Efficient Botnet Detection using Feature Ranking and Hyperparameter Tuning

Meshal Farhan AL-Anazi
College of Computer and Information Security
Naif Arab University for Security Sciences, KSA

Mostafa G. M. Mostafa
College of Computer and Information Security
Naif Arab University for Security Sciences, KSA

ABSTRACT
Botnet is considered a multifunctional malware. It can be leveraged by criminals to launch variety of malware attacks such as click fraud, DDOS, spam, etc. Moreover, the botnets pretend the normal traffic by leveraging common protocols such as IRC, HTTP, DNS and P2P for command control. Therefore, distinguishing botnet behavior is challenging because it has similarities with normal protocols behaviors. Most of previous researches focus on detecting specific type of botnet. Moreover, they rely on limited number of features. In addition, they do not select the optimal model by tuning the hyperparameters of machine learning algorithms. In this paper we use a recent dataset that containing a diverse set of botnet traces and wider flow features. We select the relevant features using several ranking algorithms. Eventually, the optimal models are selected by tuning the hyperparameters of machine learning algorithms.

General Terms
Intrusion Detection Systems (IDS), Machine Learning (ML)

Keywords
Botnet, Hyperparameter Tuning, Random Search

1. INTRODUCTION
Nowadays the botnet is one of the sever threat incorporating variety of malicious activities. The botnet helps the criminals in launching large scale and disastrous attacks. In 2015, Over 400M identities were hijacked globally from bots [1]. In 2016, a botnet was generating over 50,000 HTTP requests per second during a targeted DDoS attack [2]. These incidents clearly show the risks constituted by botnets of this size and scope, as well as the expected increased scale and complexity of future attacks.

Botnet is a network of compromised computers (bots) under the remote control of a botmaster [3]. Bot provides the attacker with the ability to remotely control behavior of the compromised computers through specially deployed Command and Control (C&C) communication channels [4]. Based on the Command and Control (C&C) communication channel the botnet topologies can be classified into three Models: centralized, decentralized and unstructured model [5]. The botnet Command and Control (C&C) pretend as legitimate communication by using popular protocols such as IRC, HTTP and P2P protocols. In their investigation of botnet using machine learning, Matija and Jens [4] show that traffic analysis is one of the main means of identifying the botnet existence.

The performance of machine learning algorithms relies on several factors such as dataset quality, best features and appropriate algorithm hyperparameters. However, most of previous researches focus on feature selection. On other hand, they underestimate the quality of dataset by using datasets containing limited type of botnets. Moreover, they do not optimize the machine learning performance by tuning the hyperparameters.

In this paper we use a recent dataset that containing a diverse set of botnet traces and wider flow features. We select the relevant features using several ranking algorithms. Eventually, the optimal models are selected by tuning the hyperparameters of machine learning algorithms.

The rest of the paper is organized as follows: Section 2 describes the background and related work. In section 3 we explain the methodology. The results and discussions are presented in section 4. Finally, section 5 presents the conclusions.

2. BACKGROUND AND RELATED WORK
Botnet detection can be classified into host-based and network-based. Host-Based techniques detect the bots by investigating different computer evidence sources such as application and OS logs. In this paper we focus in network-based botnet detection.

The network-based techniques can be classified into signature based and behavior-based techniques. The signature-based techniques monitor and inspect all the network traffic passing through sensor such as snort. Once string of botnet match for payload, warning alert for botnet is raised. Even the signature-based techniques demonstrate very high identification accuracy, they suffer from several limitations. Signatures-based detection cannot detect unknown attacks for which there is no signature available, and some botnet adopting evasion techniques such as polymorphic [5]. Moreover, the signature-based techniques require payload inspection which resource intensive. Additionally, new bots frequently utilize encryption and other methods to obfuscate their communication and defeat packet inspection techniques [6].

Behavior-based techniques rely on traffic flow analysis to detect the botnets. A flow summarizes the traffic information for all the packets sharing the following five fields: source IP address, source port, destination IP address, destination port and protocol. The behavior-based techniques do not require deep inspection of packet payload so, they are more resilient for encryption of botnet command and control (C&C).

Livadas et al [7], proposed method to detect command and control (C2) traffic of IRC-based botnets using machine learning classification algorithms. For testing they use very small dataset called Testbed trace involves 74 flows, 38 of which were IRC flows and the rest is botnet flow. Three classifiers namely decision tree, naive Bayes, and Bayesian networks were chosen for comparison. After running the experiment several times, the decision tree and the Bayesian

networks classifiers perform very poorly. They performed exhaustive warping search with the three classifiers to find the optimal features. Unfortunately, the optimal feature sets performed worse in identifying the testbed botnet traces.

Strayer et al [8], proposed method with goal to determine if they can find evidence of C&C botnet activity by only monitoring network traffic. they performed the classification using decision trees, Naïve Bayes, Bayesian Networks. The Bayesian Networks technique got a low FPR, but higher FNR. Decision tree got a balance between FNR and FPR.

The existing botnet such as Storm-botnet and Zeus-Botnet that have unique command and control architecture. Such architecture uniqueness tends to exhibit identifiable behaviors that can be recognized by analyzing network traffic characteristics which influence Saad et al [9] to propose method to detect the P2P botnet C&C using network traffic behaviors. the true detection rate of the P2P Botnet C&C was above 90% for the Support Vector Machine, Artificial Neural Network and the Nearest Neighbors Classifier and the total error rate is less than 7%.

Zhao et al [6], propose an approach to detect the botnet activities by classifying behavior based on time intervals. They selected the features by intuition based on the behavior of variety of common protocols as well as the behavior of known botnets such as Storm, Nugache and Waledac. They used (REPTree) to improve the detection accuracy with respect of noisy data. The result of REPTree with 10-fold cross validation is above 90% with a very low false positive rate.

Many of P2P Bots utilize TCP protocol for communication such as Waledac Bot, Storm Bot, Conficker Bot and Zeus Bot. Therefore, Alauthaman et al [10], proposed approach based on TCP features to detect P2P Bots. With purpose of selecting those feature that have high discrimination power they utilized three algorithms. The first selection algorithm w classification and regression tree (CART). The second algorithm is ReliefF which rank the features based on their discrimination between the instances that are close to each other. The third one is the principal component analysis (PCA) in which each component is a linear combination of features that maximize the variance

10- fold validation was used to evaluate neural network with the first ten features from CART, ReliefF and PCA. The features based on the CART and ReliefF gave a high accuracy at around 99.2 and 97.37 respectively, while the features based on PCA gave a lower accuracy at around 91%.

Haddadi et al [11], proposed a method for identifying the most effective attributes for detecting specific botnet types. Two machine learning classifiers were used C4.5 decision tree and the symbolic bid-based (SBB) framework for evolving teams of programs to detect botnet behavior. The classifiers were evaluated using two sets of features: set.1 contains eight flow features and set.2 contains six TCP flag features. set.2 feature set performed better in terms of higher Score and lower FPR.

Since some botnets such as Conficker, Kraken and Torpig evade detection using domain generation algorithm (DGA) in which each bot algorithmically generates a large set of domain names and queries each of them until one of them is resolved and then the bot contacts the corresponding IP-address [12], therefore Haddadi et al [13], proposed an approach to detect HTTP based botnet which uses domain fluxing .They used two machine learning based classifiers: C4.5 and Naïve Bayes. The classifiers performance is evaluated using 10-fold cross validation. Although C4.5 detection rate is 88%, however the FPR is higher than desired in Zeus classification shows that differentiating Zeus traffic form normal traffic is a challenging task.

Venkatesh and Anitha [14], proposed method to identify TCP related features to detect HTTP based botnet. Using neural network, the identification rate for botnets was very high 99%.To confirm that the generated traffic is representative of the publicly available ones and the sandbox ones, Haddadi and Zincir-Heywood [15] , proposed a systematic approach to generate botnet traffic data. They used C4.5 classifier. The lowest true positive rate 79% for Zeus and the highest true positive 100% for Citadel

Yin et al [16], proposed an anomaly identification model based on Genetic Neural Network (GNN). The GNN consolidates the significant global searching function of genetic algorithm with the exact local searching feature of back propagation networks to enhance the initial weights of neural network systems. The identification rate of mixed both genetic algorithm and neural network GNN algorithm was 95.7%, while the identification rate of BP feed forward neural network was 90.3%.

Guntuku et al [17], proposed method to detect P2P bots based on Bayesian Regularized Neural Network (BRNN) which utilizing Bayesian Regularization approach to minimize the overfitting problem. Information Gain Attribute Evaluation was done using the Ranker Algorithm in order to find the most influential features of the entire feature set. The top ranked fifteen features are the input to the BRNN. The BRNN was successful in detecting unseen botnets activity with an accuracy of 99.2 %.

Mathur et al [18], proposed method to differentiate normal network traffic from the botnet traffic regardless of its type. They used CfsSubsetEval as a filter method to choose the most relevant features. Logistic Regression, Random SubSpace, Randomizable Filtered, MultiClass, Random Committee classifiers was used to build a model to make predictions. Logistic Regression and MultiClass Classifier were able to achieve the highest Accuracy of 98.4%.

Beigi et al [19], In their proposed method they used wrapper greedy algorithm that combine backward elimination and forward selection with decision tree classifier. Although their final feature set showed a high detection rate of 99% on a biased dataset containing limited number of botnets, the more truthful detection ability of these features was discovered on a much more diverse set of botnet traces was 75%.

Alejandre et al [20], presented a proposal to detect botnets in the phase of C&C using a genetic algorithm (GA) as optimizer algorithm and a classifier C4.5 to evaluate individuals in the GA. As a result of the interaction of the GA and the algorithm C4.5, the best detection rate was 96.52%.

Narang et al [21], presented a preliminary result of performance comparison for three different feature selection algorithms - Correlation based feature selection, Consistency based subset evaluation and Principal component analysis on three different Machine learning techniques. With the features obtained from each of the algorithm, machine learning models were built using C4.5 algorithm, Naïve Bayes classifier and Bayesian Network classifier. The models were built using 10-fold cross validation technique. The Accuracy and Detection rate obtained with the full feature set is compared with the results for three techniques. Using the full feature set for

classification gave a higher accuracy over the use of reduced number of features.

3. METHODOLOGY

The objective of feature selection is three-fold: improving the prediction performance of the model, providing faster and more cost-effective model [22][24]. The selection of relevant features is crucial for improving the model performance. In literature several feature selection approaches are employed. those approaches can be categorized into three types: wrapping, embedded and ranking. Wrapping approach uses certain search strategy with specific algorithm to select the optimal features. This approach of feature selection is prone to local optimum in addition to time consuming. The embedding approach is utilizing the built-in feature importance calculation in the algorithm such as decision tree. Unfortunately, not all machine learning algorithms have built in feature importance calculation. Feature ranking approach uses certain criteria to sort the features. The feature with high rank is more important than those with lower rank. This approach is not algorithm specific, fast and efficient. Feature selection techniques do not modify the original feature. On other hand, feature transformation techniques modify the original features and generate new feature to make the target labels more separable. We used four ranking approaches to select the relevant features and one transformation technique.

3.1 Feature Variance

Variance is the average of the squared deviations from the mean. The feature variance ranking does not do any correlation with target variable. The feature with high variance has more information than the one with lower variance. After the calculation the variance for each feature, the features are sorted in decreasing order.

3.2 Ranking Using ANOVA F-value

ANOVA F-value checks for and captures linear relationships between each feature and target. F-value is the variation among the group means relative to the variation within the group. A highly correlated feature is given higher F-value and less correlated feature is given lower F-value. After the calculation the F-value for each feature, the features are sorted in decreasing order.

3.3 Ranking Using Mutual Information

Mutual information can capture any kind of dependency between features and target. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. After the calculation the mutual information for each feature, the features are sorted in decreasing order.

3.4 Ranking Based on Literature

The number of usages for each feature by the discussed methods in section II is identified, then the feature that has large number of usages gets more rank than those with small number of usages. After the calculation of the features rank based on literature, the features are sorted in decreasing order.

3.5 Principal Component Analysis (PCA)

PCA is a feature transformation method in which "new" independent variables are created, where each "new" independent variable is a combination of each of the original independent variables. The new variables are ordered by how well they predict the target variable.

In machine learning, there are two types of parameters: those that are learned from the training data, for example, the weights in logistic regression, and the parameters of a learning algorithm that are optimized separately. The latter are the tuning parameters, also called hyperparameters of a model [23]. Hyperparameters is so called because it is a parameter that controls other parameters of the model [25].

The tuning and comparing different parameter settings to further improve the performance for making predictions on unseen data is called model selection, where the term model selection refers to optimal values selection of hyperparameters [23]. The use of test dataset for model selection is not a good machine learning practice, so we use cross validation on training data to evaluate the generalization performance during model selection. After the selection of model, the model is fitted with whole training data and evaluated using unseen test data.

We perform the model selection as following:

- The selected features are divided into three lists: the top five, the top ten and the top fifteen
- We use five algorithms, namely: KNN, Logistic Regression (LR), SVM, Decision Tree (DT), Random Forest (RF).
- For each algorithm we identify the key hyperparameters and the corresponding ranges.
- Randomize search cross validation is performed with top five, top ten and top fifteen features by choosing hyperparameters randomly and apply 10-fold cross validation to obtain the validation accuracy.
- The previous step is iterated ten times
- The best model is the model with highest validation accuracy.

The Gaussian Naïve Bayes (GNB) does not have hyperparameters, so we use 10-fold cross validation only to obtain the validation accuracy. The validation accuracy validates the model generalization, so it gives us indication which model could perform well with unseen test data.

The baseline accuracy is a point of reference to measure how well each model is performing. The more positive gap with baseline accuracy is better. The baseline accuracy is obtained by calculating percentage of majority target class in training data which is 63.27%.

The optimal models are fitted with whole training data then tested with unseen test data. Most of the existing studies are based on a limited number of botnet traces in their datasets. Although these approaches mostly report a high detection rates, obtaining these highly accurate results on a broader dataset is questionable [19]. So, we use broader botnet dataset which created by Canadian Institute for Cybersecurity [26].

The training and testing datasets are in PCAP format, so we extracted the features using CICFlowMeter. The CICFlowMeter is a network traffic flow generator. It generates bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) directions, hence more than 80 statistical network traffic features is extracted. The flow feature generated by CICFlowMeter are described in [26].

4. RESULTS AND DISCUSSIONS

Error! Reference source not found. shows the top fifteen selected features. We choose the top fifteen features because more features will affect the algorithms performance.

Table ii and

Table **iii** show validation and test accuracies for the best models evaluated using top five features. Using variance features KNN algorithm achieved the best validation accuracy 86.98%, but the best test accuracy 80.72% is achieved by Random Forest using ANOVA F-Value features. On other hand, using the literature features the Gaussian Naïve Bayes achieve the worst validation accuracy 37.92% and the worst test accuracy 36.91%. Random Forest and Decision Tree Using the ANOVA F-Value and Mutual Information features got the highest test accuracy.

Table i: The Top Fifteen Features Bold Features are Used in Literature

No	Variance	Anova F-Value	Mutual Information	Literature
1	**Flow_Duration**	Init_Fwd_Win_Byts	Init_Fwd_Win_Byts	Flow_Duration
2	Fwd_IAT_Tot	**Bwd_Pkt_Len_Std**	**Pkt_Len_Std**	Flow_BitsPs
3	**Flow_IAT_Max**	Active_Min	**Pkt_Len_Var**	Flow_BytsPs
4	**Fwd_IAT_Max**	**Fwd_IAT_Min**	Pkt_Size_Avg	Flow_PktsPs
5	**Fwd_IAT_Mean**	Active_Mean	**Pkt_Len_Mean**	DownPUp_Ratio
6	**Flow_IAT_Mean**	**Flow_IAT_Min**	**Flow_IAT_Max**	TotLen_Fwd_Pkts
7	**Fwd_IAT_Min**	**Fwd_IAT_Mean**	**Flow_Duration**	Tot_Bwd_Pkts
8	Idle_Max	**Pkt_Len_Max**	**Dst_Port**	Pkt_Len_Mean
9	Bwd_IAT_Tot	Bwd_Pkt_Len_Max	Fwd_PktsPs	Tot_Fwd_Pkts
10	**Flow_IAT_Min**	**Flow_IAT_Mean**	**Flow_PktsPs**	Pkt_Len_Var
11	Idle_Mean	**SYN_Flag_Cnt**	**Flow_IAT_Mean**	Flow_IAT_Mean
12	Idle_Min	**Fwd_PSH_Flags**	**Fwd_IAT_Max**	PSH_Flag_Cnt
13	Bwd_IAT_Max	**Pkt_Len_Std**	**Fwd_IAT_Mean**	Pkt_Len_Max
14	Fwd_IAT_Std	Active_Max	Bwd_Pkt_Len_Mean	Pkt_Len_Min
15	**Flow_IAT_Std**	Fwd_Seg_Size_Min	**Pkt_Len_Max**	Pkt_Len_Std

Table iv**Error! Reference source not found.** and

Table v show validation and test accuracies for the best models evaluated using top ten features. Using the mutual information features random forest algorithm achieved the best validation accuracy (88.1%), however the best test accuracy (80.93%) is achieved by decision tree using ANOVA F-Value features. KNN, Decision Tree and Random forest were performing very well during validation. Using mutual information features all algorithm test accuracies were higher than baseline accuracy. The worst test performance was with PCA and Literature features. Random Forest and Decision Tree Using the ANOVA F-Value and Mutual Information features got the highest test accuracy.

Table ii Validation Accuracies Based on Top Five Features Bold values are less than the baseline accuracy

	Variance	**ANOVA**	**MUTUAL**	**PCA**	**Literature**
KNN	86.98	84.58	84.59	86.58	73.8
LR	63.87	67.08	64.52	63.23	**62.95**
DR	69.65	81.02	84.91	83.99	72.29
RF	69.49	80.86	84.77	86.72	76.31
GNB	**62.59**	64.16	64.17	**62.21**	**37.92**
SVM	63.71	63.75	**62.95**	**62.95**	**62.95**

Table iii Test Accuracies Based on Top Five Features

Bold values are less than the baseline accuracy

	Variance	**ANOVA**	**MUTUAL**	**PCA**	**Literature**
KNN	69.8	**41.19**	66.4	69.22	**58.43**
LR	71.25	71.23	63.63	65.31	63.27
DR	63.6	79.58	72.63	**57.2**	67.76
RF	**60.22**	80.72	77.31	63.73	64.8
GNB	72.66	65.84	63.56	68.61	**36.91**
SVM	68.166	63.54	63.27	63.27	63.27

Table vi and

Table **vii** show validation and test accuracies for the best models using top fifteen features. Using mutual information features random forest algorithm achieves the best validation accuracy 88.1% and the best test accuracy 81.1. On other hand, using the literature features the Gaussian Naïve Bayes achieve the worst validation accuracy 37.86% and the worst test accuracy 36.87. using Variance and ANOVA F-Value features all algorithm validation accuracy did not fall below the baseline accuracy, however using same features the test accuracies for KNN fall severely below the baseline accuracy. Using mutual information features all algorithm test accuracies were higher than baseline accuracy. The worst test performance was with PCA and Literature lists. Random Forest and Decision Tree Using the ANOVA F-Value and Mutual Information features got the highest test accuracy.

Table iv: Validation Accuracies Based on Top Ten Features Bold values are less than the baseline accuracy

	Variance	**ANOVA**	**MUTUAL**	**PCA**	**Literature**
KNN	85.86	85.76	86	85.84	75.02
LR	63.6	66.9	63.61	62.65	**62.91**
DT	74	85	86.55	81.98	74.72
RF	74.43	87.12	88.18	86.25	77.15
GNB	**62.35**	63.53	**62.64**	**38.83**	**37.84**

SVM	**62.95**	63.73	**62.89**	63.58	**62.99**

KNN	85.88	85.86	86.32	85.88	76.05
LR	63.53	65.55	65.63	63.3	**62.86**
DT	73.35	85.14	86.14	83.5	75.73
RF	74.7	87.85	88.1	86.64	77.67
GNB	63.59	63.4	**62.39**	**39.73**	**37.86**
SVM	63.71	64.5	**62.95**	63.71	63.68

Table v Test Accuracies Based on Top Ten Features Bold values are less than the baseline accuracy

	Variance	ANOVA	MUTUAL	PCA	Literature
KNN	**42.93**	**55.69**	70.44	**42.67**	**36.87**
LR	69.42	71.33	74.08	63.28	63.12
DT	68.26	80.93	75.25	**42.74**	70.2
RF	**60.27**	67.58	78.64	**52.11**	63.52
GNB	72.54	66.97	72.22	**36.29**	**36.82**
SVM	63.27	64.98	63.26	63.16	63.27

Table vii Test Accuracies Based on Top Fifteen Features Bold values are less than the baseline accuracy

	Variance	ANOVA	MUTUAL	PCA	Literature
KNN	**44.97**	**45.35**	73.65	**50.69**	**56.15**
LR	68.4	67.19	74.83	**51.7**	65.96
DR	66.85	73.3	72.73	**41.23**	**55.61**
RF	**62.1**	73.08	81.1	**54.04**	**52.81**
GNB	72.43	67.09	71.36	**54.33**	**36.87**
SVM	63.54	**52.33**	63.27	63.32	64.02

5. CONCLUSIONS

Machine learning optimal models and their performance are empirically evaluated. Botnet dataset with wider number of botnets has been used. Four features ranking methods and one feature transformation have been conducted. Ninety machine learning models have been built by using six machine learning algorithms with different set of flow features, utilizing 10-fold cross validation to avoid variance and tuning the hyperparameters to optimize the model performance.

The results were got by simple techniques which are feature ranking and hyperparameter tuning on datasets having wider botnet types, wider flow features and large data points around 350,000 instances. An interesting finding is that the Random Forest algorithm, which was not given much attention in related work, achieved the best performance (81.1%). On the other hand, The Decision Tree performed well, which is in line with previous research results. Decision Tree and Random Forest gave improved performance with features that have more correlation with target. Our methodology for choosing features and model parameters have given improvement in accuracy over the best reported result in literature by 6%.

Table vi :**Validation Accuracies Based on Top Fifteen Features Bold values are less than the baseline accuracy**

	Variance	ANOVA	MUTUAL	PCA	Literature

6. REFERENCES

[1] "Intrenet security threat report," symantec, April 2016. [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf. [Accessed 23 December 2018].

[2] D. Cid, "Large CCTV Botnet Leveraged in DDoS Attacks," June 2016. [Online]. Available: https://blog.sucuri.net/2016/06/large-cctv-botnet-leveraged-ddos-attacks.html. [Accessed 23 Dec 2018].

[3] Z. Athichart Tangpong, ""Botnet Detection Through Fine Flow Classification," in The Pennsylvania State University, Technical Report CSE11-001, 2011.

[4] J. Stevanovic, "On the use of machine learning for identifying botnet network traffic," Journal of Cyber Security and Mobility, vol. 4, pp. 1-32, 2016.

[5] Z. Chao Li, "Botnet: Survey and Case Study," in Fourth International Conference on Innovative Computing, Information and Control, 2009.

[6] D. Zhao, "Botnet detection based on traffic behavior analysis and flow intervals," Computers & Security, vol. 39, p. 2–16, 2013.

[7] S. Carl Livadas, "Using Machine Learning Techniques to Identify Botnet Traffic," in 31st IEEE Conference on Local Computer Networks, 2006.

[8] C. Strayer W.T., "Botnet Detection Based on Network Behavior," in Botnet Detection Countering the Largest

Security Threat, vol. 36, Boston, MA, Springer, 2008.

[9] H. Sherif Saad, "Detecting P2P Botnets through Network Behavior Analysis and Machine Learning," in Ninth Annual International Conference on Privacy, Security and Trust, 2011.

[10] H. Mohammad Alauthaman, "A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks," Springerlink, 2016. [Online]. Available: Springerlink.com. [Accessed 2018].

[11] H. Fariba Haddadi, "On Botnet Behaviour Analysis using GP and C4.5," in Annual Conference on Genetic and Evolutionary Computation, Vancouver, BC, Canada , 2014.

[12] R. Sandeep Yadav, "Detecting Algorithmically Generated Malicious Domain Names," in 10th ACM SIGCOMM conference on Internet measurement, Melbourne, Australia, 2010.

[13] F. Haddadi, J. Morgan, E. G. Filho and A. N. Zincir-Heywood, "Botnet Behaviour Analysis Using IP Flows: With HTTP Filters Using Classifiers," in 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 2014.

[14] R. irubavathi Venkatesh G., "HTTP Botnet Detection Using Adaptive Learning Rate Multilayer Feed-Forward Neural Network," Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems pp, pp. 38-48, 2012.

[15] N. Zincir-Heywood, "Data Confirmation for Botnet Traffic Analysis," in Lecture Notes in Computer Science, Springer, Cham, 2015.

[16] W. Chunyong Yin, "Botnet Detection Based on Genetic Neural Network," International Journal of Security and Its Applications, vol. 9, pp. 97-104, 2015.

[17] C. Hota, "Real-time Peer-to-Peer Botnet Detection Framework based on Bayesian Regularized Neural Network," CoRR, vol. abs/1307.7464, 2013.

[18] M. R. P. A. Lakshya Mathur, "Botnet Detection via mining of network traffic flow," International Conference on Computational Intelligence and Data Science (ICCIDS 2018), p. 1668–1677, 2018.

[19] E. B. Beigi, H. H. Jazi, N. Stakhanova and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 2014.

[20] F. V. Alejandre, N. C. Cortés and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms," in International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 2017.

[21] H. Pratik Narang, "Feature selection for detection of peer-to-peer botnet traffic," in Compute '13 Proceedings of the 6th ACM India Computing Convention, Vellore, Tamil Nadu, India, 2013.

[22] A. E. Isabelle Guyon, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[23] V. M. Sebastian Raschka, Python Machine Learning Second Edition, Birmingham: Packt Publishing Ltd, 2017.

[24] Aly M. El-Semary, Mostafa G. M. Mostafa. "Distributed and Scalable Intrusion Detection System Based on Agents and Intelligent Techniques," the Journal of Information Processing Systems. Vol. 6, No.4, 481, 2010.

[25] H. Daumé, A Course in Machine Learning, http://ciml.info, 2012.

[26] "Network Traffic Flow Analyzer," 10 Jan 2019. [Online]. Available: http://www.netflowmeter.ca/netflowmeter.html.