Article QR

Title:	**Tree-Based Learning Models for Botnet Malware Classification in Real World Sub-Sample Dataset**

Author (s):	Akinyemi Moruff Oyelakin[1], Rasheed Gbenga Jimoh[2]
Affiliation (s):	[1]Crescent University, Abeokuta, Nigeria [2]University of Ilorin, Ilorin, Nigeria

Tree-Based Learning Models for Botnet Malware Classification in Real World Sub-Sample Dataset

Akinyemi Moruff Oyelakin[1*], and Rasheed Gbenga Jimoh[2]

[1]Department of Computer Science, Crescent University, Abeokuta,Nigeria
[2]Department of Computer Science, University of Ilorin, Ilorin, Nigeria

ABSTRACT The use of machine learning techniques for botnet detection has been an active area of research in security field for some years now. Some of the past machine learning-based botnet detection studies used datasets that were generated synthetically. The release of a large and real-life botnet dataset, named CTU-13, allowed researchers to build machine learning-based models from real-world data. In fact, the real-life traces in the dataset makes it more promising for being used for botnet identification studies. The current study proposed the use of a single tree-based learning algorithm in the classification of botnet evidence from sub-sampled portion of three captures in CTU-13 dataset. Random sub-sampling was used to arrive at three different datasets that was used in the study. The first step in the methodology involved experimental analyses on three captures out of the thirteen in the whole dataset. The analyses revealed the basic characteristics of the datasets which further guided the study. The missing values and categorical data types in the dataset were handled through mixed imputation and feature encoding, respectively. The big data nature of the dataset was handled through random sub-sampling technique with a view to building a botnet detection model that is less computationally intensive. The random sub-sampling technique was used without changing the data distributions in the dataset. The botnet detection models were built by using decision-tree algorithm from the three sub-sampled dataset captures. The performances of the models were evaluated by using accuracy, precision, recall, and f1-score, respectively. In all, the model built with scenario5 capture slightly performed better than the ones built using scenario 6 and scenario 7 captures, respectively.

INDEX TERMS botnet malware, bot communication, malware detection, tree learning algorithms

I. INTRODUCTION

Malware exists in different types and sizes. Cyber attackers use several variants of malware to launch sophisticated attacks in the cyber space. A malware is a kind of software that is used to perform malicious attacks and network intrusions [1]. The common categories of malware include viruses, botnets, worms, ransom ware, and Trojan horses. Botnets have been identified to be the most destructive of these malware types [2]. A botnet is a network of compromised devices that are controlled remotely by a bot herder through the use of command and control server [3],[4]. The attacks that are launched through botnets include Distributed Denial of Service (DDoS), phishing, spyware, spoofing, and spam attacks [5]. Authors in [6] emphasized that the datasets in supervised learning use labeled datasets. In supervised machine learning-based botnet detection studies, one of the key challenges has been the availability of real-world botnet

*Corresponding Author: moruff.oyelakin@cuab.edu.ng

datasets [7]. Thus, the researchers in [7] changed the landscape in botnet detection researches by releasing the CTU-13 dataset. The real-life traces in the dataset makes it more promising for botnet studies [7]as compared to others that are small or synthetically generated. Since, the behavioral patterns identified in the CTU-13 dataset are very complex, a tree-based machine learning algorithmis proposed for the detection of botnets using some identified metrics. Past studies have shown that if there is a high non-linearity and complex relationship between dependent and independent features, a tree model would be better than the linear counterpart. The current study investigated how a tree-based model behaves in the identification of botnet evidence in the chosen captures of the CTU-13 dataset.

Decision tree algorithms are powerful machine learning tools available today and are used in a wide variety of real-world applications [8]. Unlike linear models, tree-based learning algorithms map non-linear relationships quite well. Decision trees are supervised learning algorithms that are used for classification and regression problems [8], [9]. In this study, the variant of decision tree-based algorithm, named CART was used for the classification of botnets because of its popularity for handling non-linearly separable data patterns in a better way. CART was originally proposed by [9] in 1984. The study attempted to address three key issues that were observed in the chosen real-world dataset. Moreover, it also investigated how tree-based botnet detection model with improved performances can still be achieved. For instance, in each of the selected captures, large missing values were handled, the mixed dataset were attended to, while the big data nature were addressed through random sub-sampling.

In each of the three scenarios used in this study, the model learns simple decision rules inferred from the dataset features. The approach used in this study focused to investigate how the tree-based model was able to correctly identify botnet evidence from sub-sample data captures.

II. RELATED STUDIES

Authors in [10] proposed a technique named Host and Network Analysis for botnet detection (HANABot). The authors argued that the scheme works at both, network level and host level, to detect bot evidence. The study used some botnet datasets collected through honeypots and a few were selected from publicly available datasets. However, the dataset was limited in size and may not serve as good representativeness of different samples of bots. Similarly, [11]built a deep learning-based classification model for botnet attack detection. Authors used a deep learning technique for the identification of botnets in the CTU-13 datasets. However, the authors used very low number of network flows in their study. The work is equally silent on the class imbalance issue in the dataset. Most of the machine learning-based models, built for botnet so far, are assumed to be linearly.

Researchers in [12] built tree and ensemble-based botnet detection models by using CTU-13 dataset. The study claimed that the models were then trained and tested on the dataset of different attacks. The study argued that the XGBoost model showed the best performances across the selected metrics.[13] proposed a botnet detection model that used net flow traffic in CTU-13 dataset. The researchers used the reconstruction error that occurred from Auto-encoders, trained with Word2Vec network embedding. However, the author agreed that the results vary for different

captures in the CTU-13 dataset as poor performances were recorded for some datasets. Researchers in [14]came up with a multiclass machine learning-based botnet detection in Software Defined Networks (SDNs). They argued that the current SDNs-based techniques used binary classification to decide if the detected flow belongs to a botnet or not. The authors presented a multiclass machine learning-based approach to address botnet problem in SDNs. The limitation observed in the work is that the ML-based classifiers performed better in the detection of a specific type of botnet.

Moreso [15]built machine learning-based models by using a fractional part of CTU-13 dataset. The current study argued that the methodology adopted performed better than similar existing ML-learning based botnet detection models. [16] proposed a botnet detection model based on network flow summary. Deep neural network was applied to detect botnet by modeling network traffic flow. It was argued that the experimental results are promising when compared to similar studies. However, it was observed that the approach is more computationally intensive.

III. METHOD

This current study used random sub-sampling technique to select five 5%of the total samples in each of the chosen three scenarios in the CTU-13 dataset. All the experimentations were carried out in Python environment. The stages involved in the experimentation carried out in this study include:

(i)   Dataset Collection;

(ii) Dataset Exploratory Analysis,

(iii) Generate random sub sampling of samples in each dataset and then store

(iv) Perform Dataset Pre-processing, and

(v)   Build and evaluate Tree-based Botnet Classification model

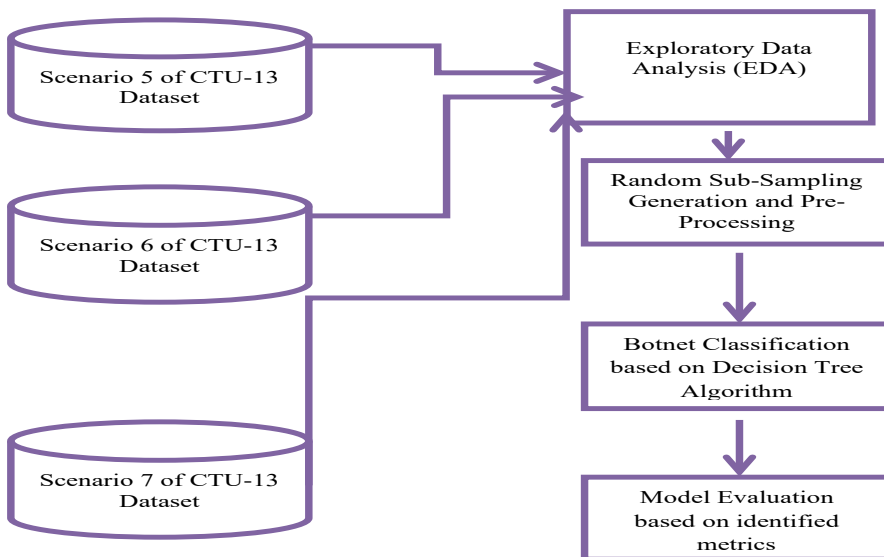The steps in the study are graphically captured as shown in figure 1.

Figure 1. Architecture of the proposed ML-based botnet malware detector

A. CTU-13 DATASET COLLECTION AND DESCRIPTION

1) COLLECTION

The dataset was collected from the stratosphere page: https://www.stratosphereips.org/datasets-ctu13. It is a labeled dataset with botnet, normal, and background traffic. The dataset was originally built at CTU University, Czech Republic.

2) DESCRIPTION OF THE CTU-13 BOTNET DATASET

The CTU-13 dataset was captured at Czech Technical University with a view to advancing researches in botnet detection [7]. The reason for choosing this dataset for the current study was that it comprises real traces, it is very large, and is net flow-based. The dataset is labeled and is contained in different thirteen captures, popularly called scenarios. On each of the captures (scenarios) in CTU-13 dataset, a specific malware, which used several protocols and performed different malicious actions, are included. Each of the scenario file contains the bidirectional netflows. The bidirectional net flow files were generated with Argus, containing all the traffic as well as the labels. The input features contained in the dataset are fourteen in number.

IV. RESULTS

A. RESULTS OF DATA EXPLORATORY ANALYSIS

The current study used three captures out of the thirteen in the dataset. For this reason, the results of the analyses, herein, are shown below. Based on the exploratory analysis of the CTU-13 dataset, it was observed that each of the captures in the dataset contain fourteen input features with hundreds of thousands of instances and millions in some cases. The input features in dataset include: StartTime, Duration, Protocol, SrcAddr, SrcPort, Direction, DstAddr, Destination port, State of the transmission, sTos, dTos, TotPkts, Totbytes, SrcBytes. The target feature is named label which is multi-class in nature.It was also observed that the dataset contains mixed data types as shown in Table I. Therefore, it would be required that any researcher using this dataset to build botnet detection models address the multi-class, mixed data type, and big data issues in the dataset. The mixed data type in the dataset is as captured is shown in Table I.

TABLE I
CATEGORICAL AND NON-CATEGORICAL DATA TYPES IN EACH CAPTURES OF CTU-13 DATASET

Feature	Description	Feature Data Type
StartTime	Start Time of the Netflow	Object
Dur	Duration of the flow	float64
Proto	Protocol	Object
SrcAddr	Source Address	Object
Sport	Source port	Object
Dir	Traffic Direction	Object
DstAddr	Destination Address	Object
Dport	Destination Port	Object
State	State of the flow	Object
sTos	Type of Service from service to source	float64

Feature	Description	Feature Data Type
dTos	Type of Service from destination to source	float64
TotPkts	Total Packets	int64
TotBytes	Total Bytes	int64
SrcBytes	Source Bytes	int64

B. SUMMARY STATISTICS OF NUMERICAL AND CATEGORICAL FEATURES IN THE DATASETS

Furthermore, the exploratory analysis was used to determine the statistical summary of both, the numerical and categorical features in the dataset. Figures 1 to 13 show the statistical summary of the numerical features in the thirteen scenarios of the CTU-13 dataset.

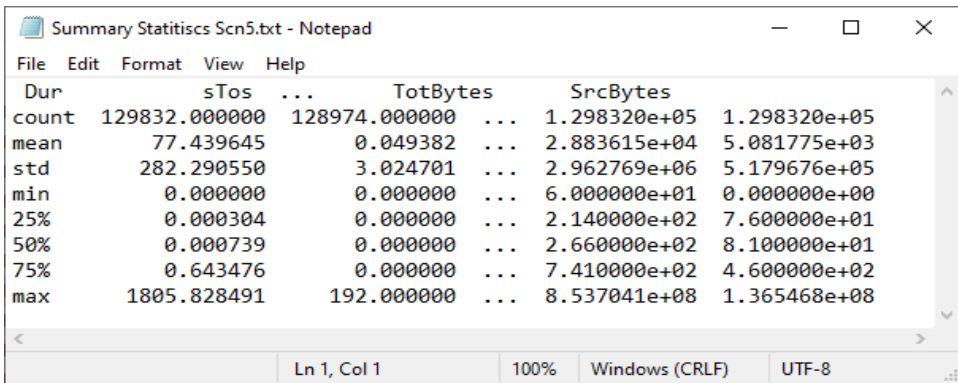### C. STATISTICAL SUMMARY OF NUMERICAL FEATURES

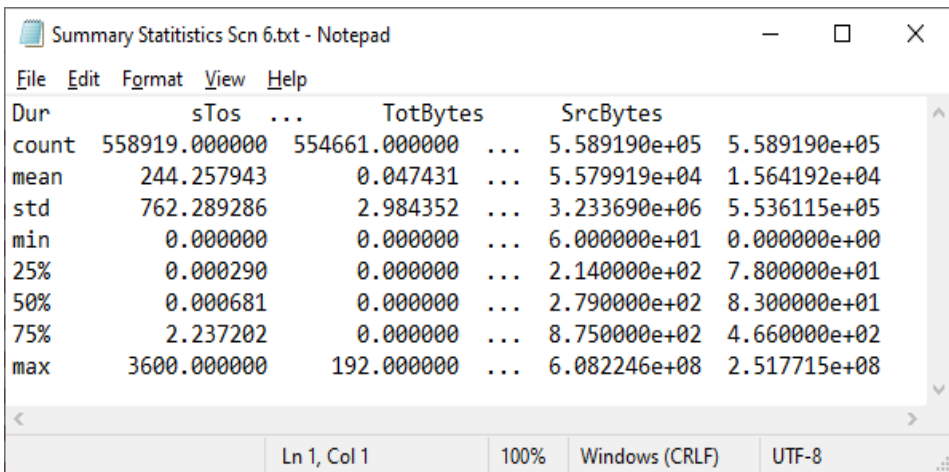**Figure 2.** Statistical summary of numerical features in scenario 5

Figure 3. Statistical summary of numerical features in scenario 6

Figure 4. Statistical summary of numerical features in scenario 7

Figure 5. Statistical summary of categorical features in scenario 5

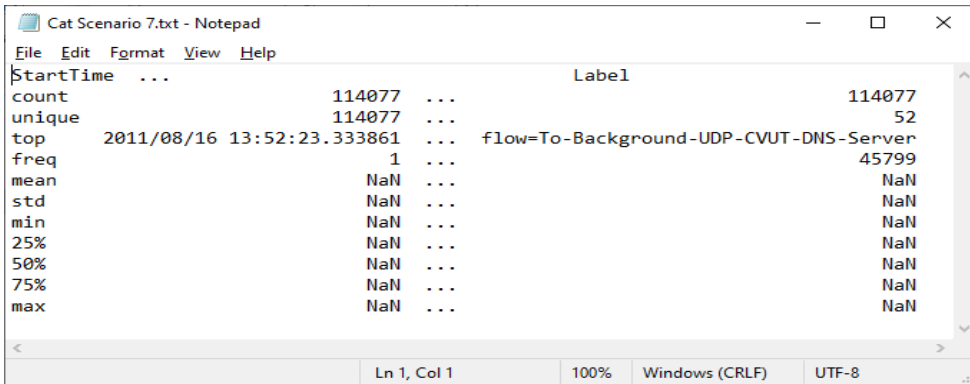**Figure 6.** Statistical summary of categorical features in scenario 6

Figure 7. Statistical summary of categorical features in scenario 7

D. STATISTICAL SUMMARY OF CATEGORICAL FEATURES

The statistical summary of the categorical features in the thirteen captures is also shown in figures 4 to 6.

The statistical summary for both the numerical and categorical features enabled to understand the data distribution in the dataset better and, thus plan to handle them in building the botnet classification model.

E. RESULTS OF DATA PRE-PROCESSING

The data preprocessing focuses on handling the missing values and encoding of the categorical variables.

F. RESULTS OF THE MISSING VALUES HANDLING

TABLE II
SUMMARY OF INSTANCES IN CTU-13 DATASET BEFORE AND AFTER
MISSING VALUES HANDLING

Scenario 1	No of Input features	No of original samples	Total number of missing values	No of reduced samples after deletion
1	14	2,824,636	205,296	2,619,340
2	14	1,808,122	273,815	1,534,307
3	14	4,710,638	534,903	4,166,735
4	14	1,121,076	89,301	1,031,775
5**	14	129,832	7,683	122,149
6**	14	558,919	37,729	521,190
7**	14	114,077	7,637	106,440
8	14	2,954,230	185,759	2,768,471
9	14	2,087,508	181,829	1,905,679
10	14	1,309,791	199,261	1,110,530
11	14	107,251	17,833	89,368
12	14	325,471	30,201	295,270
13	14	1,925,149	147,586	1,777,563

The missing values in the scenario asterisked are the ones handled through imputation. In each of the scenarios, it was observed that the number of missing values is very large and deleting them may be biased to the classification results [17]. Since the dataset contains mixed data (numerical and categorical), 'means imputation' and 'most frequent' strategies are suggested for handling the imputation in the dataset. It is argued, herein that the approach is better than deleting the missing values which may bring bias into the botnet classification results. The dataset captures marked with double asterisks in table 3 are the ones used in this study.

G. ENCODING

Aside the target class, there are other categorical features as well. All the categorical attributes in the selected captures of the dataset were converted into numerical equivalent before being fed in the chosen classification algorithm used for building the model. Label encoding technique was used.

H. RANDOM SUB-SAMPLING OF THE DATASET

The challenges brought by big data are in terms of space and time complexity. It was then argued by [18] that sampling seems to be an important approach for exploring large datasets in any classification problem. Thus, authors in [18] proposed techniques to handle sub-sampling of big data. In this study, the random sub-sampling approach was used to select samples from each of the selected captures and each dataset was used to appropriately detect botnet without losing the statistical significance from the larger dataset. The screenshots of dataframes of random sub-sampling of the dataset are shown in figures 7 to 9.

Figure 8. Dataframe for the sub sampled capture 5
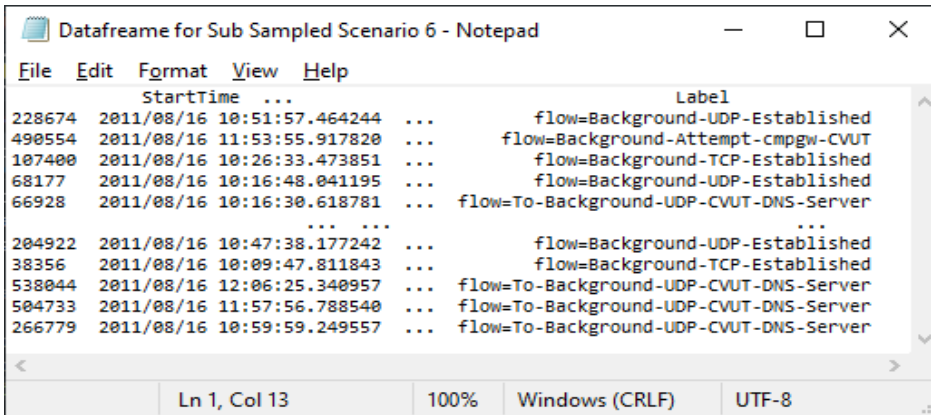
Figure 9. Dataframe for the sub sampled capture 6
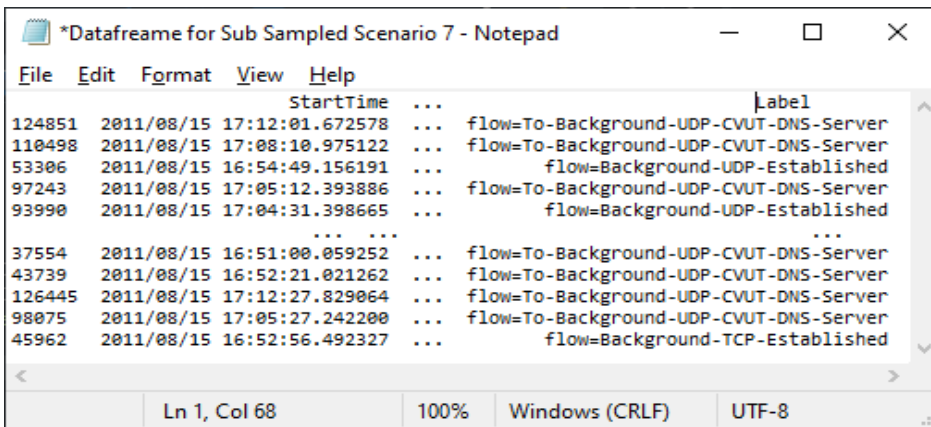
Figure 10. DataFrame for the sub sampled capture 7

TABLE III

SUMMARY OF THE SUB SAMPLED SAMPLES

CTU-Dataset	Actual Samples	New Samples (5% Random Sub Sampling)
Scenario 5	129,832	6,492
Scenario 6	558,919	27,946
Scenario 7	114,077	5,704

The summary of the sub-sampled dataframes for the selected scenarios in CTU-13 dataset is shown in Table 3. With this sub-sampling, the issue of computational complexity, that may arise by using the whole samples for the training and testing of the model, is eliminated.

I. FEATURE SELECTION

The feature space in the dataset is not too large as the input features are fourteen in number. Yet, it is important to still select the most promising features for the training and testing of the botnet detection model. As reported in literature, some of the

attribute selection methods used in tree-based learning models for classification problems include Information Gain, entropy, Gini Index (Gini Impurity), Gain Ratio, and more. The feature selection method used in the current study is Gini Index. Eleven features were later selected out of the fourteen input features based on the promising Gini score.

J. THE BOTNET CLASSIFICATION MODEL

The algorithm used to build the three botnet classification models in this work is named Classification and Regression Trees (CART). It is a variant of decision tree algorithms. The botnet detection models were built from the three separate captures by using split test ratio of 80%-20%, respectively. For every set of input supplied, the target is to achieve a promising model that has good results across the metrics chosen. Several runs were carried out by using different split test ratio. However, the best results were achieved at 80% and 20% as training and test sets, respectively.

K. PERFORMANCE METRICS

The tree-based botnet detection models that were built from the three captures were evaluated by using the metrics, namely: accuracy, precision, recall and f1-score. The results of the classification are shown in Table 4.

TABLE IV

PERFORMANCES OF THE TREE-BASED MODELS

ML Algorithm	CTU-Dataset	Accuracy	Precision	Recall	F1-Score
Decision Tree	Scenario 5	0.9691	0.9685	0.9690	0.9685
Decision Tree	Scenario 6	0.8519	0.8543	0.8520	0.8496
Decision Tree	Scenario 7	0.9531	0.9541	0.9534	0.9532

V. DISCUSSION

The results of the exploratory data analysis (EDA) clearly revealed some of the basic characteristics of the dataset which were informative for the other stages in the study. For instance, the EDA showed that the dataset contains complex patterns; it is made up of real life net-flows with mixed data types and a lot of missing values. The dataset is contained in thirteen different parts called 'captures'. Based on the identified characteristics of the dataset, new dataframes were arrived at through pre-processing. All the missing values in the selected datasets were handled through imputation. The categorical features were encoded and moderate data samples were arrived at based on the random sub-sampling method. The sub-sampling produced sizeable samples that prevent from having computational intensive experimentations. The decision-tree botnet detection models showed brilliant performances across the four metrics used in the study. All the tree-based models, built from the three different scenarios of the dataset, provided promising results in terms of botnet identification. However, the model that was built using scenario 5 showed the best classification performances across the four metrics than those of scenarios 6 and 7. The approach in this study further established that promising results are obtained in the sub-sampled captures of the CTU-13 dataset.

A. CONCLUSION

The current study investigated the strength of a decision tree (DT) algorithm in the detection of botnet in real life net-flow

traces. It was argued that the choice of the decision tree is due to the observations made from the exploratory analysis of the chosen datasets. The study argued that unlike linear models, DT-based models map non-linear relationships quite well as evident in some of the past studies. The ML algorithm was proposed by [9]. Various exploratory analysis tasks in the current study provided good insights to understand the problem in a better way and, thus proposed a better approach to handle the pre-processing and classification issues. The missing values and categorical data types in the dataset were handled through mixed imputation and feature encoding, respectively. The big data nature was also handled through random sub-sampling with a view to building a botnet detection model that is less computationally intensive. The random sub-sampling technique was used without unnecessarily changing the data distributions in the dataset. The performances of the model in botnet identification were reported. The metrics chosen for the evaluation of the model were accuracy, precision, recall, and f1-score, respectively. It was observed that the model built with scenario5 captures slightly performed better than the ones built by using scenario 6 and scenario 7 captures, respectively based on the four metrics. The results showed that the model built from the sub-sampled data captures was able to classify botnet evidence at reasonable level.

REFERENCES

[1] P. Wang, B. Aslam, and C. C. Zou, "Peer-to-peer botnets," in *Handbook of Information and Communication Security*, P. Stavroulakis and M. Stamp, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 335–350. doi: https://doi.org/10.1007/978-3-642-04117-4_18

[2] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, and J. Zhang, "Botnet: classification, attacks, detection, tracing, and preventive measures," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, no. 1, Art. no. 692654, Dec. 2009, doi: https://doi.org/10.1155/2009/692654

[3] A. Zand, G. Vigna, X. Yan, and C. Kruegel, "Extracting probable command and control signatures for detecting botnets," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, Gyeongju Republic of Korea: ACM, Mar. 2014, pp. 1657–1662. doi: http://doi.org/10.1145/2554850.2554896

[4] A. Almomani, "Fast-flux hunter: a system for filtering online fast-flux botnet," *Neural. Comput. Appl.*, vol. 29, no. 7, pp. 483–493, Apr. 2018, doi: https://doi.org/10.1007/s00521-016-2531-1

[5] S. Lagraa, J. François, A. Lahmadi, M. Miner, C. Hammerschmidt, and R. State, "BotGM: Unsupervised graph mining to detect botnets in traffic flows," in *2017 1st Cyber Security in Networking Conference (CSNet)*, IEEE, 2017, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8241990/

[6] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H. T. Lin, *Learning from data*. AML Book, 2012.

[7] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014, doi: https://doi.org/10.1016/j.cose.2014.05.011

[8] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: https://doi.org/10.1007/BF00058655

[9] L. Breiman, *Classification and regression trees*. Routledge, 2017, doi: https://doi.org/10.1201/9781315139470

[10] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P botnet detection scheme based on decision tree and adaptive multilayer neural networks," *Neural. Comput. Appl.*, vol. 29, no. 11, pp. 991–1004, June. 2018, doi:https://doi.org/10.1007/s00521-016-2564-5

[11] A. A. Ahmed, W. A. Jabbar, A. S. Sadiq, and H. Patel, "Deep learning-based classification model for botnet attack detection," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 7, pp. 3457–3466, July. 2022, doi: https://doi.org/10.1007/s12652-020-01848-9

[12] A. R. Vishwakarma, "Network trafficbased botnet detection using machine learning," Master thesis, Dep. Comput. Sci., San Jos State Univ. San Jos, California, 2020. [Online]. Available:https://scholarworks.sjsu.edu/etd_projects/917/

[13] K. Ramström, "Botnet detection on flow data using the reconstruction error from Autoencoders trained on Word2Vec network embeddings," Master thesis, Dep. Info. Technol., Uppsala Univ.,Uppsala, Sweden, 2019. [Online]. Available:https://www.diva-portal.org/smash/record.jsf?pid=diva2:1352441

[14] F. Tariq and S. Baig, "Multiclass machine learning based botnet detection in software defined networks," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 3, p. 150–156, 2019.

[15] S. Harun, T. H. Bhuiyan, S. Zhang, H. Medal, and L. Bian, "Bot classification for real-life highly class-imbalanced dataset," in *2017 IEEE 15th Int. Conf. Depend. Auton. Secur. Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, IEEE, 2017, pp. 565–572. [Online]. Available:https://ieeexplore.ieee.org/abstract/document/8328446/

[16] A. Pektaş and T. Acarman, "Botnet detection based on network flow summary and deep learning," *Int. J. Netw. Manag.*, vol. 28, no. 6, Art. no. 2039, Nov. 2018, doi: https://doi.org/10.1002/nem.2039

[17] M. Swamynathan, *Mastering Machine Learning with Python in Six Steps*. Berkeley, CA: Apress, 2017, doi: http://doi.org/10.1007/978-1-4842-2866-1

[18] J. A. R. Rojas, M. B. Kery, S. Rosenthal, and A. Dey, "Sampling techniques to improve big data exploration," in *2017 IEEE 7th Symposium on Large Data Analysis and Visualization (LDAV)*, IEEE, 2017, pp. 26–35. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8231848/