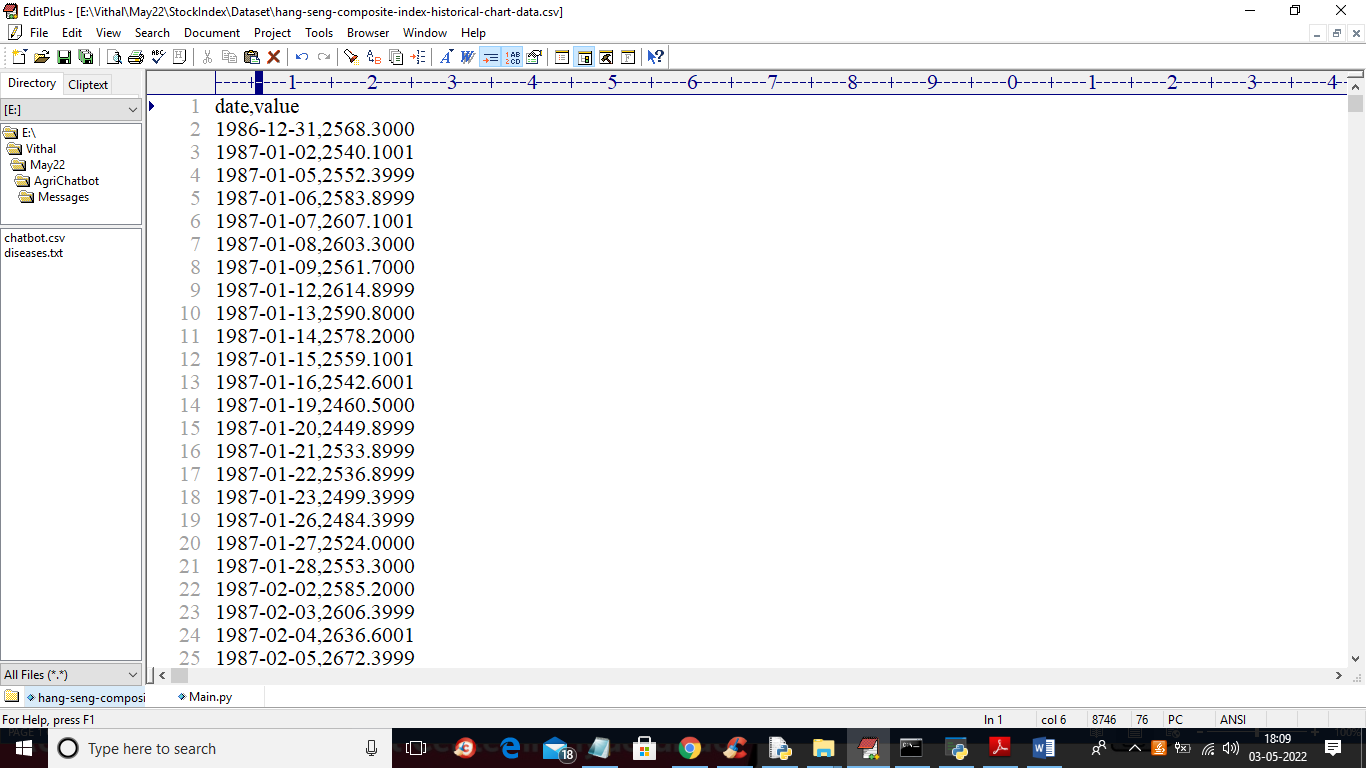
Discovery and Prediction of Stock Index Pattern via Three-Stage Architecture of TICC, TPA-LSTM and Multivariate LSTM-FCNs

In this paper to predict stock index author is using multivariate time series (data which contains time information) data and then extracting repeated values from that data by using Toeplitz Inverse Covariance-Based Clustering (TICC) as this algorithm put similar data in same cluster. Patterns will be extracted by using Temporal Pattern Attention and Long-Short-Term Memory (TPA-LSTM) and then extracted pattern will get trained with Multivariate LSTM-FCNs (fully connected network) to predict stock index.

All existing algorithms such as ARIMA work on time series data but not extract any patterns so its prediction accuracy will be low and Relative Absolute Error will be high.

Author compare propose algorithm TPA-LSTM with various existing algorithms such as SVM, Random forest and Naïve Bayes and evaluate their performance in terms of accuracy and RAE.

To implement this project author has used Hang-Sang dataset and I am also using same dataset and below screen showing dataset details



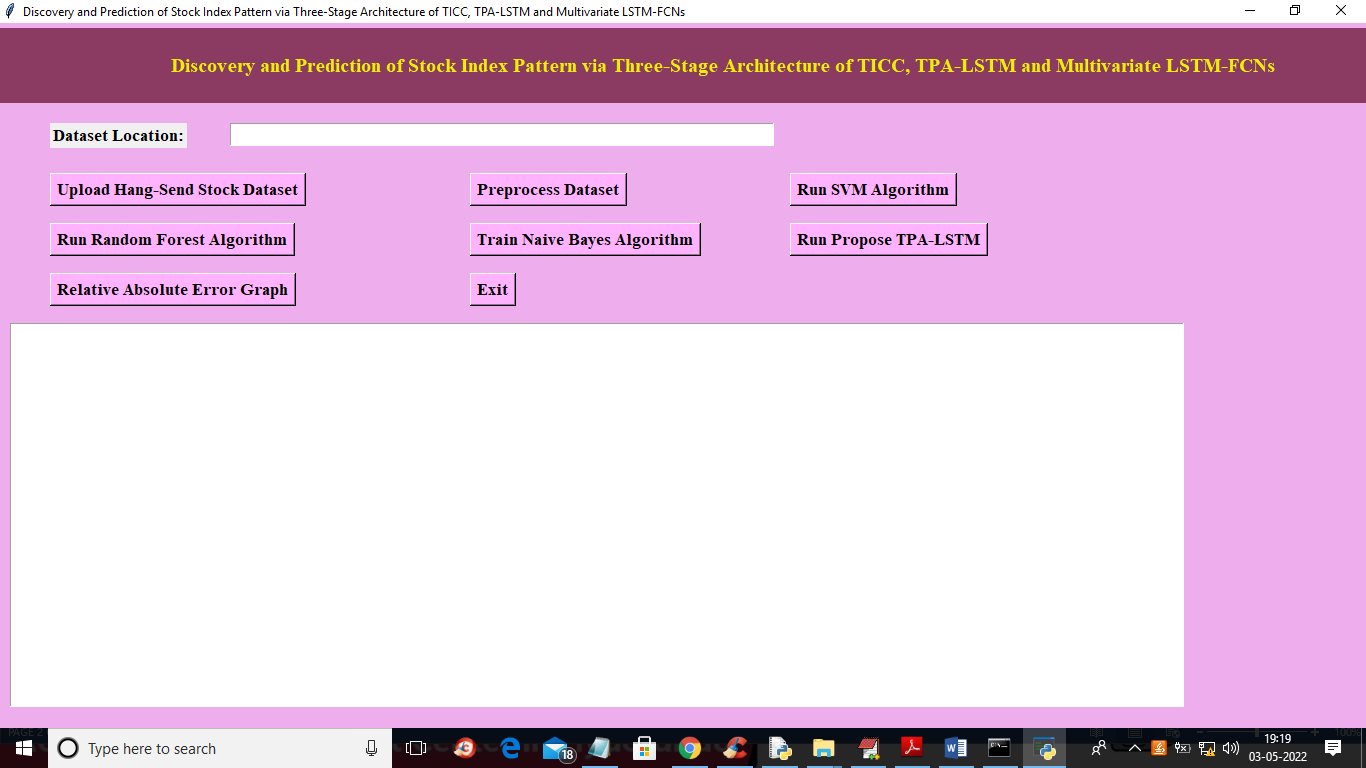
In above dataset we have date and stock value and by using this time series and stock data we will train algorithms and then predict stock index and then find difference between original stock index and predicted index as RAE error.

To implement this project we have designed following modules

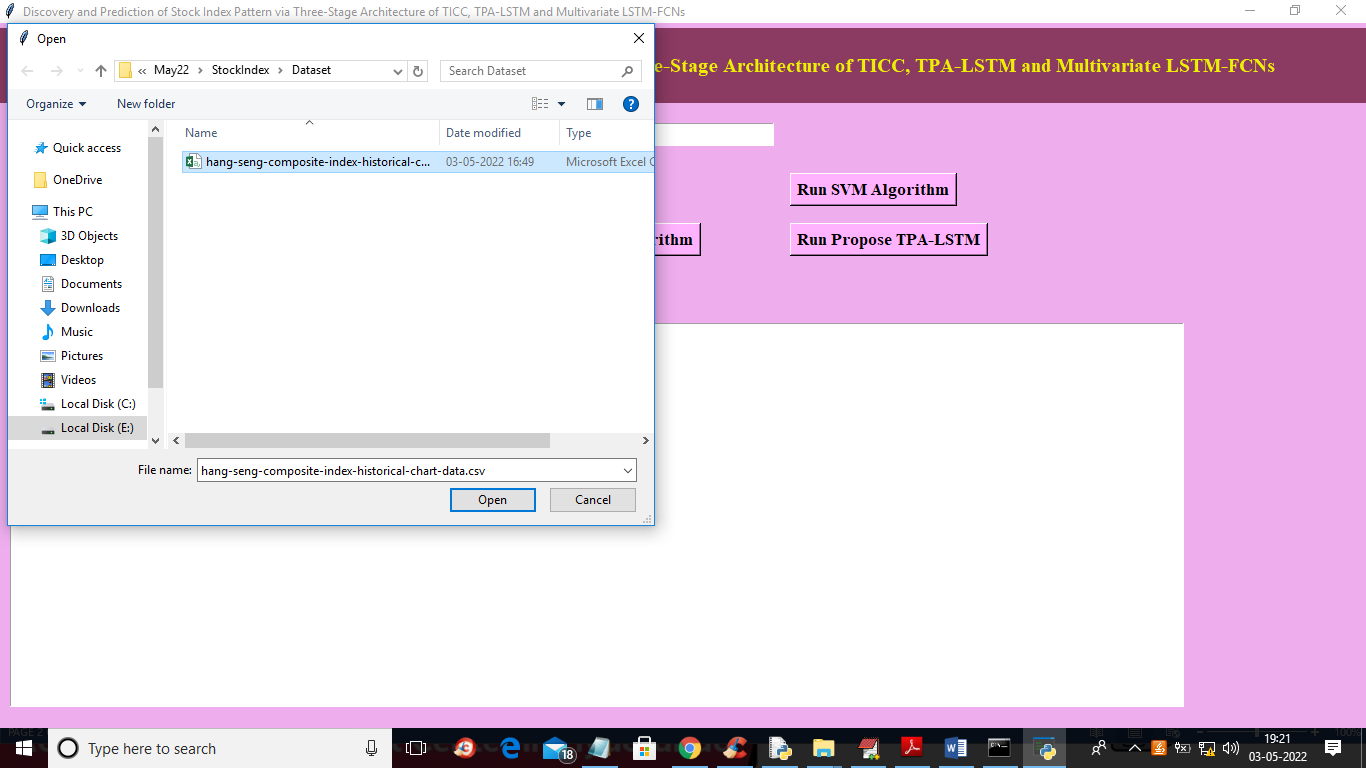
1. Upload Hang-Send Stock Dataset: using this module we will upload dataset to application
2. Preprocess Dataset: using this module we will read all dataset values and then normalize values using MIN-MAX scaler
3. Run SVM Algorithm: using this module we will split dataset into train and test and then train SVM on training dataset and then calculate accuracy and RAE on test data prediction
4. Run Random Forest Algorithm: using this module we will split dataset into train and test and then train Random Forest on training dataset and then calculate accuracy and RAE on test data prediction
5. Train Naive Bayes Algorithm: using this module we will split dataset into train and test and then train Naïve Bayes on training dataset and then calculate accuracy and RAE on test data prediction
6. Run Propose TPA-LSTM: using this module we will split dataset into train and test and then train TPA-LSTM on training dataset and then calculate accuracy and RAE on test data prediction
7. Relative Absolute Error Graph: using this module we will plot RAE (relative absolute error) graph between all algorithms.

SCREENSHOTS

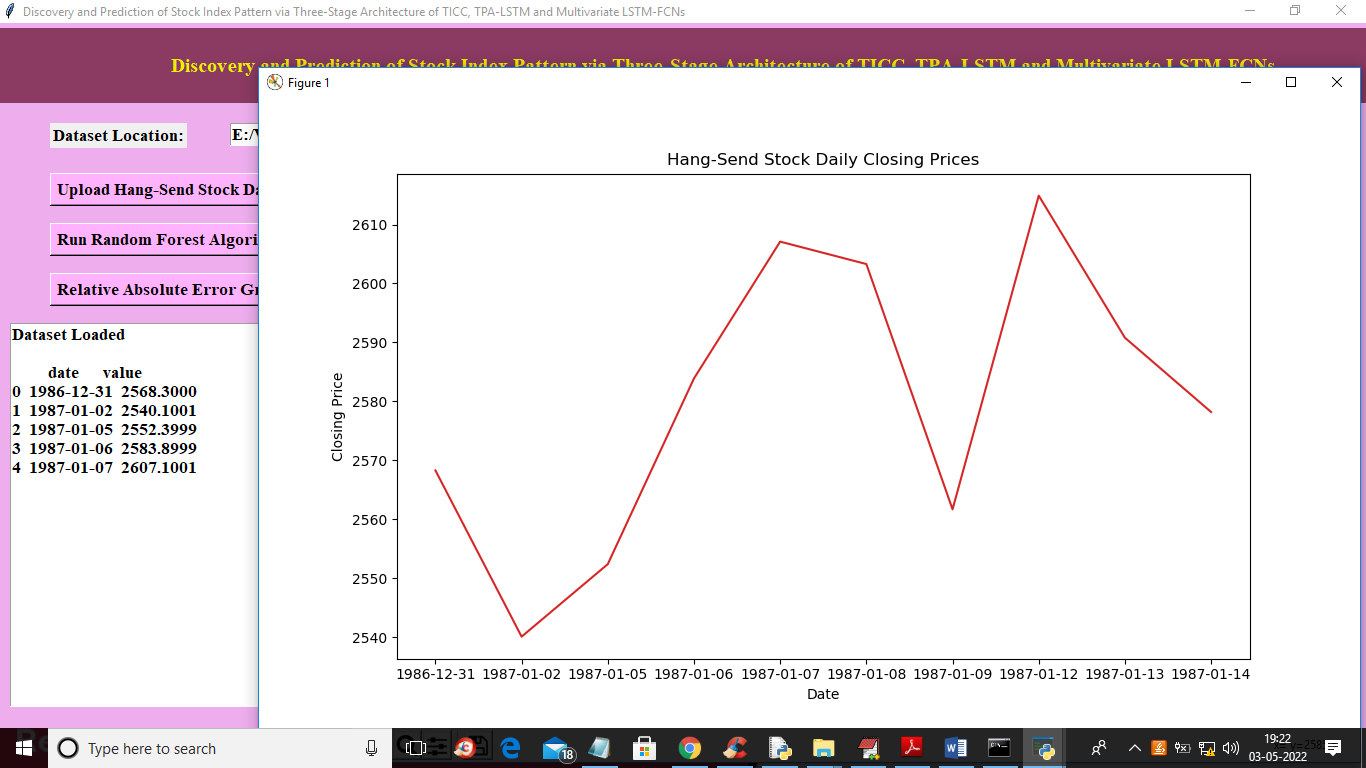
To run project double click on ‘run.bat’ file to get below screen



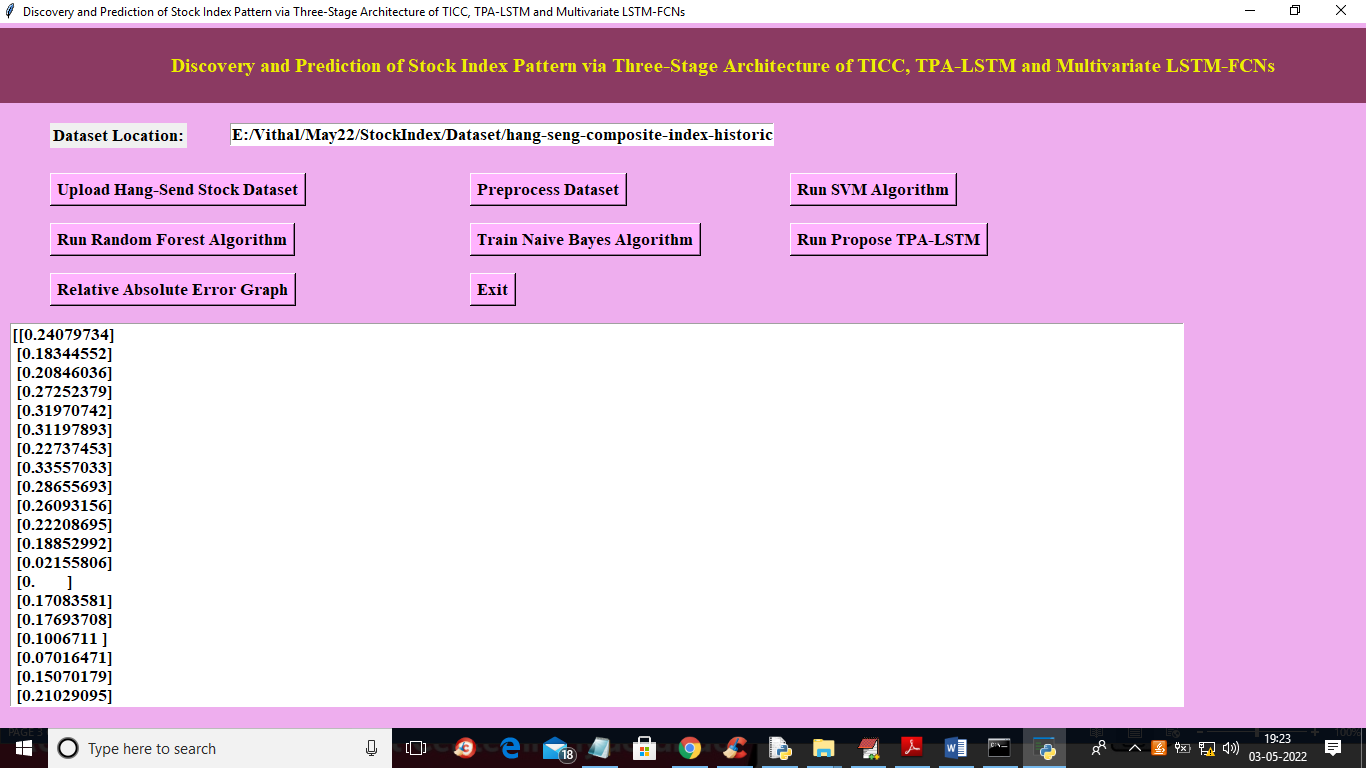
In above screen click on ‘Upload Hang Send Stock Dataset’ button to upload dataset and to get below output



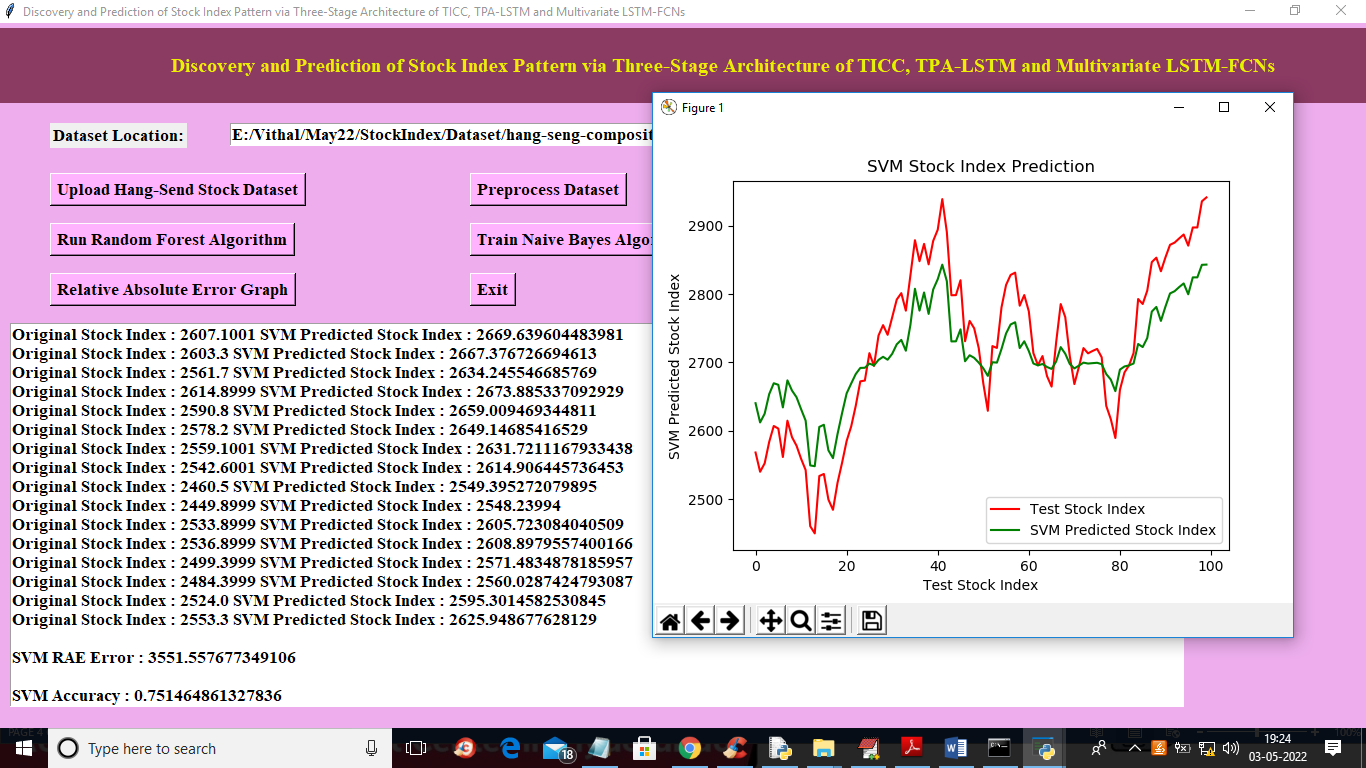
In above screen selecting and uploading ‘Hang Seng’ dataset file and then click on ‘Open’ button to load dataset and to get below output



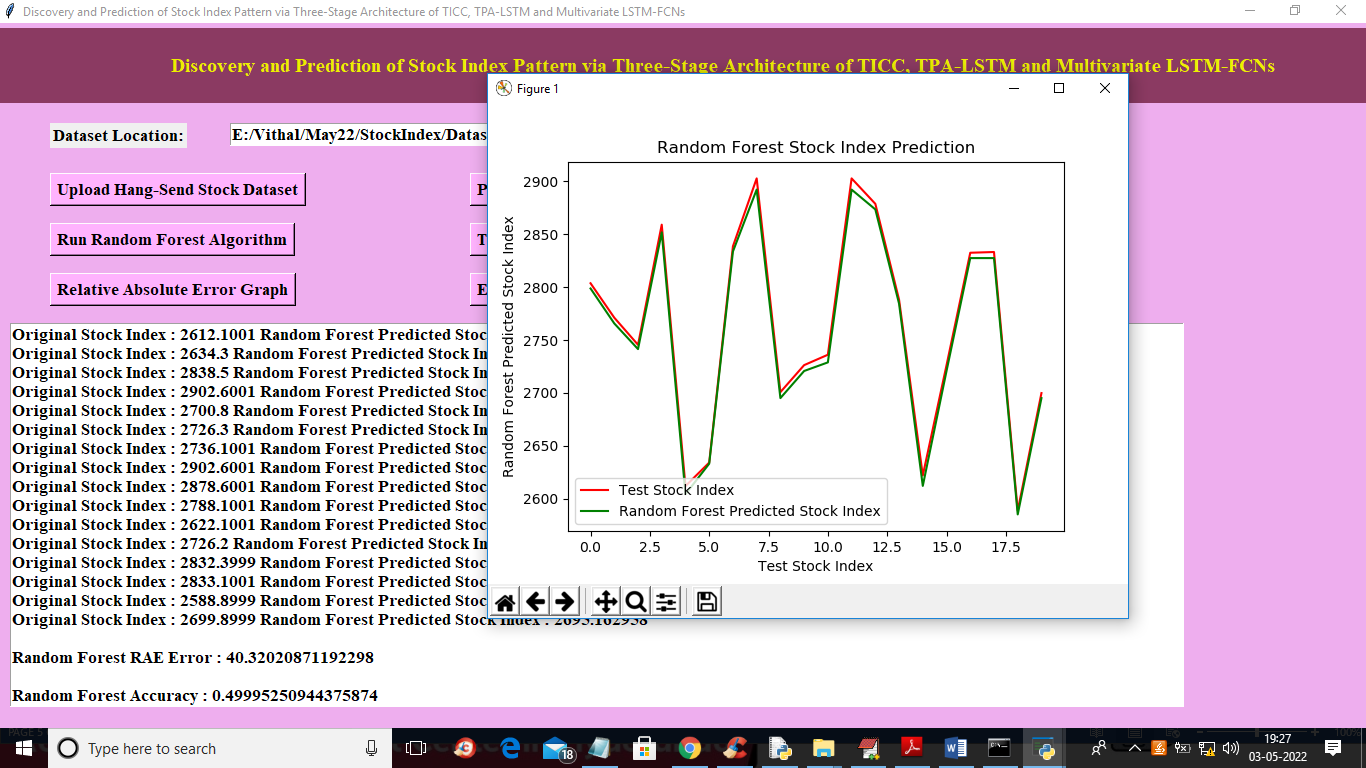
In above screen in text area we can see dataset loaded and in graph x-axis represents DATE and y-axis represent stock value on that date and now close above graph and then click on ‘Preprocess Dataset’ button to read all values and then normalize them and get below output



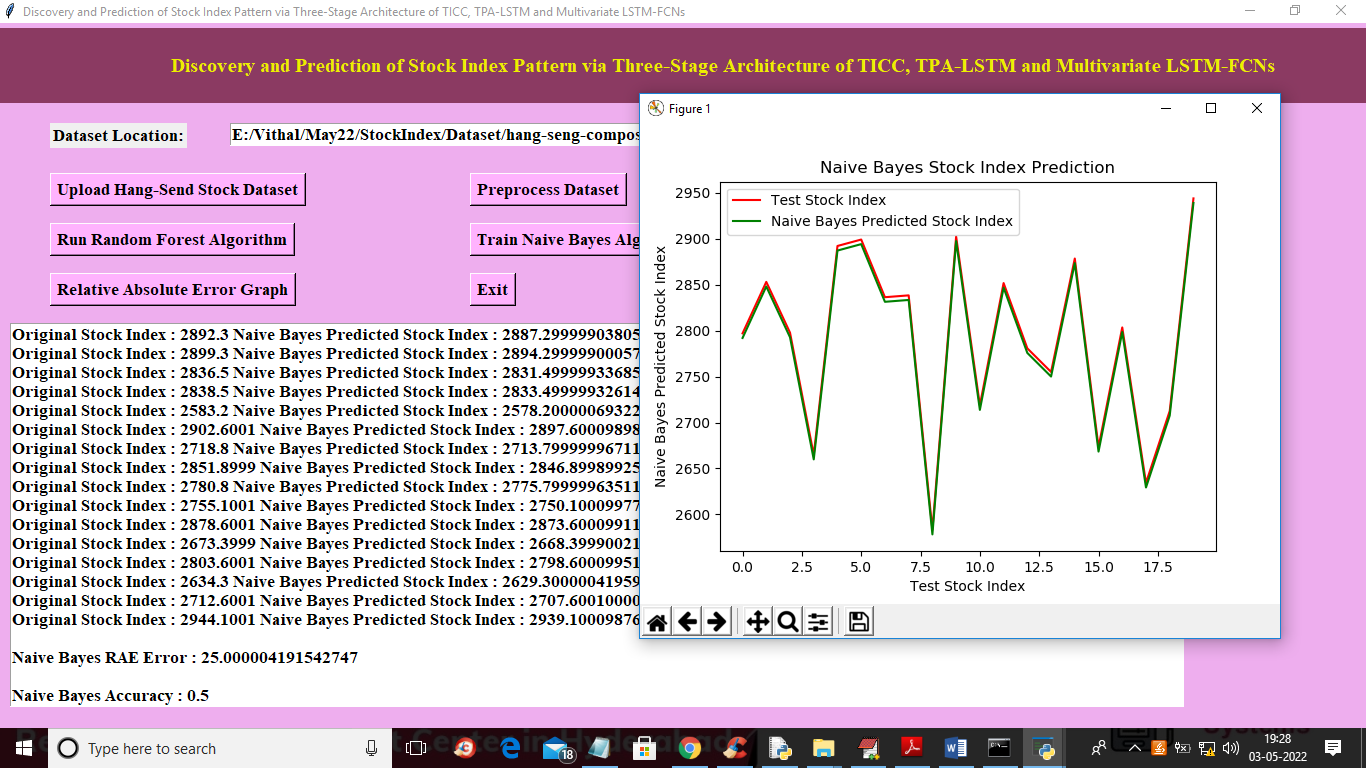
In above screen all stock values are normalize between 0 and 1 and now click on ‘Run SVM Algorithm’ button to get below output



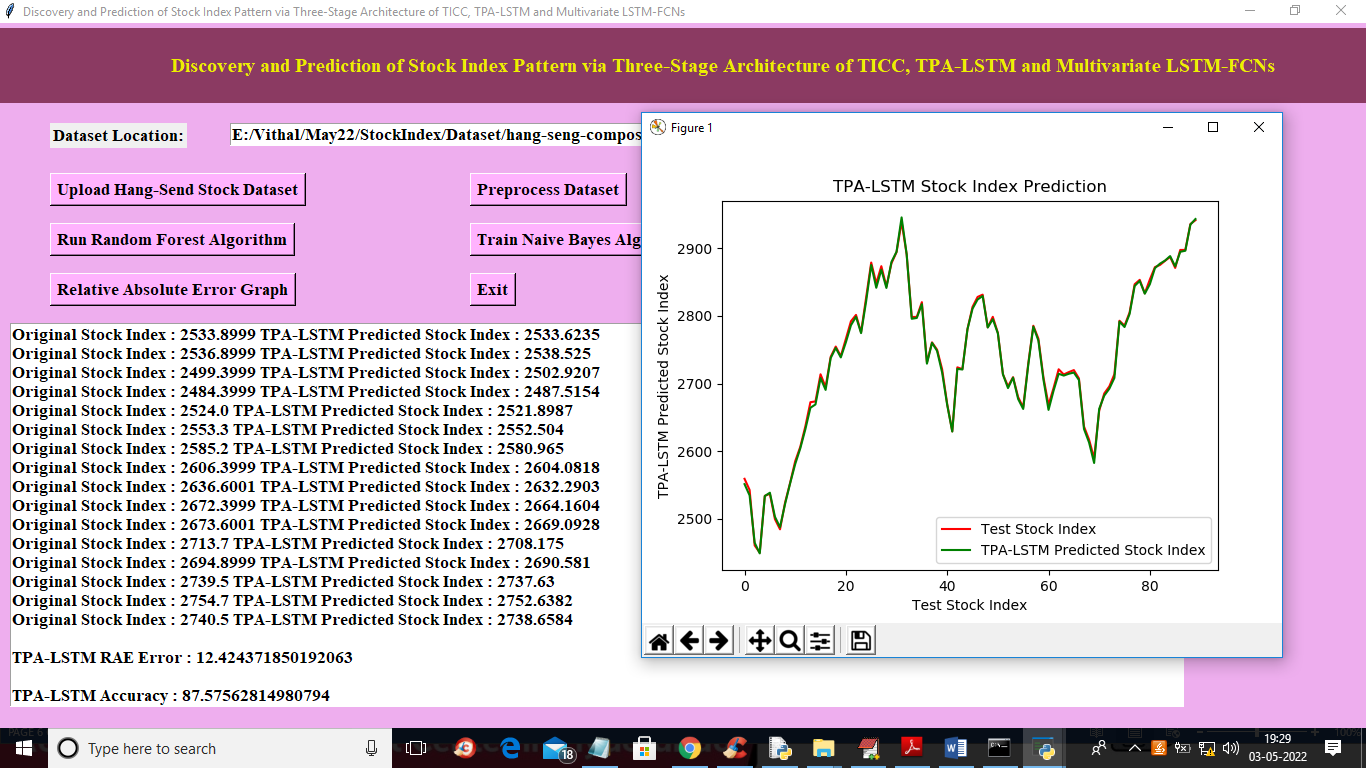
In above screen text area we can see original test value and predicted value from SVM and then calculate difference between original test value and predicted value as RAE and we got SVM RAE as 3551 and accuracy as 75% and in graph x-axis represents days and y-axis represents stock values and RED line represents original test value and green line represents predicted value and we can see there is huge gap between red and green line so prediction is not accurate and if prediction is accurate then both lines get overlap and now close above graph and then click on ‘Run Random Forest’ button to get below output



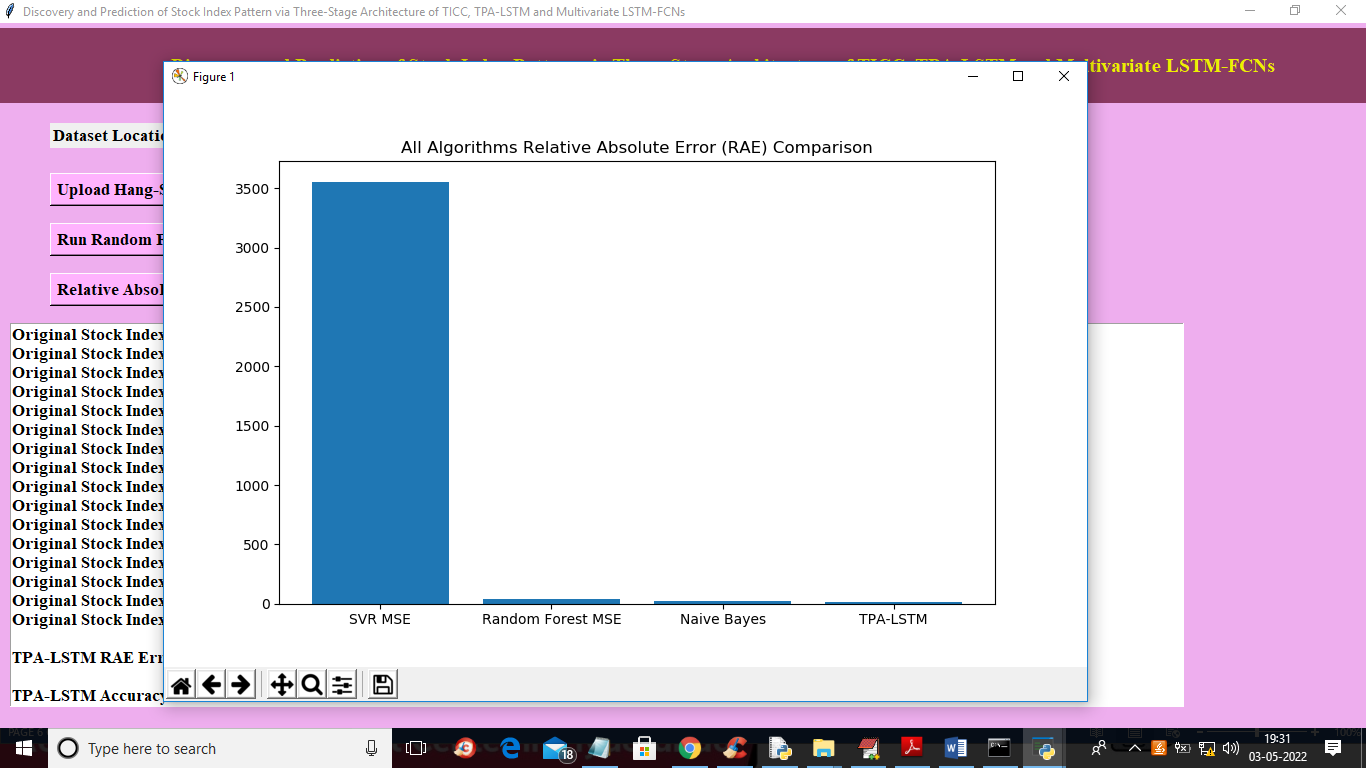
In above screen with random forest both lines are overlapping so its RAE error reduce to 40 and its prediction is little accurate and now close above graph and then click on ‘Run Naïve Bayes Algorithm’ button to get below output



In above screen with Naïve Bayes we got 25% error rate and both lines are overlapping so its prediction is also little accurate and now close above graph and then click on ‘Run Propose TPA-LSTM’ button to train propose algorithm and get below output



In above screen with propose LSTM-TPA we got RAE as 12% and accuracy as 87% and we can see both lines are overlapping without any gap so propose algorithm prediction is accurate. Now close above graph and then click on ‘Relative Absolute Error Graph’ button to get below graph



In above screen x-axis represents algorithm names and y-axis represents RAE error and in all algorithms propose LSTM-TPA got less error rate so its performance is good