ISSN 0146-4116, Automatic Control and Computer Sciences, 2019, Vol. 53, No. 5, pp. 419–428. © Allerton Press, Inc., 2019.

Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset

Naveen Bindra[a], * and Manu Sood[a], **

[a] *Department of Computer Science (HPU), Shimla, India*

**e-mail: naveenjb@hotmail.com*

***e-mail: soodm_67@yahoo.com*

Received November 12, 2018; revised March 22, 2019; accepted March 25, 2019

Abstract—Recent trends have revealed that DDoS attacks contribute to the majority of overall network attacks. Networks face challenges in distinguishing between legitimate and malicious flows. The testing and implementation of DDoS strategies are not easy to deploy due to many factors like complexities, rigidity, cost, and vendor specific architecture of current networking equipment and protocols. Work is being done to detect DDoS attacks by application of Machine Learning (ML) models but to find out the best ML model among the given choices, is still an open question. This work is motivated by two research questions: 1) which supervised learning algorithm will give the best outcomes to detect DDoS attacks. 2) What would be the accuracy of training these algorithms on a real-life dataset? We achieved more than 96% accuracy in the case of Random Forest Classifier and validated our results using two metrics. The outcome was also compared with the other works to confirm its adequacy. We also present a detailed analysis to support our findings.

Keywords: DDoS detection, DDoS attack, Machine Learning, security, network threats, Scikit-learn, classification

DOI: 10.3103/S0146411619050043

1. INTRODUCTION

Computer Networks in today's era have become a backbone of many establishments including government, research, and defense organizations. Botnet-based DDoS attack infects computers with malware to act as a Botnet and command and control rest with the attacker [1]. A DDoS attack not only chokes the compute resource but also the available bandwidth, thereby bringing the network to halt. The vigor of DDoS attacks has come a long way as now the attackers even ask for the ransom. The attack on seven South Korean banks to extort money in lieu of leaving their work back to normalcy is the latest example of the advancement in cybercrimes [2]. Similarly, the attack by 'wannacry-ransomware' and the Equifax and Deloitte's breach, are few instances among the long list of attacks [3], which caught these organization off their guard and remained the focus of discussions among the security professionals and researchers for quite some time now. The opening/extending of networks to accommodate Datacenters, clouds, IoT etc. has further raised the challenge to keep networks secure.

Such incidents have forced various establishments to adopt innovative strategies to deal with such situations and losses. Various reports [4] suggest that DDoS attacks are increasing exponentially and thus caught researchers' attention to devise effective and optimum strategies in dealing with such attacks. The DDoS attacks are not new but their sophistication and scale have definitely reached an alarming stage. Traditional networks are not good enough to handle these attacks due to their ossified architecture. The challenge of detecting DDoS with the traditional strategies i.e. signature-based Intrusion prevention system at the periphery of network and statistical methods for the analysis of whole network flows, fall short of addressing these attacks due to the ever-changing anatomy of DDoS attacks. Recent trends have revealed that DDoS attacks contribute to a majority of network attacks [5]. The testing and implementation of DDoS strategies are not easy to deploy due to complexities, rigidity, cost, and vendor specific architecture of current networking equipment and protocols. Networks face challenges in distinguishing between legitimate and malicious flows. Application of Machine Learning algorithms for DDoS detection, though have just taken off but so far has been promising [7–10, 13]. Once trained on extensive data, they perform fantastically. The performance of Machine Learning models vastly depends on the selection

of features, quality and quantity of training data. Most of the studies, to the best of our knowledge, have used datasets like KDD 99, NSL-KDD, DARPA and CAIDA etc., which excludes recent DDoS attacks. This makes our work more relevant as we investigate the application of supervised learning algorithms on a real-life dataset, having contemporary DDoS attacks.

The major challenges remain in dealing with a DDoS attack is to differentiate between legitimate and malicious traffic flow. With the increase in sophistication of DDoS attacks, the packets spoofing has increased rendering the existing firewalls ineffective in identifying the malicious traffic. The other bigger problem is that they are capable of detecting only the known attacks due to their signature base identification mechanism. Machine Learning models offer a new glimmer of hope as it can address the gaps in their capabilities of IDS by detecting even "new" and "unseen" DDoS attacks. This decade until now, has seen a surge in explorations of various Machine Learning based models by the researchers to classifying legitimate network traffic from malicious one, but still, it falls short of tackling this issue completely. To effectively deal with these attacks, one must train the Machine Learning models with contemporary datasets and employ efficient feature selection/creation strategies. An efficient ML based classifier would be the one with low rates of false positives/negatives.

Achieving accuracy and performance of Machine Learning are still evolving technologies. This study sets out to investigate the applicability of Machine Learning in network traffic classification by using the latest exhaustive dataset to train our Machine Learning-based classifiers. The results achieved are encouraging and ushered in our journey to improve upon the performance and accuracy with the use of efficient preprocessing of datasets. Our work brings forth the importance of having an optimal DDoS detection scheme to secure networks.

Our work is more relevant in the sense that it investigates the application of supervised learning algorithms on a dataset, which includes recent DDoS attacks imitating the real scenario. We validated our results using two metrics. We also present a detailed analysis to support our findings. This article is organized into six sections. Section 2 of this paper gives an overview of existing flavors of DDoS attacks and their operating mechanisms. This section also gives a brief description of the five supervised Machine Learning algorithms actually applied in this work. Section 3 has the details about the types of datasets used in literature for network traffic classification. It also provides details about the experimental testbed used for this study. Section 4 outlines the work done by researchers in the literature to explore network flow classification. Section 5 is the most vital one as it divulges details and the manner in which these results obtained and the conclusion is drawn in the final section 6 scope for future work is likewise indicated in this section.

2. RELATED WORK

Previous studies have shown that Machine Learning applies for detection of DDoS attacks but the choices of datasets so far are not that inclusive. Authors in [6] studied the performances of various Machine Learning algorithms and ranked them so. This work sketches the outcome of the accuracy of ensemble-based classifiers. These classifiers were used to build models to categorize network traffic. The datasets used were namely DARPA, CAIDA DoS attacks 2007. However, this work falls short of illustrating the detection of contemporary DDoS attacks. The work of [7] assesses the application of Machine Learning for detection of BOTNET. They used publically available Zeus Botnet toolkit and substantiate the detection of newer versions after training their model on older versions. They also perceived similarity in command and control communication used by various versions of Zeus Botnet toolkit. An SVM based scheme for predicting the number of zombies in a DDoS attack is well illustrated by [8]. This work used SVM to detect the zombies in a DDoS attack. [9] Used intelligent decision prototype for detecting and tracing DDOS attacks. Their strategy focused on packet marking and deterministic packet marking traceback schemes for identifying DDoS attacks. In [10], the performance of Machine Learning algorithms for detecting network intrusion is evaluated. The author compared performances of decision tree algorithm with neural network and support vector machine. The metrics used for comparison were accuracy, detection rates, false alarm rate, and the accuracy of the four categories of attacks. The work of [11] evaluates supervised learning algorithms i.e. CART decision tree and naïve bays classifiers vis-à-vis NIDS (Bro and Corsano). In reference [12], the authors presented a solution for detecting P2P Botnets using network behavior analysis and Machine Learning. Botnets are used to carry out DDoS attacks. This work assessed the application of Machine Learning to detect Botnets. They studied the command and control phase for detection of DDoS attacks before they are actually launched. Machine Learning algorithms used were SVM, artificial neural network, k nearest neighbor classifier, Gaussian-based classifier, and naïve based classifiers. The limitation of this work is that it considered the detection of a single compromised host and could not detect a whole of Botnet.

From the review of the literature, it is concluded that most of the research has been focused on the datasets which do not entail the latest DDoS attacks which otherwise is necessary for the training of Machine Learning models to detect DDoS attacks in a real life scenario and match the sophistication of DDoS attacks. Therefore, the performance of supervised ML algorithms over the latest real-life dataset having contemporary DDoS attacks is still awaited.

3. DDoS DETECTION TECHNIQUES AND SUPERVISED MACHINE LEARNING

A. DDoS Attacks

DDoS attacks were a natural choice for our exploration as it contributes towards the major share in overall network attacks. DDoS attacks are DoS attacks, which are carried out by using a number hosts used as Bots (generally without their knowledge owner of the hosts) on a single or multiple networks. DDoS attacks can render a network ineffective and cause heavy monitory and reputational losses to organizations. The most difficult aspect of DDoS attacks is the rise in their sophistication and now they can even mimic the legitimate flows making their detection even more difficult for the current detection techniques.

DDoS attacks [13] are generally classified into three categories namely application-based DDoS attacks, volume-based DDoS attacks, and protocol based DDoS attacks. Application-based DDoS attacks target the vulnerabilities in the software e.g. windows, open BSD and Apache. Ping of death and Synflood are categories as Protocol based dos attacks. Volume-based DDoS attacks comprise TCP, UDP flooding or ICMP ping. A number of tools and techniques are available for easily carrying out these attacks [1]. DDoS attacks generally inflict damage by 1) Consumption of network bandwidth, the processing power of networking equipment 2) Denial of services to the legitimate users and 3) Disruption of decision making of switches by modifying the changes in their flow tables in Software Defined Networks.

The most common amongst these attacks are ping flooding, Synflooding, ping of death, Smurf and Fraggle distributed DoS attacks. Ping flood is simply ICMP echo command, which is used by network administrators or users to check the reachability of network equipment. It becomes lethal when used with "-n" and "-i" option as they provide the option to the user to decide on the number of times a ping request is sent and size of data respectively. Ping of Death is another form of DDoS attack, which exploits the outer limit of TCP for receiving data. When the attacker sends the packet greater than 65.535 bytes' in fragments and these are assembled at the receiving end, confuse the receiver. This causes a buffer overflow and computer/ server crashes. However, this anomaly has been taken care of in computers made after 1998. 'Synflood attacks' exploits Transport Control Protocol's three-way handshaking. The attackers spoof the IPs of the sender computer and send the SYN request to the server. However, since, they do not respond to a TCP-acknowledge request of the server thus leaving the TCP connection half-open. Secondly, the attacker does not send the acknowledge request at all. Smurf attacks are carried out by spoofed IPs. However, the most networking equipment manufacturer has taken care of this to some extent and but the total mitigation is still evasive. "Fraggle" is just another flavor of Smurf attack caused by using UDP packets, unlike TCP packets. DNS amplification attack is initiated sending a DNS name lookup request to a public DNS server. The source IP address of the targeted victim is spoofed. The attacker tries to request as much zone information as possible, thus amplifying the DNS record response that is sent to the targeted victim. The traffic directed at a target is increased manifold easily by even a smaller request size. SNMP and NTP are used as a reflector in an amplification attack.

B. Machine Learning Algorithms

The concept of Machine Learning of algorithms is not new. However, this decade has witnessed renewed importance of Machine Learning in many fields. These algorithms are used to solving complex problems. Any algorithm should be able to work optimally and correctly for the intended work. Machine Learning can be applied to DDoS detection and it has found to be far better than "signature" based detection. Machine Learning algorithms are trained to detect anomalous behavior and they themselves do the anomaly detection. Machine Learning algorithms [13–15] are of four types: (a) supervised learning, (b) Unsupervised learning, (c) Semi-supervised Learning, and (d) Reinforcement Learning. These algorithms are trained with some training data and the evaluation is done by using the test data. To check the accuracy of the algorithm, the test data is required. In supervised learning, the algorithm is presented with some dataset and is asked to make a prediction. However, in the case of unsupervised learning, the algorithm or system makes a decision themselves without any support data. Semi-supervised learning falls in between the supervised and unsupervised learning algorithms. Finally, in reinforcement learning, the system working in a real environment is provided with feedback in terms of rewards and punishments. Alter-

natively, the Machine Learning algorithms can be classified based on the desired outputs namely regression, clustering, density estimation, dimensionality reduction etc. K-mean and fuzzy c algorithm come under the classification algorithm. In the k-mean algorithm, k is the number of clusters selected randomly and then the distance of data points from these centroids is calculated. The iterations go on until the centroids coincide and further changes in position stop. Fuzzy C is another classification algorithm where overlapping attributes of data are also taken into consideration. Here conditional probability is used. Machine Learning also includes but doesn't limit to Neural Network, Bayesian, Genetic algorithm, Decision Tree, Support Vector Machine etc.

C. Supervised ML Algorithms

Our study examined the performances of five supervised Machine Learning classifiers [14] namely Support Vector Machine (SVM), Gaussian Naïve Bays (GNB), K nearest neighbor and Random Forest. A classifier can be defined as a function allocating a population's element value from one of the available categories. SVM is a supervised Machine Learning algorithm for classification/regression problems. SVM is trained on data and based on it, classifies new data. It classifies the data into different classes by finding a line (hyperplane) which separates the training dataset into classes. SVM uses margin maximization by trying to maximize the distance between various classes. If the line that maximizes the distance between the classes is identified, the probability to generalize well to unseen data is increased. An advantage of Using SVM is that it offers the best classification performance (accuracy) on the training data. The best thing about SVM is that it does not make any strong assumptions on data. Moreover, it does not over-fit the data. These algorithms find the boundary that separates classes by as wide a margin as possible. When the two classes cannot be clearly separated, the algorithms find the best boundary they can. Linear SVM is a flavor of SVM, which runs quickly. These Machine Learning algorithms work well with feature-intensive data like text or genomic. In addition, they utilize only a modest amount of memory. For a given training dataset comprised of n points, a hyperplane can be depicted as a set of

$$\vec{x} \text{ if } \quad \vec{w}, \vec{x} - b = 0,$$

where w is the normalized vector plane to the hyperplane.

Naïve Bayes [14] is another algorithm chosen by us for study and it performs well when the input variables are categorical. A Naïve Bayes classifier converges faster, require relatively little training data than other discriminative models like logistic regression when the Naïve Bayes Classifier is amongst the most popular learning methods grouped by similarities that work on the popular Bayes Theorem of Probability. Segregating DDoS from legitimate traffic is a simple classification of words based on Bayes Probability Theorem for subjective analysis of content. If the instances have several attributes. Given the classification parameters, attributes, which describe the instances, should be conditionally independent.

For a given vector X having n features represented as $x1, x2....xn$, NB presents a conditional probability model for "m" possible outcomes as follows:

$$p(Cm|x) = \frac{p(Cm)\, p(x|Cm)}{p(x)}.$$

Following equation is derived from the repeated application of chain rule to the conditional probability

$$p(Cm|x1, x2 \ldots xn) \propto p(Cm, x1, x2 \ldots xn) = p(Cm).$$

The binary classifier can be constructed as a function after combining it with a decision rule that assigns class a label e.g. $\hat{y} = Cm$ as follows; where $\hat{y}$ is a label:

$$\hat{y} = \underset{m \in (1,\ldots M)}{\operatorname{argmax}} \; p(Cm) \prod_{i=1}^{n} p(Xi|Cm).$$

Logistic regression [14] known as a sigmoid function in Machine Learning is an idea borrowed from statistics. It uses [14] equation like linear regression but uses binary values

$$y = e \wedge (b0 + bl * x)/(1 + e \wedge (b0 + bl * x)).$$

Here y is the predicted output, b0 is the bias and b1 is the coefficient for the single input value (x); b is the coefficient, an associated value in every column and it must be learned from training data.

Forest is an ensemble of decision trees. Random Forest [14] is the group learning method used for classification. In this algorithm, groups of decision trees with the random subset of the data are formed. A model is trained several times on a sample selected from dataset randomly to achieve good prediction per-

formance from the random forest algorithm. Here, the output of all the decision trees in the random forest is combined to make the final prediction. The final prediction of the random forest algorithm is derived by polling the results of each decision tree or just by going with a prediction that appears the most times in the decision trees. For a given training set $X = x1, x2...xn$ with a response set $Y = y1, y2...yn$, the algorithm selects a random sample and after training, the prediction for new or unseen samples x' can be made averaging all regression tree f_b over x' as follows:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x').$$

K-nearest neighbor algorithm [15] is generally used for both classification and regression. This is also known as the lazy learning algorithm as calculations are prolonged until classification happens. Euclidean and Hamming distances are used for ascertaining the distance metric. Here k is user-defined unlabeled vector and is labeled during the classification.

Linear Discriminant Analysis is another ML algorithm which generally is applied when the number of classes for classification is more than two. The two assumptions of this model are 1) the data is Gaussian and 2) and every attribute of the data has the same variance.

4. EXPERIMENTAL SETUP AND DATASET

We established our experimental setup over an "Ubuntu" operating system installed as a virtual operating system on a Windows machine having Windows 10 Professional operating system. The specifications of the laptop were Intel Core (TM) 2 Duo CPU, 4 GB RAM. Various research tools were utilized. A python based application "DDoSD" (DDoS detector) was developed using Scikit-learn [18], a very popular Machine Learning based libraries to build five Machine Learning based models for network classification.

We chose Intrusion Detection Evaluation Dataset [19] from the Canadian Institute of Cybersecurity. This Institute has provided various benchmark datasets since 1998 to investigate various solutions for network security. They were kind enough to offer us the dataset. The vital aspect of this dataset is that it is contemporary unlike other datasets used until now for network threat detection. This dataset reflects the latest attacks and benign traffic as well to create a more "normal" or "real" scenario. This dataset has included the most common attacks based on the 2016 McAfee report, such as Web-based, brute force, DoS, DDoS, Infiltration, Heart-bleed, Bot, and Scan covered in this dataset. Dataset is huge running into GBs and their derived features along with flows are presented as labeled data, which we used to train and test our Machine Learning models for network classification. Boosting the accuracy for the prediction of results requires a "right dataset." After considering many datasets, we picked CIC IDS 2017 dataset for our research. This is not only the latest but comprehensive also. Other datasets like KDD99, etc are old and have lost their relevance as much new kind of DDoS attacks have surfaced recently and we wanted a dataset, which can mirror the real world. Therefore, CIC IDS 2017 dataset was an ideal choice. The dataset we chose to train our models has about 85 features and 2, 25,725 instances. A detailed comparative analysis of the Intrusion Datasets by Gharib et al. proved its edge over other Intrusion Detection datasets [25]. The authors evaluated various datasets over eight criteria e.g. Attack Diversity, Anonymity, Available Protocols, Complete Capture, Complete Interaction, Complete Network Configuration, Complete Traffic, Feature Set. From their study, it emerged that CICIDS2017 dataset; definitely was the latest and superior to the likes of CAIDA, KDD99, DEFCON, ADFA, LBNL, KYOTO etc.

5. RESULTS AND DISCUSSIONS

A. Findings

We ran the experiments initially without pre-processed data. We used python for development of software application and the pseudo code for the application is illustrated in Fig. 1. The time taken by most of the classifiers was astonishingly high. When we narrowed down to the problem areas, it was revealed this happened due to the huge dataset size and application of Machine Learning algorithms is CPU and memory intensive. After pre-processing and fine-tuning of a laptop for best performance i.e. choosing 2GB of storage space as virtual memory in addition to the existing RAM, the hiccups disappeared and paved the way for the successful execution of experiments. Data cleansing and pre-processing of data are considered as the first steps before building any Machine Learning model. They help in improving the models' performance. Our dataset contained blank, NaN and Infinity values. There are many ways in python to deal

```
IMPORT important libraries including Scikit Learn
IMPORT the dataset
PRE-PROCESSING to impute missing values, replace NaN
(Not a Number) and Infinity values in the dataset
SCALE the data
STOREvarious Machine Learning Models in a variable 'models'
SET scoring equal to accuracy and ROC
SET Name as name of the Machine Learning models
FOR Name, Model in models:
      Store value of model_selection using 10 splits in a variable
      Calculate and store results using cross_val_score method
of model_selection in skleam by imputing Train and Testing
data
      Append results in list of existing results
     Print mean accuracy and standard deviation
END FOR
```

Fig. 1. Pseudo Code for comparing ML models.

with such data. These values are necessary to deal with for optimally running the Machine Learning models. Pre-processing of data is very crucial as it boosts the accuracy of Machine Learning models. We impute the missing values and replaced the NaN values with the median of the values in that column and Infinity values with "0." "SelectPercentile" method was used for the feature selection and we successfully brought down the number of the feature was reduced from 85 to 12 to further boost the classifiers' performance. These many steps reduced the time taken by classifiers considerably. We also replaced the categorical labels i.e. BENIGN and DDoS with "0" and "1" respectively as the Machine Learning models work only on a float/numerical values.

The Machine Learning models perform poorly due to "underfitting" or "overfitting." Overfitting refers to learning of a Machine Learning model over the data and noise as well as impacting its performance negatively. Underfitting is another unwanted phenomenon and it happens when Machine Learning models fail to fit the training data well and thus fails to make out the underlying trends. Underfitting creeps in when we subject our models to split and train/test strategy. Practically splitting renders lesser data for training. The left out data might contain important patterns, which induce an error by bias. *K* Fold cross-validation does help us out in avoiding these situations. We, therefore, used *K* fold cross-validation while choosing $k = 10$ for evaluation and comparison of our Machine Learning models and thereafter we calculated averages for ascertaining the mean accuracy. Generally, *k* is taken as 5 or 10 to have good results but it is not a thumb of the rule and one can choose any value for *k*. In *K* Fold cross-validation, the data is divided into k subsets and for every subset split and train/test strategy is applied to the model. The *k*-1 subsets are used to train the model. In this strategy, every data point enters validation as well as training dataset helping in reduction of bias.

We used "Classification accuracy" and Receiver Operating Characteristic (ROC) curve as metrics to evaluate the performance of these classifiers, though the other metrics like Confusion Matrix and classification reports etc are also available. Classification accuracy is the ratio of the number of correct predictions made to all predictions made. ROC is plotted as a true positive rate (Sensitivity) function of the false positive rate (Specificity) for different points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to particular decision thresholds. Results of our experiment for classification metrics are illustrated in Fig. 1. The experiment revealed that random forest classifier though takes relatively higher time but the accuracy of predictions is definitely better than other classifiers.

We also tried fine tuning of the supervised algorithms. We picked top two performers i.e. and Random Forest when the experiments were carried out with default values. We then explored the optimization of parameters of KNN and Random Forest and the outcome was intriguing.

The by the default value of n neighbors (number of neighbors) is 5 and if we reduce it to 3 and 1 the performance of KNN is improved but still remain lower than Random Forest. Similarly, by default, the value of *n*_estimator that is a parameter of Random Forest is 10 and when we increased its value to 20, the performance improved and on the contrary if it reduces on decreasing the value below 10. The other by default important parameter in case of Random Forest Classifier is job=one, which is the number of

Table 1. Results of Classification Accuracy

#	Results for classification accuracy		
	machine learning models	mean accuracy	standard deviation
[1]	Logistic Regression	0.824770	0.003
[2]	K Nearest Neighbour	0.943640	0.001
[3]	Gaussian NB	0.810424	0.002
[4]	Random Forest	0.961341	0.001
[5]	Linear SVM	0.823536	0.002
[6]	KnearestNeighbors with n_neighbors=3	0.951	0.001
[7]	Random Forest with n_estimator = 20	.965	0.001
[8]	Linear Discriminan Analysis	.821	0.003

Table 2. Results of ROC

#	Results for ROC		
	machine learning models	ROC	standard deviation accuracy
[1]	Logistic Regression	.873	.002
[2]	K Nearest Neighbour	.975	.001
[3]	Gaussian NB	.848	.002
[4]	Random Forest	.990	.001
[5]	Linear SVM	.877	.002
[6]	K Nearest Neighbour with n_neighbors=1	.938	.002
[7]	Random Forest with n_estimator = 20 n_job=1	.993	.00
[8]	LinearDiscriminanAnalysis	.870	0.002

CPUs. Further, if the machine has more CPUs, than the performance of Random Forest will surely improve as the jobs will be shared between the CPUs for simultaneous execution and the time taken for completion will surely decrease.

By plotting Machine Learning model evaluation results, we evaluated the spread and the mean accuracy of each model as depicted in Fig. 1. The overall results of average mean accuracy and standard deviations are given in Table 1. The performance of Random Forest Classifiers was reasonably good i.e. 96.13%. We also used "roc_auc (Receiver Operating Characteristic curve)" as a metric to evaluate our Machine Learning models, the results obtained are tabulated in Table 2. Random forest showed .99 as mean accuracy for ROC metric and in ROC, the value closer to one is the better. What stands out from the results obtained after applying both the metrics is that Random forest has performed better than every other classifier. This may be considered as further validation of our results. If we compare our results with the works of [21–24] for application of Random Forest Classifiers utilized in several other fields (Table 2), it came out that our, model achieved better accuracy. In the context of the dataset, we used and the methods of pre-processing, Random Forest scored over others.

From our findings, we extrapolated that supervised Machine Learning algorithms are very potent and effective techniques in the classification of network flows, of course, if applied with pre-processed data. The results illustrated in Figs. 1 and 2 addressed both research questions i.e. about best performer among the supervised Machine Learning models and the accuracy results of using real-life dataset i.e. CIC IDS 2017 dataset.

B. Limitations

The clear drawback of our work is that we have used a single technique for data pre-processing for evaluation of all the Machine Learning models. On the contrary, the performance of every Machine Learning models can be boosted by using specific data processing techniques. Another limitation of our work is the time taken by the Random Forest classifier despite outperforms other Machine Learning models. In our

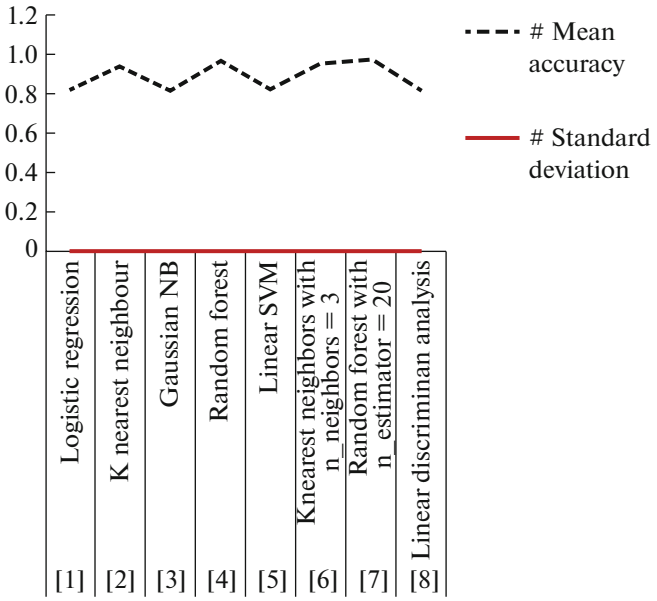

Fig. 2. Performance of Machine Learning models.

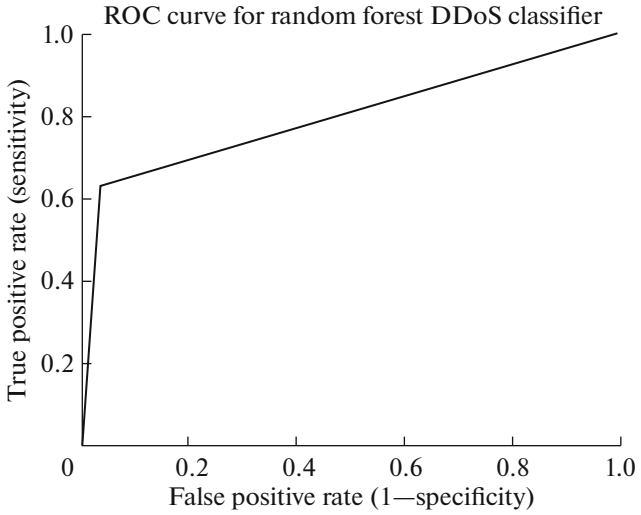

Fig. 3. ROC for RFC.

future work, we will try to address these shortcomings to shorten the detection time and increase the overall performance of these models.

6. CONCLUSIONS

Due to increased sophistication in the DDoS attack invoking methodologies and easy availability of related tools over the internet for, the detection and mitigation of the same has become very difficult. Machine Learning models are anomaly detection techniques, which are accurate and practical methods to identify DDoS traffic from legitimate traffic. These ML-based classifiers can be trained and tested using a real-life Intrusion Detection datasets, for better performance in real scenarios.

Keeping the above in mind, we contrast five such ML algorithms. The results obtained in the trials have revealed that our study successfully addressed both the research questions raised in this paper. Our study has also lead to the conclusions: (1) the Random forest classifier is a better choice in case of DDoS detection and (2) the accuracy achieved in the trials is over 96% over a real-life dataset. It clearly emerges out

Table 3. Comparison of Outcomes for RFC

#	Performance of random forest classifiers (RFC)		
	authors of research work	dataset used	prediction accuracy
[1]	Bharathidason and Venkataeswaran [27]	Multiple datasets	61–96%
[2]	Mellor, Haywood Stone and Jones [28]	776 Land Cover Maps	96
[3]	Almseidin, Alzubi, Kovacs and Alkasassbeh	KDD Intrusion	93.7
[4]	Bindra and Sood	CIC IDS2017 (Friday)	96.2

from TABLE 3, after comparing our work with that of others, that the prediction accuracy of Random Forest Classifier in our work is comparable or even better to them. We also substantiated the outcomes by using two metrics in our experiment i.e. Classification accuracy and Receiver Operating Characteristic (ROC). In addition, the use of K-fold cross-validation made our results more acceptable as the same does away with the problem of "underfitting." There are many interesting leads, which can be pursued to further this research work. One interesting work would be to evaluate and compare the influence of other data pre-processing techniques on the accuracy of these Machine Learning models.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

1. Cybersecurity Trends, 2018. https://www.incapsula.com/ddos/attack-glossary/high-orbit-ion-cannon.html. Accessed February 5, 2018.
2. DDoS Attack, 2018. https://en.wikipedia.org/wiki/Denial-of-service_attack. Accessed February 8, 2018.
3. Hacking Incidents, 2018. https://en.wikipedia.org/wiki/List_of_security_hacking_incidents. Accessed February 15, 2018.
4. Transformation of DDoS attacks in Global warefare, 2018. https://qz.com/860630/ddos-attacks-have-gone-from-a-minor-nuisance-to-a-possible-new-form-of-global-warfare/. Accessed January 1, 2018.
5. DDoS attacks Trend Report, 2018. https://www.cdnetworks.com/CDNetworks_Q3_2017_DDoS%20Attack%20Trends%20Report_EN_201712.pdf. Accessed February 26, 2018.
6. Robinson, R. and Thomas, C., Ranking of machine learning algorithms based on the performance in classifying DDoS attacks, *Proceedings of the IEEE Recent Advances in Intelligent Computational Systems (RAICS),* Trivandrum, 2015, pp. 185–190.
7. Azab, A., Alazab, M., and Aiash, M., Machine learning based Botnet identification traffic, *2016 IEEE Trustcom/BigDataSE/ISPA,* Tianjin, 2016, pp. 1788–1794.
8. Agrawal, P.K., Gupta, B.B., and Jain, S., SVM based scheme for predicting number of zombies in a DDoS attack, *2011 European Intelligence and Security Informatics Conference,* Athens, 2011, pp. 178–182.
9. Chonka, A., Zhou, W., Singh, J., and Xiang, Y., Detecting and tracing DDoS attacks by intelligent decision prototype, *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom),* Hong Kong, 2008, pp. 578–583.
10. Jalil, K.A., Kamarudin, M.H., and Masrek, M.N., Comparison of machine learning algorithms performance in detecting network intrusion, *2010 International Conference on Networking and Information Technology,* Manila, 2010, pp. 221–226.
11. Balkanli, E., Alves, J., and Zincir-Heywood, A.N., Supervised learning to detect DDoS attacks, *2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS),* Orlando, FL, 2014, pp. 1–8.
12. Saad, S., et al., Detecting P2P Botnets through network behavior analysis and Machine Learning, *2011 Ninth Annual International Conference on Privacy, Security and Trust,* Montreal, QC, 2011, pp. 174–180.
13. Application of Machine Learning, 2018. https://medium.com/app-affairs/9-applications-of-machine-learning-from-day-to-day-life-112a47a429d0. Accessed February 5, 2018.
14. Ayon Dey, Machine learning algorithms: A review, *Int. J. Comput. Sci. Inf. Technol.,* 2016, vol. 7, no. 3, pp. 1174–1179.
15. Logistic Regression, 2018. https://machinelearningmastery.com/logistic-regression-for-machine-learning/. Accessed December 16, 2017.

16. Types of Machine Learning Algorithms, 2017. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861. Accessed December 12, 2017.
17. Supervised Machine Learning, 2017. https://en.wikipedia.org/wiki/Supervised_learning#Algorithms. Accessed October 2, 2017.
18. Sci-kit Learn, Machine Learning in Python, 2017. http://scikit-learn.org/stable/. Accessed November 5, 2017.
19. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, *4th International Conference on Information Systems Security and Privacy (ICISSP),* Portugal, 2018.
20. DDoS Attacks, 2017. https://en.wikipedia.org/wiki/Denial-of-service_attack. Accessed November 14, 2017.
21. Chaudhary, A., Kolhe, S., and Kamal, R., An improved random forest classifier for multi-class classification, *Inf. Process. Agric.,* 2016, vol. 3, no. 4, pp. 215–222.
22. Bharathidason, S. and Venkataeswaran, C.J., Improving classification accuracy based on random forest model with uncorrelated high performing trees, *Int. J. Comput. Appl.,* 2014, vol. 101, no. 13, pp. 26–30.
23. Mellor, A., Haywood, A., Stone, C., and Jones, S., The performance of random forests in an operational setting for large area sclerophyll forest classification, *Remote Sens.,* 2013, vol. 5, no. 6, pp. 2838–2856. https://doi.org/10.3390/rs5062838
24. Almseidin, M., Alzubi, S., and Kovacs, M., Alkasassbeh, Evaluation of machine learning algorithms for intrusion detection system, *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 2017, pp. 277–282.
25. Gharib, A., Sharafaldin, I., Lashkari, A.H., and Ghorbani, A.A., An evaluation framework for intrusion detection dataset, *Proc. 2016 International Conference on Information Science and Security (ICISS),* 2016, pp. 1–6.