

RESEARCH

Open Access



B-CAT: a model for detecting botnet attacks using deep attack behavior analysis on network traffic flows

Muhammad Aidiel Rachman Putra¹, Tohari Ahmad^{1*} and Dandy Pramana Hostiadi²

*Correspondence:
tohari@its.ac.id

¹ Department of Informatics,
Institut Teknologi Sepuluh
Nopember (ITS), Kampus ITS
Keputih Sukolilo, Surabaya 60111,
Indonesia

² Department of Magister
Information Systems, Institut
Teknologi Dan Bisnis STIKOM Bali,
Bali, Indonesia

Abstract

Threats on computer networks have been increasing rapidly, and irresponsible parties are always trying to exploit vulnerabilities in the network to do various dangerous things. One way to exploit vulnerabilities in a computer network is by employing malware. Botnets are a type of malware that infects and attacks targets in groups. Botnets develop quickly; the characteristics of initially sporadic attacks have grown into periodic and simultaneous. This rapid development has proved that the botnet is advanced and requires more attention and proper handling. Many studies have introduced detection models for botnet attack activity on computer networks. Apart from detecting the presence of botnet attacks, those studies have attempted to explore the characteristics of botnets, such as attack intensity, relationships between activities, and time segment analysis. However, there has been no research that explicitly detects those characteristics. On the other hand, each botnet characteristic requires different handling, while recognizing the characteristics of the botnet can help network administrators make appropriate decisions. Based on these reasons, this research builds a detection model that can recognize botnet characteristics using sequential traffic mining and similarity analysis. The proposed method consists of two main processes. The first is training to build a knowledge base, and the second is testing to detect botnet activity and attack characteristics. It involves dynamic thresholds to improve the model sensitivity in recognizing attack characteristics through similarity analysis. The novelty includes developing and combining analytical techniques of sequential traffic mining, similarity analysis, and dynamic threshold to detect and recognize the characteristics of botnet attacks explicitly on actual behavior in network traffic. Extensive experiments have been conducted for the evaluation using three different datasets whose results show better performance than others.

Keywords: Botnet, Bot detection, Information security, Intrusion detection system, Network infrastructure, Network security, Botnet attack characteristics

Introduction

Cybercrime cases have accelerated in recent years [1], involving malware attacks and various illegal software applications. This attack can be a botnet that runs malware performing malicious activities such as phishing, Distributed Denial of Services (DDoS),

and spamming [2–6]. A botnet consists of bots, a collection of machines infected with malware that communicate and connect to carry out attacks controlled by the botmaster [7–10]. In practice, the attacks start from infecting devices to form a bot network before reaching the target machine [11]. This infection process runs underground [12, 13], sometimes leading to detection difficulty. The infected bots are connected via the Command and Control (C&C) service that the botmaster [9, 14] uses by sending commands, exchanging information, and updating the botnet control system code, affecting big data [15, 16].

Botnet architecture evolves from centralized to decentralized [17–19]. The centralized botnets connect and communicate directly with botmasters via C&C, such as IRC or HTTP [4, 20], and are typically anticipated by identifying botmaster activity on C&C servers [14, 21]. When botmaster communication access is blocked, all bots in the network cannot attack the target. On the other hand, in the decentralized structure, each bot client can become a bot master dynamically by utilizing the peer-to-peer (P2P) model [22, 23], making it more difficult to detect [17, 23].

Botnets have unique attack characteristics, different from other attack types. Based on their attack period and intensity, botnets can be categorized into sporadic, periodic, and simultaneous [24–26]. Sporadic and periodic botnets use a botnet type, like Neris, Rbot, or NSIS.ay, to attack the target. Sporadic and periodic botnets differ in their attack period [26, 27], where the former has systematic attacks repeated every certain period [25–27]. It can be seen from the peak of its attack activity spreading over time. Meanwhile, sporadic botnets tend to be more random in determining the attack time [21, 28], concentrating on a particular time without a regular pattern. Their attack tends to peak at a specific time in a scenario [21, 27]. Unlike those two attacking groups, the simultaneous botnets have many botnet types performing attacks parallelly and consistently high starting from their inception [27].

Previous studies have introduced botnet detection models with several approaches, such as mining-based [10, 15], behavior analysis-based [29], signature-based [13, 30], deep learning [6, 7, 20, 31], graph-based [10, 18, 32–34], and anomaly-based [21, 35]. In addition, optimization is implemented in the pre-processing stage to improve detection accuracy, such as segmentation time-based activity analysis [16, 36] and optimization model on feature selection [37, 38]. As a result, the model can more accurately recognize bot (attack) and normal (non-attack) activities.

Some research can analyze activity scenario patterns [39], detect P2P bots [23, 40, 41], analyze botnet group activity correlation [32], and analyze communication patterns within botnets [17]. However, they still focus on detecting particular attacks, where practically, many attacks combine sporadic, periodic, and simultaneous attacks. Therefore, an appropriate and accurate model to detect the attack activity based on their characteristics is essential.

This research proposes Botnet Characteristic Attack Type (B-CAT), a new model to detect botnet activities and their attack characteristics, whether periodic, sporadic, or simultaneous categories, using several analytical approaches based on knowledge base extraction. It aims to detect and recognize botnet attack characteristics from network traffic data. The sequentially linked activities are analyzed from the mining network traffic process, including the source and destination parameters and the

time distance between network traffic. This research takes three botnet datasets with different characteristics extracted from the knowledge base. Each incoming network traffic passing the sequential traffic mining process is analyzed based on its similarity with existing patterns in the knowledge base.

The detection results provide more information to network security administrators to appropriately handle attacks based on their characteristics to anticipate botnet attacks. Thus, this research novelties can be summarized as follows:

- Detecting and recognizing bot attack characteristics by combining sequential traffic mining, similarity analysis, and dynamic threshold analysis.
- Developing a knowledge base extraction mechanism of botnet activity to determine the characteristics of an actual botnet based on the sporadic, simultaneous, and periodic activity category on large network traffic data types.

This study is structured as follows: “[Introduction](#)” section explains the botnet background and basic concepts; “[Related works](#)” section describes related works; the details of the proposed model are in “[Proposed method](#)” section; the evaluation results and discussion are provided in “[Result and discussion](#)” section; finally, “[Conclusions](#)” section concludes the research and depicts possible future works.

Related works

Botnet activity detection models have been developed [7, 16, 18, 23, 42], considering their attack characteristics, which are sporadic [21, 24], periodic [16, 25], and simultaneous [21, 27]. The experiment in those previous studies used a corresponding public dataset for periodic [25], simultaneous [27], and sporadic [28] attacks, which can be either single [7, 42] or group bot attacks [16, 24, 32]. A bot group activity [32], which is an activity performed by some bots having a similar characteristic pattern [18], has a causal relationship with each other [16, 17, 39] and forms a patterned attack stage [17, 39].

Improving the detection model performance involves optimizing the recognition of the attack characteristics, such as optimizing the pre-processing in the feature selection [17, 30, 38] and feature extraction [18, 23]. Some approaches have been implemented, such as machine learning [10, 42, 43] and deep learning [7, 20, 30]. However, none of these detection models focus on recognizing the characteristics of botnet attacks.

On the other hand, a dataset that can be used as a benchmark for an attack is needed to recognize the characteristics of botnet. Several studies have simulated botnet attacks to be used as a basis for research to identify botnet characters [28, 44]. Garcia et al. [28] simulated a botnet attack by executing seven types of malwares: neris, rbot, virut, menti, sogou, murlo, and NSIS.ay. Randomly executed attacks produce records of different attack activities such as SPAM, Distributed Denial of Service (DDoS), Click Fraud, and others. Thus, Alomari et al. [44] simulated an HTTP botnet attack that carried out a DDoS attack. Flooding attacks with the HTTP-GET method are executed by two botnet types: Black Energy v.1.8 Bot and Zemra bot. The simulation, which was compiled using 40 VMWare zombies, 1 Command and Control (C&C) server, and a target, succeeded in recording botnet attacks in real-time.

Chowdhury et al. [18] introduce a detection model with a clustering approach based on each bot activity, represented in seven features: in-degree, out-degree, in-degree weight, out-degree weight, clustering coefficient, the node between centrality, and eigenvector centrality. Overall, these features construct clusters using Self Organizing Map (SOM). The filtering technique for inactive nodes is to improve detection performance. Their experiment shows that the method can detect botnets optimally. Nevertheless, the details of false positive values are not provided, and it can only detect sporadic attacks.

Dollah et al. [42] use machine learning and classification to build the detection system. The bot attack characteristics are extracted from HTTP traffic and DNS queries involving five types of botnets: Dorkbot, Zeus, Citadel, SpyEye, and Cutwail, known as malicious network traffic. In the training process, data traffic is marked by labeling 1 for malicious and 0 for non-malicious traffic. The data are cleaned from meaningless noise by ignoring the IP and port number in the pre-processing stage. The detection process uses four types of machine learning algorithms, namely: Decision Tree, k -Nearest Neighbor (k -NN), Naïve Bayes (NB), and Random Forest (RF). The experimental results show that the k -NN algorithm produces the best botnet detection. However, this study does not explicitly detect the attack type. It only informs the presence of malicious bot activity in the network traffic.

A hybrid detection technique, which combines several approaches, has been introduced [7, 16, 23] to optimize the detection results. Khan et al. [23] focus on detecting the P2P botnet through a multi-layer concept with machine learning as the basis. Each layer in this detection model has specific objectives, such as reducing data traffic, which refers to the issue of device resource limitations and detection computational efficiency. At the second layer, network traffic is filtered and classified into two traffic classes: P2P and non-P2P. Then, each traffic class is analyzed based on data packets, data streams, and session Layers. After obtaining two classes, the process continues in the third layer: feature extraction and feature reduction. This layer focuses on obtaining and using the best features in the detection model, which takes a Decision Tree algorithm as a classification model. It shows relatively good detection accuracy. However, more information about other botnet attack types is needed, as the sporadic botnet attack does not only occur in P2P communication.

Implementing the fuzzy logic method with hierarchical Artificial Neural Networks (ANN) is also considered [7] by generating features in traffic data and analyzing them using feature participation. The experiment shows that the model can detect the fourth hidden layer in the CTU-13 dataset. Similar to the previous research, it can only recognize sporadic attacks. Indeed, botnet attacks can be periodic, repeated, and simultaneous in computer networks.

Hostiadi and Ahmad [16] investigate the bot group activity, adopting correlation analysis to get the causality of each bot attack based on the activity time. Sliding window-based time segmentation analysis is used to obtain the transition of attack activity in the analyzed period. In the segmentation stage, the system searches for frequent host activities in network traffic and extracts them to obtain periodic and intense patterns. It measures the activity similarity in each time segment using the Mean Absolute Error (MAE) and the causality analysis using the Pearson correlation. The research can detect the causality based on the time of attack occurrence and the analysis of forward and

backward correlation. Furthermore, the system can detect the characteristics of sporadic and periodic botnet attacks. However, similar to other studies, it has not been able to detect coinciding or simultaneous botnet attacks.

Botnet detection using the signature-based or rule-based approach has relatively good speed accuracy. Alzahrani and Ghorbani [45] proposed a signature-based detection to detect Short Message Service (SMS) botnet activity. SMS is taken as a medium to carry out infections, attacks, or share other malicious applications. The proposed method uses pattern matching and rule-based techniques to label incoming or outgoing SMS suspicious or normal. The evaluation results show that the proposed model performs well for recognizing malicious SMS. The rule-based botnet detection model was also introduced by Alieyan et al. [46], which detects Domain Name System (DNS) queries and responses that are considered abnormal. There are three stages in the proposed model: the first stage is for network traffic filtration and data cleansing; the second stage focuses on feature analysis; and the third stage is implementing DNS rule-based abnormality detection. The evaluation results show that the proposed model can detect DNS botnets better than the others. Ayo et al. [47] also introduce an approach to improve the performance of Fast Flux Botnet detection (FFB). The problem of detection speed and large feature dimensions are investigated using the Genetic Algorithm (GA) and k -Nearest Neighbor (k -NN) approaches. The proposed model comprises four main modules: extractor, filter, resolver, and detector. This approach builds decision tree rules and K -dimensional (KD) trees from the k -NN algorithm to classify DNS. The evaluation results show that the proposed method can improve the performance of FFB.

Some previous studies have considerable detection performance, but they do not recognize the characteristics of each attack, which is crucial to overcoming the attack. Bot attacks can occur in real-time on computer networks simultaneously with various characteristics and resemble specific attack patterns. Therefore, characteristic recognition is required to determine the proper response to botnet attacks on computer networks.

Proposed method

This paper proposes a detection model to detect botnet activity through similarity analysis of sequential activity. The detection process consists of several processes, from traffic splitting to building a knowledge base, sequential activity mining, repetitive sequential activity analysis, sequential activity transformation, similarity analysis, and activity measurement. The entire detection process and flowchart of the proposed method are presented in Figs. 1 and 2, respectively.

Traffic splitting

The traffic is divided into the botnet and normal by analyzing the *Label* feature in the dataset, which is to be adopted as a knowledge base for botnet attack activities. The splitting process is done with a regular expression (Regex), a notation used to define patterns of words to be searched. In the traffic splitting phase, Regex finds network traffic with a label feature with the keyword 'botnet'. At first, the value in the *Label* feature is converted to lowercase. Suppose network traffic has a lowercase *Label* value matching the predefined pattern; the traffic will be grouped into the botnet attack

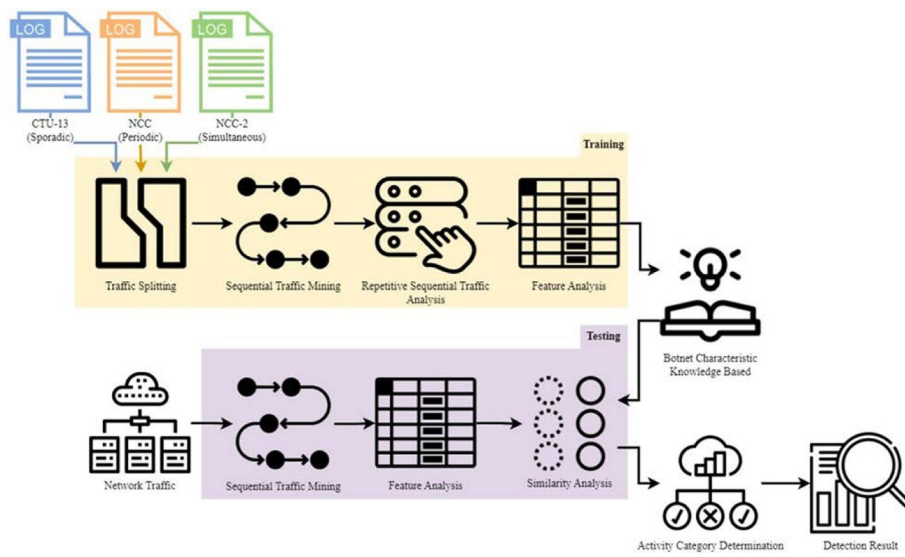


Fig. 1 Proposed method

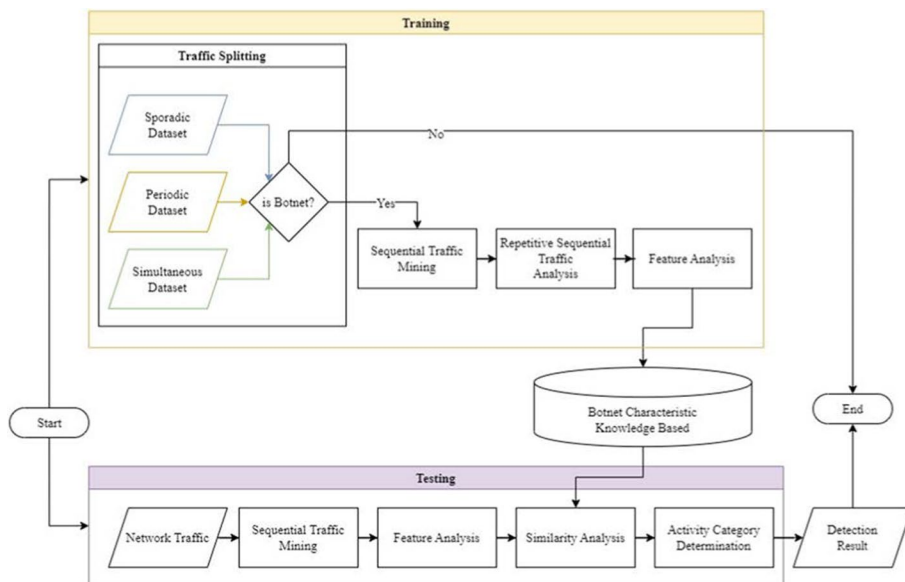


Fig. 2 Flowchart of the proposed method

category. On the other hand, data that does not have a lowercase *Label* value according to a predefined pattern is grouped in the normal activity category. The activity group with the botnet attack category is carried forward to the next stage. This traffic-splitting process is illustrated in Fig. 3.

Sequential traffic mining

Botnet data separated at the previous traffic splitting stage has had network header information called features. Each network traffic record (*Trf*) has fifteen

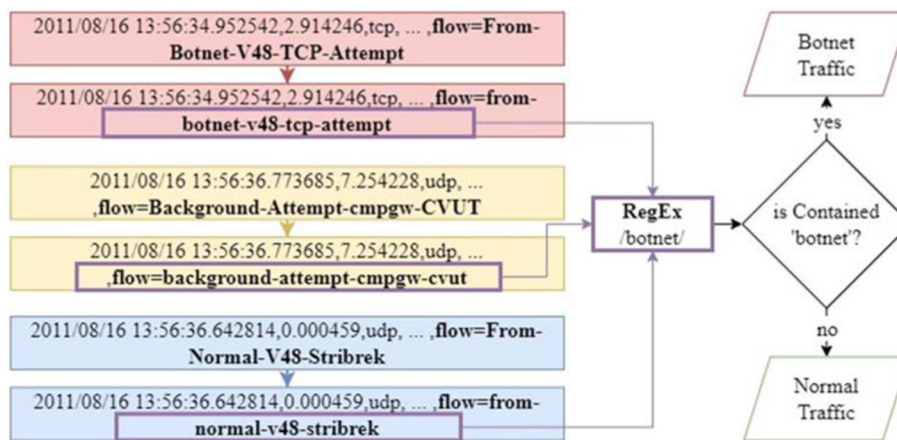


Fig. 3 Traffic splitting

features: *StartTime*, *Dur*, *Proto*, *SrcAddr*, *Sport*, *Dir*, *DstAddr*, *Dport*, *State*, *sTos*, *dTos*, *TotPkts*, *TotBytes*, *SrcBytes*, and *Label*. Thus, traffic (*Trf*) is denoted as $Trf = \{StartTime, Dur, Proto, SrcAddr, \dots, Label\}$. Sequentially, each traffic in the activity period can relate to the activity in the next period [39]. Incoming network traffic records will first go through *SrcAddr* and *DstAddr* analysis to form a pairwise. The pairwise of *SrcAddr* feature in $Trf(Trf^{SrcAddr})$ and *DstAddr* feature in $Trf(Trf^{DstAddr})$ is called the communication ID, denoted as *c*. In other words, if the set of pairwise in a dataset is *C*, then $C = \{c_1, c_2, \dots, c_m\}$ where $c \in C$. The value of *m* in *C* is the number of elements in the set of communication Id between *SrcAddr* dan *DstAddr* as the pairwise in botnet data. Furthermore, each *Trf* having the same pairwise is grouped into a group of activities sequentially according to the occurrence time of the activity represented by the value in the *StartTime* feature. This group of activities in the order of occurrence time is called Sequential Traffic (*SeqA*), so it becomes $SeqA = [Trf_1, Trf_2, \dots, Trf_n]$, where *n* is the number of *Trf* in a *SeqA*. In addition to having the same *c*, *SeqA* must meet the requirement that the time interval between the occurrence of *Trf*₁ and the occurrence of *Trf*_{*n*} must be less than or equal to 60 min which is called the segment period (ϑ). The determination of value ϑ is based on the time required in one botnet attack cycle, which is 60 min [48]. Sequential activity in each segment period can have a different or similar sequence pattern. If *Trf* has the same pairwise *c* but the time interval between *Trf*₁ in *SeqA* is greater than ϑ , then a new *SeqA* will be formed with other *Trf* groups that follow. The sequential activity analysis process is carried out until the traffic record in the last segment period obtains a group of sequential pairs of activities. Analysis of sequential traffic mining is shown in Algorithm 1.

Algorithm 1 Sequential Activity Mining

INPUT: Trf ; $Trf \leftarrow \{StartTime, SrcAddr, DstAddr, \dots, Label\}$
OUTPUT: $SeqA$
 /* c : communication id between source and destination from Trf */
 /* ∂ : constant of the maximum duration of sequential activity; 3600 seconds */
 $l \leftarrow$ number of Trf
 $SeqA \leftarrow [Trf_1, Trf_2, \dots, Trf_n]$; $n \leftarrow$ number of sequential activity
 $c_1 \leftarrow \{Trf_1^{SrcAddr}, Trf_1^{DstAddr}\}$
 $C \leftarrow \{c_1, c_2, \dots, c_m\}$; $m \leftarrow$ number of elements in the set of communication Id

Step 1: Communication id analysis
for $index \leftarrow 1$ to l **do**
 if $c_{index} \notin C$ **do**
 $C \leftarrow C \cup c_{index}$
 else do
 go to step 2
 end if

Step 2: Time interval analysis between traffic
if $SeqA(c_{index})$ exists **do**
 if $(Trf_{index}^{StartTime} - SeqA(c_{index})[Trf_n^{StartTime}]) > \partial$ **do**
 $C \leftarrow C \cup c_{index}$
 go to step 3
 else do
 go to step 3
 end if
else do
 go to step 3
end if

Step 3: Assign Trf to $SeqA$
 $SeqA(c_{index}) \leftarrow SeqA(c_{index}) \cup Trf_{index}$
end for

Algorithm 2 Repetitive Sequential Activity Analysis

INPUT: $[SeqA_1, SeqA_2, SeqA_3, \dots, SeqA_o]$
OUTPUT: kb
 /* kb : a set of sequential activities without repetition */
 /* o : number of sequential activities */
 $kb \leftarrow \{SeqA_1, SeqA_2, \dots, SeqA_o\}$

for $index \leftarrow 1$ to o **do**
 if $SeqA_{index} \notin kb$ **do**
 $kb \leftarrow kb \cup SeqA_{index}$
 else do
 $kb \leftarrow kb$
 end if
end for

Repetitive sequential traffic Analysis

This stage analyzes the *Label* feature. First, a trace for the same pattern is carried out by taking the *Label* feature from *Trf*, which is in *SeqA*. For example, if there is an $SeqA = [Trf_g, Trf_h, Trf_i]$ where $g < h < i$; g, h, i is an index of *Trf* in *SeqA*, $Trf_g^{Label} = \text{"flow=From-Botnet-V48-UDP-DNS"}$, $Trf_h^{Label} =$

"flow=From-Botnet-V48-TCP-WEB-Established", and $Trf_h^{Label} =$ "flow=From-Botnet-V48-TCP-Attempt", then a $SeqA$ is declared as ["flow=From-Botnet-V48-UDP-DNS", "flow=From-Botnet-V48-TCP-WEB-Established", "flow=From-Botnet-V48-TCP-Attempt"]. The proposed system traces the similarity of the $SeqA$ values contained in kb . If there is no similar pattern, $SeqA$ is stored in kb as a knowledge base; otherwise, the $SeqA$ is included in Repetitive Sequential Traffic, which is ignored. Algorithm 2 presents a repetitive sequential traffic analysis process flow to form a kb containing a set of $SeqA$ and is denoted as $kb = \{SeqA_1, SeqA_2, \dots, SeqA_o\}$ where o is the number element $SeqA$ in the kb set.

Feature analysis

At this stage, two features of Trf in $SeqA$ are analyzed: $StartTime$ and $Label$. The analysis begins by extracting the $StartTime$ feature to generate a new feature called $Diff$, obtained using Eq. (1) that if Trf_i with $i = 0$, then $Diff$ is 0. The $Label$ feature extraction is performed to get a description of the activity of a Trf in $SeqA$. For example, if a $Label$ has the information "flow=From-Botnet-V48-TCP-WEB-Established", then there are sections that are deleted, namely: "flow=From", "V48", and "Botnet". The "flow=From" part is removed because it is a meaningless series of words. The extraction results change the Label to "TCP-WEB-Established" and store it in a new feature called $NetworkActivity$. This feature extraction process is illustrated in Fig. 4.

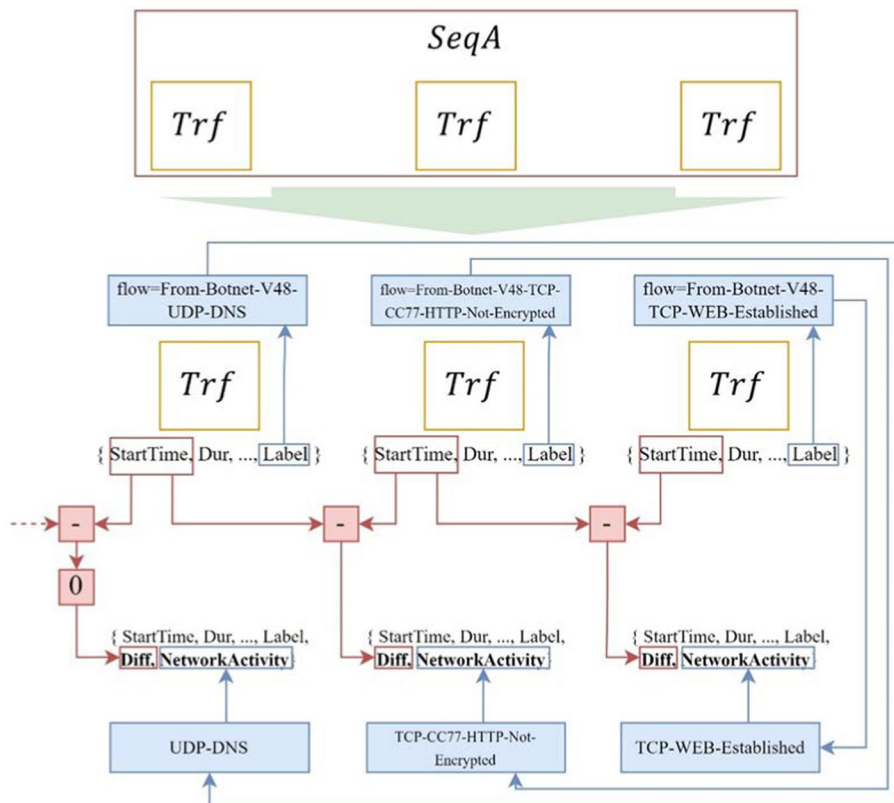


Fig. 4 Feature analysis (extraction)

$$Trf_i^{Diff} = \begin{cases} 0; & \text{if } i = 0 \\ Trf_j^{StartTime} - Trf_i^{StartTime}; & \text{if } i \neq 0 \end{cases} \quad (1)$$

where $j = i + 1$; Trf_i and Trf_j in the same $SeqA$

Similarity analysis

The botnet activity detection process identifies botnet attack characteristics by measuring the similarity between the new traffic data, known as traffic testing data, and data patterns stored in kb . It is to find how close $SeqA$ is to kb with the categories of the periodic knowledge base (kb^{per}), sporadic knowledge base (kb^{spo}), and simultaneous knowledge base (kb^{sim}).

The system scores the *NetworkActivity* similarity to get the bot's suspected activity. A score of 1 is given if *NetworkActivity* in *Trf* testing has the same information as *NetworkActivity* in one of the types of kb , calculated with Eq. (2). The similarity of $SeqA$ is generated based on the average scores obtained; this process is called pattern similarity analysis (*Psim*) and can be expressed in Eq. (3), where $x = 0, 1, 2, \dots, n$; n is the number of *Trf* in $SeqA$ or the number of $kbTrf$ in $kbSeqA$ at a single characteristic.

The next step is measuring the time difference similarity (*Dsim*) to identify the attack characteristics by comparing the *Diff* feature of *Trf* in the testing data with that in kb . *Dsim* is determined by adopting the MAPE error rate calculation, denoted with Eq. (4), and the similarity analysis value is calculated using Eq. (5), taking the previous *Psim* and *Dsim*. Each $SeqA$ similarity analysis result is collected to get the highest value of $SeqA$ with each kb characteristic. This process is repeated until all $SeqA$ have obtained the highest similarity to each kb having different characteristics. Algorithm 3 explains how to obtain the highest similarity to each characteristic ($perSc_{max}$, $spoSc_{max}$, $simSc_{max}$).

$$SimNetA(Trf_x^{NetAct}, kbTrf_x^{NetAct}) = \begin{cases} 1; & \text{if } Trf_x^{NetAct} = kbTrf_x^{NetAct} \\ 0; & \text{if } Trf_x^{NetAct} \neq kbTrf_x^{NetAct} \end{cases} \quad (2)$$

$$PSim(SeqA, kbSeqA) = \frac{1}{n} \sum_{x=0}^n SimNetA(Trf_x^{NetAct}, kbTrf_x^{NetAct}) \quad (3)$$

$$DSim(SeqA, \zeta SeqA) = 1 - \left(\frac{1}{n} \sum_{x=0}^n \left| \frac{Trf_x^{Diff} - kbTrf_x^{Diff}}{kbTrf_x^{Diff}} \right| \right) \quad (4)$$

$$Sim(SeqA, \zeta SeqA) = \frac{1}{2} (PSim(SeqA, kbSeqA) + DSim(SeqA, kbSeqA)) \quad (5)$$

The previous measurement inferred the similarity level between bot activity and each kb characteristic that a heuristic approach is implemented to find a threshold value (\bar{T}) determining this level. Each $Max(perSc_{max}, spoSc_{max}, andsimSc_{max})$ is tested, ranging between 0 and 1. That threshold is employed in Eq. (6) to determine whether an activity is either botnet or normal.

$$Act = \begin{cases} \text{Botnet} ; & \text{if } \text{Max}(\text{perSc}_{max}, \text{spoSc}_{max}, \text{simSc}_{max}) \geq \bar{T} \\ \text{Normal} ; & \text{if } \text{Max}(\text{perSc}_{max}, \text{spoSc}_{max}, \text{simSc}_{max}) < \bar{T} \end{cases} \quad (6)$$

Algorithm 3 Similarity Analysis

INPUT: $SeqA, kb^{per}, kb^{spo}, kb^{sim}$
OUTPUT: $perSc_{max}, spoSc_{max}, simSc_{max}$
 /* kb : knowledge-base, there are three different kb according to the characteristics of the dataset (Periodic, Sporadic, Simultaneous) */
 /* $sizeOf(x)$: number of elements in x */
 /* Sim : calculation of similarity analysis */
 /* Sim_{max} : maximum value from all similarity analysis in every characteristic */
 /* k^{th} sequential activity in kb */
 $kb^{per} \leftarrow \{perSeqA_1, perSeqA_2, \dots, perSeqA_p\}; p \leftarrow \text{number of SeqA in } kb^{per}$
 $kb^{spo} \leftarrow \{spoSeqA_1, spoSeqA_2, \dots, spoSeqA_r\}; r \leftarrow \text{number of SeqA in } kb^{spo}$
 $kb^{sim} \leftarrow \{simSeqA_1, simSeqA_2, \dots, simSeqA_s\}; s \leftarrow \text{number of SeqA in } kb^{sim}$
 $perSc_{max} \leftarrow \text{Max}(perSc_1, perSc_2, \dots, perSc_k); perSc(SeqA, perSeqA_k)_k$
 $spoSc_{max} \leftarrow \text{Max}(spoSc_1, spoSc_2, \dots, spoSc_k); spoSc(SeqA, spoSeqA_k)_k$
 $simSc_{max} \leftarrow \text{Max}(simSc_1, simSc_2, \dots, simSc_k); simSc(SeqA, simSeqA_k)_k$
 $k \leftarrow 1, 2, 3, 4, \dots, m$

Step 1: Find patterns with the same length
for $index \leftarrow 1$ to k **do**
 if $sizeOf(SeqA) = sizeOf(perSeqA_{index})$ **do**
 go to step 2
 else if $sizeOf(SeqA) = sizeOf(spoSeqA_{index})$ **do**
 go to step 3
 else if $sizeOf(SeqA) = sizeOf(simSeqA_{index})$ **do**
 go to step 4
 else do
 do nothing
 end if

Step 2: Get max value of cosine similarity with kb^{per}
 $perSim_{index} \leftarrow Sim(SeqA, perSeqA_{index})$
if $perSim_{index} > perSim_{index-1}$ **do**
 $perSim_{max} \leftarrow perSim_{index}$
else do
 $perSim_{max} \leftarrow perSim_{index-1}$
end if

Step 3: Get max value of cosine similarity with kb^{spo}
 $spoSim_{index} \leftarrow Sim(SeqA, spoSeqA_{index})$
if $spoSim_{index} > spoSim_{index-1}$ **do**
 $spoSim_{max} \leftarrow spoSim_{index}$
else do
 $spoSim_{max} \leftarrow spoSim_{index-1}$
end if

Step 4: Get max value of cosine similarity with kb^{sim}
 $simSim_{index} \leftarrow Sim(SeqA, simSeqA_{index})$
if $simSim_{index} > simSim_{index-1}$ **do**
 $simSim_{max} \leftarrow simSim_{index}$
else do
 $simSim_{max} \leftarrow simSim_{index-1}$
end if
end for

Step 5: Return All Sc_{max}
Return $perSim_{max}, spoSim_{max}, simSim_{max}$

Activity category detection

Network traffic activity detected as a botnet attack can be either sporadic, periodic, or simultaneous during the same attack period. A measurement of the proximity to the $SeqA$ value, which has the highest probability of being specific characteristics of a botnet attack, determines how precise the emerging characteristics are. Thus, the maximum value of the three highest similarities is measured using Eq. (7) to categorize the attack.

$$\begin{aligned}
 & \text{Botnet Attack Characteristic} \\
 & = \begin{cases} \text{Periodic;} & \text{if } \text{Max}(\text{perSim}_{avg}, \text{spoSim}_{avg}, \text{simSim}_{avg}) = \text{perSim}_{avg} \\ \text{Sporadic;} & \text{if } \text{Max}(\text{perSim}_{avg}, \text{spoSim}_{avg}, \text{simSim}_{avg}) = \text{spoSim}_{avg} \\ \text{Simultaneous;} & \text{if } \text{Max}(\text{perSim}_{avg}, \text{spoSim}_{avg}, \text{simSim}_{avg}) = \text{simSim}_{avg} \end{cases} \quad (7)
 \end{aligned}$$

Result and discussion

The method is implemented in Python 3.10 using an Intel Core i7-9700F 3.00 GHz processor, 256 GB SSD Storage, and 16 GB RAM. Three datasets are used: CTU-13 [28], NCC-1 [25], and NCC-2 [27], representing sporadic, periodic, and simultaneous activities, whose characteristics are provided in Tables 1 and 2. The CTU-13 and NCC-1 datasets consist of 13 sub-datasets called scenarios, which describe some entities, like the differences in attack types, botnet names, and protocols used. Differently, NCC-2 consists of 3 sub-datasets called sensors, whereas NCC-2 records network activity consisting of more than one attack scenario. From those three datasets, botnet data are generated with 1,784,699 records.

The similarity measurement has a range of values from 0 to 1. The substantial value of similarity between sequential activities and the characteristics of botnets in the knowledge base is determined based on the measurement of the threshold value (\bar{T}). If this similarity level is higher than the threshold, the sequential activity is detected as a botnet. The threshold is specified using a heuristic approach [49] in the range of between 0.10 and 0.90 by paying attention to changes in the True Positive (TP) and False Positive (FP) values shown in Fig. 5. This figure shows that the FP value began to decrease at a threshold of 0.50, so a more specific heuristic test was carried out with a value range between 0.45 and 0.60. The result of the same evaluation value with the threshold between 0.45 and 0.60 is presented in Fig. 6. It is tested on three sub-dataset samples: scenario 3 of CTU-13, scenario 3 of NCC-1, and sensor 3 of NCC-2, resulting in $\bar{T} = 0.51$, whose FP decreases to almost 0.

Sequential traffic mining and repetitive sequential traffic analysis

The model formed the sequential traffic at this stage, as shown in Table 3, and then removed the repeated sequential traffic to generate a knowledge base. The repetitive

Table 1 Detail of the datasets

Dataset source	Sub dataset total	Botnet traffic	Normal traffic	Total traffic
CTU-13	13	444,699	19,532,001	19,976,700
NCC-1	13	536,000	15,078,345	15,614,345
NCC-2	3	804,000	13,975,083	14,779,083
Total	29	1,784,699	48,585,429	50,370,128

Table 2 Detail of the sub-datasets

Dataset source	Scenario/ sensor	Botnet name	Number of bots source IP	Attack type
CTU-13	1	Neris	1	IRC, SPAM, CF
	2	Neris	1	IRC, SPAM, CF
	3	Rbot	1	IRC, PS, US
	4	Rbot	1	IRC, DDos, US
	5	Virut	1	SPAM, PS, HTTP
	6	Menti	1	PS, HTTP
	7	Sogou	1	HTTP
	8	Murlo	1	PS
	9	Neris	10	IRC, SPAM, CF, PS
	10	Rbot	10	IRC, DDos, US
	11	Rbot	3	IRC, DDos, US
	12	NSIS.ay	3	P2P
	13	Virut	1	SPAM, PS, HTTP
NCC-1	1	Neris	1	IRC, SPAM, CF
	2	Neris	1	IRC, SPAM, CF
	3	Rbot	1	IRC, PS, US
	4	Rbot	1	IRC, DDos, US
	5	Virut	1	SPAM, PS, HTTP
	6	Menti	1	PS, HTTP
	7	Sogou	1	HTTP
	8	Murlo	1	PS
	9	Neris	10	IRC, SPAM, CF, PS
	10	Rbot	10	IRC, DDos, US
	11	Rbot	3	IRC, DDos, US
	12	NSIS.ay	3	P2P
	13	Virut	1	SPAM, PS, HTTP
NCC-2	1	Rbot, Neris, Sogo, NSIS.ay, Virut	10	IRC, PS, DDos, US, SPAM, CF, HTTP, P2P
	2	Rbot, Neris, Menti, Virut	10	IRC, PS, DDos, US, SPAM, CF, HTTP
	3	Rbot, Neris, Murlo, NSIS.ay, Virut	10	IRC, PS, DDos, US, IRC, SPAM, CF, P2P, HTTP

sequential traffic analysis process causes traffic reduction in each dataset, as shown in Table 4. In the CTU-13 dataset, the highest reduction value is in scenario 3, at 99.97%. For NCC-1, the highest activity reduction value is 98.00%, obtained from scenario 6. For the NCC-2 dataset, the activity reduction value is on sensor 3, 93.74%. Repetitive sequential analysis reduces sequential traffic with the same attack pattern and different targets, so a spread attack has a high reduction value. The Port Scanning (PS) attack determines the targets randomly [50] and causes repetitive sequential activities with the same pattern removed. It leads to a high reduction value obtained in the sub-dataset with the PS attack type.

Meanwhile, the lowest reduction value in the CTU-13 dataset is 0 in scenario 11, the same as in the NCC-1 dataset in scenarios 10 and 11. In contrast, the NCC-2 dataset has the lowest reduction value in sensor 1, at 89.32%. Sequential traffic is formed by analyzing the intensity, pattern, source, and target of attacks so that an intense and

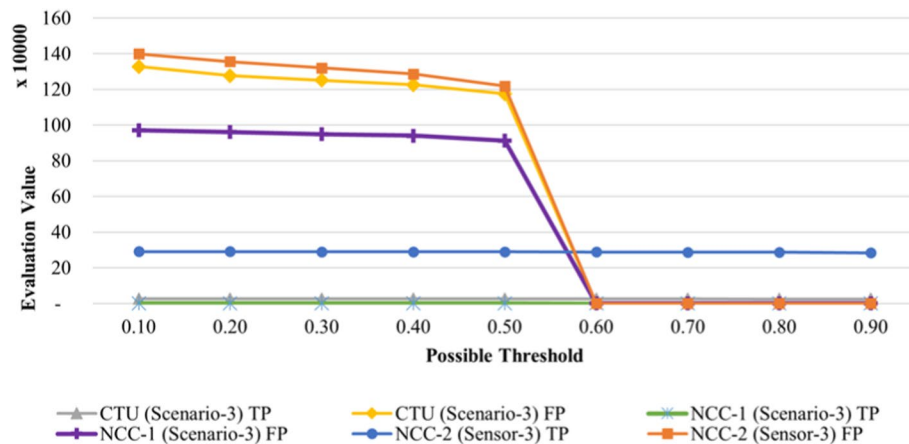


Fig. 5 Evaluation threshold values with possible threshold, from 0.10 to 0.90 using several datasets

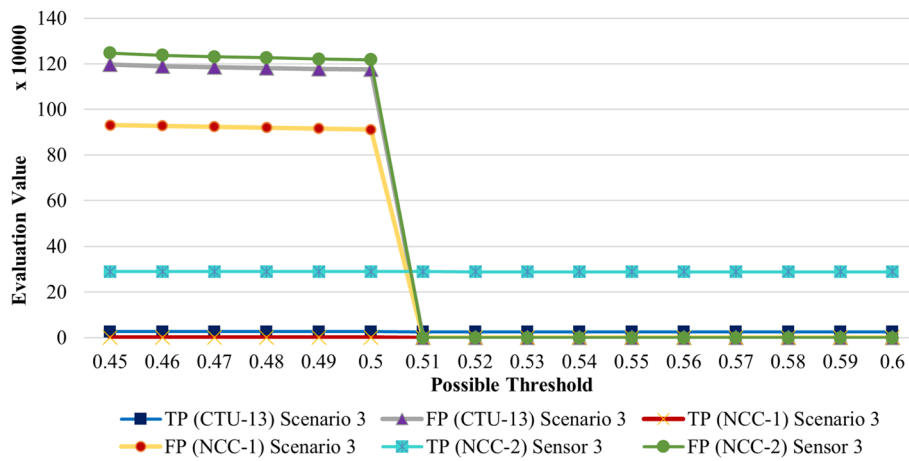


Fig. 6 Evaluation threshold values with possible threshold, from 0.45 to 0.60 using several datasets

concentrated attack creates a complex and non-repeating pattern. This is consistent with the lowest reduction value of 0 obtained from a sub-dataset with a DDoS attack type aiming at flooding a specific target [51].

Detection result with similarity approach

The results of the similarity analysis produce network traffic in the form of *SeqA*, which has a substantial similarity based on the threshold value. Three parameters are used to evaluate the performance: accuracy, precision, and recall, the results of which are provided in Table 5.

The detection result shows that the method works the best on the CTU-13 dataset, whose highest average accuracy is 99.97% and average recall is 97.38%. This proposed method has the highest average precision with 100% on NCC-2. Nevertheless, it underperforms on NCC-1, affected by the poor results in scenarios 3 and 12, which are 44.38% and 31.98%, respectively. It contrasts the other sub-datasets, which get a precision score of 100%. The detection results for sub-datasets other than scenarios 3 and 12 show the lowest recall value of 95.66%, much higher than that from scenarios

Table 3 Sequential activity mining results

Dataset source	Scenario/sensor	Sequential activity detected total	Sequence length		
			Min	Max	Average
CTU-13	1	5167	1	7420	8
	2	1927	1	2814	11
	3	26,789	1	3	1
	4	596	1	1304	4
	5	183	1	275	5
	6	1651	1	199	3
	7	17	1	21	4
	8	1833	1	833	3
	9	27,944	1	15,112	7
	10	163	1	8018	652
	11	14	1	4144	583
	12	1687	1	70	1
	13	2179	1	21,407	18
NCC-1	1	1865	1	1610	12
	2	936	1	1526	26
	3	19	36	1000	105
	4	624	1	1994	18
	5	300	1	1000	63
	6	643	1	1000	9
	7	17	119	1000	529
	8	736	1	1047	19
	9	15,854	1	3000	14
	10	88	181	1819	682
	11	12	1000	1000	1000
	12	939	1	1000	10
	13	853	1	1896	22
NCC-2	1	3637	1	4198	40
	2	18,630	1	6887	20
	3	18,687	1	5543	16

3 and 12, which are 39.90% and 31.34%, respectively. It indicates that the generated knowledge base cannot cover their features. It is worth noting that this dataset focuses on periodic botnet activities.

Botnet characteristic

The measurement in the previous stage obtained bot activity based on its average similarity to kb^{per} , kb^{spo} and kb^{sim} , where the highest average value indicates their attack characteristic, as depicted in Table 6. As described in the previous section, those three datasets have different characteristics: CTU-13, NCC-1, and NCC-2 represent sporadic, periodic, and simultaneous attacks, respectively.

In certain cases, an attack characteristic may have a relatively high probability of meeting all characteristics, although only one is chosen. For example, the similarity level of scenario 12 of the CTU-13 dataset is more than 60% for all characteristics. In that scenario, the IP numbers 147.32.84.165, 147.32.84.191, and 147.32.84.192 have sporadic activity because their attack spreads to the target computers. Some

Table 4 Repetitive sequential traffic analysis results

Dataset source	Scenario/ sensor	Sequential activity total (after repetitive analysis)	Sequence length			Reduction (%)
			Min	Max	Average	
CTU-13	1	379	1	7420	72	92.65
	2	106	1	2814	161	94.56
	3	8	1	3	2	99.97
	4	15	1	1304	105	97.47
	5	39	1	275	19	79.03
	6	26	1	199	58	98.42
	7	7	1	21	8	56.25
	8	44	1	833	33	97.59
	9	1349	1	15,112	105	95.19
	10	70	1	8018	1518	55.97
	11	14	1	4144	583	0
	12	43	1	70	8	97.45
	13	106	1	21,407	328	95.14
NCC-1	1	146	1	1610	139	92.18
	2	101	2	1526	222	89.46
	3	15	36	1000	120	54.55
	4	25	1	1994	404	96.01
	5	57	7	1000	324	81.00
	6	13	10	1000	396	98.00
	7	16	119	1000	555	5.88
	8	46	8	1047	277	93.75
	9	1064	1	3000	184	93.32
	10	88	181	1819	682	0
	11	12	1000	1000	1000	0
	12	37	5	1000	192	96.06
	13	110	2	1896	159	87.31
NCC-2	1	395	1	4198	352	89.32
	2	1257	1	6887	263	93.32
	3	1178	1	5543	221	93.74

botnet activities repeatedly attack the same targets at different times with different intensities, representing periodic. Furthermore, simultaneous attacks at a specific period with different targets make them simultaneous activities. Overall, the method can fully detect a correct botnet attack characteristic.

In the NCC-1 dataset, the method can recognize 12 out of 13 scenarios, where three are detected with 100%, which are scenarios 3, 7, and 11. In scenarios 3, 9, and 10, the method delivers more than 80% of being simultaneous; however, their similarity to the periodic attack is still detected closer. In scenario 12, the attack is incorrectly detected as simultaneous instead of periodic. It is because the method detects the attack simultaneously carried out by different IP numbers: 147.32.84.165 and 147.32.84.191 at a specific period.

Those results inferred that a botnet attack can be either sporadic, periodic, simultaneous, or even a combination of them. For example, the DDoS attack, aiming to flood the target with traffic and form a sequential activity, has a series of activities [10, 34, 51]. Scenarios 10 and 11 of the NCC and CTU-13 datasets result in extensive

Table 5 Detection result with similarity approach

Dataset source	Scenario/ sensor	TP	FP	TN	FN	Accuracy (%)	Precision (%)	Recall (%)
CTU-13	1	38,654	0	2,783,675	2307	99.92	100	94.37
	2	20,050	0	1,787,181	891	99.95	100	95.75
	3	26,759	63	4,683,753	63	100	99.77	99.77
	4	2580	0	1,118,496	0	100	100	100
	5	901	0	128,931	0	100	100	100
	6	4042	0	554,289	588	99.89	100	87.30
	7	63	0	114,014	0	100	100	100
	8	5794	0	2,948,103	333	99.99	100	94.57
	9	183,196	0	1,902,521	1791	99.91	100	99.03
	10	106,352	5	1,203,434	0	100	100	100
	11	8164	0	99,087	0	100	100	100
	12	2143	25	323,278	25	99.98	98.85	98.85
	13	38,547	0	1,885,146	1456	99.92	100	96.36
		Average					99.97	99.89
NCC-1	1	22,576	0	2,089,224	424	99.98	100	98.16
	2	23,944	0	1,441,182	56	100	100	99.77
	3	798	1000	2,902,611	1202	99.92	44.38	39.90
	4	10,957	0	713,388	43	99.99	100	99.61
	5	19,000	0	73,917	0	100	100	100
	6	5954	0	506,021	46	99.99	100	99.23
	7	8881	0	74,473	119	99.86	100	98.68
	8	13,804	0	2,857,217	196	99.99	100	98.60
	9	217,452	0	1,353,304	2548	99.84	100	98.84
	10	57,397	0	924,369	2603	99.74	100	95.66
	11	12,000	0	18,964	0	100	100	100
	12	2821	6000	259,186	6,179	95.56	31.98	31.34
	13	18,864	0	1,857,489	136	99.99	100	99.28
		Average					99.60	90.49
NCC-2	1	140,364	0	4,749,158	5636	99.88	100	96.14
	2	354,918	0	5,634,133	9082	99.85	100	97.50
	3	289,620	0	3,591,792	4380	99.89	100	98.51
		Average					99.87	100

SeqA lengths over 60 min with DDoS attacks. A botnet may target different or the same machine, repeating it at different or the same period. On the other hand, an average activity length affects the formation of *SeqA*, as shown in scenario 4 of the CTU-13 and NCC-1 datasets, even though the number of bot actors is only one. In the case of Port Scanning (PS) and Controlled by Us (US) activities, the length of sequential activity is relatively small. It is due to the attack technique being carried out in a scattered manner without looking at the similarity of the target or focusing on only one target. So, there is no repeated scanning activity at different periods, which makes it referred to as sporadic.

Comparison with other studies

This research compares the experiment results with previous studies to analyze the method further, as provided in Table 7. In general, the detection result using the CTU-13 dataset performs better than previous studies, with an accuracy of 99.97%, a precision

Table 6 Botnet characteristic detection results

Dataset source	Scenario/ sensor	Average similarity value comparison with every characteristic (%)			Predicted
		Sporadic	Periodic	Simultaneous	
CTU-13	1	96.68	13.45	13.31	<i>Sporadic</i>
	2	98.59	9.21	10.29	<i>Sporadic</i>
	3	99.97	50.02	50.01	<i>Sporadic</i>
	4	95.54	20.84	19.79	<i>Sporadic</i>
	5	99.18	18.68	16.67	<i>Sporadic</i>
	6	94.46	32.36	35.60	<i>Sporadic</i>
	7	100	23.03	28.07	<i>Sporadic</i>
	8	98.17	34.79	31.80	<i>Sporadic</i>
	9	97.95	14.70	14.37	<i>Sporadic</i>
	10	100	0.14	0.14	<i>Sporadic</i>
	11	100	0.11	0.11	<i>Sporadic</i>
	12	99.34	61.23	63.48	<i>Sporadic</i>
	13	98.98	5.76	5.45	<i>Sporadic</i>
NCC-1	1	10.57	94.92	34.40	Periodic
	2	6.33	96.93	39.94	Periodic
	3	0	100	81.58	Periodic
	4	7.41	95.96	40.40	Periodic
	5	2.03	98.21	52.94	Periodic
	6	13.36	93.15	23.84	Periodic
	7	0	100	59.33	Periodic
	8	7.31	96.82	64.23	Periodic
	9	10.03	95.17	89.41	Periodic
	10	0.01	99.80	95.93	Periodic
	11	0	100	16.67	Periodic
	12	54.17	66.04	76.27	Simultaneous
	13	7.63	96.25	22.63	Periodic
NCC-2	1	3.47	66.13	97.86	Simultaneous
	2	7.13	65.53	95.43	Simultaneous
	3	8.88	79.03	94.05	Simultaneous

of 99.89%, and a recall of 97.38%. The proposed method has a higher accuracy of 0.03% compared to [7]. At the same time, the precision value is 0.04%, slightly lower than [42] with the Decision Tree. The proposed method shows that the recall value is still 2.07% below [42] with Naïve Bayes. The lower recall value is because the proposed method measures the similarity of *SeqA* with the same size *kb*, while the attack stages may have more or fewer steps depending on the attack cycle. Detection performance can be increased by improving the ability of the proposed method to measure the similarity of *SeqA* with different sizes *kb*.

In the NCC-1 dataset, the method has better accuracy than previous studies, with a value of 99.60%, slightly below [16]. While in precision, the proposed method still needs to be better than [42] with the Decision Tree. Furthermore, its recall is 10.25% lower than [42], dropped by scenarios 3 and 12.

The detection results on the NCC-2 dataset show good precision with a value of 100%, much higher than in previous studies. As for accuracy, the proposed method is

still 0.12% below [42]. For recall, the proposed method gets a score of 97.38%, which is still 2.58% below [42] with Random Forest. The tree-based detection model is superior because this algorithm fits the characteristics of the NCC-2 dataset, which has a large enough botnet traffic data and improves the detection rules.

The proposed model generally has a better detection performance than previous studies. The average value indicates it in those three datasets with different attack characteristics, which results in a detection accuracy of 99.82%, precision of 96.79%, and recall of 94.64%. Table 7 shows that Joshi et al. [7] has a higher average accuracy than the proposed method of 99.94%. It is worth noting that it is an average value, which is the total accuracy obtained from the test results on each dataset divided by the number of datasets used in the testing. In this case, Joshi et al. [7] only used one dataset, namely CTU-13 with sporadic attack characteristics. In contrast, this proposed method is tested on three datasets with different attack characteristics. By using the same dataset (CTU-13) only, the proposed method has better accuracy than [7]. In addition to detecting botnet attacks, this method can further recognize the characteristics of botnet attacks required to determine appropriate handling and anticipation once actual attacks occur on computer networks. In contrast, the other methods cannot do it, as shown in Table 7, because they do not analyze the attack intensity and period.

Discussion of the complexity and overhead

The sequential activity mining process shown in Algorithm 1 is divided into three steps, where overall, the most dominant complexity is in step 2 (time interval analysis). In the worst case, its complexity is $O(m)$, where m is the number of c in C . Meanwhile, the overhead may occur in step 1 when tracking c in C . Thus, the proposed model is efficient with $O(1)$ on average. Still, if more Trf is processed, it will cause a bit of computational overhead. The sequential activity mining process also allows updates to occur continuously on both set C and set $SeqA$, allowing overhead due to the memory reallocation process.

In Algorithm 2 (repetitive sequential activity analysis), complexity again comes from the looping process ($O(o)$) where o is the number of $SeqA$. The greater the number of sequential activities that have been formed, the longer the looping process will be. Comprising a set of kb will also cause space complexity for the same reason, depending on the number of unique $SeqA$ ($O(o)$). On the other hand, the repetitive sequential activity analysis process is made as efficient as possible. It keeps overhead to a minimum from the management set and iteration process.

The similarity analysis presented in Algorithm 3 consists of four steps; the time complexity of each step depends on the iteration process ($O(z)$) where z is the number of $SeqA$ in kb . Meanwhile, space complexity is tied to the storage management of $perSim$, $spoSim$, and $simSim$. Besides these three variables, kb size also contributes significantly to space complexity. On the other hand, the overhead of Algorithm 3 comes from the iterative comparison process followed by the similarity calculation over kb . The entire process will become increasingly complex depending on the data size. Overall, the complexity of the proposed model comes from the iteration process. In contrast, the opportunity for overhead to occur comes from the memory reallocation process, depending on how large the $SeqA$ and kb are processed.

Table 7 Comparison with other studies

Detection model	Dataset used	Botnet detection result (%)			Average result (%)			Botnet characteristic detection
		Accuracy	Precision	Recall	Accuracy	Precision	Recall	
Khan et al. [23]	CTU-13	98.70	–	–	98.70	–	–	No
Joshi et al. [7]	CTU-13	99.94	–	–	99.94	–	–	No
Dollah et al. [42]								
Decision tree	CTU-13	92.20	99.93	84.47	97.27	99.83	94.59	No
	NCC-1	99.63	99.85	99.41				
k-NN	NCC-2	99.98	99.70	99.88				
	CTU-13	75.16	73.18	51.52	90.56	90.21	81.54	
	NCC-1	96.67	99.15	94.18				
Naïve Bayes	NCC-2	99.85	98.31	98.91				
	CTU-13	69.34	62.28	99.45	74.69	54.08	92.48	
	NCC-1	65.38	66.77	82.82				
Random Forest	NCC-2	89.36	33.20	95.17				
	CTU-13	73.83	49.99	47.67	74.99	66.47	49.99	
	NCC-1	51.16	49.55	2.34				
Hostiadi et al. [16]	NCC-2	99.99	99.86	99.96				
	CTU-13	99.18	42.29	91.55	86.33	39.26	96.19	
	NCC-1	99.73	75.14	99.29				
Proposed Method	NCC-2	60.09	0.36	97.73				Yes
	CTU-13	99.97	99.89	97.38	99.82	96.79	94.64	
	NCC-1	99.60	90.49	89.16				
	NCC-2	99.87	100	97.38				

Conclusions

This research proposes a model to detect the characteristics of botnet attacks using a sequential traffic mining approach and similarity analysis. It consists of six main processes, starting with traffic splitting, which aims to get botnet traffic. The second part is sequential traffic mining, focusing on mining and forming sequentially related activity groups. Then, repetitive sequential traffic analysis is used to eliminate repetitive activities. The following process calculates the similarity between the sequential activity, and by using the threshold value, it successfully detects botnet activity. Finally, the method is successful in determining the characteristics of botnet attacks.

In the detection process, the method can detect botnet activity optimally and shows better results than most research, with a detection accuracy of 99.97% on the CTU-13 dataset and 99.60% on the NCC-1 dataset. The proposed model gives better results in precision measurement than previous studies, specifically on the NCC-2 datasets. On average, the proposed model got an optimal accuracy value of 99.82% with the three datasets. In calculating the precision value, the proposed method achieved an average of 96.79%. At the same time, the average recall calculation of the proposed method is 94.64%. The proposed method gets the most optimal value in accuracy compared with previous research. The most optimal values for accuracy mean that the proposed method has the best performance in recognizing botnet activity compared to detection methods in previous studies.

The proposed model has advantages in analyzing the characteristics of botnet attacks. Each detected botnet is analyzed to see the closeness of the similarity of characteristics and see the dominant characteristics contained in each dataset, which can be sporadically, periodically, and simultaneously. First, the model can recognize sporadic activity as an attack by one bot to several targets at different times and carried out with high intensity. Second, on periodic activity, the model can recognize the attack of one bot or group of bots on the same and different targets repeated at different periods. Finally, the model successfully detects simultaneous activity, which identifies an attack by one bot or group of bots simultaneously in the same period. This model provides more information to assist network security administrators in handling botnet attacks based on their characteristics.

In the future, the development will focus on better understanding botnet activities to reduce FP values and optimize recall values. Reducing FP and optimizing recall value can be made by extracting the pattern of sequential activity to have a more varied pattern and the ability to measure the similarity of *SeqA* with different sizes *kb*. Efficiency and effectiveness will also be the main focus of future research while maintaining the complexity and overhead of the proposed model. In addition, it also needs to be tested with other datasets.

Abbreviations

<i>Trf</i>	A record of network traffic
<i>c</i>	Communication Id, the pairwise of <i>SrcAddr</i> and <i>DstAddr</i> feature in <i>Trf</i>
<i>C</i>	Set of Communication Id
<i>SeqA</i>	Sequential traffic, a group of <i>Trf</i> in the order of occurrence time
<i>kb</i>	Knowledge base
<i>per</i>	Botnet periodic characteristic
<i>spo</i>	Botnet sporadic characteristic
<i>sim</i>	Botnet simultaneous characteristic
\bar{I}	Threshold value
∂	Segment period, based on the time required in one botnet attack cycle
<i>sizeOf(x)</i>	Number of elements in <i>x</i>
<i>Max(a, b, c)</i>	Maximum value between <i>a</i> , <i>b</i> , and <i>c</i>
<i>PSim</i>	Pattern similarity
<i>DSim</i>	Time difference similarity
<i>Act</i>	Detection result, which determine the traffic is either botnet or normal

Acknowledgments

The authors thank the PMDSU Scholarship from the Ministry of Education, Culture, Research and Technology, Republic of Indonesia, and Institut Teknologi Sepuluh Nopember (ITS) for supporting this research.

Author contributions

MARP: Conceptualization, methodology, visualization, investigation, software, validation, data curation, writing—original draft; TA: Conceptualization, methodology, writing—review & editing, supervision, funding acquisition, project administration; DPH: Conceptualization, validation, methodology, investigation, formal analysis, writing—review & editing. All authors reviewed the manuscript.

Funding

This study was supported by the PMDSU research grant from the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia and Institut Teknologi Sepuluh Nopember.

Data availability

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare no competing interests.

Received: 18 November 2022 Accepted: 6 March 2024

Published online: 10 April 2024

References

1. Malware. AV-TEST—Indep. IT-Security Inst; 2022. <https://www.av-test.org/en/statistics/malware/>. Accessed 15 Nov 2022.
2. Böck L, Fejrskov M, Demetzou K, Karuppayah S, Mühlhäuser M, Vasilomanolakis E. Processing of botnet tracking data under the GDPR. *Comput Law Secur Rev*. 2022;45: 105652.
3. Melo R, Macedo D, Dantas M, Bona LC. A novel immune detection approach enhanced by attack graph based correlation. In: *IEEE symposium on computers and communications*; 2019. p. 1–6.
4. Abrantes R, Mestre P, Cunha A. Exploring dataset manipulation via machine learning for botnet traffic. *Procedia Comput Sci*. 2022;196:133–41.
5. Krishna E, Arunkumar T. Hybrid Particle swarm and gray wolf optimization algorithm for IoT intrusion detection system. *Int J Intell Eng Syst*. 2021;14:66–76.
6. Priyadarshini R, Barik RK. A deep learning based intelligent framework to mitigate DDoS attack in fog environment. *J King Saud Univ Comput Inf Sci*. 2022;34:825–31.
7. Joshi C, Ranjan RK, Bharti V. A Fuzzy Logic based feature engineering approach for Botnet detection using ANN. *J King Saud Univ Comput Inf Sci*. 2021;34:6872–82.
8. Hoang XD, Nguyen QC. Botnet detection based on machine learning techniques using DNS query data. *Futur Internet*. 2018;10:43.
9. Huancayo Ramos KS, Sotelo Monge MA, Maestre Vidal J. Benchmark-based reference model for evaluating botnet detection tools driven by traffic-flow analytics. *Sensors*. 2020;20:4501.
10. Wang W, Shang Y, He Y, Li Y, Liu J. BotMark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. *Inf Sci (Ny)*. 2020;511:284–96.
11. Zeidanloo H, Tabatabaei F, Vahdani Amoli P, Tajpour A. All about malwares (malicious codes). *Secur Manag*. 2010.
12. Shetu SF, Saifuzzaman M, Moon NN, Nur FN. A survey of botnet in cyber security. In: *2nd international conference intelligent communication and computational techniques ICCT*; 2019. p. 174–7.
13. Daneshgar FF, Abbaspour M. A two-phase sequential pattern mining framework to detect stealthy P2P botnets. *J Inf Secur Appl*. 2020;55: 102645.
14. Muhammad A, Asad M, Javed AR. Robust early stage botnet detection using machine learning. In: *International conference on cyber warfare and security*; 2020. p. 1–6.
15. Kumari K, Mrunalini M. Detecting denial of service attacks using machine learning algorithms. *J Big Data*. 2022;9:56.
16. Hostiadi DP, Ahmad T. 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*. 2022;34:4219–32.
17. Putra MAR, Ahmad T, Hostiadi DP. Analysis of botnet attack communication pattern behavior on computer networks. *Int J Intell Eng Syst*. 2022. <https://doi.org/10.22266/ijies2022.0831.48>.
18. Chowdhury S, Khanzadeh M, Akula R, Zhang F, Zhang S, Medal H, Marufuzzaman M, Bian L. Botnet detection using graph-based feature clustering. *J Big Data*. 2017;4:14.
19. Gaonkar S, Dessai NF, Costa J, Borkar A, Aswale S, Shetgaonkar P. A Survey on Botnet Detection Techniques. In: *International conference on emerging trends in information technology and engineering*; 2020. p. 1–6.
20. Tuan TA, Long HV, Taniar D. On detecting and classifying DGA botnets and their families. *Comput Secur*. 2022;113: 102549.
21. Choi H, Lee H, Lee H, Kim H. Botnet detection by monitoring group activities in DNS traffic. In: *7th IEEE international conference on computer and information technology*. Institute of Electrical and Electronics Engineers (IEEE); 2008. p. 715–20.
22. Zeidanloo HR, Manaf AB, Vahdani P, Tabatabaei F, Zamani M. Botnet detection based on traffic monitoring. In: *International conference on networking and information technology*; 2010. p. 97–101.
23. Khan RU, Zhang X, Kumar R, Sharif A, Golilarz NA, Alazab M. An adaptive multi-layer botnet detection technique using machine learning classifiers. *Appl Sci*. 2019;9:2375.
24. Kwon J, Kim J, Lee J, Lee H, Perrig A. PsyBoG: Power spectral density analysis for detecting botnet groups. In: *Proceedings of the 9th international conference on malicious and unwanted software: the Americas*; 2014. p. 85–92.
25. Hostiadi DP, Ahmad T. Dataset for Botnet group activity with adaptive generator. *Data Br*. 2021;38: 107334.
26. Choi H, Lee H, Kim H. BotGAD: Detecting botnets by capturing group activities in network traffic. In: *4th international ICST conference on COMMunication System softWARE and middleware*; 2009. p. 1–8.
27. Putra MAR, Hostiadi DP, Ahmad T. Botnet dataset with simultaneous attack activity. *Data Br*. 2022;45: 108628.
28. García S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. *Comput Secur Elsevier Ltd*. 2014;45:100–23.
29. Yahyazadeh M, Abadi M. BotCatch: Botnet detection based on coordinated group activities of compromised hosts. In: *7th international symposium on telecommunications*; 2014. p. 941–5.
30. Asadi M, Jabrael Jamali MA, Parsa S, Majidnezhad V. Detecting botnet by using particle swarm optimization algorithm based on voting system. *Futur Gener Comput Syst*. 2020;107:95–111.
31. Homayoun S, Ahmadzadeh M, Hashemi S, Dehghantanha A, Khayami R. BotShark: a deep learning approach for botnet traffic detection. *Adv Inf Secur*. 2018;70:137–53.
32. Hostiadi DP, Wibisono W, Ahmad T. B-corr model for bot group activity detection based on network flows traffic analysis. *KSII Trans Internet Inf Syst*. 2020;14:4176–97.
33. Khodadadi R, Akbari B. Ichnaea: Effective P2P botnet detection approach based on analysis of network flows. In: *International symposium on telecommunications*; 2014. p. 934–40.

34. El-Sofany H. A new cybersecurity approach for protecting cloud services against DDoS attacks. *Int J Intell Eng Syst.* 2020;14:205–15.
35. Prasad KM, Reddy ARM, Rao KV. BARTD: Bio-inspired anomaly based real time detection of under rated App-DDoS attack on web. *J King Saud Univ Comput Inf Sci.* 2020;32:73–87.
36. Hostiadi DP, Ahmad T. Sliding time analysis in traffic segmentation for botnet activity detection. In: 5th international conference on computing and informatics; 2022. p. 286–91.
37. Alejandre FV, Cortés NC, Anaya EA. Feature selection to detect botnets using machine learning algorithms. In: International conference on electronics, communications and computers; 2017.
38. Bueno Silva LF, Nunes Utimura L, Pontara Da Costa KA, Aparecida Zanoli Meira E Silva M, Das Gracias Domingues S. Study on machine learning techniques for botnet detection. *IEEE Lat Am Trans.* 2020;18:881–8.
39. Hostiadi DP, Ahmad T, Wibisono W. A new approach to detecting bot attack activity scenario. *Adv Intell Syst Comput.* 2021;1383 AISC:823–35.
40. Al-Hakbani MM, Dahshan MH. Avoiding honeypot detection in peer-to-peer botnets. In: IEEE international conference on engineering and technology; 2015. p. 1–7.
41. Wang CY, Ou CL, Zhang YE, Cho FM, Chen PH, Chang JB, Shieh CK. BotCluster: A session-based P2P botnet clustering system on NetFlow. *Comput Networks Elsevier.* 2018;145:175–89.
42. Mohd Dollah RF, Faizal MA, Arif F, Masud MZ, Xin LK. Machine learning for HTTP botnet detection using classifier algorithms. *J Telecommun Electron Comput Eng.* 2018;10:27–30.
43. Dong X, Dong C, Chen Z, Cheng Y, Chen B. BotDetector: an extreme learning machine-based Internet of Things botnet detection model. *Trans Emerg Telecommun Technol.* 2021;32: e3999.
44. Alomari E, Manickam S, Gupta BB, Singh P, Anbar M. Design, deployment and use of HTTP-based botnet (HBB) testbed. In: 16th international conference on advanced communications technology; 2014. p. 1265–9.
45. Alzahrani AJ, Ghorbani AA. Real-time signature-based detection approach for SMS botnet. In: 2015 13th annual conference on privacy, security trust; 2015. p. 157–64.
46. Alieyan K, Almomani A, Anbar M, Alauthman M, Abdullah R, Gupta BB. DNS rule-based schema to botnet detection. *Enterp Inf Syst.* 2021;15:545–64.
47. Ayo FE, Awotunde JB, Folorunso SO, Adigun MO, Ajagbe SA. A genomic rule-based KNN model for fast flux botnet detection. *Egypt Inf J.* 2023;24:313–25.
48. Wang A, Chang W, Chen S, Mohaisen A. Delving into internet DDoS attacks by botnets: characterization and analysis. *IEEE/ACM Trans Netw.* 2018;26:2843–55.
49. Ahmad T, Hasbiya T, Ijtihadie R, Wibisono W. Detecting malicious activities in a computer cluster for developing dynamic honeypot. *ICIC Express Lett Part B Appl.* 2018;9:257–64.
50. Marnerides AK, Mauthe AU. Analysis and characterisation of botnet scan traffic. In: International conference on computing, networking and communications; 2016. p. 1–7.
51. Karthik MG, Krishnan MBM. 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.* 2021;14:113–23.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.