Intrusion Detection Using Multiple Machine Learning Models

Mustavi Ibne Masum, Yumna Islam, and Farhana Azad
180204040, 180204046, and 180204068
Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh

***Abstract*—Since the very beginning of information transmission, there has been a challenge with protecting data flows and information. Many strategies have been developed to safely transfer and protect such information. Yet, as communications and technology develop and information management systems become more potent and widespread, the issue has grown more intricate and is now a significant difficulty. The severity of the issue has grown as a result of the growing use of wired and wireless communication networks, the internet, web applications, and computing. IDS keeps an eye out on a computer network for attempts to steal information. High detection rates and low False Alarm Rates are possible outcomes of applying ML. In this research, we've used different feature selection and ML techniques in order to detect intrusion.**

***Index Terms*—Machine Learning, IDS, PCA, NSL-KDD**

I. Introduction

One of the best resources for learning about the current world is the Internet, which has advanced with time. The internet is a significant tool for both professional and academic endeavors. Hence, data sent via the Internet must be secure. Internet security is one of the major problems in the world today. Because the Internet is continuously under assault, it is imperative to develop a system to protect both the users who use the data and the data itself. In order to fulfill that need, the intrusion detection system (IDS) was developed. To stop malicious attempts, network administrators modify intrusion detection systems.

In order to find out whether there are any unusual behaviors against the rules of the system or violent signs in the network, intrusion detection has emerged to collect and analyze a number of important points in computer systems and networks. IDS stands for hardware and software intrusion detection system.

Technologies for detecting intrusions are used to spot two main types of attacks: misuse detection and anomaly detection. By watching patterns that differ from established standards, anomaly detection aims to "learn" the characteristics of event patterns that represent normal activity and can identify when an intrusion has taken place. In order to spot misuse, user behavior is compared to well-known network attack techniques. Both misuse detection tools and antivirus software using rule-based approaches to distinguish between known attacks by comparing the attack's pattern to a database of signatures.

Many machine learning models, including Random Forest, Decision Trees, K-Nearest Neighbors, Naive Bayes classifier, Logistic Regression, and AdaBoost classifier, were used to perform an extensive investigation on intrusion detection in this paper. First, we preprocessed the NSL-KDD dataset [1] and three different feature selection methods are used: F Mutual Information Classifiers, Chi2, F Classifiers, and PCA (Principal Component Analysis). Then, we carried out the intersection and union operations. We retrieved 28 features from the union operation and 12 features from the intersection operation.

The Random Forest classifier has the highest accuracy in both union and intersection cases, whereas the Naive Bayes classifier has the lowest accuracy.

II. Related Works

Nutan Farah Haq et al. (2015) [2] worked on a survey paper 'Application of Machine Learning Approaches in Intrusion Detection System: A Survey'. This paper conducted a relative study on intrusion detection systems which enlisted 49 papers from 2009 to 2014 focusing on classifier design(single, hybrid, and ensemble). Statistical comparison of classifier design, algorithms, datasets, and feature selection have been included in this paper.

Iram Abrar et al. (2020) [3] did a study on 'A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset'.In order to categorize the data in this study as normal or intrusive, machine learning (ML) classifiers including support vector machines (SVM), K-nearest neighbors (KNN), logistic regression (LR), Naive Bayes (NB), multi-layer perceptrons (MLP), random forests (RF), Extratree classifier (ETC), and decision trees (DT) were used. Four feature subsets from the NSLKDD dataset that were retrieved were used to analyze the model performance.Results show that for all attack classes using different feature subsets, the performance of RF, ETC, and DT was above 99%. The recommended model thus reduces computational complexity by removing irrelevant components and has a high rate of accurate prediction.

Nutan Farah Haq et al. (2015) [4] worked on a paper 'An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA)'. This study looked at various intrusion detection strategies that used wrapper approaches and machine learning techniques. This research primarily focuses on the classification precision of three different classifiers employing the smallest number

of features chosen by three different wrapper search algorithms on the well-known public type NSL-KDD dataset and providing differences between them. The Bayesian Network, Naive Bayes, and J48 are the three fundamental classifiers.The results of this paper suggested ensemble technique outperformed those of Naive Bayes, Bayesian Network, and J48 classifiers, according to the experiment.

Jinxin Liu et al. (2020) [5] proposed a study on 'Machine Learning-Driven Intrusion Detection for Contiki-NG-Based IoT Networks Exposed to NSL-KDD Dataset'. The NSL-KDD dataset, specifically the Routing Protocol for Low-Power and Lossy Networks (RPL) and 6LoWPAN networks, employing the Contiki-NG operating system, is used in this study to examine potential DoS and probing attacks. The resulting dataset is also fed into eleven ML algorithms to test their capacity to categorize various threats. In comparison to the remaining approaches, Bayes Network, Naive Bayes (NB), and Adaboost, tree-based methods and ensemble algorithms like XGBoost, Decision Trees (DTs), Bagging Trees, and Random Forest performs well and attain more than 96% accuracy.

Ravipati Rama Devi et al. (2019) [6] has worked on a review paper 'Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets -A review paper'. In this research, classifiers for the KDD99 and NSL-KDD datasets as well as an overview of several machine learning techniques for the Intrusion Detection Systems (IDS) and various detection methodologies have been presented. The experiment's findings demonstrate that KNN has a high false-positive and false-negative detection rate while the AdaBoost method has a very low false-positive and as compared to other supervised algorithms, this approach has a higher detection rate and runs more quickly.

III. Background Study

We used the following feature selection algorithms to choose the important features:

A. Mutual Information Classifiers

Mutual Information calculates mutual information for continuous target variables in regression problems or fixed categories in classification tasks. The entropy of the variables is the foundation of mutual information. Mutual information (MI), which assesses the interdependence of two random variables, has a non-negative value. Higher values indicate greater dependence, and it equals 0 only when two random variables are independent.

B. Chi-Square

A statistical technique called the chi-square test is used to compare actual outcomes with predictions. The objective of the chi-square is to determine whether a disparity between observed and anticipated data is due to chance or a correlation between the variables under study. So, using a chi-square test is a fantastic alternative for better understanding and interpreting the relationship between our two category variables.

C. f-Classif

It is a feature selection technique that computes the ANOVA F-value for the provided sample. f-class If is one of the classes of the sklearn.feature selection module and these classes are useful for feature selection and dimensionality reduction on sample sets, either to increase the accuracy scores of estimators or to enhance their performance on extremely high-dimensional datasets.

D. PCA(Principal Component Analysis)

Principal component analysis (PCA) is a method for lowering the dimensionality of such datasets, improving interpretability while minimizing information loss. It accomplishes this by producing fresh, uncorrelated variables that maximize variance one after the other.

E. MinMaxScaler

MinMaxScaler divides by the range after dividing each value in a feature by its minimal value. The range is the difference between the maximum and smallest starting values. The feature that MinMaxScaler returns has a default range of 0 to 1. The original distribution's shape is maintained by MinMaxScaler. The information that was already present in the original data is not significantly altered.

F. K-nearest Neighbor

The k-nearest neighbors (KNN) algorithm is a supervised machine learning algorithm that can be used for both classification and regression problems.By calculating the distance between the test data and all of the training points, KNN tries to predict the proper class for the test data. Next choose the K points that are closest to the test data. The KNN method determines which classes of the "K" training data the test data will belong to, and the class with the highest probability is chosen. The value in a regression situation is the average of the 'K' chosen training points.

G. Naive Bayes Classifier

The Bayes Theorem is the foundation of the probabilistic machine learning method known as Naive Bayes, which is utilized for a variety of classification problems. Naive Bayes is based on two assumptions. It makes the assumption that a Naive Bayes model's predictors are conditionally independent. It also assumes that each feature affects the outcome equally. Each variable just needs to have one probability, which simplifies the computation of the model. The classification method performs effectively despite this irrational independence assumption, especially with small data sets.

H. Logistic Regression

Modeling the probability of a discrete result given an input variable is what logistic regression does. The most popular types of logistic regression models a binary result, such as true or false, yes or no, and so on. Using multinomial logistic regression, events with more than two distinct possible outcmodelan be modeled.Classification problems are a suitable place to use logistic regression as an analysis tool when trying to determine which category a new sample most closely matches.

I. Random Forest

Random forest is a Supervised Machine Learning algorithm that is frequently employed in Classification and Regression problems.Using various samples, it constructs decision trees and uses their average for classification and majority vote for regression.It is capable of handling data sets with continuous variables, such as those used in regression and categorical variables, as in the case of classification.

J. Decision Tree

For classification and regression, decision trees (DTs) are a non-parametric supervised learning technique.The objective is to learn straightforward decision rules derived from the data features in order to build a model that predicts the value of a target variable.In a decision tree, which resembles a flowchart, each internal node indicates a test on an attribute, each branch shows the test's result, and each leaf node (or terminal node) has a class label.

K. Adaptive Boosting(Adaboost)

Ada-boost or Adaptive Boosting is an iterative ensemble boosting classifier.AdaBoost classifier combines a number of weak classifiers to create a strong classifier that has a high degree of accuracy. Adaboost's core idea is to train the data sample and set the classifier weights in each iteration to ensure accurate predictions of unusual observations.

IV. Dataset

The NSL-KDD [1] dataset is taken from Kaggle. The NSL-KDD data set is an update to the KDD'99 dataset. This dataset can be used as a benchmark data set to effectively compare various intrusion detection strategies. One training set and two testing sets make up the environment. The redundant data is removed after combining the training and testing datasets, and the combined dataset is then split into two portions: training data made up 80 percent and testing data 20 percent.

In this dataset, there is no redundant data included in the training set, hence the classifiers won't be biased toward frequent records. The suggested test sets do not contain duplicate data; as a result, the learners' performance is unaffected by approaches with higher detection rates for frequent records. The proportion of records in the original KDD data set that is selected from each difficulty level group is inversely related to the number of records chosen from those groups. Because of this, there is a broader range of variation in the classification rates of various machine learning algorithms, which makes it easier to evaluate various learning methods accurately.

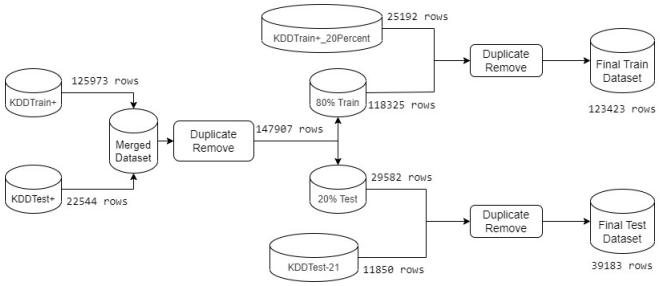

Fig. 1. Dataset Preprocessing

The training dataset is composed of 118325 data and the test dataset is composed of 29582 data where we have 41 features. The features of this dataset are given below:

TABLE I
FEATURES

Index	Feature Name	Index	Feature Name
1.	duration	22.	is_guest_login
2.	protocol-type	23.	count
3.	service	24.	srv-count
4.	flag	25.	serror-rate
5.	src-bytes	26.	srv-serror-rate
6.	dst-bytes	27.	rerror-rate
7.	land	28.	srv-rerror-rate
8.	wrong-fragment	29.	same-srv-rate
9.	urgent	30.	diff-srv-rate
10.	hot	31.	srv-diff-host-rate
11.	num-failed-logins	32.	dst-host-count
12.	logged-in	33.	dst-host-srv-count
13.	num-compromised	34.	dst-host-same-srv-rate
14.	root-shell	35.	dst-host-diff-srv-rate
15.	su-attempted	36.	dst-host-same-src-port-rate
16.	num-root	37.	dst-host-srv-diff-host-rate
17.	num-file-creations	38.	dst-host-serror-rate
18.	num-shells	39.	dst-host-srv-serror-rate
19.	num-access-files	40.	dst-host-rerror-rate
20.	num-outbound-cmds	41.	dst-host-srv-rerror-rate
21.	is-host-login		

There is no null value for any of the features in the dataset. We have combined several classes for programming convenience and replaced them with new class names. As a result, we get 5 classes instead of 40 classes: Normal, Dos, Probe, R2L, and U2R.

V. Methodology

We first performed some preprocessing on our dataset in order to identify the essential features before running several feature selection algorithms to detect intrusion. Last but not least, we tested a few machine-learning models and displayed the performance metrics.

The steps we took are as follows:

1. Preparing Dataset
2. Execution of Feature selection algorithm
3. Execution of Machine learning Models
4. Performance Metrices

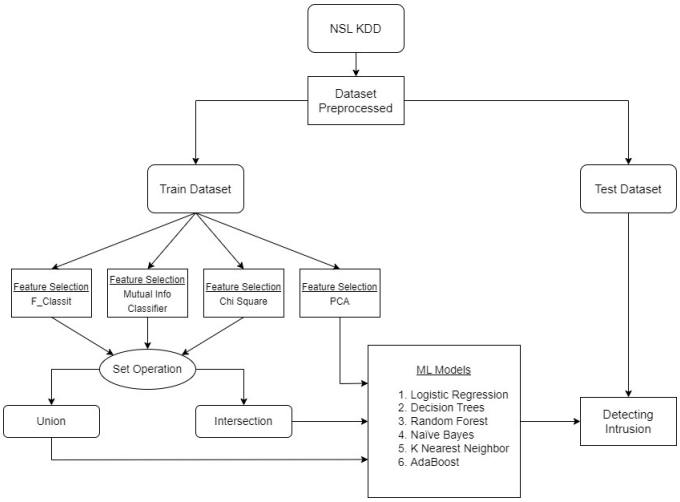

Fig. 2. Methodology

A. Preparing Dataset

Our dataset was obtained through Kaggle. Here, we preprocessed our dataset. Initially, we combined the train and test datasets, and then we deleted any duplicate data. After that, we divided the combined dataset, taking data for training purposes in the amount of 80% and data for testing purposes in the amount of 20%. Next, we adjusted our labels. Since the original dataset had 40 classes, we combined some of them to make 5 new classes rather than 40.

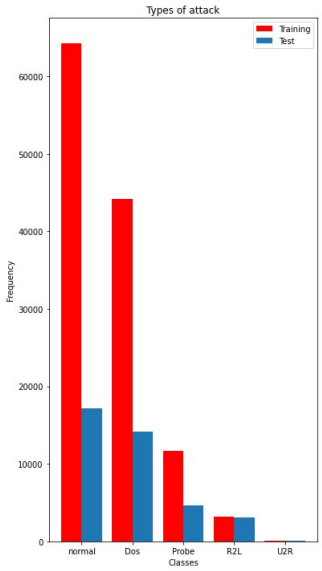

Fig. 3. Frequency according to minimized classes

B. Execution of Feature Selection Algorithm

Feature selection is a technique used in machine learning to improve accuracy. By focusing on the most important variables and removing the redundant and unimportant ones, it also improves the algorithms' ability to anticipate outcomes.To determine the most important features for our study, we applied three different feature selection algorithms: F Mutual Information Classifiers, Chi2, F Classif. We selected the top 20 features from the output after applying each of the algorithms, and we continued our research using those features.Using those features obtained from the output of each algorithm, we then performed union and intersection operations. From union operation, we obtained 28 features, and from intersection operation, 12 features. We have also used PCA algorithm on our dataset.

TABLE II
INDEX OF SELECTED FEATURES

Set Operation	Feature Index
Union	1,3,4,5,6,10,12,13,16,23,24,25,26,27, 28,29,30,31,32,33,34,35,36,37,38,39,40,41
Intersection	4,12,23,25,26,29,32,33,34,36,38,39

C. Execution of Machine learning Models

To obtain the class on the testing data, we have used a variety of machine learning models. The Random Forest Classifier, Decision Trees, K-Nearest Neighbors, Naive Bayes Classifier, Logistic Regression, and AdaBoost classifier are among the six machine learning models that we have tested. In both union and intersection scenarios, the Random Forest classifier has the highest accuracy and Naive Bayes Classifier has the lowest accuracy.

D. Performance Metrices

Using the Accuracy metric is the easiest approach to assess a classifier's performance. Each data point's actual class and anticipated class are compared in this case, and a match counts as one accurate prediction.The accuracy is then calculated as the proportion of accurate forecasts to all other incorrect ones.

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

The proportion of accurately categorized positive samples (True Positive) to the total number of positively classified samples is known as precision (either correctly or incorrectly). Precision is calculated as follows:

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

The recall is determined by dividing the total number of Positive samples by the number of Positive samples that were correctly identified as Positive. The model's capacity to identify positive samples is gauged by the recall. More positive samples are found when recall is higher. The formula for recall is as follows:

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

F1 takes into consideration both Precision and Recall. It is calculated as follows:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

F1-score is basically the harmonic mean of precision and recall and provides a balance between them.
The receiver operating characteristic curve (ROC curve) is a graph that displays how well a classification model performs across all categorization levels. Two parameters are plotted on this curve, which are the true positive rate and the false positive rate. TPR vs. FPR are plotted on a ROC curve at various categorization criteria. More items are classified as positive when the classification threshold is lowered, which raises the number of both False Positives and True Positives. The abbreviation "Area under the ROC Curve" is AUC. In other words, AUC calculates the area in two dimensions that lies beneath the complete ROC curve.AUC has a value between 0 and 1. A model with 100 percent incorrect predictions has an AUC of 0, while a model with 100 percent correct predictions has an AUC of 1.

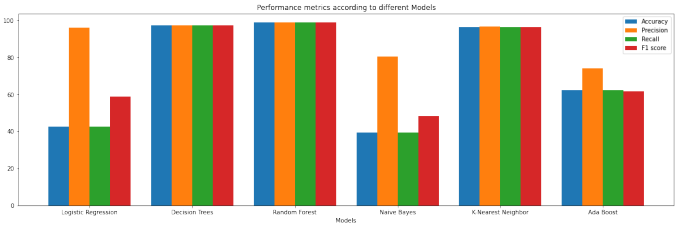

Fig. 4. Performance matrices of Different models on PCA

Macro averaging reduces our multiclass predictions down to multiple sets of binary predictions, which then calculates the associated metric for each of the binary cases and averages the outcomes.

VI. Experimental Result

We achieve good accuracy for the random forest classifier (98.6219 %) and decision tree (97.1952 %) after applying PCA to our original dataset. Here, the Naive Bayes classifier's (39.554) and Logistic Regression's(42.4751 %) accuracy are also low.

TABLE III
Performance on Union features

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	82.9186	89.3291	82.9186	85.3976
Decision Trees	94.6150	95.3421	94.6150	94.7504
Random Forest	98.1574	98.3114	98.1574	98.2005
Naive Bayes	56.8639	62.6411	56.8639	53.4127
K-Nearest Neighbor	96.5010	96.7928	96.5010	96.6288
Ada Boost	79.0802	80.3575	79.0802	78.5824

We used minmaxscaler algorithm on the union and intersection dataset in order to normalize data. For the Union dataset, the Random Forest classifier (98.1574%) and K-nearest neighbors (96.5010%) both do quite well. Ada Boost's accuracy is modest (79.0802%), and the naïve bayes classifier (56.8639%) performs poorly.

TABLE IV
Performance on Intersection features

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	75.5710	84.0964	75.5710	79.1565
Decision Trees	95.3322	95.5268	95.3322	95.3965
Random Forest	95.4419	95.7371	95.4419	95.5293
Naive Bayes	58.3059	64.1100	58.3059	52.8410
K-Nearest Neighbor	91.7439	92.7362	91.7439	92.0985
Ada Boost	45.0068	61.1897	45.0068	50.4278

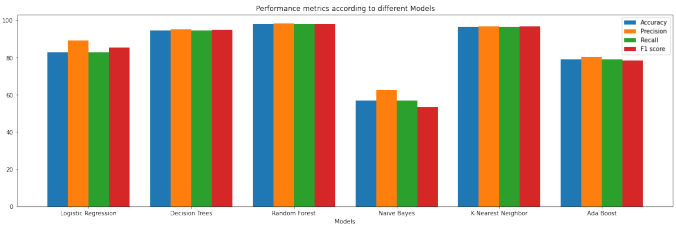

Fig. 5. Performance matrices of Different models on union features

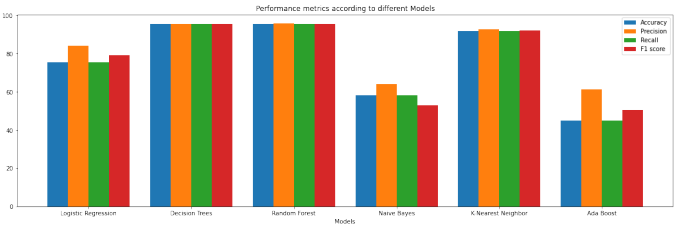

Fig. 6. Performance matrices of Different models on intersection features

In the intersection dataset, we get high accuracy for the random forest classifier(95.4419%) and decision tree(95.3322%) and low accuracy for the naive bayes classifier(58.3059 %) and Ada boost(45.0068%).

VII. Result Analysis

Using decision trees, random forest classifiers, and nearest neighbors, we consistently obtain good accuracy. After examining every scenario, we can conclude that logistic regression depends on number of features because it performs poorly for PCA and intersection and performs well for union. Hence, we

TABLE V
Performance on PCA features

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	42.4751	95.8147	42.4751	58.8505
Decision Trees	97.1952	97.2065	97.1952	97.1958
Random Forest	98.6219	98.7142	98.6219	98.6428
Naive Bayes	39.1854	80.4338	39.1854	48.3410
K-Nearest Neighbor	96.1846	96.4246	96.1846	96.2816
Ada Boost	62.1673	74.1156	62.1673	61.7189

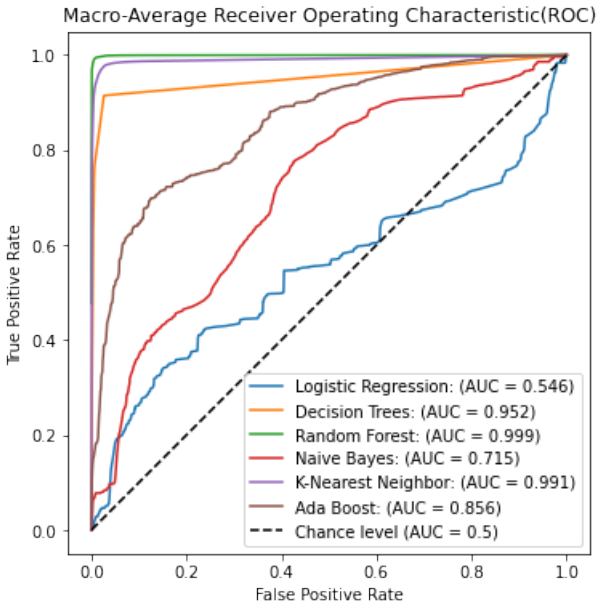

Fig. 7. Macro Avg ROC Curve for PCA

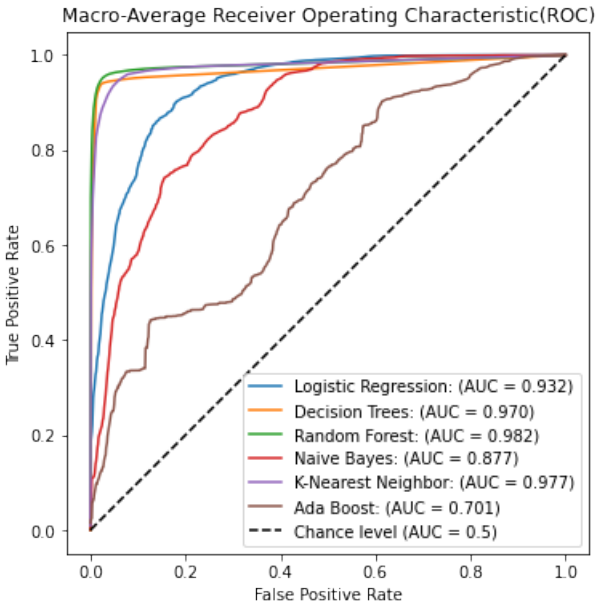

Fig. 9. Macro Avg ROC Curve for Intersection Features

can say that it might function well with a lot of features. For union, we obtain a decent ROC curve. Although ada boost is an ensemble learning strategy, we should be able to achieve a good result from it, but we aren't. Moreover, the naïve Bayes classifier struggles in all three situations.

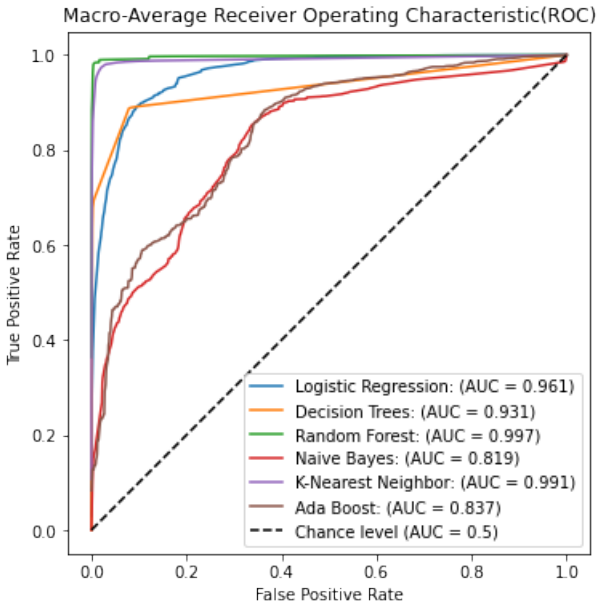

Fig. 8. Macro Avg ROC Curve for Union Features

VIII. Conclusion

The protection of information from harmful behavior or attackers is essential to the quick growth of information technology. The application of various classifier techniques in intrusion detection systems is a new area of research in machine learning. Many machine learning models, including the Random Forest Classifier, Decision Trees, K-Nearest Neighbors, Naive Bayes Classifier, Logistic Regression, and AdaBoost classifier, were utilized in this study to identify network intrusion, and the effectiveness was measured using the NSL-KDD. After preprocessing the dataset, the model was trained and tested based on the significant attributes.

References

[1] M. H. Zaib, "Nsl-kdd," Apr 2019. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[2] N. F. Haq, A. R. Onik, M. A. K. Hridoy, M. Rafni, F. M. Shah, and D. M. Farid, "Application of machine learning approaches in intrusion detection system: a survey," *IJARAI-International Journal of Advanced Research in Artificial Intelligence*, vol. 4, no. 3, pp. 9–18, 2015.

[3] I. Abrar, Z. Ayub, F. Masoodi, and A. M. Bamhdi, "A machine learning approach for intrusion detection system on nsl-kdd dataset," in *2020 international conference on smart electronics and communication (ICOSEC)*. IEEE, 2020, pp. 919–924.

[4] N. F. Haq, A. R. Onik, and F. M. Shah, "An ensemble framework of anomaly detection using hybridized feature selection approach (hfsa)," in *2015 SAI Intelligent Systems Conference (IntelliSys)*. IEEE, 2015, pp. 989–995.

[5] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for contiki-ng-based iot networks exposed to nsl-kdd dataset," in *Proceedings of the 2nd ACM workshop on wireless security and machine learning*, 2020, pp. 25–30.

[6] R. D. Ravipati and M. Abualkibash, "Intrusion detection system classification using different machine learning algorithms on kdd-99 and nsl-kdd datasets-a review paper," *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, vol. 11, 2019.