Assignment No. 3

Student No: 300318105

In this Assignment, the task is to generate a model to determine the anomalies in DNS queries using binary supervised classification techniques.

Data Analysis:

In Exploratory data analysis, it can be noticed that the data is not imbalanced. There is a total of 268074 records for the static dataset. By using a Counter from python's collection, we can measure the class distribution for the target variable. There are approximately 147179 records for the class label '1' and 120895 for the class label '0' which is 55% and 45% of the data is '0' and '1' labelled.

Data Preprocessing and Feature Engineering:

For performing the statistical analysis of data, Kernel density estimation is used. KDE is used to estimate the probability density function of the continuous random variable. The area under the curve is highlighted. From the KDE plots on the input features, it is evident that most of them have a multimodal distribution with spikes in data, probably indicating anomalies or outliers in data.

There are 16 input features corresponding to various attributes in the DNS query which are as follows: 'timestamp', 'FQDN_count', 'subdomain_length', 'upper', 'lower', 'numeric', 'entropy', 'special', 'labels', 'labels_max', 'labels_average', 'longest_word', 'sld', 'len', 'subdomain',' Target Attack'.

For the column 'longest_word', the column is transformed such that if the column value is a string then the value is replaced with its length. Similarly, label encoding is applied on columns 'sld', and 'subdomain_length' to convert categorical data to numeric values. Hashing is another alternative used to encode the text field but the encoded data with negative values was generated which was not suitable for statistical feature selection approaches like the chi-square test. Thus, the label encoder is used while pre-processing. An index column was added as the data records had duplicate values. Missing values in the dataset were dropped.
After performing feature selection techniques on data, a few columns from the data frame are dropped. The input features dropped from the data are 'FQDN_count', 'upper', and 'labels_average'.

After performing data cleaning, the data is split using the hold-out method train-test ratio of 77:33. Feature selection is done by considering the training data. After selecting features, K-Fold cross-validation is performed for ML models KNN, Random Forest Classifier and XGBoost classifier. K-Fold with 10 splits is performed. Training data is split into 10 partitions and the model is trained to select one partition randomly for testing and training on others. Then the mean accuracy over 10 fold is calculated. This is generally used to evaluate the model's performance over k different training datasets and avoid overfitting of data.

For the Dynamic dataset, Kafka consumer is used to dealing with the live stream of data. The input bytes of data are first decoded, cleaned and then transformed to be used with the model developed with static data. After cleaning data, the data is split into train-test data of a 7:3 ratio without K-fold cross-validation. After performing the split that train is trained using the dynamic data model. Data is trained only when the accuracy level falls below 0.8.

Feature selection:

Statistical analysis of data is performed to select the best features from the data. There are three techniques used chi-square test, ANOVA-F test and mutual information gain.

The chi-Square test helps to find the relationship between variables which are independent of each other. After applying the chi-square test, f1 measure values and p-values are generated. The more the value of the f1-score, the more important the feature is for determining the target. Conversely, the less the p-value, the lesser the importance of the feature.

ANOVA-F stands for ‘Analysis of variance’ which helps to determine if two or more samples of data are from the same distribution ie. if the samples belong to a particular distribution or not. ANOVA-F test determines how similar two classes are to each other. According to this test, 'labels_average', 'labels_max', 'upper', and 'entropy' are the least important features according to the target class.

Mutual Information gain helps to find the relative importance between the given variables. It can generate negative or positive values but the absolute difference is considered while setting up the threshold to select features. In this case, it is 0.001.
The following features are selected based on the mutual information gain test: 'timestamp', 'subdomain_length', 'lower', 'numeric', 'entropy', 'special', 'labels', 'labels_max', 'longest_word','sld', 'len', 'subdomain'.

Model Selection, Evaluation and Parameter tuning:

The Random Forest, KNN, and XGBoost models are used on the dataset. All three models are used for classification. Among all the models, XGBoost appears to show the best performance in terms of accuracy as well as overall model performance

The performance metric chosen to select the model is the K-fold Evaluation technique. For the model evaluation, accuracy, recall, precision and ROC curves are used and the confusion matrix for the model is then evaluated. For the dataset given, recall values are quite high but the accuracy for almost all the models is stagnant at 0.826. Accuracy for Random Forest, KNN, and XGBoost are 0.8251, 0.8138, and 0.8262 respectively.

From the ROC curves, it is quite evident that all the models are performing the same with the average area under the curve of 0.8 with an AUC score of 0.806,0.799,0.808 for Random Forest, KNN and XGBoost respectively.

All the ML models are trained and evaluated after tuning the parameters. Tunning certain parameters has helped increased the accuracies to great extent.

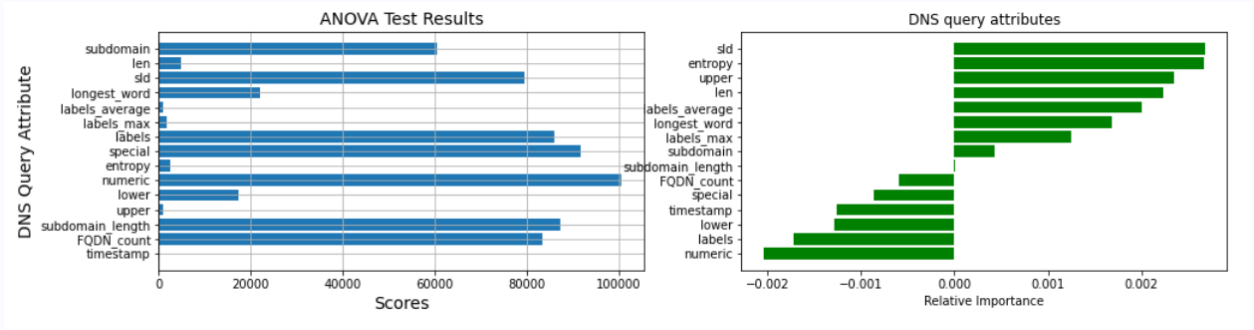
Discuss and analyze the results

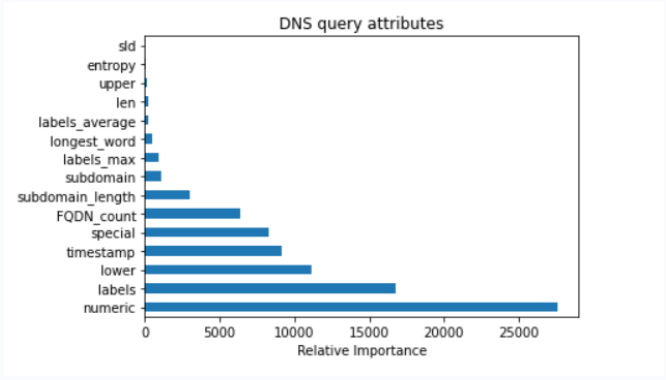
After training the model on the given static dataset, performing data cleaning and feature selection it is found that XGBoost provides the best results from all the algorithms considered. Thus the XGBoost model will be used to train the dynamic dataset using Kafka.

For the static and dynamic models, it can be easily seen that static models' performance deteriorates when the model is evaluated against the dynamic stream of data. The dynamic model learns for the first 150-200 windows and then has consistent accuracy.

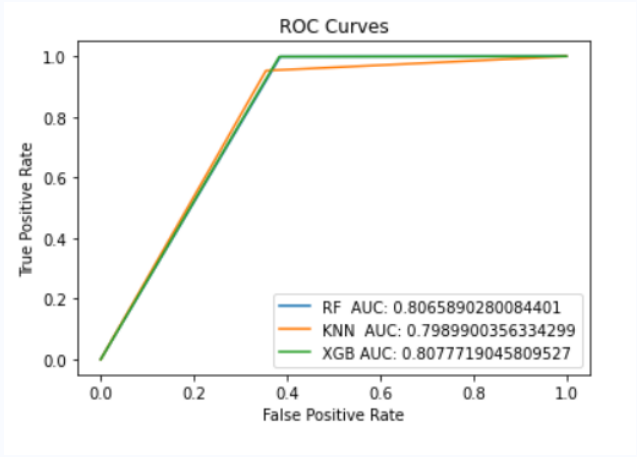
Plots:

Feature Selection:





ROC curves:



Performance of Statics and Dynamic Model: