

**DMA COURSE PROJECT**

**[18ECSC301]**

**Report on**

**NETWORK TRAFFIC ANALYSIS**

**[ 5DMACPO8]**

**Submitted By: -**

|  |  |  |
| --- | --- | --- |
| **Name** | **USN** | **R.NO** |
| Aayush Rajwade | 01FE18BCS005 | 105 |
| Aman Kumar | 01FE18BCS029 | 129 |
| Annapoorna pattar | 01FE18BCS040 | 140 |
| Ayush Utsav | 01FE18BCS059 | 159 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE NO** |
| 1. | Introduction | 4 |
| 2. | Problem Statement | 5 |
| 3. | Exploratory Data Analysis | 5-9 |
| 4. | Data Preprocessing | 10-12 |
| 5. | ML Models | 13-15 |
| 6. | Optimal Model | 16-18 |
| 7. | Result | 19 |
| 8. | References | 20 |

1. **INTRODUCTION**

Advancement in technology have provided a lot of data for analysis. The 21st century also known as digital era because of rapid increase in the amount of internet devices connected together and therefore comes the important task of Network traffic classification which is classification of network traffic into appropriate classes.

Network traffic classification is vital for many applications such as QOS (Quality of service control), resource allocation and the most important malware detection particularly from security point of view and increasing cyber Attacks.

These days researchers are employing learning methods with flow statistical features since they do not rely on port number and payload itself.

1. **PROBLEM STATEMENT**

**“**For the given NetML dataset, prediction of top-level annotation for binary classification of malware.**”**

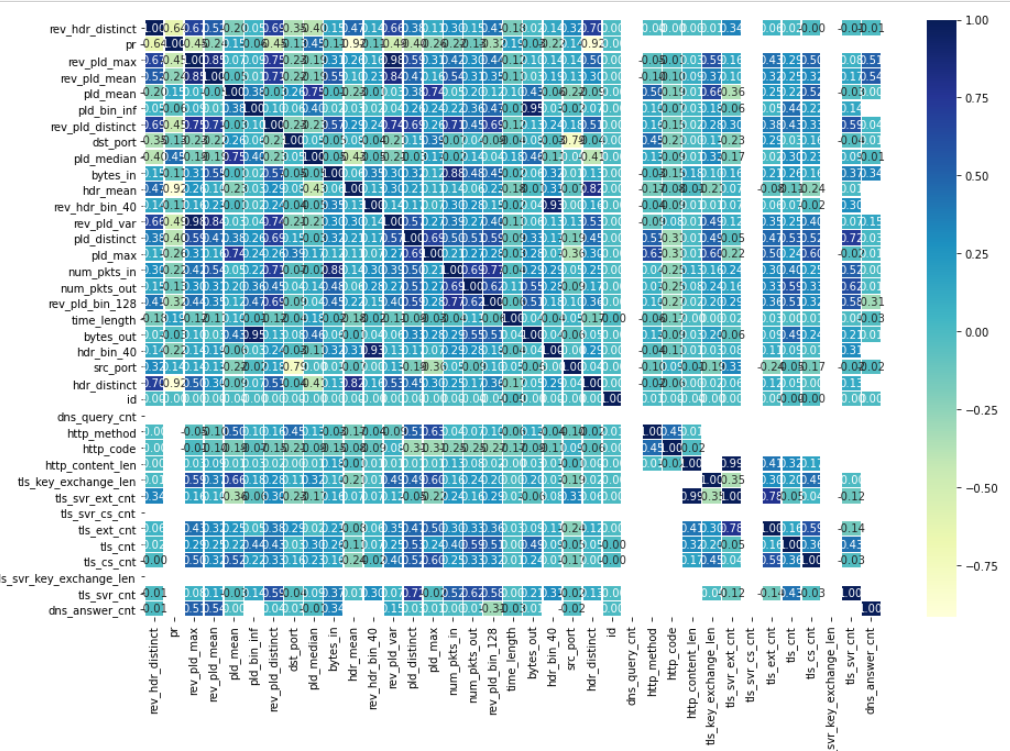
**Description:-**

According to the problem statement the expectations are the prediction of top-level annotation for binary classification of malware for the given NetML dataset.

Usually network traffic analysis uses port-based approaches but because of advancement in science and application development newer application uses dynamic port allocation instead of using standard registered port numbers.

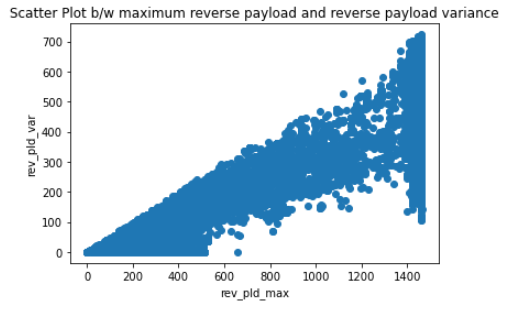
So in this scenario ML methods with flow statistical features may be employed, since they do not rely on port numbers or payload itself.

1. **EXPLORATORY DATA ANALYSIS**
   1. **Dataset description:-**
      1. The training dataset consists of 387268 rows and 62 columns.
      2. These 62 columns are the summation of 4 set of features namely-metadata, TLS, DNS and HTTP.
      3. TLS, DNS and HTTP are protocol dependent features whereas metadata features are protocol independent.
      4. Metadata set of features have 32 columns, rest 30 column belongs to protocol-based set of features.
      5. Each row is identified by unique id whose o/p label contains binary categorical values: malware and benign, stored in a separate o/p file.
      6. We have 387268 unique ids same as the number of rows.

****

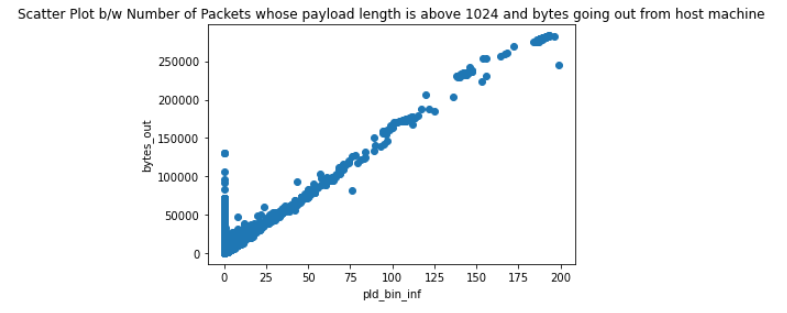
**Figure: 3.1**

**Figure: 3.1** shows the Heat Map that is created for finding strong correlation between input columns. Since the domain information require expertise in this field so correlation matrix is considered to be the most apt one.

****

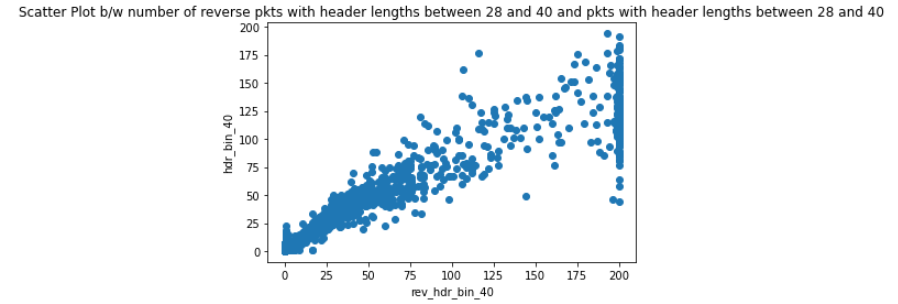
**Figure: 3.2**

**Figure: 3.2** shows the Scatter plot with correlation value of 0.97 between reverse payload variance and maximum reverse payload value. In spite of a large correlation between them there is no logical correlation.



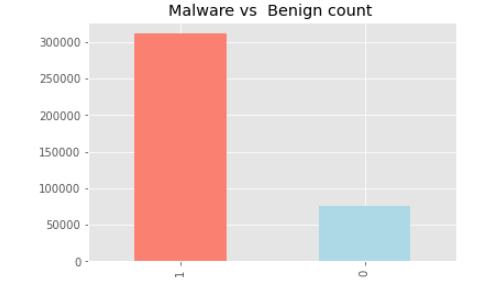
**Figure: 3.3**

**Figure: 3.3** shows the Scatter plot with the correlation value of 0.94 between number of bytes out from the host machine and number of packets whose payload length are above 1024. This implies that the host machine might be infected by malware since it is sending payload length of more than 1024 as well as data which is represented here in terms of bytes out is being send from host to attacker or a server.



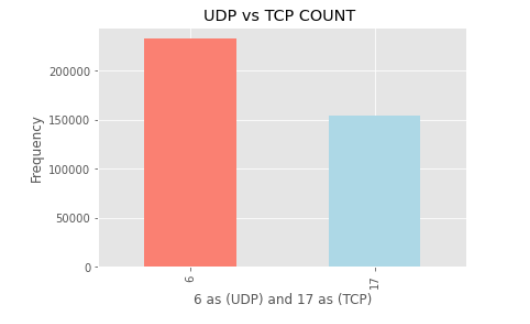
**Figure: 3.4**

**Figure: 3.4** shows the Scatter plot with the correlation value of 0.9286 between number of packets between header length 28 and 40 and reverse of number of packets between header length 28 and 40. This implies that there is a connection and reverse connection between host and attacker machine, potentially indicating malware.



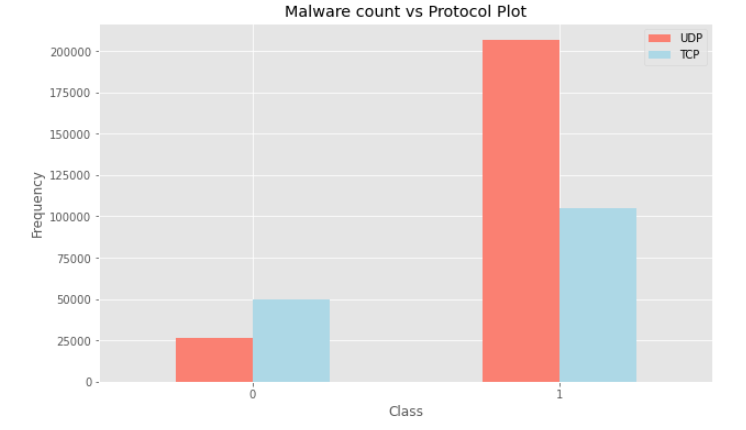
**Figure: 3.5**

**Figure: 3.5** shows the graph between the Malware versus Benign count that shows the Malwares count compared to the Benign count present in the dataset.



**Figure: 3.6**

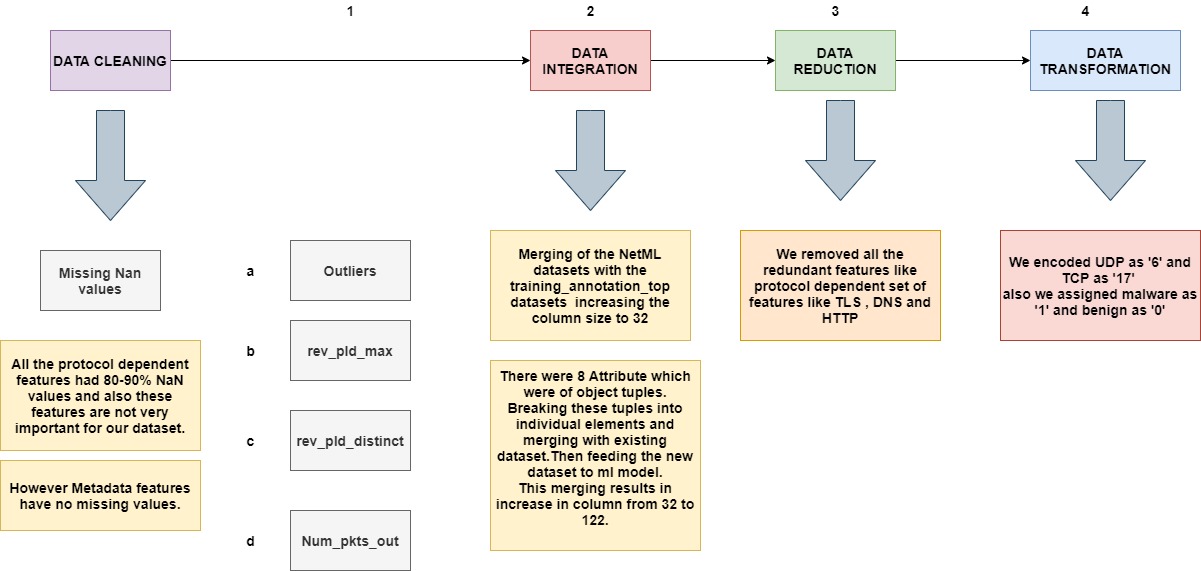
**Figure: 3.6** shows the graph for UDP count versus TCP count. From the graph it is inferred that UDP count is greater in number as compared to TCP count.



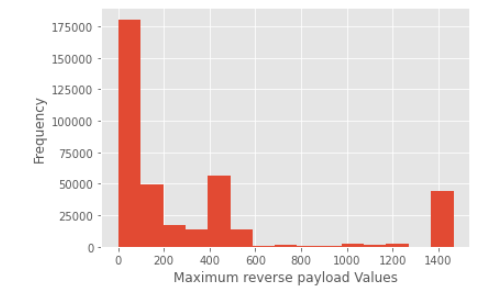
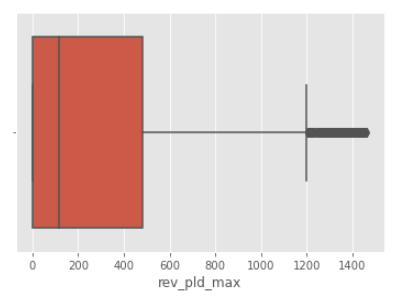
**Figure: 3.7**

**Figure: 3.7** shows the graph between the output predicting class count of Malware/Benign versus the protocol attribute. The Malware frequency is represented on y-axis and Malware is represented on x-axis as 1 and Benign as 0. From the graph it is inferred that most of the network traffic flow are using UDP connection and they may be using applications which uses UDP connection and since UDP connection is not the secure therefore there is a lot of chance of Malware attacks.

1. **DATA PREPROCESSING**

****

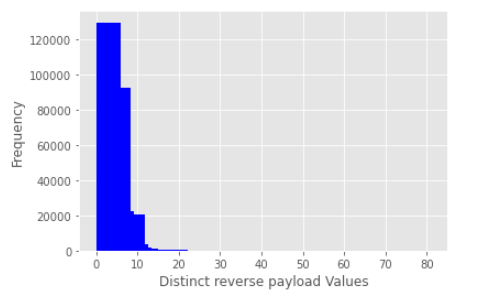
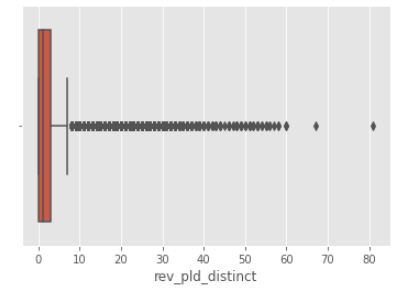
* 1. **Data Cleaning**
     1. **Missing Values:-** The NetML dataset had 4 types of flow features that is categorized into 2 main features metadata features and protocol-based features. So protocol-based features had 80-90 % missing values which was significantly higher so these features were dropped.
     2. **Outlier Analysis:-**



**Figure: 4.1 Figure: 4.2**

**Figure: 4.1** shows the histogram to see the skewness and outliers of maximum reverse payload feature.

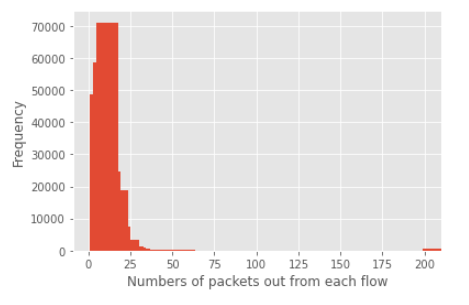
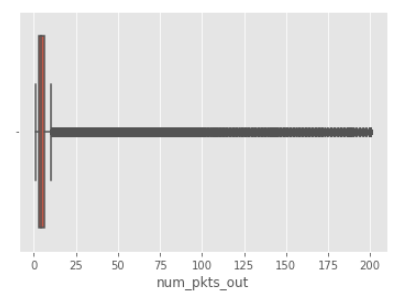
**Figure: 4.2** shows the Box plot of the maximum reverse payload feature for the analysis of the outliers present in it and also to determine the skewness.



**Figure: 4.3 Figure: 4.4**

**Figure: 4.3** shows the histogram to see the skewness and outliers of Distinct reverse payload feature.

**Figure: 4.4** shows the Box plot of the Distinct reverse payload feature for the analysis of the outliers present in it and also to determine the skewness.



**Figure: 4.5 Figure: 4.6**

**Figure: 4.5** shows the histogram to see the skewness and outliers of number of packets out feature.

**Figure: 4.6** shows the Box plot of number of packets out feature for the analysis of outliers present in it and also to determine the skewness.

* 1. **Data Integration**
     1. Merging of the NetML dataset with the training\_annotation\_top dataset increasing the column size to 32.
     2. Breaking the tuples that were in the form of object tuples into individual elements and merging with existing dataset and feeding the dataset to the ML model. The merging results in increase in the column from 32 to 122.
  2. **Data Reduction**
     1. Removal of all the redundant features like protocol

dependent set of features (TCP, DNS and HTTP).

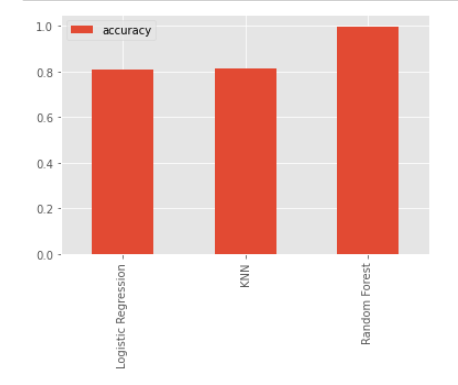
* 1. **Data Transformation**
     1. Encoding the Protocol feature UDP as ‘6’ and TCP as ‘17’.
     2. Encoding Malware as ‘1’ and Benign as ‘0’.

1. **ML MODELS**

The ML model used for Binary Classification of the NetML dataset are as follows:-

* 1. Random Forest Classifiers
  2. Logistic Regression
  3. KNN Classification

**Model comparison**



**Figure: 5.1**

**Figure: 5.1** shows the comparison between the three used ML models. From the graph it is inferred that Random Forest model gives the highest accuracy among all other used models.

Accuracy scores of the models are shown below:-

1. Random Forest Classifiers => 99.77%
2. Logistic Regression => 81.05%
3. KNN Classification => 81.53%

**Bias-Variance Tradeoff:-**

1. Random Forest

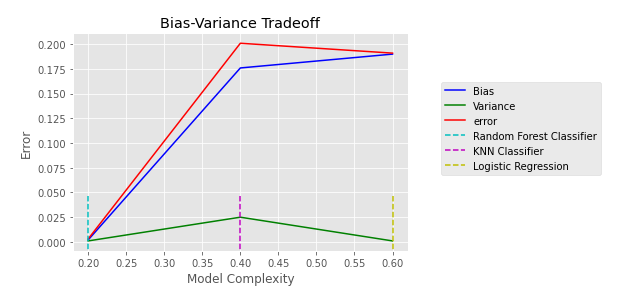


1. Logistic Regression



1. KNN Classification





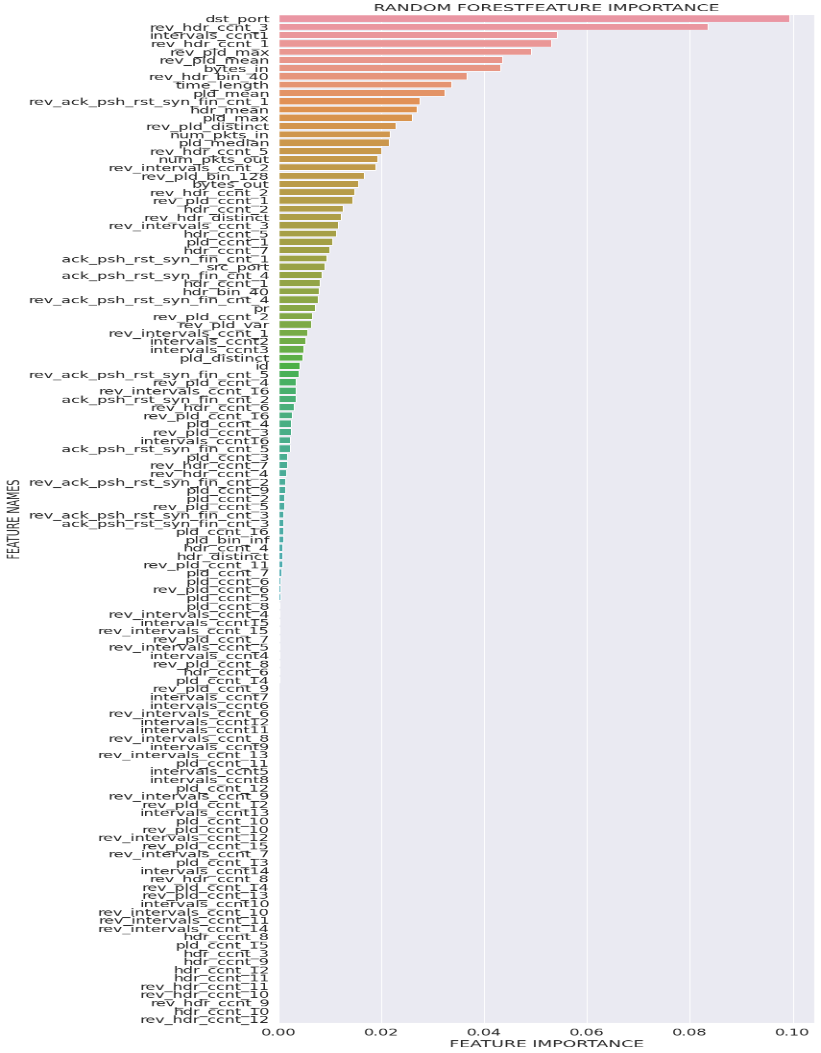
**Figure: 5.2**

**Figure: 5.2** shows the Bias-Variance Tradeoff comparison between the three models that were considered for Binary Classification.

1. The comparison clearly shows that Random Forest has the least bias as well as the least variance among the three models which suggests that the model that needs to be chosen is Random Forest.
2. So the chosen model needs to be refined with hyperparameters and best features.
3. **OPTIMAL MODEL**

The optimal model selected that is used for the Binary Classification is “Random Forest”.

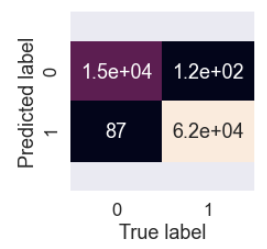
* 1. **Reason for selecting RF Model**
     1. The accuracy score is the highest in RF Classifier model.
     2. The Bias-Variance tradeoff clearly showed the least error of 0.3%.
     3. The Domain Knowledge required for the feature extraction is difficult in short time so this model is the best for such situations.
  2. **Post Model Analysis**
     1. **Important parameters/features selected**

****

**Figure: 6.1**

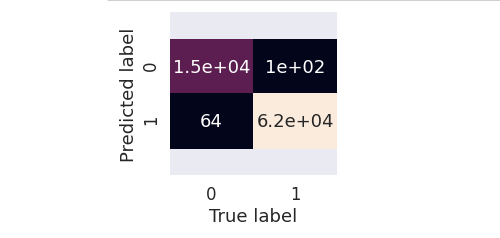
**Figure: 6.1** shows the most important features graphically in ascending order. The tuned model consists of top 10 features for predictions and the rest less significant features are discarded.

* + 1. **Confusion Matrix**

****

**Figure: 6.2**

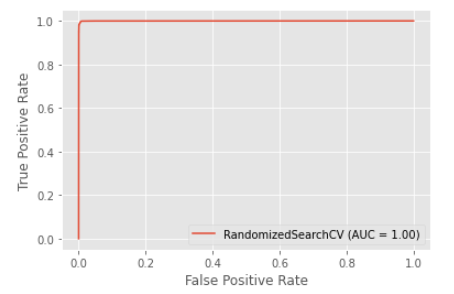
**Figure: 6.2** shows the Confusion Matrix of RF Classifier model before hyperparametric tuning. The Confusion Matrix indicates 207 incorrect predictions whereas 77,000 correct predictions with model accuracy score of 99.7792%.



**Figure: 6.3**

**Figure: 6.3** shows the Confusion Matrix after hyperparameter tuning and selecting important features. The Confusion Matrix shows 164 incorrect predictions whereas 77,000 correct predictions with the accuracy score increasing to 99.7856%.

* 1. **ROC Curve**

****

**Figure: 6.4**

**Figure: 6.4** shows the ROC (Receiver Operating Characterstics) curve of the tuned RF Classifier model. Since **perfect classifier**is a combination of two straight lines both moving away from the baseline towards the top-left corner. It can be inferred that the tuned model is perfect classifier model.

1. **RESULT**
   1. The final submission file generated from the tuned model gave the TPR (True Positive Rate) value of 0.99931 and FAR (False Alarm Rate) value of 0.00590.
   2. The overall accuracy obtained is 0.99341 on test dataset.
   3. The final leaderboard ranking is 6th position as shown in Fig:7.1 in the NetML top challenge.



**Figure: 7.1**

1. **REFERENCES**

* [https://eval.ai/web/challenges/challenge-page/526/overview](https://arxiv.org/pdf/2004.13006v1.pdf)
* <https://arxiv.org/pdf/2004.13006v1.pdf> [literature survey]
* [https://scikit learn.org/stable/auto\_examples/classification/plot\_classifier\_comparison.html](https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)
* [https://pandas.pydata.org/docs/user\_guide/index.html#user-guide](https://pandas.pydata.org/docs/user_guide/index.html)
* [What Is ROC Curve in Machine Learning using Python? ROC Curve Example (intellipaat.com)](https://intellipaat.com/blog/roc-curve-in-machine-learning/#:~:text=Introduction%20to%20ROC%20Curve%20in%20Machine%20Learning%20Let%E2%80%99s,to%20measure%20the%20accuracy%20of%20a%20classification%20model.)