IDS for Improving DDoS Attack Recognition Based on Attack Profiles and Network Traffic Features

Amer A. Sallam
Faculty of Engineering & Information Technology
Taiz University
Taiz, Yemen
amer.sallam@gmail.com

Muhammad Nomani Kabir*
Faculty of Computing
Universiti Malaysia Pahang
Pahang, Malaysia
nomanikabir@ump.edu.my

Yasser M. Alginahi,
Department of Electrical and Computer Engineering
University of Windsor
Windsor, Canada
alginah@uwindsor.ca

Ahmed Jamal
Faculty of Engineering & Information Technology
Taiz University
Taiz, Yemen

Thamer Khalil Esmeel
Faculty of Computing
Universiti Malaysia Pahang
Pahang, Malaysia

***Abstract*— Intrusion detection system (IDS) is one of the important techniques in security domains of the present time. Distributed Denial of Service (DDoS) detection involves complex process which reduces the overall performance of the system for a large database, and consequently, it may incur inefficiency to the network causing critical failure. In this paper, the attacks database is split into a set of smaller groups by classifying the attack types in terms of the most dominant features that define the profile of each attack along with the sensitive network traffic features. Decision Tree, AdaBoost, Random Forest, K-Nearest Neighbors and Naive Bayes are then used to classify each attack according to their profile features. DDoS attack was considered in all chosen classifiers. It is found that the average classification accuracy with F-measure for the above-mentioned algorithms is 97.24%, 97.21%, 97.20%, 94.77% and 84.70%, respectively, providing plausible results when comparing to other existing models.**

***Keywords*— Signature-Based Intrusion Detection, Anomaly-Based Intrusion Detection, Decision Tree, Naive Bayes, K-Nearest Neighbors, AdaBoost, Random Forest.**

I. Introduction

With the overgrowth of the computer networks in the past decade, the number of Internet users has exceeded 4 billion, and the number of computer devices expected to be 3.5 per capita worldwide in 2021. This growth is continuing rapidly and leads to almost 106 Terabytes per second of global Internet traffic [1]. In parallel to these developments, building a reliable network is not an easy task and is faced with several challenges. These challenges are surrounded with many types of attacks that threaten confidentiality, Integrity and Availability (CIA) of computer networks [2].

Zero-day is the unintended security flaw that has merely learned by the developers and becomes publicly known, but fixing the effect has not been released or its patch is not released yet. During this stage the malware can be dangerous and progressively widespread. Using signature based intrusion detection method to expose such vulnerability or detecting such attacks is often not sufficient. To cope with such problem, an anomaly intrusion detection method has been proposed with a diverseness of classification algorithms to obtain accurate results of detection due to their efficiencies and auto learning abilities. The aim of this study is to appraise some of those algorithms on detecting the most popular attack named Distributed Denial of Service (DDoS).

The DDoS attack is considered as one of the most popular and harmful attacks that continuously deny several services of the end users by consuming network resources and overloading the system with undesired requests [3, 4]. To cope with such attacks, different security mechanisms have been proposed with multi-layered defence approach. Thus, if an attacker bypasses one layer, another layer can stand and prevent that kind of attack to provide robust protection in the network. Among those mechanisms, the Intrusion Detection System (IDS) is considered one of the most common and significant tools. IDS can monitor the whole passing traffic and alert the administrator of any suspicious behavior. Thus, it plays a vital role in minimizing the threat of attacks in a timely manner by installing it on the edge point of a network [5, 6].

However, IDSs are classified into two classes: signature-based IDS (SIDS) and anomaly-based IDSs (AIDS). SIDS can easily identify the signature attributes of certain attacks against the well-known and updated characteristics that are saved in the local database. On the other hand, AIDS can easily recognize “zero-day” attacks and distinguish the behavior of the suspicious profile from the normal ones inside the networks. Therefore, AIDS is considered a better approach to detect DDoS attacks due to its ability on classifying the models of different attacks and detecting the attributes of each unknown attack. To facilitate the automation process of building such models, AIDS uses variety of classification algorithms that can accurately detect various types of attacks [7].

Many studies have been conducted in IDSs field specifically in the Machine Learning domain and the contributions of some of those studies could be summarized as follows: The study in [8], proposes a hybrid architecture through combining two feature selection algorithms including Naive Bayes (BN) and Classification and Regression Trees (CART) in order to improve the performance of intrusion detection system by reducing the number of features that have been used during detection process of the attacks. This study uses the hybrid model on the (KDD cup 99) intrusion detection dataset, the accuracy rates obtained by this study are 100% for normal, 100% for probe, 100% for DoS, 84% for U2R, and 84% for R2L.

The study in [9] attempted to improve the intrusion detection system by using Support Vector Machine (SVM) algorithm on the (DARPA 1998) dataset. However, the

This research was supported by Fundamental Research Grant Scheme (FRGS) No. RDU1901150 from the Ministry of Higher Education, Malaysia and Postgraduate Research Grant Scheme PGRS1903112 from Universiti Malaysia Pahang.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

achieved accuracy results were not as expected: 98% for normal, 88% for probe, 84% for DoS, 0% for U2R, and 18% for R2L. To enhance that result, the authors try to combined SVM with Dynamically Growing Self-Organizing Tree (DGSOT) algorithm to improve the training time of the SVM algorithm. The accuracy rates are: 95% for normal, 91% for probe, 97% for DoS, 23% for U2R, and 43% for R2L.

In [10], Suresh and Anitha, used the chi-square and information gain feature selection mechanisms for selecting the serious attributes in the (CAIDA) dataset using different Machine Learning algorithms such as (NB, SVM, K-Nearest Neighbors (KNN), Fuzzy c-means, K- means clustering, and Decision Tree (DT) is developed for choosing the suitable attribute. The accuracy was achieved as 98.7% for Fuzzy C Means, 97.2% for NB, 96.4% for SVM, 96.6% for KNN, 95.6% for DT, and 96.7% for K-Means. In [11] a Feature Vitality Based Reduction Method (FVBRM) is pro-posed using NB classifier to calculate the execution for the following types of at-tacks: Dos, probe, R2l, and U2R using (NSL-KDD) dataset, the results of this study are represented as: 98.7% for normal, 98.8% for probe, 64% for U2R, and 96.1% for R2L.

In [12], Yassin et al., combined two classification methods, i.e., K-Means clustering (KMC) and NB classifier in order to minimize false alarms while maximizing detection and accuracy rates. The performance of this method was evaluated against (ISCX 2012) Dataset and the accuracy of detection rate was significantly improved (99% for KMC and 98.8% for NBC) while decreasing the false alarm to 2.2%. In this method, the data can be accurately categorized, except for the types of L2L and R2L attacks. The research in [13] implemented some common Machine Learning algorithms such as RF, C5.0, NB, and SVM to evaluate (CICIDS 2017) dataset. as a result, the RF and C5.0 classifiers surpass the others with average accuracy of 86.80%, 86.45% respectively and the obtained precision was 99%. The false positive rate of RF and C5.0 was 0.050%, 0.046% respectively and the maximum of the false positive rate was 75% with SVM. In contrast, the SIDSs method detects the attacks that only exist in its database. However, this method is quite successful, but the data-base needs to be frequently updated and the information of new attacks should be processed in advance, in order to be recognized. Otherwise, the new attacks will remain unknown and will not be detected. AIDS is more efficient in overcoming the above mentioned problem and more effective in protecting the entire system against any kind of suspected behavior [14 – 20] and it is also possible to perform other kinds of decision making under uncertainty with the help of Machine Learning algorithms. Those algorithms can be build using mathematical models based on the samples, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task [21-28].

There are numerous intrusion detection datasets available in computer networks. The most popular datasets are DARPA [29-31], KDD99 [32, 33], CAIDA [34,35] and ISCX [36]; however, all exist with some shortcomings. In this study, five classification algorithms (i.e., DT, NB, KNN, AdaBoost and RF) are used to detect the DDoS attacks, then, their performance are compared and evaluated using labelled datasets (CICIDS2017) [37] that include the possible scenarios of the targeted attack.

II. Methodology

In this framework, different pre-processing and actual application are performed to detect DDoS attack in computer network using Machine Learning methods. The input instances of attacks are segmented and the data cleansing technique is used for cleaning up the dataset from any form of errors or defects such as blanks, duplication and non-numeric values. The proposed framework provides two modes to be selected as Included or Excluded Mode. Then, the dataset is divided into two parts: training, and testing part. Fig. 1 illustrates the proposed framework which follows the standard procedure s of training and testing phase based on the selected mode.

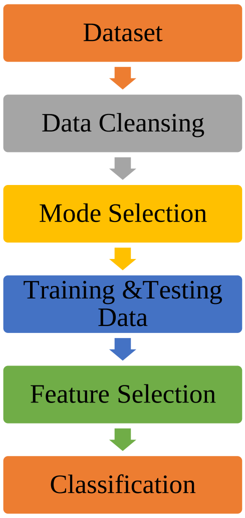

Fig. 1. The phases of the proposed framework.

The process of features selection will be triggered to be used by the given classifiers. Finally, the performance of each classifier is evaluated. The components of the pro-posed framework are explained as follows:.

A. Dataset

The chosen dataset (CICIDS2017) [37] contains 3119345 instances and the distribution of these instances is described in Table I.

TABLE I. THE DISTRIBUTION OF THE INSTANCES IN THE CICIDS2017 DATASET

Label Name	Instances
Benign(healthy)	2359289
Faulty with errors (blanks)	288602
DoS Hulk	231073
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS Slowloris	5796
DoS SlowHTTPTest	5499
Bot	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652
Infiltration	36
Web Attack – SQL Injection	21
Heartbleed	11

CICIDS dataset was created by the Canadian Institute Cybersecurity [37] and it is chosen in this research as a benchmark dataset for the following reasons: it is up-to-date dataset, it provides wide attack diversity, and it contains various network protocols (e.g. HTTP, HTTPS, FTP, SSH, and Mail services). This dataset represents real data (PCAPs) and contains more than 80 features that define the network traffic analysis labelled with flows based on the time stamp, source IP, destination IP, source port No., destination port No., and other protocols.

B. Data Cleansing

In order to achieve more accurate and desired results, it is necessary to clean up the dataset from any form of errors or defects that may exist within the chosen dataset (CICIDS2017) as possible. Thus, the whole instances of dataset are examined in the cleansing phase and 288602 out of 3119345 instances found with error (empty) as shown in Table I. Those empty instances filled with zeros to be accepted for further processing. The feature in the column 41st named 'Fwd Header Length' found to be duplicated in column 62nd, and the duplicated feature (error) has been removed.

Another change, that needs to be done in the dataset, is to convert the categorical and string values of the network traffic features (i.e., Flow ID, Source IP, Destination IP, Timestamp and External IP) into numerical values to be accepted for use in Machine Learning algorithms. This can be done with LabelEncoder from Sklearn classes. In this method, various string values can be converted into integer values between 0 and n-1 and will become more suitable for further processing [38]. However, although the 'Label' tag is a categorical feature, but it doesn’t need to be change. This is because the original categories are required during the processing steps to classify the types of attacks in different forms.

Finally, some minor structural changes should be made to the dataset, including:

• The Label feature, the character "-" that used to identify the web attack types must be replaced with the character "_” because the default codec of Pandas library that is used in this work does not recognize it.

• "Flow Bytes/s", "Flow Packets/s" features include the "Infinity" and "NaN" values are adjusted to -1 and 0 respectively to make them suitable for further processing.

C. Mode Selection

The proposed framework provides two modes to be selected as follows:

Included Mode: this mode is used to deal with specific type of attack features as required. Based on the selected mode, the process of features selection will be triggered and sorted based on their weights to be used by the given classifiers. All features of targeted attack along with the healthy instances are saved as a dataset for this mode in one single file.

Excluded Mode: in this mode the all instances of the entire dataset which represent the various types of attacks are selected and saved in separate files. Each data in these files are named by the type of attack it contains.

D. Training and Testing Data

The CICIDS2017 dataset does not have separate files dedicated for training or testing phases; however, it contains a single unbundled dataset. Therefore, the dataset needs to be cleansed and partitioned into training and testing data. So that, the classifier algorithm gets to know the training data and acquires the required knowledges to apply it on the testing data for building more reliable models for each attack.

For this purpose, train-test_split [39] [40] is used. Sklearn command used to divide the dataset into two parts according to the sizes specified by the user. Generally preferred partitioning [39] is 20% for testing and 80% for training. The train-test_split command does the random selection when creating data groups. This process is known as cross-validation.

E. Feature Selection

The dataset is evaluated to determine which features are important to define the attack. The feature selection step is applied for both modes (Included and Excluded Mode) in order to determine the most important features for each mode. The Random Forest Regressor (RFR) algorithm [41] is used to calculate and select the most dominant weighted features. This algorithm is used to create a decision-forest in order to construct the decision-tree. When the process is finished, the features with important weights are compared and sorted accordingly. The total of the important weight in the decision tree is given based on the sum of the weights for all the selected features. The information about the importance of any feature in the decision tree is given by the comparison of the weight of any feature to the weight of the whole tree. However, the network traffic features (Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, External IP) should be excluded when the importance weight is calculated. Because, it is likely that the attacker would prefer not to use well-known ports to escape control or use generated / fake IP addresses. Also, many ports are used dynamically, and many applications are transmitted over the same port. Therefore, the use of the port number may be misleading. In this context, during the determination of the leading attributes that define the attack profile, the more generic and invariant attributes should be chosen, because having much more information about the profile makes it easier to decide whether the suspected attack is recognized or not. Thus, all the features that have been processed during the pre-processing stage and contained the entire streams that identify both the attack information and the data for both modes, have been randomly selected as 30% attack and 70% benign.

TABLE II. THE MOST SIGNIFICANT FEATURES THAT DEFINES DDOS ATTACK IN INCLUDED MODE

Features	Importance
Bwd Packet Length Std	0.469963
Total Backward Packets	0.094493
Fwd IAT Total	0.013731
Total Length of Fwd Packets	0.007831
Flow Duration	0.006176
Flow IAT Min	0.005831
Flow IAT Std	0.005461
Flow IAT Mean	0.005329

As mentioned so far, RFR is used to cover the process of feature selection for both Included and Excluded Modes. In Included Mode, the selection process is only applied on the specific part of the dataset that represents the DDoS attributes in particular along with healthy instances, so the features of other attacks are neglected, while the instances of the entire dataset are selected for the Excluded Mode. As a result, the distributions of the most significant features that defines

DDoS profile in both Included and Excluded modes are illustrated in Tables II and III respectively.

TABLE III. THE MOST SIGNIFICANT FEATURES THAT DEFINES DDOS ATTACK IN EXCLUDED MODE

Features	Importance
Bwd Packet Length Std	0.246627
Flow Bytes/s	0.178777
Total Length of Fwd Packets	0.102417
Fwd Packet Length Std	0.063889
Flow IAT Std	0.009898
Flow IAT Min	0.006946
Fwd IAT Total	0.005121
Flow Duration	0.004150

F. *Classification*

Five classifiers, namely, KNN [25], NB [25], DT [26], RF [27] and AdaBoost [28] were used to classify the features selected from previous step. Each classifier is separately used to work on each mode and its selected features. The performance of each classifier using different parameters are evaluated and discussed in the next section.

III. RESULTS AND DISCUSSION

CICIDS2017 dataset is used for each mode, 80% of the instances are taken for training and the rest of the instances were taken for testing. The results are then evaluated depending on four criteria namely: accuracy, precision, recall and f-measure [31], shown in equations (1) – (4). All these criteria take a value between 0 and 1.

Accuracy: is the ratio of correctly classified data to total data.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

Precision: is the ratio of correctly classified data as the attack to total data classified as the attack.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

Recall (Sensitivity): is the ratio of correctly classified data as the attack to total attack in data.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$F_{measure}$: is the weighted average of Precision and Recall.

$$F_{measure} = 2 * \frac{Recall*Precision}{Recall+Precision} \tag{4}$$

In calculating these four items, the four values summarized below are used:

True Positive (TP): The abnormal data classified as attack.

False Positive (FP): The normal data classified as attack.

False Negative (FN): The abnormal data classified as normal data.

TABLE IV. THE CONFUSION MATRIX

		Actual Class	
		Attack	Normal
Predicted Class	Attack	TP	FP
	Normal	FN	TN

This distribution is presented by the confusion matrix in Table IV. Extensive experiments were carried out for each feature selection algorithm with each classifier on each mode and the detection classification results are provided in Tables V and VI.

TABLE V. ACCURACY OF DIFFERENT CLASSIFIERS WITH INCLUDED MODE

Algorithms	Accuracy	Precision	Recall	F-measure	Time(s)
Random Forest	96.70	99.29	95.98	97.61	6.6974
Decision Tree	95.94	99.85	94.33	97.02	0.4442
AdaBoost	97.34	99.67	96.52	98.07	12.7102
K Nearest Neighbors	93.01	96.52	93.38	94.92	6.0036
Naive Bayes	80.94	84.54	89.07	86.74	0.3151

TABLE VI. ACCURACY OF DIFFERENT CLASSIFIERS WITH EXCLUDED MODE

Algorithms	Accuracy	Precision	Recall	F-measure	Time(s)
Random Forest	94.67	94.09	99.87	96.89	871.4547
Decision Tree	94.68	94.20	99.76	96.90	9.3605
AdaBoost	94.02	94.50	98.55	96.48	362.8278
K Nearest Neighbors	92.21	92.01	96.23	95.11	586.3621
Naive Bayes	85.28	88.30	94.88	91.48	4.643

TABLE VII. COMPARISON OF THE PERFORMANCE OF INCLUDED MODE AGAINST THE EVALUATION CRITERIA WITH OTHER STUDY BY ABDULRAHMAN ET AL. [13]

Algorithms	**Proposed Method**		**Abdulrahman et al [13]**	
	F-Measure	**Accuracy**	**F-Measure**	**Accuracy**
Random Forest	97.61	96.70	92.48	86.80
Decision Tree	97.02	95.94	92.33	86.46
Naive Bayes	86.74	80.94	88.00	79.97
K Nearest Neighbors	94.92	93.01	-	-
AdaBoost	98.07	97.34	-	-

The final results for accuracy of different classifiers with Included Mode and Excluded modes are listed in Tables V and VI, demonstrate that the proposed method achieves a high degree of accuracy. The best result is obtained for features selected in Included Mode, 97.34% for AdaBoost. On the other hand, Decision Tree provides the best result (i.e., 94.68%) for Excluded Mode. It reveals that the selection of specific classification approach does not affect the recognition rate much. Feature selection approach is the main reason behind the increase in recognition rate of the system. The features of Included Mode are demonstrated to be more accurate than the features of *Excluded Mode* because of the robustness of its profile features which are invariant to targeted attack (i.e., DDoS), and this may not be significantly invariant among all types of attacks in Excluded Mode. For example, there are a number of samples in Included Mode which may not overlap with those in features group of Excluded Mode in terms of DDoS attack detection. It was also found that the average classification accuracy and the F-measure of the proposed method obtained plausible results compared to existing models (i.e., Abdulrahman et al. [13]) as shown in Tables VII and VIII.

TABLE VIII. COMPARISON OF THE PERFORMANCE OF EXCLUDED MODE AGAINST THE EVALUATION CRITERIA WITH OTHER STUDY BY ABDULRAHMAN ET AL. [13]

Algorithms	Proposed Method		Abdulrahman et al [13]	
	F-Measure	Accuracy	F-Measure	Accuracy
Random Forest	96.89	94.67	92.48	86.80
Decision Tree	96.90	94.68	92.33	86.46
Naive Bayes	91.48	85.28	88.00	79.97
K Nearest Neighbors	95.11	92.21	-	-
AdaBoost	96.48	94.02	-	-

IV. CONCLUSION AND FUTURE WORK

In this paper, an attack classification method using features selection analysis of both attack modes, Included and Excluded, is presented. Tests on instances obtained from CICIDS2017 attacks database that contains more than 80 features which define various network protocols and wide attack diversity are conducted and experiments using 3119345 instances are carried out. The implementation goes through different stages, starting with pre-processing then data cleansing, mode selection, training and testing data, feature selection, and classification. Two modes are proposed, first mode to deal only with DDoS attack features along with healthy instances. While the second mode, feature selection is applied to the entire dataset instead of the only DDoS attack. The RFR algorithm is used as a feature selection technique in order to elect the most dominate and important features based on the weight calculation of each mode. Finally, five different classification algorithms with different qualities viz. DT, NB, KNN, AdaBoost and RF are used to classify such features in order to detect Zero-day attack and to obtain more accurate results of detection due to their efficiencies and auto learning abilities. The performance of each algorithm is evaluated based on F-Measure, Accuracy, Recall and Perception. The results show that AdaBoost is the most successful DDoS detector for the first mode (97.34%), while Decision Tree is the best for the second mode (94.68%). However, the results for the first mode is shown to be more accurate. It is also found that the average classification accuracy and the F-measure of the proposed method achieved plausible results compared to some existing models. Unfortunately, this method is not practically viable in real-time systems. However, such problem can be solved by adapting modules that can catch the real-time network data on the fly and make it workable with the Machine Learning algorithms.

REFERENCES

[1] Miniwatts Marketing Group. "Internet Growth Statistics," Available: https://www.Internetworldstats.com/emarketing.htm. [Accessed December 2 2019].

[2] M. N. Kabir and Y. Alginahi, "Introduction," in Authentication Technologies for Cloud Computing, IoT and Big Data: Institution of Engineering and Technology, 2019, ch. 1, pp. 1-12.

[3] S. Shamshirband, N. B. Anuar, M. L. M. Kiah, and A. Patel, “An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 9, pp. 2105-2127, 2013.

[4] H. A. S. Ahmed, and M. F. B. Zolkipli, “Data Security Issues in Cloud Computing,” International Journal of Software Engineering and Computer Systems, vol. 2, no. 1, pp.58-65, 2016.

[5] H. Alginahi, Y. Alginahi, M. N. Kabir, “Data Protection Laws,” in Authentication Technologies for Cloud Computing, IoT, and Big Data: Institution of Engineering and Technology, 2019, ch. 12, pp. 309-337.

[6] T. Shon, and J. Moon, “A hybrid Machine Learning approach to network anomaly detection,” *Information Sciences*, vol. 177, no. 18, pp. 3799-3821. 2007.

[7] O. E. Elejla, M. Anbar, B. Belaton, and B. O. Alijla, “Flow-Based IDS for ICMPv6-Based DDoS Attacks Detection,” *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp.7757-7775, 2018.

[8] S. Chebrolu, A. Abraham, J. P. Thomas “Feature deduction and ensemble design of intrusion detection systems,” *Computers & Security*, vol. 24, no. 4, pp. 295-307, 2005.

[9] L. Khan, M. Awad, and B. Thuraisingham, “A new intrusion detection system using support vector machines and hierarchical clustering,” *The VLDB Journal*, vol. 16, no. 4, pp. 507-521, 2007.

[10] M. Suresh, and R. Anitha. “Evaluating Machine Learning Algorithms for Detecting DDoS Attacks,” In International Conference on Network Security and Applications, 2011, pp. 441-452.

[11] S Mukherjee, and N Sharma, “Intrusion Detection using Naive Bayes Classifier with Feature Reduction,” *Procedia Technology*, vol. 4, pp. 119-128, 2012.

[12] W. Yassin, N. I. Udzir, Z. Muda and M. N. Sulaiman, “Anomaly-based intrusion detection through k-means clustering and naives bayes classification,” In Proc. 4th Int. Conf. on Comput. Informatics, ICOCI, no. 49, 2013, pp. 298 – 303.

[13] A. A. Abdulrahman, “Evaluation of DDoS attacks Detection in a New Intrusion Dataset Based on Classification Algorithms,” *Iraqi Journal of Information and Communications Technology*, vol. 1, no. 3, pp. 49-55, 2018.

[14] O. Depren, M. Topallar, E. Anarim, M. K. Ciliz, “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks,” *Expert systems with Applications*, vol. 29, no. 4, pp. 713-722, 2005.

[15] M. Ahmed, A. N. Mahmood, J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016.

[16] S. Acharya, N. Tiwari, “Survey of DDoS attacks based on TCP/IP protocol vulnerabilities,” *IOSR Journal of Computer Engineering*, vol. 18, no. 3, pp. 68-76, 2016.

[17] A. A. Ahmed, “Investigation model for DDoS attack detection in real-time,” International Journal of Software Engineering and Computer Systems, vol. 1 no. 1, pp.93-105, 2015.

[18] S. Haris, R. Ahmad, and M. Ghani, “Detecting TCP SYN flood attack based on anomaly detection,” In proc. Second International Conference on Network Applications, Protocols and Services, 2010, pp. 240 – 244.

[19] R. V. Deshmukh, K. K. Devadkar, “Understanding DDoS attack & its effect in cloud environment,” *Procedia Computer Science*, vol. 49, pp. 202-210, 2015.

[20] H. Mukhtar, K. Salah, and Y. Iraqi, “Mitigation of DHCP starvation attack,” *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1115-1128, 2012.

[21] G. Hackeling, *Mastering Machine Learning with scikit-learn*. Birmingham, UK: Packt Publishing Ltd, 2017.

[22] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine Learning: Algorithms and Applications*, Florida, USA: Crc Press, 2016.

[23] R. Sathya, A. Abraham, “Comparison of supervised and unsupervised learning algorithms for pattern classification,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34-38, 2013.

[24] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-Supervised Learning,” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542-542, 2009.

[25] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised Machine Learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3-24, 2007.

[26] K. Kaur, D. Bhutani, “A review on classification using decision tree,” *IJCAT-International Journal of Computing and Technology*, vol. 2, no. 2, pp. 42 – 46, 2015.

[27] Stack Abuse. “Random forest algorithm with python and scikit learn,” Available: https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/. [Accessed April 9 2019].

[28] R. E. Schapire, “The boosting approach to Machine Learning: An overview,” *Nonlinear estimation and classification*. New York, NY, USA: Springer, p. 149-171. 2003.

[29] MIT Lincoln Laboratory, “1998 DARPA Intrusion Detection Evaluation Data Set,” Available: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-data-set. [Accessed March 05, 2019].

[30] C Brown, A Cowperthwaite, A Hijazi and A. Somayaji, “Analysis of the 1999 darpa/lincoln laboratory ids evaluation data with netadhict,” In Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1-7.

[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani. “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,” In Proc. International Conference on Information Systems Security and Privacy, 2018, pp. 108 – 116.

[32] University of California, “KDD Cup 1999 Data,” Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Accessed 20 Mar 2019].

[33] A Özgür, H Erdem, “A review of KDD99 dataset usage in intrusion detection and Machine Learning between 2010 and 2015”, in PeerJ Preprints, [online document], 2016. Available: https://peerj.com/preprints/1954.pdf. [Accessed June 12, 2019].

[34] Center for Applied Internet Data Analysis. CAIDA OC48 Peering Point Traces dataset, Available: https://www.caida.org/data/passive/passive_oc48_dataset.xml. [Accessed March 25, 2019].

[35] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “An Evaluation Framework for Intrusion Detection Dataset,” In Proc. International Conference on Information Science and Security (ICISS), 2016, pp. 1-6.

[36] Canadian Institute for Cybersecurity, “UNB Intrusion detection evaluation dataset (ISCXIDS2012),” Available: http://www.unb.ca/cic/datasets/ids.html. [Accessed March 27 2019].

[37] Canadian Institute for Cybersecurity, “UNB. Intrusion Detection Evaluation Dataset (CICIDS2017),” Available: http://www.unb.ca/cic/datasets/ids-2017.html. [Accessed March 29 2019].

[38] Scikit Learn, Support Vector Machines - scikit-learn 0.19.1 documentation, “sklearn.preprocessing.LabelEncoder,” Available: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html. [Accessed May 19 2019].

[39] A. Géron, *Hands-on Machine Learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*," Champaign, IL, USA: O'Reilly Media, Inc., 2017

[40] Scikit Learn, scikit-learn 0.19.1 documentation, rain_test_split, “sklearn.model_selection.train_test_split,” Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html. [Accessed November 20, 2019].

[41] J. Brownlee. “Feature Importance and Feature Selection With XGBoost in Python,” Available: https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/. [Accessed December 21, 2019].