International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 3, June 2020, pp. 2543~2550
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i3.pp2543-2550

Botnet detection using ensemble classifiers of network flow

Zahraa M. Algelal[1], Eman Abdulaziz Ghani Aldhaher[2], Dalia N. Abdul-Wadood[3], Radhwan Hussein Abdulzhraa Al-Sagheer[4]
[1,2,4]Department of Computer Science, Faculty of Education for Girls, University of Kufa, Iraq
[3]College of Medicine, University of Bagdad, Iraq

Article Info

Article history:

Received Mar 19, 2019
Revised Oct 26, 2019
Accepted Nov 22, 2019

Keywords:

Botnet
Ensemble
Machine learning
Network flow
Network security

ABSTRACT

Recently, Botnets have become a common tool for implementing and transferring various malicious codes over the Internet. These codes can be used to execute many malicious activities including DDOS attack, send spam, click fraud, and steal data. Therefore, it is necessary to use Modern technologies to reduce this phenomenon and avoid them in advance in order to differentiate the Botnets traffic from normal network traffic. In this work, ensemble classifier algorithms to identify such damaging botnet traffic. We experimented with different ensemble algorithms to compare and analyze their ability to classify the botnet traffic from the normal traffic by selecting distinguishing features of the network traffic. Botnet Detection offers a reliable and cheap style for ensuring transferring integrity and warning the risks before its occurrence.

Corresponding Author:

Radhwan Hussein Abdulzhraa Al-Sagheer,
Department Of Computer Science,
Faculty Of Education For Girls,
University Of Kufa,
Najaf, Iraq.
Email: radhwan.hu@uokufa.edu.iq

1. INTRODUCTION

Day by day the dependency on the Internet has increased in our daily lives, mainly in many important fields such as educational organizations, communication companies, government facilities, banking, and e-commerce. This adds many difficulties in managing the web and utilizing the application, for example, protecting the user data, integrity, privacy, and availability [1]. All these reasons changed the consideration of attackers to thinking about financial advantages, the attackers utilize diverse malware to accomplish their objectives. Among the different sorts of malware, Botnet is one of the most genuine ways of doing the crime online on the web [2]. Therefore, financial benefits are the main aim of generating botnets by the attacker [3]. McAfee's Threat Report for the first quarter of 2019 showed that the number of newly discovered malware threats has achieved more than 60 million threats. The whole malware estimated to reach more than 800million before the end of 2018 [4]. Moreover, the statistics revealed by CenturyLinkin the first half of 2019 showed that the average number of threats amounted to 3.8 million unique threats per month, and explained that the top five countries suspected for the movement of botnets attack are the United States, Spain, India, Indonesia, and Turkey [5]. This huge number of malware threats caused by botnets have been planned, each one becoming more resilient, unsafe, and smart. Fortunately, botnet detection methods have also developed, which employ different approaches such as traffic analysis [6-8], DNS based methods [9] and machine learning such as decision trees [10], Neural Network [11] and clustering [12].

The botnet detection modelin this study focuses on network traffic analysis under the behavior characteristic that is flows generated by bots be different from normal flows. With this characteristic, machine learning (ensemble classifier algorithms) can be attempted to classify flows depending on their behavior with the possible highest accuracy. It is important to select the essential features by using some

methods, such as information gain. The process of feature selection consequently guaranteed high accuracy and reduced training time when performing, which is mentioned by [13]. The results of the detection methods were verified using CTU-13 Dataset and 10 fold cross validation was adopted to evaluate the proposed model performance.

2. RELATED WORKS

Recently, there has been growing attention in strategies for Botnet detection. Whereas it is important to learn how a botnet has an infection on the PCs, it is more serious to determine the infested device prior to it is exploited to set mischievous actions. There are various techniques have been introduced to detect Botnets. These methods can be categorized into signature based, anomaly based, DNS based and data mining techniques [14]. The signature based techniques, Behal [14] have proposed the "N-EDPS" which is a signature based system for botnet detection and prevention. Through monitoring the outbound traffic, their system concentrates on discovering and stopping malware infections especially botnets. They employed the current freely available software which is open source usually. For detection, they utilized "BotHunter" and "Snort Inline" for the prevention [6]. By using several network traffic anomalies, the anomaly basedhas tried to identify Botnet. For example, high volumes of traffic, traffic passing to unusual ports, high network latency and anomalous behavior may indicate the existence of bots in the network [15]. These trends can detect new types of the bot. Karasaridis [7] have presented an approach to detect IRC botnet controllers from Netflow. Their approach was able to detect the botnet communications which are encrypted. It can supply extra BotHunter evidence-trails for infection actions [7].

Another method to detect Botnet has been developed by Wang and Paschalidis in 2017, their proposed method has two phases, the first phase suggests two techniques in order to create the empirical distribution. The two techniques are flow based approach and graph based approach. The flow based approach is for approximating the histogram of quantized flows and the graph based approach for approximating the grade distribution of node communication graphs. The second phase uses the social network community to detect the Bots, this was done by a graph that captures the associations of connections among nodes over time. They utilized real-world botnet traffic in the experiment which is CTU-13 dataset [8]. DNS-based detection techniques are utilized DNS-related network traffics generated by the botnet. These techniques are similar to anomalous detection techniques where similar anomaly detection algorithms are applied on DNS traffic. In 2019, Alieyan et al. proposed DNS rule-based detection technique for botnet detection. They defined some rules to detect IPs that exhibition anomalies in DNS requests and DNS replies. This rule technique is using to enable users to detect the existence of irregular behaviors of DNS requests and DNS replies. These behaviors are proposed for the detection of any existence of DNS based botnets and any source IP that shows such behaviours [9]. Mining based Detection techniques which are considered as effective techniques for botnet detection. In 2013, Garg et al. presented a method for the detection of P2P Botnets using several mining algorithms such as K-nearest neighbor, Naïve Bayes and decision tree (J48).

The ability of these algorithms to detect P2P networks has been analyzed and compared by using many of the features of network traffic [16]. K-medoids and K-means [12] are utilized to derive a set of rules to decide which connections should be considered as a botnet. Datasets were extracted from the sources ISOT and ISCX. Results on K-medoids were better for almost all these experiments than K-means. As a methodology, Liao [17] used packet size to differentiate between P2P Botnet traffic and normal P2P traffic. They provided the following observations. Initially, P2P Bots attempts to update information for other Bots instead of remaining inactive. Next, the Bot mainly transfers data with lower communication rate. In order to classify network traffic, three methods were used: Naïve Bayes, Bayesian networks, and J48. However, the size of packets in P2P Botnet was found small compared with other P2P applications [17]. Others proposed neural networks-based botnet detection techniques to identify the legal and illegal patterns. Through using some of the TCP-based features, a multi-layer neural network have been trained to detect HTTP botnets. The results showed that this method is effective and can detect HTTP botnets at a low false positive rate [18]. Graphical Turing tests "VISUALCOM", "IMGCOM", and "AD-IMGCOM" have been used in building the model to prevent and detect the DDoS attacks in cloud computing from a botnet. This model is implemented with a queuing model [19].

3. BOTNET OVERVIEW

Botnets are networks comprising of a huge number of PCs infected by Bots. These infected PCs, remotely controlled by "botmasters" to implement specific malicious activities. The attacker arranges a communication station to direct instructions to the Bots and to obtain results from them [20].

This communication channel is called the command and control (C&C) channel. TheC&C is the main feature that distinguishes between Botnet and other types of malware [21]. Botnets may be categorized based on the C&C mechanism into two major types: centralized and decentralized C&C [22]. The attacker or botmaster is usually used the C&C server to direct a command to the bots in centralized botnets as illustrated in Figure 1(a). Due to its uncomplicatedness, the centralized botnet is widely used via numerous botnet groups. The IRC-based botnets and HTTP Botnet are considered among the most famous of botnet approaches. However, the single point of failure C&C server in centralized Botnet is the major problem in it. A shutdown of the C&C server might result in a lack of communication among the bots and the botmaster [23]. The next generation of botnets, attackers have started to structure Botnets based on a decentralized architecture, such as, the Peer-to-Peer botnet [24] which it adopted via many forms of the botnet, for example, Waledac, Storm, and Conficker [25]. Peer-to-Peer botnet is a form that adopted a decentralized architecture to avoid having any single point of failure. In P2Pbotnet as illustrated in Figure 1(b), there is no central server, and bots are linked to each other topologically and act as a bot (client) and C&C (server) at the same time. For this situation, the botmaster can direct instructions to the infected peers to implement any order or requesting information at any time [26].

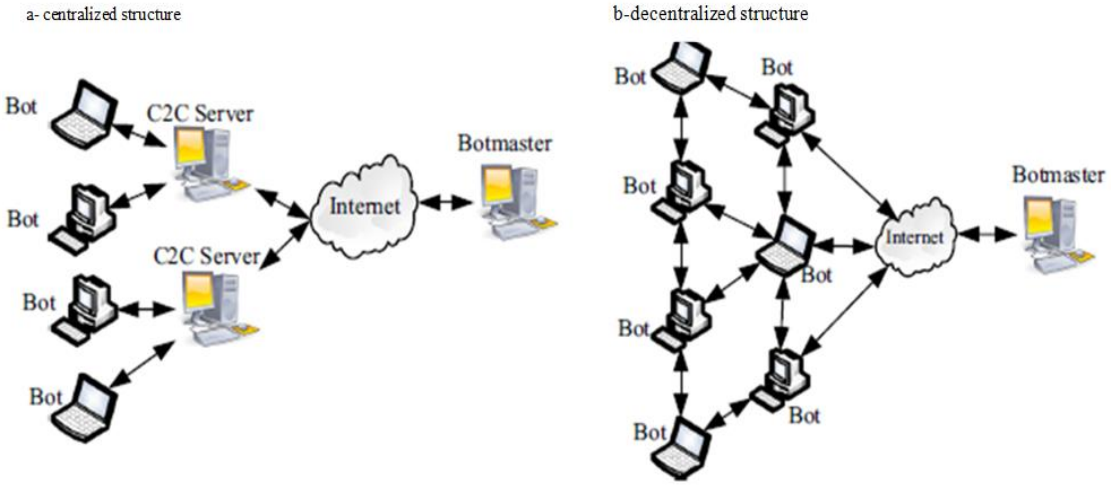

Figure 1. Structures of botent

4. ENSEMBLER CLASSIFIER FRAMEWORK

Ensemble method constructs a set of classifiers (base learners) from training data and combines them to classify new data examples by taking a vote (typically by weighted or un-weighted) of their decisions [27]. The main idea behind the ensemble learning is to employ several individual classifiers and combine their predictions to obtain a classifier that can work better than each of them [28]. In this research, the most three common ensemble approaches: Bagging, Boosting and random forest methods have been used, as shown in Figure 2 [29].

4.1. Bagging

Bagging or bootstrap aggregating is a method to get multiple learners, where the training data set for each learner is produced by random uniformly sampling with replacement from the original data set [30]. Bagging is consists of two parts: bootstrap and aggregation. A significant reduction in error could produce when the combination of independent base learners happens, thus, it is essential to keep the base learner independent as possible. The bootstrap distribution is utilized via the bagging technique to generate diverse base learners. Using random sampling and replacement, the bagging method produces bootstrap sampling of the training data, it implemented bootstrap sampling [31] to generate data subsets to train the base learners. Moreover, several repeats of the original dataset are formed through utilizing random selection with replacement. Next, every dataset is utilized to form a new learner and the formed set of learners is used to construct an ensemble. For aggregating the outputs of the base learners, bagging utilizes one of the most common methodologies for classification, which is voting while it uses an averaging approach to dealing with the regression problem.

4.2. Boosting

Boosting technique also called ARCing "Adaptive Resampling and Combining" [28]. It is related to the algorithms that can convert weak learners to strong learners. Generally, we can be defined as the weak learner as the learner which is slightly better than the random guess. Oppositely, the strong learner is very close to a perfect result. Boosting is a common method utilized to improve learning method performance. The concept behind boosting is that a weak learner can be boosted to a strong learner Schapire [32] proposed the boosting technique for that purpose. Boosting is consider as an advancing additive model and it utilizes the whole dataset for each stage. This technique merges the outputs from various classifiers with the aim of produce an effective classifier [33].

4.3. Random forest

The random forest belongs to the family of ensemble approaches. It grows many decision trees by utilizing randomly partitioning the training data and features, where each tree is built depends on the values of an independent set of random vectors of the training dataset. These random vectors produced from a fixed probability distribution since the probability distribution is diverse to concentrate on instances, which has difficulties to classify [34]. The randomization aids in reducing the correlation among decision trees to improve the generalization error of the ensemble [30].

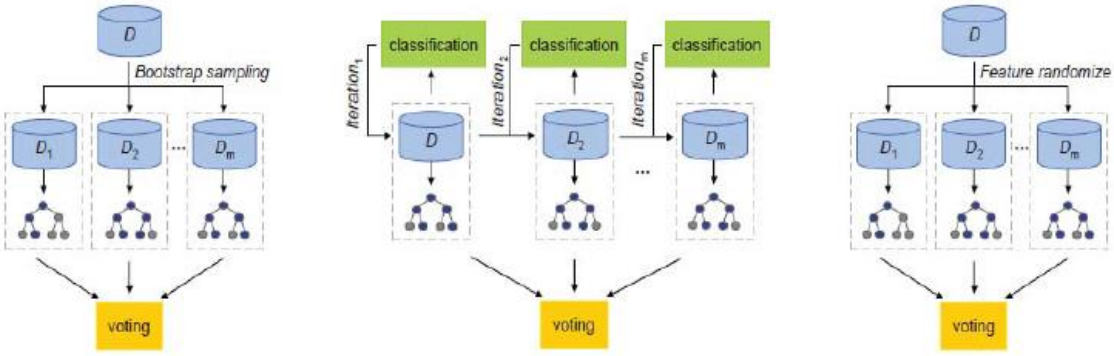

Figure 2. Shows the basic workflow for (a) Boosting, (b) Bagging, and (c) Random forest

5. PROPOSED MODEL

The proposed system for the Botnets detection, the classification of network traffic is achieved by applying three different Ensemble classifier algorithms: Bagging, Boosting and Random Forest. The results of the detection methods were verified using CTU-13 Dataset and 10 fold cross validation was adopted to evaluate the proposed model performance. The framework of our system is described in Figure 3.

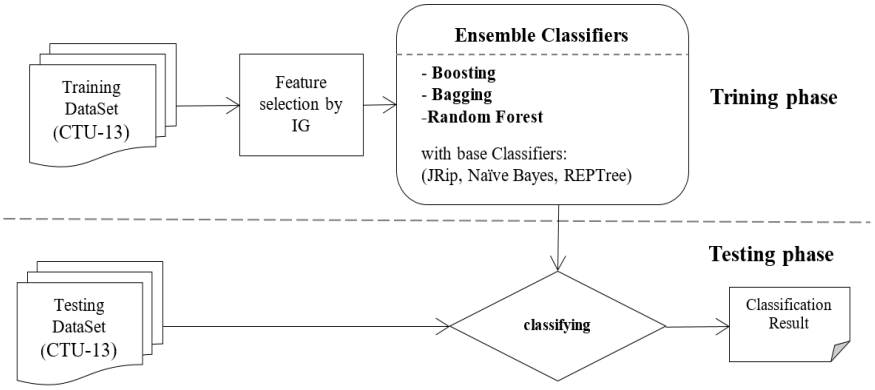

Figure 3. The proposed framework for botnet detection

5.1. Dataset

The CTU-13 dataset [35] is one of the largest NetFlow datasets available that contains botnet traffic as well as normal and background labeled data. These data were collected by the Czech Technical University (CTU), 2011. The CTU-13 dataset has 13 datasets (called scenarios) of different botnet samples. In addition to that, each of these scenarios has been recorded in a separate file as a NetFlow which using CSV notation. These NetFlow files include the following attributes: Start Time, Duration, Source IP address, Source Port, Direction, Destination IP address, Destination Port, Protocol State (e.g., UTP, TCP), SToS (Type Of Service), Total Packets (exchanged between source and destination), Total Bytes, and Label (e.g., background, normal, and botnet).

5.2. Feature selection

In the Botnet detection technique, one of the essential parts is feature extraction. By experimenting not all features have similarly contributed to the result, some of them are significant and pertinent than the other to the learning and analysis process. The redundancy of features may cause a reduction in the accuracy, to rank the features in this paper, the information gain (1) measure has been used [36].

$$IG(A) = H(S) - \sum_t \frac{S_t}{S} H(S_i) \ldots \quad (1)$$

where H(S) is the entropy of the given a training set S and H(Si) is the the entropy of the ith subset of the training set Since the attribute A is observed. The gained information is utilized to assist in ranking the attribute in machine learning and the attribute with the high IG is ranked higher than the other attributes because it has a stronger power in classifying the data. Figures 4 show that the classification of the (12) attributes of the CTU-13 dataset sorted in descending order by information gain. After ranking the attributes using information gain the best ones are selected Therefore the top 8 attributes based on their importance value are considered in this work. The selected attributes are: < Source IP, Destination IP, Start Time, duration, IP protocol, protocol state, the total number of packets and total bytes exchanged>.

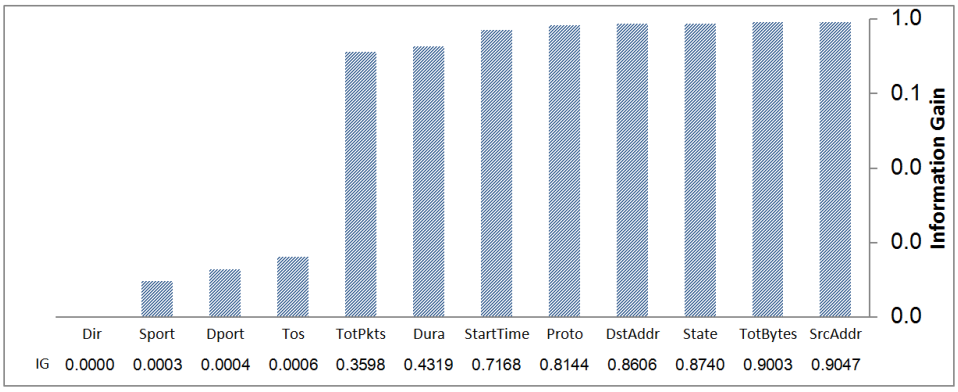

Figure 4. The information gain for each attribute (A base-10 log scale is used for the Y axis)

5.3. Detection methods

The research introduces three Ensemble methods to identify between botnet and normal traffic by classifying the corresponding flows. We have used bagging, AdaBoost, Random Forest method of the ensemble-based classifier. The machine learning algorithms like JRip, Naïve Bayes and REPTree have been deployed as a base classifier on ensemble methods.

- JRip: This class applies a rule-suggestion learner, "Repeated Incremental Pruning to Produce Error Reduction" (RIPPER).
- Naïve Bayes: It depends on what is called the Bayesian theorem, It's particularly appropriate if the input dimensions are high.
- REPTree: "Reduced Error Pruning Tree (REPT)" Builds a decision tree using information gain as the partitioning criterion and prunes it using reduced error.

6. EXPERIMENTAL RESULTS

In our experiments, we have used CTU Botnet Dataset (Scenario 11), which already contains labeled bidirectional net flows, The selected attributes by information gain are: Source IP, Destination IP, Start Time, duration, IP protocol, protocol state, the total number of packets and total bytes exchanged, as shown in Figure 4. A data mining software called WEKA has been used to apply ensemble algorithms to this dataset. WEKA is a group of machine learning algorithms for solving data mining tasks. The algorithms can either directly applied by using GUI or called from Java code. Because the size of the downloaded data is too large to be processed by the available PC machines, so to deal with this problem a small part of the data was randomly selected that can be handled by the available devices. This sample of data was entirely random selected to guarantee that the results of the analysis stay unbiased by the selective process.

Five different measures were utilized to evaluate the performance of the proposed method, those measures are Accuracy, False Positive Rate, Precision, Recall, and F-measure. The ten-fold cross-validation technique was adopted to estimate the accuracy of the proposed method where the dataset is split at random manner into similarly exclusive and equal-sized subsets. Also, the cross-validation method guarantees that every part of the basic dataset is utilized in a similar number of times in training and testing. The generated results usingensemble methods with the three different classification schemes (JRip, Naïve Bayes and REPTree as a base classifier) are given in Table 1.

Table 1. Performance comparison table of classifiers

Methods		Accuracy%	FPR	Precision	Recall	F-measure
AdaBoost	JRip	99.84	0.002	0.998	0.998	0.998
	Naïve Bayes	98.12	0.038	0.982	0.981	0.981
	REPTree	85.48	0.307	0.88	0.855	0.841
Bagging	JRip	99.84	0.002	0.998	0.998	0.998
	Naïve Bayes	99.1	0.018	0.991	0.991	0.991
	REPTree	85.48	0.307	0.88	0.855	0.841
Random Forest		95.11	0.103	0.954	0.951	0.95

Table 1 present the comparison of ensemble algorithms over the 10 fold cross-validation concerning different comparison measures. JRip classifier achieves the highest classification accuracy (99.84%) in both AdaBoost and Bagging compared with the accuracy of Naïve Bayes (98.12%) and REPTree (85.48%) in AdaBoost and with the accuracy of Naïve Bayes (99.1%) and REPTree (85.48%) in Bagging. Furthermore, Table 1 can conclude the JRip classifier gives the lower false positive rate (0.002) in both AdaBoost and Bagging and the highest false positive rate from REPTree (0.307) and it has a low accuracy too. Random Forest also achieves high detection accuracy (95.11%) and a low false positive rate (0.103). The Ensemble with JRip Classifiers model has been compared with five different methods which are clustering, Neural Network, Recurrent Neural Network [37, 38], K-medoids, K-means [12], Long Short-Term Memory (LSTM) [11], anddecision trees [10]. The comparative of results in Table 2 show that our proposal Ensemble with JRip Classifiers model achieves better detection accuracythan the existing systems for botnet detection.

Table 2. A comparison of the proposed model with other algorithms

Author	Data set	Methods	Accuracy (%)
Bansal and Mahapatra[37]	ISCX &	Clustering	98.39
	CTU-13	Neural Network	89.38
		Recurrent Neural Network	83.09
Alejandre et al. [12]	ISOT&	K-medoids	69.99
	ISCX.	and K-means	73.37
Sinha K. [11]	CTU-13	Long Short-Term Memory (LSTM)	96.2%
Khan R. et al [10]	ISOT &	decision trees	98.7%.
	CTU-13		
Proposed model	CTU-13	AdaBoost+ JRip	99.84
(Ensemble Classifiers)	(Scenario 11)	Bagging+ JRip	

7. CONCLUSION

In this paper, we have presented an approach to deal with botnet detection problem, which is considered as a serious and critical threat of internet security. One approach to handle this problem is by recognizing botnet actions and infected devices to provide vital safety measures. The proposed model was

based on “ensemble classifiers methods” which are performing better performance through combining multiple algorithms in the process of botnet analysis. Also, through the feature selection process, the most significant features were extracted for the analysis process to increase the accuracy and decrease the time as well as resources. To evaluate this proposed methodology, we have performed experimental assessments on the CTU botnet dataset and the performance of the proposed model was assessed utilizing 10 fold cross-validation. The results showed that the proposed model was effective and has promising results.

REFERENCES

[1] Stevanovic M, Revsbech K, Pedersen JM, Sharp R, Jensen CD, "A collaborative approach to botnet protection," *In: International Conference on Availability, Reliability, and Security. Springer*, Berlin, Heidelberg, pp. 624-638, 2012.

[2] Silva SSC, Silva RMP, Pinto RCG, Salles RM, "Botnets: A survey," *Computer Networks*, vol. 57(2), pp. 378-403, 2013.

[3] Rodríguez-Gómez RA, Maciá-Fernández G, García-Teodoro P, "Survey and taxonomy of botnet research through life-cycle," *ACM Computing Surveys (CSUR)*, vol. 45(4), pp. 45, 2013.

[4] McAfee, "McAfee labs threats report," 2019. [Online], Available from: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2019.pdf. [Accessed 2019 Sep 14].

[5] CenturyLink (NYSE: CTL), "Century Link 2019 Threat Report," 2019. [Online], Available: https://www.centurylink.com/asset/business/enterprise/report/2019, 2010threat-research-report.pdf. [Accessed 14 September 2019].

[6] Behal S, Brar AS, Kumar K., "Signature-based botnet detection and prevention," In: *Proceedings of International Symposium on Computer Engineering and Technology*, pp. 127-132, 2010.

[7] Karasaridis A., Rexroad B., Hoeflin D., "Wide-scale botnet detection and characterization," In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pp. 7, 2007.

[8] Wang J., Paschalidis IC., "Botnet detection based on anomaly and community detection," *IEEE Trans Control Netw Syst*, vol. 4(2), pp. 392-404, 2017.

[9] Alieyan K, Almomani A, Anbar M, Abdullah R, Gupta B B., "DNS rule-based schema to botnet detection," *Enterprise Information Systems*, Taylor & Francis, pp. 1-20, 2019.

[10] Khan R U, Zhang X, Kumar R, Sharif A., "An Adaptive Multi-Layer Botnet Detection Technique Using Machine Learning Classifiers," *Applied Sciences*, vol. 9(11), pp. 2375, 2019.

[11] Sinha K, Viswanathan A, Bunn J., "Tracking Temporal Evolution of Network Activity for Botnet Detection," *arXiv preprint arXiv: 1908.03443*, 2019.

[12] Alejandre F V, Cortés N C, Anaya, E A., "Botnet detection using clustering algorithms," *Research in Computing Science*, vol. 118, pp. 65-75. 2016

[13] Omara H, Lazaar M, Tabii Y., "Effect of feature selection on gene expression datasets classification accuracy," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8(5), pp. 3194-3203, 2018.

[14] Feily M, Shahrestani A, Ramadass S., "A survey of botnet and botnet detection," *In 2009 Third International Conference on Emerging Security Information, Systems and Technologies, IEEE*, pp. 268-273, 2009.

[15] Zeidanloo HR, Shooshtari MJZ, Amoli PV, Safari M, Zamani M., "A taxonomy of botnet detection techniques," *In Computer Science and Information Technology (ICCSIT), IEEE*, pp. 158-162, 2010.

[16] Garg S, Singh AK, Sarje AK, Peddoju SK., "Behaviour analysis of machine learning algorithms for detecting P2P botnets," In *Advanced computing technologies (ICACT), IEEE*, pp. 1-4, 2013.

[17] Liao W-H, Chang C-C., "Peer to peer botnet detection using data mining scheme," *In Internet Technology and Applications, 2010 International Conference on, IEEE*, pp. 1-4, 2010.

[18] Nogueira A, Salvador P, Blessa F., "A botnet detection system based on neural networks," In *Digital Telecommunications (ICDT), IEEE*, pp. 57-62, 2010.

[19] Saravanan A, SathyaBama S, Kadry S, Ramasamy L K., "A new framework to alleviate DDoS vulnerabilities in cloud computing," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9(5), pp. 4163-4175, 2019.

[20] Lu W, Rammidi G, Ghorbani AA., "Clustering botnet communication traffic based on n-gram feature selection," *ComputCommun*, vol. 34(3), pp. 502-14, 2011.

[21] Zeidanloo HR, Manaf AB, Vahdani P, Tabatabaei F, Zamani M., "Botnet detection based on traffic monitoring," in *Networking and Information Technology (ICNIT), IEEE*, pp. 97-101, 2010.

[22] Han K-S, Im EG., "*A survey on P2P Botnet detection*," in *Proceedings of the International Conference on IT Convergence and Security 2011*, Springer, pp. 589-593, 2012.

[23] Ludl C, McAllister S, Kirda E, Kruegel C., "On the effectiveness of techniques to detect phishing sites," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, pp. 20-39, 2007.

[24] Felix J, Joseph C, Ghorbani AA., "Group behavior metrics for P2P botnet detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, pp. 93-104, 2012.

[25] Davis CR., Fernandez JM., Neville S., "Optimising sybil attacks aga inst P2P-based botnets," In *2009 4th International Conference on Malicious and Unwanted Software*, IEEE, pp. 78-87, 2009.

[26] Wang P, Aslam B, Zou CC., "Peer-to-Peer Botnets," in *Handbook of Information and Communication Security*, Springer, pp. 335-350, 2010.
[27] Biau G, Devroye L, Lugosi G, "Consistency of random forests and other averaging classifiers," *J Mach Learn Res*, vol. 9, pp. 2015-2033, 2008.
[28] Rokach L., *Pattern Classification Using Ensemble Methods*, World Scientific, 2009.
[29] Saini R, Ghosh SK., "Ensemble classifiers in remote sensing: A review," Proceeding *IEEE 2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 1149-1152, 2017.
[30] Tan P-N, Steinbach M, Kumar V., "Introduction to Data Mining. 2nd Edition. person addison Wesley," 2006.
[31] Efron B J, Tibshirani R., "An introduction to the bootstrap," *Journal of the American Statistical Association. CRC press*, vol. 89, pp. 436, 1993.
[32] Schapire RE., "The strength of weak learnability," *Mach Learn*, vol. 5(2), pp. 197-227, 1990.
[33] Friedman Jerome H., "Greedy function approximation: a gradient boosting machine," *Ann Stat*, vol. 29, pp. 1189-1232, 2001.
[34] Belgiu M., Drăguţ L., "Random forest in remote sensing: A review of applications and future directions," *ISPRS J Photogramm Remote Sens*, vol. 114, pp. 24-31, 2016.
[35] Garcia S., Grill M., Stiborek J., Zunino A., "An empirical comparison of botnet detection methods," *ComputSecur*, vol. 45, pp. 100-23, 2014.
[36] Kullback S, Leibler RA., "On information and sufficiency," *Ann Math Stat*, vol. 22(1), pp. 79-86, 1951.
[37] Bansal A, Mahapatra S., "A comparative analysis of machine learning techniques for botnet detection," *In Proceedings of the 10th International Conference on Security of Information and Networks*, pp. 91-98, 2017.
[38] Radhwan Hussein Abdulzhraa Al-Sagheer1, K. I. Mohammed, "Impact of crack length into pipe conveying fluid utilizing fast fourier transform computer algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9(4), pp. 2541-2547, 2019.