A Study of AI and ML Algorithms for DDoS Cyberthreats Detection

Harleen Kaur Taunque and Somesh Kumar Gupta

University of Waterloo, Waterloo, ON, CA, N2L

1 Abstract

Distributed Denial of Service (DDoS) is a type of Cybersecurity threat which is one of many versions of Denial of Service(DoS) that uses IP addresses to attack a particular server/victim. DDoS threats are well-coordinated attacks that uses a compromised secondary victims to target the single or multiple victim systems may it be a large firm server or a small scale system. The DDoS threats are very costly in terms of bandwidth and power and they result in loss of confidential data as well. Therefore it has become of much importance to devise better algorithms to detect different types of DDoS Cyberthreats with higher accuracy while considering the computation cost in detecting these threats. Most of the study conducted in literature assumes detection of DDoS threats as a binary classification problem and their results give out whether an attack was attempted or not. However, in order to effectively defend the network from causing extensive damage, it is of paramount importance to know which type of DDoS attack is targeting the network or the system. This study, presents an Ensemble Classifier that combines the performance of top 4 performing algorithm and compares it with different Artificial Intelligence and Machine Learning (AI and ML) algorithms to effectively detect the different types of DDoS threats by converting the problem to a multilabel classification problem.

2 Introduction

Distributed Denial of Service (DDoS) [1] are flooding threats that deny a legitimate user from accessing its intended service. It is one of the most prevalent threats that the Cybersecurity industry is facing as per the recent market research. Therefore it is of foremost importance to understand and detect different types of DDoS threats. There are mainly two major types of DDoS attacks Refection Based and Exploitation Based as given in [2] and each of the DDoS attacks falls in either of the two categories.

Reflection Based DDoS threats are those kinds of attacks in which the hacker hides its identity by using third-party tools and components. The attack is initiated by sending the data packets to a reflector server with the source and IP address of the victim/target. These attacks are executed through the Application layer protocol using either Transport Control Protocol (TCP), User Datagram Protocol or by using both of them simultaneously. The TCP based attacks include MSSQL, SSDP whereas UDP based attacks include NTP, TFTP and TCP/UDP based combined attacks include DNS, LDAP, NETBIOS and SNMP.

The Exploitation based attacks are similar to that of Reflection based attacks and it also uses third party software and components to remain anonymous when initiating the attack. It has TCP based attacks and UDP based attacks, TCP based attack consists of SYN Flood whereas UDP based attacks consist of UDP-Flood(UDP) and UDP-lag. The hacker initiates the UDP threat by sending a huge amount of UDP packets at a very high rate on the random ports of the target machine which leads to the exhaustion of bandwidth and thus the system crashes. UDP-lag works by disrupting client-server connection which is carried out by a lag(hardware) switch or software that hogs the bandwidth of the network thereby slowing the user. SYN attack works by exploiting the TCP-three-way-handshake which includes sending out SYN packets continuously until the system crashes.

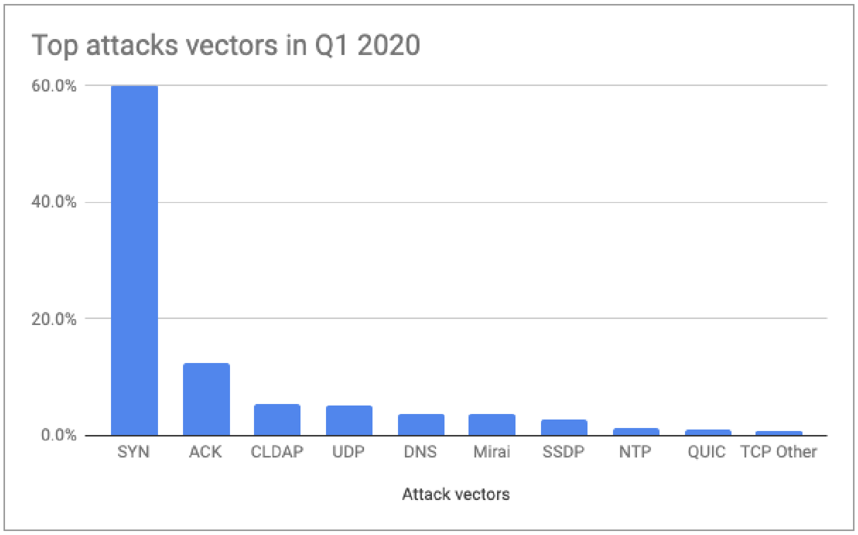

Fig. 1. Top attack vectors in Quarter 1 2020 [3]

The figure 1, taken from the recent 2020 article by Cloudflare [3] depicts the seriousness to classify the type of DDoS attack as SYN type took up to 60% of attack vectors in first Quarter of 2020, so a need to correctly detect them as early as possible arises. A study conducted by Kerbson Security mentioned in the study by Li.et.al [4] reveals that for each one of DDoS attacks it may have incurred a cost of $\$323,973.75$ to the device owners with an additional cost to the excess use of power and bandwidth. Figure 2 also taken from Cloudflare [3] depicts that the bit rate of the network layer of attacks is increasing especially during the COVID-19 crisis as March 2020 had maximum attack bit rate (550 Gbps) in the four quarters. However, this source didn't capture the DDoS attack on Amazon in February 2020 in which the company experienced the largest DDoS attack in history. The Company experienced an attack of 2.3 Terabit per second (Tbps) DDoS attack which lasted more than the average duration for these types of attacks.

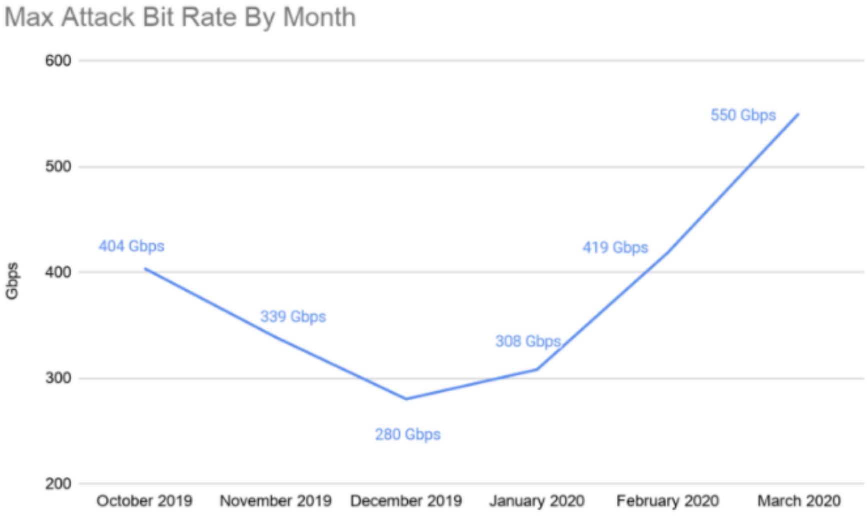

Fig. 2. Maximum Attack Bit Rate by Month [3]

The criticality of the situation makes this one of the major problems of internet security and many statistical detection methods can be found in literature as highlighted in [[5], [6],[7],[8] and [9]] like wavelet-based, port entropy-based , destination entropy-based etc. However, all these methodologies are very time consuming and ineffective as the internet is a widely dynamic field that is constantly changing. Therefore, to solve these issues, many researchers resorted to Machine Learning and Artificial Intelligence methodologies to detect the DDoS attack.

However, there is a lack in the literature to classify the DDoS attacks into the various types which motivated us to use efficient Machine Learning and Artificial Intelligence techniques to capture the variability of the changing internet domains as it is easy to update the ML and AI models. However, as the dataset is large and has 87 features the computational complexity and the prediction time increase. To tackle that we applied feature selection – ExtraTrees classifier technique to select the best relevant 20 features.

The study applies the baseline models and advanced models described in detail in section 4. In addition, the study presents an Ensemble Classifier MV-4 which combines the performance of the top 4 AI/ML models to give a good performing classifier for this dataset. The study is divided into the following section. Section 3 gives an overview of relevant research and study conducted for DDoS Cyberthreat detection, section 4 deals with the description of the methods and parameters used in this study, section 5 gives information about the Dataset and software used along with Metrics used for evaluating AI and ML models. Section 6 discusses testing techniques and section 7 consists of results and analysis for the used model, in the final section we give conclusions, and possible future works.

3 Literature Review

The work present in the literature deals with only binary classification of the DDoS attacks.The study by Balkanli.et.al [10] used Bro and Corsaro open-source systems and Classification and Regression Trees(CART) and Naive Bayes Machine learning classifiers to detect DDoS attacks. They were able to use these machine learning algorithms to obtain high performance without using IP addresses and port numbers. They obtained an accuracy score of 0.99 with a recall value of 1.00 along with an F1-Score of 0.97 on the training dataset using Decision Tree and an accuracy of 0.98, recall value of 0.93 and an F1-Score of 0.82 on the training using Naive Bayes. However, the algorithms devised were highly focused on detecting backscatter traffic and detection of whether there was a DDoS attack or not without focusing on the type of attack.

The study of Siaterlis.et.al [11] uses the Multilayer Perceptron network to successfully classify normal and attack state (binary) or DDos source, DDos victim or normal(3 types) of DDoS type attack with high true positive and low false-positive rate, majorly focusing on UDP attacks. They used a combination of metrics basically complementary passive like packet capturing and Netflow based traffic monitoring and ROC curves to make a reliable decision. Multilayer Perceptron was also used by Barati.et.al [12] to improve binary DDoS detection rate with an accuracy of 99.9971%. A hybrid methodology to select features using the wrapper Genetic Algorithm(GA) and MLP to classify the attack was implemented in the study.

Nguyen.et.al [13] uses k-Nearest Neighbour (K-NN) in his study to classify the status of the network into Normal, pre-attack and attack DDoS class status in-order to pre-identify the DDoS attack. It used the packet types(UDP, ICMP, TCP, SYN), source/destination IP addresses and port numbers as features and uses a selection of handlers and agents, communication and compromise to classify the packets into two phases of pre-attack, 3rd phase of Attack and Fourth Normal phase. Other algorithms such as Naive Bayesian, C4.5 and K-Means have been explored to achieve an accuracy of 91.4%, 98.8% and 85.9% respectively in the paper by Zekri.et.al [14] to binary classify DoS attack majorly targeting layer 3 and layer 4 of OSI 7 layer model.

Enhanced Multi-Class Support Vector (EMCSVM) is used to operate on KDDCup 99 dataset in [15]. The data set used in the study contains only six types of DDoS attacks hence the classification was only limited to six types. In addition, Linear, Polynomial, Radial Bias kernel functions are used with radial bias giving the best accuracy among the other kernel functions. KiruthikaDevi.et.al [16] discussed a new proposed model called Hop count inspection algorithm (HCF)-Support Vector Machine (SVM) to binary classify DDoS attacks as Normal and attack with an accuracy of 98.99% and compared it with other baselines classifiers like Random Forest and Decision Tree with an accuracy of 93% and 61.76% respectively.

The study by arun.et.al [17] uses Genetic fuzzy and Neuro-fuzzy methods as a subsystem of the ensemble to reduce errors and improve detection accuracy. NFBoost algorithm is proposed to finally classify the DDoS attack as normal(no attack) and attack. Cost per sample and detection accuracy of 99.2% were the two metrics used to understand its performance.

Li.et.al [4] highlights the performance of various algorithms like Back Propagation(BP), PCA-BP, Long Short Term Memory(LSTM), PCA-LSTM, PCA-Support Vector Machines(SVM), Recurrent Neural Network (RNN) and Principal Component Analysis-Recurrent Neural Network (PCA-RNN) on basis of accuracy, sensitivity, precision, F1 and prediction time. It concluded that the PCA-RNN method has higher detection efficiency and accuracy using KDD-99 [18] dataset. To understand the network fully, maximum network traffic characteristics were considered, and then PCA was applied for dimensionality reduction to decrease time complexity. However, the dataset used is mostly suitable to detect the wider range of attacks and not particularly DDoS attack [19] because of which the proposed method is only suitable in case of binary classification of DDoS attack. Also, KDD-99 [18] data set is an older version of NSL-KDD [20] which removes many problems of KDD-99 [19] hence the algorithm proposed is not very reliable.

In addition, Rahman.et.al [21] explored J48, RF, SVM and KNN to binary classify DDoS attacks and they concluded that J48 is the best classifier in terms of accuracy, sensitivity, specificity, precision, recall and F1-score. C4.5 algorithm was devised in [22] to detect DDoS attacks by forming a decision tree to effectively detect the signatures of these attacks. A comparison of C4.5 with a machine learning algorithm showed that it has an accuracy of 98.8% for classifying correctly a threat as an attack or normal. However, the previous models only predicted whether there is an attack or not. They do not detect the types of attacks that are critical in CyberSecurity to effectively deploy countermeasures which is the major distinguishing factor in our study.

4 Selected Algorithms

The algorithms for different types of DDoS Cyberthreats detection are selected based on the Computational Complexity of the AI and ML algorithms [23]. This criterion is important in choosing a low Computational Complexity algorithm when the performance of different algorithms is similar. The Table 1 below shows the Computational Complexity of each of the algorithms used in this study.

Table 1. Computational Complexity of Artificial Intelligence and Machine Learning Algorithms [23]

Algorithm	**Training Complexity**	**Prediction Complexity**
SVM	$\mathcal{O}(n^2\ f\ +\ n^3)$	$\mathcal{O}(n_{sv}\ f)$
Decision Tree	$\mathcal{O}(n^2\ f)$	$\mathcal{O}(f)$
XGBoost	$\mathcal{O}(n\ f\ n_{trees})$	$\mathcal{O}(f\ n_{trees})$
Random Forest	$\mathcal{O}(n^2\ f\ n_{trees})$	$\mathcal{O}(f\ n_{trees})$
AdaBoost	$\mathcal{O}(n\ f)$	$\mathcal{O}(f\ n_{trees})$
Neural Network	-	$\mathcal{O}(fn_{l_1} + n_{l_1}n_{l_2} + ...)$

where in Table 1, n is the number of training samples, f being the total number of features in each of the training samples. In addition, n_{sv} is the number of support vectors used in the SVM algorithm. For Neural Networks n_{l_i} is the number of neurons in layer i for a given Neural Network.

4.1 Machine Learning Algorithms

Support Vector Machines (SVMs) SVMs algorithms are a set of supervised learning algorithms that are majorly used for classification and regression.

SVM separates different classes by a hyperplane and produces a classifier that works well on unseen examples which is why it generalizes very well. The hyperparameters were tuned in order to prevent overfitting. In this study, LinearSVC is used from Sci-Kit Learn [24] with a multiclass parameter as one-vs-rest ”ovr” to successfully use it for multilabel classification. Square-hinge was used as the loss function because this function uses simple mathematics which gives computationally effective results. The regularization parameter or the cost parameter was taken as 1 which means that the samples inside the margin will be penalized more so hence they try to correctly classify with a higher value of C. A large value of C was not taken to prevent overfitting and if C was taken too low that is lesser than 1 then it was leading to soft margin. For penalty, l2 normalization was used in penalization as l1 normalization resulted in too sparse coefficients.

Decision Tree Decision Trees are supervised learning algorithms that sort the tree starting from the root node to a leaf node. The leaf node gives the classification that is the label names. Decision trees use hyperplanes/ axis-parallel rectangles that divide the feature space into the classification. Decision Trees are not prone to outliers so lesser data processing is needed. It is used as a baseline benchmark for other algorithms. While implementing Decision Trees we used Gini Index as the splitting criteria and 3 as the sample split value. The best split is used as the splitting strategy at each node. Pruning was not performed to maintain the cost complexity. The features considered to look for the best split were taken to be equal to the maximum number of features.

4.2 Ensemble Learning Algorithms

Random Forest Random based classifier is set of decision trees that are randomly selected from a subset of the training set and then the vote is aggregated from all the decision trees randomly and the final class of the object tested is given. This classifier is mainly used as it is quite efficient with big datasets, it can also handle a large number of input variables without removing any variables. Besides, it also avoids overfitting by improving the accuracy score while training on the dataset. Additionally, as the forest building happens it produces unbiased generalization error estimates. The parameters which give the best accuracy score for this classifier are: 100 number of estimators, minimum sample leaves as 1, minimum sample split as 2 and the Gini criterion is used to measure the quality of the split.

Xtreme Gradient Boosting Extreme Gradient Boosting (XGBoost)[25] is a powerful algorithm that makes use of the gradient Boosting method (GBM) and works best for unstructured data. In this type of boosting technique, the gradient descent algorithm is used to minimize the errors. However, XGBoost further optimizes the GBM by using parallelization, tree pruning, hardware optimization, regularization, sparsity awareness and cross-validation. This algorithm is highly scalable in all scenarios and has higher computational speed on many memory restricted systems [25]. We used ”friedman mse” as a measure of the quality of a split. Mean Squared Error(MSE) is used with Friedman's improvement score. In this study, this criterion provides the best approximation. In addition, it uses deviance as loss function which is equivalent to logistic regression to classify using probabilistic outputs.

Adaptive Boosting Adaptive Boosting or Adaboost is an ensemble method that learns from weak classifier's mistakes and changes it to strong classifiers by using an iterative methodology. This makes this algorithm better than other learning algorithms which predict by random guesses [26]. The base estimator considered for this classifier is Decision Trees. Hence, Adaboost helps in increasing the accuracy of the base estimator. However, it tends to be computationally slow as compared to XGBoost and is very sensitive to outliers and noise. Besides, for this algorithm Gini criterion is used to measure the quality of the split.

Majority Vote Classifier Majority Voting is an Ensemble learning technique that selects the class label that has been predicted by the majority of the classifiers if a class label has received more than 50% of the votes. In this study, we combine the top 4 performing classifiers and get the best approximate for the different class labels for different DDoS threats. This combination of the top 4 classifiers is called MV-4 classifier in this study mainly to separate it from other classifiers. From Table 2 is clear that the top 4 performing models are Random Forest, AdaBoost, Decision Tree and XGBoost. Therefore, we combine these four Classifiers using Majority Voting Technique to come up with an MV-4 classifier. To achieve the combination of 4 different algorithms the code was developed in Python3.7 from scratch.

4.3 Deep Learning Supervised Algorithms

Multi Layer Perceptron (MLP) Multilayer Perceptron [27] (MLP) is a supervised learning algorithm that uses Backpropagation learning method [27] by learning a function $f(\cdot) : R^f \to R^o$ by training procedure on the dataset. Here f is the number of input features and o is the number of class labels. It has the ability to learn non-linear functions in real-time (on-line learning) which increases the application area for MLP. However, it needs proper optimization algorithms due to its concave loss function along with tuning for hyperparameters. Therefore in this study, optimizer used is adadelta along with categorical crossentropy as the loss function. To achieve multiclass classification the Softmax activation function on its output layer and relu activation function in the hidden layers are used to achieve the maximum accuracy score.

Long Short Term Memory (LSTM) Long Short Term Memory (LSTM) [28] is based on Recurrent Neural Network (RNN) architecture that is majorly used in Deep learning with feedback connections which is different from that of Feedforward Neural Networks. LSTM works well for classification, processing and making predictions on time-series datasets. It was mainly developed to overcome the Vanishing and Exploding gradient problem encountered in RNN architecture. The model developed for this study uses adam as an optimizer along with categorical cross-entropy as the loss function. For best accuracy, 2 layers of LSTM were used each with 8 units along with softmax as the output layer for prediction of multiclass labels.

In addition to the Ensemble Learning method, other algorithms are considered mainly because ensemble-based classifiers itself are prone to overfitting when the number of features is higher. Besides, the performance of all the algorithms that are discussed in this study has not been studied in the literature for multilabel classification for different DDoS threats at all. So we decided to study the various algorithms and their performance in detail.

5 Implementation

5.1 Dataset

The dataset used in this study is CICDDoS2019 [2], this dataset is an improvement on its predecessor and improves most of the shortcomings from the previous dataset. The main benefit of using this dataset is that it has analyzed new attacks that are mainly carried out using the TCP/UDP protocols using the Application layer. In addition to the new threats, the dataset also gives priority to profile the abstract behaviour of human interactions by using B-Profile System [29] generating realistic benign background traffic. It has built the abstract behaviour of 25 users based on HTTP, HTTPS, FTP, SSH and email protocols as mentioned in [2]. The dataset [2] divides different types of attack mainly into two categories Reflective and Exploitative Attacks. It is divided into different categories based on the protocols used. It has modern reflective attacks such as PortMap, NetBios, LDAP, MSSQL, UDP, UDP-lag, SYN, NTP, DNS and SNMP.

Using this dataset for multi-class classification for AI and ML algorithms approach is a challenging task as there are many categories of attacks and no research paper has yet used this dataset for multi-class classification for various types of DDoS Attacks detection. In addition, the size of the dataset is almost 26 GB which cannot be directly used for this study on standard machines as to process this amount of data more computation resources are required. So it is of paramount importance to reduce the dataset to effectively train AI and ML models. For reducing the dataset while maintaining the integrity of data along with the distribution Scikit-learn Python Library [24] is used.

5.2 Software used for Implementation

The primary software on which all the programs for each of the Machine Learning algorithms are implemented is in Python 3.7 along with the use of popular libraries such as Numpy and Pandas Python modules. In addition, for implementing the Artificial Intelligence models Keras [30] is used as the application layer and Tensorflow library [31] is used as backend support on Python 3.7.

5.3 Evaluation Metrics

The metrics used for evaluating the Artificial Intelligence and Machine Learning (AI and ML) models are as mentioned below:

Accuracy Score The accuracy score gives the measure of closeness to a specific value. The formula for Accuracy score as given in Sci-Kit learn [24] manual is as given below:

$$Accuracy(y, y') = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(y'_i = y_i) \qquad (1)$$

In equation 1 y is the true label whereas y' is the predicted label which is given by an algorithm after prediction. For multilabel classification, the Accuracy Score gives us the accuracy of the subset. When the entire set of the predicted values given by the algorithm matches the true label the accuracy score is given as 1.0 otherwise it is 0.0.

F1-Score F1-Score which is also known as F-Score or F-Measure. It is a measure of the Test set accuracy. F-Measure calculates the harmonic mean of precision(P) and recall(R) to find the score. The formula for F1-Score is given by:

$$F1\ Score = 2 * \frac{P * R}{P + R} \tag{2}$$

The maximum value attained by F1-score is 1 and it is also known as the Dice similarity coefficient.

Receiver Operating Characteristic Curve The Receiver Operating Characteristic curve (RoC) is a useful tool to evaluate any models for classification based on their performance by considering the False Positive Rate (FPR) and True Positive Rate (TPR). The values of the TPR and FPR are computed by shifting the decision threshold of the classifier. On the RoC curve, the TPR feature lies on the Y-axis whereas, the FPR feature lies on the X-axis. The ideal point for the classifier is at the top left corner where the FPR is zero and TPR is 1 making the classifier ideal for the problem. The larger the area under the curve the better the performance of the classifier.

6 Testing

This section presents the results obtained on the test dataset for each of the AI and ML algorithms. The data in Table 2 shows the accuracy score for each of the algorithms obtained on the multilabel test dataset.

Among the Machine Learning algorithms, the SVM classifier has an accuracy score of 92.75% on a multilabel dataset which is the lowest accuracy score among all the algorithms given in section 4. Besides, the accuracy score for Decision Tree is 98.99% which is higher than that of SVM.

In the case of AI algorithms, the performance of LSTM is slightly better as it has better accuracy score than that of Multilayer Perceptron (MLP) on the test dataset. The accuracy score for LSTM is 98.17% whereas that of MLP is 97.33% which is 0.84% higher than MLP.

The Ensemble learning has four algorithms and the accuracy score for Random Forest is 99.24% which is higher than that of the accuracy score of Adaptive Boosting by 0.23%. XGBoost algorithm has an accuracy score of 98.59% which is lower than that of the Adapive Boosting and Random Forest. In addition, the Majority Voting (MJV-4) algorithm has an classification accuracy of 99.01%. The accuracy score is obtained on the Test dataset and is higher than any other algorithm presented in section 4 except Random Forest. The accuracy score for each of the algorithms is consistent with the F1-Score which is shown in Table 3 and the RoC Curve which is shown in Figure 3-8.

7 Results and Analysis

In this section, we discuss and analyze the results obtained from each of the algorithms on the CICDDoS2019 [2] dataset as mentioned in section 5.1.

Support Vector Machine (SVM) has an accuracy score of 92.75% which is shown in Table 2. It is around 6.24% lower than Decision Tree and approximately 5.84% lower than XGBoost. Besides, the accuracy score of SVM is the lowest among all the algorithms mentioned in Table 2. In addition, Table 3 shows the F1-Score for each of the DDoS CyberThreats. The F1-Score for SYN attack is 1.00 which means it can perfectly detect the SYN Cyberthreats on the system. WebDDoS attacks have an F1-Score of 0.52 which shows that SVM doesn't detect this attack properly. For Benign Traffic, the F1-Score is 0.90 which shows that the algorithm can differentiate well between Normal and abnormal traffic. Also, the SSDP attack has an F1-Score of 0.80 which is lower than other attack types which have a good F1-Score of more than 0.89. By the analysis of RoC Curve for SVM as shown in Figure 3 it is clear that for some attacks such as DNS, NetBios, SNMP etc. have a higher area under the curve while for other attacks such as LDAP, SSDP, etc have a lower area under the curve which is consistent with the F1-Score obtained in Table 2. The macro-average for SVM is 0.98 which better than Decision Tree and XGBoost.

Table 2. Accuracy Score of Artificial Intelligence and Machine Learning Models

Algorithm	**Accuracy Score**	**Design Parameters**
SVM	92.75%	C = 1.0, penalty='l2', loss='squared hinge'
Decision Tree	98.99%	criterion='gini', samples split=3
Random Forest	99.24%	criterion='gini', samples split=2
XGBoost	98.59%	criterion='friedman mse', loss='deviance'
AdaBoost	99.01%	criterion='gini', splitter='best'
MJV-4	99.01%	Random Forest, AdaBoost Decision Tree, XGBoost
MLP	97.33%	activation='relu', optimizer='adadelta'
LSTM	98.16%	activation='softmax', optimizer='adam'

The multilabel accuracy score for Decision Tree is 98.99% which is higher than SVM by approximately 6%. In addition, the F1-Score for Decision Tree is much better for all the attack types except for WebDDoS. The F1-Score for WebDDoS is 0.38 which is lower than the F1-Score of SVM. This algorithm can detect Benign traffic with a score of 0.90 and it can detect the SYN attack perfectly. All the other attack types have an F1-Score of 0.99 which is better than that of SVM F1-Score. In Figure 4, the RoC Curve for each of the attack types for Decision Tree algorithm, from the figure it is evident that most of the attack type the area under the curve is 0.99 as seen in the top left corner and the macro-average is 0.96 which is slightly lower than that of SVM mainly because the WebDDoS attack has a very low area under the curve of 0.62. Besides, for the majority of the attacks, the curves lie on the top left corner and from the metrics analysis it evident that it performs better than that of SVM for this dataset.

The XGBoost algorithm has an accuracy score of 98.59% which is higher than that of SVM and slightly lower than that of the Decision Tree. The F1-Score for this algorithm is shown in Table 3 and it is evident that it doesn't detect WebDDoS and Benign threat properly. The F1-Score of Benign Traffic is 0.46 which is lowest among all the algorithms and it shows that XGBoost doesn't detect the normal traffic properly. However, the SYN threat is perfectly detected which was not the case for SVM and Decision Tree. Apart from this, all other threats have an F1-Score of either 0.98 or 0.99. The RoC Curve for the XGBoost algorithm is shown in Figure 5. It is evident that Benign traffic has a lower area under the curve and the value is 0.45. Besides, for WebDDoS Cyberthreat the area under the curve is 0.00 which tells us that it doesn't detect this attack type at all. The area under the curve for the macro average is 0.86 which is lower than both SVM and Decision Tree.

Table 3. F1-Score of Artificial Intelligence and Machine Learning Models

Threats	**SVM**	**Decision Tree**	**Random Forest**	**XGBoost**	**AdaBoost**	**MJV-4**	**MLP**	**LSTM**
BENIGN	0.90	0.90	0.95	0.46	0.90	0.90	0.92	0.89
DNS	0.90	0.99	0.99	0.98	0.99	0.99	0.97	0.98
LDAP	0.95	0.99	0.99	0.98	0.99	0.99	0.98	0.98
MSSQL	0.89	0.99	0.99	0.98	0.99	0.99	0.99	0.99
NTP	0.98	0.99	1.00	0.99	0.99	0.99	0.99	1.00
NetBIOS	0.93	0.99	0.99	0.99	0.99	0.99	0.97	0.97
SNMP	0.97	0.99	0.99	0.99	0.99	0.99	0.97	0.97
SSDP	0.80	0.99	0.99	0.98	0.99	0.99	0.97	0.98
UDP	0.89	0.99	0.99	0.99	0.99	0.99	0.95	0.97
Syn	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
UDP-lag	0.95	0.99	1.00	0.99	0.99	0.99	0.97	0.98
WebDDoS	0.52	0.38	0.53	0.00	0.32	0.38	0.00	0.00

Random Forest has an accuracy score of 99.24% which is higher among all the AI and ML algorithms presented in this study. This accuracy score is better than the SVM algorithm by approximately 6.50% and by 0.65% from XGBoost. However, the performance of Random Forest and Decision Tree is closely matched and only differ by 0.25%. The F1-Score for Random Forest is given in Table 3 and it is evident that it has the best F1-Score among all the other algorithms present in Table 3. It has three threats SYN, UDP-Lag and NTP which are perfectly detected which is evident from the F1-Score of 1.00. In addition, it has an F1-Score of 0.95 for Benign traffic which is better than all other algorithms which points out that it can effectively differentiate between normal and abnormal traffic. It has an F1-Score 0.53 for WebDDoS attack which is better than the above-discussed algorithms. The RoC Curve for Random Forest is given in Figure 6 and it looks like an ideal classifier for this dataset as the area under the curve for detecting all the attack types is 1.00. Also, the micro and macro average have an score of 1.00.

Adaptive Boosting algorithm is an Ensemble learning approach and in this study, it uses Decision Tree as the base estimator. The accuracy score for this algorithm is 99.01% which is higher than SVM, Decision Tree and XGBoost but slightly lower than the Random Forest algorithm. Besides, the F1-Score for all the different threats is given in Table 3. The F1-Score for Benign traffic is 0.90 which better than XGBoost by a margin of 0.44 and is well detected in this algorithm. Besides, majority of the threats have an F1-Score of 0.99 and SYN Cyberthreat has a perfect F1-Score of 1.00 which indicates that they are very well detected by this algorithm. Figure 7 shows the RoC Curve for this algorithm and, the area under the curve for WebDDoS Cyberthreat is 0.62 which is lower than that of Random Forest. Besides, the micro and macro average are also lower than that of Random Forest. From this RoC we can say that this algorithm performs better than SVM, Decision Tree and XGBoost but slightly lower than that of Random Forest.

The Majority Voting Classifier (MV-4) combines the performance of Random Forest, Adaboost, Decision Tree and XGBoost algorithms which gives an accuracy score of 99.01% for MV-4 which is same as that of AdaBoost and slightly lower than that of Random Forest. From the F1-Score in Table 3 it is evident that most of the threats have an F1-Score of 0.99 whereas SYN has a score of 1.00. In addition, WebDDoS threat has lower score of 0.38 and Benign has a score of 0.90 which is lower than that of Random Forest. Figure 8 show the RoC Curve for MV-4 classifier and the area under the curve is 1.00. Besides, the area under the curve for micro and macro average is 1.00 which shows that the Classifier behaves as an ideal classifier for this dataset [2].

The performance of AI algorithms is slightly lower as compared to that of Machine Learning algorithms. The Accuracy Score for MLP is 97.33% which is higher than SVM by 4.58% and lower than that of Random Forest by nearly 2%. Table 3 shows the F1-Score for MLP algorithm for different types of DDoS Cyberthreats. The F1-Score for WebDDoS Cyberthreat is 0.00 which indicates that it is unable to detect this threat. Besides, Benign traffic has an F1-Score of 0.92 which better than XGBoost and slightly lower than Random Forest. All the other threats have good F1-Score. The RoC Curve for MLP is shown in Figure 9. In this figure, the area under the curve for WebDDoS is 0.52 which affects the area under the curve for macro average to 0.96. Apart from WebDDoS Cyberthreat, all other Cyberthreat has a score of 1.00 which indicates that these threats are properly detected by this algorithm.

The Accuracy Score for LSTM is 98.16% which is slightly better than that of MLP. Besides, this score is 5.41% higher than SVM and around 1% lower than that of Random Forest which is shown in Table 2. In addition, the F1-Score for LSTM on different Cyberthreats is shown in Table 3. From the table, it is evident that the scores are slightly better in all of the cases when compared with the MLP algorithm except for Benign traffic as it decreases from 0.92 to 0.89 from MLP to LSTM. WebDDoS threat has an F1-Score of 0.00 which is the same as that of MLP. The Figure 10 shows the RoC for different attack type using LSTM algorithm. It is evident from the figure that the area under the curve for all the different Cyberthreats is 1.00 along with micro and macro average score of 1.00 which is similar to that of Random Forest. From the RoC curve, it is evident that the performance for LSTM based on the RoC curve is close to an ideal classifier similar to that of Random Forest.

8 Conclusion and Future Work

The multiclass classification for DDoS Cyberthreat was performed by using different AI and ML algorithms and each of the threats was individually identified and validated by using different metrics. An Ensemble Classifier MV-4 was presented for multiclass DDoS Cyberthreat detection which has an accuracy score of 99.01%. In addition, a comprehensive study of different AI and ML algorithms was performed for DDoS multiclass Cyberthreat detection and among all the algorithms Random Forest Classifier has the highest accuracy score of 99.24% followed by MV-4 and AdaBoost Classifier both of which have an accuracy score of 99.01%. However, the F1-Score and results from the RoC curve show that AdaBoost lags behind Random Forest and MV-4 in terms of not detecting a few threats correctly. The F1-Score of Random Forest and MV-4 is very similar to Random Forest having a slight advantage as it detects 3 threats perfectly. The RoC Curve for all the algorithms was presented for a comprehensive analysis for each of the algorithms. From the RoC Curve, it is evident that the performance of MV-4 and Random Forest classifier behaves as an ideal classifier as the area under the curve for both the classifiers for all the different types of Cyberthreats is 1.00. In addition, the micro and macro average have a perfect score of 1.00 making both the classifiers behave like an ideal classifier on this dataset.

As the detection of different types of DDoS threats is successfully implemented which is the main aim of this paper. For future work, our goal is to deploy different solutions for each of the attack types to defend the network from such attacks by using AI and ML algorithms. This work will further be extended to develop a system that can successfully detect DDoS Cyberthreats and deploy countermeasures to prevent critical CyberSecurity threats.

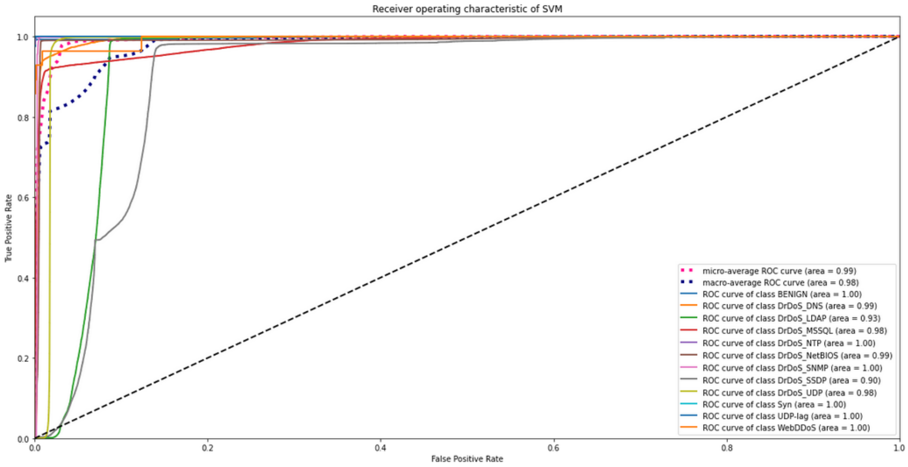

Fig. 3. RoC Curve for Support Vector Machine (SVM)

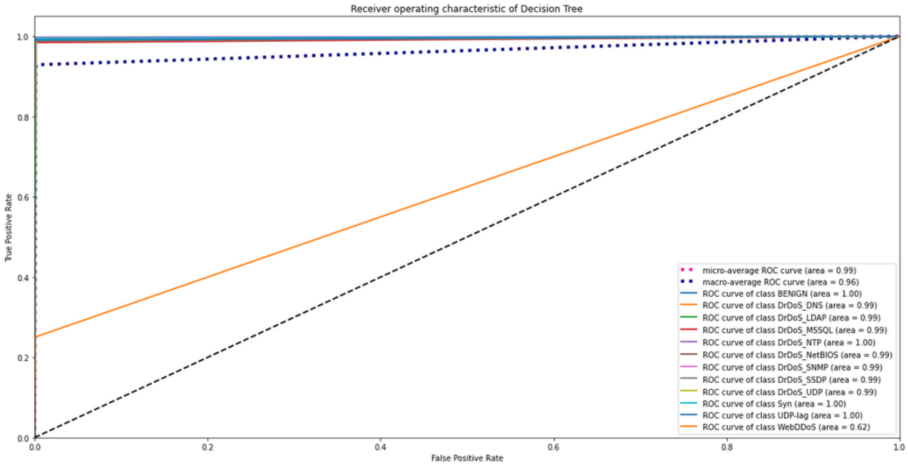

Fig. 4. RoC Curve for Decision Tree

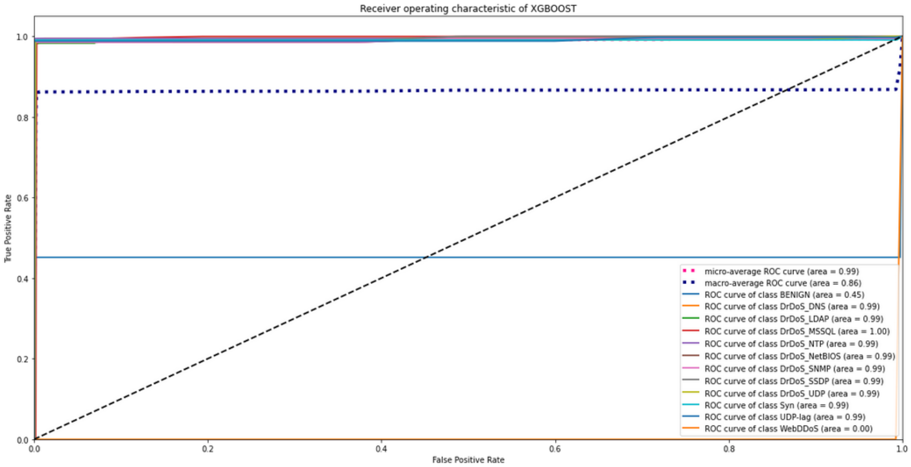

Fig. 5. RoC Curve for XGBoost Classifier

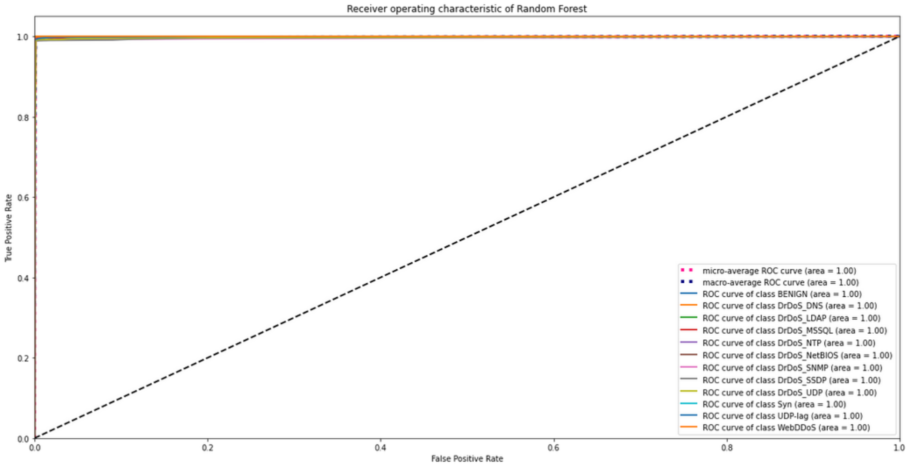

Fig. 6. RoC Curve for Random Forest

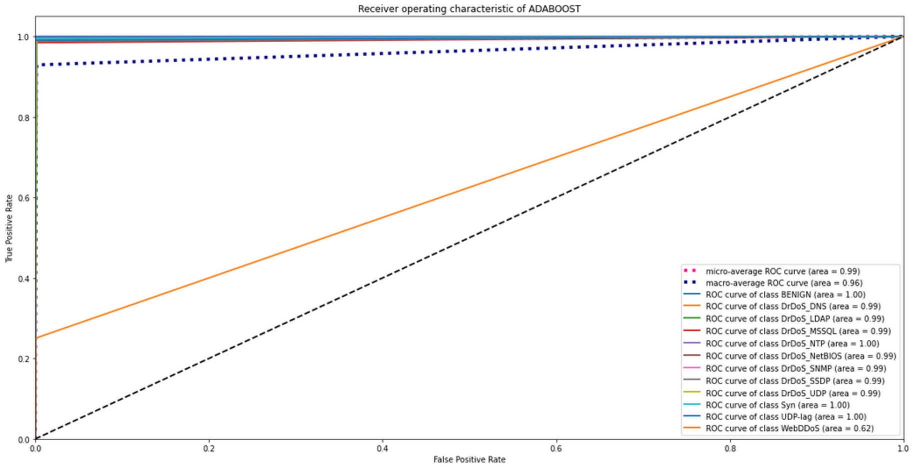

Fig. 7. RoC Curve for Adaptive Boosting Classifier (Adaboost)

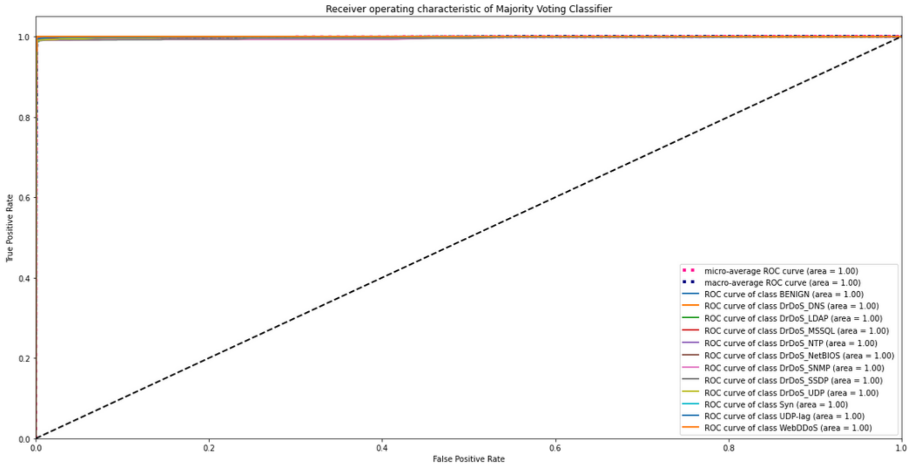

Fig. 8. RoC Curve for Majority Voting Classifier (MV-4)

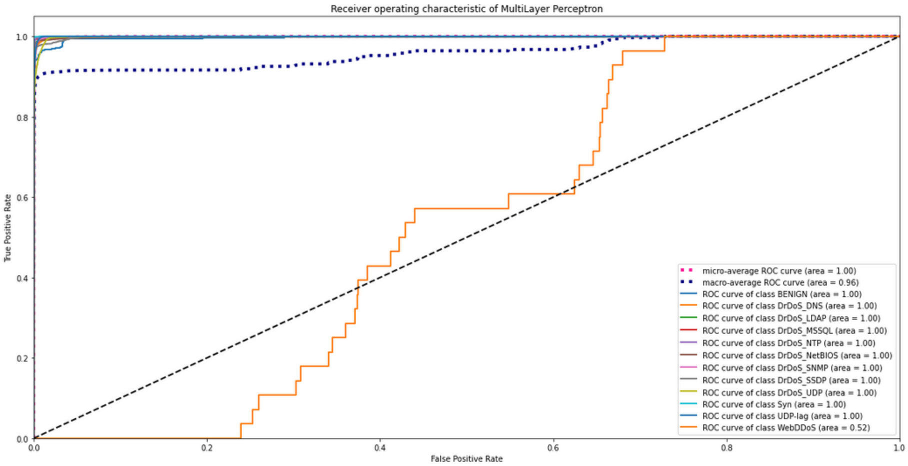

Fig. 9. RoC Curve for MultiLayer Perceptron (MLP)

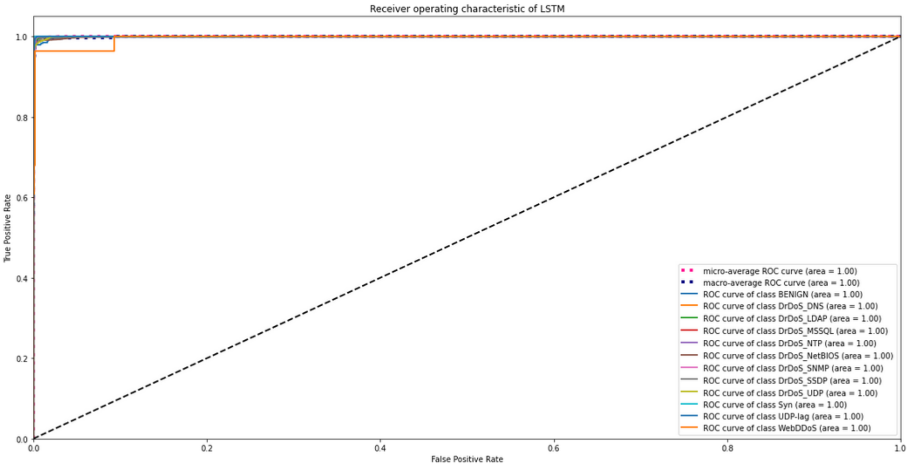

Fig. 10. RoC Curve for Long Short Term Memory (LSTM)

References

1. S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.
2. I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–8, 2019.
3. O. Yoachimik and A. Singh, "Network-layer ddos attack trends for q1 2020," 2020 (accessed August 12, 2020). https://blog.cloudflare.com/network-layer-ddos-attack-trends-for-q1-2020/.
4. Q. Li, L. Meng, Y. Zhang, and J. Yan, "Ddos attacks detection using machine learning algorithms," in *Digital TV and Multimedia Communication* (G. Zhai, J. Zhou, P. An, and X. Yang, eds.), (Singapore), pp. 205–216, Springer Singapore, 2019.
5. M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, "Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions," *The Computer Journal*, vol. 57, pp. 537–556, 03 2013.
6. Y. Tao and S. Yu, "Ddos attack detection at local area networks using information theoretical metrics," pp. 233–240, 2013.
7. S. Mousavi and M. Sthilaire, "Early detection of ddos attacks against sdn controllers," p. 77–81, 2015.
8. X. Ren, R. Wang, and H. Wang, "Wavelet analysis method for detection of ddos attack based on self-similar," *Frontiers of Electrical and Electronic Engineering in China*, vol. 2, pp. 73–77, 01 2007.
9. P. Dong, X. Du, H. Zhang, and T. Xu, "A detection method for a novel ddos attack against sdn controllers by vast new low-traffic flows," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2016.
10. E. Balkanli, J. Alves, and A. N. Zincir-Heywood, "Supervised learning to detect ddos attacks," in *2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pp. 1–8, 2014.
11. C. Siaterlis and V. Maglaris, "Detecting incoming and outgoing ddos attacks at the edge using a single set of network characteristics," in *10th IEEE Symposium on Computers and Communications (ISCC'05)*, pp. 469–475, 2005.
12. M. Barati, A. Abdullah, N. I. Udzir, R. Mahmod, and N. Mustapha, "Distributed denial of service detection using hybrid machine learning technique," in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, pp. 268–273, 2014.
13. H.-V. Nguyen and Y. Choi, "Proactive detection of ddos attacks utilizing k-nn classifier in an anti-ddos framework," vol. 4, no. 3, 2010.
14. M. Zekri, S. E. Kafhali, N. Aboutabit, and Y. Saadi, "Ddos attack detection using machine learning techniques in cloud computing environments," in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pp. 1–7, 2017.
15. T. Subbulakshmi, K. BalaKrishnan, S. M. Shalinie, D. AnandKumar, V. GanapathiSubramanian, and K. Kannathal, "Detection of ddos attacks using enhanced support vector machines with real time generated dataset," in *2011 Third International Conference on Advanced Computing*, pp. 17–22, 2011.
16. B. S. Kiruthika Devi, G. Preetha, G. Selvaram, and S. Mercy Shalinie, "An impact analysis: Real time ddos attack detection and mitigation using machine learning," in *2014 International Conference on Recent Trends in Information Technology*, pp. 1–7, 2014.

17. P. Kumar and S. Selvakumar, “Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems,” *Computer Communications*, vol. 36, p. 303–319, 02 2013.
18. “Kdd cup dataset 2019,” Oct.2019 (accessed May 30, 2020). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
19. Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, “Network intrusion detection based on supervised adversarial variational auto-encoder with regularization,” *IEEE Access*, vol. 8, pp. 42169–42184, 2020.
20. “Nsl-kdd dataset,” Oct.2019 (accessed May 30, 2020). https://www.unb.ca/cic/datasets/nsl.html.
21. O. Rahman, M. A. G. Quraishi, and C. Lung, “Ddos attacks detection and mitigation in sdn using machine learning,” in *2019 IEEE World Congress on Services (SERVICES)*, vol. 2642-939X, pp. 184–189, 2019.
22. M. Zekri, S. E. Kafhali, N. Aboutabit, and Y. Saadi, “Ddos attack detection using machine learning techniques in cloud computing environments,” in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pp. 1–7, 2017.
23. RUser4512, “Computational complexity of machine learning algorithms.”
24. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python ,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
25. T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *2016 kDD’16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.
26. Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *ICML*, 1996.
27. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*, p. 696–699. Cambridge, MA, USA: MIT Press, 1988.
28. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
29. I. Sharafaldin, A. Gharib, A. Habibi Lashkari, and A. Ghorbani, “Towards a reliable intrusion detection benchmark dataset,” *Software Networking*, vol. 2017, pp. 177–200, 01 2017.
30. F. Chollet *et al.*, “Keras.” `https://keras.io`, 2015.
31. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.