Local Dynamic Neighborhood Based Outlier Detection Approach and its Framework for Large-Scale Datasets

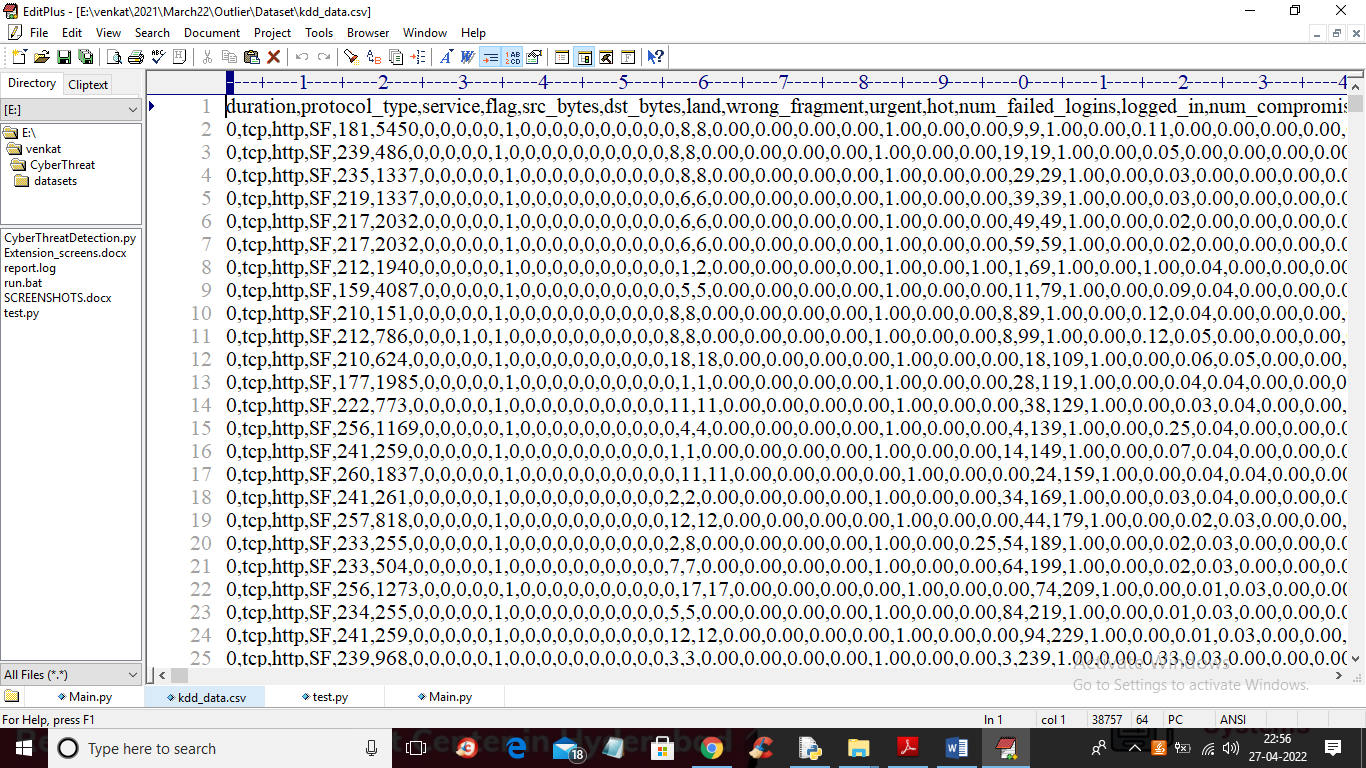
All machine learning algorithms get trained on past dataset to predict new data labels but this dataset often contains garbage or incorrect values called as Outliers and this outliers can effect machine learning algorithms and to overcome from this problem many outlier detection algorithms are introduced but their detection rate is not good when dataset size is huge. To increase outlier detection rate author of this paper has introduced novel algorithm called LDNOD (local dynamic neighbourhood based outlier detection). Propose algorithm will apply KMEANS algorithm to group similar data in clusters and then this clusters will be evaluated with neighbour instances to get similarity score. Sum of similarity score of all instances with the compare has good matching score then that record will be accepted otherwise that record will be consider as outlier and removed out from dataset.

To implement this project we have used KDD dataset and then train Random Forest algorithm with and without outlier detection and then compare their accuracy and other metrics.

To implement this project we have designed following modules

1. Upload KDD Dataset: using this module we will upload dataset to application
2. Preprocess Dataset: dataset often contains missing values and non-numeric characters data so by using this values we will replace missing values with 0 and then encode all non-numeric data to numeric id.
3. Run Random Forest on Full Dataset: using this module we will train Random Forest algorithm on full dataset without outlier detection and then calculate its accuracy
4. Run K-Means Algorithm: using this module we will cluster entire dataset into different groups
5. LDNOD Outlier Detection with Random Forest: using this module we will read each record from cluster and then calculate its similarity score with its neighbourhood records and if similarity score is high then we will select that record otherwise that record will be consider as outlier. After outlier detection we will retrain data with Random Forest and calculate its accuracy and compare with the one without applying outlier.
6. Comparison Graph: using this module we will plot comparison graph between with and without outlier detection.

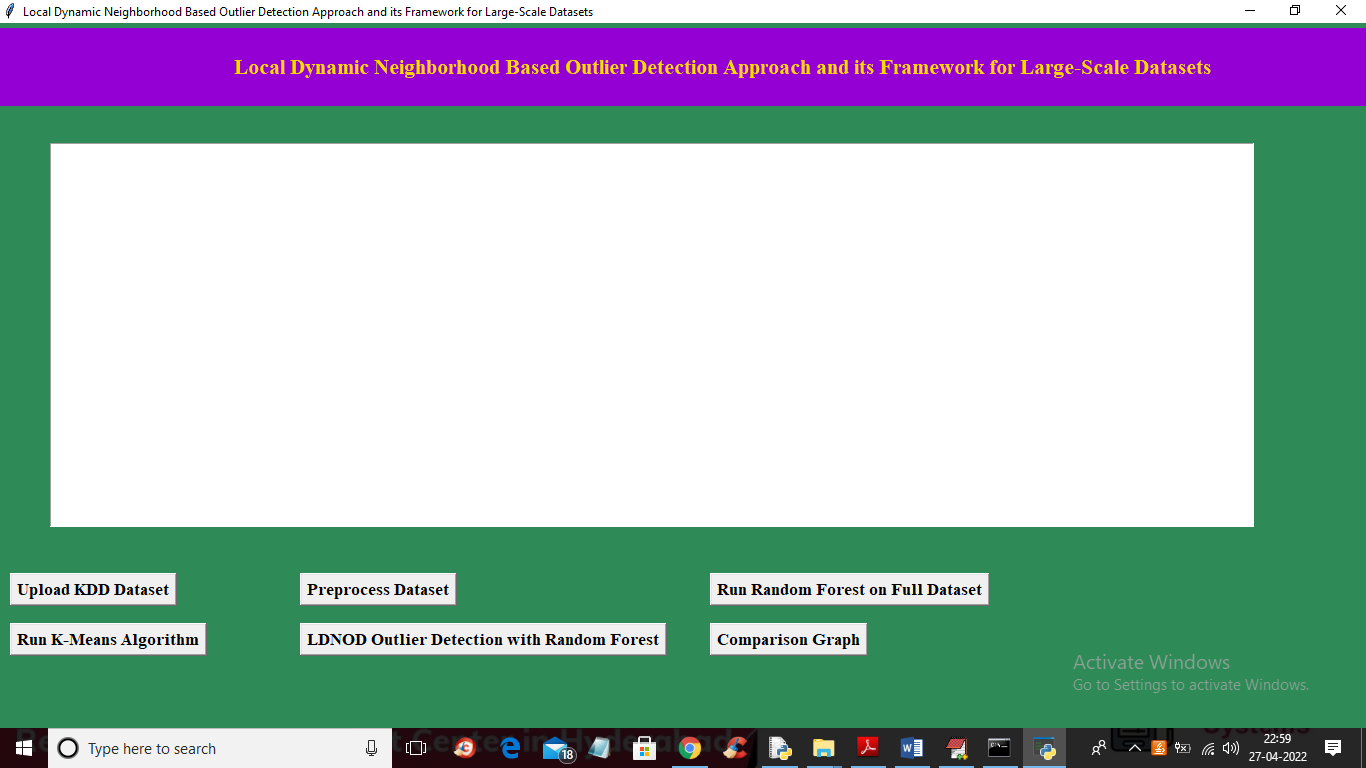
Below screen showing records from dataset



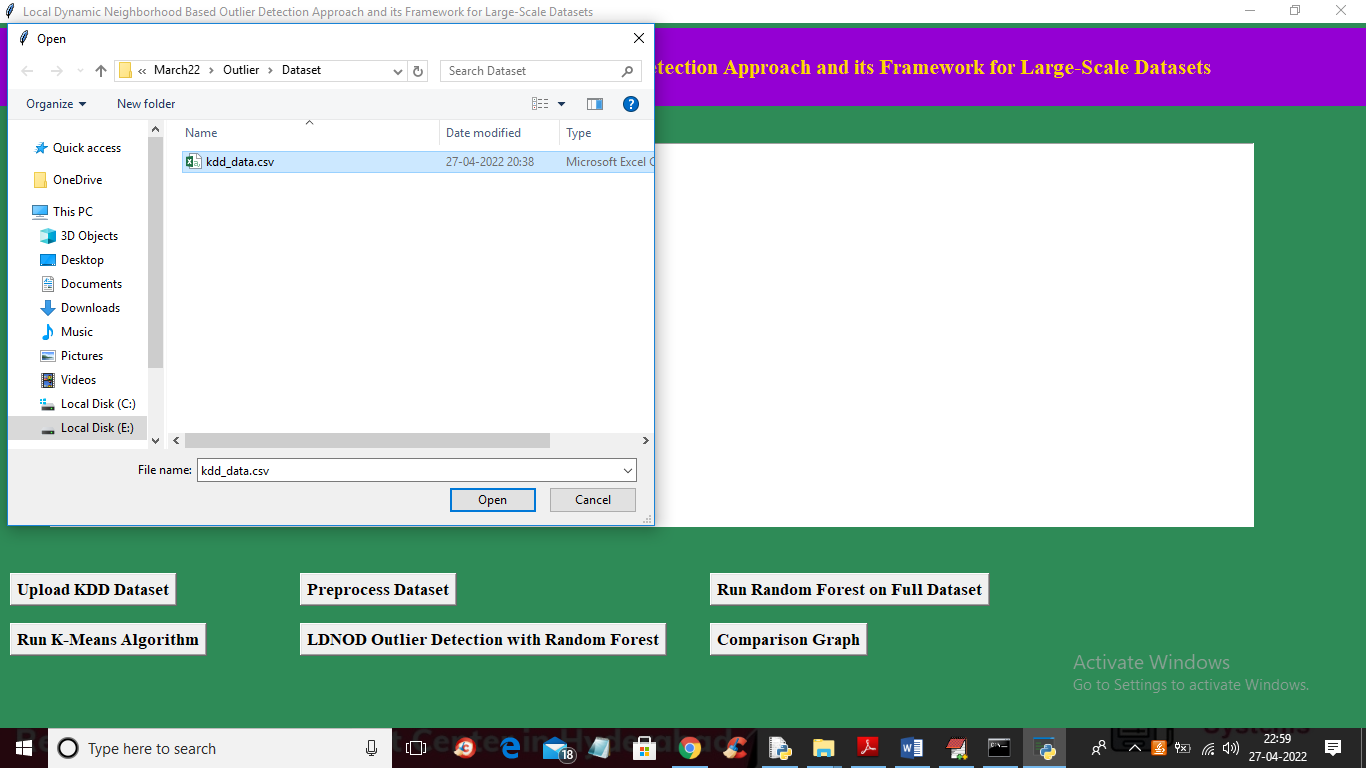
In above screen first row contains dataset column names and remaining rows contains dataset values. In above screen we can see some values are non-numeric so by applying data processing technique we can encode those values to numeric.

SCREEN SHOTS

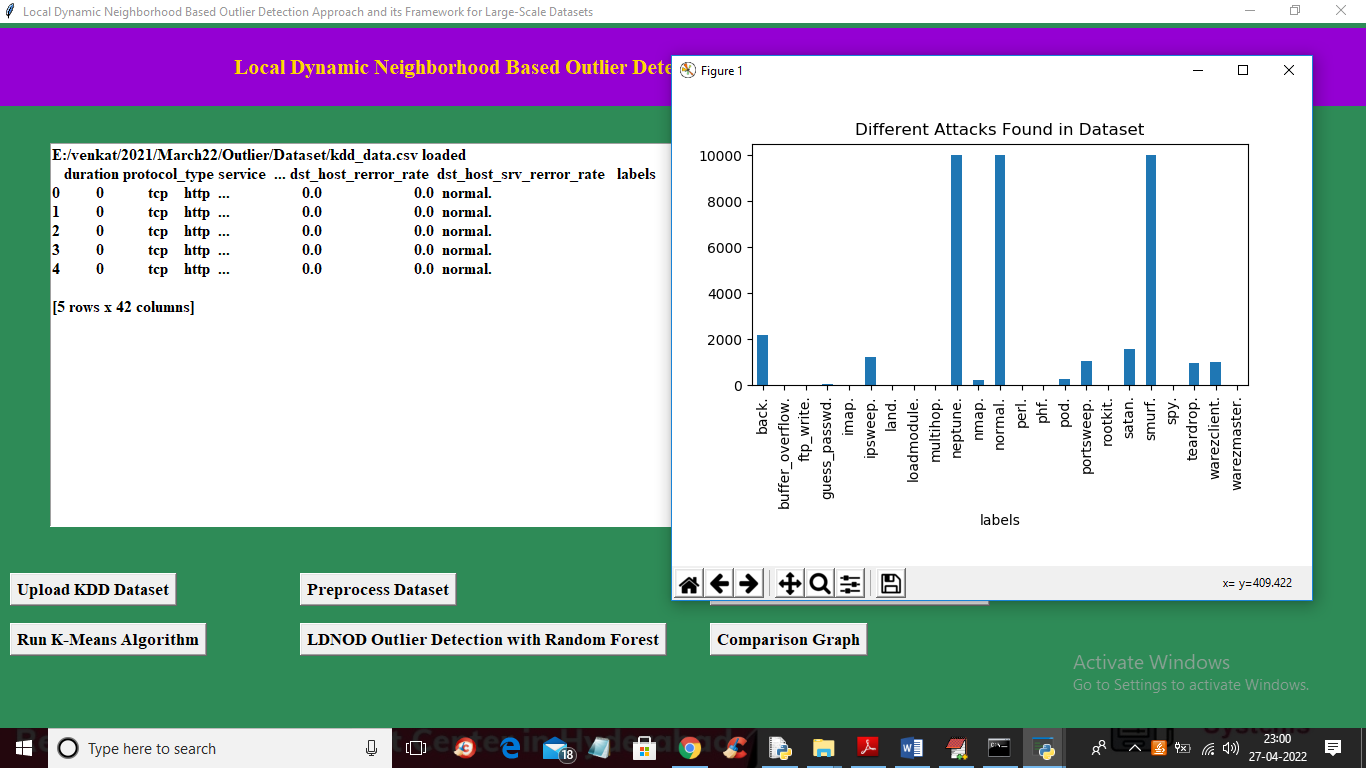
To run project double click on ‘run.bat’ file to get below screen



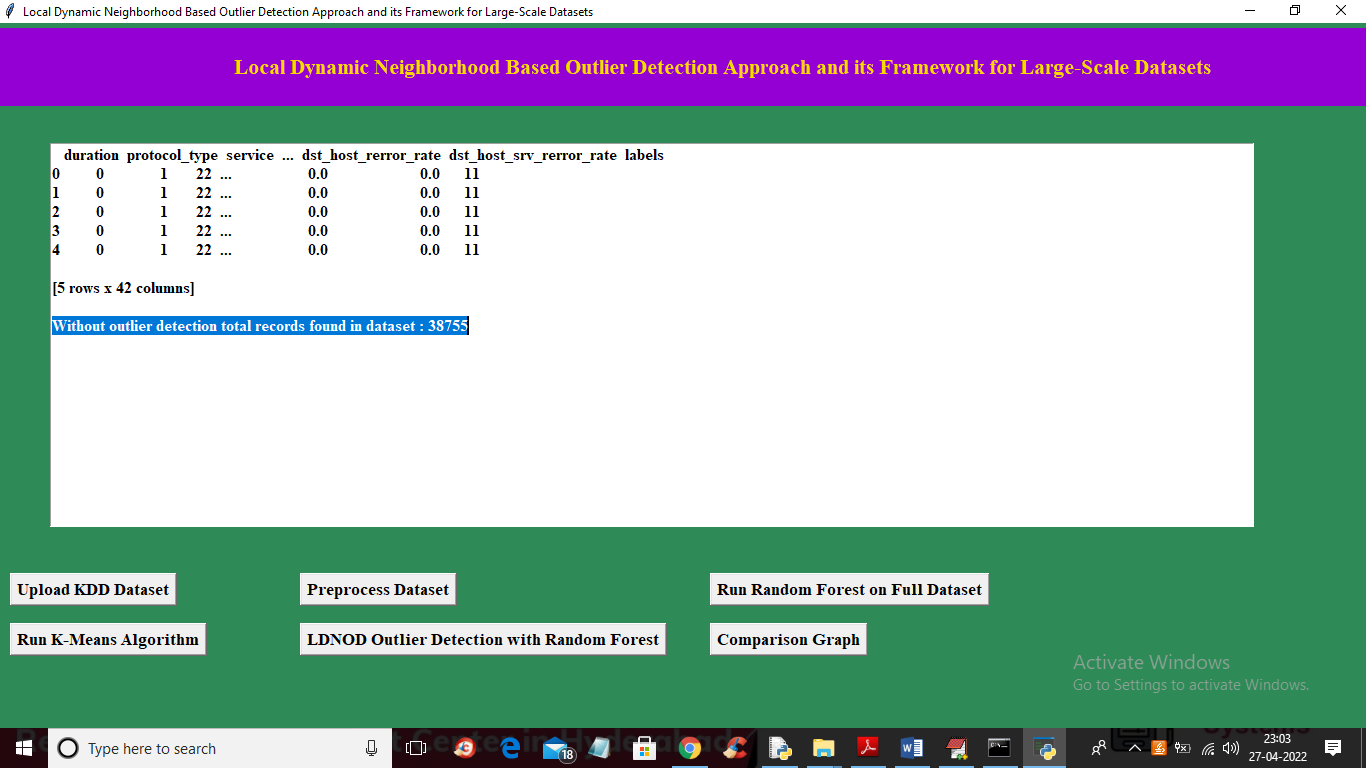
In above screen click on ‘Upload KDD Dataset’ button to upload dataset and to get below screen



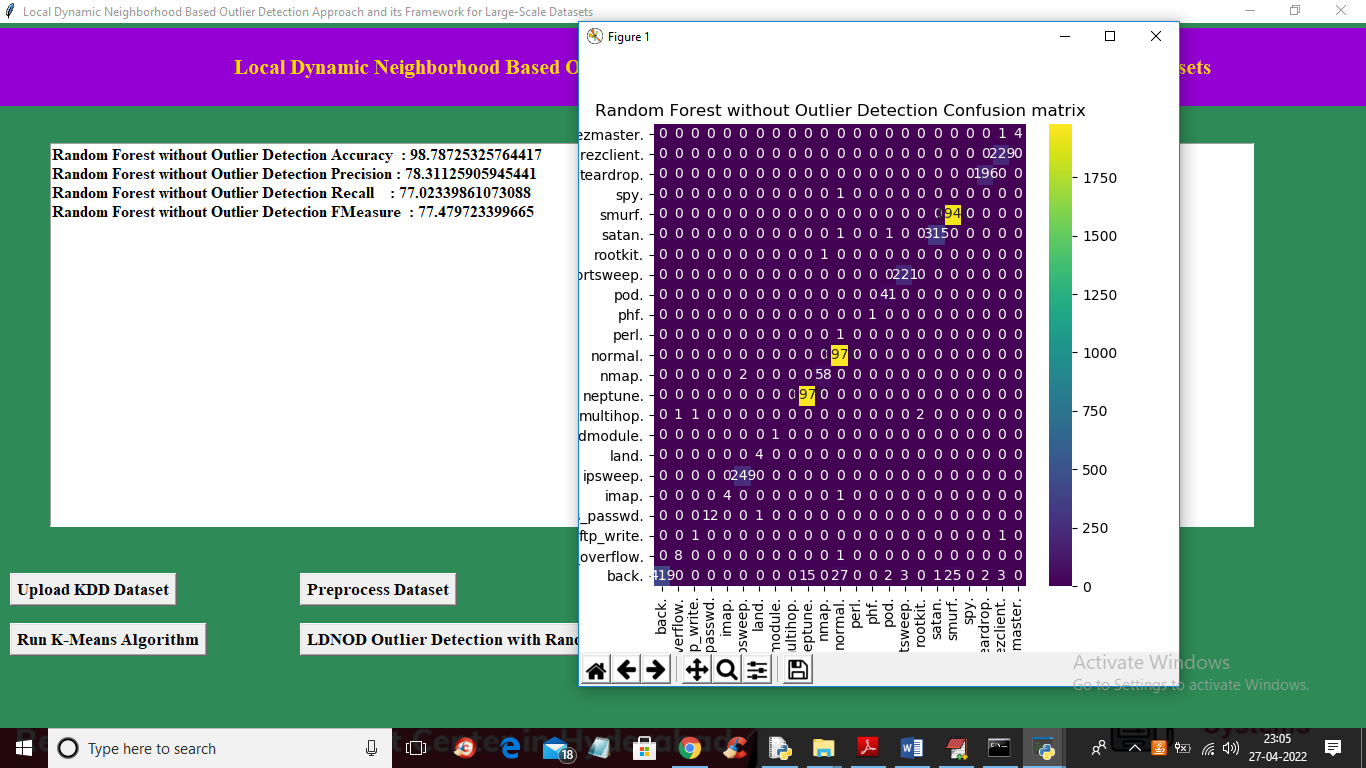
In above screen selecting and uploading ‘kdd dataset’ file and then click on ‘Open’ button to load dataset and to get below output



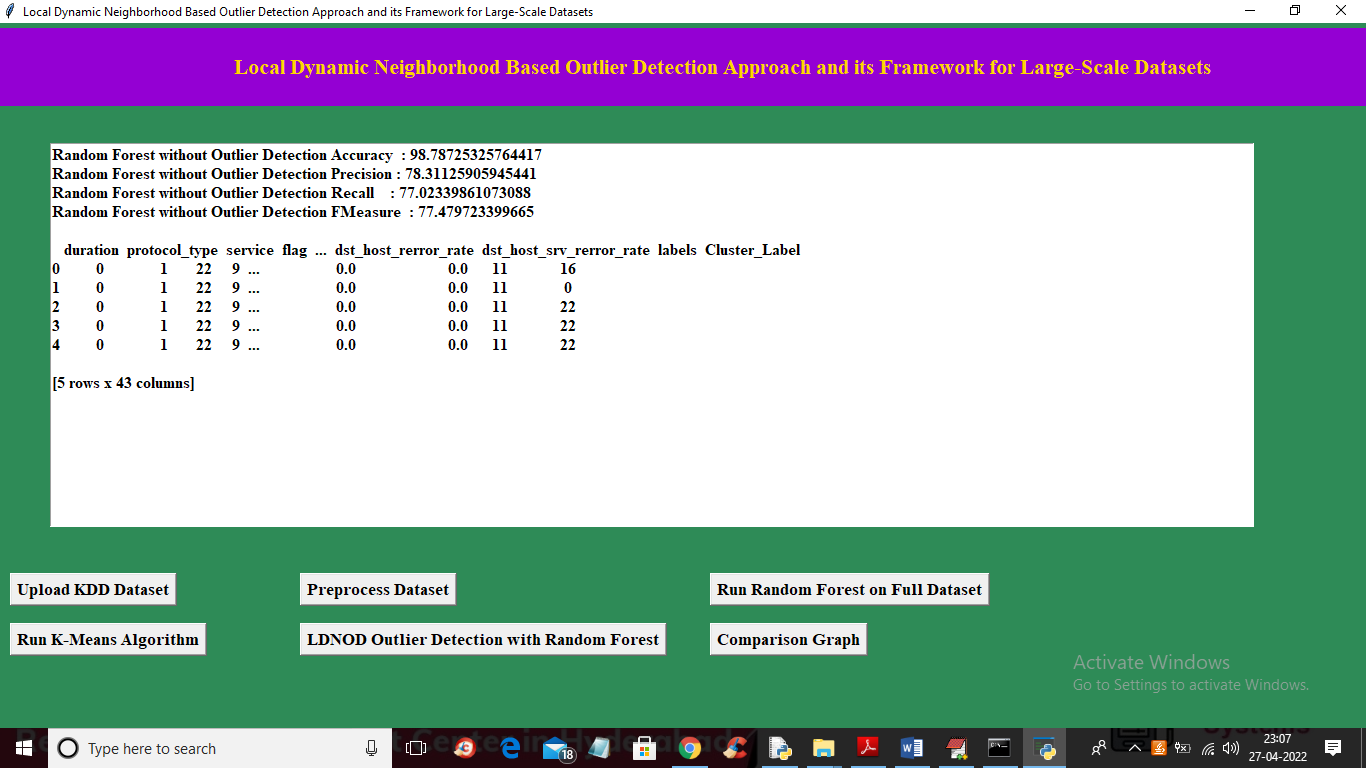
In above screen dataset loaded and displaying some values and dataset does not contains any cluster labels and after applying KMEANS will get cluster label. In above graph x-axis contains attack names and y-axis contains count of each attack found in dataset. In above screen dataset contains some non-numeric data so close above graph and then click on ‘Preprocess Dataset’ button to process data and get below output



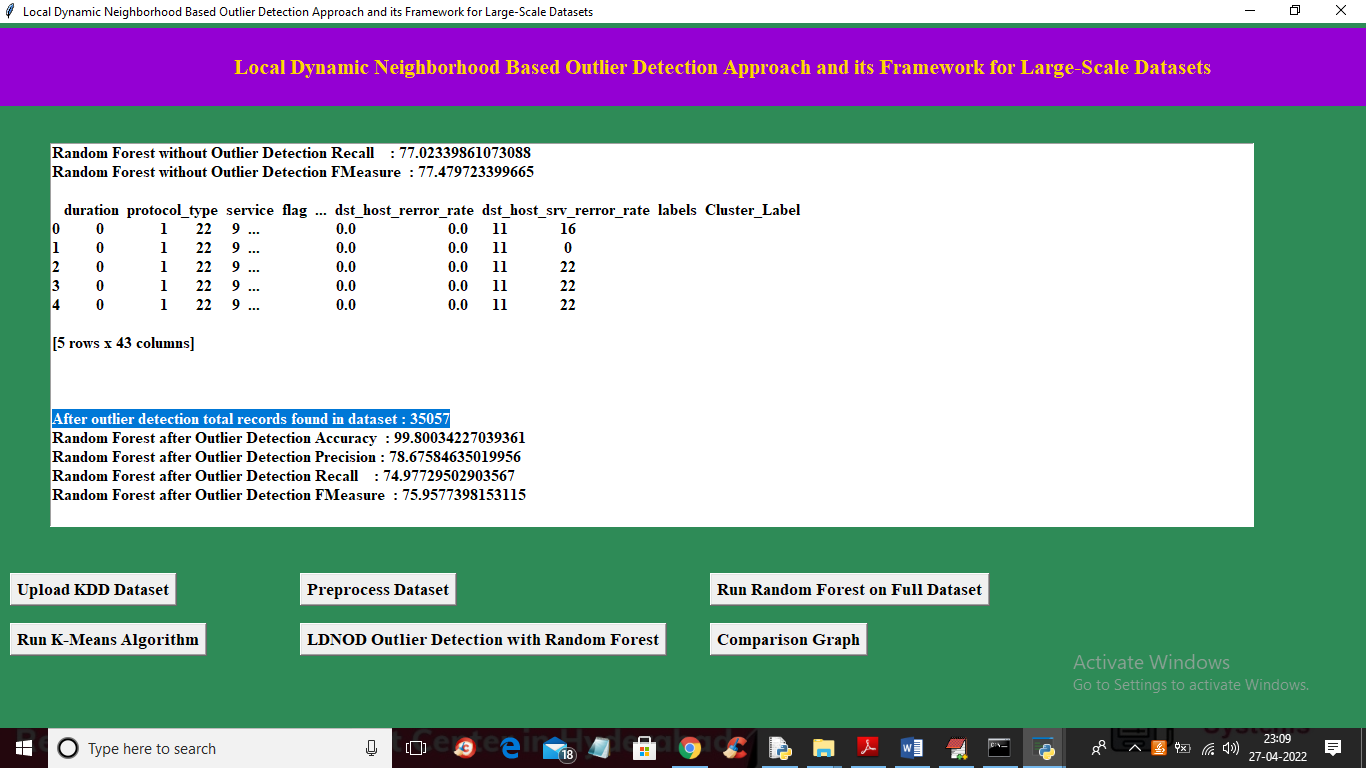
In above screen we can see all values are converted to numeric data and in blue line we can see dataset contains 38755 records without outlier detection and now click on ‘Run Random Forest on Full Dataset’ button to train Random Forest and get below output



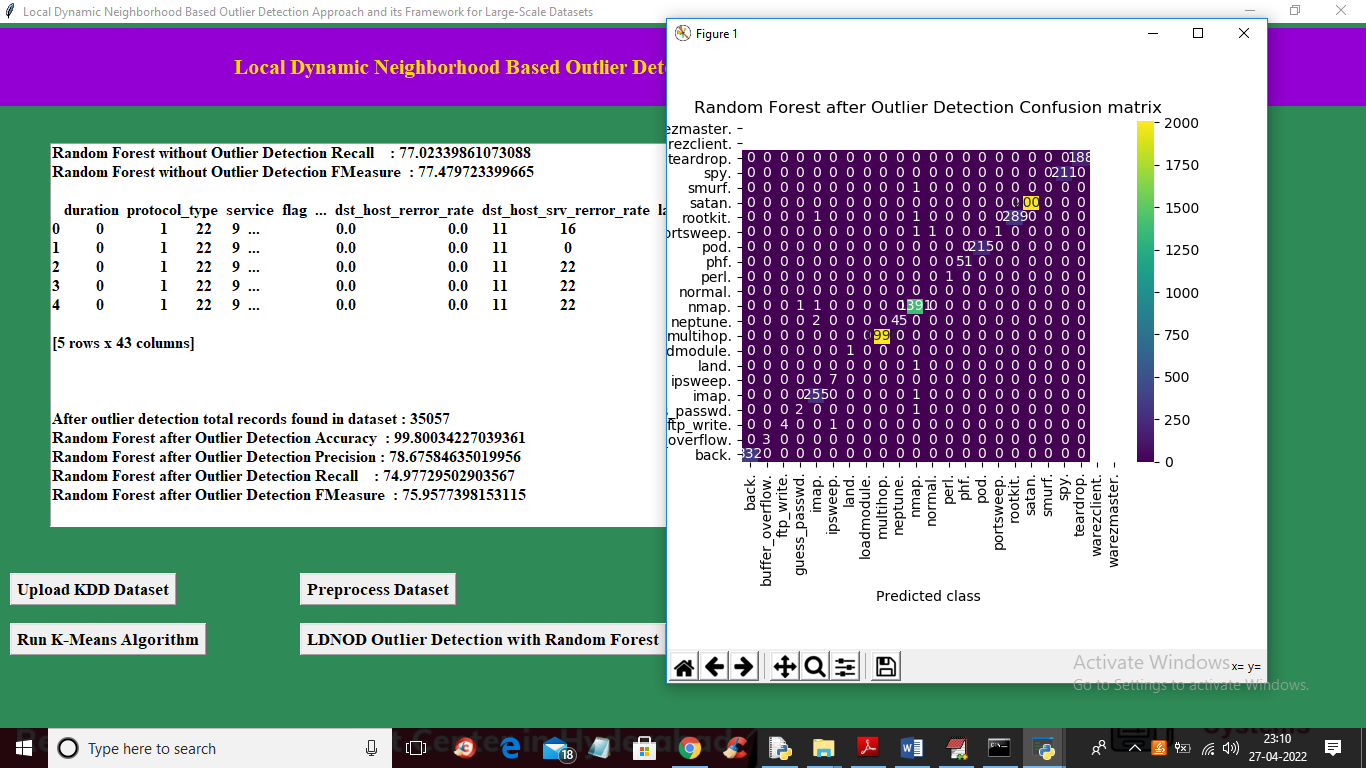
In above screen on full dataset with Random Forest we got 98% accuracy and in confusion matrix graph x-axis represents predicted attacks and y-axis represents TRUE test data attacks and in diagnol we can see both predicted and true labels are matching and now close above graph and then click on ‘Run K-Means Algorithm’ button to cluster entire dataset and get below output



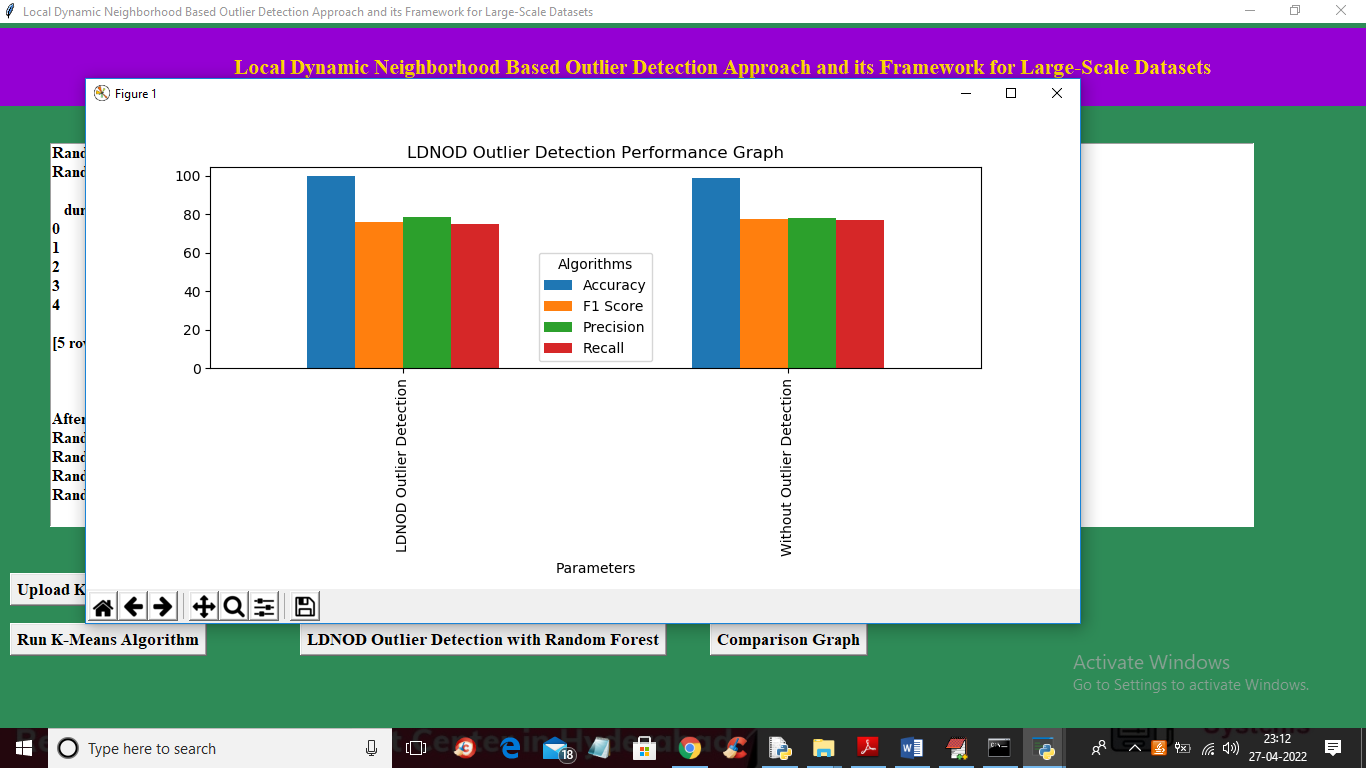
In above screen we can see cluster label added in last column and 16, 0, 22 are the cluster ID and now click on ‘LDNOD Outlier Detection with Random Forest’ button to apply LDNOD algorithm to compute similarity score between current record and neighbour records and if score is high then we will take that record and get below accuracy



In above screen in blue colour text we can see after applying outlier dataset size reduced to 35000 from 38000 and after applying outlier we got accuracy with same random forest as 99.80 and below is the confusion matrix graph



In above graph after applying outlier confusion diagnol boxes contains more number of correct prediction and now close above graph and then click on ‘Comparison Graph’ to get below graph



In above graph x-axis represents technique name and y-axis represents accuracy and other metrics such as precision, recall and FSCORE with different colour bars and in both algorithm we got high accuracy after outlier detection.