Journal of Theoretical and Applied Information Technology
31st July 2021. Vol.99. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THRESHOLD IDENTIFICATION FOR P2P BOTNET DETECTION USING LOGISTIC REGRESSION APPROACH

[1] MOHD FAIZAL ABDOLLAH*, [2]WAN AHMAD RAMZI W. YA, [1]WARUSIA YASSIN, [1]RIZUAN BAHARON, [3]MOHD FAHMI ARIF

[1] Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.

[2] Department of Computer System and Electrical Technology, Masjid Tanah Community College, Melaka

[3] Fakulti Teknologi Industri, Institut Teknologi Nasional Bandung, Indonesia

Email: [1]faizalabdollah@utem.edu.my, [1]rizuan.baharon@utem.edu.my, [1]s.m.warusia@utem.edu.my, [2]ramzi016@gmail.com, [3]fahmi.arif@itenas.ac.id.

ABSTRACT

Nowadays, all matters involving communication network that covers various fields such as banking, business, learning, and social media should be monitored as they are exposed to various cyber crimes and aggression committed by the irresponsible. Identifying an appropriate threshold value is essential to differentiate between normal and abnormal P2P network traffic to detect the abnormalities of the botnet behavior. The suitable value of the threshold can minimize the false positive rate of P2P botnet activity and increase the detection rate. Therefore, in this paper, the appropriate static value of the threshold will be identified for detecting P2P botnet. The likelihood ratio tests and classification table were two tests from a logistic regression that will be used to access the fitness of the model. Based on the result, the selected threshold manages to detect around 98% of overall detection. This result is supported by the validation process which also manages to detect 98% of overall detection. Such value has confirmed that threshold value is acceptable discrimination to be used in detecting P2P botnet activity.

Keywords: *P2P Botnet Protocols, Threshold, Logistic Regression, Detection*

1. INTRODUCTION

The number of attacks caused by Botnet has been rising every year, thus become a threat to the country. The implication of the botnet attack can result in huge losses to the organization and at the same time may cause a modification of data especially in the banking and government sector. Based on Malaysia Botnet Drones and Malware Infection 2019 reported by Malaysia Cyber Security, as shown in figure 1, the leading with a big number of cases which is 3 261 023 of unique IP detected by Cybersecurity Malaysia [1] is a botnet activity. Therefore, it is necessary to detect botnet behavior especially the P2P botnet where nowadays downloading files or movies from the internet become more popular.

P2P botnet detection can be divided into Signature-based, Anomaly-based, DNS-based and Mining-based detection. In this paper, anomaly-based detection will be employed in the experiment. It is because of the difficulty of discovering the unique communication patterns in the network traffic which do not imitate the signature-based approaches [2]. Furthermore, anomaly detection techniques also able to detect the abnormalities of the network behavior through high network latency, traffic on unfamiliar ports, high volumes of traffic, and abnormal system behavior caused by botnet infection. Besides that, anomaly-based detection possessed the capability to detect botnets and even novel attacks [3]. Although there are many advantages of using anomaly detection, this technique also has a limitation. According to [4], an anomaly-based approach has difficulties in determining the value of threshold due to incomplete profiles from behavior which can lead to the false alarm.

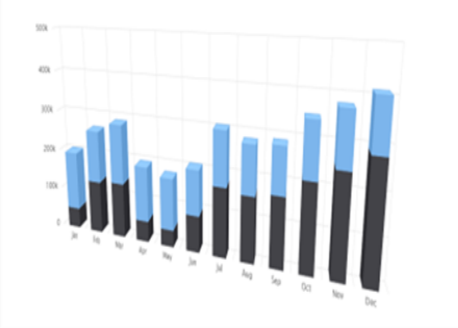

Figure 1: Malaysia Botnet Drones and Malware Infection [1].

As a result, a new technique to identify the value of the threshold is necessary, especially for P2P botnet detection. By selecting the appropriate threshold value, will help to minimize the false positive but the proper method of selection still becomes an issue [5]. This statement is motivated by [5] which stated that the appropriate value of threshold to minimize the false positive still becomes an issue that needs to be solved. Setting an inappropriate value of the threshold will generate a false alarm of the botnet activity. The author [6], claimed that identifying a good threshold can minimize the false positive rate. Hence, a new technique to find the suitable threshold value is required to decrease false alarms produced by the behavior detection for P2P botnet recognition.

The rest of this paper is presented as follows: Part 2 discusses previous studies and part 3 explains the approach used for the research. Part 4 presents some analyses of the findings. Part 5 concludes the article and suggests upcoming work.

2. RELATED WORK

Research in P2P botnet has become more popular nowadays since the major threat caused by their activities. P2P botnet which categorized in the decentralized group of a herd of malware-compromised to work together for an attacker without the knowledge of the owner. The decentralization concept causes the detection of a botnet to become harder for a security expert to translate and understand the access communications compared to previous botnet topology. Figure 2 illustrates the topology of the Command and Control of the P2P botnet [7]

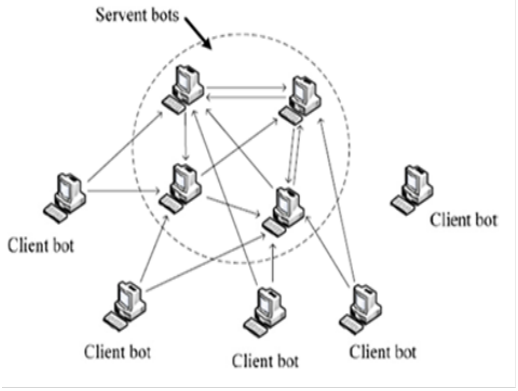

Figure 2: Topology Of P2p Botnet

Every bot opens a connection with each other with their neighbor list which only involved the bot's community IP addresses and connects to every bot through their neighbor list to form a P2P botnet [7]. According to [8,9], P2P botnet granted any bots to be used by a botmaster to do propagation of the commands through peers and collect useful information from them. The dynamic condition of the P2P botnet which peers can change OIDs frequently to get firmly connected to other peers and keep updating search periodically within themselves to find nearby peers. P2P botnets are well known as no central point of failure which has high similarity with normal P2P traffic, making them enigmatic and even difficult to detect [10].

According to [11], the botnet evolves and becomes more complicated and makes it difficult to find, hard to shut down, hard to prosecute and even harder to analyze by reverse engineering. Moreover, the report generated by Spamhaus (2019) [12] in 2019 shows that the number of botnets keeps increasing every year and predicting that there are more challenging threat attack patterns that will emerge with new evasion techniques and characteristics. Although, the P2P Botnets have been on the populate every year, combating and reducing their attacks is still a challenge.

Therefore, this research is to introduce a new technique to detect the P2P botnet by identifying appropriate threshold values in anomaly-based detection in terms of P2P botnet intrusive behavior because the behavior of normal and abnormal behavior of the P2P is more likely the same and difficult to distinguish them. By identifying appropriate threshold values with a systematic technique may be able to distinguish these behaviors and detecting the P2P botnet.

A botnet is a serious disease that infects the internet since the beginning, many previous studies

have examined the botnet problem and found a way how to recognize the pattern in network traffic. A detection model aims to recognize anomalous traffic that indicates the existence of a bot inside the target machine. However, the following section classifies a botnet detection approach into Signature-based, Anomaly-based, DNS-based and Mining-based detection [11].

2.1 Signature-based Detection

A Signature-based approach has been used to identify the machine behavior on the network as viral code. Most Intrusion Detection System is configured with a set of protocols or signatures to capture traffic which perceived to be suspicious [13]. However, a signature-based technique only can be used to detect known botnets [14]. Thus, this solution is not useful for unknown bots [15].

2.2 DNS-based Detection

The monitoring of the DNS becomes easy to detect Botnet anomalies in network traffic. According to [16], bots normally carry out DNS queries to locate and access the C&C server whereby the C &C server is typically hosted by DDNS (Dynamic DNS) provider. DDNS allows rapid and frequent updates of the IP address which allows the C&C server to keep exchanging location and allows the bot to quickly find IP's updates of the host. However, this method is not effective due to the new variants of botnets that have been designed to communicate with a minimum number of DNS queries [17]. Thus, it makes DNS approaches difficult to extract communication on botnets propagation and how they infect the target [18]. Moreover, this kind of approach ineffective to known- DNS based botnets.

2.3 Anomaly-based Detection

Anomaly-based detection has been used to detect the behavior of network traffic latency, amount traffic, unpopular port number and odd system behavior that could indicate the existence of malicious bots in the network [20]. This is approach is different from signature-based systems, which only able to detect attacks bot based on signature has previously been created [21]. This technique is good in identifying new and emerging threats because it will detect any new traffic or abnormal behavior in the network. However, the implementation required time and effort to capture what constitutes normal behavior and susceptible to high rates of false positives.

An anomaly detection system used profile baseline which was created as a baseline system, network or program activity. If there are behavior found deviates compared to the normal indicator, it will be considered as a possible attack [22]. Dorothy E Denning (1987) [23] in the seminar paper expressed the idea that the intruder can be traced to a computer system with the assumption that users of the network or computer system well behave with automatic profiling mode. In other words, the model able to learn traffic behavior, whether good or otherwise. In this paper, the researcher also mentioned several models based on statistics, Markov chains, time series.

Ghafouri et al, (2016) [24] proposed an optimal detection threshold based on anomaly detection to determine the threshold in implementing dynamical systems in the face of a strategic attack. In this paper, the researcher formulates the problem as an attacker defender to determine a threshold to achieve an optimal trade-off between the detection delay and the false-positive rates. To optimize the threshold value, they perform an algorithm that computes the optimal fixed threshold that is independent of time. The results show that the adaptive threshold approach getting a better overall detection delay-false positive trade-off and minimize the losses. Meanwhile, [25] proposed a new technique to identify static threshold values from the derived features in a fast attack on the victim perspective. This new technique shows the properly selected threshold value on the influence feature gives contributes to the detection for IDS to detect anomalies in the network. Moreover, this approach is proved in identifying a fast attack in real-time. Author [26] proposed Network-based statistical anomaly detection which implementing a Statistical anomaly detection technique that calculates an anomaly score for each packet that it sees and forwards the packets to a correlation engine for intrusion detection purposes when a predefined threshold was crossed.

Xiang, Y et al, (2011) [27] used a static threshold to identify the anomaly behavior. The author stated that the attack is detected when the distance between the probability distribution of packet sizes is greater than the value of the threshold. Hajamydee, AI et al, (2016) [28] studies the detection of an intrusion at the host level or network level by using log analysis. The log events have been cluster and use a filtering threshold to decrease the size of events for examination. The result of this shows the filtering threshold significantly impacts the result of identifying the anomalies at the network or host with the rate of detection is about 87.26% and 85.24% of anomalous events. Other

than that, [29] defines a significant static threshold value for botnet identification which can discover the unknown properties of the normal traffic patterns. Based on the threshold selection which is 0.2, the average percentage of correctly recognized bots is moderately large (> 80%). Nevertheless, [30] suggested the method of structural analysis-based learning to categorize between the botnet and benign application. The research used a machine learning method to achieve high detection of accuracy. The result shows the value of the threshold is set to 0.05 as an acceptable value to detect botnet application. Conversely, [31], reported that 0.9 is the optimum value for the threshold to distinguish between benign and botnet. His research has proven that with the value of the threshold, the detection of zero-day fast-flux botnets can be recognized.

In recent times, various alternative techniques have been proposed by the researcher in distinguishing botnet detection. Nonetheless, the method still lacks in distinguish the behaviors of malware and affects the rate of false negatives. Besides, none of the researchers investigate or propose a technique to identify the static threshold as a baseline to distinguish between normal and abnormal behavior. Therefore, a new technique will be introduced in this research to discover a suitable static threshold value to recognize botnet behavior by implementing Logistic Regression.

3. METHODOLOGY

Figure 3 illustrated the testbed environment used for the experiment. The setup consists of used four networked PCs with 4G memory as an experimental platform. They installed Vmware and control programs to PCs and simulate 4 P2P zombie hosts. P2P experiment lab is conducted to capture abnormal activities of P2P Botnet. The testbed setup that has been used by [32] consists of a router, switches and personal computer/peer with a Windows 7-64 bit and Linux OS. Network Time Protocol services are installed together with a script to capture the packet process. A malicious file from P2P Botnet is contributed by CyberSecurity Malaysia. Variants of botnets that are examined at the testbed are cryptocall ,neris,kelihos ,srvcp exe, tnnbtib.exe and kido. Each variant will be executed and been captured 24 hours long. There are software and source use for this research for the testbed such as Tcpdump for capturing packets, Tcptrace for analyzing the data and Ubuntu from Linux based for an operating system. The data collected from the testbed will be used inside the P2P threshold Botnet Detection Module for identifying the suitable threshold value for P2P botnet detection.

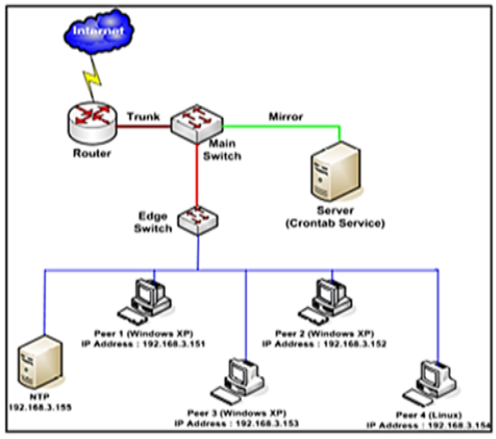

Figure 3: Testbed For P2p Botnet

The methodology of the P2P Botnet Detection Module is based on the methodology proposed by [25] as depicted in figure 4. The module by the previous researcher consists of five main modules to detect fast attack intrusion activity and it is targeted on the Time Based Module and Threshold detection module. The module adapted from the previous researcher is TCPDUMP module, Feature Extraction Module, and Threshold module, all these modules have similar functionality and can be applied. The time-based module has been replaced by the Logistic regression module to suit this study. Feature Selection which applied the data mining technique and Logistic regression module functionality is to find the best fit to determine significant features while the Threshold Detection module is to distinguish between the normal and abnormal traffic in a P2P botnet attack. The Result Module will keep the result of the detection.

3.1 TCPDump

Tcpdump is a tool that is used to sniff and analyze network traffic. It allows users to investigate or troubleshooting an issue caused by network communication. The TCPdump application can read and write real-time network traffic using a libpcap library which capable of capturing network traffic in real-time and saved to a file. libpcap was originally developed by the Network Research Group at Lawrence Berkeley Laboratory. In this study, tcpdump module works to read and interpret the raw network traffic from binary to ASCII form to allow the content of traffic information is

understandable. The tcpdump file will be feed into tcptrace application to generate 89 features for flow analysis. These features will be passed to the feature selection module for identifying the suitable feature for identifying the threshold value (Wan Ahmad Ramzi Wan Yusuf et al,. 2017). The feature selection module has been discussed in detail [33].

3.2 Feature Selection Module

Features selection module functionality is capable to remove features that are redundant or unnecessary that will cause false correlations inside the learning process of the classifiers if the feature is selected. Only the feature that gives a major impact will be selected as an input for the classifier for producing an accurate model. In this Feature Selection Module, the Wrapper model is chosen because of its capability to optimize the classifiers and able to operate with large dimensional data. To determine the top operator for feature selection technique, forward, backward and optimize feature selection will be tested and the highest accuracy of detection will be selected for the next process.

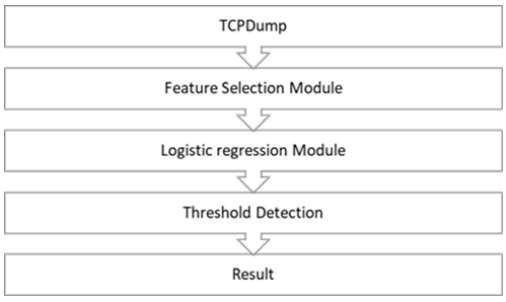

Figure 4: P2P Threshold Botnet Detection Module

Logistic regression Module functionality is to analyze a dataset that has one or more independent parameters to determine the dichotomous outcome in which there is only two possible value 1 (TRUE, Success, Attack) or 0 (FALSE, Failure, normal). It is also a technique to identify the best fitting of model probabilistic systems to predict future events. If the model is fitted, then it can be concluded that the model is proper and good in predicting the outcome variable. Therefore, detection accuracy also becomes higher. There are two approaches applied in this research to review the fitness of the model which are a log-likelihood test and classification table. Both the approach will be discussed in detail below.

3.3.1 Log-Likelihood Test

The log-likelihood test, also known as the likelihood ratio test, is based on the 2LL (deviance). The likelihood ratio tests the model by identifying the difference between -2LL for the full model and the -2LL for the initial chi-square in the null model. The null model is also called the initial model which contains only a constant. The null model is also called a baseline model. The reduction of the value of the -2LL should be less than the value when only a constant is included in the model. The reduction tells that the model is better in predicting the event. Moreover, a well-fitting model is significant at the 0.05 level or better, meaning that the researcher model is significantly good in predicting the event. Therefore the equation of the log-likelihood ratio test is stated in equation (1).

x2 = 2 [Log Likelihood(New(with predictor) – Log Likelihood(Baseline(without predictor)) (1)

3.3.2 Classification Table

The performance of the outcome will be analyzed based on performance evaluation practice. Empirically observing the efficiency of the intrusions detection system, the confusion matrix is being utilized as shown in Figure 5 [34]. From Figure 5, True Positive (TP) represents correctly classify attack examples indicate as 1 (attack); True Negative (TN) represents correctly classify non-malicious examples indicate as 0 (normal); False Positive (FP), which represents misclassify non-malicious examples indicate as 1 and False Negative (FN) represents misclassify attack examples indicate as 0. Using the parameters of the confusion matrix the performance measures are stated as follows:

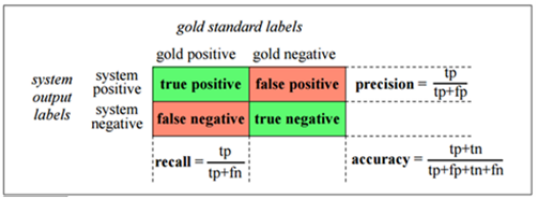

Figure 5: Contingency Table

a) Detection Rate (DR), DR = TP/ (FN+TP)

b) False Positive Rate (FPR), FPR = FP/ (FP+TN)

c) False Negative Rate (FNR), FNR = FN/(FN+TP)

d) Overall Success Rate (OSR), or Accuracy, OSR= (TN+TP)/(TP+FP+TN+FN)

e) The analysis of False Rate (FR) is the summation of FP and FN for each classifier. FR=FP+FN. This module used binary logistic regression to filter all the features selected in the feature selection module and try to find the best significant features with the best accuracy

detection. The significant features selected then will be used in the Threshold Detection Module. The goodness of fit test and classification table has been used to verify the threshold selection for this category.

3.4 Threshold Detection Module

This module is classified as Classification thresholds in which the data from the logistic regression model has been fitted and grouped into two classes 'normal' and 'attack' using a specified or estimated probability of occurrence. It has been most commonly estimated for binary data and is also known as occurrence thresholds [35]. Placing a low threshold value may generate excessive false positives while placing too high, it may cause the system to miss a less aggressive scanner [36]. Therefore, it is important to select a suitable threshold value to detect P2P botnet attack activity. The fitted model from the logistic regression module is used to draw the probability graph using a logistic regression equation to identify abnormal traffic.

3.5 Technique On Threshold Selection

The fitted model from the logistic regression module will be chosen to identify the threshold for each significant feature and the probability graph using logistic regression equation will be computed and constructed to identify suitable thresholds. For the validation of the threshold effectiveness, observation to map with real traffic will be done and explained in the subsection.

3.5.1 Constructing Logistic Function "S" Shape (Sigmoid Curve)

The output from the logistic regression module will be used to construct the logistic function based on equation (2). From the graph, the threshold value is determined by the cut-off value which is determined by the researcher to use 0.8 as a cut-off value. This is support by [37], where the researcher is set threshold values of 0.8 - 0.9 in their experiment to identify malware and produced a good result. From the graph, the threshold value is produced to determine abnormal network traffic.

$$P(Y) = e^{a+bx}/(1+e^{a+bx}) \quad (2)$$

3.5.2 Observation

Once getting the threshold value has been computed from the selected features. The next process is to verify the threshold by mapping to real network traffic by using one sample dataset that obtains 100 records of traffic. Three influence features containing the data will be tested with the threshold value by applying a heuristic approach. Any values that exceed the threshold value is considered abnormal activities while values under the threshold are considered normal activities in the network. It means thresholds are used to indicate whether a host is clean or infected.

4. RESULT AND DISCUSSION

The feature selection phase has selected six features which are: pushed data pkts_a2b, pushed datapkts_b2a,Max_Win_Adv_a2b,Max_Win_Adv_ b2a, Pure_acks_sent_b2a, throughput. Binomial logistic regression is used to test and analyzed all of these features. This test is to explore the most influential feature that will be used to improve the P2P botnet detection [33].

Based on the classification table, pushed_data_pkts_b2a and pure_act_pkts_a2b are two features that influence detecting P2P botnet. Therefore, this feature will be passed to the Threshold Detection Module for recognizing suitable values. Here is the result for feature influence for pushed_data_pkts_b2a and pure_act_pkts_a2b.

4.1 Discovering Feature Influence for Pushed_data_pkts_b2a

Pushed_data_pkts_b2a shows a significant influence on the model in predicting botnet detection. Table 1 shows the summary of Pushed_data_pkts_b2a where The -2LL values have been reduced to 9758.518 from the original which is 11961.505 after we include the feature inside the model. The reduction indicates that the features have a significant influence in predicting the outcome (botnet).

Table1: *Model Summary Pushed_data_pkts_b2a*

-2 Log Likelihood	Nagelkerke R^2
9758.518	0.525

Table 1 also shows the Nagelkerke's value for the new model which is 0.525. The values are near to one which indicates that the feature has a good

influence on detecting the P2P botnet. The feature will be selected as one of the features to detect P2P botnet. A threshold value will be identified to distinguish between normal and abnormal behavior.

Table 2: Model Coefficient Pushed_data_pkts_b2a

Chi-square	Df	Sig.
2202.986	1	0.000

To make further verification, Chi-square (x2) test has been used to verify whether the feature is significant or not. Based on Table 2, The feature is significant and this is proven by investigating the p-value which is highly significant at 0.05 and 0.001 levels. Thus, Pushed_data_pkts_b2a gives a good effect to the model in predicting the outcome.

4.2 Discovering Feature Influence for Feature pure_act_pkts_a2b.

Pure_act_pkts_a2b shows a significant influence on the model in predicting botnet detection. Table 3 shows the summary of pure_act_pkts_a2b where the -2LL values have been reduced to 8201.782 from the original which is 9758.518 after we include the feature inside the model. The reduction indicates that the features have a significant influence in predicting the outcome (botnet).

Table 3: Model Summary pure_act_pkts_a2b

-2 Log Likelihood	Nagelkerke R^2
8201.782	0.608

Table 3 depicted the Nagelkerke's value for the new model which is 0.608. The values are near to one which indicates that the feature has a good influence on detecting the P2P botnet. The feature will be selected as one of the features to detect P2P botnet. Threshold value will be identified to distinguish between normal and abnormal behavior

Table 4: Model Coefficient pure_act_pkts_a2b

Chi-square	Df	Sig.
1556.736	1	0.000

Chi-square (x2) test has been used to verify whether the feature is significant or not. Based on Table 4, The feature is significant and this is proven by investigating the p-value which is highly significant at 0.05 and 0.001 levels. Thus, pure_act_pkts_a2b gives a good effect to the model in predicting the outcome. A threshold value will be identified to distinguish between normal and abnormal behavior.

4.3 Threshold Identification

The threshold identification was based on the estimated probability of the logistic model. The estimated probability of a logistic model can build threshold identification. The regression equation of the model is based on the equation:

$$P = (Y) = \frac{e^{\beta_1+\beta_0 X}}{1 + e^{\beta 1+ \beta o\, x}} \quad (1)$$

Where:

P(Y) indicates the probability of attack success.

β_0 is the constant of the equation

β_1 indicate the slope of the logistic function

From the equation logistic regression module, logistic function or sigmoid curve is constructed using Wolfram Mathematica 10.4. From the graph, the threshold value is determined by the cut-off value which has been set by the researcher to use 0.8 as a cut-off value. This is support by [37], who stated that by setting up a threshold value of 0.8 - 0.9 in the experimental, it succeeds to identify malware and produced a good result. From the graph, the threshold value is produced to determine abnormal network traffic.

4.3.1 Threshold for Pushed_data_pkts_b2a Feature

Figure 6 shows influence features, Pushed_data_pkts_b2a that were produced by a fitted logistic regression equation to identify threshold as a baseline to differentiate normal and abnormal traffic in the network. The logistic regression equation that computed the threshold are:

$$P(Y) = \frac{e^{-3.763+4.339x}}{1 + e^{-3.763+4.339x}}$$

The threshold for this feature was identified when the probability of 80% was applied to the model. The value of the cut-off was 1.259 and hence the selection of 2 packets per second made by a single host can be considered as an attack, while if the value falls under threshold value is considered normal.

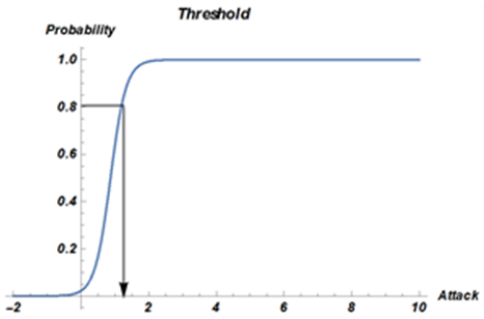

Figure 6: Threshold for Pushed_data_pkts_b2a

4.3.2 Threshold for pure_act_pkts_a2b Feature

Figure 7 shows the graph generated from fitted logistic regression for the pure_act_pkts_a2b feature. Cut off the value of 0.8 of the threshold was 1.337, as a decimal value does not reflect the packet, then the threshold value is rounded to 2. Thus, the value of 2 packets per second is considered as attack traffic inside the real-time network. The logistic equation which generates the graph are:

$$P(Y) = \frac{e^{-3.763+3.763x}}{1+e^{-3.763+3.763x}}$$

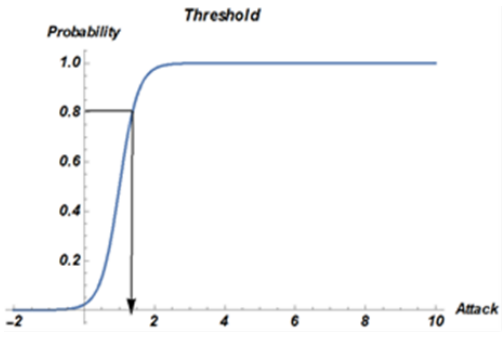

Figure 7: Threshold for Pushed_data_pkts_b2a

4.3.3 Threshold Identification of the Overall Feature

Figure 8 shows the graph of an overall threshold for the selected feature which is Pushed_data_pkts_b2a and pure_act_pkts_a2b. The 'sigmoid curve' or logistic function is constructed by the value given from the logistic regression model. The cut-off value of 0.8 determines the threshold value of 3.044, as a packet cannot be assumed in a decimal number, then the threshold value is rounded to 4. Thus, the packet exceeds the baseline of the threshold by a single host and is considered an attack activity. The logistic regression equation that computed the threshold are:

$$P(Y) = -3.763+(0.794y)+(0.931z)$$

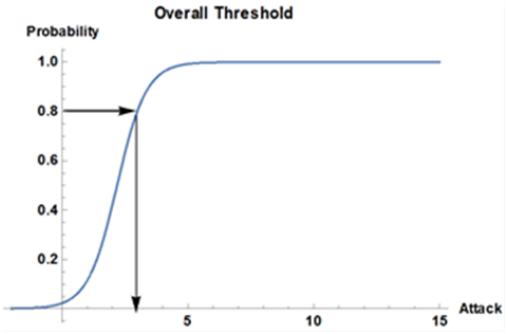

Figure 8: Threshold for the overall feature

4.4 Result Validation

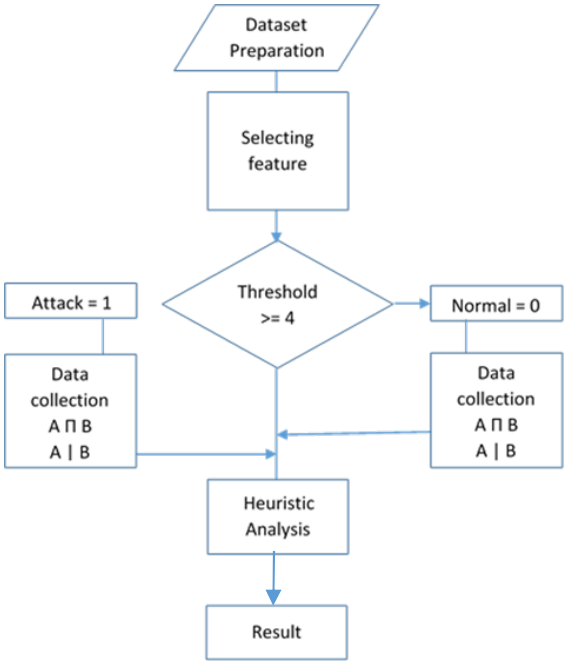

Figure 9: Result Validation Process

The validation phase is a process to verify whether the threshold selected from the network traffic flow that is exceeded the value is considered as an attack while values below the threshold value are considered as normal. The validation process is depicted in Figure 9. The mapping to the actual network traffic with the baseline of threshold value obtained from the Threshold for overall feature influenced will produce the result which to prove that it may contribute to the P2P botnet detection.

Figure 9 shows the flowchart process for the result validation process. The test is to verify the baseline of threshold value will distinguish normal and attack traffic in the network. A sample dataset with 100 records of traffic consisting of normal and abnormal traffic obtained from the real traffic is used for testing. The dataset in the form of spreadsheets then will select two influenced features which are Pushed_data_pkts_b2a represented by 'A' and pure_act_pkts_a2b represented by 'B'. The next process is to compare each feature with the Overall Threshold value derived by using the logit formula:

$$P(Y) = -3.763 + (0.794y) + (0.931z)$$

The threshold value will be manually compared to data provided in the dataset and the result will be organized according to two combinations of two influence features. The threshold value will be manually compared to data provided in the dataset and the result will be organized according to two combinations of two influence features involves 'AND' process and 'OR' process comparison as follow:

Table 5: 2^2 Combination Features 'AND' and 'OR

2^2 Combination		
A ∏ B		A |B
B ∏ A		B | A

The comparison process is based on a heuristic approach which is a set of general rules of thumb, that is used to critique, score or radar diagrams to visualize weaknesses in the system which end up with overall usability score. Any values that exceed the threshold value is considered abnormal activities while values under the threshold are considered normal activities in the network. It means Thresholds are used to indicate whether a host is clean or infected.

Table 6 shows the result of real traffic mapping by using a heuristic approach. The result shows all combination produces high accuracy with a range 97% to 98% of overall detection rate which is not much different with the Full Model overall detection rate 98.3%. To conclude, the equation model presented in this section may contribute to the detection system and can be used to detect P2P botnet.

Table 6: Result Validation

A ∏ B				A |B			
Normal	38	2	98%	Normal	37	3	97%
Botnet	0	60		Botnet	0	60	
B ∏ A				**B | A**			
Normal	38	2	98%	Normal	37	3	97%
Botnet	0	60		Botnet	0	60	

Selecting a significant feature and an appropriate threshold value need to be selected with a proper procedure or technique. Without adequate data and significant features, it will affect the result of the threshold value in distinguish P2P botnet activity. For instance, the value of the threshold for previous research cannot be used in this research as the study is not focusing on P2P botnet detection. It is because different malware may have different behavior. The value of the threshold should minimize the rate of false alarm and consequently, the rate of accuracy will be increased with the improved false alarm rate. Nowadays, the type of botnet attack has undergone significant changes and difficult to be identified as the P2P botnets hide their communication through P2P traffic. This remains an open challenge in the research community. Moreover, constraints on botnet detection (i.e. cannot differentiate and recognize the new botnet activity precisely) need to be improved.

5. RESEARCH SIGNIFICANT

Before selecting an appropriate threshold value, identify the significant feature that is necessary. The proposed feature has been revealed in [33]. The researcher [24] introduces the fixed threshold identification without considering the feature for the detection. Furthermore, the algorithm used by [24] is time-consuming to be generated. Author [25] also used logistic regression in detecting malware activities but more focused on fast attack detection. The behavior of fast attack is different compare to P2P botnet. Furthermore, the author does not consider flow analysis in determining the threshold

value as we have done in [33]. [28] used unsupervised learning to generate threshold values. The threshold was selected based on the number of occurrences of the event. This may produce a false alarm where the number of the event will keep changes based on the behavior of the P2P botnet. Therefore introducing a new technique in selecting a proper threshold value for P2P botnet and considering a proper strategy such as flow analysis, feature identification is needed to reduce the false alarm.

6. OPEN RESEARCH ISSUE

Introducing a technique in selecting a threshold value is still open research. This is due to the nature of the P2P botnet behavior that keeps changing the pattern of attack. Although, there is a researcher who has proposed a value for the threshold this cannot be used because the research concentration is different. P2P botnet behavior is different from other malware behavior. Therefore identifying a new threshold with proper strategies and technique is necessary. This may remain open research to the research community. Moreover, constrain on P2P botnet detection to differentiate and recognize their new behavior need to be improved.

7. CONCLUSION

In a conclusion, this research has introduced a new technique to identify the threshold value with higher accuracy. This threshold value can be used to detect the P2P botnet activity within the variant used in this research. Further improvement by developing new techniques is necessary especially when the P2P botnet will evolve frequently. Besides that introducing a dynamic threshold value is a good approach to tackle a new variant of a botnet where the behavior keeps changing over time. Since this study only focusing on TCP protocol, the intention to apply other protocols such as UDP is also recommended. Future research also can be considered the IDS log, as an input to determine a suitable threshold that will contribute to the accurate detection of IDS to differentiate normal and abnormal activities in the network.

8. ACKNOWLEDGEMENT

This work has been supported under Universiti Teknikal Malaysia Melaka. The authors would like to thank the Center for Research and Innovation Management, Faculty of Information and Communications Technology and Universiti Teknikal Malaysia Melaka and all members of INSFORNET research group for their incredible supports in this project.

REFERENCES:

[1] MyCert, “Malaysia Botnet Drones and Malware Infection”, Available at https:// https://www.mycert.org.my/portal/statistics, Last accessed on February 2, 2019

[2] Karim A, Salleh RB, Shiraz M, Shah SA, Awan I, Anuar NB, (2014), “Botnet detection techniques: review, future trends, and issues”, Journal of Zhejiang University Science C, 11, pp. 943-983

[3] Abdullah, R.S., Abdollah, M.F., Noh, Z.A.M., Mas’ud, M.Z., Selamat, S.R. and Yusof, R., (2013), “Revealing the criterion on botnet detection technique”, IJCSI International Journal of Computer Science Issue Vol.10, Issue 2, pp. 208-215.

[4] Jing-xin W, Zhi-ying W, Kui D., (2004), “A network intrusion detection system based on the artificial neural networks”, In Proceedings of the 3rd International Conference on Information Security, 14-16 November, Shanghai, China: ACM, pp. 166-170.

[5] Derrick EJ, Tibbs RW, Reynolds LL, (2007), “Investigating new approaches to data collection, management and analysis for network intrusion detection”, In Proceedings of the 45th Annual Southeast Regional Conference, 23-24 March, Winston-Salem, North Carolina: ACM, pp. 283-287.

[6] Fredrikson, M., Jha, S., Christodorescu, M., Sailer, R. and Yan, X., (2013), “Synthesizing nearoptimal malware specifications from suspicious behaviors”, In: Malicious and Unwanted Software:" The Americas"(MALWARE), 2013 8th International Conference, 22-24 Oct. 2013 Fajardo, PR, USA: IEEE. pp. 45-60.

[7] Han, Q., Yu, W., Zhang, Y., & Zhao, Z. (2014). “Modeling and evaluating of typical advanced peer-to-peer botnet”. Performance Evaluation, Volume 72, pp. 1–15.

[8] J.Zhang, R.Perdisi, W.Lee, X. L. and U. S. et al. (2015). “Building a Scalable System for Stealthy Peer to Peer Botnet Detection”, Vol. 2, pp. 6–10.

[9] Alauthaman, M., Aslam, N., Zhang, L., Alasem, R., & Hossain, M. A. (2018). “A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks”. Neural

computing & applications, Volume 29 issue 11, pp.991–1004.

[10] Obeidat, Atef. (2017). A Novel Botnet Detection System for P2P Networks. Journal of Computer Science. Volume 13 issue 8, pp. 329-336.

[11] Ramesh, S.R, Emmanuel S. P and R. C. Joshi, (2018), "Survey of Peer-to-Peer Botnets and Detection Frameworks". International Journal of Network Security, Vol.20, No.3, pp.547-557

[12] "Spamhaus Botnet Threat Report 2019", Available at https:// https://www.spamhaus.org/news/article/793/spamhaus-botnet-threat-report-2019. Last accessed on February 2, 2019.

[13] Sharma, N., Patnaik, T., & Kumar, B. (2013). "Recognition for Handwritten English Letters: A Review". International Journal of Engineering and Innovative Technology (IJEIT), Volume 2 issue 7, pp. 318–321.

[14] P. Panimalar and K. Rameshkumar, (2014) "A review on taxonomy of botnet detection," 2014 International Conference on Advances in Engineering and Technology (ICAET), Nagapattinam, 2014, pp. 1-4.

[15] Feily, Maryam & Shahrestani, Alireza & Ramadass, Sureswaran. (2009). A Survey of Botnet and Botnet Detection. The Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009, 18-23 June 2009, pp. 268-273.

[16] R. Villamarin-Salomon and J. C. Brustoloni, (2008), "Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic," 2008 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, 2008, pp. 476-481.

[17] M. Eslahi, R. Salleh and N. B. Anuar, (2012) "Bots and botnets: An overview of characteristics, detection and challenges," 2012 IEEE International Conference on Control System, Computing and Engineering, Penang, 2012, pp. 349-354

[18] Bailey, M., Cooke, E., Jahanian, F., Xu, Y., & Karir, M. (2009). "Survey of botnet technology and defenses". Proceedings - Cybersecurity Applications and Technology Conference for Homeland Security, CATCH 2009, 299–304.

[19] Alieyan, Kamal & Almomani, Dr.Ammar & Manasrah, Ahmad & Kadhum, Mohammad. (2015). "A survey of botnet detection based on DNS". Neural Computing and Applications.

[20] Baruah, Sangita, (2019), "Botnet Detection: Analysis of Various Techniques". International Journal of Computational Intelligence & IoT, Vol. 2, No. 2.

[21] Jwala Sharma, Samarjeet Borah, (2019), "Botnet Detection Techniques – An Analysis", International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue-1, pp. 2130-2137.

[22] Xiaodan Xu, Huawen Liu, and Minghai Yao, (2019), "Recent Progress of Anomaly Detection", Complexity, Volume 9, pp.1-11

[23] Dorothy E. Denning, (1987), "An Intrusion-Detection Model" Ieee Transactions On Software Engineering, Vol. Se-13, No. 2, February 1987, pp. 222-232.

[24] Ghafouri A., Abbas W., Laszka A., Vorobeychik Y., Koutsoukos X. (2016), "Optimal Thresholds for Anomaly-Based Intrusion Detection in Dynamical Environments". Decision and Game Theory for Security. GameSec 2016. Lecture Notes in Computer Science, vol 9996.

[25] Abdollah, M. F. and Mas'ud, M. Z. and Yusof, R. and Selamat, S. R. (2010), "Statistical Approach for Validating Static Threshold in Fast Attack Detection". Journal of Advanced Manufacturing Technology, 4 (1). pp. 53-71.

[26] S. Staniford, J. Hoagland, and J. McAlerney,(2002) "Practical automated detection of stealthy portscans," J. Comput. Secur. Volume 10.

[27] Xiang, Y., Li, K., Zhou, W. (2011), "Low-rate DDoS attacks detection and trace back by using new information metrics", IEEE Transactions on Information Forensics and Security, Vol. 6, Issue 2, pp. 426-37

[28] Hajamydeen, AI., Udzir, NI., Mahmod, R. and GHANI, AAA. (2016) "An unsupervised heterogeneous log-based framework for anomaly detection", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 24, No. 3, pp. 1117-1134.

[29] Mulay, P. (2016) "Threshold computation to discover cluster structure: a new approach", International Journal of Electrical and Computer Engineering, Vol. 6, No.1, pp.275-282.

[30] Kirubavathi, G. and Anitha, R. (2018) "Structural analysis and detection of android botnets using machine learning techniques", International Journal of Information Security, Vol. 17, No. 2, pp.153-167.

[31] Almomani, A. (2018), "Fast-flux hunter: a system for filtering online fast-flux botnet", Neural Computing and Applications, Vol. 29, No. 7, pp.483-493.

[32] R. S. Abdullah, M. F. Abdollah, Z. A. M. Noh, M. Z. Mas'ud, S. Sahib and R. Yusof, (2013), "Preliminary study of host and network-based analysis on P2P Botnet detection," International Conference on Technology, Informatics, Management, Engineering and Environment, Bandung, 2013, pp. 105-109.

[33] Wan Ahmad Ramzi Wan Yusuf, Faizal M. A, Rudy Fadhlee M. D, Nur Hidayah M. S, (2017), "Revealing Influenced Selected Feature for P2P Botnet Detection", International Journal of Communication Networks and Information Security (IJCNIS), Volume 9, No 3.

[34] Daniel Jurafsky & James H. Marti, (2019), "Naive Bayes and Sentiment Classification", Speech and Language Processing"

[35] Toms, J. D., and M.-A. Villard. (2015), "Threshold detection: matching statistical methodology to ecological questions and conservation planning objectives". Avian Conservation and Ecology 10(1).

[36] Jaeyeon Jung, V. Paxson, A. W. Berger and H. Balakrishnan, (2004), "Fast portscan detection using sequential hypothesis testing," IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004, Berkeley, CA, USA, 2004, pp. 211-225.

[37] Shankarapani, M.K., Ramamoorthy, S., Movva, R.S. et al. (2011), "Malware detection using assembly and API call sequences". J Comput Virol 7, 107–119 (2011)