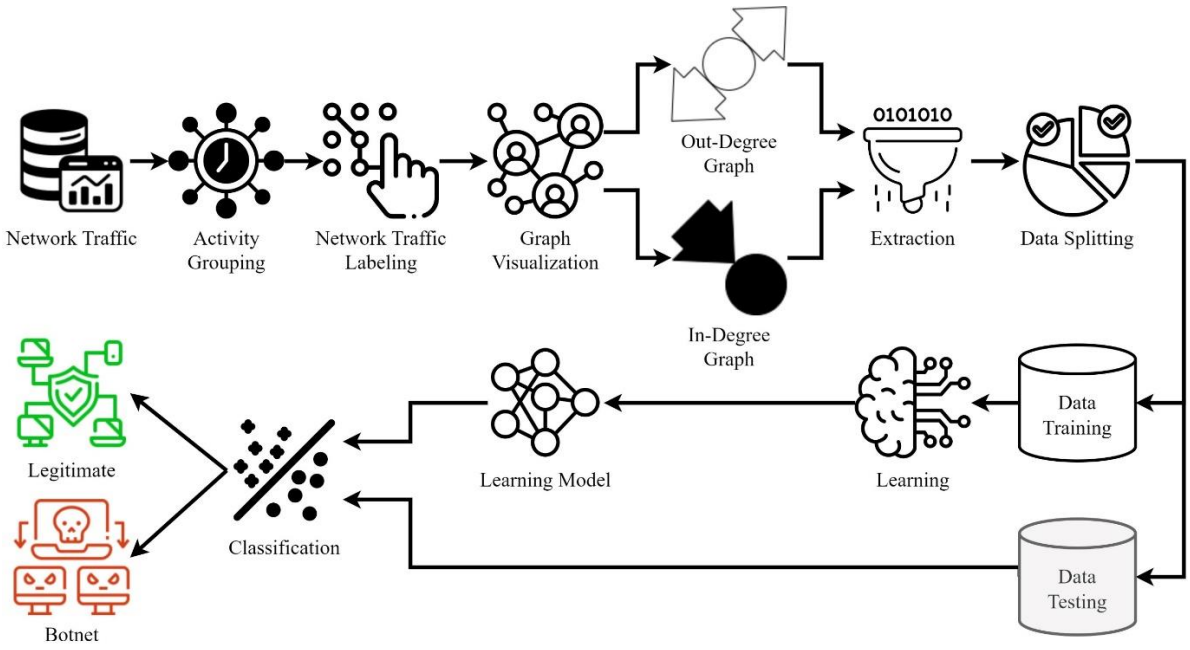

Figure. 1 Proposed method

anomaly analysis is to find C&C servers. The presented model performs well detecting botnets and adapts to complex network environments.

Several previous studies have used graph neural networks (GNN) to detect botnet nodes in graphs. Lo et al. [26] use deep GNN to analyze the characteristics and detect botnets on large-scale networks. Their model uses reversible residual connections and graph isomorphism networks to detect botnets before certain network flows and botnet nodes are highlighted with GNNExplainer. As a result, their model performs well in detecting centralized and decentralized botnets. Besides, the method uses a special graph botnet dataset extracted from CTU-13. In the other research, the GNN method is considered reliable for analyzing attack flow characteristics that cannot be detected by traditional botnet detection with network flow [26, 27]. The detection model with GNN is resistant to changes in the attack flows but it needs improvement to optimize the loss function.

Botnet activity in network traffic is quite difficult to detect because its character is similar to normal activity [25, 28]. Comprehensive analysis is needed to build a knowledge base to detect botnets [6]. The intensity of incoming and outgoing communications analysis from a node on a network is one way to build a knowledge base of botnet attacks. Thus, our research proposes a new technique for analyzing the characteristics of botnet attacks through network traffic graph visualization. This proposed model focuses on analyzing the communication interactions of nodes in networks. In-depth analysis is carried out by calculating the communication intensity, such as the incoming direction as in-degree and the outgoing direction as out-degree at each vertex, and applying a weighting method based on intensity into weighted-in-degree and weighted-out-degree. Network traffic that has been virtualized into a graph is then extracted, resulting in the form of in-degree, out-degree, weighted-in-degree, and weighted-out-degree, which are used in the classification phase to detect the presence of botnets.

3. Proposed method

This research focuses on analyzing the characteristics of botnet attacks based on graph visualization. Each direction of communication is visualized in a graph and weighted by measuring the intensity of bot activity. Then, the generated graph is extracted to get in-degree, out-degree, weighted-in-degree, and weighted-out-degree data used for the classification process. The proposed analysis method is shown in Fig. 1. The terms used to describe the proposed method are listed as follows.

Definition 1. (Network traffic) Each network traffic is a tuple, $nT = (StartTime, Dur, Proto, .., Label)$. In this research, two features in the tuple are used: $SrcAddr$, $DstAddr$, and $Label$.

Definition 2. (Graph) Each network traffic is visualized into a directed graph $G = (V, E)$, where V represents the set of vertices and $E \in V \times V$ represents the set of edges. In this research, $nT(SrcAddr)$ serves as a vertex further noted as v, then $V =$

$(v_1, v_2, \dots, v_n)$, where n is the number of unique $SrcAddr$ in the dataset.

Definition 3. (Edge Weight) Edges in the graph are weighted based on the intensity of communication between v. If the edge is denoted as e and the weight is denoted as w, then the weight of the edge between vertex a (v_a) and vertex b (v_b) can be denoted as $w(e_{ab})$.

3.1 Activity grouping

This phase focuses on grouping network traffic based on $StartTime$ feature. Botnets peform attacks in stages [6, 13], so activity grouping is needed to ensure the attack activity is a series. Previous research grouped activities into static segments of one hour [30, 31] and developed them into segments with sliding time windows [13]. In this research, activity grouping is carried out by analyzing the distance between attacks. Botnets have their characteristics in carrying out attacks, where the time interval for attack activity is shorter than normal. By setting the proper threshold value, the attack and normal activities can be distinguished. The time distance between activities is analyzed to obtain this threshold value.

Next, the threshold value obtained is used as a reference for dividing the graph according to the activity time of each e. If there is a set $E_{SrcAddr\ DstAddr}$ where $E_{SrcAddr\ DstAddr} = \{e_1, e_2, \dots, e_m\}$, then $StartTime$ of activity e_2 minus $StartTime$ of activity e_1 lower than the threshold value (G), then e_1 and e_2 are in the same activity group. On the other hand, if $StartTime$ of activity e_2 minus $StartTime$ of e_1 is greater than G, then e_1 and e_2 are in different activity groups. Equation 1 explains the grouping process based on the time the activity appears.

$$Group = \begin{cases} true;\ if\ e_2^{StartTime} - e_1^{StartTime} < G \\ false;\ if\ e_2^{StartTime} - e_1^{StartTime} > G \end{cases} \quad (1)$$

Where $e_1, e_2 \in E_{SrcAddr\ DstAddr}$ and $e_2^{StartTime} - e_1^{StartTime}$ is a time gap between e_2 activity $StartTime$ and e_1 activity $StartTime$.

3.2 Network traffic labeling

This phase processes the label feature on network traffic ($nT(Label)$) to simplify it into two categories: botnet and normal. The labeling process uses regular expressions (RegEx) by detecting whether $nT(Label)$ has the words "botnet" or not. If $nT(Label) = "flow = From - Normal - V48 - CVUT - WebServer"$, the value will be $nT(Label) = "normal"$ after the network traffic labeling phase because there is no "botnet" word in $nT(Label)$.

3.3 Graph visualization

In this phase, the graph is constructed from each nT by analyzing $SrcAddr$ and $DstAddr$. When processed, an nT will produce a vertex component with a directed connection from $v_{SrcAddr}$ to $v_{DstAddr}$, or the same as $e_{SrcAddr\ DstAddr}$. The proposed method aims to visualize attack activity into a directed graph so the connection between vertex i and vertex j forms an edge that is not the same as the connection between vertex j to i, or can be denoted as $e_{ij} \neq e_{ji}$. Two types of graphs are produced in this phase: in-degree and out-degree graphs.

3.4 Edge weighting

The edges between v are weighted based on $SrcAddr$ and $DstAddr$ to form a new set $nT(SrcAddr, DstAddr)$. So, the weight of the edge between vertex $SrcAddr$ and vertex $DstAddr$ is the total value of members in the set $nT(SrcAddr, DstAddr)$. Besides, if the set $nT(SrcAddr, DstAddr)$ is equal to $E_{SrcAddr\ DstAddr} = e_1, e_2, \dots, e_m$, where m is the number of e from $v_{SrcAddr}$ to $v_{DstAddr}$ in a graph, then the value of $w(e_{ab}) = m$.

3.5 Extraction

Directed graphs formed and grouped based on activity groups have various metadata that can be used for classification. Some metadata that can be extracted are vertex in degree, out degree, weighted in degree, and weighted out-degree. In this phase, the metadata is extracted and arranged to form new tabular data with the features: $StartTime$, $Address$, $OutDegree$, $InDegree$, $WeightedOutDegree$ and $WeightedInDegree$. All five features are carried over to the following process, namely classification, to detect addresses that are botnets and those that are not.

3.6 Data splitting

The extracted data is divided into two groups to be used in two phases: training and testing. Data splitting is done by calculating the amount of data in the two classes, botnet and normal. Then, the data are randomly divided into the two classes at a proportion of 80%:20%. Finally, the data sets from both classes are combined again according to the exact

proportions: botnet 80% combined with normal 80% as training data and botnet 20% combined with normal 20% as testing data.

3.7 Classification

The classification process utilizes the training and testing data. Thus, classification is done using seven machine learning algorithms: random forest (RF), decision tree (DT), support vector classification (SVC), naïve bayes (NB), *k*-nearest neighbors (*k*-NN), logistic regression (LR) and XG boost (XGB). The previous series of processes produced four data types: in-degree training, out-degree training, in-degree testing, and out-degree testing. The four data types are in tabular form with four features:$Address$, $Degree$, $WeightedDegree$, $Label$. The training phase includes $\{Degree, WeightedDegree, Label\}$ with $Label$ as the target feature. Meanwhile, two features are used for prediction in the testing phase: $\{Degree, WeightedDegree\}$, and then $Label$ used as an evaluation feature.

In the botnet dataset, some addresses only have either out-degrees or in-degrees. This condition can happen because those addresses are only network traffic listeners or broadcasters. Thus, the two data types (in-degree and out-degree) produce different detection results. In the evaluation phase, the prediction results on the two data types are combined based on $Address$. If an $Address$ is detected as a botnet even once from both in-degree and out-degree data, then the final evaluation will determine the $Address$ as a botnet. However, if neither data detects an $Address$ as a botnet, the $Address$ is defined as a legitimate. Finally, the detection results are compared with actual data to get the performance of the detection model.

4. Result and discussion

This research supports the analysis process performed in a common environment, such as a personal computer with Intel Core i7-9700F 3.00GHz and 16 GB RAM. For the experiment, we implemented the method in Python 3.10 and several libraries for visualization, such as the NetworkX python library and pandas for data processing and analytics.

4.1 Dataset

Testing was carried out using three different botnet datasets, namely CTU-13 [32], NCC-1 [33], and NCC-2 [34], with a bidirectional network flow (binetflow) format, which has a large number of network traffic records. CTU-13 is a botnet dataset built at CTU University, Czech Republic, by simulating a botnet-type malware attack on campus, which used several protocols and performed different actions. This malware attacks sporadically and simultaneously includes normal and background activities. This simulation is recorded in ".pcap" form and extracted into a bidirectional network traffic flow (binetflow) file. The 13 botnet attack recording results have been identified based on Argus and are referred to as the CTU-13 dataset. The description of the CTU-13 dataset is shown in Table 1.

The NCC-1 dataset was built at the network centric computing laboratory of institut teknologi sepuluh nopember, indonesia, in 2021 by extracting botnet activity on the CTU-13 dataset and generated into a group botnet attack dataset [13] through modeling in [33]. Group botnet activity has periodic and intense characteristics. There are 13 scenarios with different botnet types in each scenario, and the attack duration in each scenario is 8 hours. A description of the NCC-1 dataset is shown in Table 2. Then, the NCC-1 dataset was extended into the NCC-2 dataset [34] in 2022. It has simultaneous and distributed attack characteristics, whose description shown in Table 3. There are three sub-datasets based on the detection sensor type built and distributed in a computer network.

In this research, three datasets with different characteristics: sporadic (CTU-13), periodic (NCC-1), and simultaneous (NCC-2), are used for the analysis process. In the CTU-13 and NCC-1 datasets, scenarios 9, 10, 11, and 12 are used. Meanwhile, in the NCC-2 dataset, three sub-datasets are used. The reason for using four scenarios in the CTU-13 and NCC-1 datasets and the three sub-datasets in NCC-2 is that there is more than 1 type of bot in each scenario dataset.

4.2 Time gap analysis on activity grouping

Time gap analysis is carried out by looking at the distribution of time distance data between each botnet activity. Eight sub-datasets from CTU-13 were used as testing data [32]. Fig. 2 presents the data distribution of the time interval between botnet attack activities, visualized in a box plot. The data distribution shows that each sub-dataset that records different botnet activities has a different maximum distance between activities. Scenario 11, which is a recording of DDoS botnet activity, has the smallest distance between attack activities, namely 0 seconds. This distance indicates that DDoS botnet attack activities were carried out simultaneously. Other DDoS attack activities in scenario 10 also have a

Table 1. CTU-13 Dataset details

Scen.	Bots	Botnet Name	SPAM	CF	PS	DDoS	IRC	HTTP	P2P
1	1	Neris	✓	✓	-	-	✓	-	-
2	1	Neris	✓	✓	-	-	✓	-	-
3	1	Rbot	-	-	✓	-	✓	-	-
4	1	Rbot	-	-	-	✓	✓	-	-
5	1	Virut	✓	-	✓	-	-	✓	-
6	1	Menti	-	-	✓	-	-	✓	-
7	1	Sogou	-	-	-	-	-	✓	-
8	1	Murlo	-	-	✓	-	-	-	-
9	10	Neris	✓	✓	✓	-	✓	-	-
10	10	Rbot	-	-	-	✓	✓	-	-
11	3	Rbot	-	-	-	✓	✓	-	-
12	3	NSIS.ay	-	-	-	-	-	-	✓
13	1	Virut	✓	-	✓	-	-	✓	-

Scen.: Dataset Scenario; CF: Click Fraud; PS: Port Scanning; DDoS: Distributed Denial of Services; IRC: Internet Relay Chat; HTTP: Hypertext Transfer Protocol; P2P: Peer-to-peer.

Table 2. NCC-1 Dataset details

Scen.	Bots	Botnet Name	SPAM	CF	PS	DDoS	IRC	HTTP	P2P
1	1	Neris	✓	✓	-	-	✓	-	-
2	1	Neris	✓	✓	-	-	✓	-	-
3	1	Rbot	-	-	✓	-	✓	-	-
4	1	Rbot	-	-	-	✓	✓	-	-
5	1	Virut	✓	-	✓	-	-	✓	-
6	1	Menti	-	-	✓	-	-	✓	-
7	1	Sogou	-	-	-	-	-	✓	-
8	1	Murlo	-	-	✓	-	-	-	-
9	10	Neris	✓	✓	✓	-	✓	-	-
10	10	Rbot	-	-	-	✓	✓	-	-
11	3	Rbot	-	-	-	✓	✓	-	-
12	3	NSIS.ay	-	-	-	-	-	-	✓
13	1	Virut	✓	-	✓	-	-	✓	-

Scen.: Dataset Scenario; CF: Click Fraud; PS: Port Scanning; DDoS: Distributed Denial of Services; IRC: Internet Relay Chat; HTTP: Hypertext Transfer Protocol; P2P: Peer-to-peer

Table 3. NCC-2 Dataset Details

Scen.	Bots	Botnet Name	SPAM	CF	PS	DDoS	IRC	HTTP	P2P
1	10	Rbot, Neris, Sogo, NSIS.ay, Virut	✓	✓	✓	✓	✓	✓	✓
2	10	Rbot, Neris, Menti, Virut	✓	✓	✓	✓	✓	✓	-
3	10	Rbot, Neris, Murlo, NSIS.ay, Virut	✓	✓	✓	✓	✓	✓	✓

Scen.: Dataset Scenario; CF: Click Fraud; PS: Port Scanning; DDoS: Distributed Denial of Services; IRC: Internet Relay Chat; HTTP: Hypertext Transfer Protocol; P2P: Peer-to-peer.

relatively low maximum distance between attack activities (3 seconds). Meanwhile, the highest time interval for attack activity comes from scenario 12, which records P2P botnet attack activity. The maximum time interval of 13 seconds in scenario 12 was chosen as the threshold value G, meaning the distance between botnet attack activities in a series should be no more than 13 seconds.

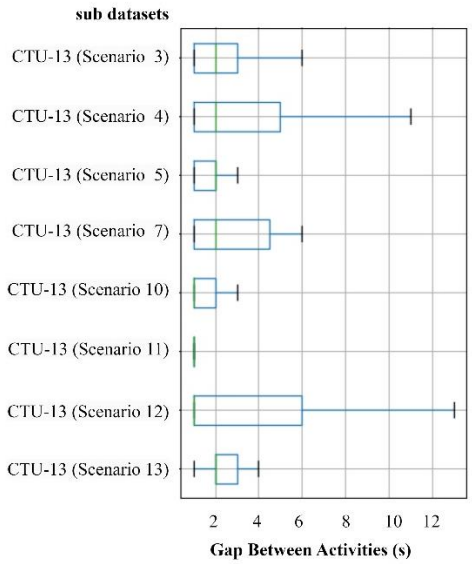

Figure. 2 Distribution of the time interval between botnet attack activities

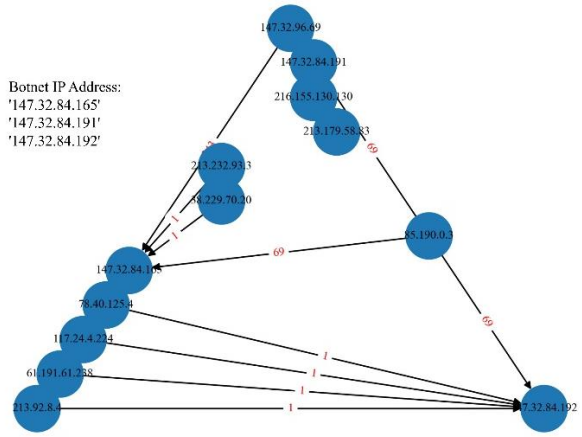

Figure. 3 Directed in-degree metric for botnet activities in 0.26 hours of CTU-13 (Scenario 11)

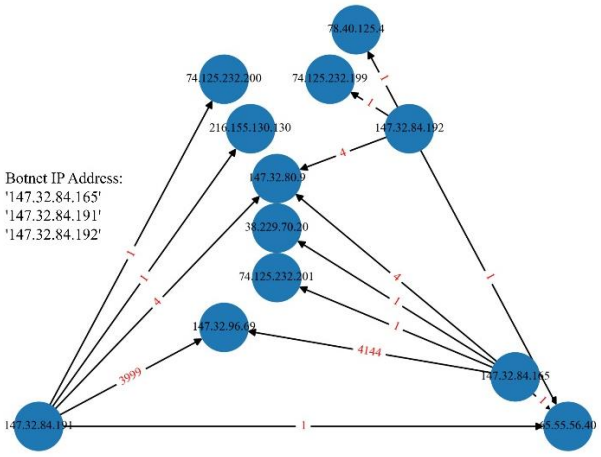

Figure. 4 Directed out-degree metric for botnet activities in 0.26 hours of CTU-13 (Scenario 11)

4.3 Botnet graph analysis

The collection of botnet activity in network traffic is visualized into a directed graph using the $DiGraph()$ class in NetworkX. A vertex in a directed graph is an IP device from network traffic. Besides, the communication links between devices are represented as edges. Fig. 3 is one of the results of the following graph visualization, with the weight of each edge being the weighted in-degree values. Besides, Fig. 4 is a visualization with the weighting of edges based on out-degree. Four graph parameters can be analyzed in this research: in-degree, out-degree, weighted-in-degree, and weighted-out-degree. The distribution of in-degree, out-degree, weighted-in-degree, and weighted-out-degree values for each device in the sub-dataset is converted into a boxplot shown in Fig. 5. The mapping results on the CTU-13 dataset are shown in Fig. 6. Analysis of the NCC-1 dataset is shown in Fig. 7, and the mapping of the NCC-2 dataset is shown in Fig. 8.

4.3.1. Attack type

Botnets have types of activity when carrying out attacks on targets. In this research, the three datasets, namely CTU-13, are shown in Table 1, the NCC-1 dataset in Table 2, and the NCC-2 dataset in Table 3. There are four types of attacks, namely: SPAM, Click Fraud (CF), Port Scanning (PS) and Distributed Denial of Services (DDoS). In scenario 9 of the CTU-13 dataset, scenario 9 of NCC-1, and all sensors in NCC-2, there is a series of interrelated activities, namely SPAM, PS, and CF activities. A series of activity groups shows that the bot attack was carried out by sending messages sporadically to the target. This attack characteristic causes the formation of a tree graph visualization. Each bot and target are visualized as a connected vertex communicating through edges. From the results of the visualization analysis, the series of SPAM, CF, and PS activities produce high weighted-out-degree and out-degree values, especially in scenario 9 in the CTU-13 and NCC-1 datasets.

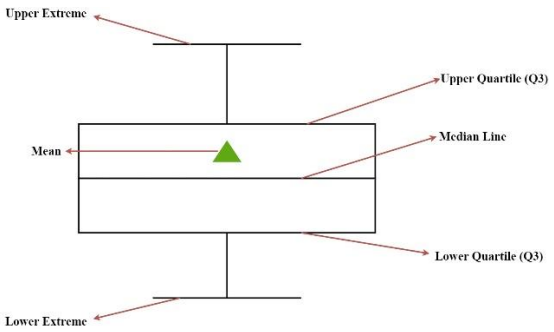

Figure. 5 Illustration of box plot

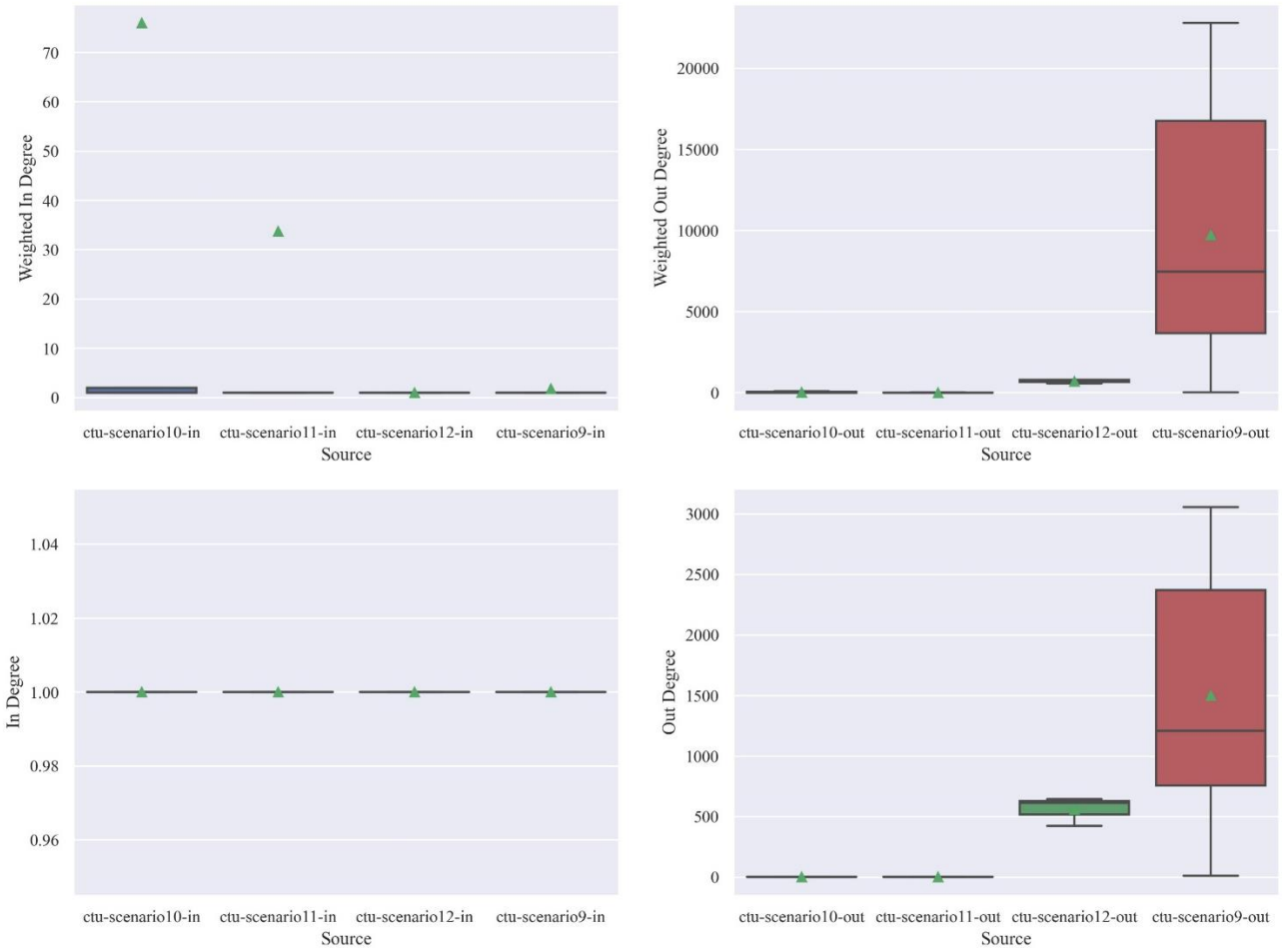

Figure. 6 Botnet attack graph edge distribution on CTU-13 Dataset

The type of DDoS attack is flooding data packets to clients, which is carried out in a distributed manner [25, 35–37]. In a botnet attack, DDoS is launched to flood the target with many high-intensity requests, forcing the target to serve each request until a deadlock occurs. In Figs. 6, 7, and 8, the analysis results show that the DDoS attack type has a high distribution of in-degree and weighted-in-degree values in scenarios 10 and 11 on the CTU-13 and NCC-1 datasets. Two things can cause the high value of the weighted-in-degree average. The first is the number of responses from the target, the same as the number of requests sent by the botnet because DDoS attack forced the target give response for every request. Second, the bot's function causes it as a C&C medium in the botnet communication network. Thus, a deeper analysis is needed based on transmitted data packages to distinguish high in-degree and weighted-in-degree due to responses from targets or responses from bots to C&C.

4.3.2. P2P Botnet

Peer-to-peer (P2P) is a type of decentralized botnet communication. This communication model is found in scenario 12 of CTU-13, scenario 12 of NCC-1, sensor 1, and sensor 3 of NCC-2. The P2P botnet has the same upper extreme, lower extreme, median, and mean values, as shown in Figs. 6 and 7. These equal upper extreme, lower extreme, median, and mean values are caused by the character of the P2P network, which applies collaborative and distributed processing. Every device connected to a P2P network will work together to carry out distributed processing, making bot activities increasingly difficult to distinguish from collaborative P2P processing activity. Thus, a deeper analysis to differentiate between attack and normal activity on P2P networks is needed to optimize the detection model, which can be focused on in further research.

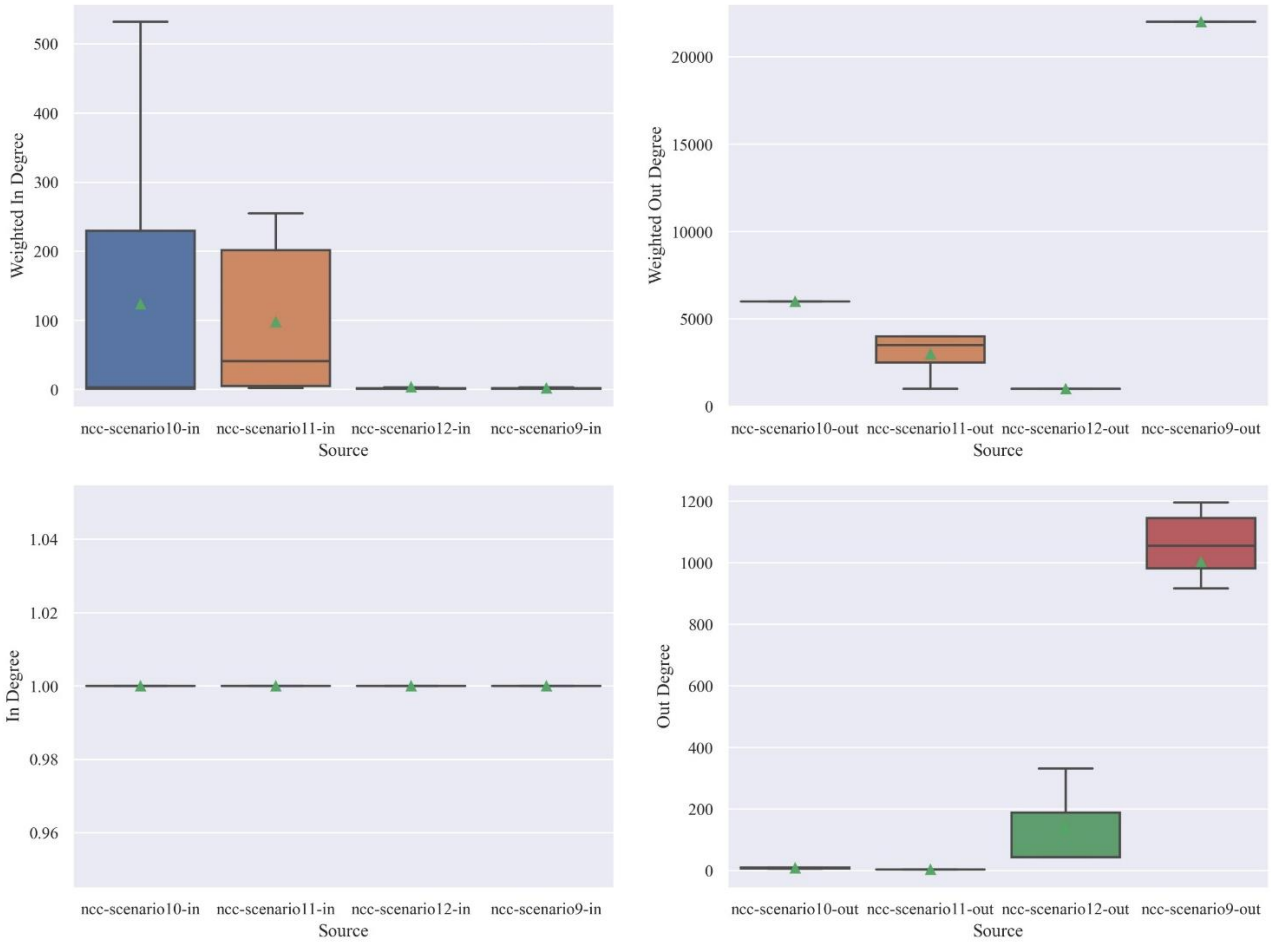

Figure. 7 Botnet attack graph edge distribution on NCC-1 dataset

4.3.3. Botnet attack characteristics

The characteristics of botnet attacks are sporadic, periodic-intense, and simultaneous [38–40]. The results of the visualization analysis in Figs. 6, 7, and 8 show that simultaneous botnet attacks have a much higher average out-degree and weighted-out-degree compared to sporadic and periodic characteristics. Thus, simultaneous botnets also have relatively high-intensity attacks, for example, bot 147.32.84.165 on sensor 2. It has a weighted-out-degree value of 104,001, much higher than other bots. Differences in the weighted-out-degree values in a bot can cause the calculated average value to be higher than the upper extreme value. Thus, the average weighted-out-degree value on sensor 2 is 36,401 higher than the highest weighted-out-degree 104,001.

Besides, periodic botnets tend to have consistent attack intensity in each scenario, or each bot's attack intensity is distributed equally. For example, in scenario 9, which executes a SPAM attack with ten bots, there is a constant weighted-out-degree value of 22,000 or 22,002. The well-distributed attack intensity produces a straight boxplot graph because the data's average, highest value, lowest value, and high value are the same. Weighted-in-degree parameter analysis shows that sporadic botnets with DDoS attacks have the highest value. Meanwhile, the in-degree value in every characteristic is equal, indicating that a botnet attack cycle does not always receive a response from the target.

4.3.4. Number of attackers

Based on the analysis results, DDoS attacks that cause flooding on targets have a higher intensity if carried out by ten bots than by three bots. These activities are shown in Fig. 6, which compares two scenarios that have DDoS attacks with different numbers of attacking bots, namely scenarios 10 and 11. In sporadic botnets, the weighted-out-degree value of 1,120 in a DDoS attack with ten bots in scenario 10 is higher than the DDoS attack with three bots in scenario 11 with a weighted-out-degree value of 384 in one attack cycle. Similarities can be seen in periodic botnets on NCC-1; scenario 11 with three bots has weighted-out-degree value of 1,002, while scenario 10 with 10 attacking bots has an average out-degree value of 6,000 in one attack cycle.

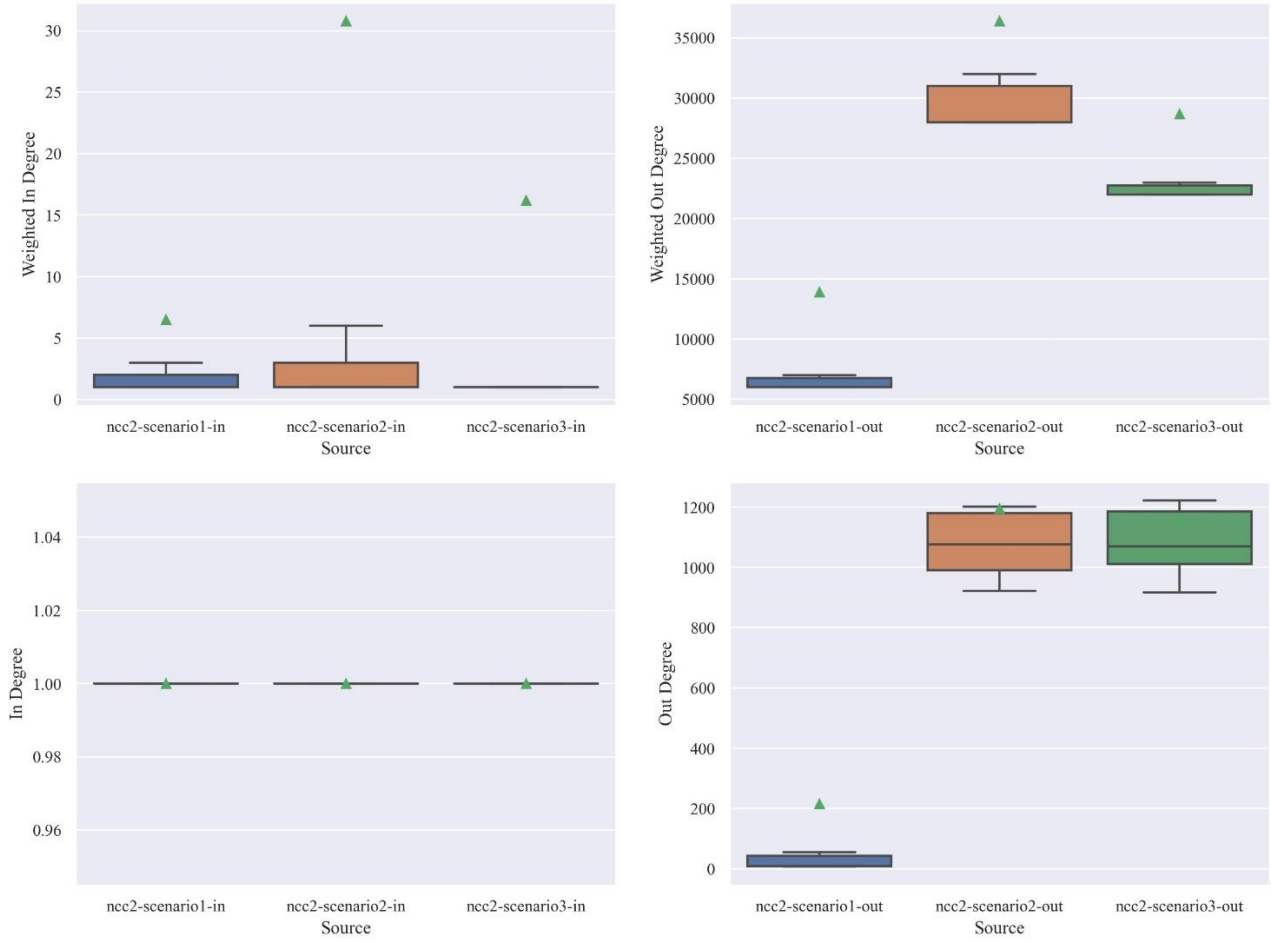

Figure. 8 Botnet attack graph edge distribution on NCC-2 dataset

Meanwhile, the received communication intensity by the botnet (in-degree / weighted-in-degree) with ten attackers is also consistently higher than with three attackers for both sporadic and periodic botnets. In the botnet dataset, simultaneous DDoS attacks present in all sub-datasets and cannot be appropriately analyzed using graph visualization alone.

4.4 Classification result

This research does not perform hyperparameter tuning; all hyperparameters use the Scikit Learn library predefined values by default [41]. For evaluating the performance, this paper uses three matrixes: accuracy (Acc), precision (Pre), and recall (Rec). Tables 4 and 5 present the detection results on two types of in-degree and out-degree data, respectively. The machine learning classification model works better in detecting botnets based on out-degree data according to the characteristics of botnets (especially SPAM botnets) that target many devices for intense attack. A high number of out-degrees or weighted-out-degrees in a series of activities is easier to recognize than in-degrees and weighted-in-degrees. However, combining the results of in-degree and out-degree detection can improve the performance of the detection model. For example, RF in NCC-2 was previously only able to detect 5 out of 10 botnets on out-degree data; combining detection results with in-degree increases the performance of detecting 7 botnets. Table 6 shows the detection results from each machine-learning method.

The detection model has good performance for detecting botnets with sporadic attack characteristics. This good performance can be seen in Table 6, where XGB, DT, RF, and *k*-NN get a perfect score of 100%. Even so, the model still has a high false alarm value because the average Pre value in testing with CTU-13 is still below 20%. While the model achieved the best average performance when detecting periodic botnets, the Acc, Pre, and Rec values remained stable. The performance in periodic and simultaneous botnet detection is low because the formed graph has not come from a proper attack cycle. Simultaneous and periodic botnets attack targets at the same time, while periodic botnets attack targets intensely [33, 34, 39, 40]. These two characteristics of botnet attacks cause the time distance between attacks tend to be lower

Table 4. Detection result with in-degree data

Dataset	Method	TN	FP	FN	TP	Acc. (%)	Pre. (%)	Rec. (%)
CTU-13	XGB	53060	1	6	4	99.99	80.00	40.00
	DT	53060	1	6	4	99.99	80.00	40.00
	RF	53060	1	6	4	99.99	80.00	40.00
	NB	53061	0	10	0	99.98	-	0.00
	LR	53061	0	10	0	99.98	-	0.00
	k-NN	52944	1	8	2	99.98	66.67	20.00
	SVC	52945	0	10	0	99.98	-	0.00
NCC-1	XGB	28484	0	10	0	99.96	-	0.00
	DT	28479	5	10	0	99.95	0.00	0.00
	RF	28479	5	10	0	99.95	0.00	0.00
	NB	28484	0	10	0	99.96	-	0.00
	LR	28484	0	10	0	99.96	-	0.00
	k-NN	28507	0	10	0	99.96	-	0.00
	SVC	28507	0	10	0	99.96	-	0.00
NCC-2	XGB	123722	1	5	5	100.00	83.33	50.00
	DT	123721	2	5	5	99.99	71.43	50.00
	RF	123721	2	5	5	99.99	71.43	50.00
	NB	123723	0	10	0	99.99	-	0.00
	LR	123723	0	10	0	99.99	-	0.00
	k-NN	123861	3	7	3	99.99	50.00	30.00
	SVC	123864	0	10	0	99.99	-	0.00

Table 5. Detection result with out-degree data

Dataset	Method	TN	FP	FN	TP	Acc. (%)	Pre. (%)	Rec. (%)
CTU-13	XGB	107841	103	0	10	99.90	8.85	100.00
	DT	107843	101	0	10	99.91	9.01	100.00
	RF	107844	100	0	10	99.91	9.09	100.00
	NB	107912	32	6	4	99.96	11.11	40.00
	LR	107942	2	10	0	99.99	0.00	0.00
	k-NN	107520	44	0	10	99.96	18.52	100.00
	SVC	107564	0	10	0	99.99	-	0.00
NCC-1	XGB	60331	6	1	8	99.99	57.14	88.89
	DT	60337	0	1	8	100.00	100.00	88.89
	RF	60337	0	2	7	100.00	100.00	77.78
	NB	60320	17	1	8	99.97	32.00	88.89
	LR	60337	0	9	0	99.99	-	0.00
	k-NN	28507	0	10	0	99.96	-	0.00
	SVC	28507	0	10	0	99.96	-	0.00
NCC-2	XGB	380246	22	3	3	99.99	12.00	50.00
	DT	380267	1	1	5	100.00	83.33	83.33
	RF	380266	2	1	5	100.00	71.43	83.33
	NB	380228	40	2	4	99.99	9.09	66.67
	LR	380268	0	6	0	100.00	-	0.00
	k-NN	379714	3	1	4	100.00	57.14	80.00
	SVC	123864	0	10	0	99.99	-	0.00

than sporadic botnet activity. A comprehensive analysis of the time gap is needed in future work by considering attack activity that occurs intensely or simultaneously. Thus, a deeper analysis is needed in future works to decide the appropriate time gap value to form activity groups.

Meanwhile, DT obtained the highest average performance from Acc, Pre, and Rec in three different

Table 6. Final detection result combining in-degree and out-degree data

Dataset	Method	TN	FP	FN	TP	Acc. (%)	Pre. (%)	Rec. (%)
CTU-13	XGB	153927	104	0	10	99.93	8.77	100.00
	DT	153929	102	0	10	99.93	8.93	100.00
	RF	153930	101	0	10	99.93	9.01	100.00
	NB	153999	32	6	4	99.98	11.11	40.00
	LR	154029	2	10	0	99.99	0.00	0.00
	k-NN	153551	45	0	10	99.97	18.18	100.00
	SVC	153596	0	10	0	99.99	-	0.00
NCC-1	XGB	86379	6	2	8	99.99	57.14	80.00
	DT	86380	5	2	8	99.99	61.54	80.00
	RF	86380	5	3	7	99.99	58.33	70.00
	NB	86368	17	2	8	99.98	32.00	80.00
	LR	86385	0	10	0	99.99	-	0.00
	k-NN	28507	0	10	0	99.96	-	0.00
	SVC	28507	0	10	0	99.96	-	0.00
NCC-2	XGB	483676	23	4	6	99.99	20.69	60.00
	DT	483696	3	3	7	99.9988	70.00	70.00
	RF	483695	4	3	7	100.00	63.64	70.00
	NB	483659	40	6	4	99.99	9.09	40.00
	LR	483699	0	10	0	100.00	-	0.00
	k-NN	483340	6	5	5	100.00	45.45	50.00
	SVC	123864	0	10	0	99.99	-	0.00

Table 7. Comparative analysis with previous research

Method		CTU-13			NCC-1			NCC-2			Data Format
		Acc (%)	Pre (%)	Rec (%)	Acc (%)	Pre (%)	Rec (%)	Acc (%)	Pre (%)	Rec (%)	
Dollah et al. [42]	DT	92.20	99.93	84.47	99.63	99.85	99.41	99.98	99.70		Network flow based
	k-NN	75.16	73.18	51.52	96.67	99.15	94.18	99.85	98.31	98.91	
	NB	69.34	62.28	99.45	65.38	66.77	82.82	89.36	33.20	95.17	
	RF	73.83	49.99	47.67	51.16	49.55	2.34	99.99	99.86	99.96	
Hostiadi and Ahmad [43]		99.18	42.29	91.55	99.73	75.14	99.29	60.09	0.36	97.73	Network flow based
Proposed	XGB	99.93	8.77	100.00	99.99	57.14	80.00	99.99	20.69	60.00	Graph based
	DT	99.93	8.93	100.00	99.99	61.54	80.00	100.00	70.00	70.00	
	RF	99.93	9.01	100.00	99.99	58.33	70.00	100.00	63.64	70.00	
	NB	99.98	11.11	40.00	99.98	32.00	80.00	99.99	9.09	40.00	
	LR	99.99	0.00	0.00	99.99	-	0.00	100.00	-	0.00	
	k-NN	99.97	18.18	100.00	99.96	-	0.00	100.00	45.45	50.00	
	SVC	99.99	-	0.00	99.96	-	0.00	99.99	-	0.00	

datasets. The best average performance of DT is 99.97% of Acc, 46.82% of Pre and 83.33% of Rec. DT can get high performance because tree-based algorithms can generate a deep tree model that covers all conditions from training data. Therefore, other tree-based algorithms, such as RF and XGB, also perform above-average.

4.5 Comparative analysis

This paper compares the proposed model's performance with several previous studies [42, 43]. Table 7 provides the performance comparison of each botnet detection model in three metrics: Acc, Pre, and Rec. It is shown that the proposed model receives the

highest Acc score among previous research in testing with the entire dataset. The highest Acc shows that the proposed method can adapt to large network traffic. However, it is still not good enough compared to previous research on Pre and Rec. Furthermore, it is generally quite good in detecting botnet nodes of directed graph data visualized from network traffic. However, it needs further improvement to reduce the possibility of false alarms to increase the low Pre value. Previous research detected network traffic, whereas the proposed method detects nodes or addresses. Meanwhile, the proposed method can adapt to changes in the characteristics of botnet network traffic because it uses a graph-based approach instead of network traffic-based.

5. Conclusion

This research presents a directed graph visualization of botnet attack activity to analyze the characteristics of botnet attacks with four parameters: in-degree, out-degree, weighted-in-degree, and weighted-out-degree. The weighted-in-degree and weighted-out-degree parameters result from weighting in-degree and out-degree by combining aspects of attack intensity. After in-degree and out-degree graphs are generated, four graph parameters are extracted and used for classification using several machine learning algorithms. Classification results from in-degree and out-degree are then combined to get the final detection results.

Experiments using three public datasets show that the proposed method can reliably detect nodes suspected of being botnets with 99.97% of accuracy, 46.82% of precision, and 83.33% of recall on average. Network administrators can use visualization to investigate attacks on the network by botnets, which are represented as vertices. Meanwhile, the analysis results show that each botnet has a character based on the type of attack, botnet name, attack characteristics, and number of attacking bots. The unique characteristics of botnets can be used as a knowledge base to build more sophisticated botnet detection models based on graph analysis.

Future research will analyze aspects of the packets transmitted by each vertice to use as a new weighting method. Analysis of transmitted packets can be useful for distinguishing bots tasked with attacking targets or those acting as C&C. In addition, weighting based on transmitted packets may be a new approach in anomaly-based botnet detection models. Additionally, future research should analyze attacks based on the number of attackers. All these approaches need to be carried out to improve the performance of the detection model, especially in precision and recall, with efficiency in processing complexity.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, MARP, TA, DPH, RMI, and PM; methodology, MARP, TA, DPH, RMI, and PM; software, MARP, DPH; validation, MARP and DPH; formal analysis, MARP, TA, DPH, RMI, and PM; investigation, MARP, TA, DPH, RMI, and PM; resources, MARP and DPH; data curation, MARP; writing—original draft preparation, MARP; writing—review and editing, TA and DPH; visualization, MARP; supervision, TA and RMI; project administration, TA and RMI; funding acquisition, TA.

Acknowledgments

This work has been supported by the Institut Teknologi Sepuluh Nopember (ITS) and PMDSU Scholarship from the Ministry of Education, Culture, Research and Technology, The Republic of Indonesia.

References

[1] 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”, *Appl. Sci.*, Vol. 9, No. 11, 2019, doi: 10.3390/app9112375.

[2] M. Alshamkhany, W. Alshamkhany, M. Mansour, M. Khan, S. Dhou, and F. Aloul, “Botnet Attack Detection using Machine Learning”, In: *Proc. of the 2020 14th International Conference on Innovations in Information Technology, IIT 2020*, Nov. 2020, pp. 203–208, doi: 10.1109/IIT50501.2020.9299061.

[3] R. Abrantes, P. Mestre, and A. Cunha, “Exploring Dataset Manipulation via Machine Learning for Botnet Traffic”, *Procedia Computer Science*, Vol. 196, pp. 133–141, 2022, doi: https://doi.org/10.1016/j.procs.2021.11.082.

[4] A. M. Manasrah, T. Khdour, and R. Freehat, “DGA-based botnets detection using DNS traffic mining”, *J. King Saud Univ. - Comput. Inf. Sci.*, 2022, doi: https://doi.org/10.1016/j.jksuci.2022.03.001.

[5] H. Suryotrisongko and Y. Musashi, “Hybrid Quantum Deep Learning and Variational Quantum Classifier-Based Model for Botnet DGA Attack Detection”, *Int. J. Intell. Eng. Syst.*,

Vol. 15, No. 3, pp. 215–224, 2022, doi: 10.22266/ijies2022.0630.18.

[6] F. F. Daneshgar and M. Abbaspour, "A two-phase sequential pattern mining framework to detect stealthy P2P botnets", *J. Inf. Secur. Appl.*, 2020, doi: 10.1016/J.JISA.2020.102645.

[7] H. R. Zeidanloo, A. B. Manaf, P. Vahdani, F. Tabatabaei, and M. Zamani, "Botnet detection based on traffic monitoring", In: *Proc. of International Conference on Networking and Information Technology*, pp. 97–101, 2010, doi: 10.1109/ICNIT.2010.5508552.

[8] T. Oh, S. Jadhav, and Y. H. Kim, "Android botnet categorization and family detection based on behavioural and signature data", In: *Proc. of 2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 647–652, 2015, doi: 10.1109/ICTC.2015.7354630.

[9] F. E. Ayo, J. B. Awotunde, S. O. Folorunso, M. O. Adigun, and S. A. Ajagbe, "A genomic rule-based KNN model for fast flux botnet detection", *Egypt. Informatics J.*, Vol. 24, No. 2, pp. 313–325, 2023.

[10] A. J. Alzahrani and A. A. Ghorbani, "Real-time signature-based detection approach for SMS botnet", In: *Proc. of 2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pp. 157–164, 2015, doi: 10.1109/PST.2015.7232968.

[11] M. A. R. Putra, T. Ahmad, and D. P. Hostiadi, "Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks", *Int. J. Intell. Eng. Syst.*, Vol. 15, No. 4, 2022, doi: 10.22266/ijies2022.0831.48.

[12] C. Joshi, R. K. Ranjan, and V. Bharti, "A Fuzzy Logic based feature engineering approach for Botnet detection using ANN", *J. King Saud Univ. - Comput. Inf. Sci.*, 2021, doi: 10.1016/j.jksuci.2021.06.018.

[13] D. P. Hostiadi, T. Ahmad, and W. Wibisono, "A New Approach to Detecting Bot Attack Activity Scenario", *Adv. Intell. Syst. Comput.*, Vol. 1383 AISC, pp. 823–835, 2021, doi: 10.1007/978-3-030-73689-7_78.

[14] M. A. R. Putra, T. Ahmad, and D. P. Hostiadi, "Botnet Dataset Overview Using Statistical Approach Based on Time Gap Activity Analysis", In: *Proc. of 2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, 2023, doi: 10.1109/ISDFS58141.2023.10131832.

[15] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, and L. Bian, "Botnet detection using graph-based feature clustering", *J. Big Data*, Vol. 4, p. 14, 2017, doi: 10.1186/s40537-017-0074-7.

[16] R. Khodadadi and B. Akbari, "Ichnaea: Effective P2P botnet detection approach based on analysis of network flows", In: *Proc. of International Symposium on Telecommunications (IST)*, pp. 934–940, 2014, doi: 10.1109/ISTEL.2014.7000837.

[17] C. Y. Wang, C. L. Ou, Y. E. Zhang, F. M. Cho, P. H. Chen, J. B. Chang, and C. K. Shieh, "BotCluster: A session-based P2P botnet clustering system on NetFlow", *Comput. Networks*, Vol. 145, pp. 175–189, Oct. 2018, doi: 10.1016/j.comnet.2018.08.014.

[18] T. A. Tuan, H. V. Long, and D. Taniar, "On Detecting and Classifying DGA Botnets and their Families", *Comput. Secur.*, Vol. 113, p. 102549, 2022, doi: https://doi.org/10.1016/j.cose.2021.102549.

[19] S. Homayoun, M. Ahmadzadeh, S. Hashemi, A. Dehghantanha, and R. Khayami, "BoTShark: A deep learning approach for botnet traffic detection", *Adv. Inf. Secur.*, Vol. 70, pp. 137–153, 2018, doi: 10.1007/978-3-319-73951-9_7.

[20] Z. Chu, Y. Han, and K. Zhao, "Botnet Vulnerability Intelligence Clustering Classification Mining and Countermeasure Algorithm Based on Machine Learning", *IEEE Access*, Vol. 7, pp. 182309–182319, 2019, doi: 10.1109/ACCESS.2019.2960398.

[21] E. Papadogiannaki and S. Ioannidis, "Pump Up the JARM: Studying the Evolution of Botnets Using Active TLS Fingerprinting", In: *Proc. of 2023 IEEE Symposium on Computers and Communications (ISCC)*, pp. 764–770, 2023, doi: 10.1109/ISCC58397.2023.10218210.

[22] D. P. Hostiadi, W. Wibisono, and T. Ahmad, "B-Corr Model for Bot Group Activity Detection Based on Network Flows Traffic Analysis", *KSII Trans. Internet Inf. Syst.*, Vol. 14, No. 10, pp. 4176–4197, 2020, doi: 10.3837/tiis.2020.10.014.

[23] M. Rabzelj, C. Bohak, L. Š. Južnič, A. Kos, and U. Sedlar, "Cyberattack Graph Modeling for Visual Analytics", *IEEE Access*, Vol. 11, pp. 86910–86944, 2023, doi: 10.1109/ACCESS.2023.3304640.

[24] H. Studiawan, T. Ahmad, A. M. Shiddiqi, B. J. Santoso, and B. A. Pratomo, "Forensic Event Reconstruction for Drones", In: *Proc. of 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 41–45, 2021, doi: 10.1109/ISRITI54043.2021.9702864.

[25] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu,

"BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors", *Inf. Sci. (Ny).*, Vol. 511, pp. 284–296, 2020, doi: 10.1016/j.ins.2019.09.024.

[26] X. Zhu, Y. Zhang, Z. Zhang, D. Guo, Q. Li, and Z. Li, "Interpretability Evaluation of Botnet Detection Model based on Graph Neural Network", In: *Proc. of IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2022, doi: 10.1109/INFOCOMWKSHPS54753.2022.9798287.

[27] F. Alizadeh and M. Khansari, "An Analysis of Botnet Detection Using Graph Neural Network", In: *Proc. of 2023 13th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 491–495, 2023, doi: 10.1109/ICCKE60553.2023.10326235.

[28] M. Eslahi, W. Z. Abidin, and M. V. Naseri, "Correlation-based HTTP Botnet detection using network communication histogram analysis", In: *Proc. of IEEE Conf. Appl. Inf. Netw. Secur.*, Vol. 2018-Janua, pp. 7–12, 2017, doi: 10.1109/AINS.2017.8270416.

[29] X. D. Hoang and Q. C. Nguyen, "Botnet detection based on machine learning techniques using DNS query data", *Futur. Internet*, Vol. 10, No. 5, 2018, doi: 10.3390/FI10050043.

[30] H. Choi, H. Lee, and H. Kim, "BotGAD: Detecting botnets by capturing group activities in network traffic", In: *Proc. of 4th International ICST Conference on Communication System Software and Middleware*, pp. 1–8, 2009, doi: https://doi.org/10.1145/1621890.1621893.

[31] M. A. R. Putra, U. L. Yuhana, T. Ahmad, and D. P. Hostiadi, "Analyzing The Effect of Network Traffic Segmentation on The Accuracy of Botnet Activity Detection", In: *Proc. of 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, pp. 1–6, 2022, doi: 10.1109/CENIM56801.2022.10037365.

[32] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods", *Comput. Secur.*, Vol. 45, pp. 100–123, 2014, doi: https://doi.org/10.1016/j.cose.2014.05.011.

[33] D. P. Hostiadi and T. Ahmad, "Dataset for Botnet group activity with adaptive generator", *Data Br.*, Vol. 38, 2021, doi: https://doi.org/10.17632/4vftxh97m8.1.

[34] M. A. R. Putra, D. P. Hostiadi, and T. Ahmad, "Botnet dataset with simultaneous attack activity", *Data Br.*, Vol. 45, p. 108628, Dec. 2022, doi: 10.1016/J.DIB.2022.108628.

[35] M. G. Karthik and M. B. M. Krishnan, "Securing an Internet of Things from Distributed Denial of Service and Mirai Botnet Attacks Using a Novel Hybrid Detection and Mitigation Mechanism", *Int. J. Intell. Eng. Syst.*, Vol. 14, pp. 113–123, 2021, doi: 10.22266/ijies2021.0228.12.

[36] K. M. Prasad, A. R. M. Reddy, and K. V. Rao, "BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 32, No. 1, pp. 73–87, Jan. 2020, doi: 10.1016/J.JKSUCI.2017.07.004.

[37] H. E. Sofany, "A New Cybersecurity Approach for Protecting Cloud Services against DDoS Attacks", *Int. J. Intell. Eng. Syst.*, Vol. 14, pp. 205–215, 2020, doi: 10.22266/ijies2020.0430.20.

[38] M. A. R. Putra, D. P. Hostiadi, and T. Ahmad, "Simultaneous Botnet Dataset Generator: A simulation tool for generating a botnet dataset with simultaneous attack characteristic", *Softw. Impacts*, Vol. 14, p. 100441, Dec. 2022, doi: 10.1016/J.SIMPA.2022.100441.

[39] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic", In: *Proc. of 7th IEEE International Conference on Computer and Information Technology (CIT)*, pp. 715–720, Apr. 2008, doi: 10.1109/cit.2007.90.

[40] J. Kwon, J. Kim, J. Lee, H. Lee, and A. Perrig, "PsyBoG: Power spectral density analysis for detecting botnet groups", In: *Proc. of the 9th IEEE International Conference on Malicious and Unwanted Software (MALCON)*, pp. 85–92, 2014, doi: 10.1109/MALWARE.2014.6999414.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python", *J. Mach. Learn. Res.*, Vol. 12, pp. 2825–2830, 2011.

[42] R. F. M. Dollah, F. M. A., F. Arif, M. Z. Mas'ud, and L. K. Xin, "Machine Learning for HTTP Botnet Detection Using Classifier Algorithms", *J. Telecommun. Electron. Comput. Eng.*, Vol. 10, Nos. 1-7 SE-Articles, pp. 27–30, Feb. 2018.

[43] D. P. Hostiadi and T. Ahmad, "Hybrid model for bot group activity detection using similarity and correlation approaches based on network traffic flows analysis", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 34, No. 7, pp. 4219–4232, 2022, doi: 10.1016/J.JKSUCI.2022.05.004.