Mehran University Research Journal of Engineering and Technology
Vol. 40, No. 1, 215- 229, January 2021
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.2101.19

Machine Learning Classification of Port Scanning and DDoS Attacks: A Comparative Analysis

Muhammad Aamir[1a], Syed Sajjad Hussain Rizvi[1b], Manzoor Ahmed Hashmani[2], Muhammad Zubair[3], Jawwad Ahmad[4]

RECEIVED ON 05.05.2019, ACCEPTED ON 29.11.2019

ABSTRACT

Cyber security is one of the major concerns of today's connected world. For all the platforms of today's communication technology such as wired, wireless, local and remote access, the hackers are present to corrupt the system functionalities, circumvent the security measures and steal sensitive information. Amongst many techniques of hackers, port scanning and Distributed Denial of Service (DDoS) attacks are very common. In this paper, the benefits of machine learning are taken into consideration for classification of port scanning and DDoS attacks in a mix of normal and attack traffic. Different machine learning algorithms are trained and tested on a recently published benchmark dataset (CICIDS2017) to identify the best performing algorithms on the data which contains more recent vectors of port scanning and DDoS attacks. The classification results show that all the variants of discriminant analysis and Support Vector Machine (SVM) provide good testing accuracy i.e. more than 90%. According to a subjective rating criterion mentioned in this paper, 9 algorithms from a set of machine learning experiments receive the highest rating (good) as they provide more than 85% classification (testing) accuracy out of 22 total algorithms. This comparative analysis is further extended to observe training performance of machine learning models through k-fold cross validation, Area Under Curve (AUC) analysis of the Receiver Operating Characteristic (ROC) curves, and dimensionality reduction using the Principal Component Analysis (PCA). To the best of our knowledge, a comprehensive comparison of various machine learning algorithms on CICIDS2017 dataset is found to be deficient for port scanning and DDoS attacks while considering such recent features of attack.

Keywords: Classification, DDoS Attacks, Machine Learning, Port Scanning, Supervised Learning.

1. INTRODUCTION

Port scanning and DDoS attacks are very common techniques of cyber attackers to scan for vulnerabilities and exhaust the resources of a target respectively. When port scanning is in process, a scanning tool identifies the open ports in the target, informs about the running services, and enumerate in the form of target's status such as operating system in use, memory occupied, and processing information. The aim of port scanning is to identify the vulnerable areas of a target resource through which the exploitation may be possible. According to Oracle®, the anatomy of a typical attack involves five steps i.e. reconnaissance, enumeration, penetration, exfiltration and sanitation [1].

[1] Shaheed Zulfikar Ali Bhutto Institute of Science & Technology (SZABIST), Karachi, Pakistan
Email: [a]aamir.nbpit@gmail.com (Corresponding Author), [b]dr.sajjad@szabist.edu.pk
[2] Department of Computer and Information Sciences, High Performance Cloud Computing Center (HPC3), Universiti Teknologi PETRONAS, Seri Iskander, Malaysia. Email: manzoor.hashmani@utp.edu.my
[3] Iqra University, Karachi, Pakistan. Email: zubair@iqra.edu.pk
[4] Usman Institute of Technology, Karachi, Pakistan. Email: jawwad@uit.edu

After an attacker gathers sufficient information such as IP scheme, datacenter locations and target profile during the reconnaissance phase, port scanning is involved as an early stage of the enumeration step. NMAP is used as the most common tool of port scanning worldwide [2]. A common example is scanning for ports supporting the Transmission Control Protocol / Internet Protocol (TCP/IP) traffic. A scanner sends the synchronization (SYN) signal to the target. If target responds with Synchronization & Acknowledgment (SYN+ACK) signal, it means that the port is open. Now in order to close the connection after knowing that the port is open, the scanner may send rest (RST) signal. On the other hand, leaving the connection open may drive this scenario to a kind of denial of service attack known as TCP SYN attack. This is shown in Fig. 1.

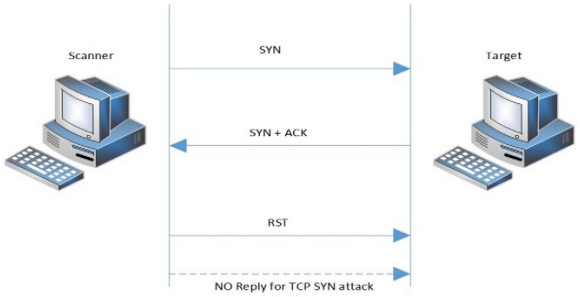

Fig. 1: Scanning for TCP ports.

DDoS attacks are another major area of cyber security concern where attackers are able to flood the target (also known as the victim) with malformed traffic to exhaust its resources in such a way that it is impossible for the victim to respond to legitimate requests. DDoS is a huge problem due to the fact that it is possible to create denial of service at every layer of the Open Systems Interconnection (OSI) communication model [3]. According to the statistics from *akamai.com* for Summer-2018 [4], an increase of 16% is observed in DDoS attacks when compared with the attacks of the last Summer-2017. In a traditional scenario of DDoS attack, an attacker has a number of machines under control which are compromised as a result of some exploitation such as malware. Such compromised machines are called zombies and they are the part of a malicious network called botnet. The zombies are used to directly attack the victim. Some high powered machines in the botnet are also selected as the handlers of zombies which are used to pass the attacker's instructions on to the zombies. The attacker's server runs the Command & Control (C&C) function which is the direct instruction of attack [5]. A typical scenario of DDoS attack is shown in Fig. 2. DDoS attack can also adjust the rate of malicious traffic sent to the victim according to the capacity of network's bandwidth. Therefore, when DDoS attack is modified as per the available network, bandwidth, and the nature of traffic acceptable on the network, it can be applied on traditional as well as more recent types of networking [6].

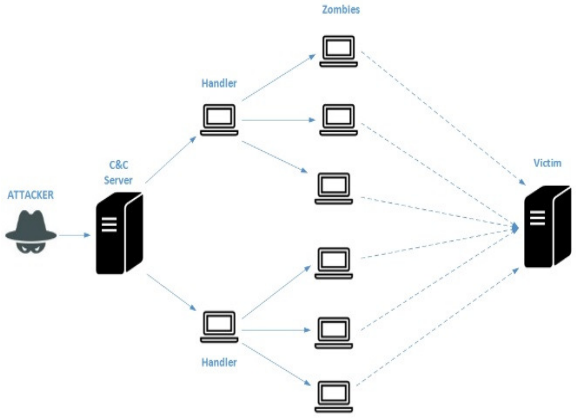

Fig. 2: Typical scenario of DDoS attack

Machine learning is one of the major tools of today to find optimal solutions of numerous real-world problems through soft computing approach. Cyber security is not an exception and researchers are using several techniques based on machine learning to address the issues of cyber attacks and find solutions for intrusion detection. The machine learning is mainly classified in two forms i.e. supervised learning and unsupervised learning. When data is available with labeled target values, supervised learning methods are used to find the response for new data with unknown output. The supervised learning is also divided into two major components i.e. classification/prediction and optimization. In the former category, the classification is made for discrete output whereas the prediction is associated with continuous output. The latter category corresponds to the machine learning approaches with evolutionary computing either for final outputs or for various other supporting solutions such as feature selection [7].

In this paper, supervised machine learning is used for intrusion detection of port scanning and DDoS attacks. The classifications are obtained in a mix of normal and

attack traffic. Different machine learning algorithms are trained and tested on a recently published benchmark dataset to identify the best performing algorithms on the data which contains more recent vectors of port scanning and DDoS attacks. The dataset used in this research is CICIDS2017 which is an intrusion detection dataset created in 2017 and published in 2018 by Canadian Institute for Cybersecurity (CIC) [8]. The relevant category of attacks has been taken from the Friday – working hours' scenarios of CICIDS2017. CIC is also the creator of other benchmark intrusion detection datasets such as NSL-KDD 2009 and ISCXIDS2012. The reason for selection of CICIDS2017 in this research is the fact that it contains more recent vectors of attacks as compared to the older techniques which are not being exercised by attackers with noticeable frequencies these days. The other machine learning based comparative analyses are available in research on older benchmark datasets of intrusion detection. On the other hand, the contributions of this paper are:

- Comparing different machine learning algorithms using a recent benchmark dataset of intrusion detection with focus on port scanning and DDoS attacks. The work is carried out with feature selection approach using correlation coefficient scores to reduce processing overhead while achieving results with minimal performance hit.
- Providing the comparative analysis of train/test accuracies and other training statistics on specified computing resource. The analysis also covers the performance observations with cross validation, area under curve evaluation, and dimensionality reduction.
- Discussing the results in the light of recent attack vectors and proposing the future line of action.

The rest of this paper is organized as follows: Section 2 provides the related work in this research area. Section 3 provides a briefing on machine learning algorithms applied in this paper. Section 4 explains the experimental setup, and Section 5 provides classification results. Section 6 discusses the machine learning analysis, and Section 7 provides experimental observations with cross validation, area under curve evaluation, and dimensionality reduction. Finally, Section 8 provides the conclusion and future work followed by the references.

2. RELATED WORK

Brahmi *et al.* [9] worked on DARPA 98 dataset [10] with four types of attacks i.e. Scan (or Probe), Denial of Service (DoS), User to Remote (U2R) and Remote to Local (R2L). Attack detection rates were obtained with multidimensional association rule mining, where six-dimensional rule mining gave the best rates. They obtained 95% and 99% detection accuracy for Scan and DoS attacks respectively. Jemili *et al.* [11] worked on KDD 99 dataset [12] to first categorize normal and attack traffic with junction tree inference module. In subsequent steps for attack category, Scan and DoS attacks were detected using anomaly detection module with accuracy rates of 99% and 89% respectively. The other types of attacks showed less accuracy due to lower number of training samples. Zhang *et al.* [13] grouped major attacks in KDD 99 dataset (Scan and DoS) and further expanded them into four attack levels. The detection accuracies achieved for the levels were 95%, 93%, 90% and 87% with constant 1% false acceptance rate using random forest classification. In [14], the authors used similar technique of random forest classification on KDD 99 dataset. However, the dataset was normalized in this work during the preprocessing stage. A comparative study was also presented with Naïve Bayes, Decision Tree and Gaussian maximum likelihood classifiers. The detection accuracies with random forest classification obtained for Scan and DoS attacks were 76% and 97% respectively.

Gao *et al.* [15] determined five features to classify distributed reflection DoS attacks with the SVM algorithm. A few experimental runs also displayed 100% accuracy without any false positive. However, the limitation of their work exists as a limited set of experiments. In [16], several feature scores including correlation ranking were input to an ensemble method of feature selection. The features exhibiting high scores under various methods easily crossed the threshold of final feature selection. In this way, 16 most important features were determined from the CAIDA 07 dataset [17]. A comparative study was presented using algorithms such as Naïve Bayes (NB), Random Forest (RF) and Multi-Layer Perceptron (MLP). High detection accuracy of 98.3% was achieved with MLP and the boosted feature set. In

[18], a comparison was presented among different machine learning algorithms to detect SYN flood attack (a variant of denial of service) in a virtualized environment of cloud computing. An intersection process was used to identify certain important features based on statistical analysis of TCP/IP header from an extended feature space. With the limited set of 25 features, Naïve Bayes, J48, neural network and supervised K-means algorithms were compared. The highest accuracy of 99.995% was achieved with J48 algorithm (a Java based decision tree algorithm in WEKA tool).

In [19], detection accuracy of SVM machine learning algorithm was compared with the accuracy of SNORT, an open source intrusion detection tool. The SVM classification was applied in libsvm (SVM Library of Java). With the evaluation metrics of true positives and false positives, the SVM provided 99% detection accuracy of attacks as compared to SNORT having 89% of accurate detections. Lu *et al.* [20] compared RF, NB and SVM algorithms to detect establishment of C&C session before the launch of DDoS attack using the feature vector of network traffic with 55 dimensions. The traffic comprising of normal and C&C session traffic was generated with HTTP and IRC protocols running on ports 80 and 6667 respectively. The RF algorithm showed better results in terms of detection accuracy as compared to NB and SVM. In [21], the authors simulated modern types of DDoS attacks at application layer along with traditional network layer attacks. The comparative analysis of attack detection was provided using machine learning algorithms of NB, RF and MLP. The highest accuracy of 98.63% was achieved with MLP followed by 98.02% and 96.91% accuracies of RF and NB respectively. In [22], the authors simulated DDoS attacks and presented a comparative analysis of attack detection using machine learning algorithms of NB, RF, MLP, logistic regression and radial basis function. The highest accuracy of 93.67% was achieved using NB algorithm configured with multinomial classifier. However, the limitation of their work exists as having quite a limited set of data samples. Robinson and Thomas [23] applied and compared a range of machine learning algorithms to detect DDoS attacks on three different datasets i.e. CAIDA 07, CAIDA Conficker and Lincoln Laboratory Scenario – Distributed Denial of Service (LLSDDoS). The highest detection accuracy of 99.96% was achieved on CAIDA Conficker dataset using RF classifier with Ada Boost.

The overall analysis of related work reveals that the machine learning based comparative studies of port scanning and DDoS attack classifications are available in literature either on older benchmark datasets or simulated network traffic. Hence, the work of presenting comparative analysis of different machine learning algorithms using a newer benchmark dataset of intrusion detection with recent vectors of port scanning and DDoS attacks is a need of community to extend the research in this domain with available information of the best performing algorithms.

3.0 MACHINE LEARNING ALGORITHMS AND DATASETS

3.1 Tree

Different variants of decision tree algorithm work in a way that the subsets of dataset are created based on splitting the samples with respect to target classes and separated by the most contributing feature in the dataset. In the subsequent phases, each subset is independently split into further subsets based on high contributing vectors. In this way, decision trees are built, and the system learns how to split the new data points with respect to the feature set to reach the classification results [24].

3.2 Discriminant Analysis

Discriminant analysis is a feature extraction technique of machine learning. From 'n' independent variables of a dataset, 'p' new independent variables ($p \leq n$) are extracted which separate most of the classes of target variable. Unlike principal component analysis where variance within feature variable is considered, DA considers the classes of target variable hence it is a supervised method of feature extraction [25].

3.3 Support Vector Machine (SVM)

Support Vector Machine is quite a popular machine learning approach for predicting and classifying data in high dimensional space. SVM brings out the information of data for separating target classes in terms of introducing hyperplanes among the feature vectors in such a way that the distance between points nearest to the hyperplanes is maximized. These points

lying the closest to the hyperplanes are termed as support vectors. It is a complex technique of machine learning due to high dimensional computations [26].

3.4 K-Nearest Neighbors (KNN)

K-Nearest Neighbors algorithm relies on the information provided by the 'K' number of already classified or trained points closest to the new data point in feature space. The voting mechanism decides the fate of new data point on assigning a class to it. The closeness factor to choose 'K' points is determined by some applied metric. Euclidean distance (straight line distance between two points in n-dimensional space) is usually the most common metric applied in KNN [27].

3.5 Ensemble Classifiers

Ensemble classifiers apply independent algorithms under the hood to solve classification problem with the help of individual results provided by the underlying algorithms. For example, the boosted tree ensemble classifier applies a preconfigured number of decision trees in such a way that the result of a tree will be used to boost its more contributing features in the subsequent tree. Hence, a series of decision tree results are used to find the weighted average for final classification. In the case of bagged tree ensemble classifier, independent decision trees are run in parallel to provide results for ensemble technique. The simple average or voting is used for the final classification [28].

There are different datasets available for research in the domain of intrusion detection. In Table1, the important datasets are mentioned with relevant information [29-31].

4. EXPERIMENTAL SETUP

From CICIDS2017 dataset, total 512212 instances are taken from the Friday–Working hours–Afternoon scenarios of port scan and DDoS attacks. There are three classes in total labeled as 0, 1 and 2 for Normal, Port scanning and DDoS traffic respectively. 225255 instances are labeled normal, 158930 instances are port scanning traffic, and remaining 128027 instances belong to DDoS attacks. There are 78 independent variables (features) in default state with no missing values; however significant feature preprocessing is required as mentioned below:

Table 1: Datasets for research on intrusion detection

Dataset	Information	Year
KDD 99 Cup	41 features representing the legitimate and attack traffic. Attacks are categorized into four classes: Denial-of-service (DoS), Probing, Remote-to-Local (R2L), and User-to-Remote (U2R) attacks.	1999
CAIDA 07	Anonymized traces of one-hour DDoS attack traffic, mainly containing the flooding traffic of SYN, ICMP and HTTP protocols.	2007
CAIDA 08	Legitimate and attack traces monitored at datacenters of Chicago and San Jose, taken on March 19 and July 17 of year 2008 respectively.	2008
NSL-KDD	Refined version of KDD 99 dataset after removal of duplicate records. The number of records are also reduced while keeping the same feature set [30].	2009
ISCX	Traffic from real-world physical test environment with centralized botnets. This dataset has 19 features with 196,032 records.	2012
UNSW–NB15	49 features covering nine different types of modern attacks, with identification of new patterns of normal traffic [31].	2015
CICIDS2017	78 features with normal traffic and attacks including botnet, cross site scripting, DoS/DDoS and SSH brute force.	2017

- *12 features* are removed having no variation in the dataset (single value features) or showing values including incalculable figures such as 'Infinity'.
- *45 features* are removed having below 20% correlation coefficient with respect to the dependent (target) variable. According to [32], labeling systems exist that roughly consider correlation coefficients which are ≤ 0.35 being the representation of low or weak correlations. Hence, it is assumed that all decisive variables are included in the final set of features after feature selection. This configuration is made using *corrcoef* function of *Numpy* package for scientific computing in Python 3.
- *21 features* remain in the dataset as the most significant features according to the configured value of correlation coefficient. These 21 features are taken for the classification.

- *Data normalization* is done using ***StandardScaler*** class of *scikit-learn* library in Python 3. The dataset is split in 70-30% ratio for training and testing in randomized manner using *train_test_split* class of scikit-learn. It provides 358548 samples of data for training (157701 normal, 111292 port scanning, and 89555 DDoS) and 153664 samples for testing (67554 normal, 47638 port scanning, and 38472 DDoS).

Classification experiments are performed in Matlab R2017a due to the availability of enriched set of algorithms in *Apps* section under Classification Learner. The 21 independent variables shortlisted for the classification's feature space are mentioned in Table 2.

Table 2: Correlation based selected features for classification

Serial No.	Feature	Correlation Coefficient
1	Destination Port	-0.29
2	Total Length of Fwd Packets	-0.20
3	Fwd Packet Length Max	-0.21
4	Fwd Packet Length Mean	-0.23
5	Bwd Packet Length Max	0.55
6	Bwd Packet Length Min	-0.38
7	Bwd Packet Length Mean	0.56
8	Bwd Packet Length Std	0.56
9	Bwd IAT Total	-0.25
10	Bwd IAT Max	-0.21
11	Min Packet Length	-0.43
12	Max Packet Length	0.44
13	Packet Length Mean	0.45
14	Packet Length Std	0.47
15	Packet Length Variance	0.45
16	PSH Flag Count	0.28
17	URG Flag Count	-0.31
18	Average Packet Size	0.45
19	Avg Fwd Segment Size	-0.23
20	Avg Bwd Segment Size	0.56
21	Subflow Fwd Bytes	-0.20

5. CLASSIFICATION RESULTS

The classification results of different algorithms with accuracy scores and other parameters of training including the confusion matrices are provided in Table 3. The numbers under "Predicted Class" columns in Table 3 show correct and incorrect classifications of respective target categories under specified machine learning algorithms. While reasonable training accuracies, true positive rates, and false negative rates are observed for all experiments, some algorithms still show large differences in training and testing accuracy scores. It shows that not all algorithms fit well in classifications of port scanning and DDoS attacks for the given dataset taking into consideration the scope and experimental settings of this research. The experiments are conducted on Intel® Core™ i7, 7500U CPU @2.70 GHz with 4 cores. It is a DELL Inc. Laptop machine with 8GB of primary storage (RAM).

6. DISCUSSION AND ANALYSIS

The classification results obtained in Table 3 reveal that some machine learning algorithms can exhibit substandard performance in classifying port scanning and DDoS attacks even after they show good training accuracies. As testing instances are different from the training set, the considerable differences of feature vectors of the two sets can make it harder for even a trained model to show better results in terms of classification accuracy. In this analysis, the best performing algorithms can be found in terms of classification accuracy of port scanning and DDoS attacks which show good training as well as classification scores. Figure 3 shows the results of all specified machine learning algorithms.

From Fig. 3 as well as Table 3, it is observed that the Fine Gaussian variant of SVM is the best performing machine learning model among the experiments which shows 99% testing (classification) accuracy as well as 99% training accuracy. For collective analysis, it is observed that all the variants of discriminant analysis and SVM provide good classification results of port scanning and DDoS attacks. On the other hand, inefficient performance in the range of 49-69% is exhibited by the tree based models as well as KNN and most of the ensemble classifier based algorithms. However, the subspace discriminant variant of ensemble classifier provides 85.5% testing accuracy which is still competitive to other high performing algorithms.

Based on the analysis, the specified machine learning algorithms are rated in Table 4 into three categories on a subjective scale i.e. Good (≥ 85%), Fair (65% - 84.9%) and Substandard (≤ 64.9%). For the analysis of average True Positive Rate (TPR) and False Negative Rate (FNR) in the training phase as shown in

Table 3: Classification Results

	Confusion Matrix			Predicted Class			True +ve Rate	False -ve Rate	Training Accuracy (%)	Testing Accuracy (%)	Prediction Speed K obs/sec)	Training Time (sec)
Tree	Simple Tree	Actual Class	0	157518	889	294	99	1	99.4	60.6	3700000	4.4884
			1	249	110453	0	99	1				
			2	183	0	89372	99	1				
	Medium Tree	Actual Class	0	157595	90	16	99	1	99.9	53.8	3200000	5.4814
			1	249	111043	0	99	1				
			2	145	0	89410	99	1				
	Complex Tree	Actual Class	0	157626	53	22	99	1	99.9	56.6	2600000	13.507
			1	125	111167	0	99	1				
			2	47	0	89508	99	1				
Discriminant Analysis	Linear Dist.	Actual Class	0	131486	12681	13534	83	17	92.6	92.8	790000	6.039
			1	5	111179	108	99	1				
			2	25	38	89492	99	1				
	Quadratic Dist.	Actual Class	0	147877	853	8971	94	6	97.1	97.1	680000	5.0499
			1	588	110640	64	99	1				
			2	42	0	89513	99	1				
Support Vector Machine	Linear SVM	Actual Class	0	144999	5634	7068	92	8	96.4	96.2	540000	529.63
			1	35	111238	19	99	1				
			2	42	8	89505	99	1				
	Quadratic SVM	Actual Class	0	154069	728	2904	98	2	98.8	96.7	7500	334.95
			1	474	110818	0	99	1				
			2	85	0	89470	99	1				
	Cubic SVM	Actual Class	0	144949	5658	7094	99	1	96.3	92.0	7200	1797.523
			1	52	111211	29	99	8				
			2	59	17	89479	99	1				
	Fine Gaussian	Actual Class	0	154708	96	2897	98	2	99.0	99.0	3200	587.01
			1	474	110818	0	99	1				
			2	86	0	89469	99	1				

	Medium Gaussian	Actual Class	0	150528	746	6427	95	1	97.7	97.8	2500	593.47
			1	821	110470	1	99	1				
			2	87	0	89468	99	1				
	Coarse Gaussian	Actual Class	0	150405	771	6525	95	5	97.6	93.0	7100	1725.879
			1	888	110393	11	99	1				
			2	126	1	89428	99	1				
KNN	Fine KNN	Actual Class	0	144121	5984	7596	91	9	96.1	65.9	2200	8636.765
			1	102	111108	82	99	1				
			2	102	34	89411	99	1				
	Medium KNN	Actual Class	0	143501	6404	7796	91	9	95.6	65.7	2400	8138.642
			1	644	110223	425	99	1				
			2	209	146	89200	99	1				
	Coarse KNN	Actual Class	0	142201	7095	8405	90	19	95.1	62.6	2700	7257.832
			1	822	110151	319	99	1				
			2	455	204	88896	99	1				
	Cosine KNN	Actual Class	0	142310	6993	8398	90	10	95.5	60.9	2600	7768.586
			1	213	111012	67	99	1				
			2	302	152	89101	99	1				
	Cubic KNN	Actual Class	0	146110	5998	5593	92	8	95.7	68.7	2800	8247.752
			1	2213	108570	509	97	3				
			2	802	219	88534	99	1				
	Weighted KNN	Actual Class	0	145072	6819	5810	92	8	95.3	66.0	2700	8061.481
			1	3010	107710	572	97	3				
			2	502	132	88921	99	1				
Ensemble Classifier	Boosted Trees	Actual Class	0	157610	83	8	99	1	99.9	54.0	79000	87.703
			1	186	111106	0	99	1				
			2	168	0	89387	99	1				
	Bagged Trees	Actual Class	0	157602	77	22	99	1	99.9	68.4	72000	47.226
			1	234	111058	0	99	1				
			2	47	0	89508	99	1				

	Subspace Discriminant	Actual Class	0	138536	17769	1396	88	12	85.5	85.5	28000	49.636
			1	49	111179	64	99	1				
			2	32644	11	56900	64	36				
	Subspace KNN	Actual Class	0	157595	70	36	99	1	99.8	60.7	7400	1780.41
			1	245	111047	9	99	1				
			2	53	0	89502	99	1				
	RUSBoosted Trees	Actual Class	0	157595	90	16	99	1	99.9	49.2	82000	464.17
			1	250	111042	0	99	1				
			2	145	0	89410	99	1				

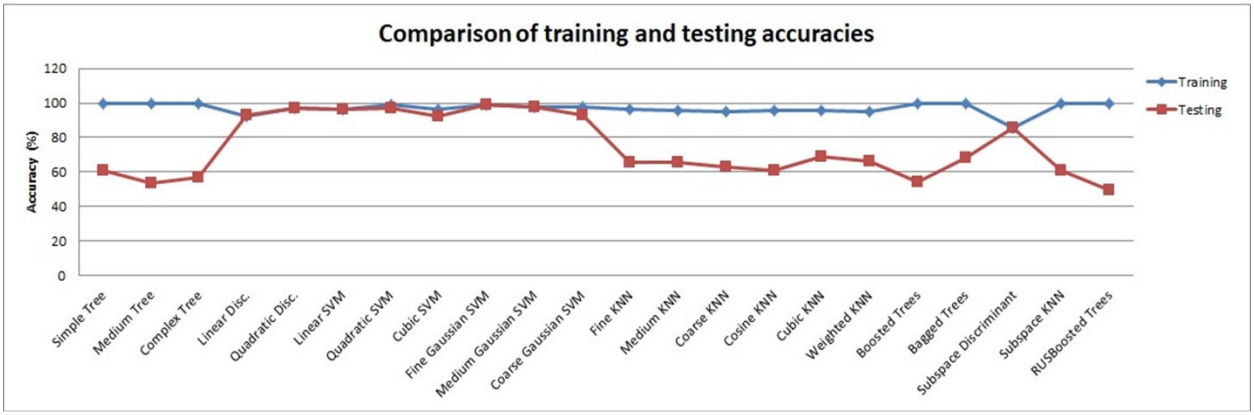

Fig. 3: Training and testing accuracies of machine learning algorithms

Fig. 4, it is observed that the majority of algorithms are capable of differentiating attacks from normal traffic with significant percentage. Although the subspace discriminant variant of ensemble classifier has the highest FNR of 16.33%, it is interesting to observe that it could still show good classification accuracy as mentioned in Table 4. On the other hand, several other classifiers such as tree based and KNN showed performance degradation in testing phase when new/unseen data was presented for classification. In general, it is observed that all the algorithms can identify the attack traffic well in training, and the misclassifications mainly belong to the normal traffic due to its higher share in the mixed traffic with a factor of noise in the data.

For the analysis of training time exhibited by the specified machine learning algorithms, it is observed that it generally increases when observations per second by the classification algorithm decreases in the training phase. As high differences of K-observations/sec and training time among various machine learning algorithms are observed in Table 3, they are shown in scatter plot in Fig. 5 with normalized values between 0 and 1. It can be seen that the less number of training observations per second requires high amount of training time to complete the learning phase of a model in general. However, some models are also comparatively efficient as they take less time to complete with small numbers of training observations per second (e.g. a few variants of ensemble classifier and SVM from Table 3). Fast

Table 4: Rating of machine learning algorithms based on classification accuracies

Algorithm Type	Variant of Model	Classification Accuracy (%)	Rating
Tree	Simple	60.6	Substandard
	Medium	53.8	Substandard
	Complex	56.6	Substandard
Discriminant Analysis	Linear	92.8	**Good**
	Quadratic	97.1	**Good**
Support Vector Machine	Linear	96.2	**Good**
	Quadratic	96.7	**Good**
	Cubic	92.0	**Good**
	Fine Gaussian	99.0	**Good**
	Medium Gaussian	97.8	**Good**
	Coarse Gaussian	93.0	**Good**
KNN	Fine	65.9	Fair
	Medium	65.7	Fair
	Coarse	62.6	Substandard
	Cosine	60.9	Substandard
	Cubic	68.7	Fair
	Weighted	66.0	Fair
Ensemble Classifier	Boosted Trees	54.0	Substandard
	Bagged Trees	68.4	Fair
	Subspace Discriminant	85.5	**Good**
	Subspace KNN	60.7	Substandard
	RUSBoosted Trees	49.2	Substandard

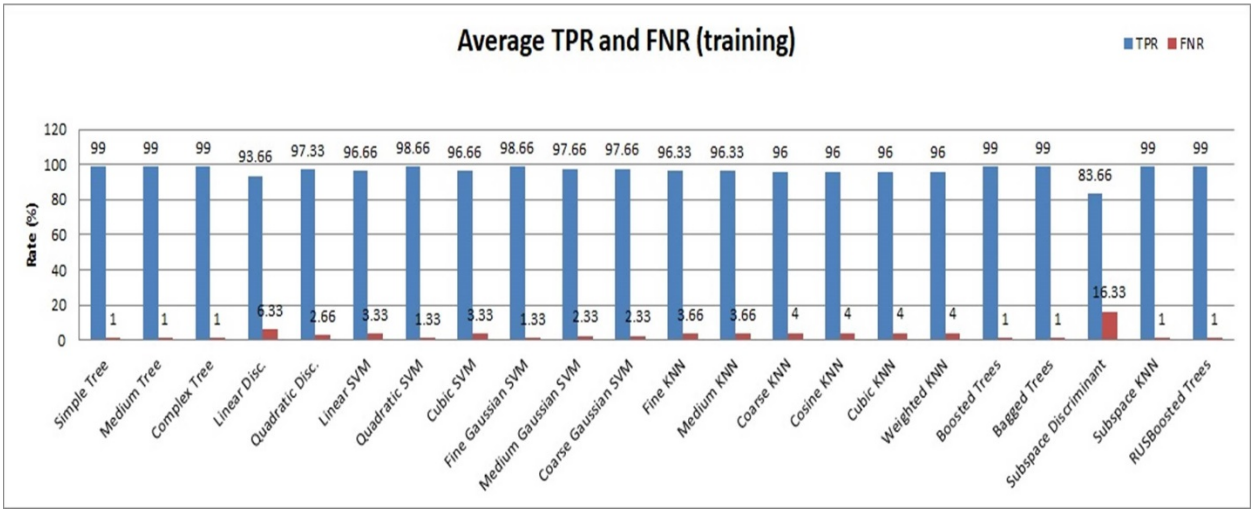

Fig. 4: Average TPR and FNR of machine learning algorithms during training

training completions are provided by the tree and discriminant analysis based algorithms. Hence in terms of fast training and high training/testing accuracy scores, discriminant analysis based machine learning models in this comparative study are found to be more accurate and efficient to classify port scanning and DDoS attacks.

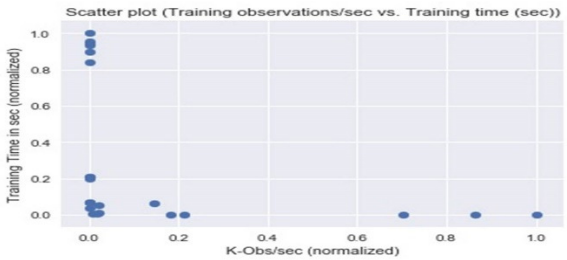

Fig. 5: Observations per Second vs. Training Time

7. COMPARATIVE ANALYSIS WITH VALIDATION, EVALUATION AND DIMENSIONALITY REDUCTION

The machine learning algorithms should not be trusted without validating the results to avoid overfitting and false sense of prediction strength. For this purpose, the steps of validation and evaluation are added in this paper to analyze whether the training part should be trusted to avoid overfitting, and evaluated through acceptable means. Fig. 6 explains the proposed scheme where k-fold cross validation and AUC analysis of ROC curves are included in the experiments. There is also a factor of dimensionality reduction, for which the Principal Component Analysis (PCA)) is used in this paper.

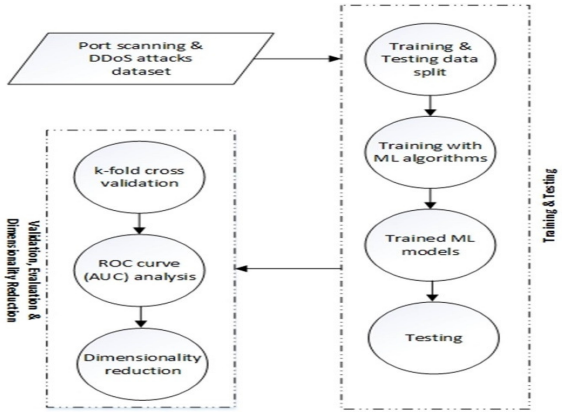

Fig. 6: Proposed scheme of analysis with validation, evaluation and dimensionality reduction.

7.1 K-Fold Cross Validation

In order to avoid overfitting during the training phase, k-fold cross validation is an effective tool by shifting the train-test splits for certain number of rounds to know whether a particular split is not an overfitting state. It can be established if other splits also produce the training accuracy close to the original one. In Fig. 7, a comparison is provided between no validation and 10-fold cross validation (k=10) of training accuracies of the machine learning algorithms. It can be noticed that for some comparisons, the average training accuracy of cross validation is slightly dropped from the one without validation as different splits can produce different accuracies (*e.g*. medium tree, linear discriminant, and coarse KNN). Hence, the average accuracy can be different with the validation step. In a few cases, it is also increased as compared to training accuracy with no validation (*e.g*. weighted KNN).

7.2 AUC Analysis of Roc Curves

Area-under-curve analysis of ROC is another effective tool of evaluation in order to avoid the *accuracy paradox* [33]. This term refers to the fact that a machine learning algorithm can provide the accuracy score which can be valid for only an instantaneous

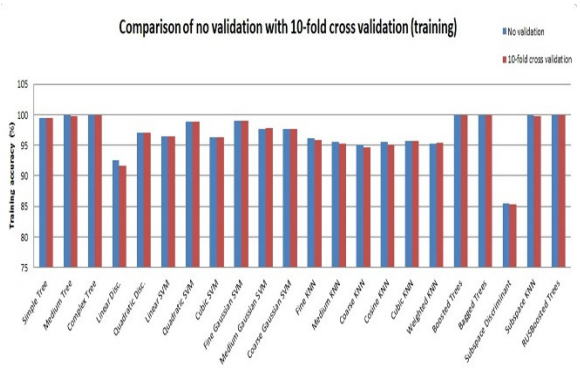

Fig. 7: Comparison of training accuracies (no validation vs. 10-fold cross validation)

operating point. Therefore, in order to avoid this state of false accuracy performance, the training ROC curve can be plotted between true positive and false positive scores to evaluate the area under curve which is the true representation of a model's accuracy; hence the accuracy paradox may be avoided. It can be established if the area under curve scores tally the training accuracy provided by the respective machine learning models. In Figs. 8, 9, and 10; ROC curves of selected algorithms are plotted. This selection is made in accordance with the *Good* rating algorithms of Table 4. In Fig. 8, both variants of discriminant analysis are plotted. In Fig. 9, three variants of SVM providing the respective highest training accuracies from Fig. 3 are plotted. In Fig. 10, the single good variant of ensemble classifier is evaluated. It can be noticed that all the mentioned algorithms show area under curve scores which tally the respective training accuracies of machine learning models, hence the accuracy paradox can be avoided. Here, the 10-fold cross validation is kept enabled for effective validation followed by the evaluation step. The curves are made with one vs. all approach i.e. normal traffic vs. attack traffic (covering both port scanning and DDoS types of attacks).

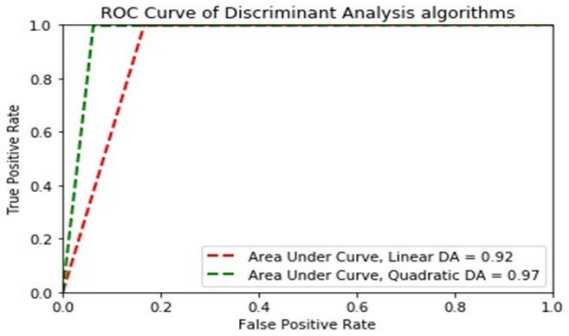

Fig. 8: AUC Analysis of Discriminant Analysis Algorithms.

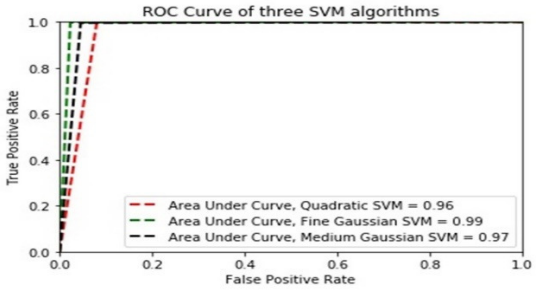

Fig. 9: AUC Analysis of three SVM Algorithms

8. DIMENSIONALITY REDUCTION WITH PCA

Principal Component Analysis (PCA) is an unsupervised tool of dimensionality reduction. It is unsupervised because it does not take into account the target classes to reduce dimensionality. It considers the variance of original features in a dataset and produces new features to preserve most of the variance. Hence, it is a feature extraction technique instead of a feature selection approach. The calculations are made by obtaining eigenvalues and eigenvectors from the covariance matrices [34]. In this analysis, training accuracies are obtained for all machine learning models with two PCA configurations, along with observing the prediction speeds. In Table 5, two different PCA settings are used i.e. PCA explaining 85% variance, and PCA explaining 90% variance. The reason behind using two different settings is the comparative analysis for different number of extracted features. There are five features extracted for PCA explaining 85% variance, and six features for PCA explaining 90% variance. It can be noticed that prediction speeds are reduced in most cases as compared to full-feature analysis for the reason that although dimensionality is reduced but 10-fold cross validation is kept enabled for effective validation along with the dimensionality reduction. Also, the prediction speed is generally lower in 90% analysis than 85% analysis due to the presence of an extra extracted feature in the latter case.

Table 5: Training accuracies and prediction speeds with PCA

Algorithm Type	Variant of Model	Training accuracy (%) with PCA explaining 85% variance	Prediction speed (K-obs/sec) with PCA explaining 85% variance	Training accuracy (%) with PCA explaining 90% variance	Prediction speed (K-obs/sec) with PCA explaining 90% variance
Tree	Simple	90.3	3300000	96.4	3300000
	Medium	99.8	3100000	99.8	3000000
	Complex	99.9	2500000	99.9	2400000
Discriminant Analysis	Linear	83.1	740000	82.5	720000
	Quadratic	91.7	650000	91.3	630000
Support Vector Machine	Linear	90.3	530000	94.8	520000
	Quadratic	92.3	7500	96.2	7000
	Cubic	91.4	7100	94.8	7000
	Fine Gaussian	98.0	3200	99.0	3100
	Medium Gaussian	94.8	2500	96.2	2500
	Coarse Gaussian	95.0	7100	97.2	7100
KNN	Fine	92.6	2300	95.2	2200
	Medium	92.3	2400	93.2	2300
	Coarse	92.0	2600	93.0	2400
	Cosine	92.7	2700	95.3	2600
	Cubic	93.0	3000	93.4	2800
	Weighted	92.6	2900	95.2	2800
Ensemble Classifier	Boosted Trees	99.9	80000	99.9	78000
	Bagged Trees	99.8	77000	99.9	75000
	Subspace Disc.	81.7	30000	83.4	28000
	Subspace KNN	99.8	9000	99.9	8000
	RUSBoosted Trees	99.9	89000	99.9	87000

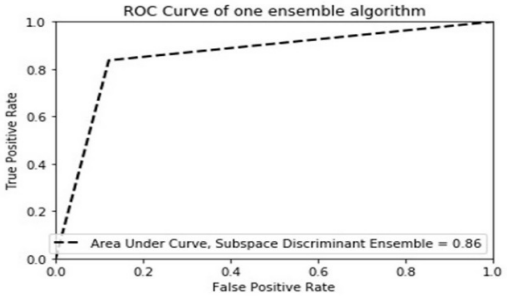

Fig. 10: AUC Analysis of one ensemble algorithm

9. CONCLUSION AND FUTURE WORK

In this paper, a comparative analysis is presented on a recently published benchmark dataset CICIDS2017 to classify port scanning and DDoS attacks in a mix of normal and attack traffic. 22 different machine learning algorithms are trained and tested to check their performance on the recent vectors of attacks. The classification results show that all the variants of discriminant analysis and SVM provide testing accuracies of more than 90%. The best accuracy score of 99% is obtained with the Fine Gaussian variant of SVM. In general, the training time increases with a decrease in number of observations per second during the training phase. The fastest convergence of training time is exhibited by tree and discriminant analysis based algorithms. Hence in terms of fast training and high training/testing accuracies, discriminant analysis based models are more productive. In the subjective rating of algorithms, 9 algorithms receive the highest rating i.e. good for showing more than 85% testing accuracy. This comparative analysis is further extended to observe training performance of machine learning models through k-fold cross validation, AUC analysis of ROC curves, and dimensionality reduction using PCA. The 10-fold cross validation shows that the average training accuracy in some cases is slightly dropped from the one without validation as different splits can produce different accuracies to bring a slight change in the average accuracy with cross validation. The AUC analysis of ROC curves shows that all the observed algorithms provide area under curve scores which tally the respective training accuracies of machine learning models. Finally, the dimensionality reduction with PCA explaining 85% and 90% variances, providing 5 and 6 extracted features respectively, shows that the prediction speeds as compared to full-feature analysis can vary with respect to the dimensionality reduction as well as enabling the 10-fold cross validation for effective results to avoid overfitting.

Machine learning is recently being explored in research for effective and efficient applications in the field of information security [35, 36]. In fact, security is always one of the top concerns in the development of automated communication systems [37]. Intrusion detection is one of the major domains under cyber security, and machine learning is being actively applied and tested in this area to get fruitful results [38]. In future work, more variants of machine learning models including neural networks (multi-layer perceptron) will be considered in conjunction with detailed feature engineering to find enriched comparisons of machine learning algorithms on recent datasets of port scanning and DDoS attacks. In addition to this, more analysis on the techniques of dimensionality reduction will be performed to decrease the performance overhead in significant manner.

ACKNOWLEDGMENT

The authors are thankful to anonymous reviewers for suggesting improvements in the quality of this paper. We are also thankful to Mehran University of Engineering and Technology (MUET) and the organizing committee of INCCST'19, where a part of this research was initially presented.

REFERENCES

[1] "Oracle | Integrated Cloud Applications and Platform Services", https://www.oracle.com/index.html, [Last Visited on 10th June 2019].

[2] "NMAP: The Network Mapper – Free Security Scanner", https://nmap.org/, [Last Visited on 5th June 2019.

[3] Nagesh K., Sumathy R., Devakumar P., Sathiyamurthy K., "A Survey on Denial of Service Attacks and Prelusions", *Proceedings of the International Conference on Information and*

Analytics, No. 118, 2016.

[4] "Content Delivery Network (CDN) & Cloud Computing Services | Akamai." https://content.akamai.com/us-en-PG11224-summer-2018-soti-web-attack-report.html, [Last Visited on 15th May 2019].

[5] Aamir M., Zaidi M.A., "A survey on DDoS attack and defense strategies: from traditional schemes to current techniques", *Interdisciplinary Information Sciences*, Vol. 19, No. 2, pp. 173–200, 2013.

[6] Aamir M., Zaidi S.M.A., "Denial-of-service in content centric (named data) networking: a tutorial and state-of-the-art survey", *Security and Communication Networks*, Vol. 8, No. 11, pp. 2037–2059, 2015.

[7] Xue B., Zhang M., Browne W.N., Yao X., "A survey on evolutionary computation approaches to feature selection", *IEEE Transactions on Evolutionary Computation*, Vol. 20, No. 4, pp. 606–626, 2016.

[8] Sharafaldin I., Lashkari A.H., Ghorbani A.A., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization.," in ICISSP, pp. 108–116, 2018.

[9] Brahmi H., Brahmi I., Yahia S.B., "OMC-IDS: at the cross-roads of OLAP mining and intrusion detection", *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 13–24, 2012.

[10] "1998 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory", https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, [Last Visited on 3rd June 2019].

[11] Jemili F., Zaghdoud M., Ahmed M.B., "A framework for an adaptive intrusion detection system using Bayesian network", *Intelligence and Security Informatics*, pp. 66–70, 2007.

[12] "KDD Cup Dataset 1999." http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, [Last Visited on 2nd June 2019].

[13] Zhang J., Zulkernine M., Haque A., "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 38, No. 5, pp. 649–659, 2008.

[14] Gharibian F., and Ghorbani A.A., "Comparative study of supervised machine learning techniques for intrusion detection", *Proceedings of the 5th Annual Conference on Communication Networks and Services Research*, pp. 350–358, 2007.

[15] Gao, Y., Feng, Y., Kawamoto, J., and Sakurai, K., "A machine learning based approach for detecting DRDoS attacks and its performance evaluation", *Proceedings of the 11th Asia Joint Conference on Information Security*, pp. 80–86, 2016.

[16] Singh K.J., De T., "Efficient Classification of DDoS Attacks Using an Ensemble Feature Selection Algorithm", *Journal of Intelligent Systems*, Vol. 2017, pp. 1–13, 2017.

[17] "CAIDA DDoS Attack 2007 Dataset." http://www.caida.org/data/passive/ddos-20070804_dataset.xml, [Last Visited on 13th June 2019].

[18] Al-Hawawreh M.S., "SYN flood attack detection in cloud environment based on TCP/IP header statistical features", *Proceedings of the 8th International Conference on Information Technology*, pp. 236–243, 2017.

[19] Li M.S.H., Vélez J.I., Castillo L., "Distributed Denial of Service (DDoS) Attacks Detection Using Machine Learning Prototype", *Proceedings of the 13th International Conference on Distributed Computing and Artificial Intelligence*, pp. 33–41, 2016.

[20] Lu L., Feng Y., Sakurai K., "C&C session detection using random forest", *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, No. 34, 2017.

[21] Alkasassbeh M., Al-Naymat G., Hassanat A.B., Almseidin M., "Detecting distributed denial of service attacks using data mining techniques", *International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 1, 2016.

[22] Singh K.J., De, T., "An approach of DDOS attack detection using classifiers", *Emerging Research in Computing, Information, Communication and Applications*, Springer, pp. 429–437, 2015.

[23] Robinson R.R., Thomas C., "Ranking of machine learning algorithms based on the performance in classifying DDoS attacks", *Recent Advances in Intelligent Computational Systems,* pp. 185–190, 2015.

[24] Patel B.R., Rana K.K., "A survey on decision tree algorithm for classification", *International Journal of Engineering Development and Research*, Vol. 2, No. 1, pp. 1–5, 2014.

[25] Kan M., Shan S., Zhang H., Lao S., Chen X., "Multi-view discriminant analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 1, pp. 188–194, 2016.

[26] Suthaharan S., "Support vector machine," Machine learning models and algorithms for big data classification, Springer, pp. 207–235, 2016.

[27] Larose D.T., Larose C.D., "*K-nearest neighbor algorithm," Discovering Knowledge in Data: An Introduction to Data Mining*, Second Edition, John Wiley & Sons, pp. 149–164, 2005.

[28] Folino G., Sabatino P., "Ensemble based collaborative and distributed intrusion detection systems: A survey", *Journal of Network and Computer Applications*, Vol. 66, pp. 1–16, 2016.

[29] Ring M., Wunderlich S., Scheuring D., Landes D., Hotho A., "A survey of network-based intrusion detection data sets", *Computers and Security*, Vol. 86, pp. 147–167, 2019.

[30] Tavallaee M., Bagheri E., Lu W., Ghorbani A.A., "A detailed analysis of the KDD CUP 99 data set", *Proceedings of the Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp. 1–6, 2009.

[31] Moustafa N., Slay J., "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *Proceedings of the Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015.

[32] Taylor R., "Interpretation of the correlation coefficient: A basic review," *Journal of Diagnostic Medical Sonography*, Vol. 6, No. 1, pp. 35–39, 1990.

[33] Aamir M., Zaidi S.M.A., "DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation", Vol. 18, No. 6, pp. 761–785, 2019.

[34] Aamir M., Zaidi S.M.A., "Clustering based semi-supervised machine learning for DDoS attack classification", *Journal of King Saud University – Computer and Information Sciences*, pp. 1–11, 2019.

[35] Stamp M., "A survey of machine learning algorithms and their application in information security", *Guide to Vulnerability Analysis for Computer Networks and Systems*, Springer, pp. 33–55, 2018.

[36] Berman D.S., Buczak A.L., Chavis J.S., Corbett C.L., "A survey of deep learning methods for cyber security", *Information*, Vol. 10, No. 4, pp. 1–35, 2019.

[37] Memon Z., Jalbani A.H., Shaikh M., Memon R.N., Ali A., "Multi-agent communication system with chatbots", *Mehran University Research Journal of Engineering and Technology*, Vol. 37, No. 3, pp. 663–672, 2018.

[38] Shah S.A.R., Issac B., "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Generation Computer Systems*, Vol. 80, pp. 157–170, 2018.