An Efficient Approach for Intrusion Detection in Sensor Networks

Sai Teja Gudupati
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
stgudipa@uwaterloo.ca

Shadman Raihan
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
s2raihan@uwaterloo.ca

Somesh Kumar Gupta
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
sk2gupta@uwaterloo.ca

***Abstract*—Wireless sensor network (WSN) is revolutionizing the field of the Internet of Things (IoT) due to its ease of deployment along with real-time applications. The number of WSN devices connected to the Internet is increasing at a rapid rate and this connectivity is also leading to many Security vulnerabilities in the WSNs. Under such scenarios, it becomes critical to detect the correct type of Intrusion as early as possible to deploy corrective measures to minimize the risk. In this study, we choose an efficient algorithm from different Artificial Intelligence and Machine Learning algorithms (AI&ML) for multi-class classification of different Denial of Service (DoS) Intrusions detection based on lowest Computational Complexity while considering the performance of the algorithm for multi-class classification as WSNs have limited resources allocated to it.**

***Index Terms*—DoS, WSNs, AI, ML**

I. Introduction

Wireless sensor networks (WSNs) are one of the most emerging technologies due to its size, cost-effectiveness and easily deployable nature. Since WSNs is becoming a globally adopted technology, the privacy and security concerns of the network have become a key issue. Due to the self-configuring capabilities of WSN devices, it is vulnerable to various types of security attacks. This can lead to loss of control over the WSNs, security breach, eavesdropping and monetary loss.

Denial of Service (DoS) is one of the most dangerous security attacks which denies an intended user by making the network resources temporarily unavailable. It does that by sending superfluous requests to overload the system. It uses a single connection to attack the server. DoS attacks can lead to a huge monetary loss for organizations and disrupt various services including healthcare, home, smart transport, smart tourism, smart energy etc [1]. Researchers around the world have introduced various Intrusion detection systems (IDSs) to address this issue. IDS may use various detection techniques including anomaly, signature and specification whereas, in this paper, we mainly analyze anomaly-based IDS. The performance of the anomaly-based IDS depends on the classification algorithm and the dataset used for training the algorithm. The classification of various types of DoS attacks is done by analyzing network traffic logs which include information about both normal and abnormal cases.

In this paper, the multi-class classification of various types of DoS attacks is performed using both Artificial Intelligence and Machine Learning (AI&ML) algorithms. A total of 5 different AI and ML algorithms is chosen based on the Computational Complexity of the algorithm. Among the Machine Learning algorithms, Gaussian Naive Bayes, Random Forest and Decision Tree are selected as they have lower Computational Complexity as compared to other algorithms. Whereas in the case of Artificial Intelligence, Multilayer Perceptrons (MLPs) and Long Short Term Memory (LSTM) algorithms are selected as the MLPs can learn detailed features whereas LSTMs have better performance for sequential data. While applying these techniques for abnormality detection, the complexity of the algorithms and their performance for multi-class classification is taken into account while selecting the best algorithm for deploying in WSNs. Since most of the WSNs are placed in a remote area without a continuous power supply, the longevity of these networks becomes a vital issue. The approximate power consumption can be represented in terms of the complexity of the algorithms since it represents the number of resources that are required to run the algorithm.

The KDD-CUP-99 [2] and NSL-KDD dataset [3] are the two most common datasets used for the evaluation of signature-based IDS algorithms. KDD-CUP-99 has redundant data in the training set which makes the algorithms biased towards frequent records. Mahbod.et.al. [3] proposed NSL-KDD dataset which was immune to above-mentioned shortcomings of KDD-CUP-99 dataset [2]. Both KDD-CUP-99 and NSL-KDD dataset has 41 features and 23 attack types. The features of the dataset include various network parameters like duration, protocol type, service, flag etc. The dimensionality of the dataset is reduced by selecting 10 most important features out of 41 features by using the chi-squared algorithm [4]. The number of samples for each of the attack types are different for both KDD-CUP-99 and NSL-KDD dataset. In KDD-CUP-99, smurf attack type has the highest number of samples in the dataset while the spy attack type has only 2 samples. On the other hand, the NSL-KDD dataset has the highest sample size for the benign attack type whereas the spy attack has the lowest sample size. Therefore, the dataset is not well balanced as it can be seen in the graph presented in Figure 1 and 2. For implementing a multi-class classification a well-balanced

dataset is required as each attack type should have a sufficient number of samples so that the algorithms can learn all the feature variations to effectively perform multi-class classification. KDD-CUP-99 and NSL-KDD dataset lack a sufficient number of samples for some of the attack types which affects the classification accuracy in some cases which is shown in the Results section. Due to the unbalanced dataset for different attack types, the performance of the Neural Networks(NNs) dropped while performing multi-class classification as NNs requires large dataset for training.

II. Literature Review

There are different studies [5], [3], [6], [7], [8] and [9] conducted for DoS Intrusion Detection algorithms which is discussed in detail. The studies [9], [5], [3], [6] and [8] has presented the problem of Intrusion Detection as binary classification problem and applied different AI&ML algorithms for classification. However, this seems ineffective as in order to perform counter measures for the type of attack it is important to detect the type of DoS attack the network is being targeted with as the counter measures differ with the type of attack. Therefore, there is a need to perform multi-class classification to detect the type of DoS attack. There are studies [6], [8] that have conducted multi-class classification using different AI&ML algorithms without considering the runtime complexity of the algorithms.

KDD-CUP-99 [2] is one of the most used datasets for bench-marking IDSs. Mei.et.al [10] proposed a robust Principal Component Analysis(PCA) based classifier for intrusion detection. The classifier uses an unsupervised learning algorithm where anomalies are considered to be an outlier. This classifier has been applied on the KDD-CUP-99 dataset which has achieved 98.94% in recall and 97.89% in precision while keeping the false alarm rate at 0.92% in case of binary classification. However, only a few IDS solutions are using anomaly detection approaches and experts think that this technology is not mature enough.

The KDD-CUP-99 dataset lacks a well-balanced sample size for each of the attack types which is detrimental for multi-class classification. Mahbod.et.al. [3] pointed out this deficiency of the KDD-CUP-99 dataset and proposed a novel solution to fix it. The authors proposed NSL-KDD dataset which includes selected data from original KDD-CUP-99 dataset but immune to above-mentioned shortcomings.
Both KDD-CUP-99 and NSL-KDD dataset contains simulated attacks which falls one of the following four types.

1) **Denial of Service Attack (DoS):** This type of attack is initiated by making computing resources busy which denies users from accessing a service.
2) **User to Root Attack (U2R):** In this class of attack, the attacker gets the access of a legitimate account by sniffing password or social engineering. The attacker then uses the root access to exploit the vulnerability.
3) **Remote to Local Attack (R2L):** This happens when the attacker uses some vulnerability to gain the local access of that machine.
4) **Probing Attack:** The attacker tries to gain information about the victim's computer network avoiding its security controls.

The Original KDD dataset includes redundant samples in the training set which makes the classifier biased towards more frequent records. This bias prevents the algorithm from learning more harmful but infrequent records like User to Root Attack (U2R). The NSL-KDD dataset improves on the KDD-CUP-99 dataset by discarding redundant records and duplicate data in the training and testing sets accordingly. The new NSL-KDD data set is still vulnerable to some of the problems discussed by McHugh [11]. However, it is still used as benchmarking data for various intrusion detection methods.

The study by Jiyeon.et.al. [6] proposes a Convolutional Neural Network (CNN) based system for detecting DoS attacks. The performance of the proposed CNN model was evaluated against a Recurrent Neural Network (RNN). Here, they have generated two different types of image datasets from the KDD-CUP-99 dataset. In the dataset, the grayscale set of images has one channel whereas the RGB images have 3 colour channels. For evaluation, 18 scenarios have been created considering various hyperparameters including image size, kernel size and the number of convolutional layers. In this study, both binary and multi-class classification is implemented however, for multi-class classification, they have considered only 3 types of DoS attacks. One possible reason for this kind of choice can be the small sample size of other types of attacks since the performance of CNN based algorithms increases with the increase of sample size. RGB images outperformed grayscale images in terms of accuracy in the case of both binary and multi-class classifications. It is also shown that the performance of the model increases when more than one convolutional layer is used for multi-class classification. Hyperparameter tuning was performed to find out the best possible model for the classification problem. Both CNN and RNN showed an accuracy of 99% for binary classification. Whereas, for multi-class classification, CNN achieved an overall accuracy score of 99% while RNN showed 100%, 80% and 85% accuracy for Smurf detection, Neptune and Benign DoS attacks respectively.

Derhab.et.al [12] proposes an Anomaly-based detection mechanism for DoS attacks that have been classified into statistical and Machine Learning-based methods. In the Statistical method, the normal traffic profile is calculated using the mean and standard deviation of normal traffic. The benefit of the Statistical approach is lower detection time with a low computational cost. In Machine Learning methods, classification and ensemble learning algorithms have been used to learn the traffic.

The study by Amma.et.al [13] is a Statistical method that uses the Class center-based triangle area vector (CCTAV) to calculate the class center of the dataset along with feature extraction. For each sample data, the triangle area vector (TAV)

is generated. In this study, two different profiles are created one for the normal traffic and the other for the abnormal (attack) traffic. The normal traffic profile was created using the mean and standard deviation of Mahalanobis distance (MahD) by taking all normal traffic into account. By using this Mahalanobis distance the traffic is predicted as normal or attack type. In addition, the study shows that CCTAV has a computational complexity of $O((e^2)^n)$ which is less as compared to the existing Statistical methods.

Nerijus.et.al [14] analyzed the effect of data pre-processing on detection accuracy with the NSL-KDD dataset. The study shows that data preprocessing plays a vital role in determining the precision and accuracy of a Machine Learning model. In addition, Suleman.et.al [15] found out some of the important features of the DoS attack by calculating the entropy and granulation in the dataset. In this approach, the weight of each feature was calculated using the entropy calculation and then the best features important for classifying the DoS attack types were presented. This method helps in reducing the dimension of the dataset.

Su.et.al. [8] uses BLSTM (Bidirectional Long Short-term memory) and attention mechanism for intrusion detection. The key features for network traffic classification have been obtained by the attention mechanism which screens the network flow vector generated by the BLSTM model. They also employed multiple layers of the convolutional layer to extract the local features of traffic data for classification. The algorithm has five major components i.e input layer, multiple Layers, BSLTM layer, attention layer and output layer. This deep Neural Network doesn't require any feature extraction as it automatically extracts the key features of the dataset by using different layers.

Subha.et.al [16] analyzed different Neural Network algorithms such as ART, ARTl, FUZZY ART, IVEBF and EBCS and pointed out their performance in improving the lifetime of a Wireless Sensor Networks (WSNs). The performance of a sensor network largely depends on the computational efficiency of the sensor node. Nowadays, a large number of smart sensor nodes are using Artificial Intelligence. So, it is of paramount importance that intelligent tools should be used in an energy-efficient way in the wireless sensor network. In addition, different types of Machine Learning techniques are introduced to reduce energy consumption along with a reduction in communication cost for WSNs.

The study by Tao.et.al [17] uses a hybrid approach for detecting DoS Cyberthreats. The hybrid approach consists of ensemble learning in which the first step is to find clusters of different labels in the dataset using Spectral clustering. In the second step, it uses a Deep Neural Network(DNN) to do multi-class classification. It then predicts whether traffic is normal or whether an attack is being attempted by specifying which type of attack is being conducted on the network with an accuracy score. However, the hybrid approach mentioned by [17] uses unsupervised learning in the first step to cluster the labels which seems unreasonable as the DoS Cyber attacks are a Supervised learning problem. As given in [18] the Computational Complexity of Spectral Clustering algorithm is $\mathcal{O}(n^3)$ which is very high. The clustering technique used increases the computational complexity with an increase in power consumption by the network. This, however, might not find large scale applications for detecting DoS Intrusions in WSNs as we know that power consumption in WSNs is a limited resource.

Wireless sensor network has constraints in terms of processing power and energy consumption. A large network such as BLSTM and CNN is more resource-intensive and is not an ideal choice for intrusion detection such as DoS attacks in a Wireless Sensor Network. Our work aims to deploy efficient and low power consuming Machine Learning algorithms while improving intrusion detection performance in detecting different DoS Cyber attacks.

III. Proposed System Design

In order to find an efficient AI and ML Network to detect DoS attacks for Wireless Sensor Networks, it is important to consider parameters such as memory requirement, power consumption by the Network, Training Computation Complexity and Prediction Computation Complexity.

For the DoS attack detection, we have shortlisted 3 Machine Learning Algorithms based on Computation Complexity for training and prediction and 2 Artificial Intelligence models based on the overall accuracy for a different type of DoS attack.

A. Computation Complexity

Computational complexity or simply complexity of an algorithm is defined as the number of resources required to run a particular algorithm. Computational Complexity gives importance to time and memory requirements for an algorithm. As the amount of resources required to run an algorithm generally varies with the size of the input, the complexity is typically expressed as a function $n \rightarrow f(n)$, where n is the size of the input and $f(n)$ is either the worst-case complexity (the maximum of the number of resources that are needed overall inputs of size n) or the average-case complexity (the average of the number of resources overall inputs of size n). Time complexity is generally expressed as the number of required elementary operations on an input of size n, where elementary operations are assumed to take a constant amount of time on a given computer and change only by a constant factor when running on a different computer. Space complexity is generally expressed as the amount of memory required by an algorithm on an input of size n.

After an analysis, we have come up with 3 Machine Learning algorithms that will be efficient for DoS attack detection. The algorithms are given below in Table I as per their Training Complexity and Prediction Complexity.

In Table I, n is the number of training samples in the training dataset and f is the number of features in the dataset whereas n_{trees} is the depth of the tree or the number of trees depending on the implementation of the Random Forest algorithm.

Table I
COMPUTATIONAL COMPLEXITY OF MACHINE LEARNING ALGORITHMS

Algorithms	Training Complexity	Prediction Complexity
Random Forest	$\mathcal{O}(n^2\ f\ n_{trees})$	$\mathcal{O}(f\ n_{trees})$
Decision Tree	$\mathcal{O}(n^2\ f)$	$\mathcal{O}(f)$
Naive Bayes	$\mathcal{O}(n\ f)$	$\mathcal{O}(f)$

From Table I, it is clear that the Naive Bayes algorithm will be more efficient than that of the Decision Tree algorithm when we consider both Training and Prediction Complexity. However, the two algorithms have the same Prediction Computation Complexity. Random Forest algorithm has higher Training and Prediction Complexity when compared to both Naive Bayes and Decision Tree by a factor of n_{trees}. Therefore, the decision to efficiently design the network for DoS Cyberthreat detection using Machine Learning algorithms depends entirely on the prediction accuracy of the Naive Bayes and Decision Tree algorithm. The higher the prediction accuracy for the dataset that particular algorithm will be chosen for the network design.

Table II
COMPUTATIONAL COMPLEXITY OF NEURAL NETWORKS

Algorithms	Prediction Complexity
Neural Network	$\mathcal{O}(fn_{l_1} + n_{l_1}n_{l_2} + ...)$

In Table II, f is the number of features in the dataset whereas, n_{l_i} is the number of neurons at layer i in a neural network. For DoS attack detection Multilayer Perceptrons (MLP) and LSTM (Long Short Term Memory) Recurrent Neural Network are considered due to their large applicability in different domains.

As of now in choosing the most efficient model for DoS Cyber threat detection is incomplete as the analysis on Power Consumption is not completed for all the models. Once, when we calculate the power consumption on all the models for the given dataset then an overall comparison for selecting the most efficient algorithm for different DoS Cyberthreats Detection.

IV. METRICS

There are three metrics used for evaluating the models they are as given below:

A. Accuracy Score

Accuracy is defined as the closeness of measurement to a particular value. Accuracy is used when the different classes in the dataset are well balanced. Accuracy Score is given by the formula below:

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

Where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

B. F1-Score

F1-Score, also known as F-Score or F-Measure is a measure of a Test's accuracy. F1-score takes the harmonic mean of the precision and recall. It computes the result by the given formula below:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2}$$

F1-Score is mainly used when the classes in the dataset are unbalanced. The maximum value attained by F1-score is 1 and it is also know as the Dice similarity coefficient.

C. RoC Curve

The Receiver Operating Characteristic curve (RoC) is a useful tool to find models for classification based on their performance by considering the False Positive Rate (FPR) and True Positive Rate(TPR). These values are computed by shifting the decision threshold of the classifier. On the RoC curve the TPR feature on the Y-axis whereas, FPR feature on the X-axis. Therefore 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.

V. EXPERIMENTAL RESULTS

In this section, we discuss the results obtained from each algorithm. The main metrics used for evaluating the algorithms are F1-score, Accuracy Score and RoC curve and analysis is laid out for each of the algorithms as per the dataset. The tables III and IV below shows the average accuracy for each of the dataset. Here for the entire dataset, the target accuracy is computed for Multi-class classification.

The Figure 1 below shows the number of samples in KDD-CUP-99 [2] dataset. As from the Figure 1, it is clear that the dataset is not balanced as few attack types have less than 100 samples whereas others have more than 1000 samples. This unbalanced dataset can affect the performance of the algorithms.

In addition, the number of samples for different DoS attack types for NSL-KDD [3] dataset is shown in Figure 2. Here similar to the KDD-CUP-99 dataset the number of samples for each of attack types is not the same and so this dataset like KDD-CUP-99 dataset is unbalanced.

Table III
ACCURACY ANALYSIS ON MODELS FOR NSL-KDD DATASET

Algorithms	Accuracy Score	Design Parameters
Random Forest	98.05 % (Test Set)	$n_{tree} = 1$
Decision Tree	98.53 % (Test Set)	$max.\ dept = 15$
Naive Bayes	61.28 % (Test Set)	No Parameters
MLP	95.83 % (Test Set)	5 hidden layers
LSTM	92.32 % (Test Set)	80x1

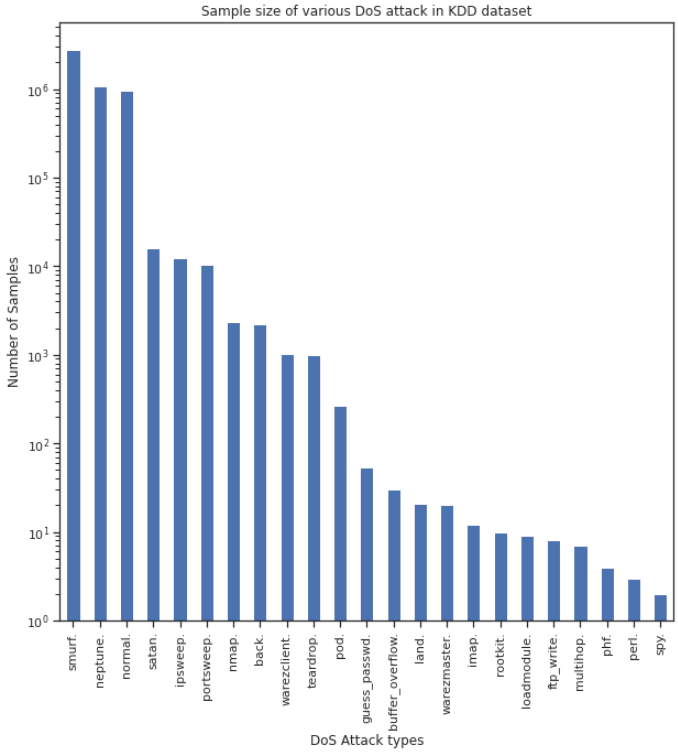

Figure 1. Plot showing the Number of Samples for Each Attack Type for KDD-CUP-99 dataset.

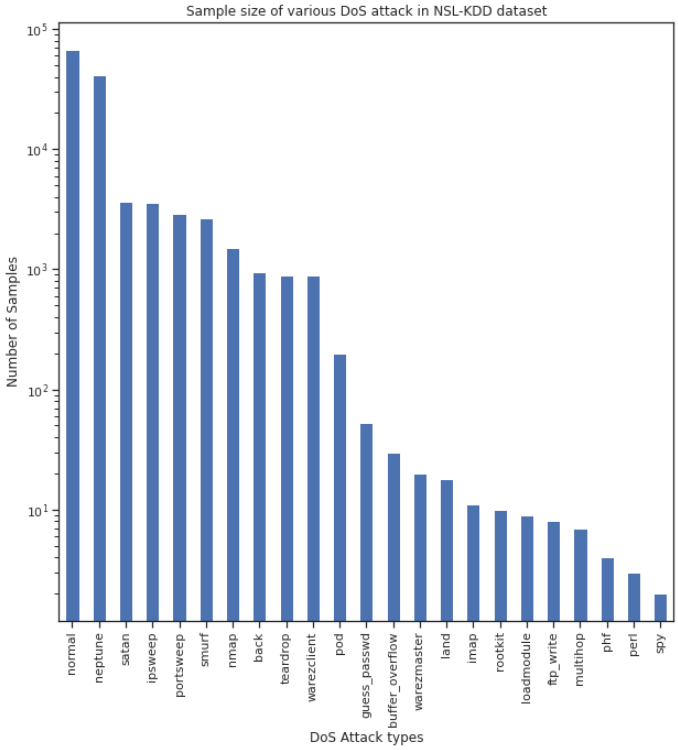

Figure 2. Plot showing the Number of Samples for Each Attack Type for NSL-KDD dataset.

Table IV
ACCURACY ANALYSIS ON MODELS FOR KDD-CUP-99 DATASET

Algorithms	Accuracy Score	Design Parameters
Random Forest	99.90 % (Test Set)	$n_{tree} = 5$
Decision Tree	99.90 % (Test Set)	max. Dept = 15
Naive Bayes	92.71 % (Test Set)	No Parameters
MLP	99.61 % (Test Set)	5 hidden layers
LSTM	99.32 % (Test Set)	80x1

A. Naive Bayes

Naive Bayes [19] is a supervised Machine Learning algorithm for Binary and multi-class classification that uses Bayes theorem. It is considered a simple or Naive algorithm for information retrieval. Despite its oversimplified assumption, the Naive Bayes algorithm works effectively for many real-world simulations. One of the main advantages of the Naive Bayes algorithm is that training time is quite low for estimating the necessary parameters. As evident from I The training complexity and prediction complexity is lowest among all the other mentioned algorithms in the table. This is the main reason Naive Bayes has been included in this study. The Performance expectation from Naive Bayes is descent and is expected to be lower than that of other algorithms.

1) KDD-CUP-99 Dataset: The accuracy score for this dataset represents the overall accuracy of all the labels in the test data. The Naive Bayes Classifier accuracy score for this dataset was the lowest among all the three classifiers at 92.71 %. In addition, there are 6 different class labels for which this classifier's accuracy score is 0%. Therefore, the Naive Bayes classifier's performance is the lowest among all the mentioned Machine Learning Models.

The results in table V show the F1-score for all the different types of DoS attacks using the Naive Bayes classifier. It is evident that Naive Bayes [19] is unable to generalize the different types of attack with a higher F1-score, for example, FtpWrite, Imap, MultiHop, Rootkit and Spy attack types have zero F1-score whereas BufferOverflow, Ipsweep, LoadModule, Nmap, Perl, Portsweep, Satan, WarezClient and Warezmaster have an F1-score below 0.50. There are 7 different attack types for which Naive Bayes gives better F1-score these include Back, Guesspasswd, Neptune, Normal, Pod, Smurf and Teardrop DoS attacks.

2) NSL-KDD Dataset: The accuracy score is 62.28% on Test set which quite lower as compared to other Machine Learning algorithms. The performance is lower as compared to KDD-CUP 99 dataset by almost 30%. This shows that the generalization of Naive Bayes on this dataset is not competitive as compared to other algorithms and gives us a performance that is lowest among all the models on this dataset.

The F1-score for the NSL-KDD dataset is similar in performance as in KDD-CUP 99 dataset. As shown in table VI there are 6 different DoS attacks for which the F1-score id

0.00 these are FtpWrite, Ipsweep, LoadModule, Perl, Rootkit and Spy which is 5 in the case of KDD-CUP 99 dataset. The attacks that have F1-score less than 0.50 are BufferOverflow, Land, MultiHop, Nmap, Portsweep, Satan, WarezClient and Warezmaster which is similar as in the case above. There are 9 different attack types for which Naive Bayes gives an F1-score greater than 0.50 these include Back, GuessPasswd, Imap, Neptune, Normal, Phf, Pod, Smurf and Teardrop DoS attacks.

B. Random Forest

Random Forest [20] is also known by the name of Random Decision Forests which is an Ensemble learning method for Classification and Regression. Random Forest Classifiers often can over-fit the dataset during the training process. It works by splitting each node during the construction of a tree. The best Split can be found by considering the input features or from a random subset of the maximum number of Features. The randomness introduced in the forest gives out decoupled prediction errors. This error is however cancelled out when we take the average of the prediction error. Random Forest Classifier achieves reduced variance with a slight increase in bias as it combines diverse trees. Due to the Variance reduction in Random Forest, the models give out better performance. As from I it is clear that using Random Forest is more Computationally expensive than the other two mentioned algorithms.

1) KDD-CUP-99 Dataset: KDD CUP dataset [2] is different from that of the NSL-KDD dataset and in this the major problem that we have encountered is that the dataset doesn't contain enough samples for all the DoS attack types. For example, for rootkit Cyberthreat the dataset contains fewer than 500 samples which are train a Neural Network or a Machine Learning model. The preprocessing of the dataset is the same for the KDD CUP dataset [2] as that of the NSL-KDD dataset. The number of features is reduced from 42 to 10 using the Chi2-Squared Test feature selection method.

In the Random Forest algorithm, the overall validation accuracy on the entire dataset is 99.90 % when the depth of the tree is 5. Besides, for multiclass classification, the accuracy scores are satisfactory for each of the attack types. However, there are 6 different classes of attacks for which the Random Forest classifier had an accuracy score of 0 %. One of the main reasons for this is the number of samples available during the testing phase and in each of the cases, the samples were less than 5. Besides, the Prediction Complexity for Random forest is higher than that of both Naive Bayes and Decision Tree Classifiers.

The F1-score shown in table V there are 6 different attack types that have an F1-score of 0.00 they are FtpWrite, Imap, LoadModule, Multihop, Rootkit and Spy. However, there is only Land attack type is having an F1-score less than 0.50 which is better than that of the Naive Bayes algorithm. Therefore, in total there are 7 DoS attack types for which Random Forest doesn't have a good F1-score.

2) NSL-KDD Dataset: The accuracy score for the Random forest algorithm is 98.05% on the Test set for all the target labels of this dataset when the parameter $n_{tree} = 1$. This accuracy is score is marginally higher than that of the Naive Bayes algorithm. However, this accuracy score is slightly (1.9%) lower than that of the KDD-CUP-99 dataset.

The F1-score for this dataset is shown in table VI and the attacks for which the F1-score is zero are FtpWrite, LoadModule, Perl, Rootkit and Spy. In addition similar to the NSL-KDD dataset only Land attack type has an F1-score of less than 0.50. So, there are only 6 different attacks that have a low F1-score which is better than Naive Bayes.

C. Decision Tree

Decision Trees [21] are non-parametric supervised learning algorithms that are mainly used for Classification and Regression. It predicts the targets by learning decision rules inferred from the input features without using any parameters. Decision trees require little data preparation however it can create overly complex trees which in some cases leads to overfitting and then the generalization of the dataset decreases. As from I the prediction complexity is similar to that of Naive Bayes algorithm [19] however its Training Complexity is higher than Naive Bayes.

1) KDD-CUP-99 Dataset: In the Case of Decision Tree Classifier, the accuracy score for the validation test is 99.90 % which is identical to that of the Random Forest Classifier. However, in this case, also there were 6 different classes for which the Classifier didn't give any accuracy score. One key point to note is that the Prediction Complexity of Decision Tree is less than that of Random forest Classifier by a factor of n_{trees} which makes Decision Tree Classifier efficient than Random Forest Classifier.

The F1-score for KDD-CUP-99 Dataset using the Decision Tree algorithm is shown in table V. The attack types FtpWrite, Imap, LoadModule, MultiHop, Perl, Phf, Spy and warezmaster all have an F1-score of 0.00. In addition, only the Land attack type has an F1-score of less than 0.50 rest all other attacks have an F1-score of more than 0.50 and this number for this dataset using the Decision Tree algorithm is 15 which is lower than Random Forest for the same dataset.

2) NSL-KDD Dataset: The Accuracy Score for NSL-KDD dataset [2] for Decision Tree is 98.53% which is highest among all the models presented in Table III. The Decision tree's performance is dependent on the dept of the Tree and for this dataset to achieve the best performance for all the attack types the maximum depth of the tree is kept to be 15 which is shown in Table III. The maximum depth of the tree does not affect the computational complexity of the algorithm which is an advantage over other algorithms.

F1-Score Analysis: The F1-score for this dataset is shown in table VI. It is evident that there are 4 different attack types for which F1-score is 0.00, they are FtpWrite, Perl, Rootkit and Spy. This number is the lowest among all the mentioned Machine Learning Algorithms. In addition, the attacks Land and LoadModule have an F1-score of less than 0.50. which is similar in performance to that of the Random Forest algorithm. So, the Decision Tree has only 6 different attack types for

which the F1-score is below 0.50 rest all have a score above 0.50 which 17 in this case.

D. MultiLayer Perceptron

Multilayer Perceptron [22] (MLP) is a supervised learning algorithm that uses Backpropagation learning method [22] by learning a function $f(\cdot) : R^f \rightarrow R^o$ 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 such as layers, hidden neurons and number of iterations for training. It has the capability for Multi-class classification by using the Softmax activation function on its output layer.

1) KDD-CUP-99 Dataset: There are only two Deep Neural Network (DNN) models chosen for this problem as there other seems to be more Computationally expensive. In the first case, an MLP with 3 hidden layers (100*50*25) was chosen for the Multiclass Classification problem. This model gave an accuracy score of 99.61 % for the overall dataset. However, there were 11 different Classes for which the model was not able to predict sufficient accuracy score. The main problem is again the low number of samples in the dataset for each of these classes.

The F1 score for various types of DoS attack is given in Table V for MLP classifier. For some of the DoS attack types, the F1 score is very low which means poor precision and recall for that particular class. For MLP classifier, Neptune, Pod, Smurf and WarezClient have the highest F1 score while BufferOverflow, FtpWrite, GuessPasswd are among the ones who have an F1 score of zero. The F1 score for various DoS attack types tends to be either near 1 or zero with the only exception of Portsweep.

2) NSL-KDD Dataset: The MLP used for NSL-KDD Dataset has 3 hidden layers (100*50*25) with softmax as activation function at the output layer. The classifier gave an overall accuracy of 96 %. Though the overall accuracy is satisfactory, the classifier was unable to correctly classify 8 types of DoS attacks out of 23 types. Since the NSL-KDD Dataset has less number of samples compared to KDD-CUP-99 Dataset, the performance of MLP classifier on NSL-KDD Dataset is poor compared to KDD-CUP-99 Dataset.

The F1 score using an MLP classifier for various types of DoS attacks has been shown in Table VI. The F1 for some of the DoS attacks is above 80 % which includes Back, GuessPasswd, Ipsweep, Neptune, Pod, Satan, Smurf, Teardrop, WarezClient and Warezmaster. But the classifier was unable to correctly identify some of DoS attacks which resulted in a lower F1 score for BufferOverflow, FtpWrite, Imap, Land, LandModule, Multihop etc.

E. Long Short Term Memory

Long Short Term Memory (LSTM) [23] 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 encounter in RNN architecture. The design of the LSTM network used in this scenario for both the dataset has 80 LSTM units and 1 dense layer with 23 units in it for classifying the different DoS attack types.

1) KDD-CUP-99 Dataset: LSTM has a high accuracy score of 99.32% on Test set for the targets which contain all the different attack labels. This score is slightly lower than MLP but better than Naive Bayes which has the lowest accuracy score of 92.71% among all the algorithms.

The F1-score for this dataset using LSTM is shown in table V. The attack types which have an F1-score of 0.00 are BufferOverflow, FtpWrite, Imap, Land, LoadModule, Multi-Hop, Nmap, Perl, Phf, Rootkit and Spy which higher than MLP. In addition, the attacks which have an F1-score less than 0.50 are GuessPasswd, Portsweep and WarezClient. Apart from these attacks, all the other attacks have an F1-score greater than 0.50. Besides, only a few of the attack types such as Neptune, Satan and Smurf have a perfect F1-score of 1.00 given out by the model.

2) NSL-KDD Dataset: The Accuracy Score of NSL-KDD for LSTM is 92.32% which shown in table III. This score is lower than that of the KDD-CUP-99 dataset by 7% for the same number of LSTM units. The LSTM classifier has 80 hidden units with a dense layer having 500 neurons. The Dropout layer has also been added which allows the neuron to work independently instead of being dependent on a small number of neurons.

The F1 score using an LSTM classifier for various types of DoS attacks has been shown in Table VI. Some of the attack types have 0 F1 scores which are FtpWrite, Land, LandModule, MultiHop, Perl, Phf and Spy. This result is similar to KDD-CUP-99 Dataset with LSTM classifier with some improvement for some attack types in terms of F1 score. For the LSTM classifier, Back, GuessPasswd, Imap, Ipsweep, Neptune, Pod attack types have the highest accuracy score.

VI. Discussion

In this section we discuss the results obtained on the two different datasets [2] mentioned in the previous section. The Naive Bayes Classifier [19] has an accuracy score of 61.28% on the Test set for the NSL-KDD dataset [2] which is the lowest score among all the model mentioned in table III. In addition, the accuracy score on the KDD-CUP-99 dataset is improved by 30% to 92.71% on the test set. However, this score is the lowest accuracy score among all the models for the KDD-CUP-99 dataset. One major reason for the improvement in the performance of the algorithm is mainly due to the increase in the number of samples in the KDD-CUP-99 dataset which is far greater than the NSL-KDD dataset. The F1-score for Naive Bayes on the two datasets is similar and in both the datasets there are attacks for which the F1-score is 0.00

Table V
ACCURACY SCORE FOR DIFFERENT ALGORITHM ON KDD-CUP-99 DATASET

CyberAttacks	Naive Bayes (F1 Score)	Random Forest (F1 Score)	Decision Tree (F1 Score)	MLP (F1 Score)	LSTM (F1 Score)
Back	0.99	1.00	1.00	0.93	0.90
BufferOverflow	0.01	0.67	0.67	0.00	0.00
FtpWrite	0.00	0.00	0.00	0.00	0.00
GuessPasswd	0.86	0.95	1.00	0.00	0.20
Imap	0.00	0.00	0.00	0.00	0.00
Ipsweep	0.02	0.97	0.96	0.89	0.88
Land	0.01	0.33	0.33	0.00	0.00
LoadModule	0.07	0.00	0.00	0.00	0.00
MultiHop	0.00	0.00	0.00	0.00	0.00
Neptune	0.98	1.00	1.00	1.00	1.00
Nmap	0.02	0.50	0.87	0.00	0.00
Normal	0.81	1.00	1.00	0.99	0.99
Perl	0.13	1.00	0.00	0.00	0.00
Phf	1.00	1.00	0.00	0.00	0.00
Pod	0.76	0.99	0.98	1.00	0.96
Portsweep	0.08	0.88	0.85	0.56	0.42
Rootkit	0.00	0.00	1.00	0.00	0.00
Satan	0.01	0.96	0.99	0.90	1.00
Smurf	1.00	1.00	1.00	1.00	1.00
Spy	0.00	0.00	0.00	0.00	0.00
Teardrop	1.00	1.00	1.00	0.94	0.99
WarezClient	0.08	0.98	0.99	1.00	0.49
Warezmaster	0.02	0.00	0.00	0.00	0.50

Table VI
ACCURACY SCORE FOR DIFFERENT ALGORITHM ON NSL-KDD DATASET

CyberAttack Types	Naive Bayes (F1 Score)	Random Forest (F1 Score)	Decision Tree (F1 Score)	MLP (F1 Score)	LSTM (F1 Score)
Back	0.98	1.00	0.99	0.95	0.95
BufferOverflow	0.25	0.77	0.67	0.00	0.57
FtpWrite	0.00	0.00	0.00	0.00	0.00
GuessPasswd	0.77	0.92	0.86	0.81	0.93
Imap	1.00	1.00	1.00	0.00	1.00
Ipsweep	0.00	0.91	0.92	0.85	0.81
Land	0.01	0.40	0.40	0.00	0.00
LoadModule	0.00	0.00	0.40	0.00	0.00
MultiHop	0.08	1.00	0.67	0.00	0.00
Neptune	0.91	0.99	0.99	0.98	0.97
Nmap	0.09	0.62	0.66	0.17	0.17
Normal	0.60	1.00	1.00	0.98	0.97
Perl	0.00	0.00	0.00	0.00	0.00
Phf	1.00	1.00	1.00	0.00	0.00
Pod	0.95	0.99	0.99	0.99	0.99
Portsweep	0.39	0.87	0.89	0.73	0.70
Rootkit	0.00	0.00	0.00	0.00	0.00
Satan	0.09	0.92	0.93	0.81	0.74
Smurf	0.99	1.00	1.00	1.00	0.99
Spy	0.00	0.00	0.00	0.00	0.00
Teardrop	0.99	1.00	1.00	1.00	1.00
WarezClient	0.29	0.99	0.97	0.81	0.73
Warezmaster	0.44	0.86	0.86	0.86	0.86

these are those attacks for which the number of samples in the dataset is very low. In some cases, there were no samples of the attack types in the test set while testing the Classifier on the dataset. However, the generalization of this classifier was less compared to other models in both the datasets. The RoC curve for the Naive Bayes algorithm on KDD-CUP-99 and NSL-KDD dataset is shown in figure 3 and 4 respectively. In figure 3 there are some attack types for which the area under the curve is below the imaginary line which shows that the classifier doesn't perform well for those attack types. Besides, most of the attack types reside on the upper left corner of the graph which tells us that on this dataset the performance is acceptable. In contrast, Figure 4 performance is lower as the area under curve decrease and in this case, the algorithm behaves far from ideal. However, One advantage of the Naive Bayes algorithm is its low training and prediction Complexity that makes it an efficient algorithm but the generalization, in this case, is poor. Therefore, the Naive Bayes algorithm is not an ideal model for the multiclass classification of KDD-CUP-99 and NSL-KDD datasets [2].

The second model is the Random Forest algorithm [20] and the accuracy score for the KDD-CUP-99 and NSL-KDD dataset is 99.90% and 98.05% respectively when the design parameter of $n_{trees} = 5$ is chosen. When the design parameter is altered the accuracy score of the multiclass classification for the algorithm decreases which in turn affects the detection of different attack types using this algorithm. In addition, the F1-score for the datasets shows us that for each of the attack types the algorithm performs better than Naive Bayes [19] algorithm. The F1-score from the KDD-CUP-99 dataset it is clear that the overall performance in predicting the attack types has improved significantly when compared to the Naive Bayes algorithm. For example, the F1-score in detecting the BufferOverflow has improved from 0.01 by Naive Bayes to 0.67 in Random Forest which effectively tells us that the attack is BufferOverflow attack and this can be said with more precision in the case of Random Forest algorithm. The RoC curve obtained in figure 5 has a better area under the curve than that of Naive Bayes however for some classes the area under the curve is lower. In comparison, Figure 6 obtained on the NSL-KDD dataset has more number of classes for which the area under the curve is lower which shows that the performance of Random Forest decreases in the case of NSL-KDD dataset which is still better than that of Naive Bayes algorithm. Besides, the Training complexity of the Random Forest algorithm as mentioned in Table I is highest among all the mentioned Machine Learning models.

Decision Tree [21] has an identical accuracy score to that of Random Forest [20] in the case of the KDD-CUP-99 dataset which is highest among all the algorithm. The performance in the case of NSL-KDD is slightly better which makes it the ideal algorithm for multi-class classification for the NSL-KDD dataset. One important point to consider is that the maximum depth of the tree is 15 which is an important parameter in order to achieve maximum performance from this algorithm for this particular dataset. Decision Tree has the best F1-scores for the different attack types among the algorithms in the NSL-KDD dataset. As shown in Table VI It has only 4 different attack types for which the F1-score is 0.00 and in two of the cases there were no test samples in the test set which is the lowest among all the presented algorithms. For the KDD-CUP dataset, the performance of the Decision Tree is almost identical to the Random forest and in some cases, it is better. For example, in Nmap the F1-score increases from 0.50 in Random forest to 0.87 in Decision Trees. One of the main reasons for low F1-score in some cases is because both the dataset is unbalanced and in some cases, lack of adequate samples makes it difficult for the classifier to classify the different attack types. The RoC curve for Decision Tree is shown in figure 7 and 8 for KDD-CUP-99 and NSL-KDD dataset respectively. The area under the curve for the KDD-CUP-99 dataset is much higher than any of the other classifiers mentioned. In comparison, the area under the curve for the NSL-KDD dataset is lower and the generalization is better than the Random Forest algorithm. Besides, one major advantage of the Decision Tree algorithm is that the prediction Complexity is the same as that of the Naive Bayes algorithm which makes it an efficient high-performance algorithm for multiclass classification for DoS attacks type detection in Sensor Networks.

In the first of Artificial Intelligence (AI) Models, MLP is used and the accuracy score for it on the KDD-CUP-99 dataset is 99.61% on the test set. Besides, the accuracy score on the NSL-KDD dataset is 95.83% which is lower than that of the KDD-CUP-99 dataset. MLP has the highest accuracy score between the two AI models and its performance is similar to the Machine Learning algorithms. In addition, the F1-score obtained for different DoS attacks KDD-CUP-99 dataset is poor. It has an F1-score of 0.00 for 13 different attack types which the worst performance amongst all the models presented. However, the performance on NSL-KDD is better and has a lower number of attack types which has an F1-score of 0.00. The RoC curve is shown in Figure 9 and 10 for KDD-CUP-99 dataset and NSL-KDD dataset. The area under the curve in Figure 9 is lower for some class labels for eg. WarezMaster attack the area is only 0.78. However, in comparison, this performance gets better on the NSL-KDD dataset in which is clear that the area under the curve is higher than that from Figure 9 which aligns with the F1-score performance and Accuracy score for each of the dataset. Besides, MLP has a large Memory requirement for storing trained parameters to do an effective prediction. The prediction complexity of MLP is higher than that of Machine Learning models which is a drawback in the case where efficiency is of paramount importance.

The second AI model used for Multi-class classification is LSTM and the accuracy score on the KDD-CUP-99 dataset is 99.32% which is slightly lower than MLP for the same case. The score decreases by almost 7% for the NSL-KDD datasets. The F1-score presented in table V is for the KDD-CUP-99 dataset and this score is similar to MLP with a low number of attack types which has an F1-score of 0.00. The

F1-score for NSL-KDD is shown in table VI and the score is better than that in table V. For eg. the attack type Warezmaster has an F1-score of 0.50 in the KDD-CUP-99 dataset and an F1-score of 0.86 in the case of the NSL-KDD dataset which is far better. In comparison to MLP, the score is similar in some cases whereas lower in some cases. The RoC curve for LSTM is shown in Figure 11 and 12 for KDD-CUP-99 and NSL-KDD dataset. The Performance-based on RoC curve for KDD-CUP-99 shows that LSTM performs better than MLP when compared to all the classes as the area under the curve is higher for LSTM than that in Figure 9 which shows RoC curve for MLP on KDD-CUP-99 dataset. However, the case is not similar for the NSL-KDD dataset when the RoC curve is compared between LSTM and MLP. The area under the curve if lower for the majority of the attack types in Figure 12 than that of Figure 10. In addition, the generalization for the two datasets using LSTM is not as expected given the Computation complexity for the algorithm. Besides, this performance of LSTM is lower for Multiclass classification when compared to more efficient Machine Learning models presented in Table I.

VII. Conclusion

A study of 5 different models was presented based on the Training and Prediction complexity of the algorithms and the analysis on each one for the KDD-CUP-99 and NSL-KDD dataset was performed in the discussion section. The Multi-class classification for each of the models was performed and the F1-score for detecting each of the attack types for the two datasets was presented in table V and VI. The accuracy score presented in table III and IV for NSL-KDD and KDD-CUP-99 dataset respectively is the overall accuracy score for the complete target set which consists of all the different attack types. This detailed analysis of multi-class classification was lacking in all the papers that we reviewed for DoS Intrusion detection in Wireless Sensor Networks in section II.

For efficient model selection, 5 different AI and ML models were presented. Among the AI and ML models, it is evident that the ML Models outperform the AI models in terms of Accuracy score, F1-score and RoC curve analysis for Multiclass classification for both the datasets. The results discussed in the above section shows that among the 3 presented Machine Learning Models Naive Bayes and Decision Tree has the lowest Prediction Complexity and then we have Random Forest which has higher Prediction Complexity than the two algorithms. The metrics for Naive Bayes on the two different dataset [2] shows that it is not an ideal algorithm for Multiclass classification for a different kind of DoS Intrusion detection. The performance between Random Forest [20] and Decision Tree [21] is similar when compared with the three metrics given above and in some cases of attack types, Decision Tree tends to outperform Random Forest by some good margin. In addition, Decision Tree has lower Prediction Complexity when compared with that of the Random Forest algorithm which makes it an ideal and efficient classifier as lower complexity means low power and memory requirement from Wireless Sensor Network (WSNs) for performing Multi-class classification and detecting all different kinds of DoS intrusion detection.

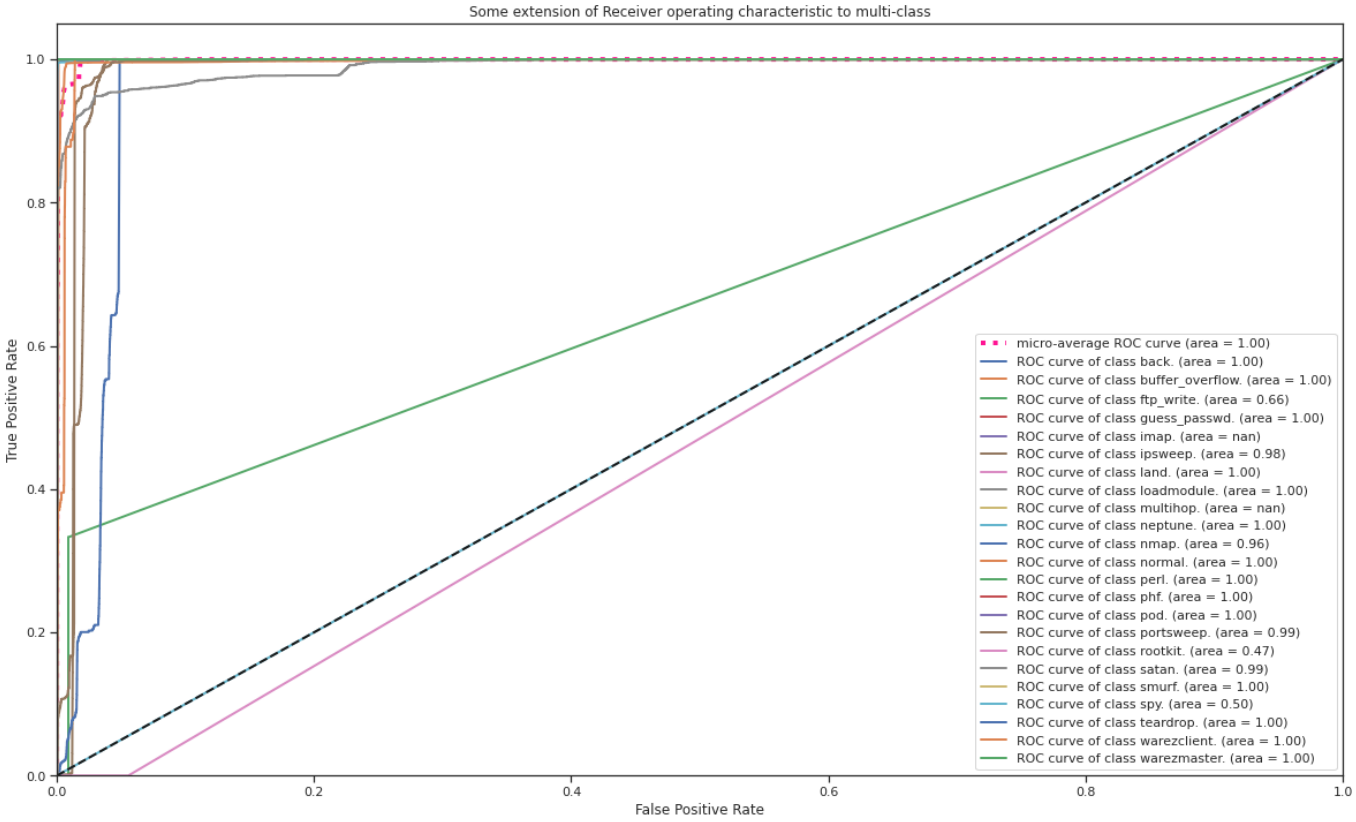

Figure 3. RoC curve of Naive Bayes on KDD-CUP-99 Dataset

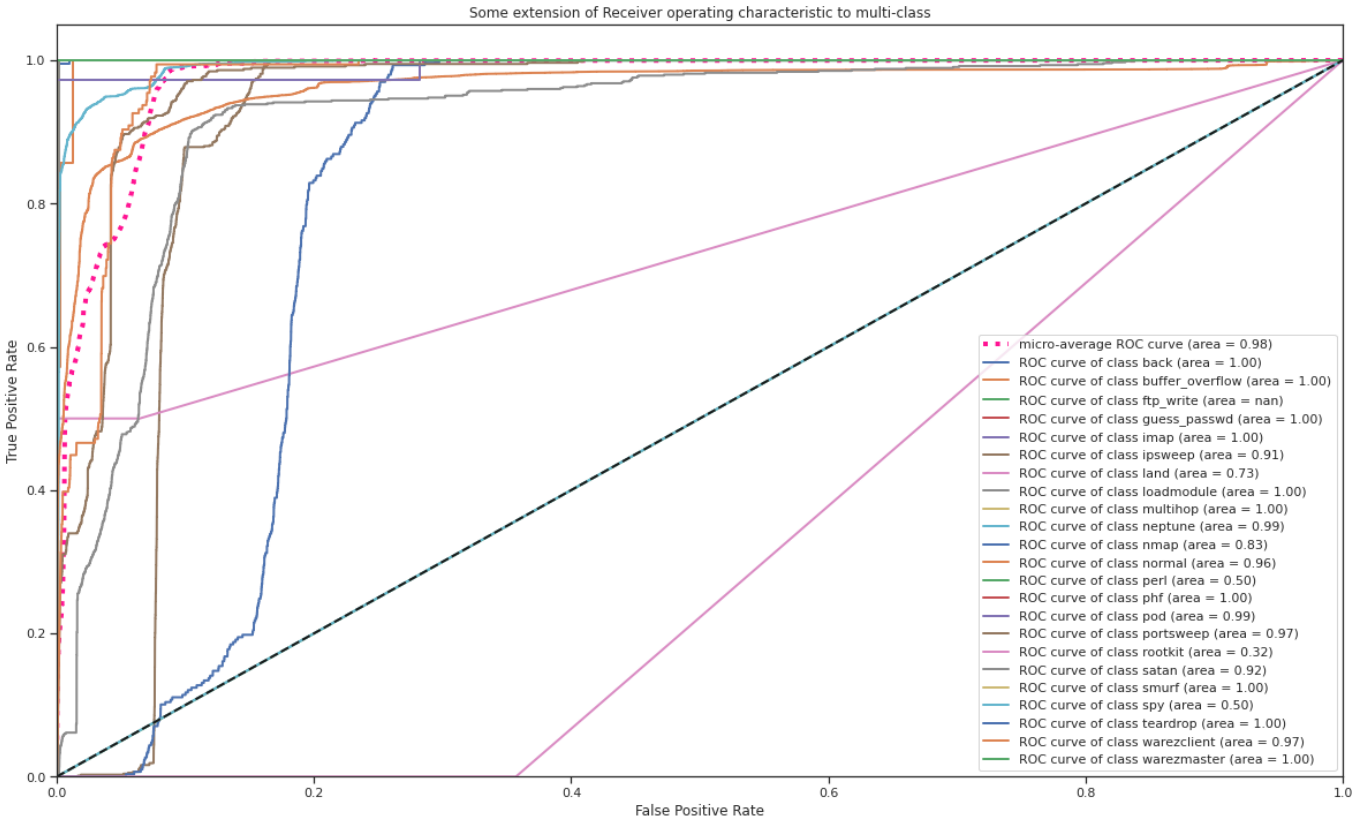

Figure 4. RoC curve of Naive Bayes on NSL-KDD Dataset

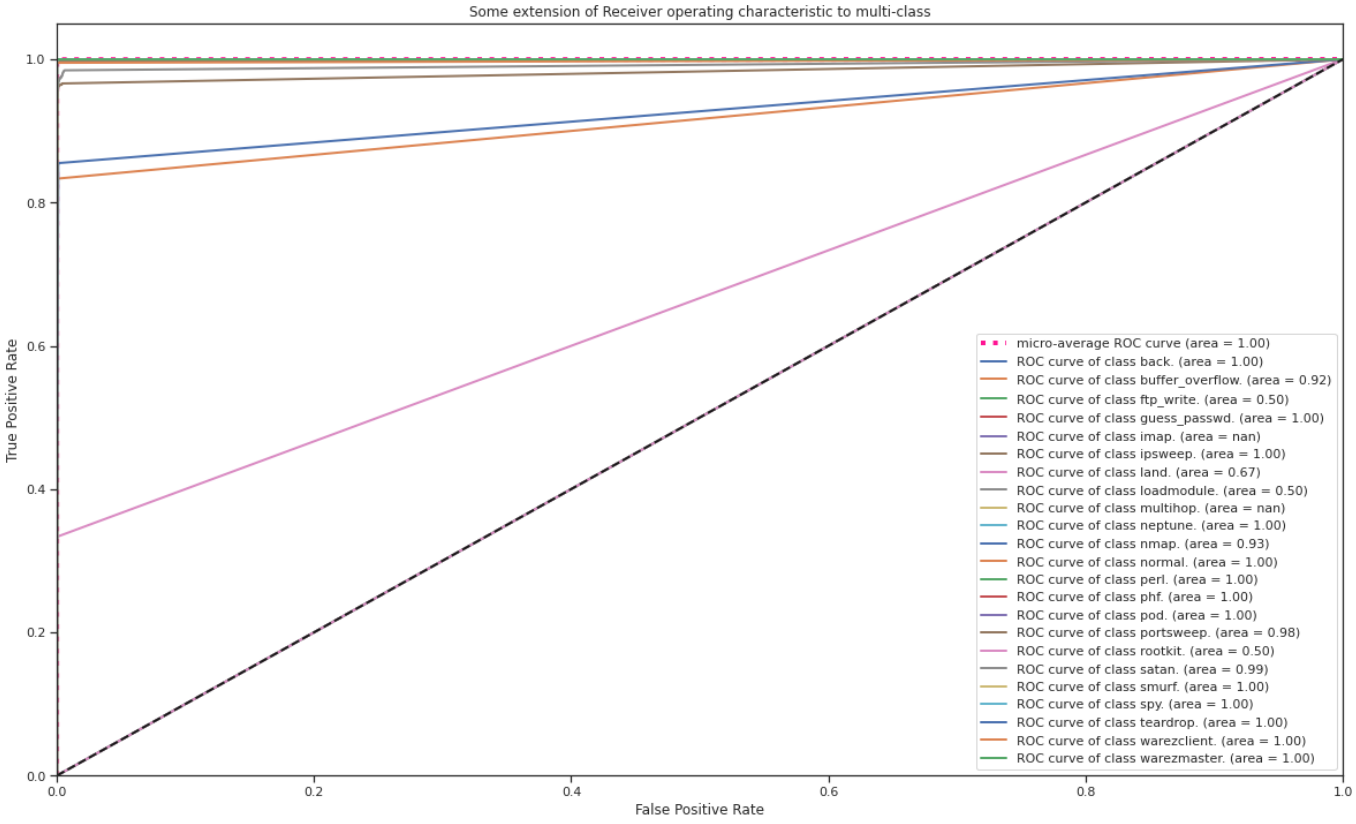

Figure 5. RoC curve of RF on KDD-CUP-99 Dataset

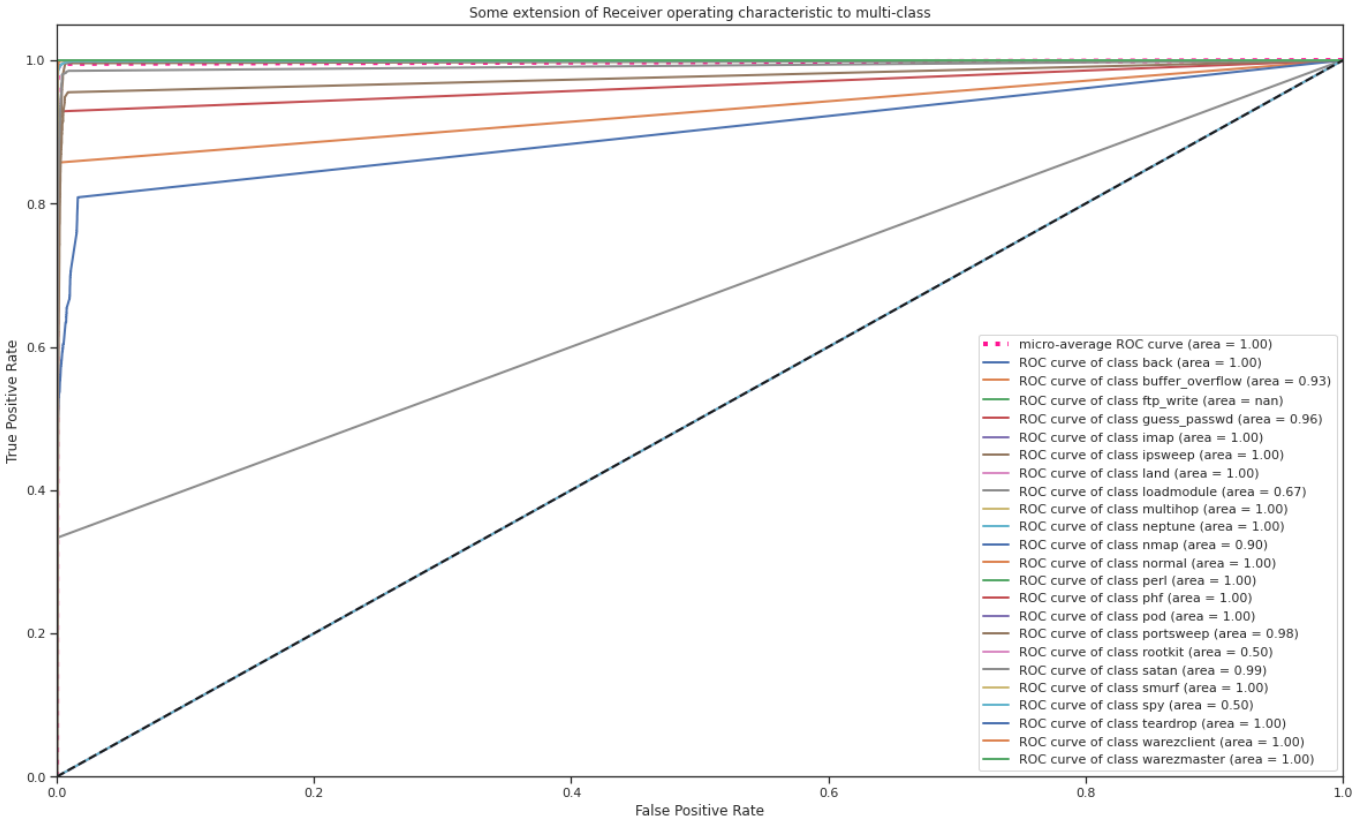

Figure 6. RoC curve of RF on for NSL-KDD Dataset

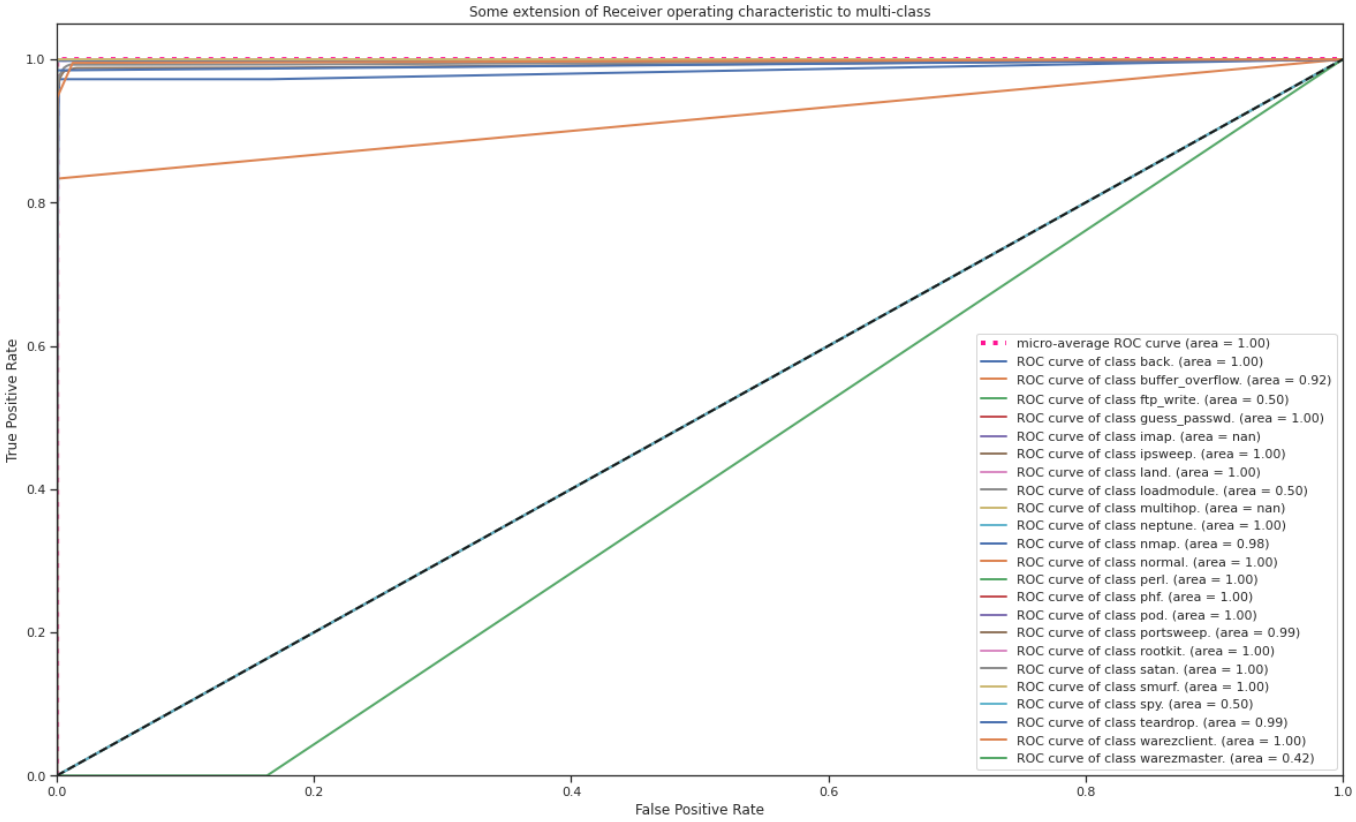

Figure 7. RoC curve of Decision Tree on KDD-CUP-99 Dataset

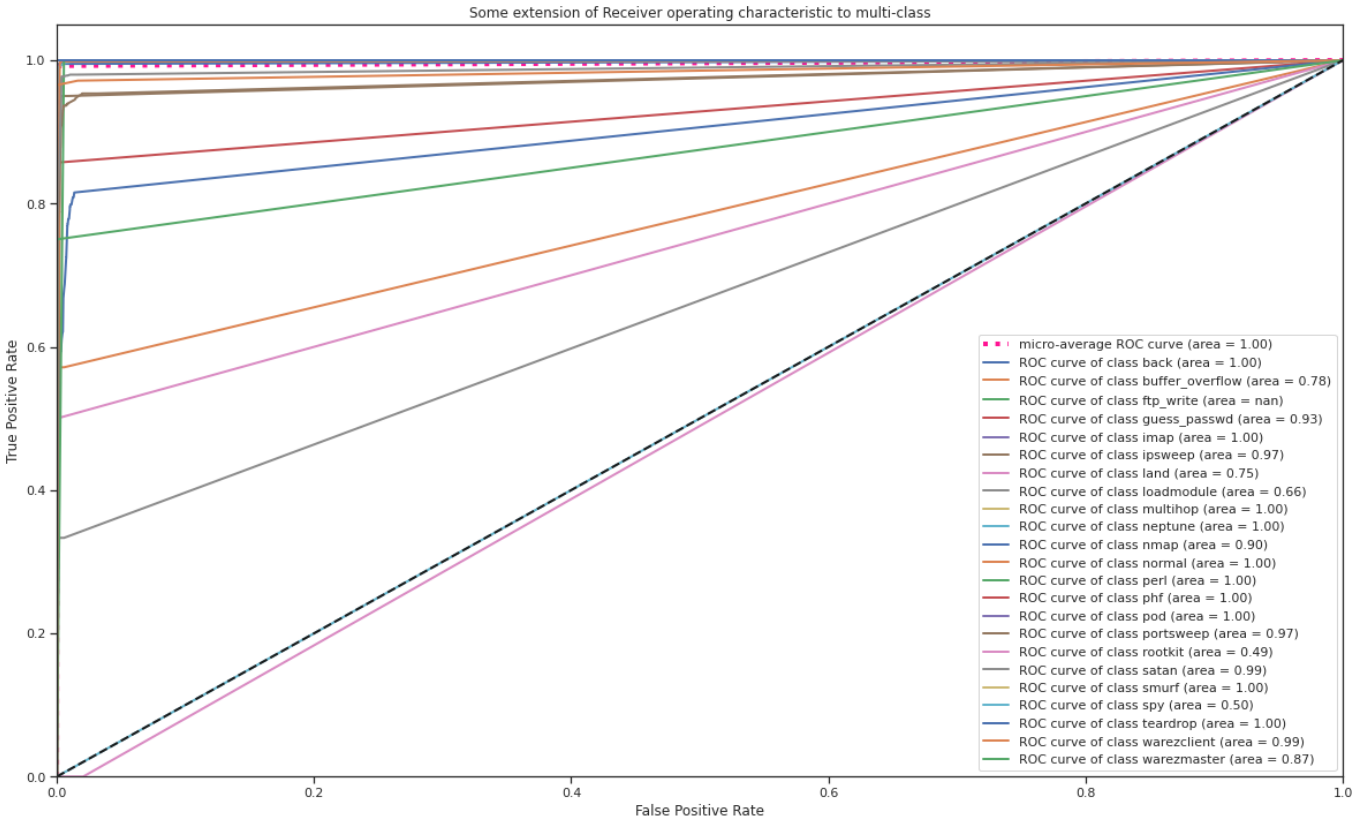

Figure 8. RoC curve of Decision Tree on NSL-KDD Dataset

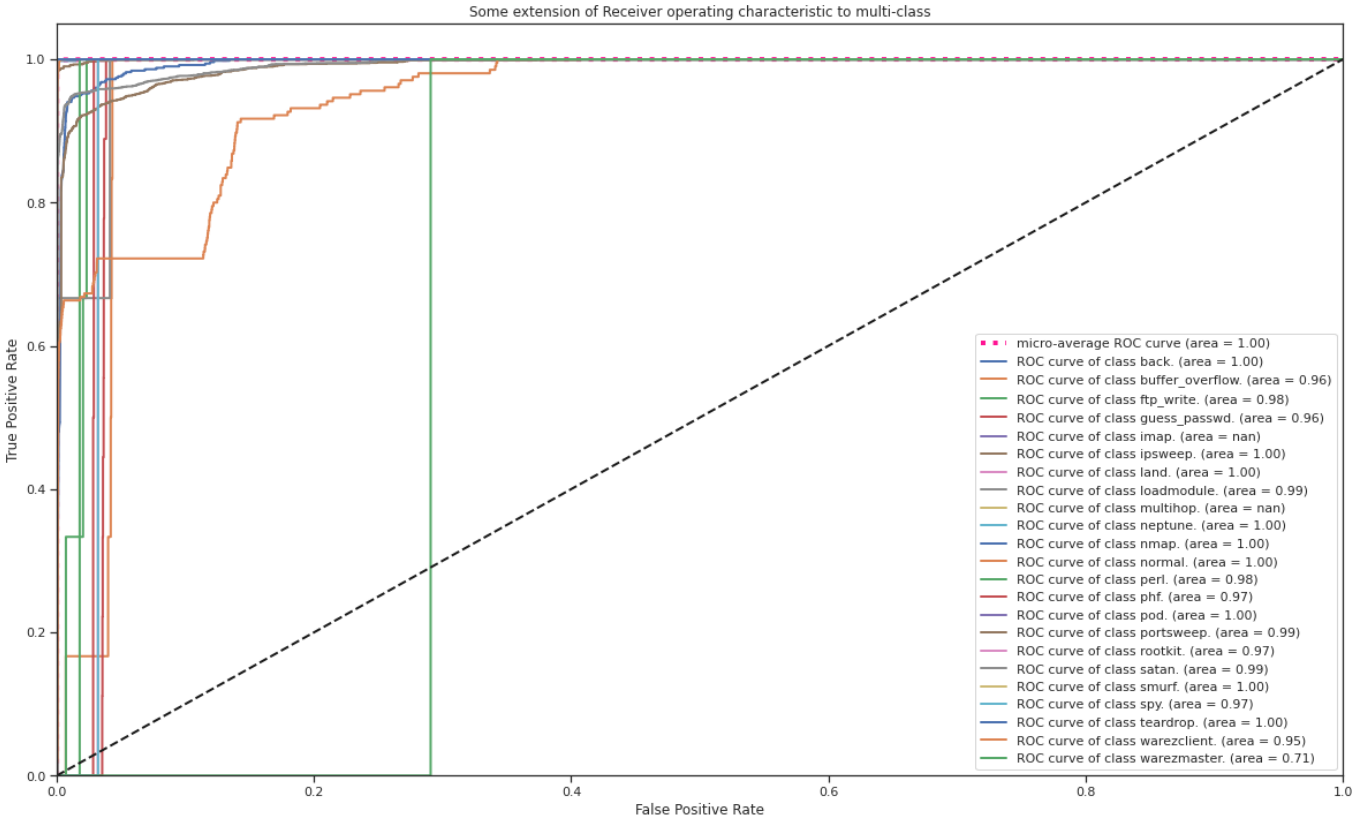

Figure 9. RoC curve of MLP on KDD-CUP-99 Dataset

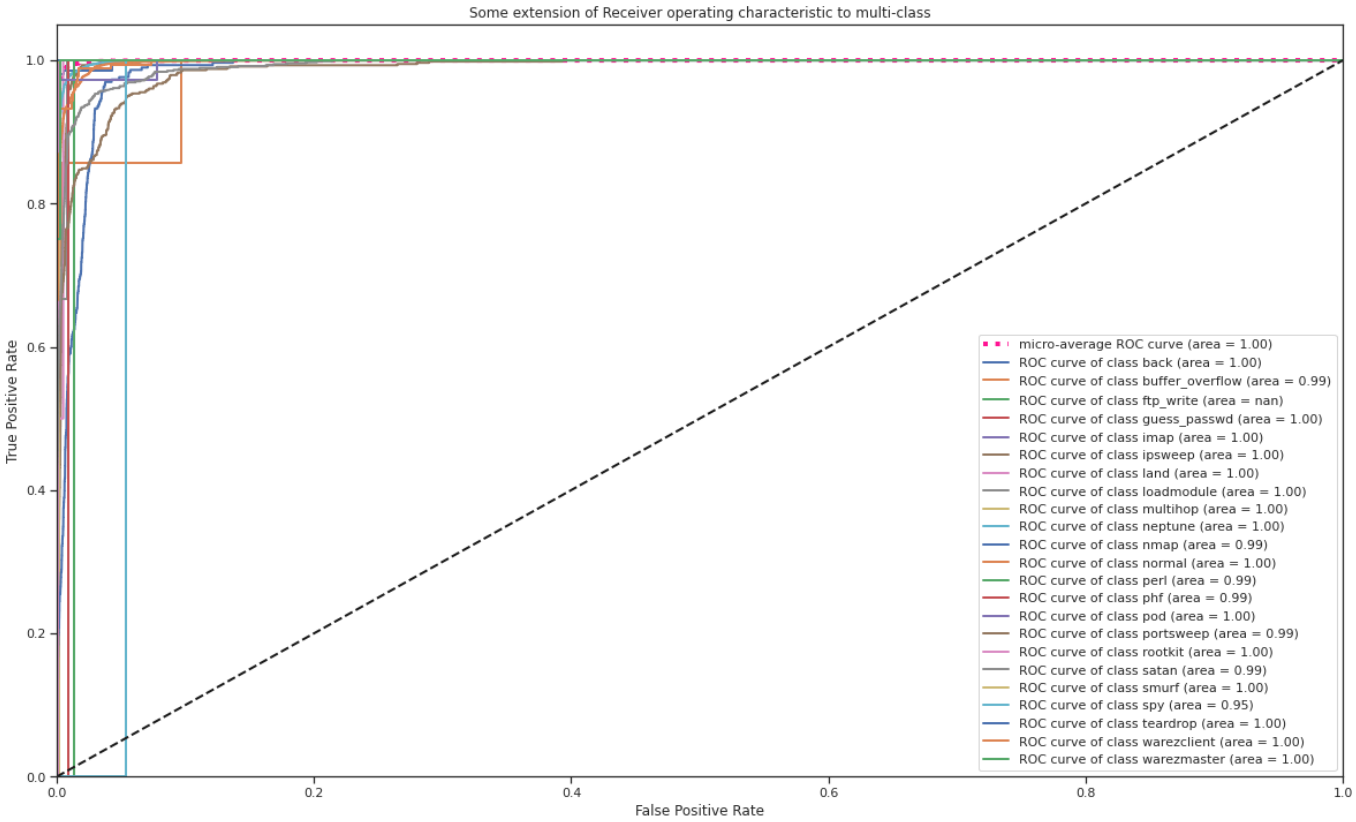

Figure 10. RoC curve of MLP on NSL-KDD Dataset

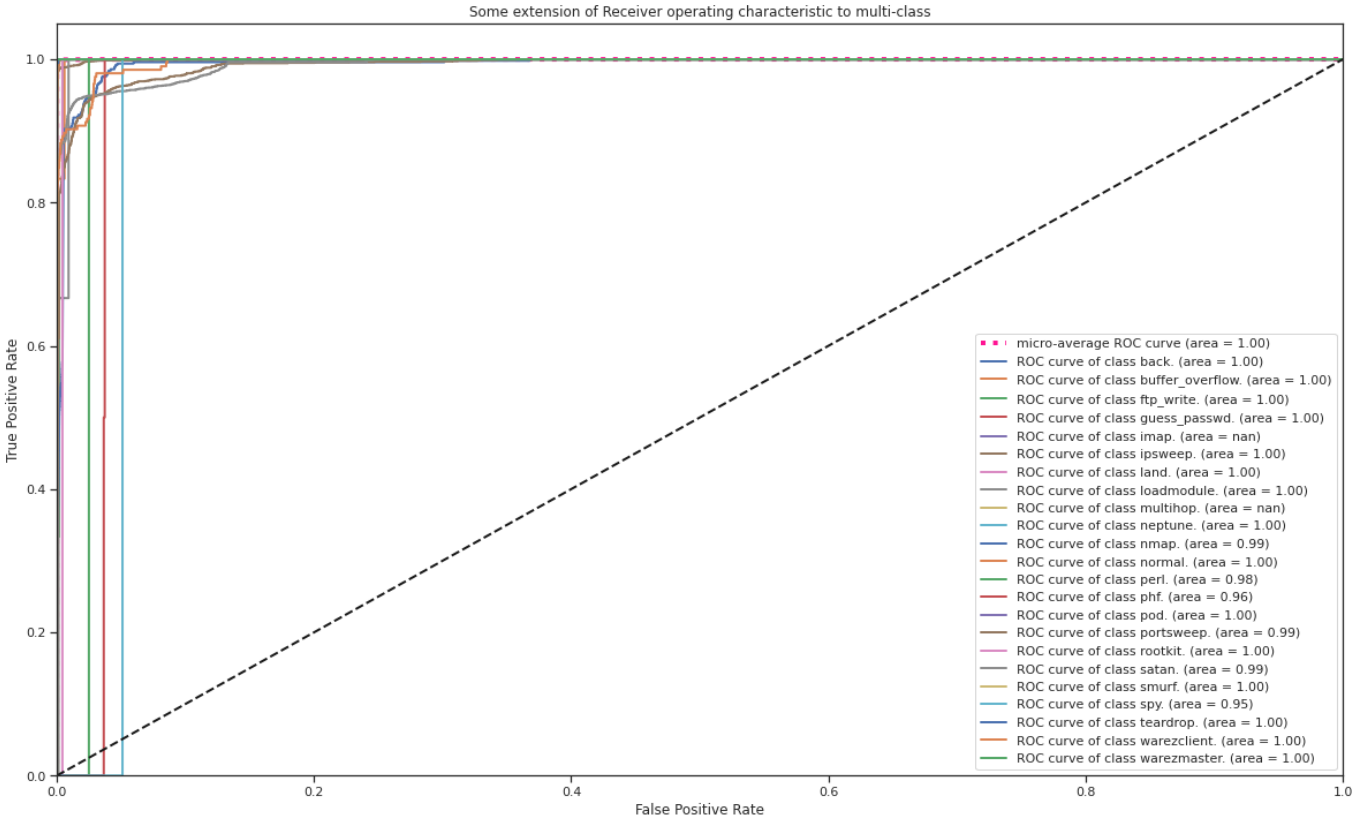

Figure 11. RoC curve of LSTM on KDD-CUP-99 Dataset

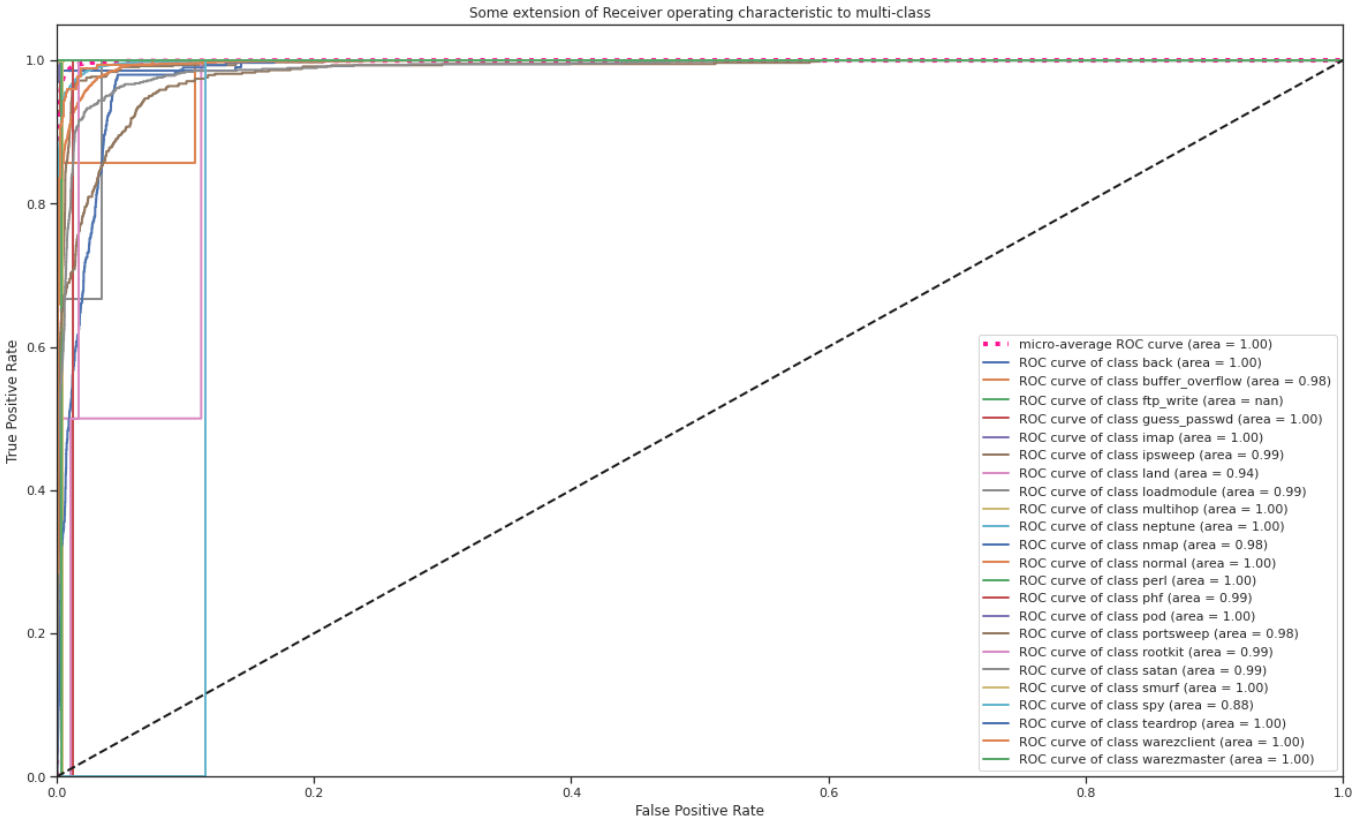

Figure 12. RoC curve of LSTM on NSL-KDD Dataset

REFERENCES

[1] Xu Li, Rongxing Lu, Xiaohui Liang, Xuemin Shen, Jiming Chen, and Xiaodong Lin. Smart community: an internet of things application. *IEEE Communications magazine*, 49(11):68–75, 2011.

[2] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[3] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. IEEE, 2009.

[4] 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*, 12:2825–2830, 2011.

[5] Abhishek Verma and Virender Ranga. Machine learning based intrusion detection systems for iot applications. *Wireless Personal Communications*, 111(4):2287–2310, 2020.

[6] Jiyeon Kim, Jiwon Kim, Hyunjung Kim, Minsun Shim, and Eunjung Choi. Cnn-based network intrusion detection against denial-of-service attacks. *Electronics*, 9(6):916, 2020.

[7] Pritesh Nagar, Hemant Kumar Menaria, and Manish Tiwari. Novel approach of intrusion detection classification deeplearning using svm. In *First International Conference on Sustainable Technologies for Computational Intelligence*, pages 365–381. Springer, 2020.

[8] Tongtong Su, Huazhi Sun, Jinqi Zhu, Sheng Wang, and Yabo Li. Bat: Deep learning methods on network intrusion detection using nsl-kdd dataset. *IEEE Access*, 8:29575–29585, 2020.

[9] Mohammad Almseidin, Maen Alzubi, Mouhammd Alkasassbeh, and Szilveszter Kovacs. Applying intrusion detection algorithms on the kdd-99 dataset. *PRODUCTION SYSTEMS AND INFORMATION ENGINEERING*, 8:51–67, 2019.

[10] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING, 2003.

[11] John McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):262–294, 2000.

[12] David J Weller-Fahy, Brett J Borghetti, and Angela A Sodemann. A survey of distance and similarity measures used within network intrusion anomaly detection. *IEEE Communications Surveys & Tutorials*, 17(1):70–91, 2014.

[13] NG Amma and S Selvakumar. A statistical class center based triangle area vector method for detection of denial of service attacks. *CLUSTER COMPUTING-THE JOURNAL OF NETWORKS SOFTWARE TOOLS AND APPLICATIONS*, 2020.

[14] Nerijus Paulauskas and Juozas Auskalnis. Analysis of data preprocessing influence on intrusion detection using nsl-kdd dataset. In *2017 open conference of electrical, electronic and information sciences (eStream)*, pages 1–5. IEEE, 2017.

[15] Suleman Khan, Abdullah Gani, Ainuddin Wahid Abdul Wahab, and Prem Kumar Singh. Feature selection of denial-of-service attacks using entropy and granular computing. *Arabian Journal for Science and Engineering*, 43(2):499–508, 2018.

[16] CP Subha, S Malarkan, and K Vaithinathan. A survey on energy efficient neural network based clustering models in wireless sensor networks. In *2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT)*, pages 1–6. IEEE, 2013.

[17] Tao Ma, Fen Wang, Jianjun Cheng, Yang Yu, and Xiaoyun Chen. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors*, 16(10):1701, 2016.

[18] Serafeim Tsironis. Accurate spectral clustering for community detection in mapreduce. 2013.

[19] Harry Zhang. The optimality of naive bayes. volume 2, 01 2004.

[20] L. Breiman. Random forests, machine learning 45. *Journal of Clinical Microbiology*, 2:199–228, 01 2001.

[21] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and regression trees*. 01 2017.

[22] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.