National College of Ireland

An approach for mitigating botnet attack on a large network.

MSc Internship

Cyber Security

Hope Micah Ayuba

Student ID: x19134771

School of Computing

National College of Ireland

Supervisor: Ross Spelman

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name: …………Hope Micah Ayuba……………………………………………………………

Student ID: …………x19134771………………………………………………………………….

Programme: ………MSc Cyber Security…… **Year:** ..2020……..……..

Module: …………MSc Internship……..……………………..………….……..…..

Supervisor: …………Ross Seplman……………………………………..………………..……

Submission Due Date: … ………17/08/2020…..………..………………………………………..…..…

Project Title: An approach for mitigating botnet attack on a large network.

Word Count: ……………4431………… **Page Count:** ……………………14……………………………………..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ……………………………………………………………………………………………

Date: ………………28/09/202……………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An approach for mitigating botnet attack on a large network.

Hope Micah Ayuba
X19134771
MSc in Cyber Security

Abstract

Botnet attacks and the various techniques of propagation has constantly been a tricky challenge for organizations to control. These attacks usually involve compromised computers and all categories of mischievous actions to cause colossal damage and loss of resources from the victim. There is a need to expose the botnet frequent methods of dissemination by implement machine learning algorithms. This research uses artificial neural networks, logistic regression, and decision tree to develop a server-based botnet detection system that maintains accuracy of 99.90%. The system detects bot/botnet that uses IRC, HTTP, and the P2P protocols by analyzing their data flows and then distinguishes their behavioural patterns on the network. Compare to other papers, this research measures performance using Accuracy, True Positive, False Negative Rate, and Precision. We got the dataset from the Stratosphere datasets repository. The dataset was netted at the Czech Technical University, Prague and these botnet samples comprise dissimilar various communication protocols and achieved different activities. Similarly, the study adheres to rigid data input that does not meet the required data within the trained botnet traffic.

Keywords: *Botnet, Detection, Network, Flow, Client, Server, Machine learning.*

1. Introduction

One of the major security challenge facing the network environment nowadays is the infection of computers by malicious programs that permits the formation of a botnet. A botnet can be described as a compromised pool of connected bots (computers), remotely controlled under the command of a botmaster. Botnet attacks to organizations are for various reasons including stealing sensitive information, the spread of spam, denial of service, or damaging IT resources [1]. To successfully perform an attack, the botheaders (botmaster) set up a communication channel to send commands to the bots and receive feedback from them. The key difference between botnet and other mischievous attacks is the command and control channel framework. Compare to other malware attacks that are used to cause havoc on the network, a botnet grows as collecting of compromised computers reliant on the command and control channel. Botnet depends on communication protocols. Botheaders normally uses the centralized (IRC, HTTP) command & control channels, or a decentralized (P2P) command & control channel to disseminate bots [2].

The earliest botnet utilizes a channel called Internet Relay Chat (IRC) that functions as a communication medium between parties on the network. Internet Relay Chat protocol was developed by Jarko Oikarinen at Oulu University Finland to replace a platform named Multi-User Talker in 1988. The design of IRC has gained remarkable recognition in communication and networking over the years and has been persistently compromised (attacked) by many hackers because of the flexible nature [3]. Usually what the bot does is to perform a scan on a system that the security configuration is weak within a networking environment. When this scanned system is fully compromised it means that the botheader (botmaster) is in total command of the open channel. The command and control channel that is waiting to be controlled by botmasters can execute over 200 commands at a go. The botheaders have adopted command and control channel tactics to commit mischievous activities using the Hypertext Transfer Protocol. HTTP botnet is like the IRC botnet with some behavioural characteristics like DNS fast-flux. This type of botnet is known for propagating DDoS attacks, malware spreading, sending unsolicited messages (Spamming), and fraud on the internet. Denial of Service was a usual attack by HTTP botnet according to [4]. According to [5] in their 2017 report, they showed that the numerical figure of malware infection caused by HTTP botnet increased massively. To date, botheaders have improved in many attack tactics and using a various network protocols to perform their enterprise. P2P botnet is another protocol that uses the P2P mechanism to perform a decentralized attack on a network. Though, Peer-to-Peer botnet has the problem of organizing bots for decentralized network architecture. A single point of failure is the issue with P2P botnets. But the nowadays P2P botnet strengthened by operating in a network concurrently, both as a server and as a client [6, 7]. The various fashion of botnet in the networking environment pose a threat and should be a battle by network operators in whatever ways or tactics it present itself [8] [3]. The botheader uses bots through the network server or client computers in recruiting zombies and subsequently force the computers into botnet. If the botheader activities on the attacked server is not curtailed, it can cause distributed denial of service using affected computers on the network stealing sensitive data or hijacking of IT resources for ransom [2] [9]. Motivation. The threat on normal Port 80 an HTTP protocol used by the regular user to access websites, vulnerable nature of IRC protocol, and that botheaders are using some communication protocols to remotely attack servers and client's on a network, causing loss and damage of valuable resources. Hence, we picked up a challenge to develop a botnet detection system to uncovering bots/botnets within the network and curtail future threats. In this paper, we develop a detection system that can classify and detect bots/botnet data flow based on machine learning models. This model will create a server-based bot/botnet detection system that can pragmatically detect in real-time botnet attack. The following are the objectives of this research:

- To properly analyze network flow and protect the network from bot/botnet attack.
- To capture bot that do not meet the required network parameter within the trained botnet traffic.
- To develop a machine learning detection system that detect IRC, HTTP, and P2P botnet on a network.

Research question: How can botnet attacks on a large network be mitigated using machine learning models?

The structure of this paper is as follows; Section I provide information on the background motivation, contribution, research question and objectives. Section II discussed related work. Section III present research methodology and specification of the proposed system. Section IV implementation. Section V evaluation. Section VI discussion and conclusion.

2. Related Work

The subject of botnet detection has been the area of research across various researchers. Many related studies emphases on the use of machine learning models in the botnet detection system. Many of these systems stressed merely on particular botnet detection channels. Some of these botnets detection systems have shortcomings such that they are incapable to detect improved bots, and some achieved not more 99.7% accuracy rate. Conversely, here are some previous detection models we reviewed.

2.1 IRC Botnet Detection

In [10], the researchers proposed a botnet detection model that is focused on metrics based flow analysis. In other to determine between fake and real IRC channels. They perform an IRC cleaning routine and execute flow based techniques on traffic. [11 and 12], developed a mechanism that joined the network and application layer analysis for botnet detection. In [13], the model used the application layer analysis to compare and validate the IRC channel activity. Other authors also used machine learning to develop botnet detection mechanisms, joining the effort in the fight against botnet. The claimed that machine learning has the capability of describing bots/botnets, provided a suitable set of representative components are considered in the selection process. The eventual result of the average detection rate is 95% [14]. The authors of [15], proposed a multidimensional model that detects botnet by analyzing network behaviour using distributed monitoring. [16], proposed a model by blacklisting and monitoring DNS activities related to a botnet. The Internet Relay Chat bot has some detection mechanisms that operators used to fight botnet and the botheader activities on the network [13]. According to [1 and 17], detection mechanisms like Domain Name System (DNS) group traffic activities and Intrusion Detection System (IDS) which centres on a single point host (IRC) design. The authors in [30] proposed a randomly determined process model, this model isolates command and control botnet communications from human communications using an Empirical Test. They called the system BotProbe with the moderately desired accuracy. According to them, the IRC based botnet has about 53% of the botnet command detected amongst huge numbers of real-world botnets, and the downloaded botnets are about 14.4%.

2.2 HTTP Botnet Detection

They carried out an HTTP bot survey specifically on botnet detection which according to [18], the survey article claimed that HTTP bot utilizes Transport Control Protocol (TCP) as the major fulcrum of its dissemination but mainly on HTTP bot. Conversely, [19] performed pragmatic research pertaining to botnet attack on a large network also known as botheaders. [20], said that botnet can also be detected by supervised learning mechanisms such as regression and classification or by clustering such as unsupervised learning mechanisms. They express this mechanism in three dimensions. A decision tree with three most popular clustering algorithms in a hybrid prototype to classify mischievous flows. From the above mechanism by [20], saw a formula of $DR = TP/TP+FN$ where, TP the total number of properly-identified mischievous flow, FN denote the number of improperly identified mischievous flow and DR means the detection rate. After an in-depth check, they claimed that K-Means can help recognize mischievous flow. In [21 and 22], they proposed a system that can intermittently detect communication in network data using a model named Degree of Periodic Repeatability. They described repeatability as a group of unique activities (i.e. clients of the same origin are traceable to an HTTP server) that is detected periodically within a defined time window by related intervals.

In the first part of their study, they planned a method that enumerates and calculates the time interval between the same origin HTTP connections that fall within a specific time. In the second part, to determine whether the time intervals were near each bot generated by the specific network activities, they calculated the standard deviation of assumed time variances. Though, botheaders can penetrate this model and create false-negative results by randomly changing the configuration of connection intervals [23]. [24], in an international conference paper titled Management and Security in the Age of Hyper-connectivity Germany 2016, the researchers claimed that Self-Organizing Network has contributed to the dissemination of various malicious threat like botnet on the network and equally helped unravel the loophole, and to build a model called Network Element Virtual Temperature that permit the firmness in the network element. Similarly, [25] developed a prototype in mobile botnet known as DeDroid detection technique, the technique allows permission on API calls and depends only on Static code analysis. Compared to dynamic code analysis, the Static code analysis utilizes a lightweight model.

2.3 Peer2Peer Botnet Detection

The network features traced to the Flow-based technique remain the same as seen in NetFlow attributes like the bytes-per-flow, bytes-per-second, and bytes-per-packet. In the last decade, there has been tremendous work on botnet detection approaches that use flow analysis. There are several network traffic flow detection approaches that have been proposed in recent times. We will discuss some approaches and their shortcomings in this section. Anomaly detection is a container for mining-based detection techniques it can extract which network traffic patterns unexpected and detect abnormal traffic including advance encrypted packets [26, 27, 28, and 29]. In [26], they proposed a comprehensive botnet detection prototype that can detect several botnets. This model analyzes the network traffic flow intermittently. Then, building an effective system classification that is executed using a statistical correlation in the network traffic flow. Irrespective of the various protocols used, this model can only identify and detect botnet averagely 99.5% accuracy rate. In [27], they proposed a Peer2Peer botnet detection model that uses a mining scheme that centred on network traffic monitoring and analyzing. They did their final assessment and outcome by the use of three data mining popular algorithms namely: Bayesian networks, Naïve Bayes, and J48. This algorithm claimed accuracy of 87%, 89%, and 98% respectively. [28], proposed a novel model for Peer2Peer to detect botnet by analyzing network traffic. Their concept involved picking twelve features from the network flow to extract flow behaviour and to enable them to analyze some pre-set time windows. In this model, they separated genuine network traffic from botnet traffic using the machine learning algorithm. They used the reduced error pruning algorithm for the selection of the decision tree and then; they selected the core discriminating feature by using the correlation features evaluator for bot/botnet detection. It can detect both the connected botnets and those not connected using their model. Similarly, their model claimed to detect bots early activity via the control center phase and detect unfamiliar bots. According to [29], they used machine learning to develop a botnet detection model. The system focused on network traffic. They base their view on extracted and flow-based characteristics from the network traffic. They included inter-arrival time, flow characteristics, and some noise to the payload for the model to achieve its target. The model claimed about 99.7% accuracy and demonstrates that it can fight more noise compared to other detection models according to the Computer Networks and Communications Journal. Zeus-botnet [31] and [32], developed their own special system of command and control design. This system exhibits a different behaviour of capturing and analyzing network flow features.

Also, it is possible that the system may fairly match the botnet behavioural patterns. Take, for instance, as with P2P botnet where botheaders normally use scripts that execute when precise events happen without human input like new bot joining the zombies. Under this scenario, our proposed system has in it the abilities to identify specific network flow features that can differentiate between botnets flow and other network flow.

To avoid vagueness, we prudently studied the several established models used in botnet detection. However, the most substantial thing in the studied models is the fact that the key principle behind all the models is to detect bot in system on a single, double, or combine methods used in having access. Another insight gained in the cause of this study is that the principal goal of botheaders is to attack network infrastructure (i.e. by stealing or takeover) for selfish interest. Though the strategy of spreading or dissemination varies. The persistence tactics of dissemination by botheaders on network servers suggest this study, which is assumed to be the most pre-emptive and effective botnet detection system which is a little above the reviewed models. We see this from the exclusive attributes of bot/botnet detection machine learning models with the systematic method of combining other known channels of botnet propagation (IRC, HTTP, and P2P) to a single system.

3. Methodology

In section III, we provide detailed clarification to substantiate the choice of research method, design, and implementation. We developed the proposed system in Python programming language version 3.7 and we used Sklearn library which features several machine classification algorithms that we need. This study implements some machine learning algorithms like; artificial neural networks, logistic regression, and decision tree to enable us to classify and categorize normal and infected data flow on a network.

3.1 Machine learning

There are three key categorizations of Machine learning: supervised, semi-supervised, and unsupervised learning. In a supervised learning approach, the system learns the input data. The supervised approach train, analyze, and labeled every input data and uses the output investigation to better the performance of its inputs. Unsupervised learning approach executes the description of unlabeled and hidden features. The data used is unlabeled, because of that the system can only be described and summarized the significant features of the input without evaluation of the system's accuracy. Semi-supervised learning is a hybrid approach. It is the joining of supervised and unsupervised learning. In pattern recognition, semi-supervised learning uses huge numbers of unlabeled input and this can decrease high accuracy [34].

3.2 Algorithms

In this study, we developed a server-based system using machine learning models which depend on the support of both supervised and unsupervised classifier algorithms. Our goal is the system performance which is why we selected the three classifiers above to enable us to evaluate the performance metrics such as accuracy in the classification of infected and normal data and also to benchmark the results of each classifier against existing researches using CTU dataset.

3.3 Data collection and Experimental Setup

We setup a simulated network test environment on a VMware for experiments. This test simulated environment targets to obtain mischievous and normal network flow, and also to construct dataset for IRC, HTTP, and P2P botnet detection investigation. The virtual network architecture is made up of five infected computers, network server, network router, and botmaster command and control server. In this study CTU labeled dataset have been used and it consist of different botnets samples which was used for this experiment specifically Neris, Rbot, Virut, Menti, Sogou, Murlo, and NSIS.ay. While the normal network flow, we carry out web browsing activities to get our normal flow, while for the mischievous network flow binary bot was executed into the computers in our test environment. We used Windump tool on the server to collect both incoming and outgoing network flow. The network flows of IRC, HTTP, and P2P is captured by Windump and stored for further analysis. The CTU dataset by [35], comprises of thirteen various captured scenarios of dissimilar botnet attacks which include denial of Service, pay per click attack, port scan, fast flux, spam traffic, and blockage of normal conversation to escalate server workload. The architecture of the network test environment is shown below.

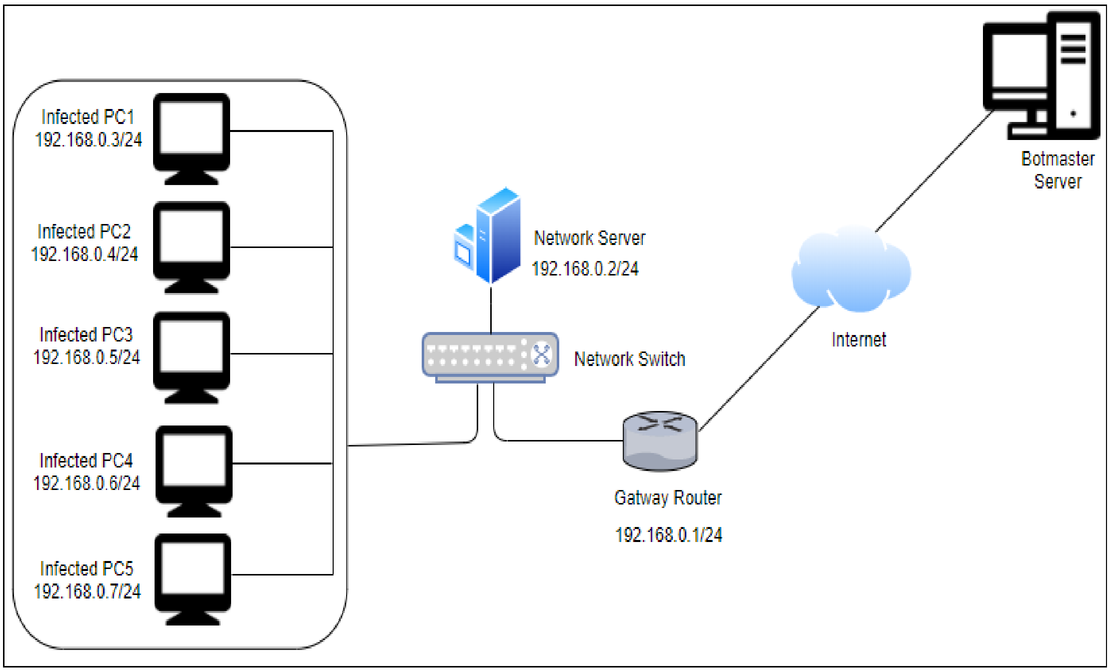

Figure 1. Network Architecture for Botnet Test Environment

3.3.1 Data Pre-Processing

In data pre-processing, both the normal and mischievous network flow is mined to traffic log parser and stored into packet capture (pcap) file format using Windump tool. We merged the normal and mischievous logs labeled with 1 for normal flow and 0 for malicious network flow. To decrease error, unnecessary noise in the result, and to curtail inaccuracy in classification, we merged both normal and mischievous network flow and performed data cleaning which is done manually [35]. In the data cleaning process, we disregarded some features like the source and destination IPs because of inefficiency in some general botnet detection systems.

3.3.2 Machine Learning Classification

We used a data modeling tool called RStudio to run the labeled data through the classifier algorithms. Artificial neural networks, logistic regression, and decision tree algorithm was used to build our system. The contribution of each classifier algorithm is to improve the general performance of our system using the same CTU-13 dataset which several researchers have used for the botnet detection system. Table 1 below described the three selected most effective classifier algorithms used in this study.

Classifier Algorithms	Description
Artificial Neural Networks	Artificial Neural Network is a concept that is centered on a group of connected nodes known as artificial neurons, which loosely depicted the animal brain [36]. The capability of the network to function and process properly relies on the inter-node connection strengths, or weights, gotten by learning from a set of training patterns. The following equation describes the linear combination function: $$\mathrm{net} = \mathrm{w1} * \mathrm{x1} + \mathrm{w2} * \mathrm{x2} + \ldots + \mathrm{wD} * \mathrm{xD} = \sum_{j=1}^{D} \mathrm{wjxj} \ = \ \mathrm{wT} \ * \ \mathrm{x},$$ Where wj is a weight linked with the input (xj). This weight shows the intensity where a specific input value influences the output value. The calculated value (net) is applied in an activation function that can be Linear, Steps, Hyperbolic Tangent, Ramp, and Sigmoid. [37] The Artificial Neural Network technique can classify non-linearly independent nodes [38], and this is suitable for our botnet detection system.
Logistic Regression	Logistic Regression is a statistical approach that is used to predict the likelihood of target variables. It has been used in the development of several detection mechanism such spam and denial of service detection technique and it has been proven to be a viable technique for botnet detection. When calculation the likelihood of target variable it uses a link function described by the following equation: $$\pi(x) = \frac{\mathrm{e}(\beta0+\beta1x1+\beta2x2+\ldots\beta ixi)}{1 + \mathrm{e}(\beta0+\beta1x1+\beta2x2+\ldots\beta ixi)},$$ Where π(x) is the probability of success when the value of the target variable is x. β1 denote a constant used for adjustment and β0 denote the coefficients target variables [39].

Decision Tree	A decision tree is a machine learning classifier that is mainly made of tree nodes, where data is divided into leaves. The node comprises features of data. Whereas the leaves show a class label. The learning tree model is generated by the branches of nodes connected to leaf nodes. One of the most potent machine learning classifier that is the classification of supervised algorithms is the decision tree [40].

Table 1. Classifier Description

4. Implementation

We discuss the application of the botnet detection system that have been developed using the CTU-13 from Stratosphere datasets repository. Then, we present detail information about our system design, evaluation metrics and experimental results.

4.1 Botnet detection system framework

We show the design of the botnet detection system in figure 2 below. We make the system up of different parts; the captured network flow, this part record the network flow and save it into packet capture (pcap) file format using Windump tool to record all the network flows on the network infrastructure, the pre-process part takes in pcap file, process them into pre-defined botnet features and output two different files. Then input them into machine learning classifier algorithms to build an efficient system from the selected features, the supplied dataset, and the test. From the pre-process part including classifier algorithms outcomes then produces our final predicted result.

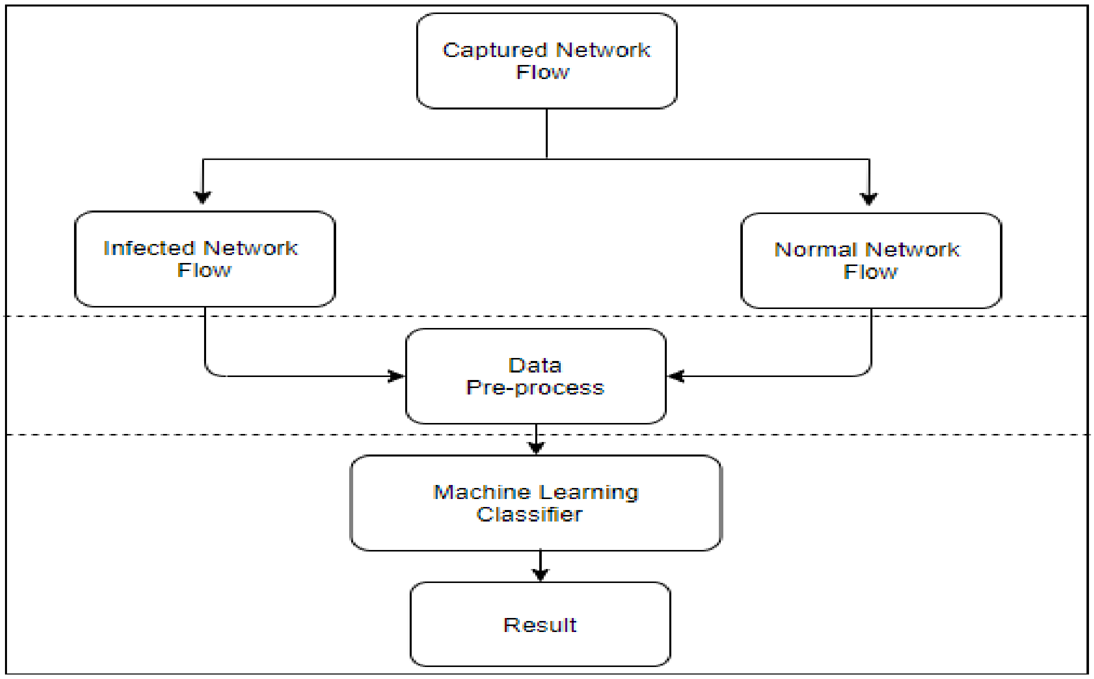

Figure 2. System Design

5. Evaluation metrics and experimental results

In this study we used k-fold crossed-validation technique to evaluate our predictive algorithms that was trained and validate their performance. We set13-fold crossed validation method in other to validate each botnet sample in our CTU-13 dataset. We divided the input sample into 13 sets. During the training process of the system we used 12 sets of sample for learning and the remaining 1 sample was used for testing the system. To obtain a better and robust result in the end, we iterate the training process 13 times and the performance of each algorithm was evaluated by using performance metrics. The labeled sample dataset that had been classified via the algorithms will produce results based on the performance metrics that our system is comparing against (Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) and Precision).

5.1 Experimental results

To evaluate the proposed system, we used dissimilar sample of datasets. The experiment was executed in the test environment with seven large datasets.

IRC, HTTP & P2P Bot	Classifiers	Accuracy (%)	Precision (%)	TPR (%)	FPR (%)
Neris	ANN	90.07	93.69	87.98	45.05
	Logistic Regression	96.97	89.90	99.87	20.10
	Decision Tree	99.90	90.89	99.98	03.05
Rbot	ANN	90.87	89.61	89.87	21.07
	Logistic Regression	95.78	90.70	97.70	15.17
	Decision Tree	99.91	99.98	97.59	11.20
Virut	ANN	90.57	89.91	87.81	30.17
	Logistic Regression	89.92	91.87	96.92	21.01
	Decision Tree	99.89	97.79	99.86	07.21
Menti	ANN	81.78	92.79	89.90	23.11
	Logistic Regression	90.89	97.71	92.99	09.19
	Decision Tree	99.92	98.79	97.79	20.09
Sogou	ANN	92.89	88.71	87.70	27.12
	Logistic Regression	96.99	92.95	99.98	10.34
	Decision Tree	99.89	97.98	99.90	12.23
	ANN	90.91	89.91	80.95	38.35

Murlo	Logistic Regression	92.82	89.89	98.89	28.18
	Decision Tree	99.90	99.85	95.91	10.01
NSIS.ay	ANN	91.89	87.99	89.97	22.12
	Logistic Regression	90.94	90.78	95.86	08.41
	Decision Tree	99.91	99.99	99.77	07.23

Table 2. The result of IRC, HTTP & P2P botnet detection using our method

5.2 Discussion

We used three machine learning classification algorithms as seen in Table 2. One among the classifiers, Logistics Regression stands out in True Positive Rate (TPR) with Neris, Rbot, Virut, Menti, Sogou, Murlo, and NSIS.ay detection archived an average of 98%. Conversely, the False Positive Rate (FPR) for Logistics Regression is a little high, this suggests that during botnet detection the classifier may produce a false alarm. Impressively, the Decision tree classifier produced the highest accuracy rate for each bot and good performance of False Positive Rate, We can express that, during detection Decision tree outperformed the other classifiers in classifying the infected and normal data flow because of high detection accuracy rate and also low false alarm. Therefore, we can say that the best classifier algorithms to detect IRC, HTTP and P2P bot for this research is the Decision tree algorithm. It has impressive performance in high accuracy, an encouraging true positive rate, and relatively low false alarm compare to the other two classifier algorithms. Conversely, in some occurrences, our system may falsely detect and misidentify normal to infected network flow in a real-life scenario. Take, for instance, when a user on a network load a particular website several times, this will send a continuous network flow to the network server. The system may see it as an attack because it looks like a botnet behavioural pattern [41]. However, for our system not to misidentify normal network flow as infected network flow and to perform effectively, we will look at selecting appropriate and required network features to avoid misidentification of network flows.

6. Conclusion

The number of IRC, HTTP, and P2P botnet attacks is on the increase daily. Hence, there is a need for discontinuing botnet's usual attacks on the network. This study targets to implement machine learning algorithms to detect IRC, HTTP, and P2P botnets on network infrastructure. The proposed system will detect botnet in the network by using network flow features. We measured the performance and tested the proposed system based on accuracy, true-positive rate, false-positive rate, and precision on seven dissimilar sample botnets. The machine learning algorithms used in our investigation are three selected classifiers namely Artificial Neural Networks, Logistic Regression, and Decision Tree. We were able to achieve our objective to develop a machine learning detection system that detects IRC, HTTP, and P2P botnet on a network.

The results we got from our experiment showed an improvement and substantial readings on classification detection of mischievous activities of IRC, HTTP, and P2P botnet in their network flow. The best classifier algorithms among the three used were Decision Tree with an average accuracy of 99.90% and True Positive Rate of 98.68%. Compare to the other two classifier algorithms the result Decision Tree shows that it be able to identify IRC, HTTP, and P2P botnet in network flow with a relatively false alarm. The result achieved in this study may be an additional knowledge in information security or cybersecurity field that machine learning classification algorithms able to detect botnet on a network.

6.1 Future work

One research to carry out is an effective botnet feature selection. In some events, our system may falsely detect and misidentify normal to infected network flow in a real-life scenario. And also to attempt more machine learning classifier algorithms by expanding botnet detection.

References

[1] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," *2014* International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, 2014.

[2] Y. Liu, X. Tao, Ma, X. Guan, J. Zheng, Q. Guo, L. Zhao, S. A novel IRC botnet detection method based on packet size sequence. In Proceedings of the 2010 IEEE International Conference on Communications (ICC), Cape Town, South Africa, 23–27 May 2010.

[3] SusanC.HerringIndiana University Available:https://www.academia.edu/37318273/ Computer-mediated_communication_on_the_internet.

[4] Kaspersky Laboratory, Statistics on Botnet-Assisted DDoS Attacks in Attacks in Q1 2015. https://securelist.com/statisticson-botnet-assisted-ddos-attacks-in-q1-2015/70071/. [Accessed: 1-July 2020].

[5] MyCERT Incident Statistics Report 2017. https://www.mycert.org.my/portal /publications ?id=7f17bda3-7d91-42e2-93fd-39476d75d35f&keyword=&year=2017. [Accessed: 1-June-2020].

[6] Pdfs.semanticscholar.org. 2020. https://pdfs.semanticscholar.org/bfa4/ 26bf57513c87f0969ba5e9e457d6f50279b6.pdf> [Accessed 24 July 2020].

[7] Coursehero.com, 2020. https://www.coursehero.com/file/44695344/securware-2018-7-10-30121pdf/. [Accessed: 24- Jul- 2020].

[8] Robert E. Creation of Internet Relay Chat Nicknames and Their Usage in English Chat room Discourse. http://www.robertecker.com/hp/re search/leet-converter.php.

[9] A. Karim, R. Salleh, M. Shiraz, S. Shah, I. Awan and N. Anuar, Botnet detection techniques: review, future trends, and issues, 2020.

[10] Detecting Botnets with Tight Command and Control - IEEE Conference Publication, Ieeexplore.ieee.org. https://ieeexplore.ieee.org/document/4116547.

[11] A Proposal of Metrics for Botnet Detection Based on Its Cooperative Behavior - IEEE Conference 2020. https://ieeexplore.ieee.org/document/4090153.

[12] Binkley R. J and Singh S. An algorithm for anomaly-based botnet detection: Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet.

[13] Evan Cooke and Farnam Jahanian, 2005, the Zombie Roundup: Understanding, Detecting, and Disrupting Botnets.

[14] C. Livadas, R. Walsh, D. Lapsley and W. T. Strayer, "Usilng Machine Learning Technliques to Identify Botnet Traffic, Proceedings. 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, 2006.

[15] A multifaceted approach to understanding the botnet phenomenon Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, Dl.acm.org. https://dl.acm.org/doi/10.1145/1177080.1177086.

[16] N. Feamster, Ramachandran A and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence.

[17]Peer-to-Peer Botnets: Overview and Case Study", Usenix.org. https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/grizzard/grizzard_html/index.html. [Accessed: 23- June- 2020].

[18]S. Chaware, A Survey of HTTP Botnet Detection, Semanticscholar.org, https://www.semanticscholar.org/paper/A-Survey-of-HTTP-Botnet-Detectio Chaware/77ea4b607184fde5a0752cd6cf31b87c1fcfeb39#citing-papers.

[19] T. Wang, X. Hu, J. Jang, S. Ji, M. Stoecklin and T. Taylor, BotMeter: Charting DGA-Botnet Landscapes in Large Networks, 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS).

[20] A comparison of clustering algorithms for botnet detection based on network flow - IEEEConferencePublication, Ieeexplore.ieee.org.https://ieeexplore.ieee.org/document/7537117

[21] The Activity Analysis of Malicious HTTP-Based Botnets Using Degree of Periodic Repeatability –IEEE Conference Publication", Ieeexplore.ieee.org, 2020. Available: https://ieeexplore.ieee.org/document/4725350. [Accessed: 06- July- 2020].

[22] Construction P2P firewall HTTP-Botnet defense mechanism - IEEE Conference Publication, Ieeexplore.ieee.org.https://ieeexplore.ieee.org/document/5953166. [Accessed: 06- July- 2020].

[23] Evasion technique and detection of malicious botnet- IEEE Conference Publication", Ieeexplore.ieee.org.https://ieeexplore.ieee.org/document/5678101. [Accessed: 06- July- 2020].

[24] Badonnel and Robert Koch and Aiko Pras and Martin Drasar and Burkhard Stiller, Management and Security in the Age of Hyperconnectivity, 2016.

[25] A. Karim, R. Salleh, M. Khan, A. Siddiqa, K. Choo, On the Analysis and Detection of MobileBotnet: http: //www.jucs.org/ujs/jucs/Journal/Volume%2022/Issue_22_4/on_the_ Analysis

[26] G. Kirubavathi Venkatesh, R. Anitha, Botnet detection via mining of traffic flow characteristics Computer Science Comput. Electr. Eng. 2016 (First Publication: 1 February 2016)

[27] Peer to Peer Botnet Detection Using Data Mining Scheme -IEEE Conference Publication", Ieeexplore.ieee.org.https://ieeexplore.ieee.org/document/5566407. [Accessed: 06-July- 2020].

[28] D. Zhao et al., Botnet detection based on traffic behaviour analysis and flow intervals, 2020.

[29] Chien-Hau Hung, Hung-Min Sun. Department of Computer Science National Tsing Hua University Hsinchu, Taiwan, A botnet detection system based on machine-learning using flow-based features, @e Twelfth International Conference on Emerging Security Information, Italy, September 2018.

[30] Yegneswaran, V, Guofei, G, et al. Active botnet probing to identify obscure command and control channels. In Proceedings of the IEEE Annual Computer Security Application Conference, Honolulu, HI, USA, 7–11 December 2009.

[31] G. Warner, A. Sprague, and C. Wei, Detection of networks blocks used by the storm worm botnet, in Proceedings of the 46th Annual Southeast Regional Conference on XX, ACM-SE 46, (New York, NY, USA, 2008.

[32] A. Boukhtouta, L. Wang, P. Sinha, A. Youssef, M. Debbabi, H. Binsalleeh, and T. Ormerod, On the Analysis of the Zeus Botnet Crimeware Toolkit, Proceedings of the 8th Annual Conference on Privacy, Security and Trust., 2010.

[33] H. Choi, H. Lee, H. Lee and H. Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic, 7th IEEE International Conference on Computer and Information Technology, Aizu-Wakamatsu, Fukushima, 2007.

[34] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[35] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova and A. A. Ghorbani, Towards effective feature selection in machine learning-based botnet detection approaches, 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, 2014, pp. 247-255, doi: 10.1109/CNS.2014.6997492.

[36] Kevin Gurney. An introduction to neural networks. https://www.inf.ed.ac.uk/teaching /courses/nlu/assets/reading/Gurney_et_al.pdf. [Accessed: 12- Aug- 2020].

[37] A. P. Engelbrecht. Computational Intelligence: An Introduction, 2nd ed. John Wiley, 2007.

[38] Prof. Dr Amit Konar. Computational Intelligence: Principles, Techniques and Applications. Springer, the Netherlands, 2005. No. of pages: 705. ISBN: 3-540-20898-4.

[39] D. W. Hosmer, Applied Logistic Regression, 2nd Ed. New York: Wiley, 2000.

[40] Classification Algorithms - Decision Tree - Tutorialspoint", *Tutorialspoint.com*, https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_decision_tree.htm. [Accessed: 12- Aug- 2020].

[41] M. Eslahi, H. Hashim and N. M. Tahir, An efficient false alarm reduction approach in HTTP-based botnet detection, 2013 IEEE Symposium on Computers & Informatics *(ISCI)*, Langkawi, 2013, pp. 201-205, doi: 10.1109/ISCI.2013.6612403.