Threat Detection Using Rf Algorithm

[1] Mrs.Sangeetha.V, [2] Balaji.S, [3] Balakrishnan.S, [4] Hariharan.G.S
Kamaraj College of Engineering And Technology

Date of Submission: 05-03-2023 Date of Acceptance: 15-03-2023

ABSTRACT:Due to the advance of information and communication techniques, sharing information through online has been increased. As a result, various online services were created. However, as increasing connection points to the internet, the threats of cybersecurity have also been increasing. Threat Detection have nowadays become a necessary component of almost every security infrastructure. Threat detection plays an important role in ensuring information security, and the key technology is to accurately identify various attacks in the network. Analyzing network flows, logs, and system events has been used for threat detection. Network flows, logs, and system events, etc. generate big data. Big Data Analytics can correlate multiple information sources into a coherent view, identify anomalies and suspicious activities, and finally achieve effective and efficient intrusion detection. In this project,an TDS(Threat Detection System) with Machine learning algorithms in Big Data Analytics was developed.It is based on Random Forest (RF) Algorithm which is an ensemble classifier and performs well compared to other traditional classifiers for effective classification of attacks. To evaluate the performance of our model, experiments on KDD99 data set was conducted.

KEYWORDS: Cybersecurity,Threat Detection,Big Data Analytics, KDD99 .

I. INTRODUCTION

Threat Detection System (TDS) solutions excel in monitoring network traffic and detecting anomalous activity. They are placed at strategic locations across a network or on devices themselves to analyze network traffic and recognize signs of a potential attack. The TDS works by looking for the signature of known attack types or detecting activity that deviates from a prescribed normal. It then alerts or reports these anomalies and potentially malicious actions to administrators so they can be examined at the application and protocol layers. This enables organizations to detect the potential signs of an attack beginning or being carried out by an attacker.

In our project, the Threat Detection System (TDS) based on Machine Learning predicts the network attacks such as DOS, probe, U2R, R2L with highest accuracy. It uses RF (Random Forest) Algorithm. It is one of the powerful methods of supervised machine learning algorithm which improves the accuracy in the classification of attacks compared to other classifiers. Experiments on KDD99 data set will be conducted to evaluate the performance of the system.

In our project, we use Random Forest Algorithm. It is one of the popular supervised classification algorithms which can be used in python. Random Forest is a powerful and versatile supervised machine learning algorithm that grows and combines multiple decision trees to create a forest. A decision tree is another type of algorithm used to classify data.

[1]. Ajith Abraham et al presented some of the computational intelligence paradigms which could be useful for designing accurate intrusion detection systems which could be also deployed in a distributed environment.Computational intelligence approaches for intrusion detection was first implemented in mining audit data for automated models for intrusion detection.Raw data is converted into ASCII network packet information, which in turn is converted into connection level information. These connection level records containconnection features like service, duration etc. Besides several machine learning techniques and artificial immune systems, several intelligent paradigms have been explored to create models to detect intrusions. Artificial neural networks (ANN) have been used both in anomaly intrusion detection as well as in misuse intrusion detection. Support vector machines (SVM) have proven to be a good candidate for intrusion detection because of its training speed and scalability. Multivariate Adaptive Regression Splines (MARS) is an innovative approach that

automates the building of accurate predictive models for continuous and binary dependent variables. It excels at finding optimal variable transformations and interactions, and the complex data structure that often hides in high-dimensional data. Decision tree induction is one of the classification algorithms in the data mining. Classification algorithm is inductively learned to construct a model from the pre-classified data set. The inductively learned model of classification algorithm is used to develop IDS.

[2]. Arafat Ali presents a proposed model of monitoring part of the Common Intrusion Detection System(CIDS) using petri-nets modeling technique.It enhances the security of the system by monitoring system activity and detecting unusual behaviour.These system collect audit data which are the only way to built a real secure system.Petri-net is a powerful technique because it detects whether there is defect or not. It also has an advantage that it solves the problem that may occur due to concurrency of activities.

[3]. Fawaz Mokbal et al proposes a robust and effective intrusion detection framework based on the ensemble learning technique using Extreme Gradient Boosting (XGBoost) and an embedded feature selection method. The CICIDS2017 most up-to-date real-world intrusion dataset is used, including the most cutting-edge and common attacks and the benign samples. All dataset files are integrated into one dataset and derivative a uniform subset of features representing all attacks. The proposal IDS performance is evaluated in both multi-classification and binary problems. The proposed framework has successfully exceeded several evaluations on a big testing dataset and achieved remarkable and significant results with an accuracy overall, precision, detection rate (sensitivity), specificity, F-score, false negative rate, false positive rate, error rate of 99.90%, 99.90%, 99.97%, 99.90%, 99.90%, 5%, 0.2%, and 0.1%, respectively, in terms of the weighted average of multi-classification. Moreover, the achieved results of binary-classification are also remarkable. The evaluation results show that the strategy used with the IDS achieves noteworthy results in both the multi classification problem and binary problem. Moreover, results show that the proposed IDS framework is able to handle the huge and imbalanced dataset problem effectively and efficiently.

[4]. Heitor Scalco Neto et al proposes a methodology to build an Online Network Intrusion Detection System by using the Computational Intelligence technique called Random Forests and an API to preprocess the network packets. The random forests technique is applied and assessed to define the method efficacy of the intrusion detection in a computational environment. An Application Programming Interface (API) was developed for the proposed NIDS to operate in a real environment. The developed API can perform experiments with various network infrastructure simulations and in a real environment. The training of the technique was performed with the ISCX network traffic database. From the developed API, an auxiliary database was created for a test to address alternative traffic types to those found on ISCX in a smaller scale network, with various operational systems. This can test the effectiveness of the method conducted on different infrastructures and modes of use. The results indicate and average score around 98% with ISCX database and Random Forest technique and 96% with the testing database. The principal findings obtained suggests the feasibility of using the Random Forests technique to solve intrusion recognition problems in computer networks.

[5]. Rajni Tewatia and Asha Mishra discussed many approaches to implement the intrusion detection system. To improve the performance of an IDS these approaches may be used in combination to build a hybrid IDS so that benefits of two or more approaches may be combined. The proposed hybrid approach gives better performance over individual approaches. The data instances include two clusters: intrusive cluster and normal cluster through the clustering algorithm and after that apply soft computing Approach to make system adaptive and automated because soft computing is a supervised approach that needs training of data. With the help of any of the above discussed techniques, algorithms may be designed so that network becomes safe and secure.

[6]. Shadi I. Abudalfa et al, focuses on developing intrusion detection system that uses supervised learning technique. It was evaluated the performance selected classifiers by using five evaluation metrics: confusion matrix analysis, Classification Accuracy, Precision, Recall and F1-Score. The CSE-CIC-IDS2018 dataset is used for training and testing data. The used dataset is publicly available for researchers. It is

collected through a collaborative project achieved by the Communications Security Establishment (CSE) and The Canadian Institute for Cybersecurity (CIC). This dataset includes a detailed description of intrusions along with abstract distribution models for applications, protocols, or lower-level network entities. The dataset includes seven attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. The used attacking infrastructure includes 50 machines and the victim organization has 5 departments includes 420 PCs and 30 servers. To make the presented NIDS user friendly, a web application using python 3 with Django framework was also developed. The experimental results show that the highest accuracy is about 96.974% with using Decision Tree (DT) technique.

[7]. Subhash Waskle et al proposed an approach to develop efficient IDS by using the principal component analysis (PCA) and the random forest classification algorithm. Where the PCA will help to organize the dataset by reducing the dimensionality of the dataset and the random forest will help in classification. The intrusion detection system works for the improvement of the system, which is affected by the intruders. This system can do the detection of the intruders. The proposed system tries to eliminate the existing problems related to the previous work. The proposed system consists of the two methods that are principal component analysis, and the other one is the random forest. The principal component analysis is used for the reduction of the dimension of the dataset; by this method, the dataset quality will be improved as the dataset may contain the correct attributes. After this, the random forest algorithm will be applied for the detection of the intruders, which provide both the detection rate and the false alarm rate in an improved manner as compared to SVM.

[8]. Yogesh and Dr. G. Suresh Reddy , conducted experiments to determine which model using NSL-KDD dataset could attain classification and accuracy. The classification algorithms are used for analyzing NSL Dataset with Attributes. The classification used in this project are support SVM, RFC, K Neighbors Classifier, Logistic Regression, Naive bayes. NSL is a data set designed to alleviate a few of the underlying issues in the old version data set. Assume it could be used as a useful standard set of data for comparing different intrusion detection methods by researchers due to a scarcity of publicly available network-based IDS data sets. For models that use the Random Forest Classifier,feature selection and cleaning are crucial. The accuracy of results using Random Forest is around 0.997, which is significantly greater than the 0.9067 accuracy of Naive Bayes. When compared to the other algorithms, the accuracy of the results produced by Naive Bayes is lower. The results show that the Random Forest classifier outperforms the Naive Bayes classifier in terms of ability and accuracy. In comparison to Naive bayes, the random forest takes less time to train and test the dataset.

II. SYSTEM DESIGN

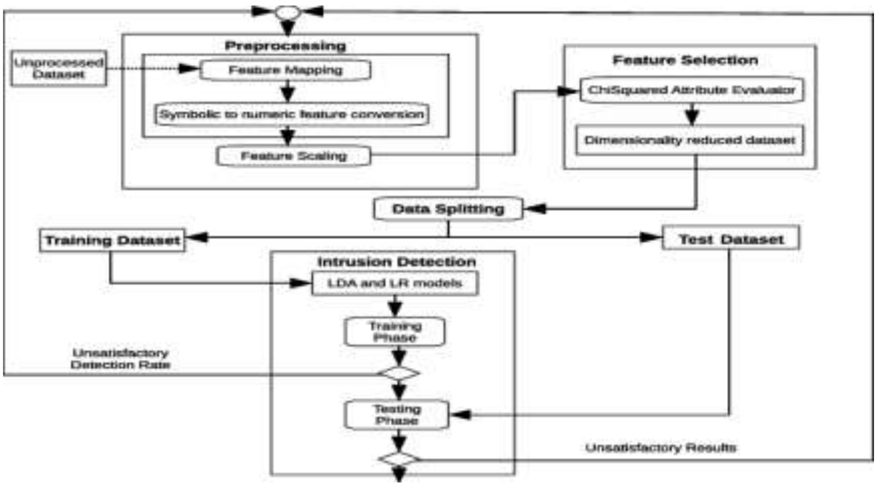


Fig. 1. THREAT DETECTION USING RF ALGORITHM

Fig.1. depicts the process in threat detection system.The threat detection system performs the following processes

- **Data Preprocessing**

The first technique on this project is to preprocess the data.In this phase, dataset reading and attack type feature adding is done where the attack type has five different values such as DoS, normal, probe, R2L,U2R. Next, find the missing values of all the features, if not found move to next process.

- **Feature Mapping**

Apply feature mapping to all the features and remove all the unimportant features before modeling. The tool used in this project is jupyter. Jupyter is an open-source web application used for data cleaning and transformation, machine learning and much more.

- **Modeling**

In modeling, first dataset splitting has done. The collected data is split in two: a training and a testing dataset. First collect training dataset and delete the unwanted data called data cleaning.

After data cleaning,feature extraction is applied to the dataset for dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.

A characteristic of these large datasets is a big wide variety of variables that require plenty of computing sources to process based on this it classify patterns and testing can be performed based on the classifier.

The attack can be detected by applying machine learning classification algorithms such as Naive Bayes, Decision Tree, SVM and Logistic Regression to create distinct models by dividing into testing and training sets.

- **Analyzing Accuracy**

The next process is to analyze the training and testing accuracy of each model and compare the results to build the predictive model.From the analysis of the each model, it shows that Random Forest yields the better accuracy.

III. IMPLEMENTATION

Random forest algorithm:

Random Forest is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of over fitting to their training set.

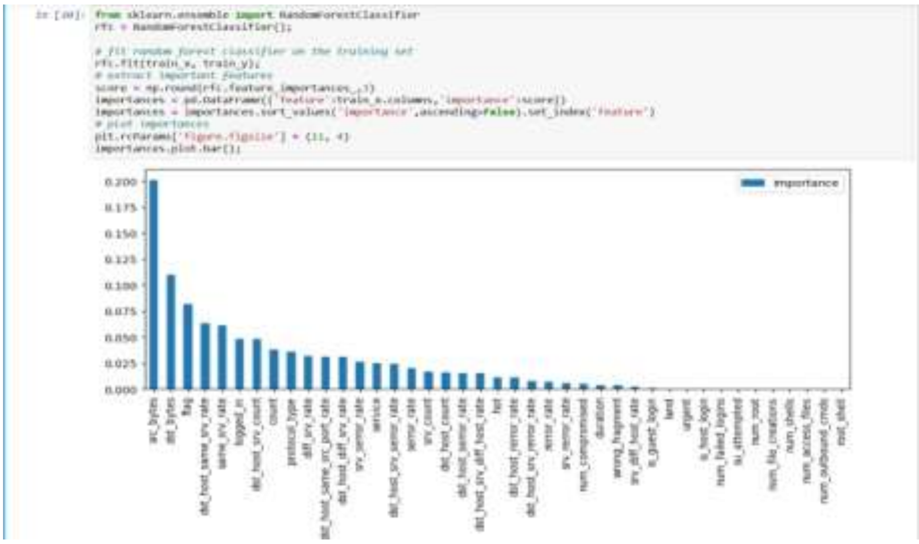
Importance Of RF Algorithm:

Random forest (RF) is an ensemble classifier used to improve the accuracy. Random forest consists of many decision trees. Random forest has low classification error compared to other traditional classification algorithms. Number of trees, minimum node size and number of features used for splitting each node.

When constructing individual trees in random forest, randomization is applied to select the best node to split on. This value is equal to √A, where A is no. of attributes in the data set. However, RF will generate many noisy trees, which affect accuracy and wrong decision for new sample. Below are estimated accuracy of detection rate for several attacks.

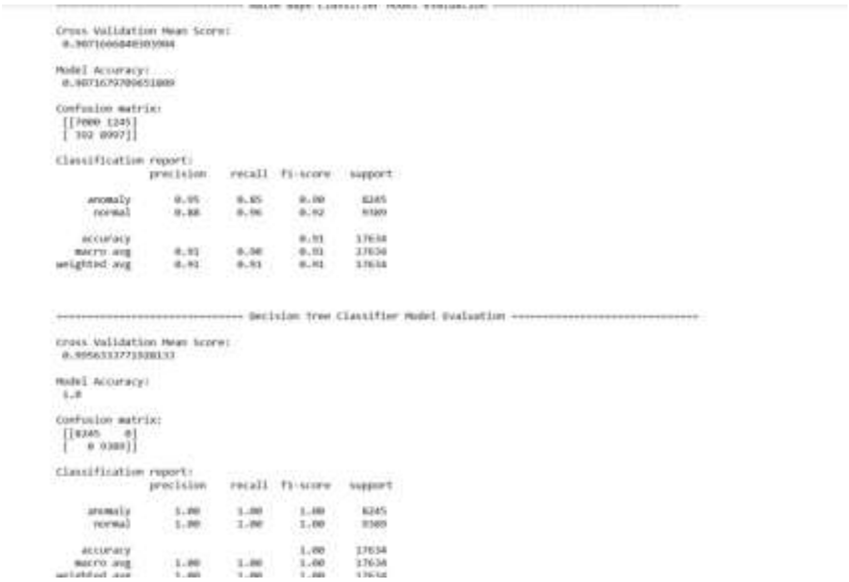
Algorithm: Random forest modeling for network IDS
Input: KDD99 dataset
Output: Classification of different type of attacks
Step 1: Load the dataset
Step 2: Apply pre-processing technique Discretization
Step 3: Cluster the dataset into four datasets.
Step 4: Partition the data set into training and test
Step 5: Select the best set features using feature subset selection measure Symmetrical uncertainty (SU) Symmetrical uncertainty compensates information gain
$SU(X, Y) = 2[I G(X/Y)/H(X)H(Y)]$
Step 6: Data set is given to Random forest for training
Step 7: The test data set is then fed to random forest for classification
Step 8: Calculate accuracy score, recall, precision, f-measure.

IV. RESULT



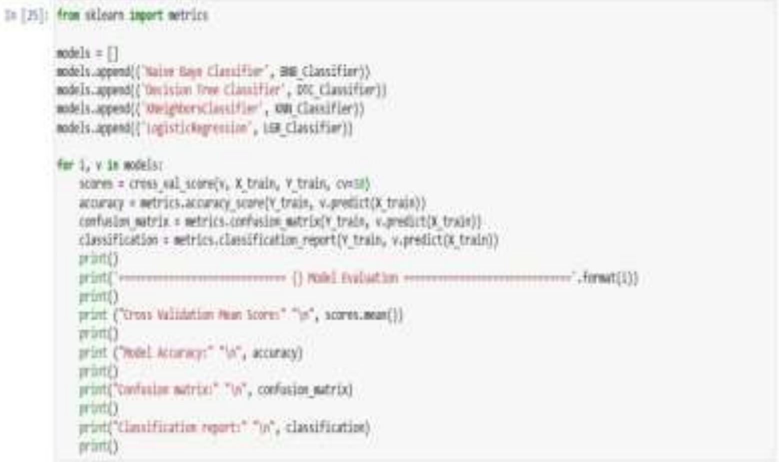
```
In [30]: from sklearn.ensemble import RandomForestClassifier
         rfc = RandomForestClassifier();

         # fit random forest classifier on the training set
         rfc.fit(train_x, train_y);
         # extract important features
         score = np.round(rfc.feature_importances_,3)
         importances = pd.DataFrame({'feature':train_x.columns,'importance':score})
         importances = importances.sort_values('importance',ascending=False).set_index('feature')
         # plot importances
         plt.rcParams['figure.figsize'] = (11, 4)
         importances.plot.bar();
```

Fig.2. Feature mapping



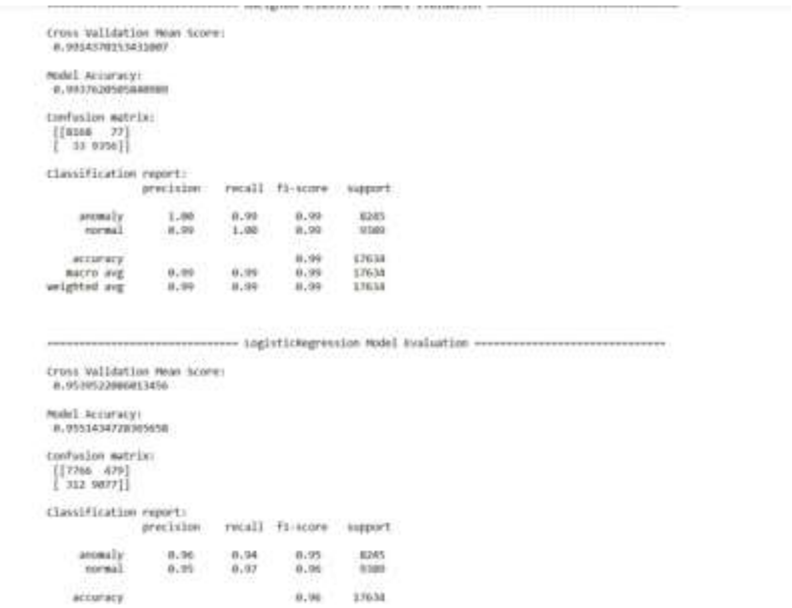
```
------------------------------ Naive Baye Classifier Model Evaluation ------------------------------

Cross Validation Mean Score:
 0.9071666848303984

Model Accuracy:
 0.9071679700651809

Confusion matrix:
 [[7000 1245]
 [ 392 8997]]

Classification report:
               precision    recall  f1-score   support

     anomaly       0.95      0.85      0.90      8245
      normal       0.88      0.96      0.92      9389

    accuracy                           0.91     17634
   macro avg       0.91      0.90      0.91     17634
weighted avg       0.91      0.91      0.91     17634


------------------------------ Decision Tree Classifier Model Evaluation ------------------------------

Cross Validation Mean Score:
 0.9956333771938133

Model Accuracy:
 1.0

Confusion matrix:
 [[8245    0]
 [   0 9389]]

Classification report:
               precision    recall  f1-score   support

     anomaly       1.00      1.00      1.00      8245
      normal       1.00      1.00      1.00      9389

    accuracy                           1.00     17634
   macro avg       1.00      1.00      1.00     17634
weighted avg       1.00      1.00      1.00     17634
```

Fig 3.Decision Tree



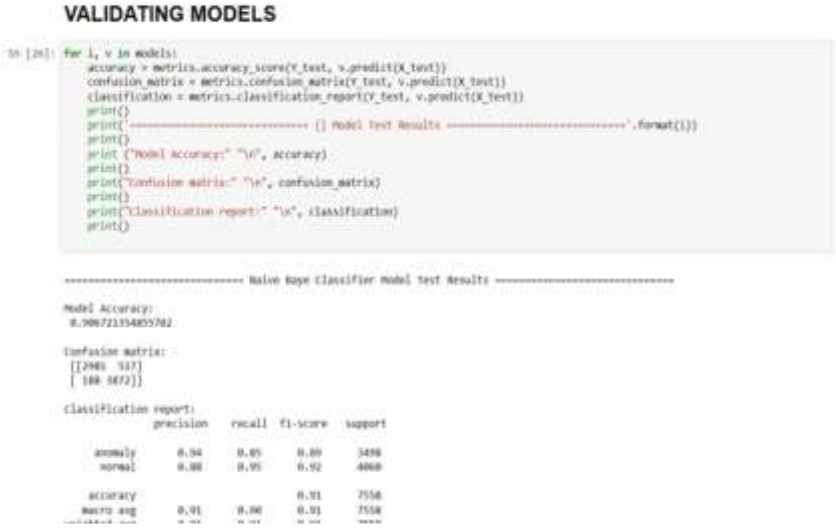
```
In [25]: from sklearn import metrics

         models = []
         models.append(('Naive Baye Classifier', BNB_Classifier))
         models.append(('Decision Tree Classifier', DTC_Classifier))
         models.append(('KNeighborsClassifier', KNN_Classifier))
         models.append(('LogisticRegression', LGR_Classifier))

         for i, v in models:
             scores = cross_val_score(v, X_train, Y_train, cv=10)
             accuracy = metrics.accuracy_score(Y_train, v.predict(X_train))
             confusion_matrix = metrics.confusion_matrix(Y_train, v.predict(X_train))
             classification = metrics.classification_report(Y_train, v.predict(X_train))
             print()
             print('------------------------------ {} Model Evaluation ------------------------------'.format(i))
             print()
             print ("Cross Validation Mean Score:" "\n", scores.mean())
             print()
             print ("Model Accuracy:" "\n", accuracy)
             print()
             print("Confusion matrix:" "\n", confusion_matrix)
             print()
             print("Classification report:" "\n", classification)
             print()
```

Fig 4. Classifiers



```
------------------------------ [illegible] Model Evaluation ------------------------------

Cross Validation Mean Score:
 0.9954370153431007

Model Accuracy:
 0.9937620505840880

Confusion matrix:
 [[8168   77]
 [   33 9356]]

Classification report:
              precision    recall  f1-score   support

     anomaly       1.00      0.99      0.99      8245
      normal       0.99      1.00      0.99      9389

    accuracy                           0.99     17634
   macro avg       0.99      0.99      0.99     17634
weighted avg       0.99      0.99      0.99     17634


------------------------------ LogisticRegression Model Evaluation ------------------------------

Cross Validation Mean Score:
 0.9539522000013456

Model Accuracy:
 0.9551434720305658

Confusion matrix:
 [[7766  479]
 [ 312 9077]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.96      0.94      0.95      8245
      normal       0.95      0.97      0.96      9389

    accuracy                           0.96     17634
```

Fig 5. Logistic Regression Model Evaluation



VALIDATING MODELS

```
In [26]: for i, v in models:
             accuracy = metrics.accuracy_score(Y_test, v.predict(X_test))
             confusion_matrix = metrics.confusion_matrix(Y_test, v.predict(X_test))
             classification = metrics.classification_report(Y_test, v.predict(X_test))
             print()
             print('============================== {} Model Test Results ==============================='.format(i))
             print()
             print ("Model Accuracy:" "\n", accuracy)
             print()
             print("Confusion matrix:" "\n", confusion_matrix)
             print()
             print("Classification report:" "\n", classification)
             print()
```

```
============================== Naive Baye Classifier Model Test Results ==============================

Model Accuracy:
 0.9067213354855782

Confusion matrix:
 [[2981  517]
 [ 188 3872]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.94      0.85      0.89      3498
      normal       0.88      0.95      0.92      4060

    accuracy                           0.91      7558
   macro avg       0.91      0.90      0.91      7558
```

Fig 6. Model Evaluation

V. ADVANTAGES

The advantages of using RF Algorithm are

1) Generated forests can be saved for future reference.

2) Random forest overcomes the problem over fitting.

3) In RF accuracy and variable importance is automatically generated

VI. CONCLUSION

This project deals the Random Forest (RF) algorithm to detect four types of attack like DOS, probe, U2R and R2L. Feature selection is applied on the data set to reduce dimensionality and to remove redundant and irrelevant features. We applied symmetrical uncertainty of attributes which overcomes the problems of information gain. The proposed approach is evaluated using KDD 99 data set. We compared our random forest modelling with Decision trees algorithm classier in terms of accuracy score, recall, precision, f-measure, I will try to apply evolutionary computation as a feature selection measure to further improve accuracy of the classifier

REFERENCES

[1]. Ajith, G. Crina, C.Yuehui (2001), " Cyber Security and the Evolution of Intrusion Detection Systems ", Information Management and Computer Security 9(4).

[2]. H. Arafat(2001), “ A new model for monitoring intrusion based on Petri Nets ”, Information Management and Computer Security 9(4),pp.175-182.

[3]. Fawaz Mokbal, Wang Dan, Musa Osman, Yang Ping, Saeed Alsamhi(2022), “ An Efficient Intrusion Detection Framework Based on Embedding Feature Selection and Ensemble Learning Technique ”, The International Arab Journal of Information Technology, Vol. 19, No. 2,pp.237-239

[4]. Heitor Scalco Neto, Wilian Soares Lacerda, Rafael Verao Francozo(2022), “ Random Forests for Online Intrusion Detection in Computer Networks ”, Journal of Computer Science, vol.17,Issue 10, pp.905-913.

[5]. Rajni Tewatia, Asha Mishra(2015), “Introduction To Intrusion Detection System: Review ”, International Journal Of Scientific & Technology Research, Volume 4, Issue 05, pp.219-222.

[6]. Shadi I. Abudalfa, Ekhlas S. Isleem, Marah J. Elshaikh Khalil, Rewaa A. Dalloul and Shaimaa M. Iqtefan (2022,), “ Evaluating Performance of Supervised Learning Techniques for Developing Real-Time Intrusion Detection System ”, International Journal of Engineering and Information Systems (IJEAIS), Vol. 6, Issue 2, pp.103-105.

[7]. Subhash Waskle, Lokesh Parashar, Upendra Singh(2022), “ Intrusion Detection System Using PCA with Random Forest Approach ”, International Conference on Electronics and Sustainable Communication Systems (ICESC 2020), pp.803-805.

[8]. B. Yogesh, Dr. G. Suresh Reddy(2022), “ Intrusion detection System using Random Forest Approach ”, Turkish Journal of Computer and Mathematics Education, Vol.13, No.02, pp.730-731