Received: May 13, 2022. Revised: June 14, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks

Muhammad Aidiel Rachman Putra[1] Tohari Ahmad[1]* Dandy Pramana Hostiadi[2]

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
[2]*Department of Informatics, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: Botnets are a severe threat to a computer network, affecting various aspects of security systems, including spreading malicious programs, phishing, sending spam messages, and click fraud. Because of their negative consequences, botnets must be identified early. Nevertheless, their different characteristics have made them challenging to detect. This research proposes a bot patterns communication detection from traffic flows analysis consisting of three main activities: bot detection, extraction, and communication behavior analysis phases. This proposed model aims to obtain a specific behavior of bot attacks, which can be used as an early warning bots attack system. The process of bot patterns communication detection depends on the accuracy of bot detection, so the model improves the pre-processing phase and uses multi-model classification. Improvement in the pre-processing phase is carried out in the feature engineering section using the concept of one-hot encoding. Several machine learning classification models are used to obtain the best detection accuracy: Decision tree, Random forest, Logistic regression, *k*-NN, and Naïve Bayes. Furthermore, the model has been tested on two different datasets, namely the NCC and CTU-13. The experimental results show that the proposed model is optimal and recognizes bot activities well. The accuracy detection is obtained at 99.99%. Besides, the model can also identify the bot's attack activity scenario and communication behavior in three types: centralized, distributed, and spread.

Keywords: Botnet, Bot detection, Intrusion detection system, Infrastructure, Network security, Bot communication behaviour.

1. Introduction

System security is needed to maintain the integrity and resources of the device in the communication process. One form of system security in question is a malicious activity detection system known as the Intrusion detection system (IDS). However, an IDS cannot detect all types of attacks accurately, one of which is botnet activity. A botnet is a malicious attack involving illegal software, also known as malware [1–3]. A botnet consists of a collection of computers that have been infected and forms a network communication[3–6]. Infected computers are known as bots or zombies and are controlled by botmaster [4, 6–8] that communicate to bot-client via a communication channel [3, 9] to attack a computer target. Bot's malicious activities can be in the form of distributed denial of service (DDoS), spreading malware, phishing, sending spam messages, and misrepresentation of multi-layer adaptive clicks [1, 10–13]. Botnets use a command & control (C&C) structure in carrying out all their activities [3–9, 14], including communication between bots and botmasters in sending commands and updating code from the botnet control system [4, 8].

In principle, botnet communication can be divided into directly connected (centralized) and decentralized. A centralized botnet has a simple communication structure, and the botmaster sends commands directly through a communication network [5, 15]. In its implementation the centralized botnet uses Internet Relay Chat IRC or HTTP to communicate [8, 12, 13, 15]. Decentralized botnets have a more complex structure than centralized ones [1, 5, 16]. Decentralized communication applies the

peer-to-peer (P2P) model [5, 17–19]. The botmaster sends messages indirectly, making its location challenging to trace and detect. When the suspected botmaster is detected or tracked, it is not immediately blocking the next possible scenarios [1, 8]. Thus, the botmaster can easily form a new topology network and use a collection of bots as an indirect communication [4, 20]. The form of dynamic topology in decentralized communication affects the botnet activities' attack pattern behavior [21].

Previous research has discussed several approaches to detect botnet activity, such as implementing machine learning classification[6, 15, 19, 20], deep learning [5], and clustering [22–24]. On the other hand, some previous studies analyze the flow-based, traffic-based, and graph-based [8] to obtain robust detection models. The previous studies produce high accuracy perform in detecting single bot activity.

Hostiadi et al. [21] introduce a botnet detection model and find that every bot activity is related, known as bot group activities. Their studies use the concept of segmentation, and the result shows that the model can obtain 231 activity scenario patterns by its stages. The scenario of a bot attack is illustrated as between 3 and 7 attacking steps. However, those previous studies have not considered the communication behavior between bots and botmasters or between bots and targets. This behavioral analysis is needed to differentiate between centralized, spread, and distributed attacks. By understanding the characteristics of bots' pattern behavior, the detection model can detect accurately.

This study proposes a new approach to analyzing the communication patterns of botnets. The aim is to collect information about bot attacks and communication patterns behavior in network traffic. This information can optimize an IDS model and be developed as an early warning system. The contribution of this paper is in the pre-processing section. This research combines some techniques, including data transformation, feature engineering with one-hot encoding, and data normalization.

This paper is constructed as follows. The general concepts of botnet structure are presented in section 1. Section 2 contains related works, while the proposed system is described in section 3. The evaluation results are discussed in Section 4. Finally, section 5 is our final remarks and discusses the challenges of future works.

2. Related Works

Some previous schemes have been proposed to identify the attacks. Chowdhury et al. [24] propose a detection methodology based on the topological features of nodes in the graph. Their proposed method creates a cluster of feature nodes in a network. A cluster containing bot nodes has a smaller size than regular nodes, and each bot can be isolated by filtering procedures that select inactive nodes from consideration. This detection model relies on the bot's behavior, and as long as the bot's behavior is different from normal nodes, it is easier to catch. Nevertheless, this research has an issue with computational costs, especially in the feature extraction process.

Dollah et al. [19] propose the detection of botnets using a machine learning approach. This study intends to detect HTTP botnets and proves that the *k*-NN algorithm has the best performance. However, in this study, the detection model does not recognize bot activity scenarios and specific behavior while attacking to computer target.

Khan et al. [15] provide a framework for detecting P2P botnets on a decision tree basis. This research consists of several layers in the detection process, starting from the first layer to filter non-botnet packets to reduce the number of network flows using domain name system queries and flow counting. Then, the packets are filtered and categorized into P2P and non-P2P in the second layer. The third layer reduces the features that are not affected in the classification process. At the last layer, P2P botnets are detected using decision tree classification. Their experimental results show that the proposed model is reliable in detecting P2P botnets. This model shows high accuracy detection but cannot describe the pattern or botnet behavior's specific characteristics.

Joshi et al. [4] propose a botnet detection approach using fuzzy logic and artificial neural network (ANN) models. The research uses a public dataset, namely the CTU-13 dataset, which consists of different characteristics of a bot scenario attack. The fuzzy logic extracts new features from existing ones in the datasets. New features have been generated and selected with several experiments. The bot detection model uses the selected features that meet the threshold values. The concept of ANN with one input layer, one output layer, and four hidden layers is used in the detecting process. This method shows a high accuracy detection in detecting botnets. However, the experiment is only carried out by one scenario in the CTU-13 dataset. In fact, the CTU-13 dataset consists of 13 scenarios of bot attack activity with different characteristics.

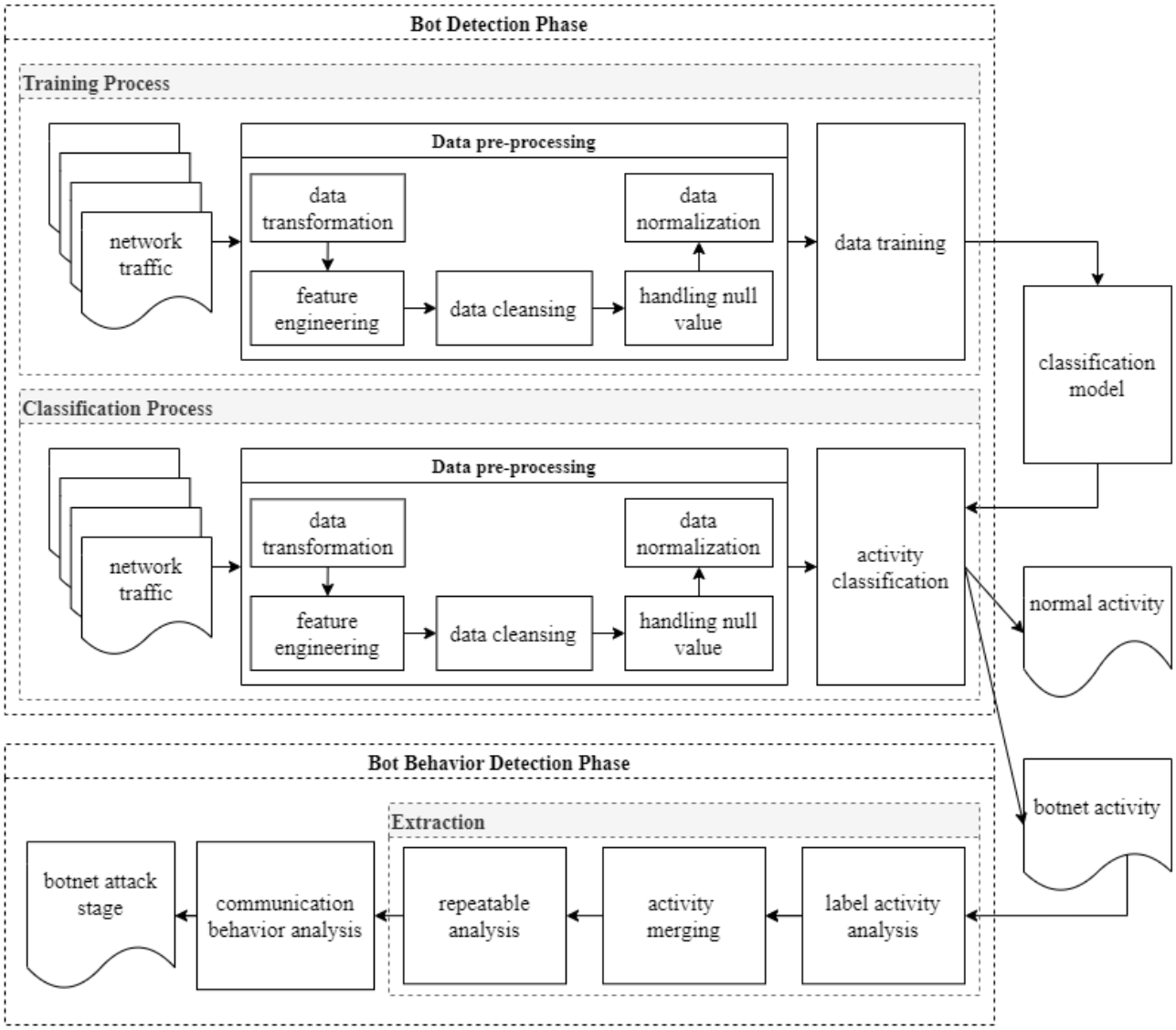

Figure. 1 The proposed model

Hostiadi et al. [21] detect botnets and obtain 23 types of bot-activity-labels. Furthermore, their proposed model can see the attack stages ranging from one to seven, applying a segmentation concept by optimizing sliding time analysis and dividing the dataset into several segments by 1 hour and 30 minutes to slides. This sliding time window is used to anticipate a possible loss of information during the transition between segments, which is affected in bot chain communication analysis. Nevertheless, that model has not analyzed the bot spread activity in each segment. In fact, the spread activity is used to trace the source of bot attacks and is categorized into intensive bot activity. Besides, the proposed method does not consider the communication of bot behaviors while attacking its target.

Several previous studies have introduced a detection model that can produce high performance and accuracy. Unlike this research, this proposed method detects botnet activity and analyzes the communication behaviors to their targets. The communication behavior patterns are needed to analyze the relationship between bot attack activities. It is to be a knowledge-based system containing bot attack behavior. Besides, it can use to anticipate and used as an early warning system from botnet attacks.

3. Proposed method

This paper proposes a method for detecting bot communication patterns behavior in carrying out attacks. The model is constructed in several main processes, including bot activity detection, bot activity extraction, and bot communication pattern analysis. Generally, the proposed model is shown in Fig. 1.

Table 1. Transforming IPv4 into numeric

IPv4	*inet_aton()*	*unpac()*
147.32.84.209	b'\x93 T\xd1'	2468369617
74.125.232.196	b'J}\xe8\xc4'	1249765572
202.127.80.17	b'\xca\x7fP\x11'	3397341201
147.32.84.165	b'\x93 T\xa5'	2468369573
….	…	…
94.63.150.63	b'^?\x96?'	1581225535

Algorithm 1. One-hot encoding

INPUT: $\{fc_i\}_{i=1}^{h}$
OUTPUT: F
/* F: all features on dataset after feature engineering*/

$F \leftarrow \{fn_j\}_{j=1}^{g}$; $newf \leftarrow \{\}$

$u \leftarrow number\ of\ network\ traffic$

Step 1: Generate new features
  for $i \leftarrow 1\ to\ h$ ***do***
    for $k \leftarrow 1\ to\ u$ ***do***
      if $fc_{i,k} \notin (F \cup newf)$:
        $newf = newf \cup \{fc_{i,k}\}$
      else
        go to step 2
      end if
    end for
  end for
Step 2: Encode
  for each f **in** $newf$ ***do***
    for $k \leftarrow 1\ to\ u$ ***do***
      if $fc_{i,k} = f$
        $f_k \leftarrow$ **1**
      else
        $f_k \leftarrow$ **0**
      end if
    end for
  end for
  $F = F \cup newf$
Step 3: Return

3.1 Bot Detection

This section divides network traffic into two classes, bots and normal, by implementing the machine learning classification as a botnet detection model. It comprises several phases: data transformation, feature engineering, data cleansing, handling null values, and data normalization.

3.1.1. Data Transformation

If NT is a dataset of collection *network traffic* with many features, it can be represented as $NT = \{f_1, f_2, \dots, f_l\}$ where l is number of features in NT. Because every f_i has different type, namely categorical and numerical, it can be represented as $NT = \{fn_1, fn_2, \dots, fn_g, fc_1, fc_2, \dots, fc_h\}$. First part $\{fn_1, fn_2, \dots, fn_g\}$ is numerical, and $\{fc_1, fc_2, \dots, fc_h\}$ is categorical. The number of categorical and numerical features is represented by g and h, where $g + h = l$. Several methods are used for the conversion of fc, using Python library, dictionary [19], and feature engineering.

Table 2. Data before normalization

Dur	SrcAddr	Sport	…	SrcBytes
2752.65625	1.21E+09	60621	…	290
1849.31555	1.32E+09	51413	…	272
2091.74731	5.30E+08	63195	…	145
1535.76941	1.98E+09	39110	…	145
…	…	…	…	…
0.00264	2.47E+09	33426	…	321

This phase focuses on converting fc using the python library and dictionary. The features requiring transformation are $SrcAddr$, $DstAddr$, and $State$. The $SrcAddr$ and $DstAddr$ are categorical features in the internet protocol version 4 format, which has a 32-bit number. IPv4 addresses are represented as four decimal numbers ranging from 0 to 255 and are separated by periods. In this case, the model uses Python's $inet\_aton()$ function to convert IPv4 to a 32-bit binary format. Since the return from the $inet\_aton()$ function is still categorical, it is passed with the $unpac()$ function to form into the specified format. The IPv4 format is a package to the output as an unsigned long data type, whose output of each IPv4 transformation process is described in Table 1. The state feature is transformed using a dictionary from Python [19], aiming to convert the data in the state feature into a predefined integer. Now, $SrcAddr$, $DstAddr$, and $State$, which were previously fc, have been converted to fn. So, after data transformation, g decreases by 3, while h increases by 3.

3.1.2. Feature engineering

In network traffic, there are several essential features to represent the activity of hosts on the network. Some basic features are used in this research, as in [6, 19]. The remaining fc can be developed into new ones through the engineering feature using one-hot encoding [25]. For example, the Dir, can be explored into features of $Dir\_$<-, $Dir\_$<, $Dir\_$<-, $Dir\_$ <->. Every fc_r where $r = 1,2, \dots, h$ has different value in *network traffic;* it can be

Algorithm 2. Activity Merging

INPUT: a
OUTPUT: $attackStages$
/* a: network activity; $attackStages$: result of activity merging */
$src \leftarrow SrcAddr$; $dst \leftarrow DstAddr$
$SA(src, addr) \leftarrow \{\}$; $attackStages \leftarrow \{\}$
$u \leftarrow number\ of\ network\ traffic$
Step 1: Create SA
  for $p \leftarrow 1$ to u **do**
    $src \leftarrow a_p(src)$
    $dst \leftarrow a_p(dst)$
    if $attackStages(src, addr) \notin SA$
      $SA = SA \cup \{attackStages(src, addr)\}$
    else
      go to step 2
    end if
  end for
Step 2: Push a to $attackStages$
  for $p \leftarrow 1$ to u **do**
    $attackStages(src, addr) = attackStages(src, addr) \cup \{a_{p(src,addr)}\}$
    /* Push a to $attackStages$ with the same src & *dst*/
  end for
Step 3: Return

Algorithm 3. Repeatable Analysis

INPUT: $attackStages$
OUTPUT: ρ
/* ρ : group activity */
$n \leftarrow number\ of\ A\ in\ attackStages$
Step 1: Check the similarity of each adjacent activity
  if $A_i = A_j$
    $attackStages \leftarrow attackStages - \{A_j\}$
  else if $i < n$
    $i = i + 1$
  else
    go to step 2
  end if
Step 2: Check the group activity
  for each k **in** $attackStages$ ***do***
    if $A_i = A_k$
      $\rho \leftarrow \{A_i, ..., A_{k-1}\}$
    else
      go to step 3
    end if
  end for
Step 3: Group Activity repeatable analysis
  while $attackStages \neq \{\}$ **do**
    if $\rho \in attackStages$
      $\rho \leftarrow attackStages \cap \rho$
    else
      $\rho \leftarrow attackStages$
    end if
  end while
Step 4: Return

represented as $fc_r = [fc_{r,1}, fc_{r,2}, \ldots, fc_{r,u}]$, where u is the number of *network traffic* in the dataset. One-hot encoding analyzes every *network traffic* to form new features. To describe the value of every *network traffic*, each new feature contains a value of 1 or 0. The mechanism of feature engineering using one-hot encoding is shown in Algorithm 1. Then, we add the $predictedLabel$ feature containing data "0" for normal and "1" for bot activity network traffics [19], and the $predictedLabel$ is used for the training process.

3.1.3. Data cleansing

Network traffic has some feature data without values. Thus, data cleansing is needed to remove those unnecessary features to process in the bot detection phase. For example, the $sTos$ feature has a null value percentage of 14% and can affect accuracy detection. Some categorical features should be eliminated for specific reasons. For example, the $sTos$ and $dTos$ are eliminated because their rate of null values is too high; the $StartTime$ feature is also removed, considering that the bot identification is not directly related to the time series.

3.1.4. Handling null value

Not all features with null values can be removed in the data cleansing process. Specific features, such as $Dport$, $Sport$, and $State$, contain necessary information. To solve this condition, in the handling null value stages, the model converts that null value to 0.

3.1.5. Data normalization

After going through several previous stages, all data have been numeric. The different data range of each feature needs to be standardized using data normalization. For example, the $SrcAddr$ and $DstAddr$, are *unsigned long* types with a different range of values. The comparison of the values of each feature can be seen in Table 2.

The numerical data are produced in pre-processing stage and formed as standard values to be

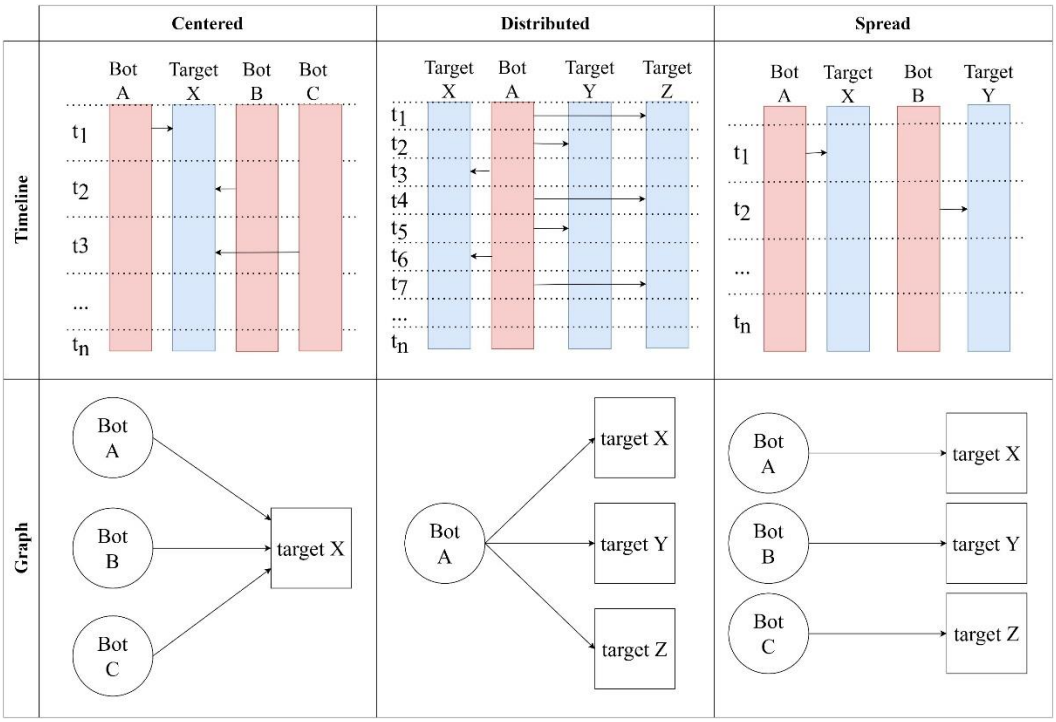

Figure. 2 Bot attack patterns

used as input in the identification process using machine learning concepts. The identification begins with modeling in the training process and ends with activity classification in the detection. The output of this process is a set of network traffic identified as a bot or normal class.

3.2 Extraction

This phase is carried out by collecting any information from network traffic identified as bots. The extraction phase consists of three main activities: Label Activity Analysis, Activity Merging, and Repeatable Analysis.

3.2.1. Label activity analysis

Label activity is analyzed in the serial grouping stage of the botnet activity. Generally, one type of bot attack in the form of activity labels can occur on different targets with different activity times. Thus, activities must be grouped to obtain the same labels. For example, serial activities like "*flow=From-Botnet-V42-TCP-Attempt-SPAM*", "*flow=From-Botnet-V43-TCP-Attempt-SPAM*", "*flow=From-Botnet-V44-TCP-Attempt-SPAM*" are assigned into "flow=*From-Botnet-TCP-Attempt-SPAM*".

3.2.2. Activity merging

Any network traffic identified as bots is an attack activity, each of which has a relationship that forms a bot attack scenario [21]. Each attack scenario denoted as SA comprises attack activity (a_i) where i is the index of attack activity with $i = 1,2,3,\dots,n$. So, it can be defined as:

$$SA = f(a_1, a_2, a_3, \dots, a_n);\ i \in a;\ i = 1,2,3,\dots,n \quad (1)$$

At this stage, each a_i has a $Label$ describing the type of activity being carried out. Any activity $Label$'s that has an exact match with the same time serial will be assigned to one bot activity label. For example, there are activities from $SrcAddr$ to $DstAddr$ with the same and have different time series of activities. Then, the same activity labels are merged until the next activity label appears. If that serial activity label is declared as $a_{i,j}$ with $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$, then the scenario activity $(SA) = \{ A_I, A_J \}$, where $A_i \neq A_J, n$ is the activity label that is repeated for a one-to-one relationship between $SrcAddr$ and $DstAddr$. The results of the merging process produce activity stages known as $attackStages$. It is necessary to analyze the stages of the attack to recognize a bot's activity pattern.

Each bot has a different behavior of attacking. Several bots can intensively or in group attacks against a computer target and need to merge the activity labels analyzed previously to obtain the pattern. The merging stage begins by grouping each a_i with features according to $SrcAddr$ and $DstAddr$. This grouping is done in stages by analyzing A_i according to the time of its appearance, referring to $StartTime$. Algorithm 2 demonstrates the

Table 3. CTU-13 dataset description

CTU Scenario	Characteristic of botnet scenario	Bot name	Bots count	Net Flow Count	Botnet Flows (%)
1	IRC, SPAM, CF	Neris	1	2,824,637	1.410
2	IRC, SPAM, CF	Neris	1	1,808,123	1.040
3	IRC, PS, US	Rbot	1	4,710,639	0.560
4	IRC, DDoS, US	Rbot	1	1,121,077	0.150
5	SPAM, PS, HTTP	Virut	1	129,833	0.530
6	PS, HTTP	Menti	1	558,920	0.790
7	HTTP	Sogou	1	114,078	0.030
8	PS	Murlo	1	2,954,231	0.170
9	IRC, SPAM, CF, PS	Neris	10	2,753,885	6.500
10	IRC, DDoS, US	Rbot	10	1,309,729	8.110
11	IRC, DDoS, US	Rbot	3	107,252	7.600
12	P2P	NSIS.ay	3	325,472	0.650
13	SPAM, PS, HTTP	Virut	1	1,925,150	2.010

Table 4. NCC dataset description

NCC Scenario	Characteristic of botnet scenario	Bot Name	Bots Count	Net Flow Count	Botnet Flows (%)
1	IRC, SPAM, CF	Neris	1	2,112,224	1.090
2	IRC, SPAM, CF	Neris	1	1,465,182	1.640
3	IRC, PS, US	Rbot	1	2,905,611	0.070
4	IRC, DDoS, US	Rbot	1	724,388	1.520
5	SPAM, PS, HTTP	Virut	1	92,917	20.450
6	PS, HTTP	Menti	1	512,021	1.170
7	HTTP	Sogou	1	83,473	10.780
8	PS	Murlo	1	2,871,217	0.490
9	IRC, SPAM, CF, PS	Neris	10	1,573,304	13.980
10	IRC, DDoS, US	Rbot	10	984,369	6.100
11	IRC, DDoS, US	Rbot	3	30,964	38.750
12	P2P	NSIS.ay	3	274,168	3.280
13	SPAM, PS, HTTP	Virut	1	2,876,489	1.100

mechanism of activity merging, whose result contains details of the communication group with attack activities sorted by time of appearance. The attack activities that have been grouped into a communication group are called attack stages.

3.2.3. Repeatable analysis

Attack stages that have been identified in the previous process contain repeatable patterns, which need to be analyzed more deeply to obtain bot communication behaviors. Algorithm 3 describes this repeatable analysis mechanism. In this phase, the aim is to find patterns of bot attack activity that have unrepeatable attack stages. If $A_{I,J}$ exists in the following time without appearing of A_K where $K \neq I \neq J$, it is expressed as $A_{I,J,K}$. This pattern is called group activity [21], denoted as ρ. Then, ρ is stored and used as a knowledge base to recognize the characteristics of the bot.

3.3 Communication Behavior Analysis

The bot's behavioral analysis is to differentiate the characters of botnet attacks. Bot attack patterns are categorized into three types: centralized, spread, and distributed attacks, as depicted in Fig. 2. Centered is a communication behavior that occurs when a target communicates, which is an attack from several bots once at a particular time. Besides, when a bot intensely carries out attack activities towards several targets at once, it is a distributed communication behavior. A botnet only performs one

Algorithm 4 Communication Behavior Analysis

INPUT: ρ
OUTPUT: Centered/Distributed/Spread Communication Behavior
/* ρ : group activity */
$src \leftarrow SrcAddr;\ dst \leftarrow DstAddr$
$u \leftarrow number\ of\ A\ in\ \rho$
Step 1: Check source intensity communication
  for $i \leftarrow 1; j \leftarrow 2$ to u **do**
    if $A_i.src \neq A_j.src$
      $S.src = \{\}$
    else go to step 2
    end if
  end for
Step 2: Check target intensity communication
  for $i \leftarrow 1; j \leftarrow 2$ to u **do**
    if $A_i.dst \neq A_j.dst$
      if $R.dst \neq \{\}$
        it is **distributed**, go to step 4
      else
        $R.dst \leftarrow \{\}$
      $S.src \leftarrow dst; R.dst \leftarrow src$
    else go to Step 3
    end if
Step 3: Analysis target
  if $(R.dst \in S.src \wedge n(S.src) = 1)$
    it is **spread**
  else it is **centered**
  end if
Step 4: Return

attack on a particular target without further activity in a specific case. For this type, we define it as spread communication behaviors. Each communication behavior can describe the characteristics of a bot. Knowing the bot's attack characteristics helps recognize a bot better and can even predict the bot's activity.

Communication behaviors analysis receives input from the previous process, namely ρ, which contains information in $SrcAddr$, $DstAddr$, $attackStages$ and $repeatableLabels$ (recurring single and group activities). The process starts by analyzing the $SrcAddr$ in each a in ρ. If in a ρ there are several identical $SrcAddr$ then they are analyzed to determine whether each $SrcAddr$ has an identical $DstAddr$. Identical $SrcAddr$ and $DstAddr$ indicate intense communication between the bot and its target. The communication activity discussed is in the form of attacks launched by bots. Intense communication is considered as a distributed communication behavior. If the ρ has identical $SrcAddr$ and $DstAddr$ without repeatable activity, its communication history is analyzed to determine when the target received an attack. Algorithm 4 describes this communication behaviors analysis process, categorizing it into centered, distributed, and spread.

Table 5. Result of bot detection with several classification algorithms

Algorithm	Accuracy (%)
Random Forest	99.998
Naïve Bayes	88.459
k-NN	99.930
Decision Tree	99.997
Logistic Regression	98.764

Table 6. Total bot-activity-step

Model	Scenario detected	Bot-activity-step		
		<3	3	>3
Hostiadi et al. [21]	231	8	64	159
Proposed Model	107	80	26	1

4. Implementation

This section provides the experimental results, including bot detection, extraction, communication patterns, and comparison with previous research.

4.1 Dataset

This study uses two datasets, namely CTU-13 [26] and NCC [27], shown in Table 3 and Table 4, respectively. The malicious bot activities in those datasets can be in the forms of IRC, SPAM, click fraud (CF), port scan (PS), fast flux (FF), and controlled by us (US). Both datasets are distinguished based on their characteristic of the attack scenario. That is, the NCC dataset presents and records periodic and intense bot activity, while the CTU-13 dataset shows the sporadic types of bot activities.

4.2 Bot detection

At the bot detection stage, we take a sample of 70% of the total data in each scenario for training. For this purpose, we mark each bot flow as 1 and 0 for normal activities [19]. The training data are proportional, consisting of 70% for each bot and normal flow from the dataset scenario. The bot detection process uses a random forest algorithm classification method due to its stable performance in

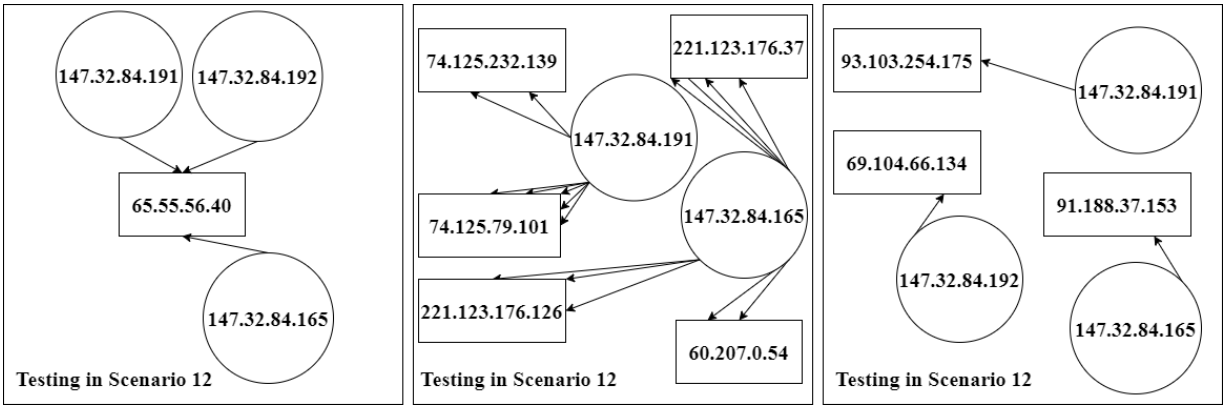

Figure. 3 Result of botnet communication behavior detection: (a) centered, (b) distributed and (c) spread

Table 7. CTU-13 communication behavior pattern experimental result

CTU scenario	Characteristics of Botnet Scenario	Bots count	Botnet flows (%)	Target	Communication pattern behavior (%)		
					Centered	Distributed	Spread
1	IRC, SPAM, CF	1	1.410	4,181	0.000	100.000	0.000
2	IRC, SPAM, CF	1	1.040	1,648	0.000	100.000	0.000
3	IRC, PS, US	1	0.560	26,715	0.000	99.996	0.004
4	IRC, DDoS, US	1	0.150	590	0.000	100.000	0.000
5	SPAM, PS, HTTP	1	0.530	180	0.000	100.000	0.000
6	PS, HTTP	1	0.790	1,578	0.000	100.000	0.000
7	HTTP	1	0.030	16	0.000	100.000	0.000
8	PS	1	0.170	493	0.000	100.000	0.000
9	IRC, SPAM, CF, PS	10	6.500	14,396	12.045	87.955	0.000
10	IRC, DDoS, US	10	8.110	33	19.469	80.531	0.000
11	IRC, DDoS, US	3	7.600	8	23.077	76.923	0.000
12	P2P	3	0.650	1,456	9.180	90.494	0.326
13	SPAM, PS, HTTP	1	2.010	1,687	0.000	100.000	0.000

Table 8. NCC communication behavior pattern experimental result

NCC scenario	Characteristics of Botnet Scenario	Bots count	Botnet flows (%)	Target	Communication pattern behavior (%)		
					Centered	Distributed	Spread
1	IRC, SPAM, CF	1	1.090	1,196	0.000	100.000	0.000
2	IRC, SPAM, CF	1	1.640	568	0.000	100.000	0.000
3	IRC, PS, US	1	0.070	19	0.000	94.737	5.263
4	IRC, DDoS, US	1	1.520	394	0.000	100.000	0.000
5	SPAM, PS, HTTP	1	20.450	175	0.000	100.000	0.000
6	PS, HTTP	1	1.170	469	0.000	100.000	0.000
7	HTTP	1	10.780	17	0.000	100.000	0.000
8	PS	1	0.490	343	0.000	100.000	0.000
9	IRC, SPAM, CF, PS	10	13.980	4,807	13.522	86.478	0.000
10	IRC, DDoS, US	10	6.100	28	20.408	79.592	0.000
11	IRC, DDoS, US	3	38.750	9	18.182	81.818	0.000
12	P2P	3	3.280	403	4.762	93.878	1.361
13	SPAM, PS, HTTP	1	1.100	508	0.000	100.000	0.000

classifying techniques, especially in network traffic analysis. The detection results using several classification algorithms are depicted in Table 5.

4.3 Extraction

The extraction is done through label activity analysis, activity merging, and repeatable analysis. In this phase, the model obtains 38 activity labels,

Table 9. Comparison between research methods

Model	Dataset	Accuracy of Bot Detection Result (%)	Number of Scenarios Detected	Consider Attack Behavior
Dollah et al. [19]	CTU-13 & Private Dataset	92.930	-	No
Khan et al. [15]	CTU-13 & ISOT	98.700	-	No
Joshi et al. [4]	CTU-13	99.940	-	No
Hostiadi et al. [21]	CTU-13	94.200	231	No
Proposed Model	CTU-13 & NCC	99.998	107	Yes

higher than [21] which can detect 23 labels. Furthermore, this research takes 107 scenarios; 80 scenarios can be detected within less than three steps, 26 with three steps in carrying out an attack, and one with more than three attack steps. In detail, the required steps to detect are provided in Table 6.

4.4 Communication behavior

Bot activity needs further analysis to obtain attack communication behavior patterns to show specific characteristics. Based on the extraction results, three characteristics of bot behavior are defined as the goal of this research, shown in Fig. 3, and the experimental results are shown in Table 7.

Each bot has different communication behaviors depending on the number of bots, the actions taken, and the intended targets. It is found that scenarios 1, 2, 3, 4, 5, 6, 7, 8, and 13 contain one bot whose communication behaviors tend to be distributed. Specifically, in scenario 3, a bot with the name *Rbot* performs spread communication behaviors on a target in the form of PS. If the bot is doing PS, no further attacks and attempts are made on similar targets. Thus, this communication only occurs once. Besides, P2P bots with bot names in NSIS.ay tend to vary their communication behaviors. Although 90% of communication behaviors are distributed, the model can define the centered communication behaviors as 9% of the total botnet flows. In addition, there is also a spread communication behavior of 6 times. The highest total for the type of centered communication behaviors is performed in DDoS activity. Is considering the purpose of the attack is to flood the target [2, 8, 9, 28]. In addition, bots like *Nerris* in Scenario number 9 do much communication, with fourteen thousands of the total targets. *The Nerris* bots perform 3628 centralized attacks or about 12% of the total botnet flow, with 88% of them being distributed attacks. In total, distributed communication behavior is very dominant in the CTU-13 dataset, which accounts for 94% of the existing bot flows.

The result of defining bot activity in the NCC dataset is similar to one bot's existing, as shown in Table 8. Several experiments show that the dataset scenario with one bot adopts the concept of a distributed communication pattern during an attack. Unique activities are *Rbots* that perform spread communication behaviors for 5% of the total botnet flow. Besides, the NCC dataset's experiment shows that DDoS attacks as centered on communication behaviors for 20% of total activities. This value is smaller than the CTU-13 dataset. Accumulatively, the NCC dataset has a higher percentage of centered communication pattern behaviors than CTU-13, with 10%.

4.5 Comparison between research methods

In this research, we compare the result of the experiment with several previous studies, shown in Table 9. Previous studies mainly use CTU-13 as the dataset, and some others take different datasets, such as ISOT [15] and private datasets [19]. For research that uses more than one dataset, the accuracy value is their average. In more detail, the proposed method obtains an accuracy value of 99.998% with CTU-13 and 99.999% with the NCC dataset. The pre-processing stage is caused by several stages, such as feature extraction, handling null values, and others. Furthermore, the detection of combination scenarios is lower than [21].

5. Conclusions

This paper proposes an approach to detect botnets based on the characteristics of the bot pattern behavior. This proposed method consists of three main processes. The first process detects bot activity using the random forest classifier. The pre-processing stage is optimized by the data transformation process, feature engineering, data cleansing, handling null values, and normalization in detecting bot activity. Then, the result of bot detection is extracted to obtain the information on bot behavior attacks.

The experiment uses two types of datasets: CTU-13 and NCC. It is found that the model can detect three distinct types, namely centered, distributed, and spread. Compared with previous studies, the proposed method is more accurate in detecting bot activity and recognizing activity. Furthermore, the distributed communication behavior often appears with 94% activity on the CTU-13 dataset and 89% on the NCC; the centered communication is mainly used by botnets carrying out specific attack activities such as DDoS, SPAM, IRC, and Click Fraud. The NCC dataset has a proportion of activity with centered communication behavior greater than CTU-13 by 10% of the total bot attack activity in the dataset. The spread communication behavior is more consistently defined in Rbot and NSIS.ay. However, this method has a weakness in extracting the steps of botnet attacks due to the smaller number of activities.

In the future, the improvement works on the extraction problem, specifically at the activity merging stage. It is to minimize the effects caused by that reduced number of activities. Besides, various botnet datasets may need to be generated.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

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

Acknowledgments

This work was supported by the Ministry of Education, Culture, Research, and Technology, the Republic of Indonesia.

References

[1] L. Böck, M. Fejrskov, K. Demetzou, S. Karuppayah, M. Mühlhäuser, and E. Vasilomanolakis, "Processing of botnet tracking data under the GDPR", *Comput. Law Secur. Rev.*, Vol. 45, p. 105652, 2022, doi: https://doi.org/10.1016/j.clsr.2021.105652.

[2] 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.

[3] 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.

[4] 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.

[5] S. Homayoun, M. Ahmadzadeh, S. Hashemi, A. Dehghantanha, and R. Khayami, "BoTShark: A deep learning approach for botnet traffic detection", *Advances in Information Security*, Vol. 70, *Springer New York LLC*, pp. 137–153, 2018. doi: 10.1007/978-3-319-73951-9_7.

[6] 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.

[7] K. S. Huancayo Ramos, M. A. S. Monge, and J. M. Vidal, "Benchmark-Based Reference Model for Evaluating Botnet Detection Tools Driven by Traffic-Flow Analytics", *Sensors*, Vol. 20, No. 16, 2020, doi: 10.3390/s20164501.

[8] 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.

[9] Y. Aleksieva, H. Valchanov, and V. Aleksieva, "An approach for host based botnet detection system", In: *Proc. of Int. Conf. Electr. Mach. Drives Power Syst.*, 2019, doi: 10.1109/ELMA.2019.8771644.

[10] R. Melo, D. Macedo, M. Dantas, and L. C. Bona, "A Novel Immune Detection Approach Enhanced by Attack Graph Based Correlation", In: *Proc. of IEEE Symp. Comput. Commun.*, pp. 1–6, 2019, doi: 10.1109/ISCC47284.2019.8969772.

[11] E. Krishna and T. Arunkumar, "Hybrid Particle Swarm and Gray Wolf Optimization Algorithm for IoT Intrusion Detection System", Int. *J. Intell. Eng. Syst.*, Vol. 14, pp. 66–76, 2021, doi: 10.22266/ijies2021.0831.07.

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

[13] 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.

[14] H. Suryotrisongko and Y. Musashi, "Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection", *Procedia Comput. Sci.*, Vol. 197, pp. 223–229, 2022, doi: https://doi.org/10.1016/j.procs.2021.12.135.

[15] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers", *Appl. Sci.*, Vol. 9, No. 11, Jun. 2019, doi: 10.3390/app9112375.

[16] L. Mathur, M. Raheja, and P. Ahlawat, "Botnet Detection via mining of network traffic flow", In: *Procedia Computer Science*, Vol. 132, pp. 1668–1677, 2018. doi: 10.1016/j.procs.2018.05.137.

[17] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches", In: *Proc. of IEEE Conference on Communications and Network Security*, pp. 247–255, 2014. doi: 10.1109/CNS.2014.6997492.

[18] 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.

[19] R. F. M. Dollah, M. A. Faizal, 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, No. 1–7, pp. 27–30, 2018.

[20] 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*, pp. 203–208, 2020. doi: 10.1109/IIT50501.2020.9299061.

[21] 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.

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

[23] 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.

[24] 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.

[25] X. Dong, C. Dong, Z. Chen, Y. Cheng, and B. Chen, "BotDetector: An extreme learning machine-based Internet of Things botnet detection model", *Trans. Emerg. Telecommun. Technol.*, Vol. 32, p. e3999, 2021, doi: 10.1002/ett.3999.

[26] 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: 10.1016/j.cose.2014.05.011.

[27] D. P. Hostiadi and T. Ahmad, "Dataset for Botnet group activity with adaptive generator", *Data Br.*, Vol. 38, 2021, doi: 10.1016/j.dib.2021.107334.

[28] 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.