Pune Institute of Computer Technology
Dhankawadi, Pune

A MINI PROJECT REPORT
ON

MALWARE CLASSIFICATION

SUBMITTED BY

Shrut Shah
Roll No. 41269
Shubhankar Gaikwad
Roll No. 41270
Under the guidance of
Prof. S. N. Girme

DEPARTMENT OF COMPUTER ENGINEERING

Academic Year 2020-21

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Mini Project report entitled

“MALWARE CLASSIFICATION”

Submitted by
Shrut Shah Roll No. 41269

Shubhankar Gaikwad Roll No. 41270

have satisfactorily completed a mini project for Lab Practices II: 'Data Mining and Warehousing' elective under the guidance of Prof. S. N. Girme towards the partial fulfillment of fourth year Computer Engineering Semester I, Academic Year 2020-21 of Savitribai Phule Pune University.

Prof. S. N. Girme
Internal Guide

Prof. M.S.Takalikar
Head
Department of Computer Engineering

Place:
Date:

Contents

List of Tables

List of Figures

1 INTRODUCTION

Cybersecurity has been an integral part of the information systems since the beginning of information revolution. The battle between code makers and code breakers is now at a critical stage. Today the amount of data in the world is manifold. Given the importance and power of data in our lives, it has now become need of the hour to protect our systems from malicious actors and softwares.

We have decided to work on malware classification topic for this reason. We have used the NSL-KDD dataset for this purpose. NSL-KDD is the benchmark for modern-day internet traffic and is used to detect attacks over the internet by many researchers. The NSL-KDD data set is not the first of its kind. The KDD cup was an International Knowledge Discovery and Data Mining Tools Competition. In 1999, this competition was held with the goal of collecting traffic records. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between “bad” connections, called intrusions or attacks, and “good” normal connections. As a result of this competition, a mass amount of internet traffic records were collected and bundled into a data set called the KDD’99, and from this, the NSL-KDD data set was brought into existence, as a revised, cleaned-up version of the KDD’99 from the University of New Brunswick.

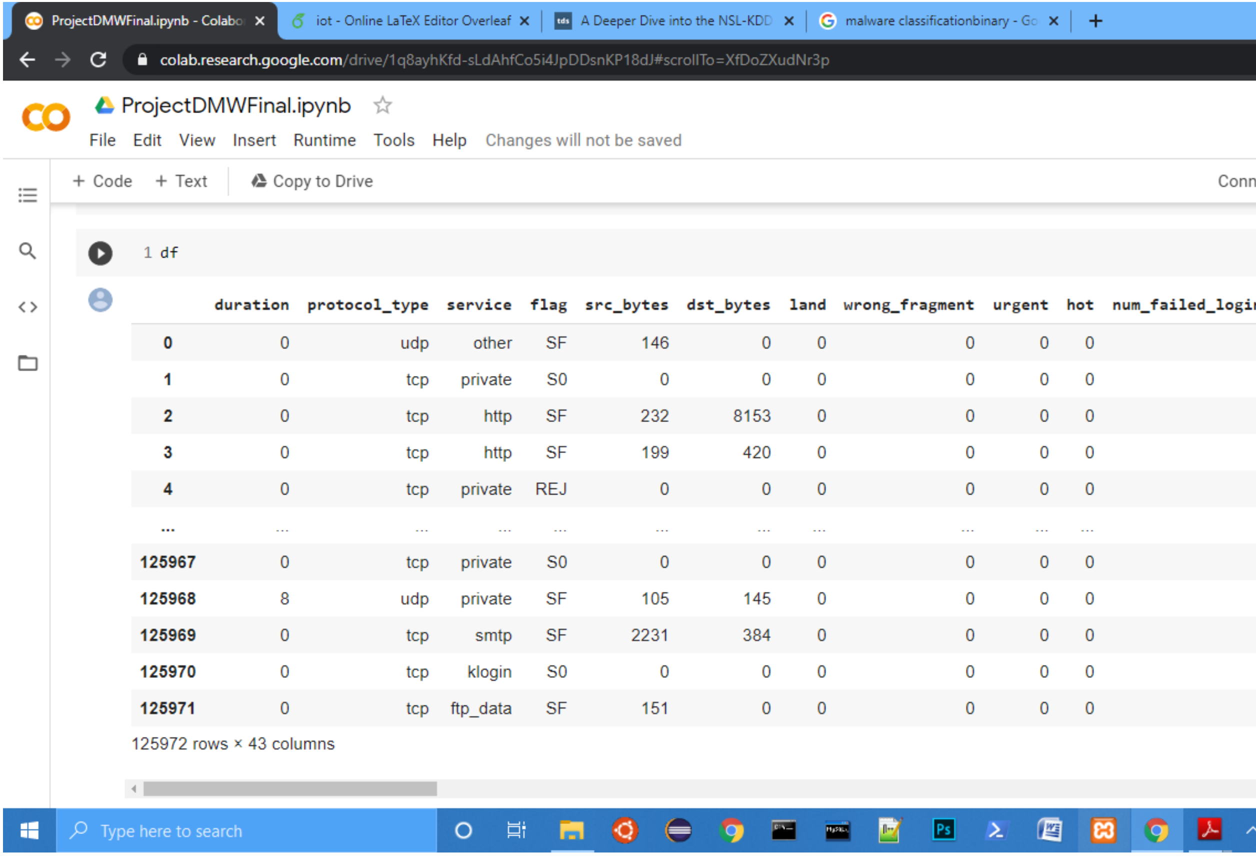
2 PROBLEM DEFINITION AND SCOPE

2.1 Problem Definition

Mini project on classification: Consider a labeled dataset belonging to an application domain. Apply suitable data preprocessing steps such as handling of null values, data reduction, discretization. For prediction of class labels of given data instances, build classifier models using different techniques (minimum 3), analyze the confusion matrix and compare these models. Also apply cross validation while preparing the training and testing datasets. For Example: Health Care Domain for predicting disease

2.2 Topic domain - Malware classification

There are systems in place to protect your valuable information held in your computer or networks. These systems that detect malicious traffic inputs are called Intrusion Detection Systems (IDS) and are trained on internet traffic record data. The most common data set is the NSL-KDD, and is the benchmark for modern-day internet traffic. These data sets contain the records of the internet traffic seen by a simple intrusion detection network and are the ghosts of the traffic encountered by a real IDS and just the traces of its existence remains.


ProjectDMWFinal.ipynb - Colabo
iot - Online LaTeX Editor Overleaf
A Deeper Dive into the NSL-KDD
malware classificationbinary - Go
colab.research.google.com/drive/1q8ayhKfd-sLdAhfCo5i4JpDDsnKP18dJ#scrollTo=XfDoZXudNr3p
ProjectDMWFinal.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Changes will not be saved
+ Code
+ Text
Copy to Drive
Conn
Type here to search

```
df
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logi
0	0	udp	other	SF	146	0	0	0	0	0	
1	0	tcp	private	S0	0	0	0	0	0	0	
2	0	tcp	http	SF	232	8153	0	0	0	0	
3	0	tcp	http	SF	199	420	0	0	0	0	
4	0	tcp	private	REJ	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	
125967	0	tcp	private	S0	0	0	0	0	0	0	
125968	8	udp	private	SF	105	145	0	0	0	0	
125969	0	tcp	smtp	SF	2231	384	0	0	0	0	
125970	0	tcp	klogin	S0	0	0	0	0	0	0	
125971	0	tcp	ftp_data	SF	151	0	0	0	0	0	

125972 rows × 43 columns

Figure 1: NSL-KDD Dataset

3 DATASET - NSLKDD

3.1 Malware types

Within the data set exists 4 different classes of attacks: Denial of Service (DoS), Probe, User to Root(U2R), and Remote to Local (R2L). A brief description of each attack can be seen below:

- DoS is an attack that tries to shut down traffic flow to and from the target system. The IDS is flooded with an abnormal amount of traffic, which the system can't handle, and shuts down to protect itself. This prevents normal traffic from visiting a network. An example of this could be an online retailer getting flooded with online orders on a day with a big sale, and because the network can't handle all the requests, it will shut down preventing paying customers to purchase anything. This is the most common attack in the data set.
- Probe or surveillance is an attack that tries to get information from a network. The goal here is to act like a thief and steal important information, whether it be personal information about clients or banking information.
- U2R is an attack that starts off with a normal user account and tries to gain access to the system or network, as a super-user (root). The attacker attempts to exploit the vulnerabilities in a system to gain root privileges/access.
- R2L is an attack that tries to gain local access to a remote machine. An attacker does not have local access to the system/network, and tries to "hack" their way into the network.

3.2 Dataset description

The data set contains 43 features per record, with 41 of the features referring to the traffic input itself and the last two are labels (whether it is a normal or attack) and Score (the severity of the traffic input itself).

4 FEATURE ENGINEERING

4.1 Description of features

The features in a traffic record provide the information about the encounter with the traffic input by the IDS and can be broken down into four categories: Intrinsic, Content, Host-based, and Time-based. Below is a description of the different categories of features:

- Intrinsic features can be derived from the header of the packet without looking into the payload itself, and hold the basic information about the packet. This category contains features 1–9.

- Content features hold information about the original packets, as they are sent in multiple pieces rather than one. With this information, the system can access the payload. This category contains features 10–22.

- Time-based features hold the analysis of the traffic input over a two-second window and contains information like how many connections it attempted to make to the same host. These features are mostly counts and rates rather than information about the content of the traffic input. This category contains features 23–31.

- Host-based features are similar to Time-based features, except instead of analyzing over a 2-second window, it analyzes over a series of connections made (how many requests made to the same host over x-number of connections). These features are designed to access attacks, which span longer than a two-second window time-span. This category contains features 32–41.

4.2 Features used for classification

- Continuous features(Column 1,5,6) - Duration, Source bytes and Destination bytes

- Categorical features (Column 2,3,4) - Protocol type , Service and Flags

5 CLASSIFICATION ALGORITHMS

5.1 K-Nearest Neighbors

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all the training tuples are stored in an n-dimensional pattern space.When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k "nearest neighbors" of the unknown tuple.

5.2 Decision Trees

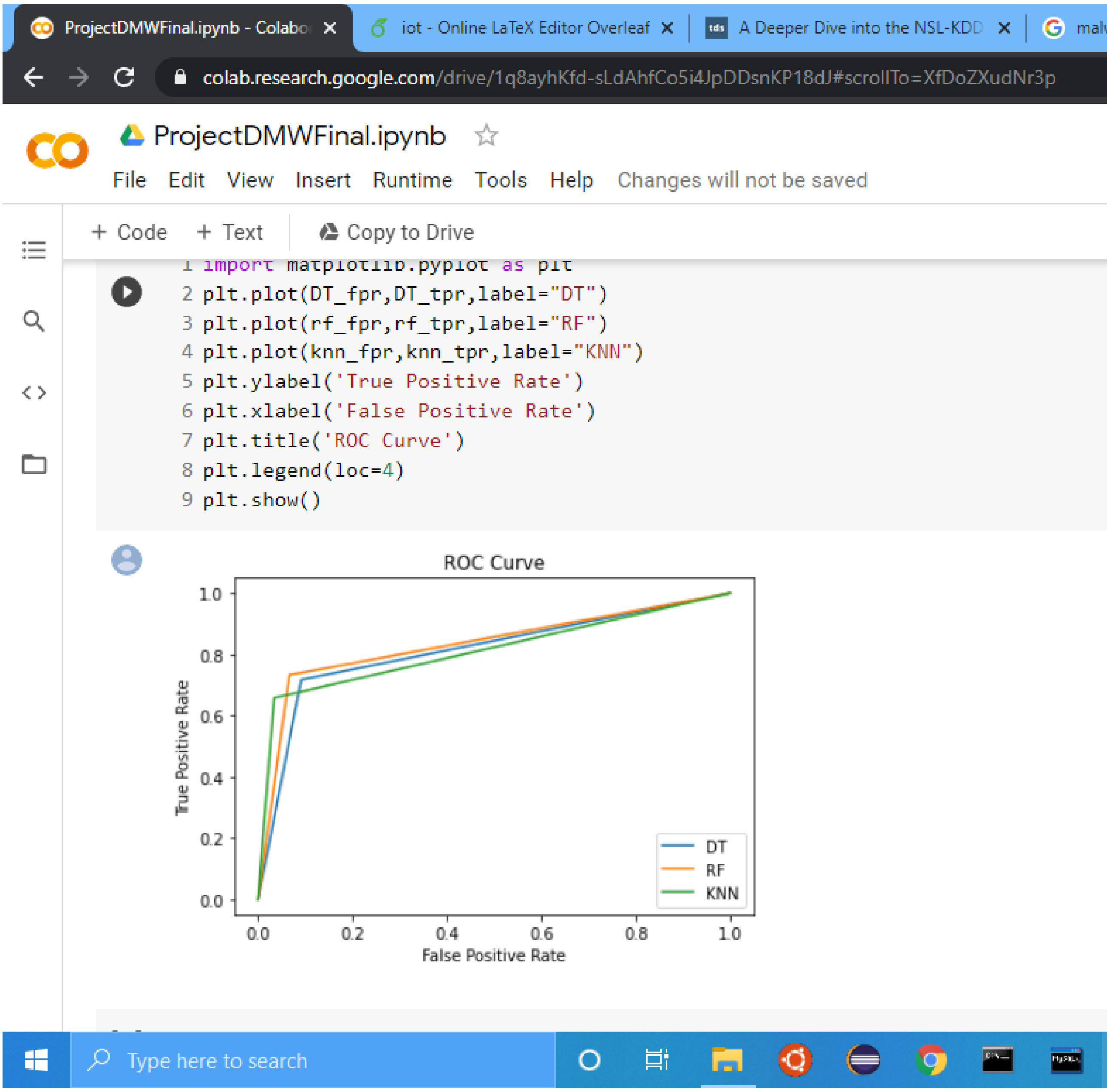
A supervised model of classification where leaf nodes contain labels and intermediate nodes are conditions. Repetitive checking is done against conditions to reach down to a label. Useful for signature based malware detection.
Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node.

5.3 Random Forests

It is similar to bagging. It creates large collection of de-correlated trees and then averages them out. Imagine that each of the classifiers in the ensemble is a decision tree classifier so that the collection of classifiers is a "forest." The individual decision trees are generated using a random selection of attributes at each node to determine the split. More formally, each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. During classification, each tree votes and the most popular class is returned.

6 Results

6.1 Implementation Results

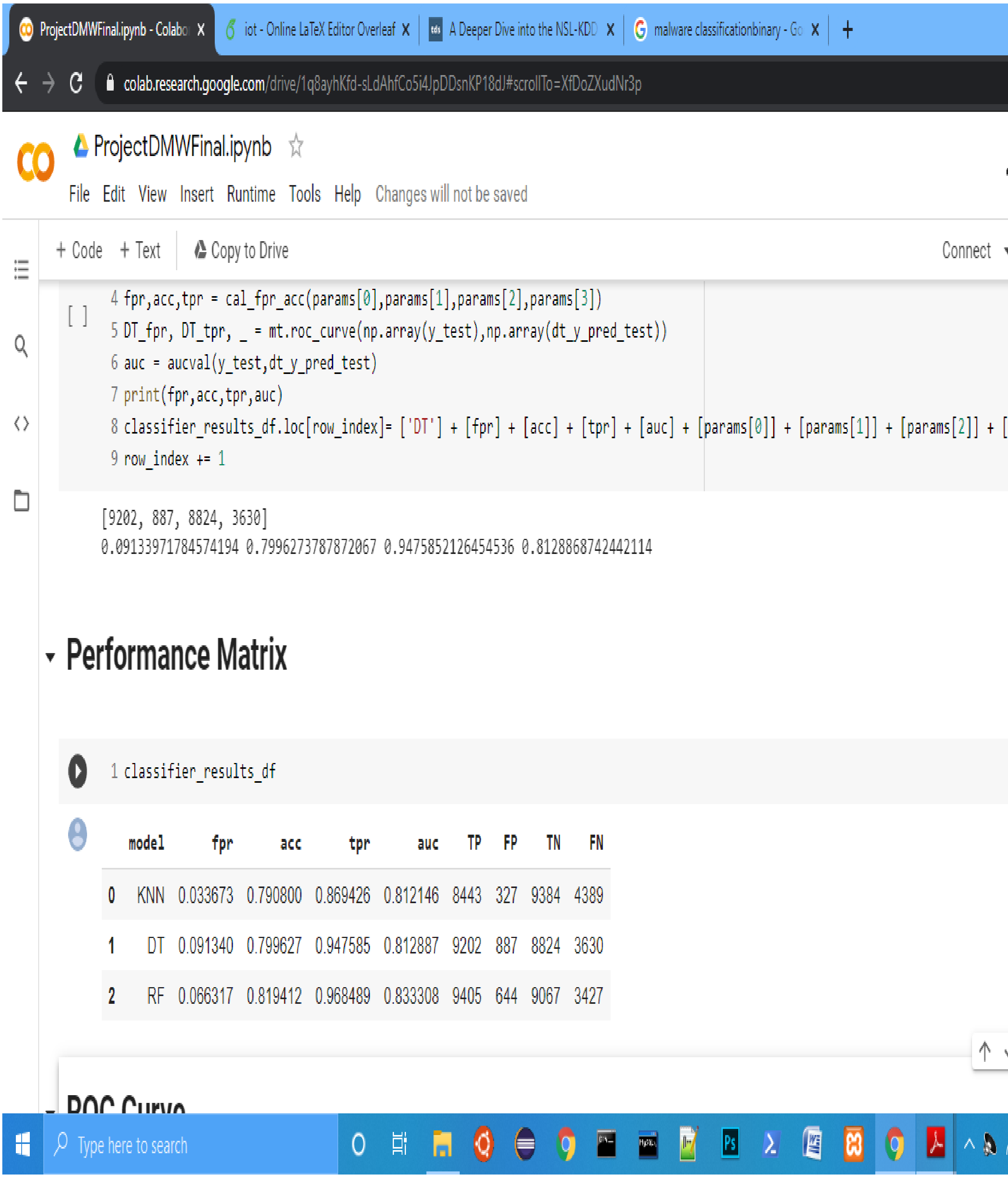


ProjectDMWFinal.ipynb - Colabo
iot - Online LaTeX Editor Overleaf
A Deeper Dive into the NSL-KDD
mal
colab.research.google.com/drive/1q8ayhKfd-sLdAhfCo5i4JpDDsnKP18dJ#scrollTo=XfDoZXudNr3p
ProjectDMWFinal.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive

```
import matplotlib.pyplot as plt
plt.plot(DT_fpr,DT_tpr,label="DT")
plt.plot(rf_fpr,rf_tpr,label="RF")
plt.plot(knn_fpr,knn_tpr,label="KNN")
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.title('ROC Curve')
plt.legend(loc=4)
plt.show()
```

ROC Curve
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
DT
RF
KNN
Type here to search

Figure 2: ROC curve for all models


ProjectDMWFinal.ipynb - Colabo X
iot - Online LaTeX Editor Overleaf X
A Deeper Dive into the NSL-KDD X
malware classificationbinary - Go X
colab.research.google.com/drive/1q8ayhKfd-sLdAhfCo5i4JpDDsnKP18dJ#scrollTo=XfDoZXudNr3p
ProjectDMWFinal.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect
Type here to search

```
fpr,acc,tpr = cal_fpr_acc(params[0],params[1],params[2],params[3])
DT_fpr, DT_tpr, _ = mt.roc_curve(np.array(y_test),np.array(dt_y_pred_test))
auc = aucval(y_test,dt_y_pred_test)
print(fpr,acc,tpr,auc)
classifier_results_df.loc[row_index]= ['DT'] + [fpr] + [acc] + [tpr] + [auc] + [params[0]] + [params[1]] + [params[2]] + [
row_index += 1
```

```
[9202, 887, 8824, 3630]
0.09133971784574194 0.7996273787872067 0.9475852126454536 0.8128868742442114
```

Performance Matrix

```
classifier_results_df
```

	model	fpr	acc	tpr	auc	TP	FP	TN	FN
0	KNN	0.033673	0.790800	0.869426	0.812146	8443	327	9384	4389
1	DT	0.091340	0.799627	0.947585	0.812887	9202	887	8824	3630
2	RF	0.066317	0.819412	0.968489	0.833308	9405	644	9067	3427

ROC Curve

Figure 3: Performance matrix

7 CONCLUSION

Malware analysis and classification is of importance for securing information systems and networks. We have successfully performed binary classification of malwares with three classifier models. Random forest classifier performs the best amongst all three.

References

[1] Jiawei Han, Micheline Kamber, Jian Pei,"Data Mining Concepts and Techniques, Third Edition" *The Morgan Kaufmann Series in Data Management Systems*

[2] Kamalakanta Sethi, Rahul Kumar, Lingaraj Sethi, Padmalochan Bera, Prashanta Kumar Patra,"A Novel Machine Learning Based Malware Detection and Classification Framework," *International Conference on Cyber Security and Protection of Digital Services, Oxford, United Kingdom, pp. 1-4, 2019.*

[3] A Deeper Dive into the NSL-KDD Data Set Available at: https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657 [Online]

[4] Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, "Elements of Statistical Learning- Data Mining, Inference, and Prediction" *Springer Series in Statistics* Available at: https://link.springer.com/book/10.1007/978-0-387-84858-7 [Online]