NYU TANDON SCHOOL OF ENGINEERING

Network Intrusion Detection

Fall 2018, FDS

Team members:
Geetesh Nikhade (gpn218)
Rahul Keshwani (ryk248)

Instructor:
Rumi Chunara, PhD

Problem Statement and Goal:

With the boom of Internet and its powerful ability to store and share data across networks, the number and diversity of cyber network attacks have also increased. Servers of many companies have been victims of such network attacks (one of the most popular examples being the data breach at Yahoo in 2014: around 3 billion accounts were compromised in this attack). With the advent of technology, these network attacks have become sophisticated and hence difficult to detect, creating an immediate need of an intelligent Network Intrusion Detection system to detect and classify anomalous behaviour.

Since the firewall systems cannot detect modern attack environments and are not able to analyse network packets in depth, in this project we used various machine learning classification techniques to first predict if the packet captured over the network is a genuine regular packet or an attack and then we leverage the data gathered to detect if it is one of the various types of network attacks like Denial of Service, Backdoors, Worms, Exploits etc.

Target Variable:

- ***Binary Classification:***

Label	0	1
Type	Benign Packet	Attack Packet

- ***Multi-class classification:*** In the table below, the first row represents labels and the second row represents the assigned attack type.

0	1	2	3	4	5	6	7	8	9
Analysis	Backdoor	DoS	Exploits	Fuzzers	Generic	Normal	Reconnaissance	Shellcode	Worms

Dataset:

The raw network packets of the UNSW-NB 15 data set was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours. The number of records in the training set is 175,341 records and the testing set is 82,332 records from the different types, attack and normal. The dataset has data related to 9 types of attacks and 48 features as shown in the tables below.

Type of Attack	UNSW_NB15_testing-set.csv	UNSW_NB15_training-set.csv
Normal	37000	56000
Fuzzers	6062	18184
Analysis	677	2000
Backdoor	583	1746
DoS	4089	12264
Exploits	11132	33393
Generic	18871	40000
Reconnaissance	3496	10491
Shellcode	378	1133
Worms	44	130

The features in the UNSW-NB15 dataset are:

srcip	dloss	dmeansz	dur	dtcpb	ct_state_ttl
sport	service	sloss	sbytes	smeansz	ct_flw_http_mthd
dstip	Sload	Sjit	dbytes	trans_depth	is_ftp_login
dsport	Dload	Djit	sttl	res_bdy_len	ct_ftp_cmd
proto	Spkts	Stime	dttl	ct_srv_src	is_sm_ips_ports
state	Dpkts	Ltime	swin	ct_srv_dst	ct_dst_sport_ltm
synack	tcprtt	Sintpkt	dwin	ct_dst_ltm	ct_dst_src_ltm
ackdat	dmeansz	Dintpkt	stcpb	ct_src_ltm	attack_cat

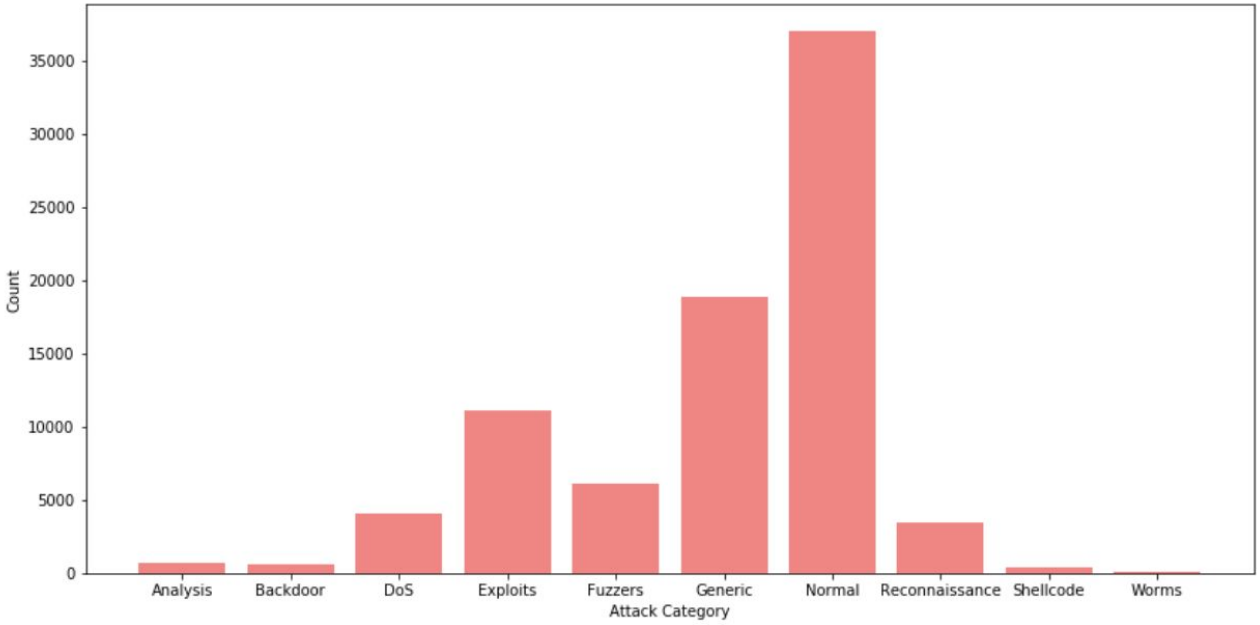
As there are a lot of features in the dataset, and it would not be possible for us to describe every attribute, we find out the most important features in the Exploratory data analysis section of this report, and have described the important features there.

Our Approach:

In this project, we first started understanding the data by gaining some preliminary knowledge of the network security domain, and preprocessing the data using Min-Max normalization, Label encoding as well as dimensionality reduction using PCA on both the training as well as testing dataset. Next, we performed exploratory analysis by plotting various different graphs, and understanding the most important features by calculating the feature importances of all the features. Finally, we used the models described below for training and then test our model on a training set based on the metrics also explained below.

Exploratory Data Analysis:

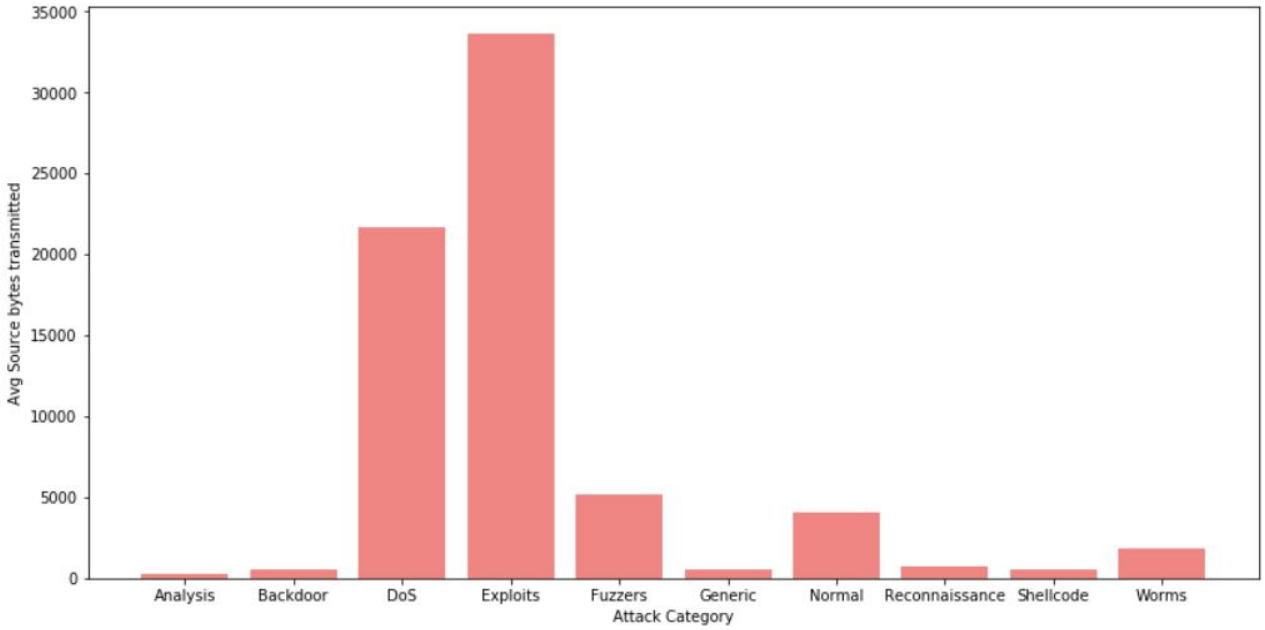
- ***Attack type distribution:***



As there seems to an imbalance in the count of each attack category. This imbalance in the data will not cause a problem while performing **Binary classification** since there is a balance between the count of attack and non-attack entries.

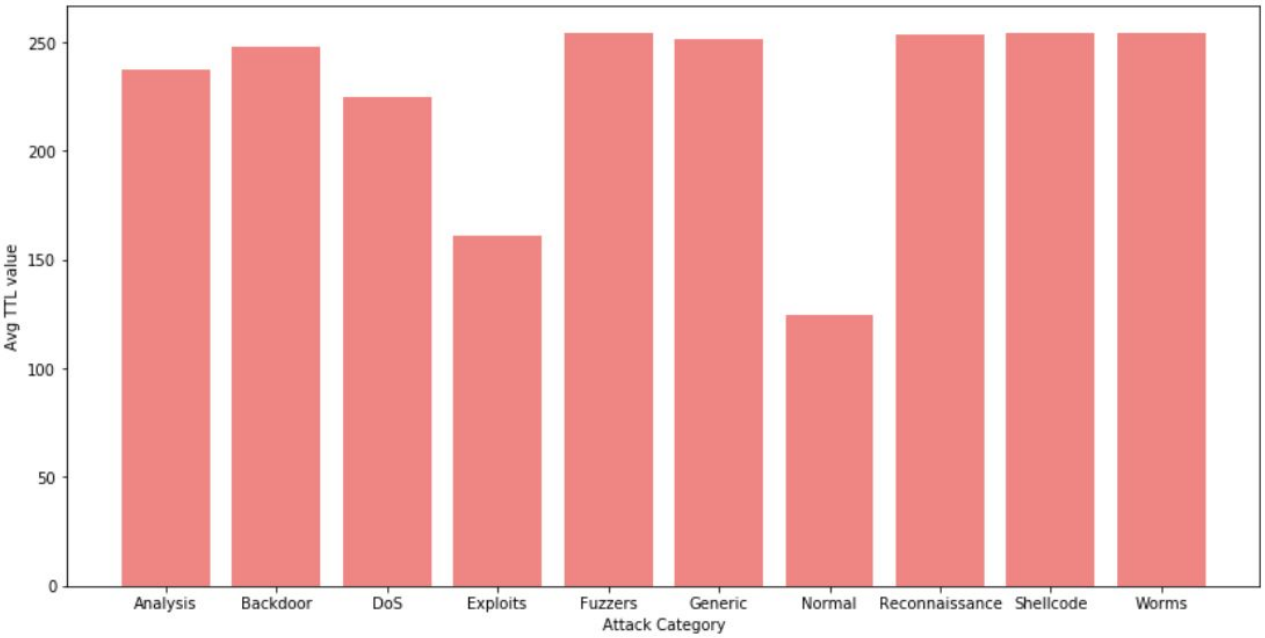
Although, the class imbalance can affect the predictions while performing **Multiclass classification** and can result in a higher number of False Negatives. Well, we can't upscale or downscale the number of samples, so we plan to do some hyperparameter tuning to attain good results.

- ***Average source bytes distribution graph:***



The graph above tells us that DoS and Exploits are 2 types of network attacks which require a large number of source bytes to be transmitted from source to destination. This makes sense since these network attacks are meant to jam the network so that genuine users are unable to use the service.

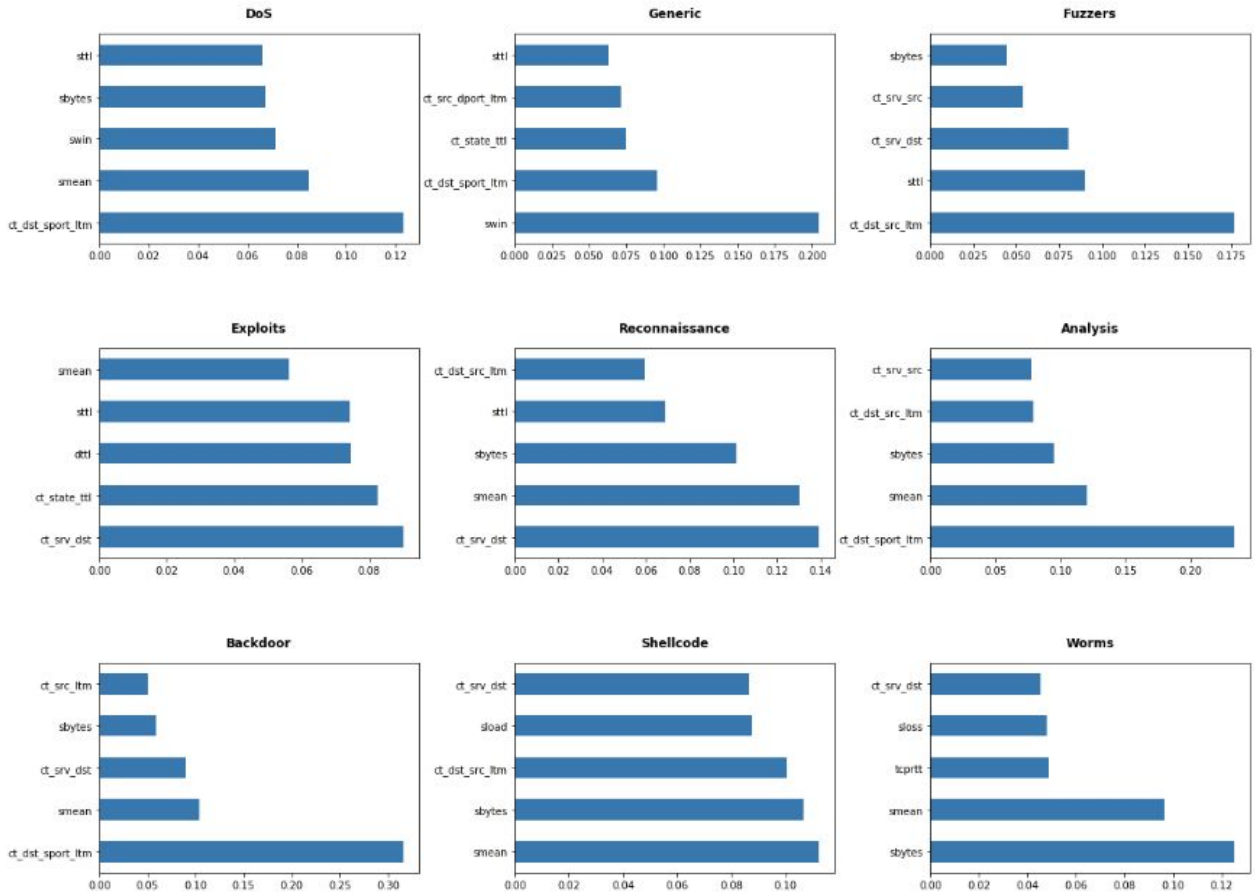
- ***Average Source to Destination TTL per attack:***



Time to Live value (ttl) defines the hop count for a network packet and can hold a maximum value of 255. From the graph above we can see that almost every attack type has an average time to live value

of around 255 which will allow them to hop on the network as far as possible. Whereas, for a normal packet it is 128 which restricts it to travel only within the continent.

- ***Feature Importances as per each attack:***



The graphs above show the feature importance for each attack category. Although, the top feature for each attack type is different, the top 5 for each of them are mostly the same (the order of occurrence is different). Hence, some of the top features are: Source Bytes (sbytes), Mean packet size transmitted from source (smean), Source Time to Live value (sttl), Count of packets from same destination to the same source port (ct_dst_sport_ltm) and Number of connections between same source and destination address (ct_dst_src_ltm). Below are short descriptions of these attributes:

Feature Name	**Description**
sbytes	Source to destination transaction bytes
smean	Mean of the row packet size transmitted by the src
sttl	Source to destination time to live value
ct_dst_sport_ltm	No of connections of the same destination address and the source port in 100

	connections according to the last time .
ct_dst_src_ltm	No of connections of the same source and the destination address in in 100 connections according to the last time.

Models Used:

- ***Random Forest Classifier:*** Initially, we thought of using Decision Tree based on its popularity, but it's prone to overfitting (high variance) and is highly dependent on the training sample distribution. So, the type of model that we finally selected is Random Forest Classifier. Since random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction, this often results in Random forest being a strong learner.
- ***XGBoost*:** This model is built on top of Decision Trees and the Boosting technique. Since we had already tried a bagging method, we thought using a boosting technique, wherein the model would learn from it's past mistakes, might help us reduce the number of misclassified network attack packets. XGBoost also performs regul

Evaluation Metrics:

In order to develop a smart Network Intrusion Decision System it is very important for us to avoid False negatives as predicting a malicious packet to be safe can bring an entire system down and hence would decrease the entire performance of Intrusion detection system. This also means that favoring True Positives and completely avoiding False Positives, building a confusion matrix for this problem statement is the most suitable method for us. We will use the following metrics based on our confusion matrix:

- ***Accuracy*** – Measures the rate of correctly classified attack instances over all classes.

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

- ***Precision (Type I Error)*** – Measures the percentage of correctly classified attacks to total number of relevant and irrelevant attacks. that are truly correct.

$$Precision = \frac{TP}{TP + FP}$$

- ***Recall (Type II Error)*** – Measures the percentage of correctly predicted relevant attacks to the actual relevant attacks.

$$Recall = \frac{TP}{TP + FN}$$

- ***ROC Curve*** - We use the ROC curve to reflect the accuracy of our prediction. ROC helps to achieve a trade-off between specificity and sensitivity.

Out of the four measures described above, we would like to provide primary focus to **Accuracy** and **Recall score**. Accuracy will give us an overall idea about the quality of the predictions. On the other hand, Recall score will help us in building a more reliable system.

Binary Classification Results:

1. *Random Forest Confusion matrix **vs** XGBoost Confusion matrix:*

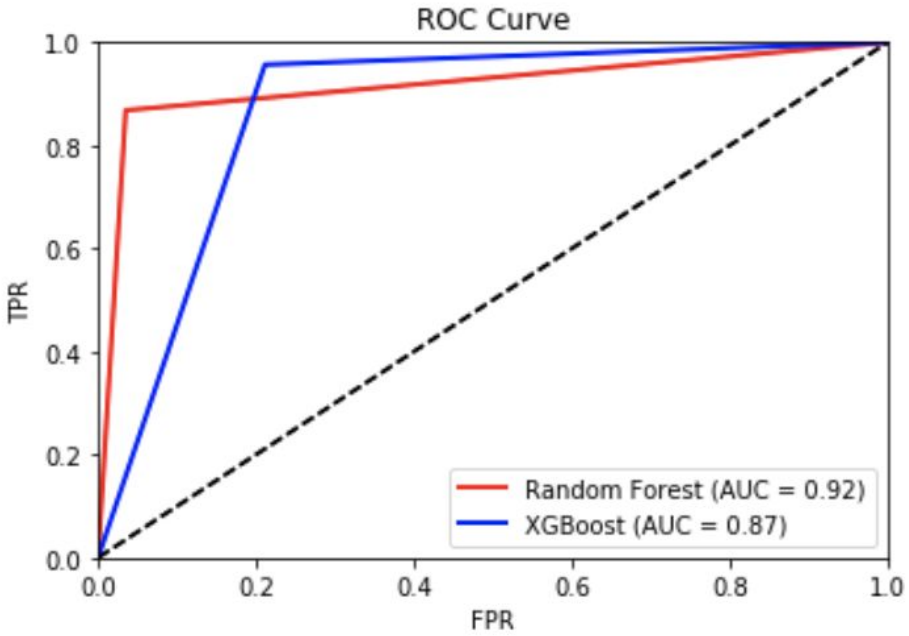
	Predicted NO	Predicted YES
Actual NO	54041	1959
Actual YES	15738	103603

	Predicted NO	Predicted YES
Actual NO	44160	11840
Actual YES	5160	114181

2. *Accuracy **vs** Precision **vs** Recall:*

	Accuracy	Precision	Recall
Random Forest	89.9070953171%	91.5336494554%	89.9070953171%
XGBoost	90.3046064526%	90.6047404798%	95.6762554361%

3. *Area Under ROC*

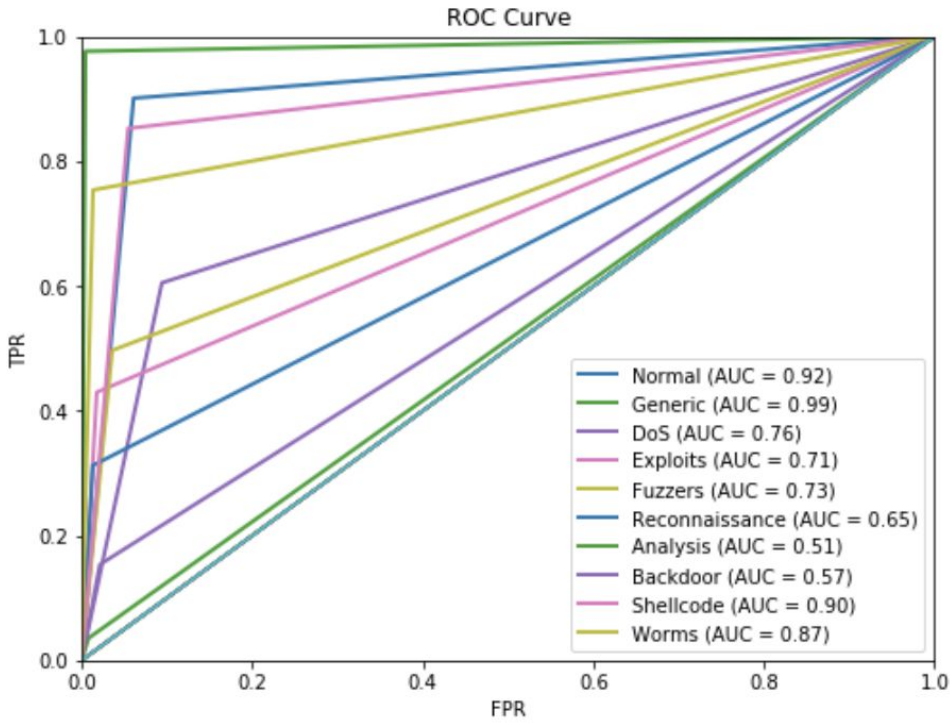


Multiclass Classification Results:

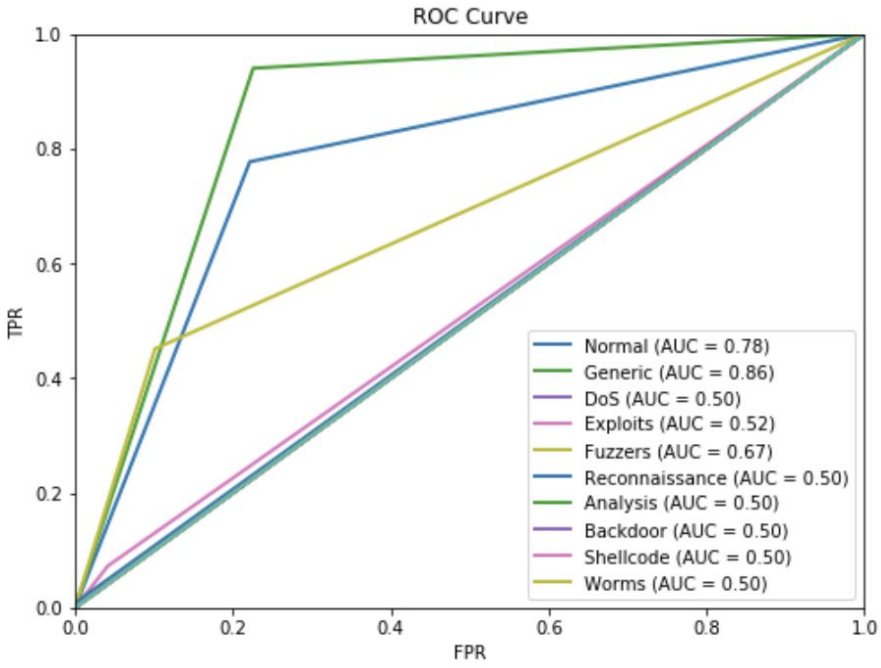
1. *Accuracy **vs** Precision **vs** Recall:*

	Accuracy	Precision	Recall
Random Forest	71.2805333607%	79.0725085912%	71.2805333607%
XGBoost	52.5598690552%	42.9791060201%	52.5598690552%

2. *Random Forest Area Under ROC*



3. *XGBoost Area Under ROC*



Analysis of the Results:

- For Binary classification, it can be seen from all the evaluation metrics, XGBoost has performed better as compared to Random Forest algorithm. As mentioned in the earlier sections, since it is important for the Network intrusion detection system to minimize the "False Negatives", XGBoost has been successful to achieve this as indicated by Recall score and the confusion matrix.
- For Multiclass classification, we cannot surely assert which model will perform better because of the major problem of class imbalance. Although, by just looking at the results, we can say that Random Forest does perform better especially in classifying some particular network attack types such as "Generic", "ShellCode", "Worms", and "DoS".

Limitations and Scope of Improvement:

- Even though we think the UNSW-NB15 dataset is a good fit for our project than the older benchmarked datasets like KDD-99 (as the old datasets do not display satisfactory results in the current technology setup), we think a newer dataset would be more relevant for the current attacks. For example, the UNSW-NB15 dataset has no data about latest attacks such as Distributed Denial of Service (DDoS), Man in the middle (MitM), SQL injection and various different attacks, which would be important to predict as per current use cases.
- Imbalance in the dataset regarding entries related to each attack type could affect the predictions while performing Multiclass predictions. Since we could not upscale/downscale the data, it would be great to have a good dataset with proper distribution of all the attack types.

- Apart from getting a richer dataset, if we could dive further deep into data science techniques, and come up with a solution to counter the class imbalance. We feel, this could be a good improvement that can be applied in the future

Changes:

- After further research, we selected Random Forest over Decision Trees since it overcomes the problem of overfitting by using the concept of Bagging. Since Random Forest creates multiple trees and for each of the tree it identifies the best split feature from a random subset of features, it reduces variance and hence reduces overfitting.
- We decided to drop the idea of using a Multi Level Perceptron, for binary as well as multi-class classification as the results we were initially getting for multiclass classification using Random Forest and XGBoost were not good enough. We decided to spend more time understanding the hyper parameters and tuning them accordingly to improve the prediction accuracy.

Data Source:

- Dataset can be found on the UNSW-NB15 source files website here.