Improving TCP Congestion Control with Machine Intelligence

Now-a-days world is connected with computer networks to pass information to any body and to manage this network activities easily we need to manage all resources such as Routing, congestion control, delay, packet deliver and many more very efficiently. Packets always get delayed if congestion window improperly managed and to avoid this problem Rule based congestion was introduced which check if one path is congested then choose alternate path but this rule based technique will take time to make decision of choosing alternate path. New Reno algorithm is based on Rule based TCP-Congestion management and to further enhance this congestion technique many other algorithms were introduced but their performance is not up to the mark.

In propose paper author is employing machine learning based supervised algorithm called Loss Prediction and Reinforcement Learning algorithms to handle TCP congestion. This algorithms will analyse network state and then learn how to handle congestion and if congestion is learned or predicted then it will choose alternate path. This algorithms will predict or learn congestion very quickly and due to this reason network performance can be increase. Learning quickly can avoid further congestion and delay will get reduced and throughput will get increased.

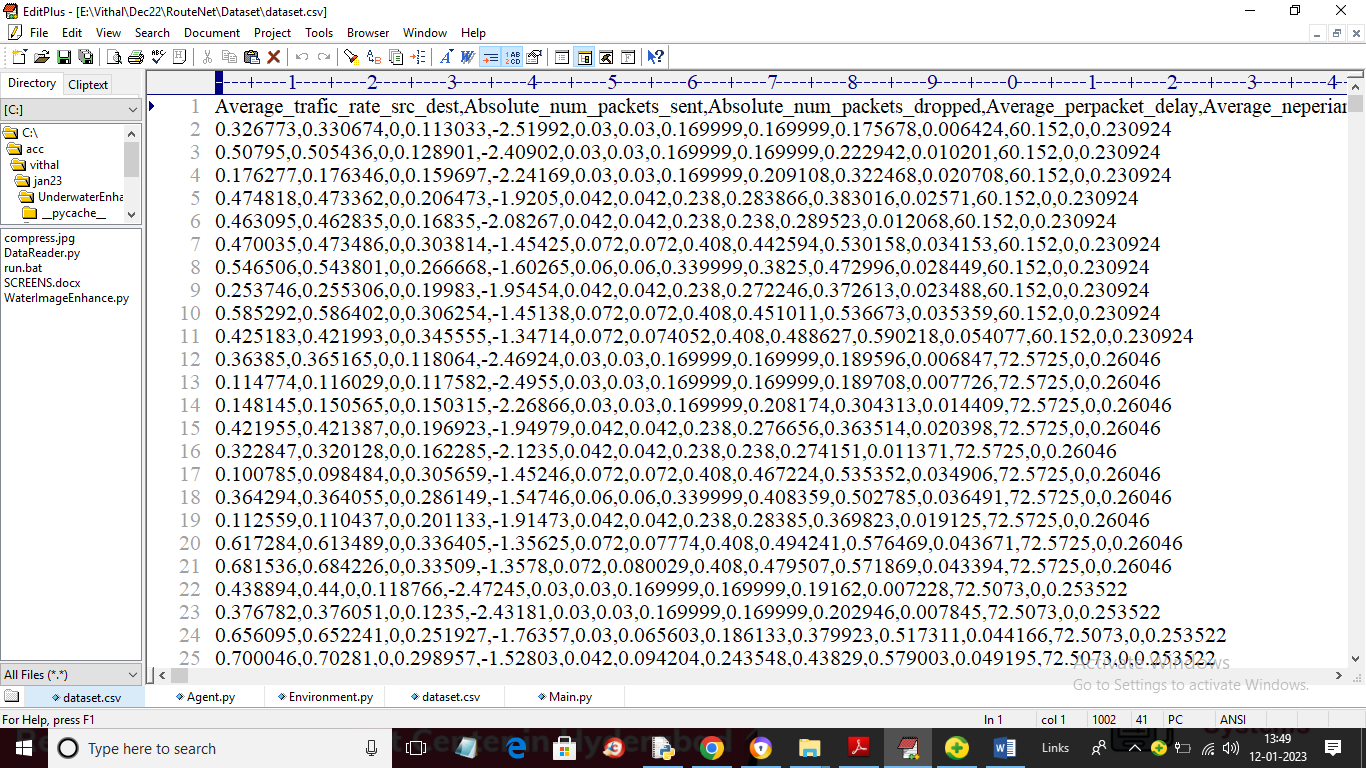
Reducing delay will help in enhancing Round trip time, quick acknowledgement and many more.

Supervised algorithm will take network current state as INPUT and then predict loss and congestion and based the predicted values path will be selected. Analysing network state may consume more time so author enhance this supervised algorithm with Reinforcement learning.

Reinforcement learning (RL-TCP) learn based on current environment state and its prediction time is less compare to other machine learning algorithms. RL algorithms consist of ENVIRONMENT, AGENT & ACTION. Current network state will be consider as ENVIRONMENT and based on Environment AGENT will take ACTION and if Action correctly predicted then RL will get rewarded and if not correctly predicted then RL will get penalised and this algorithm will trained or learn itself as long as the reward values get better and then final model can able to predict more accurate value.

To train both RL-TCP (TCP means transfer control protocol whose congestion will be improved based on RL) and LP-TCP we are using network state which contains SOURCE IP, Destination IP, delay details and other network values and by using this dataset both algorithms get trained.

In below screen we are showing dataset of network



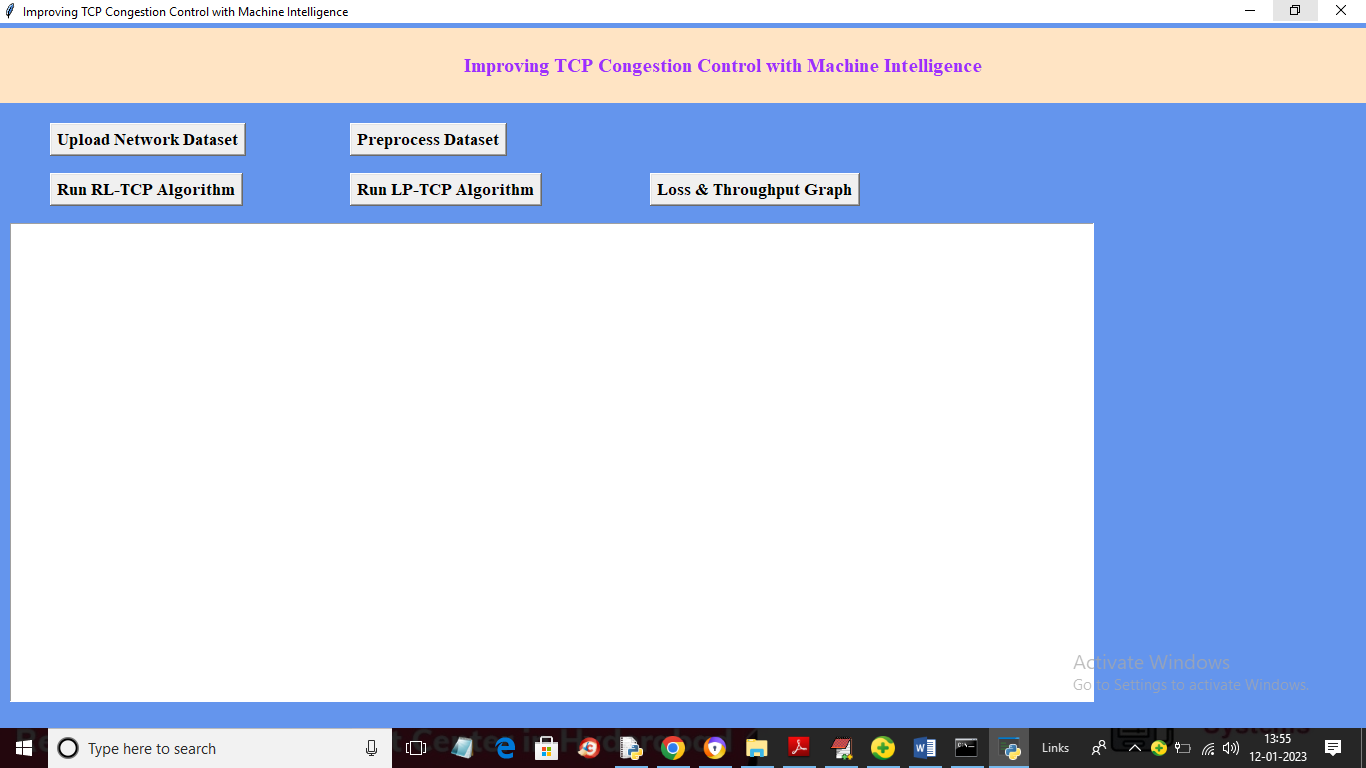
In above dataset screen first row contains dataset values and remaining rows contains dataset values and this dataset will be consider as network states.

To implement this project we are have designed following modules

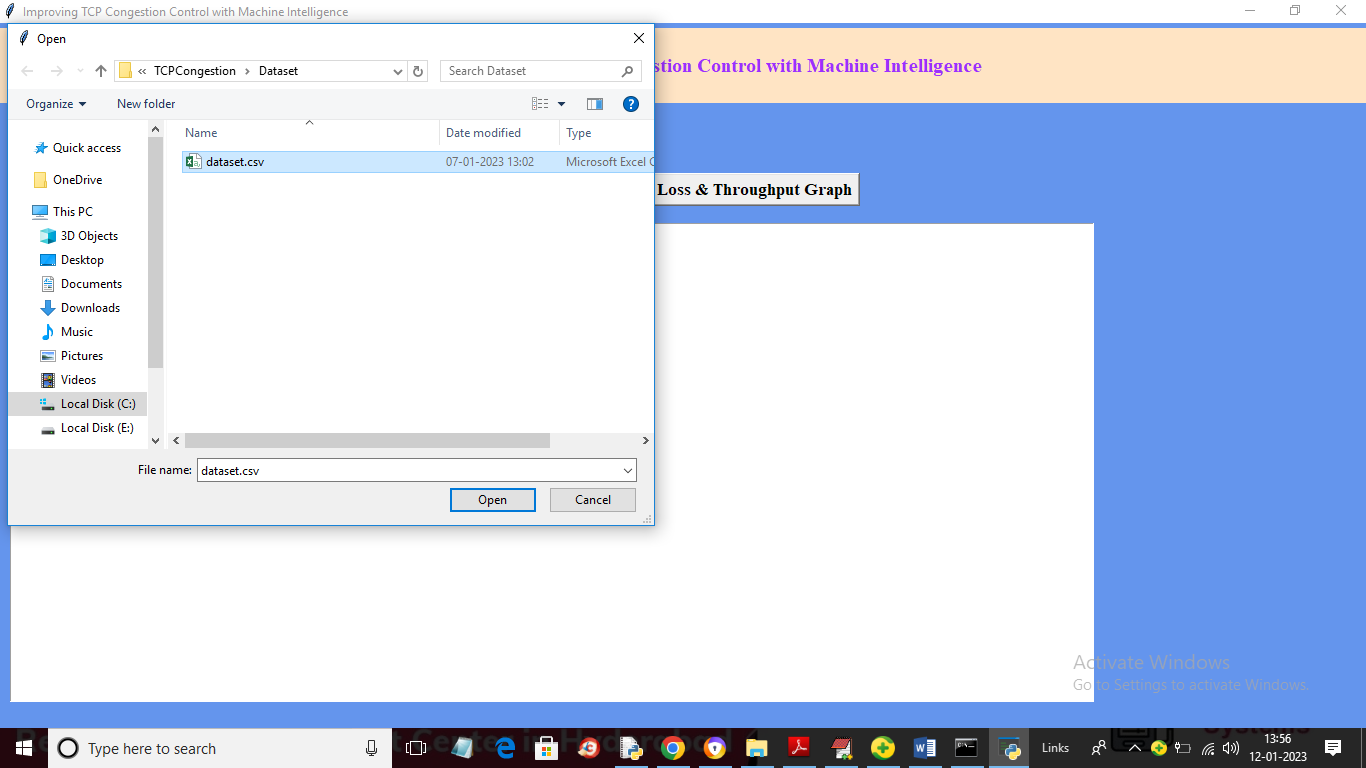
1. Upload Network Dataset: using this module we will upload network dataset to application
2. Preprocess Dataset: using this module we will read dataset and then remove missing values
3. Run LP-TCP Algorithm: now processed data will be input to LP-TCP supervise algorithm to predict congestion and based on congestion it will predict other path and based on alternate path we will calculate delay and throughput.
4. Run RL-TCP Algorithm: using this module we will trained RL-TCP algorithm to predict congestion and delay and based on prediction routing get handled and then will calculate delay and throughput. Here we will get delay for existing New-Reno and propose RL-TCP
5. Loss & Throughput Graph: using this module we will plot loss and throughput graph between both algorithms

SCREEN SHOTS

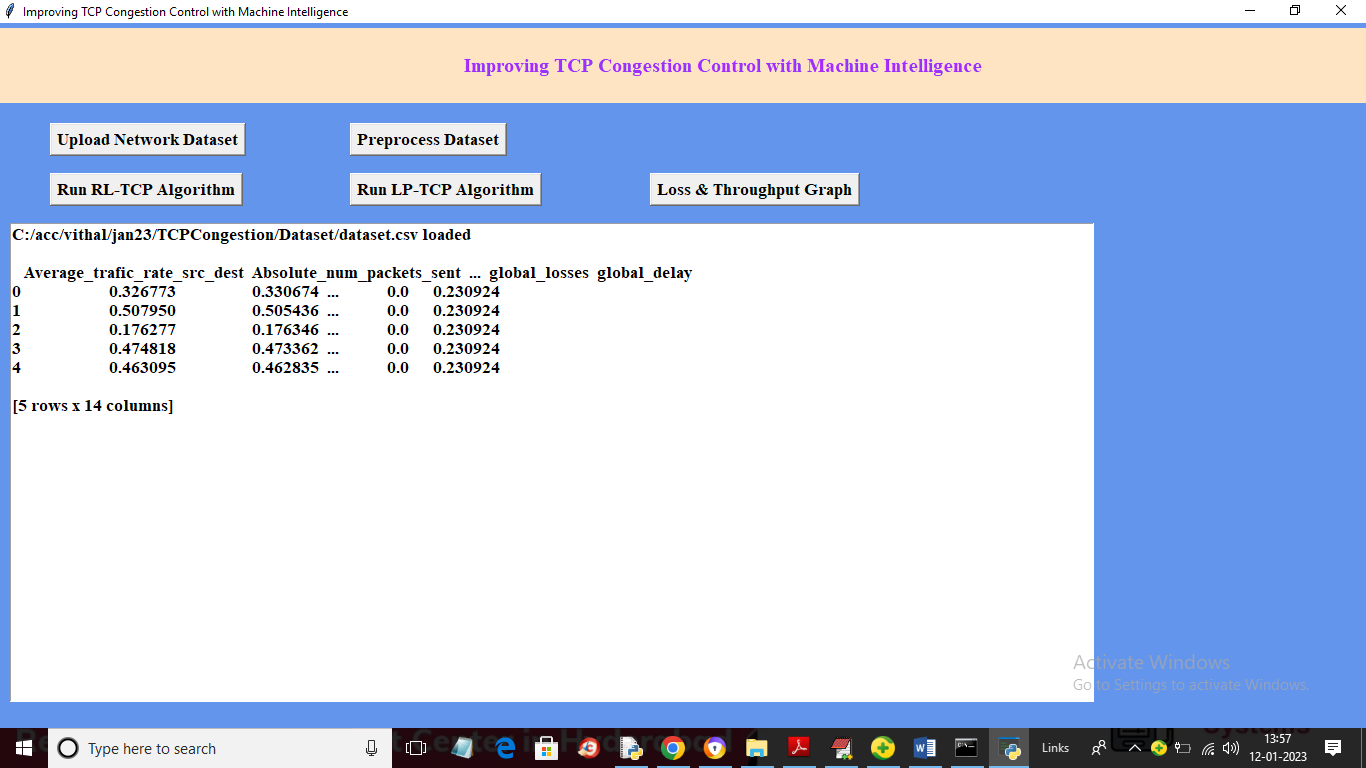
To run project double click on ‘run.bat’ file to get below screen



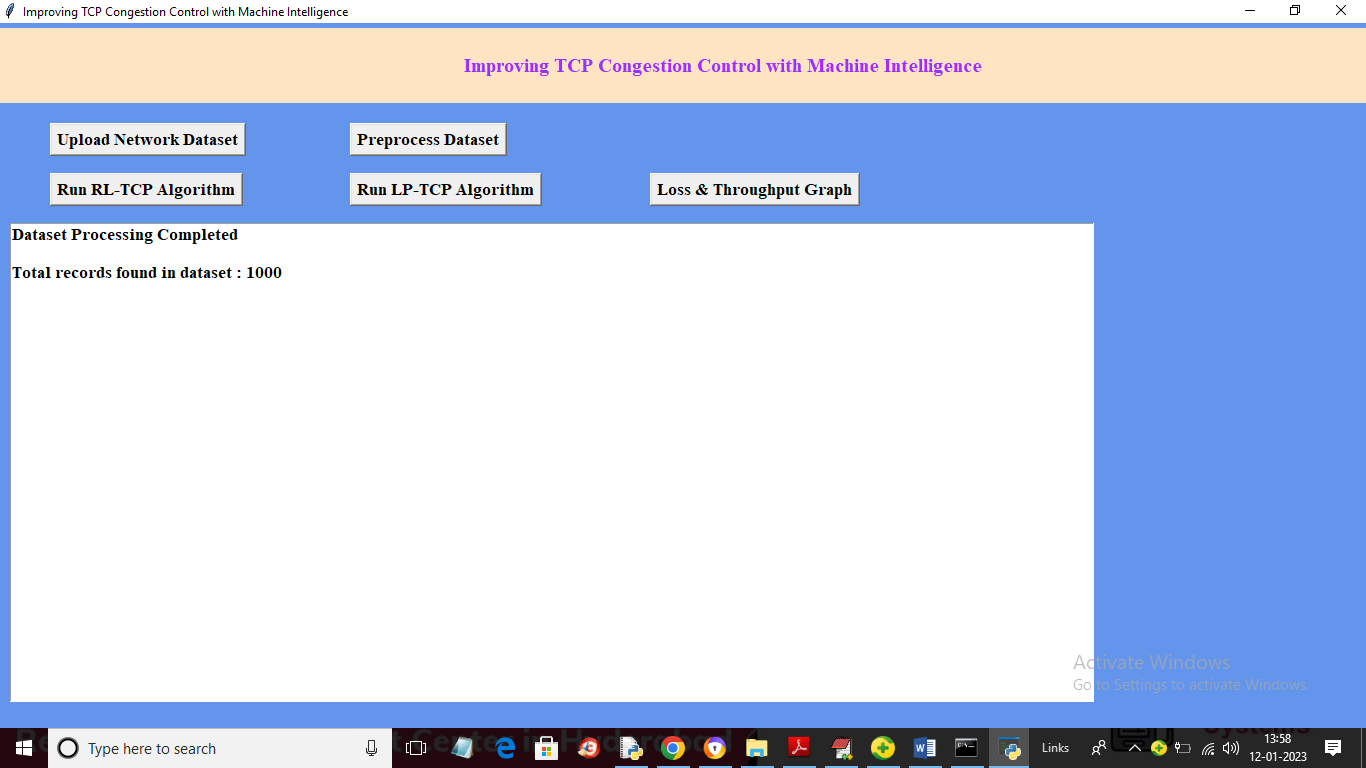
In above screen click on ‘Upload Network Dataset’ button to upload dataset and get below output



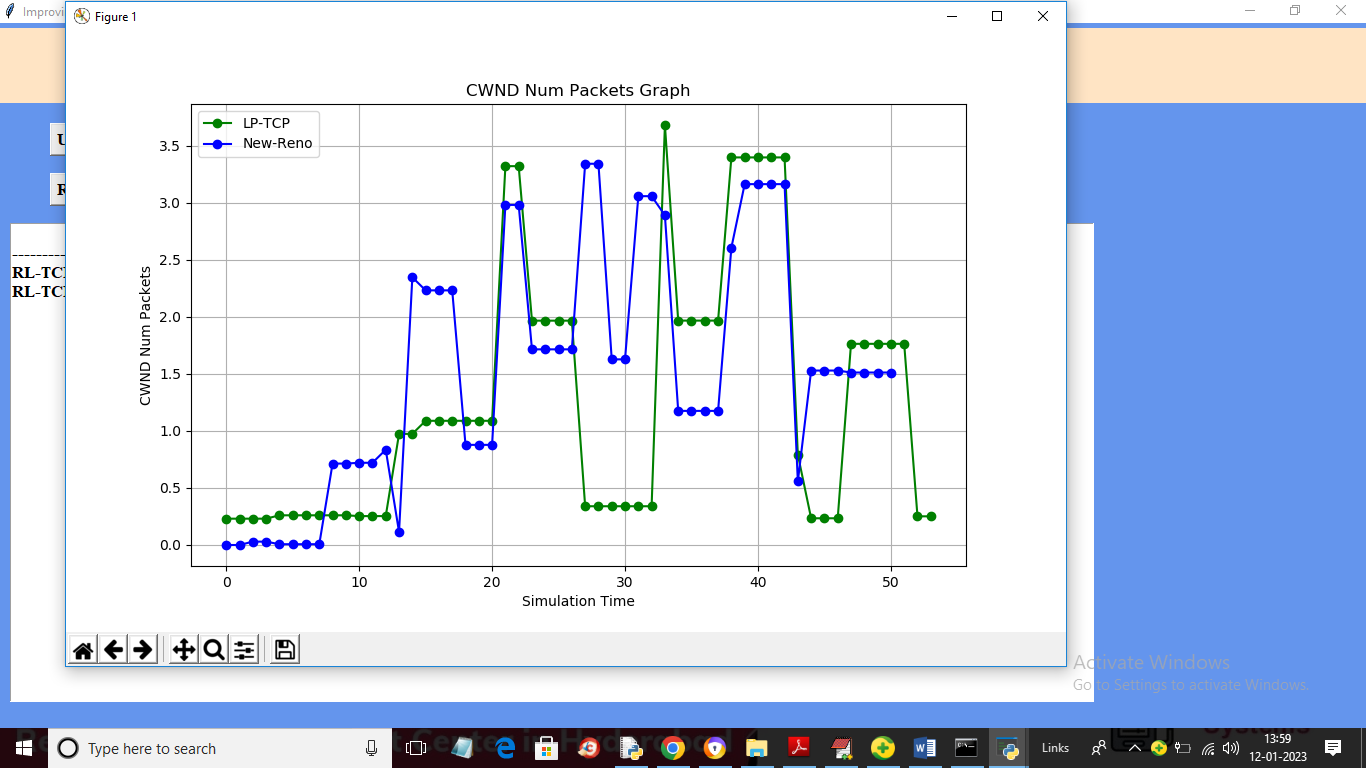
In above screen selecting and uploading dataset and then click on ‘Open’ button to load dataset and get below output



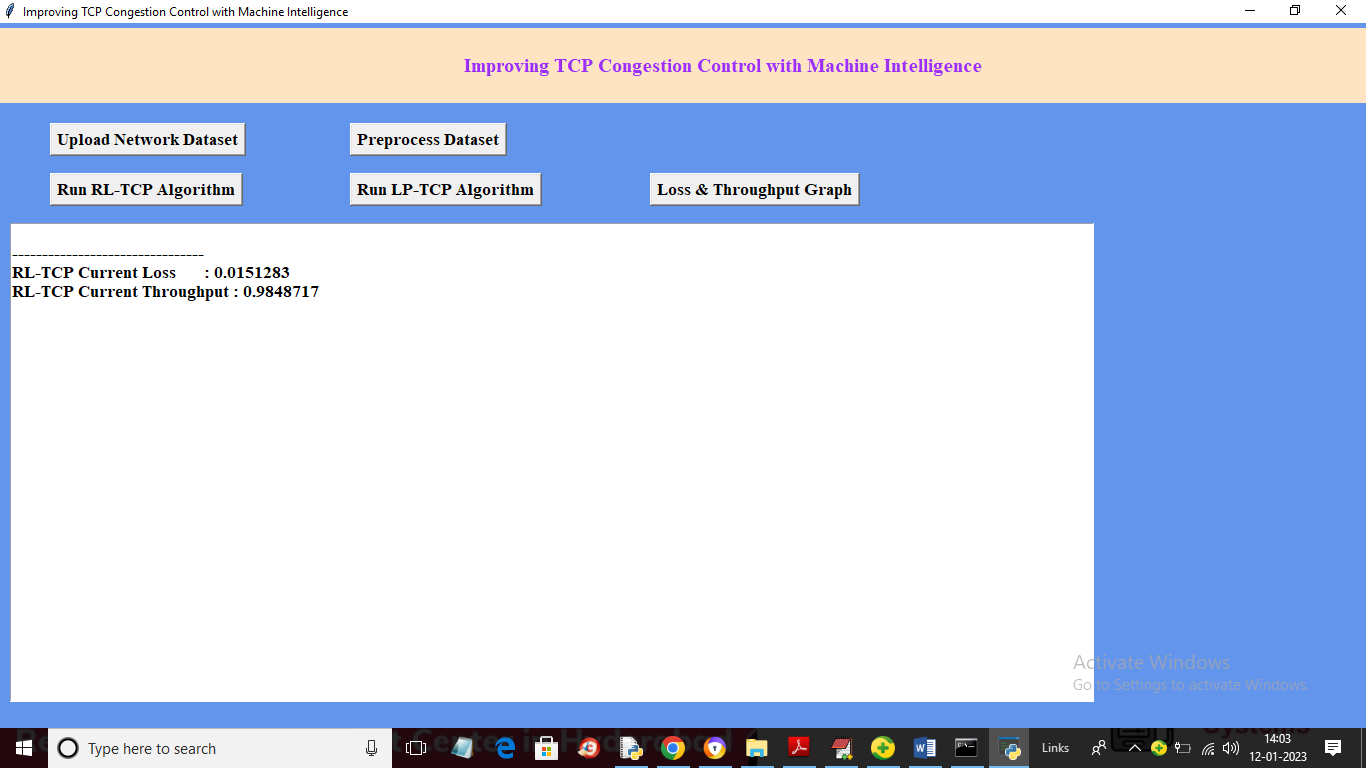
In above screen dataset loaded and now click on ‘Preprocess Dataset’ button to remove missing values and get below output



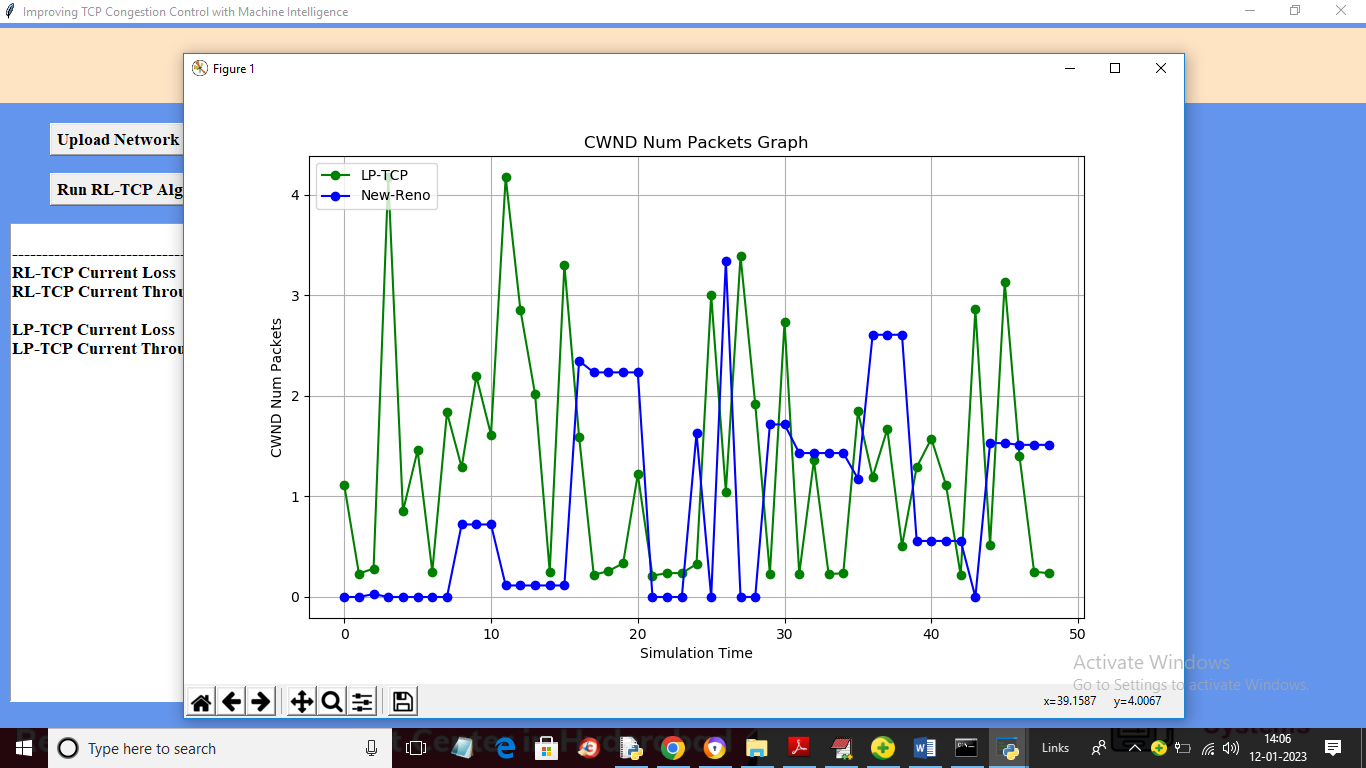
In above screen Preprocessing completed and dataset contains 1000 records and now click on ‘Run RL-TCP Algorithm’ button to train RL-TCP and get below output



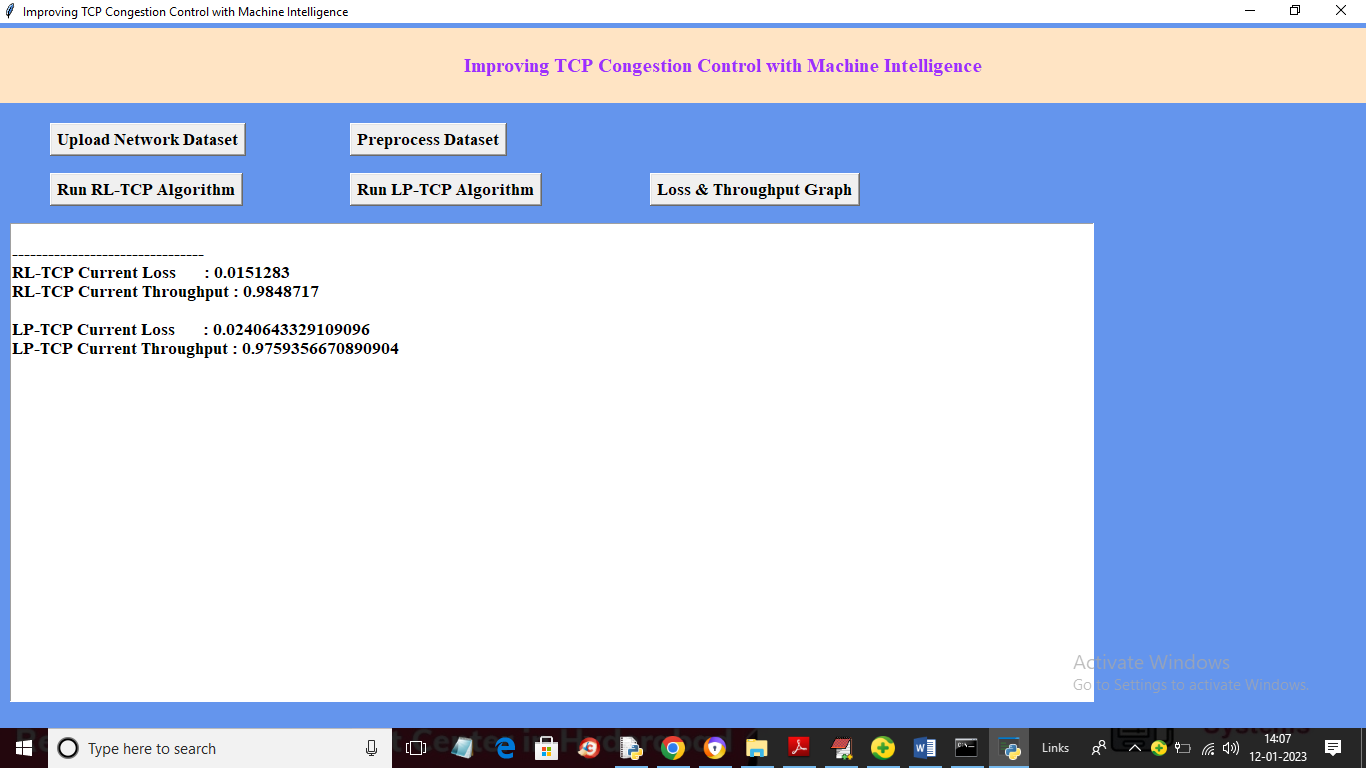
In above graph x-axis represents Simulation Time and y-axis represents CWND window size and for each packet sending we got learning or prediction rate for existing New-Reno (blue colour line) and propose RL-TCP (green colour line) and in above graph we can see RL-TCP got more packet prediction compare to existing New-Reno as RL-TCP prediction time is less so it can process more packets and its throughput will be high. Now close above graph to get below values



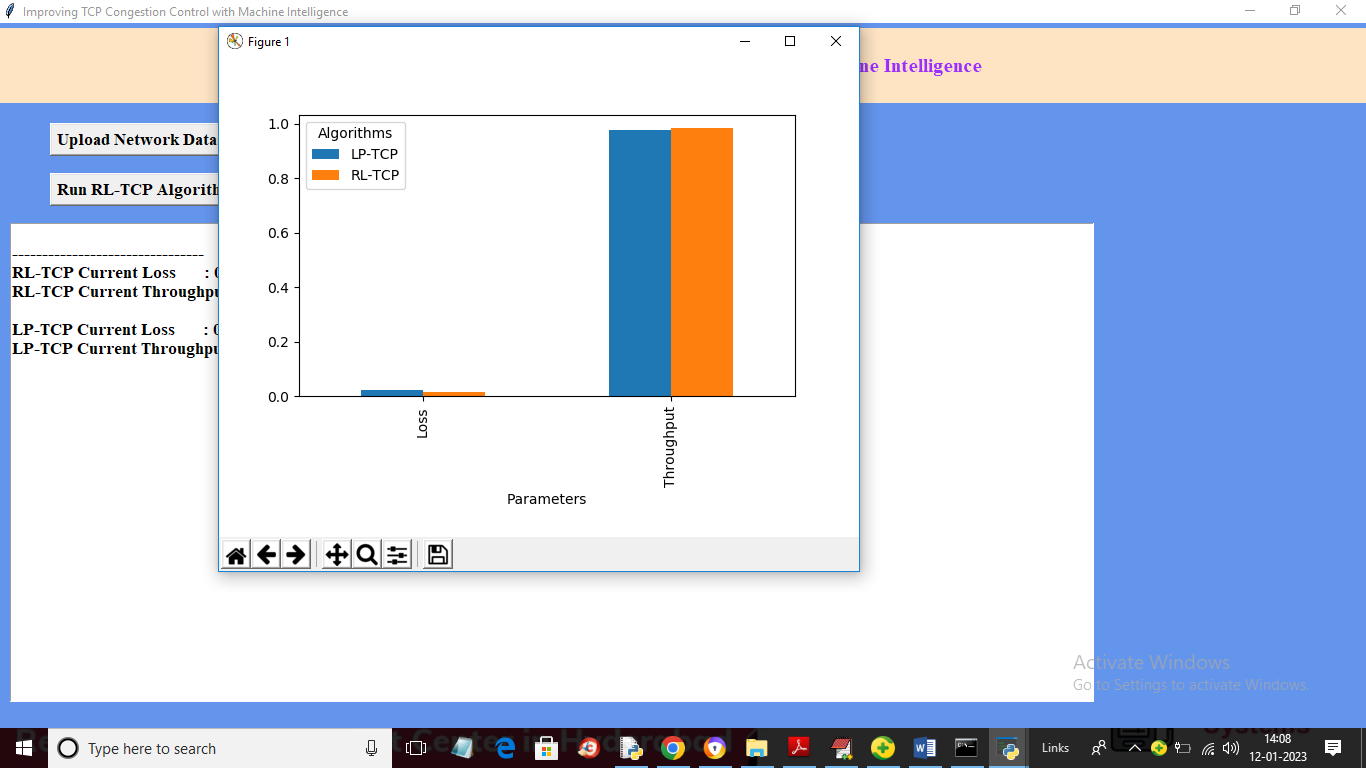
In above screen RL-TCP loss is 0.015 and its throughput is 0.98% as it is processing more packets due to less time in prediction so its throughput will be high and now click on ‘Run LP-TCP Algorithm’ button to get below output



In above screen we can see LP-TCP (green line) is also better than existing New-Reno to handle congestion and now close above graph to get below screen



In above screen with RL-TCP we got throughput as 0.98 and with LP-TCP we got 0.97 so RL-TCP is better than all other algorithms and now click on ‘Loss & Throughput Graph’ button to get below output



In above graph x-axis represents algorithm names and y-axis represents LOSS and throughout where orange bar is for RL-TCP and blue bar is for LP-TCP and in both algorithms RL-TCP got high throughput and less LOSS. So we can say with RL-TCP we can improve congestion to get less loss and high throughput