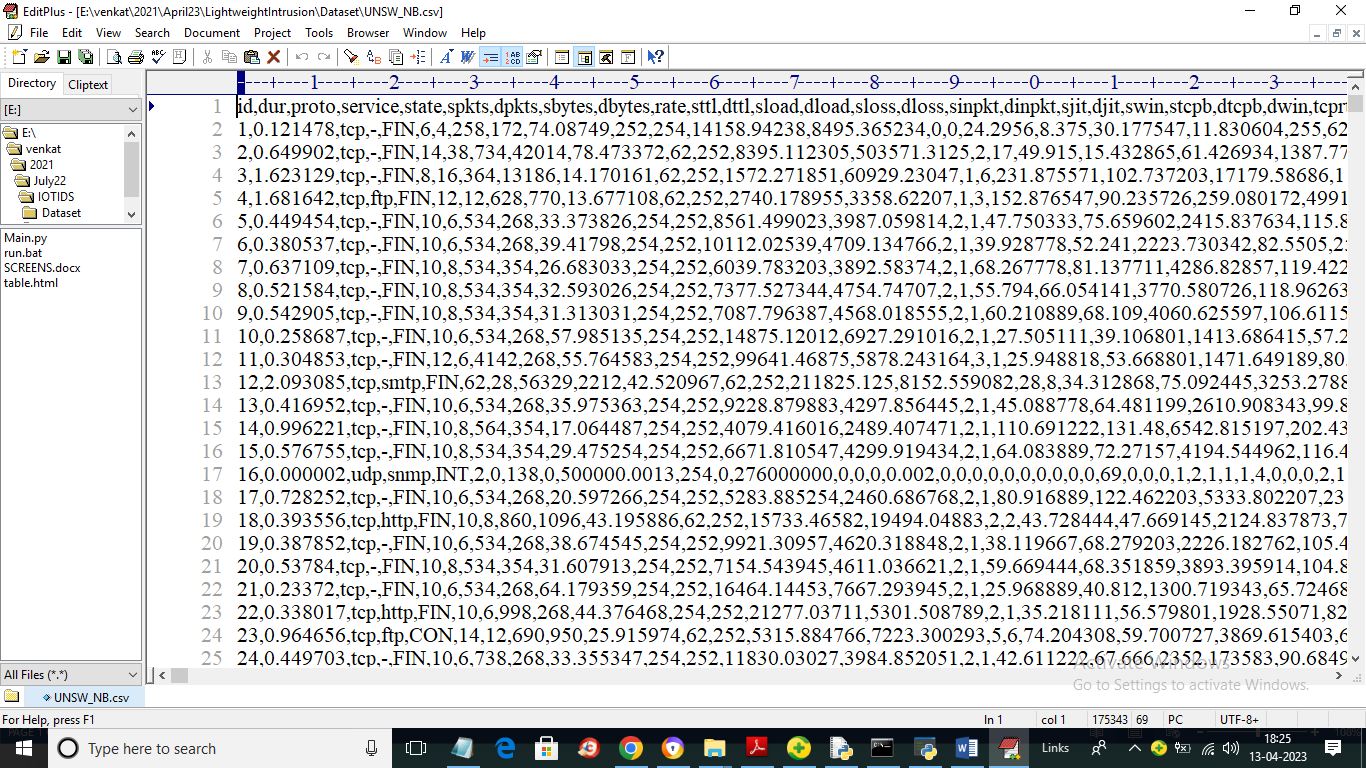
A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things

Now-a-days all devices are connected to internet and all are vulnerable to attacks and to prevent such attacks many machine learning and deep learning algorithms are introduced and they gain lots of popularity and success ratio in attack detection. This intrusion detection we cannot employ on small IOT (internet of things) devices as they runs on battery and have limited computation resources and cannot compute deep learning models to detect attacks. Often network traffic contains huge parameters and this parameters cannot be processed with little resources.

To overcome from above issue author of this paper introducing lightweight Neural Network called LNN which utilize compress neural network in the from CONV1D which took less computation and resources compare to CONV2D and 3D. To deals with huge parameters author employing PCA (principal component analysis) dimensionality reduction algorithm which select only important parameters of features and then skip all unimportant features.

Author has compare propose LNN algorithm with SVM, Random Forest, and CNN with and without PCA. Here we are implementing SVM, Random Forest and propose LNN as it’s difficult to implement all.

To train and test performance of all algorithms author has used UNSW-NB15 dataset and we are also using same dataset. Below screen showing dataset details



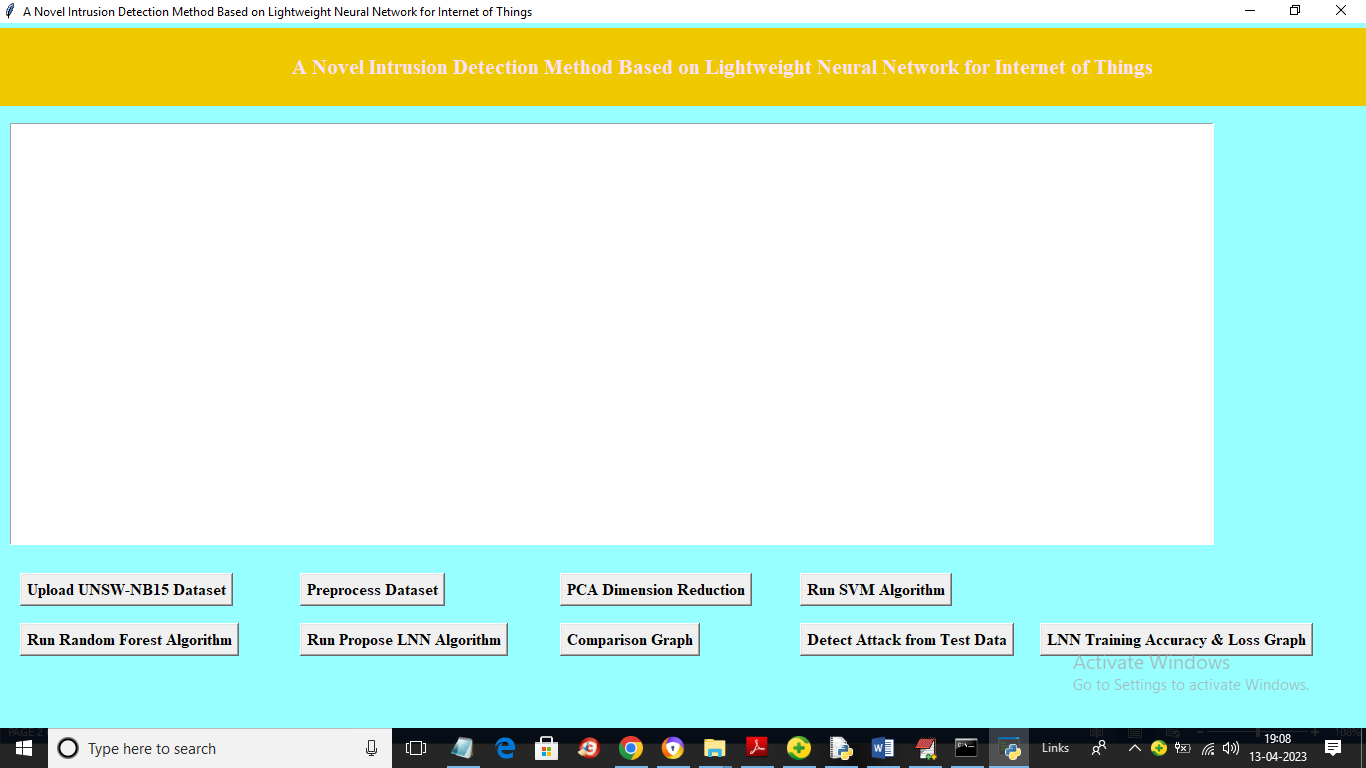
In above screen first row represents dataset column names and remaining rows represents dataset values. So by using above dataset we will train and test all algorithms.

To implement this project we have designed following modules

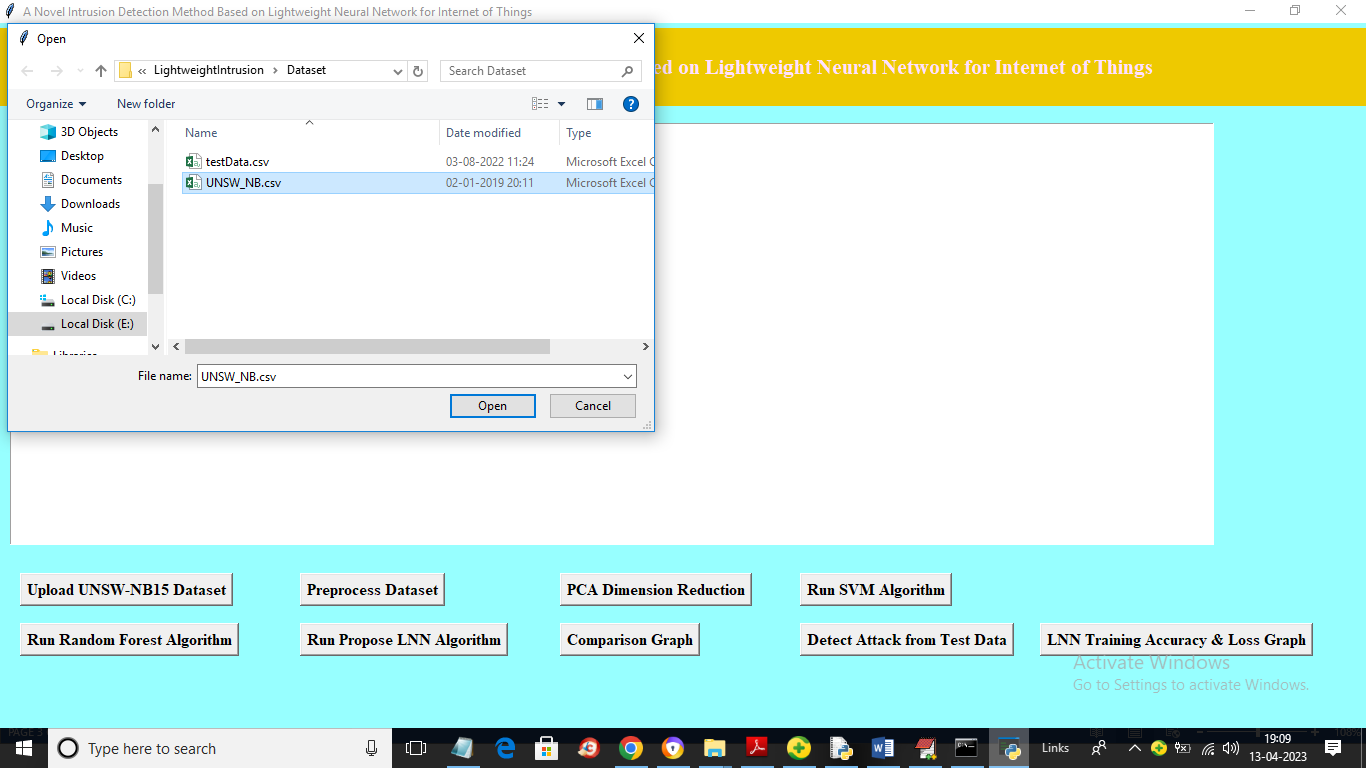
1. Upload UNSW-NB15 Dataset: using this module we will upload dataset to application and then find and plot number of normal and attack records.
2. Preprocess Dataset: dataset contains numeric and non-numeric data but machine learning accept only numeric data so by employing label encoder class we will convert all non-numeric data into numeric data. Replace all missing values with 0
3. PCA Dimension Reduction: all processed records will be input to PCA which will select all relevant features and drop all irrelevant features as dimensionality reduction. Reduced features will be split into train and test where application using 80% dataset for training and 20% for testing algorithm performance
4. Run SVM Algorithm: 80% training data will be input to SVM algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy
5. Run Random Forest Algorithm: 80% training data will be input to Random Forest algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy
6. Run Propose LNN Algorithm: 80% training data will be input to Propose LNN algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy. This algorithm will used PCA reduced and compressed model layers for training and testing model performance
7. Comparison Graph: using this module we will plot comparison graph between all algorithms
8. Detect Attack from Test Data: using this module we will upload test data and then LNN algorithm will predict weather test data is normal or contains attack

SCREEN SHOTS

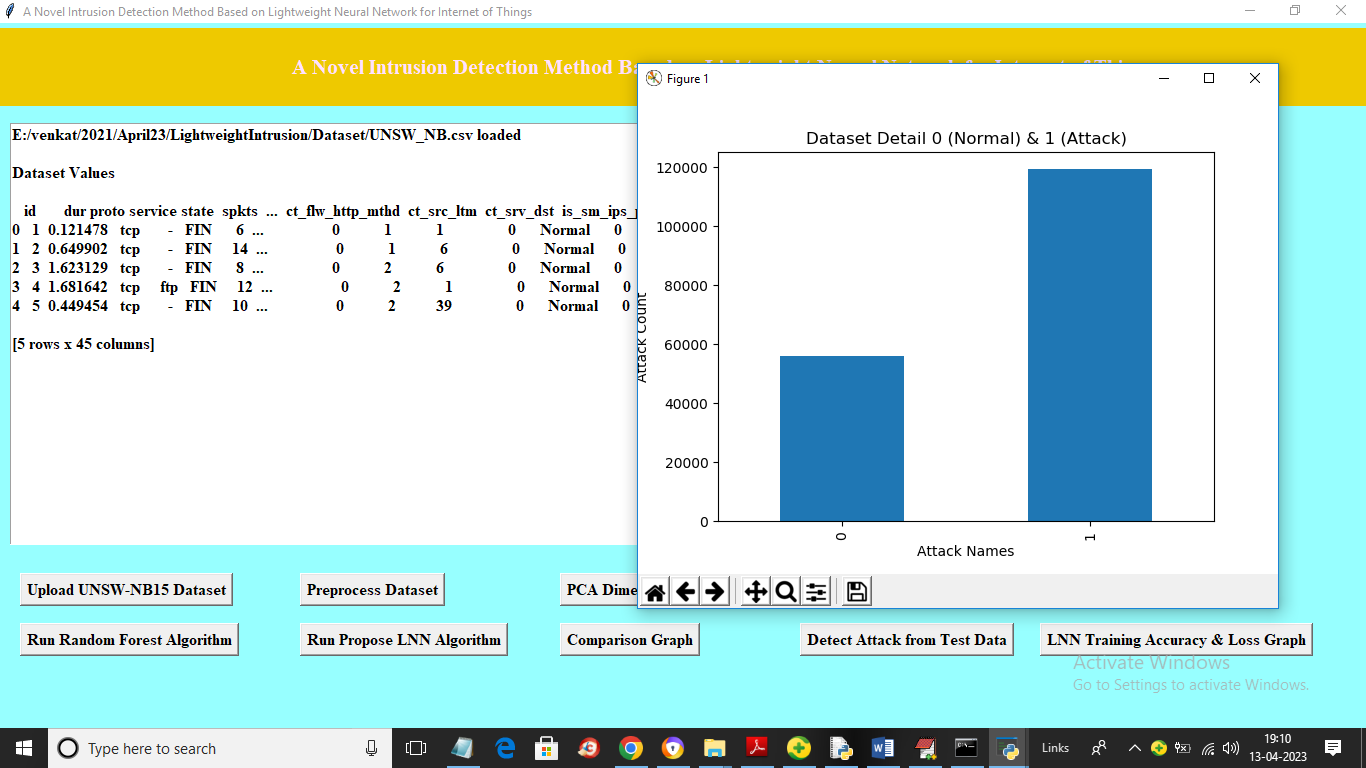
To run project double click on run.bat file to get below screen



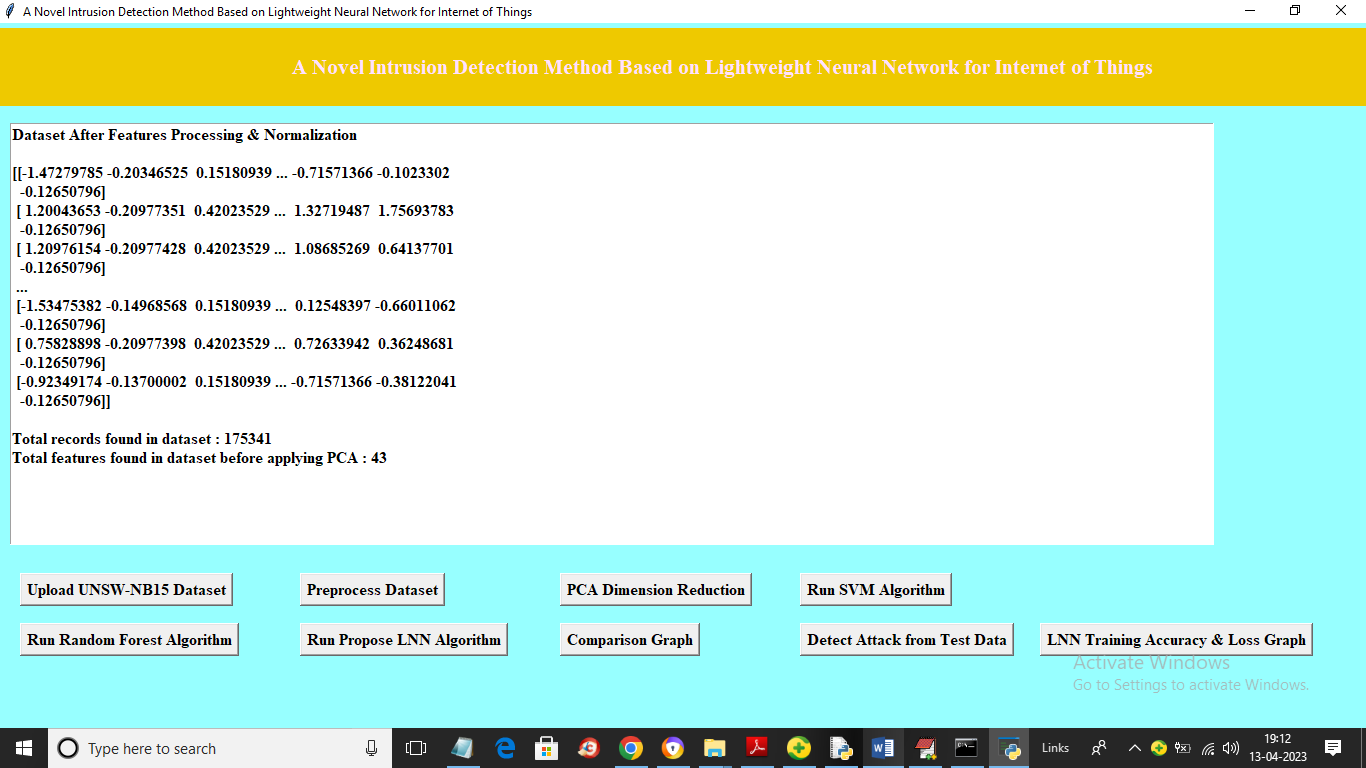
In above screen click on ‘Upload UNSW-NB15 Dataset’ button to upload dataset and get below output



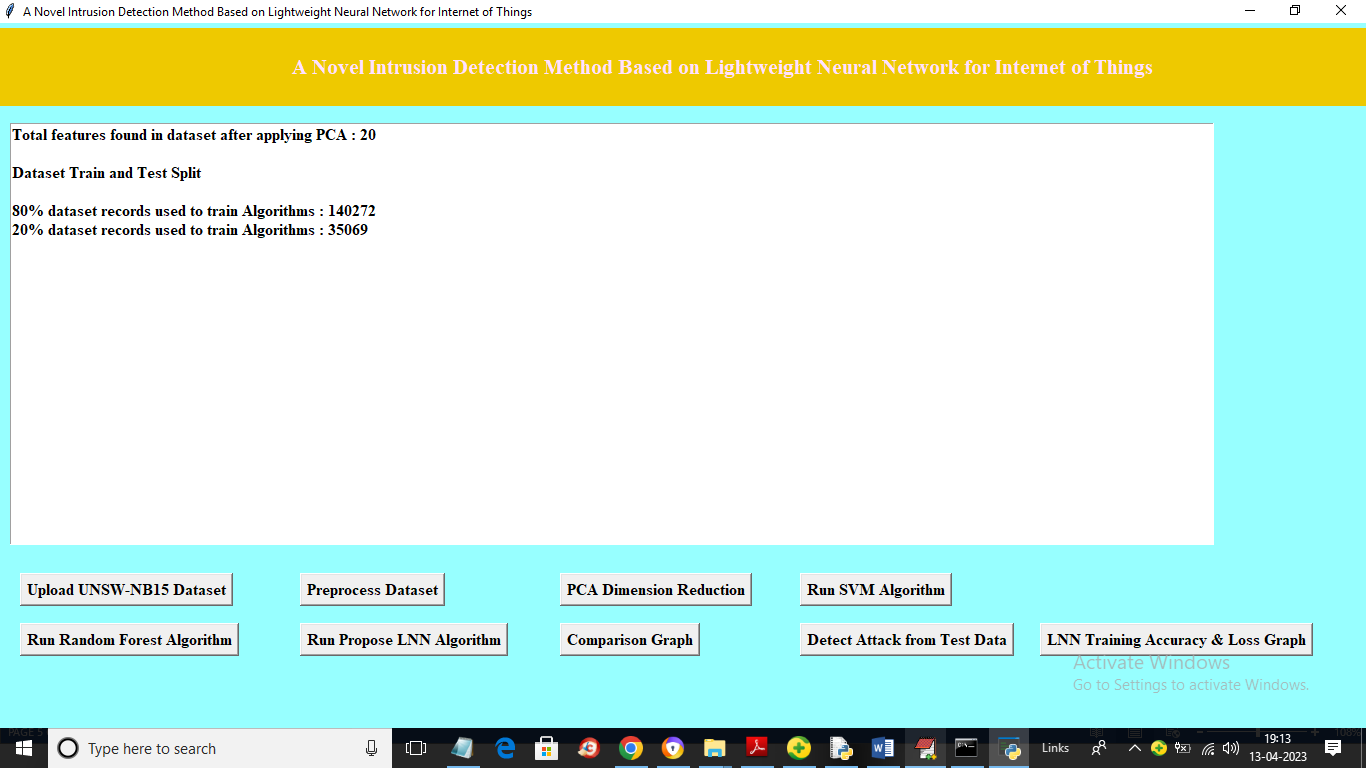
In above screen selecting and uploading dataset file and then click on ‘Open’ button to load dataset and get below output



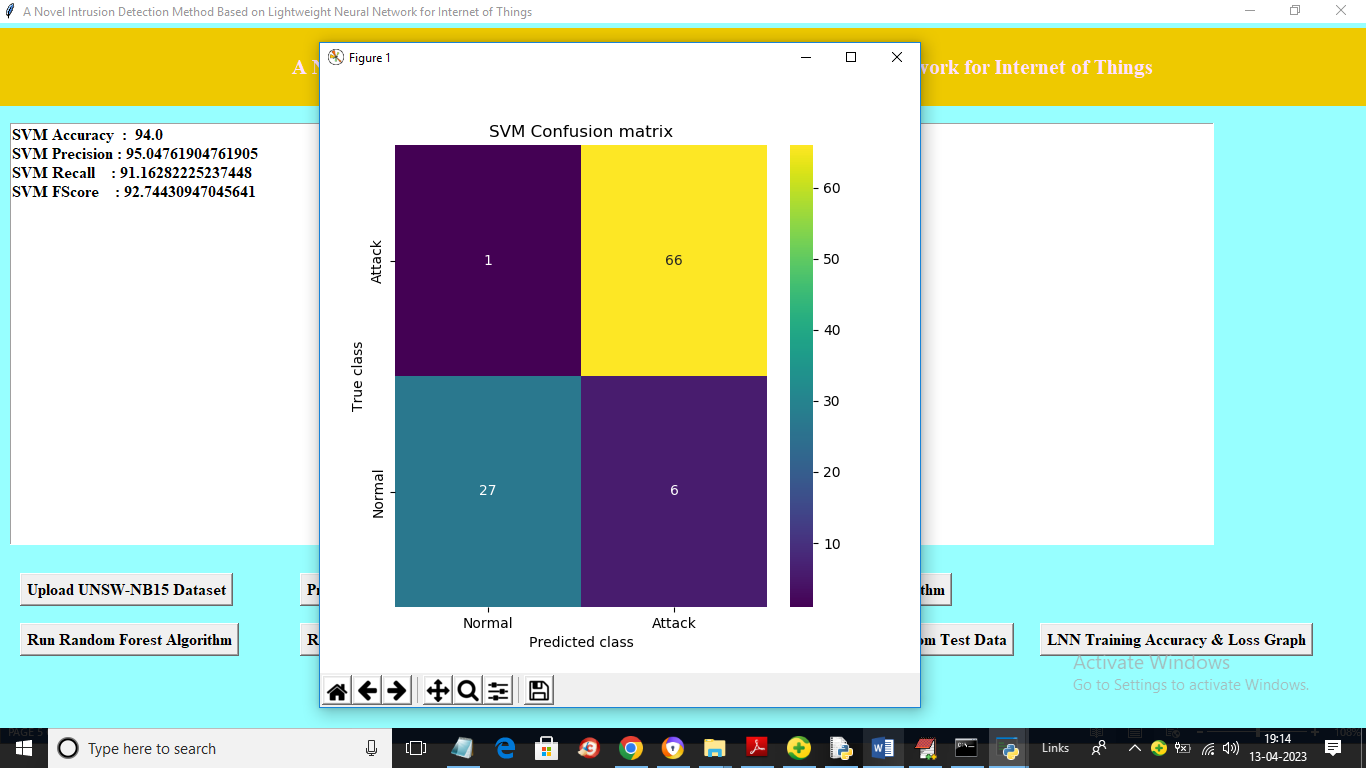
In above screen dataset loaded and we can see dataset contains both numeric and non-numeric data but algorithms accept only numeric data so we need to process data to convert to numeric. In above graph x-axis represents 0 as NORMAL and 1 as ATTACK and y-axis represents number of records available in normal and attack category. Now close above graph and then click on ‘Preprocess Dataset’ button to process dataset and get below output



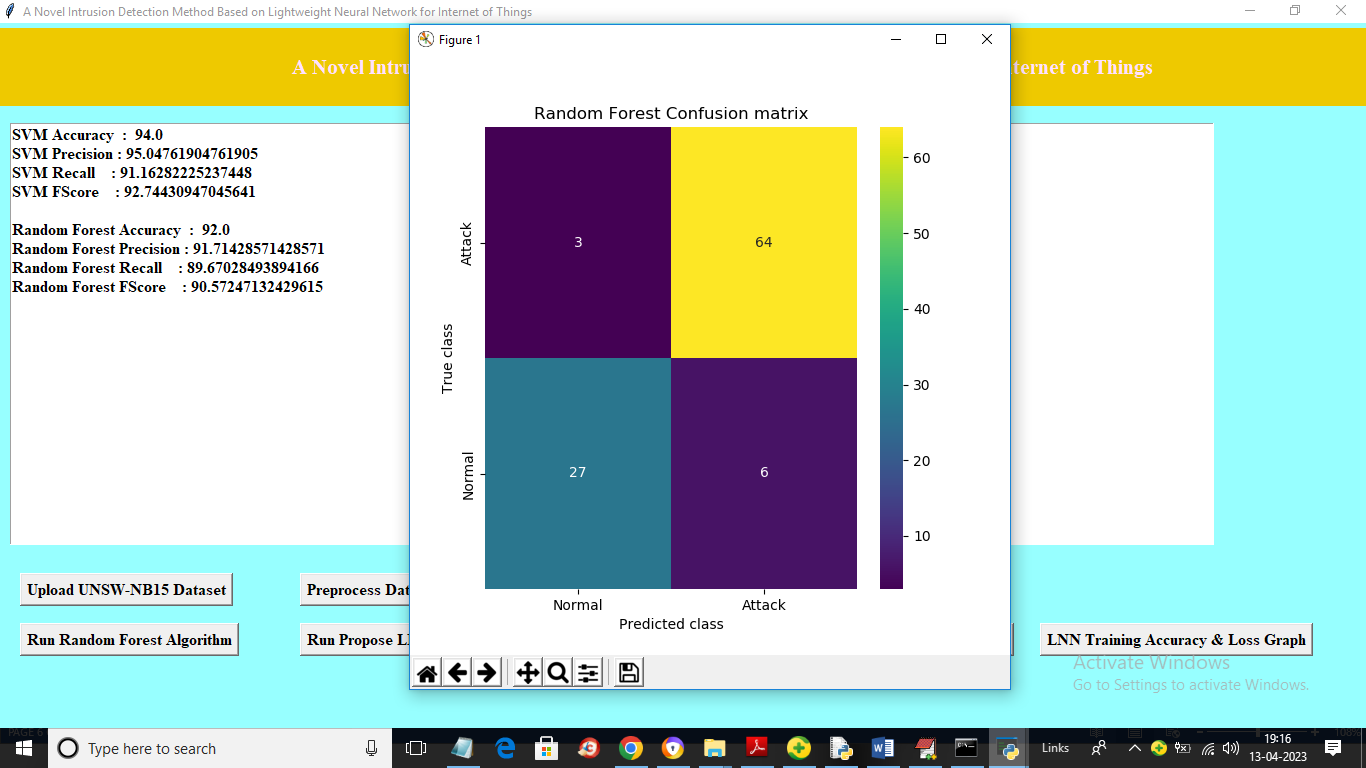
In above screen entire dataset converted to numeric format and in last two lines we can see dataset contains 175341 records and contains 43 features or column before applying PCA. Now click on ‘PCA Dimension Reduction’ button to reduce features and get below output



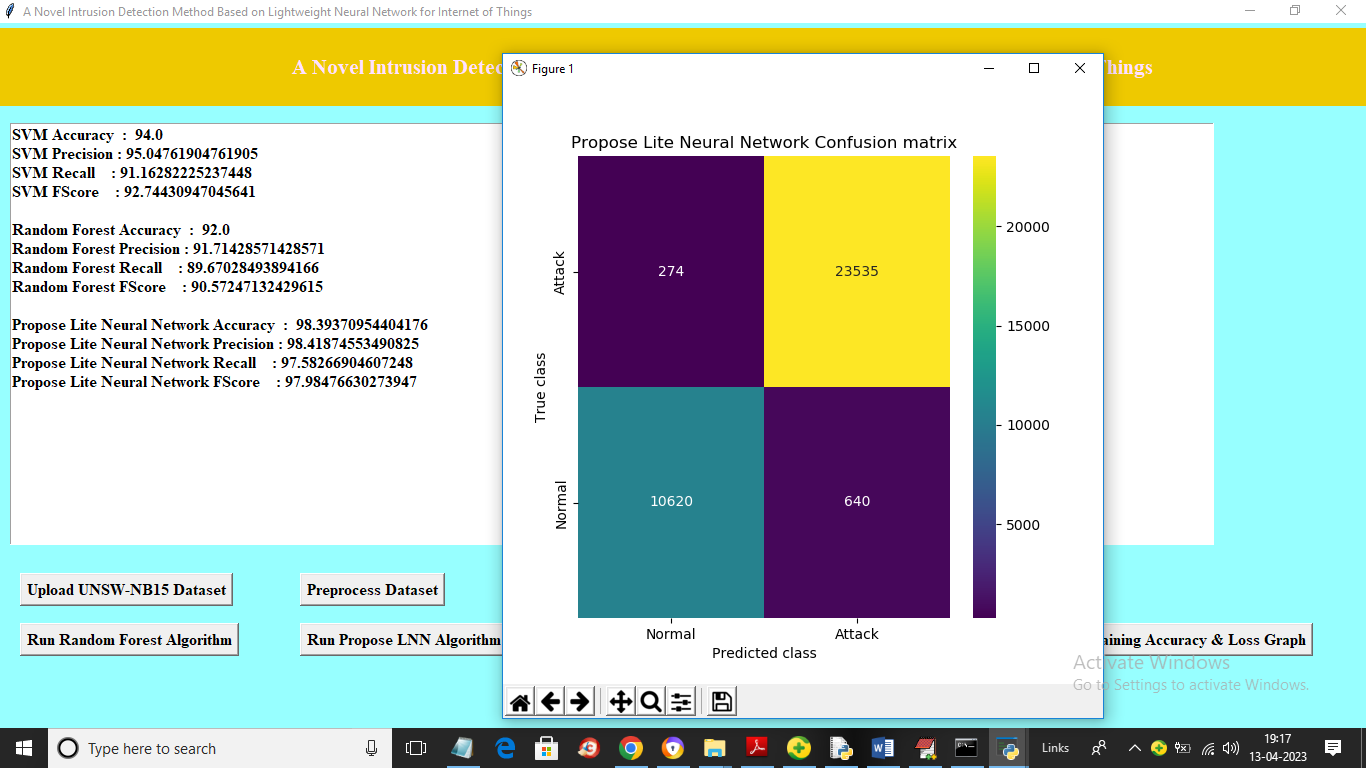
In above screen after applying PCA we got 20 features from 43 and we can see total records used for training and testing and now click on ‘Run SVM Algorithm’ button to train SVM and get below output



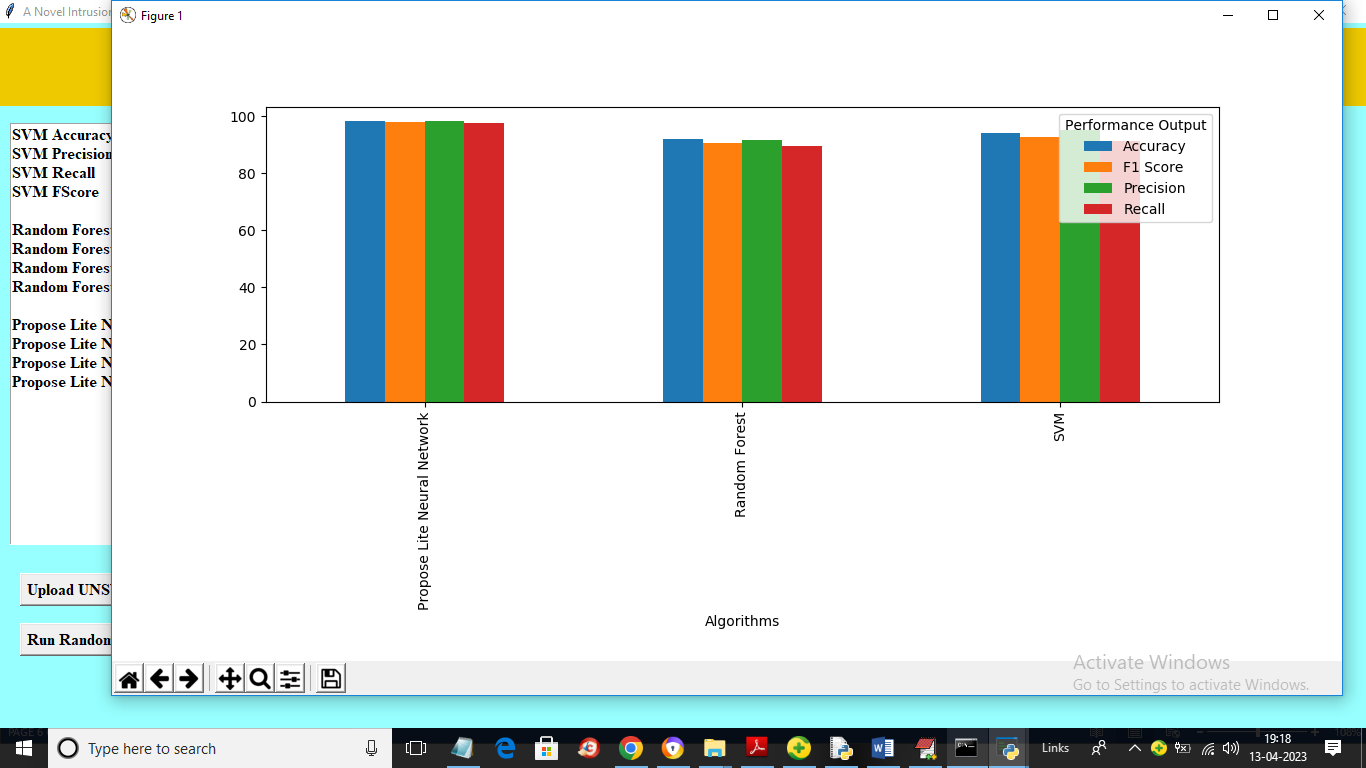
In above screen with SVM we got 94% accuracy and we can see other metrics like precision, recall and FSCORE. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents TRUE labels and light green and yellow boxes contains correct prediction count and blue boxes contains incorrect prediction count. Now click on ‘Run Random Forest’ button to get below output



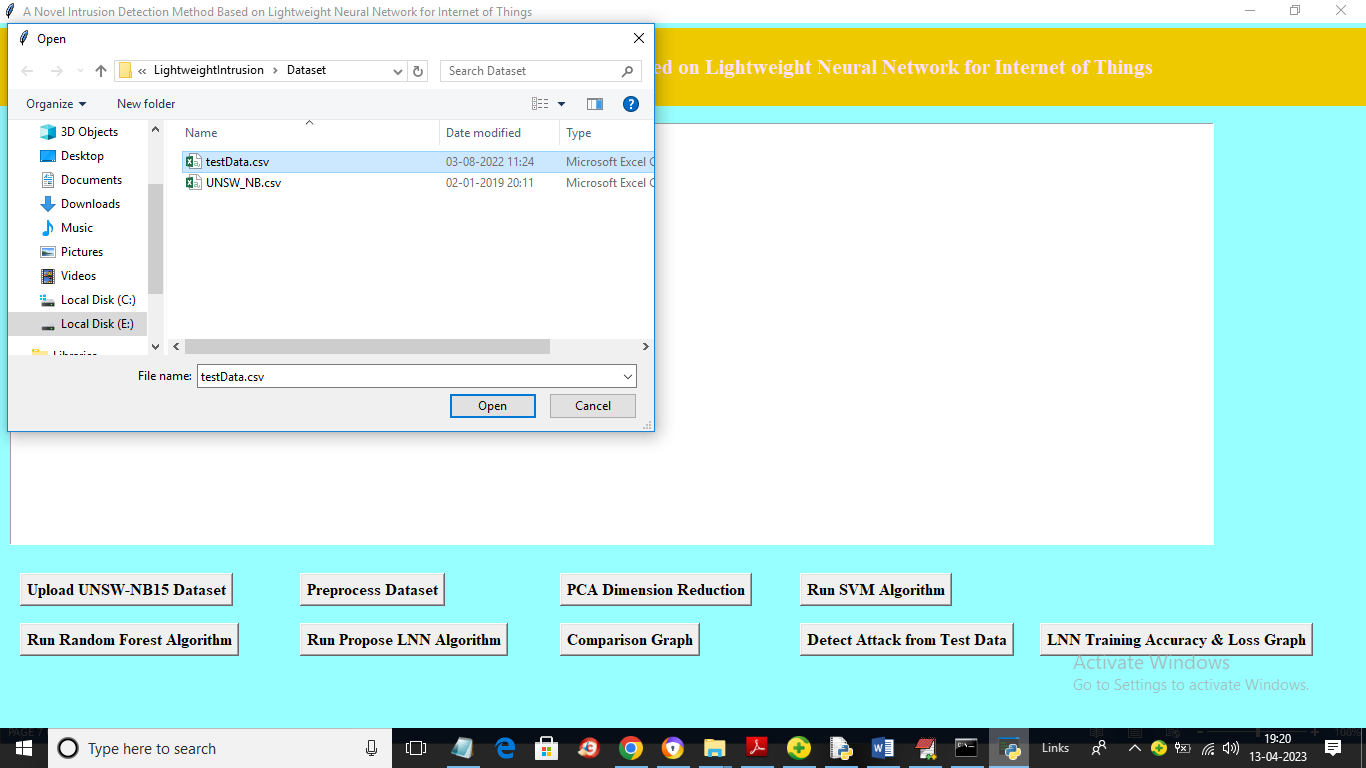
In above screen with Random Forest we got 92% accuracy and now click on ‘Run Propose LNN Algorithm’ button to run propose algorithm and get below output



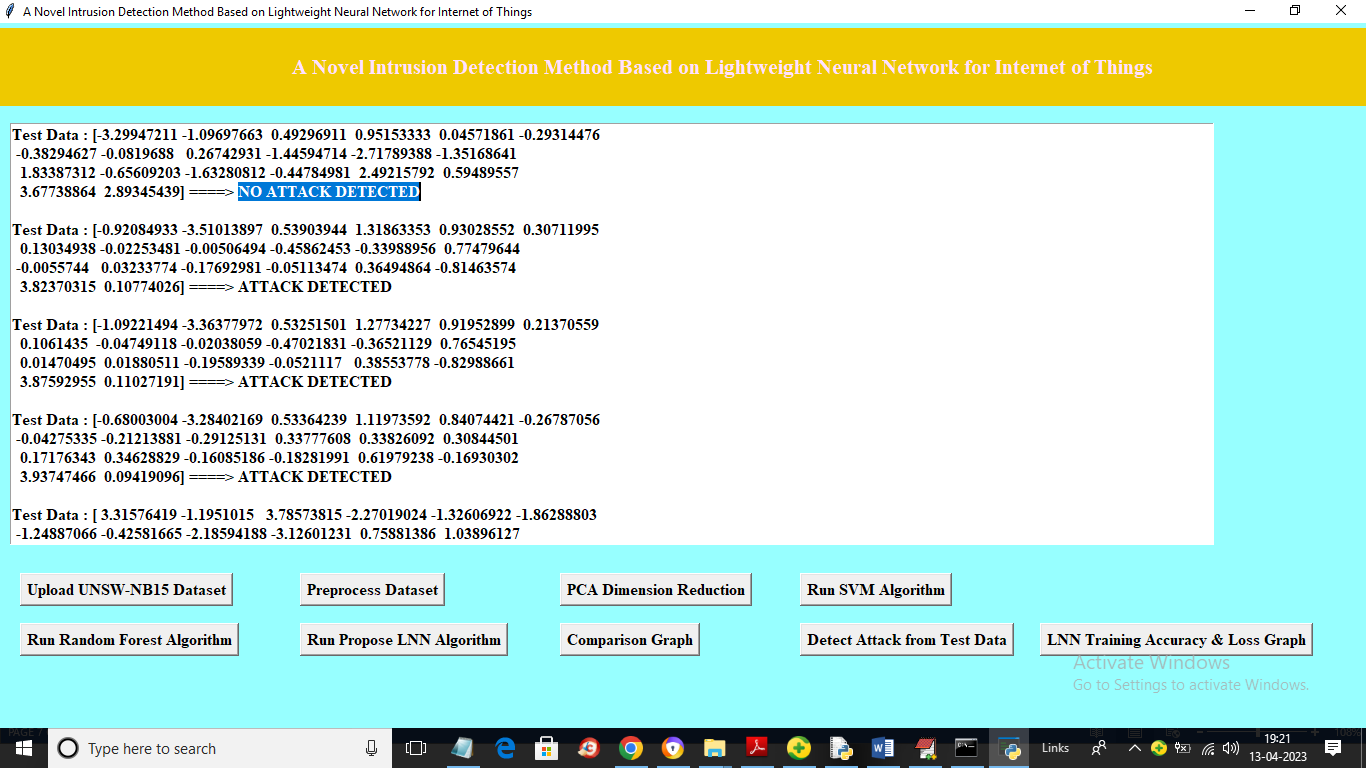
In above screen with propose LNN algorithm we got 98.39% accuracy and we can see confusion graph also and this algorithm runs on entire dataset and now click on ‘Comparison Graph’ button to get below graph



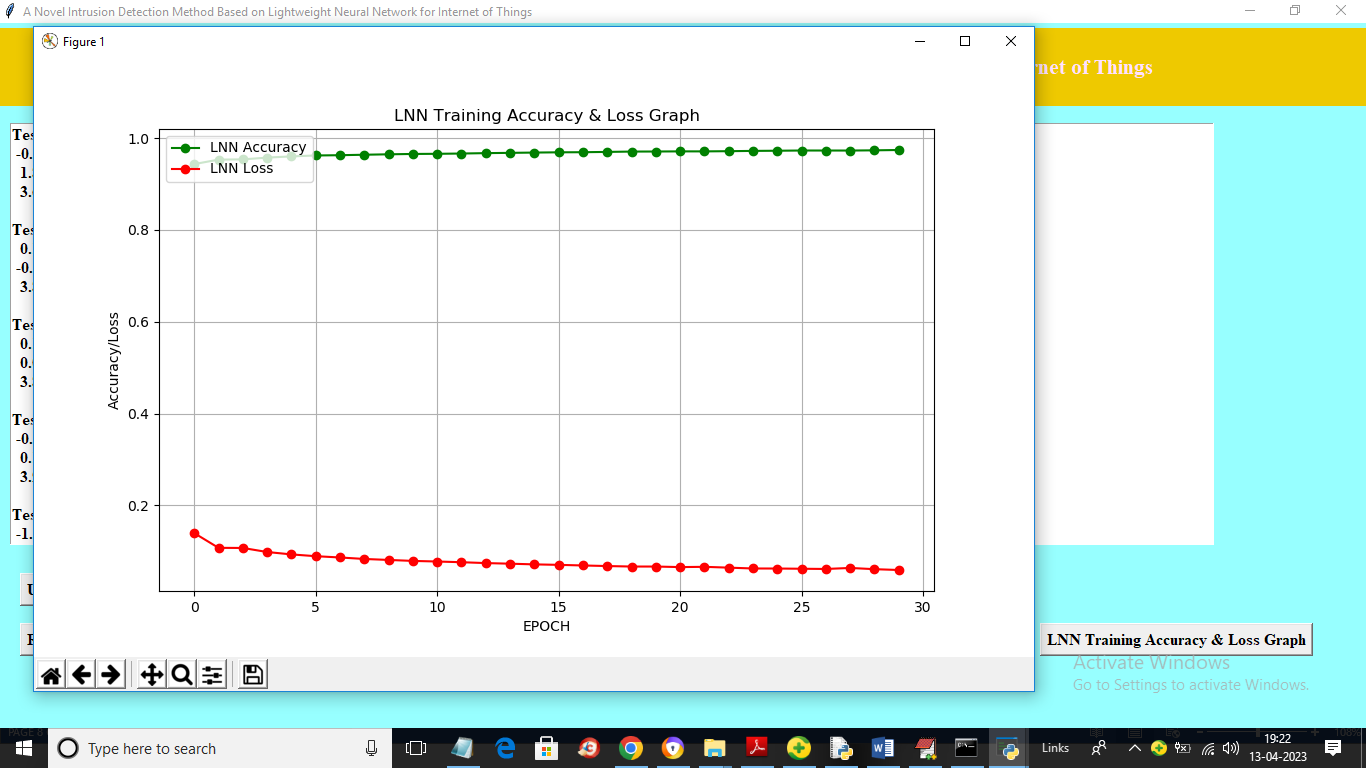
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms propose LNN got high values and now click on ‘Detect Attack from Test Data’ button to upload test data and get below output



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



In above screen in square bracket we can see test data and after arrow symbol =🡺 we can see predicted output as ‘No Attack Detected’ or ‘Attack Detected’ and now click on ‘LNN Training Accuracy and Loss Graph’ button to get below graph



In above graph x-axis represents training epoch and y-axis represents loss and accuracy and red line represents loss and green line represents accuracy and with each increasing epoch accuracy got increase and reached closer to 1 and loss got decrease and reached closer to 0.