**CEGE0042 STDM – Modelling Traffic Flow in Salt Lake City by statistical and machine learning approaches.**

**Introduction**

Traffic flow prediction has been a persistent topic of interest in urban planning. Accurate modelling can help reducing congestion, plan for temporary regulations and promote the efficient of transportation infrastructures, etc. Two popular methods for predicting traffic flow data are spatial-temporal regression from the statistical domain and graph/recurrent neural networks from the deep learning family.

ARIMA is a statistical method widely used in modelling previous records in a time-series and make predictions over certain time spans in the future. STARIMA further incorporates spatial attributes of the observations by introducing a weight matrix measurement. GNN-LSTM is considered suitable. [LIT]

This report aims to evaluate the performance of these 2 models in dealing with spatial temporal data using the example of a traffic flow network from Salt Lake City.

**Experiment outlines**

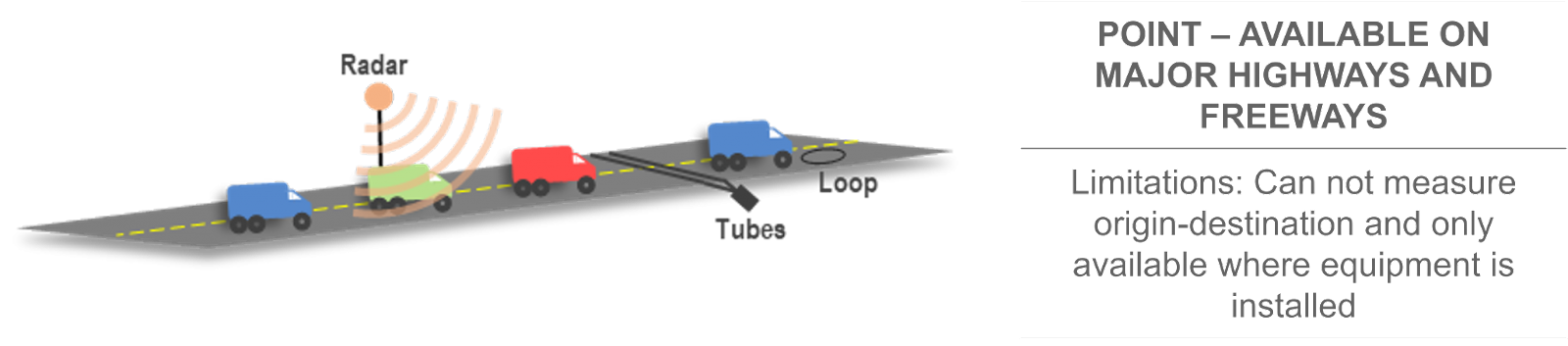
The data will be first be split into a training set and a test set by 70% and 30% respectively. The process of making predictions on traffic flow from an observation () at a certain time () and location () can be denoted as the formula: (Eqn.1). Given a set of input observations where is the observation at time and location :

and a time lag of , the function is to be defined to map input , to the output , which denotes the future measurement:

The task of this project is to model with functions , which can be learned using the proposed methods (STARIMA, GNN-LSTM) on a training dataset, by optimizing the evaluation methods accordingly to achieve a minimal difference between and a real measurement . The performance of the 2 models will be evaluated and compared.

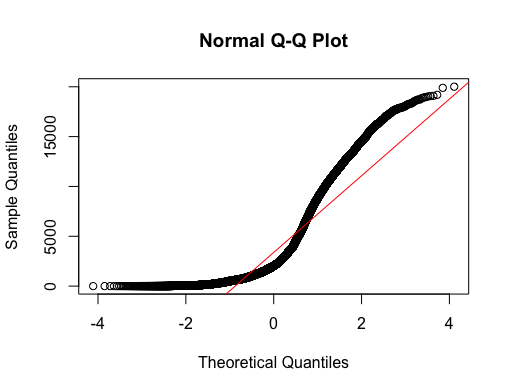
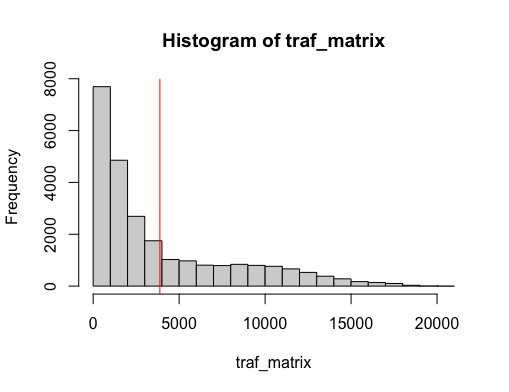
**Data Descriptions:**

The data modelled in this project is downloaded from Kaggle[ref] in 2 .csv files: ‘Utah\_Traffic.csv’ and ‘Utah\_Traffic\_Meta.csv’. The data was sourced from the Utah department of transportation and collected by sensors available on major highways and freeways[ref]. The file ‘Utah\_Traffic.csv’ contains records from 50 sensor at different locations, and the readings cover the whole of January, 2022 at 1 hour intervals (744 readings in total). The file ‘Utah\_Traffic\_Meta.csv’ provides the geometry of each station and the route number where it is located, which however does not suggest any ordinal information on how these locations are connected. Among the 50 columns of locations, 25 were found to have NULL values, which were removed from data for data quality purposes. And 1(station 636) was recognized as an identical location to station 673 in R due to precision loss in transferring the original coordinates into UTM, which was removed to avoid producing unconformable inputs in constructing the weight matrix.



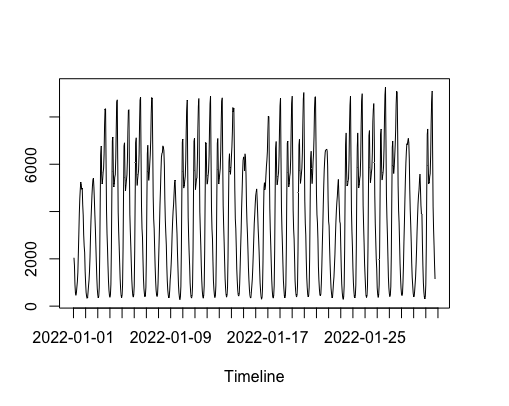
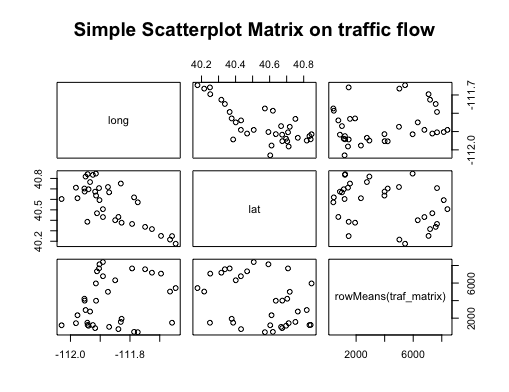
**Exploratory Data Analysis**

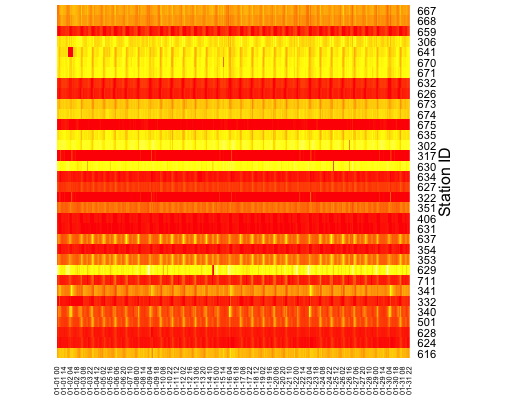
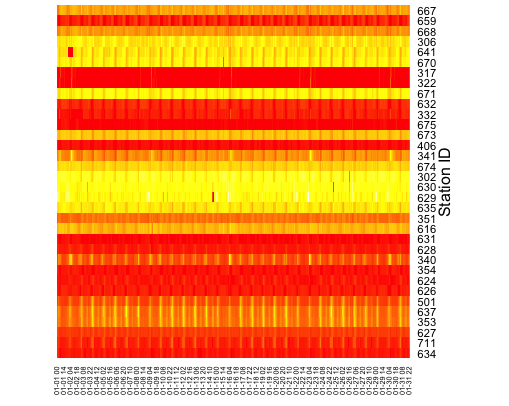
The overall mean () and standard deviation () of traffic amount was calculated by averaging all the readings from data regardless of location and timestamps, which have given results of and . The distribution is shown in fig [NUM] below, where a high positive skewness can be observed, suggesting a poisson-like distribution. This pattern can also be seen in fig[NUM], where more observations below the first two theoretical quantiles in a normal distribution are observed while less in the latter two quantiles. Spatial and temporal influences may be assumed in explaining this pattern. For examples, locations at busy areas may have more traffic flows than the others, while readings during night time may have lower traffic flow recorded.

  
Next, the data was explored according to its spatial and temporal aspects separately. As fig[NUM] shows, there may be a spatial pattern exist in traffic flow distribution, where less traffic amount is observed to the south-eastern part of the sampled area. This pattern can be also observed from fig[NUM].

Regarding the temporal dimension, fig[NUM] shows a cyclic pattern of fluctuation that corresponds to weeks, with two lower peaks that may indicate lighter traffic on weekends. On weekdays, two spikes on the same peak can be observed, representing the morning and evening rush hours where the evening rush hours appear to have more traffic flow than in the morning.

In addition, Fig[NUM] provides an illustration that shows observations in both spatial and temporal dimensions.





Fig[NUM]. heat map showing spatio-temporal distribution of traffic flow. [NUM]a. ordered by longitude on y-axis; [NUM] b. ordered by latitude on y-axis

**Methodology**

**1.Data preprocessing**

Data preprocessing is required to create equivalent time window and prediction periods, as well as weight matrix regarding the spatial dimension. The time window was set as 24 and prediction period as 1 for both models, which were later put into model as parameters.

The same weight matrix is used in both models and calculated by a distance-based spatial weights method provided by Anselin & Morrison(2019). In the processing, latitudes and longitudes for each station were first converted into UTM which takes into consideration the distance on spherical surface. The algorithm then finds the 1 nearest neighbor of a station by operating KNN with k=1. The maximum distance between paired stations is then set as the threshold for connectivity in this system. However, this threshold was added by 3000 in order to follow the assumption that all stations should be connected in the traffic network and to minimize the influence of isolated nodes in modelling. Euclidean distances between each pair of stations are then calculated. If the distance falls within9 the threshold, the two points are considered connected with the weights as the distance in between. The weight matrix was stored in R as neighbours list object and converted to matrix/arrays and saved as a csv file ‘WD\_Utah\_Traffic.csv’.

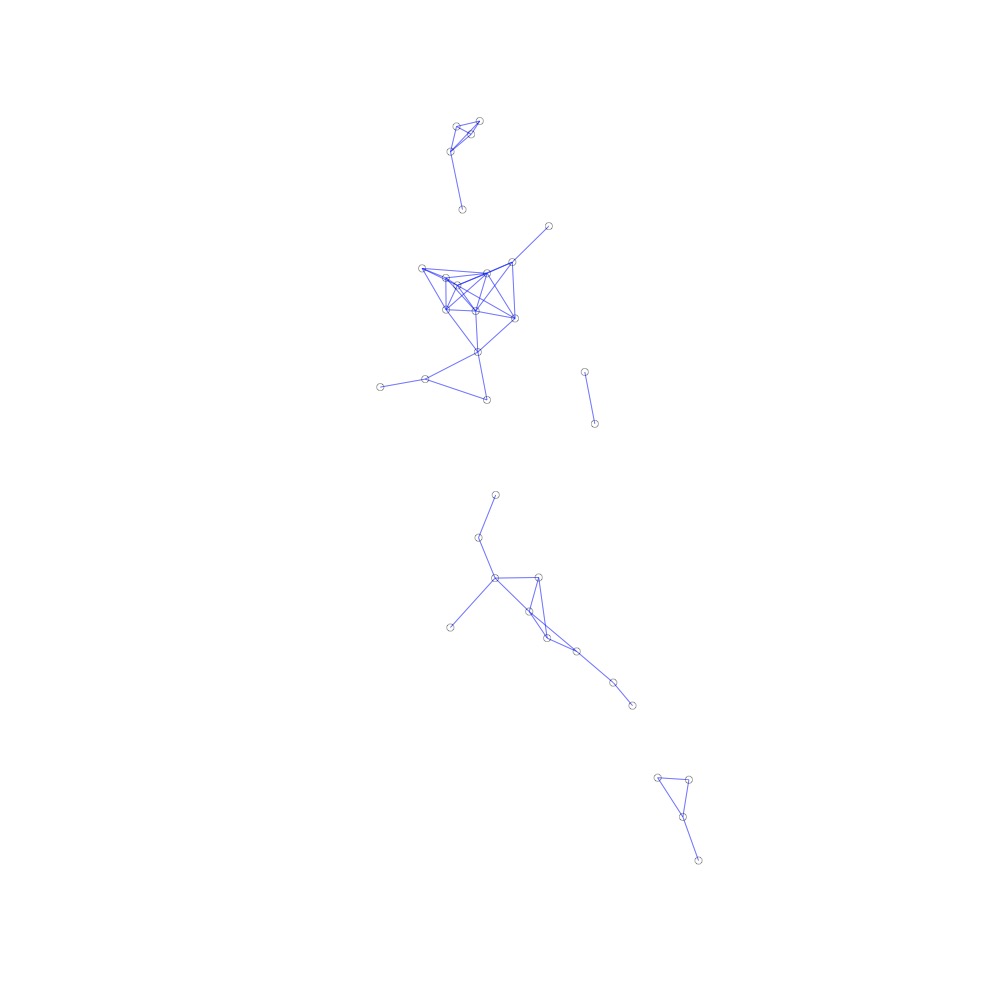
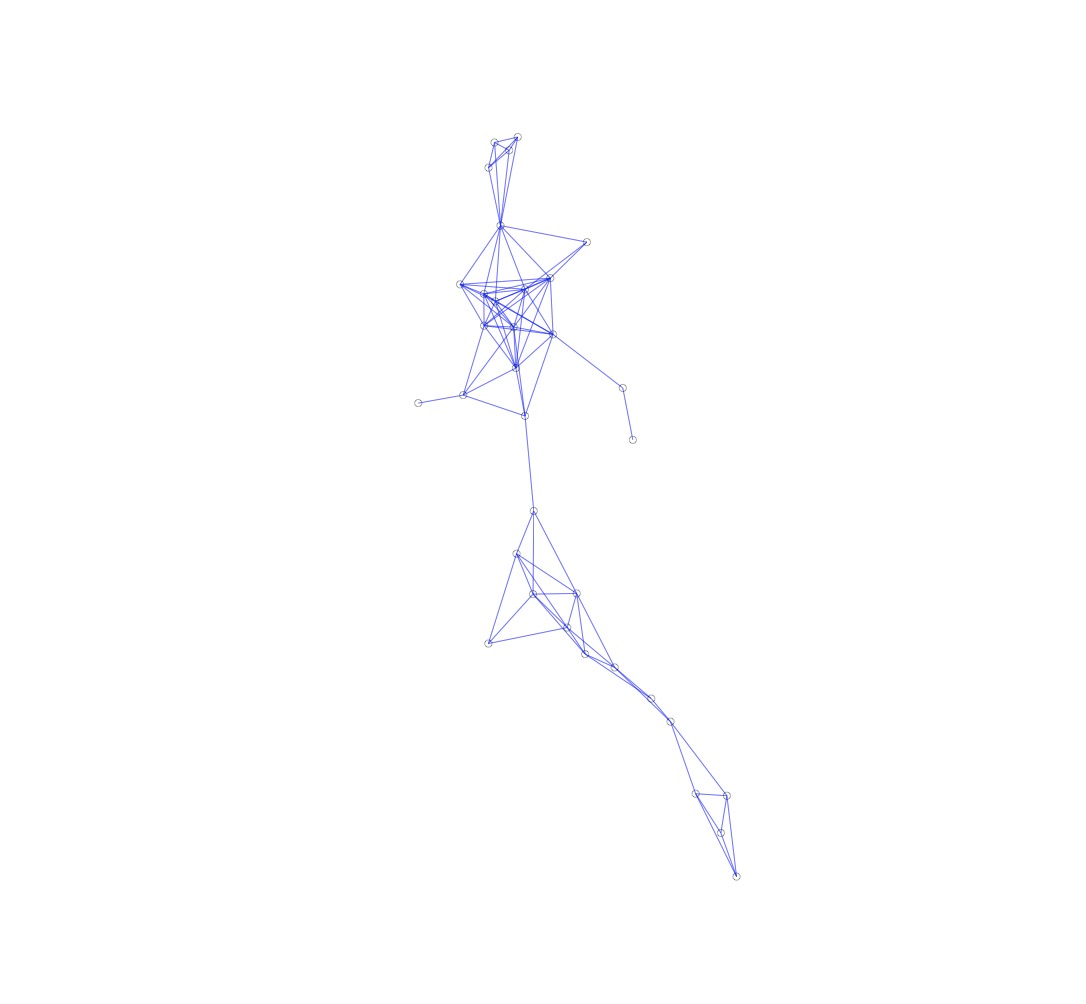
 

Fig. a. graph constructed with threshold as max distance between 2 neighbouring nodes, b. graph constructed with extra 3000 as threshold.

**2.STARIMA**

To perform the STARIMA, the package ‘STARIMA’ was sourced from Cheng[ref]. As shown in fig[NUM]a, the autocorrelation in time series have shown a cyclic pattern of 24 hours. The spatio-temporal PACF plot suggests a parameter p (time lag) of 3 and q (moving average) of range 0 to 4. The sets of parameters produces a good result with a relatively normal and stationary distribution (fig[NUM]), suggesting that most spatio-temporal information are well considered in STARIMA model.

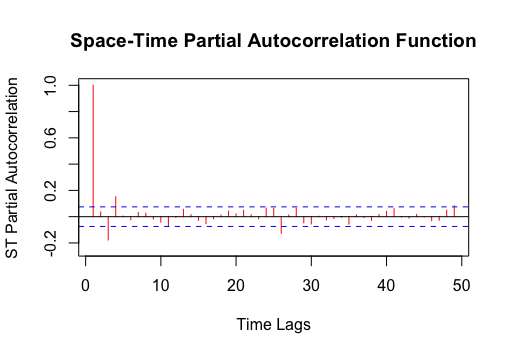
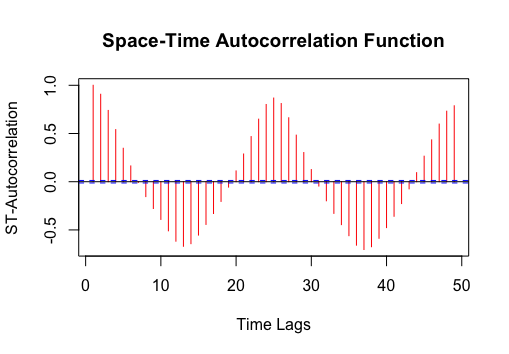


Fig a. ST-ACF covering 48 hours before differencing, b. ST-PACF covering 48 hours after 1st differencing.

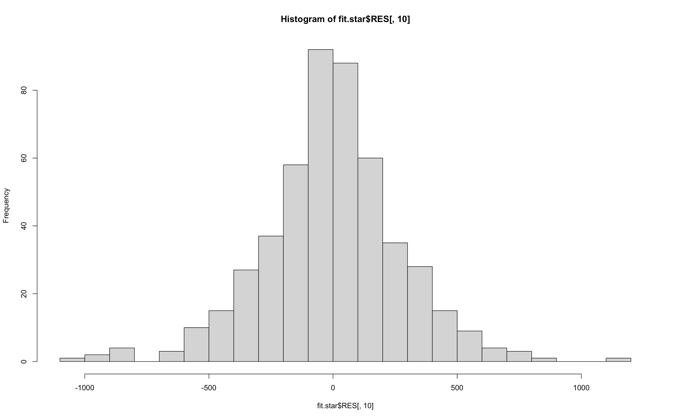
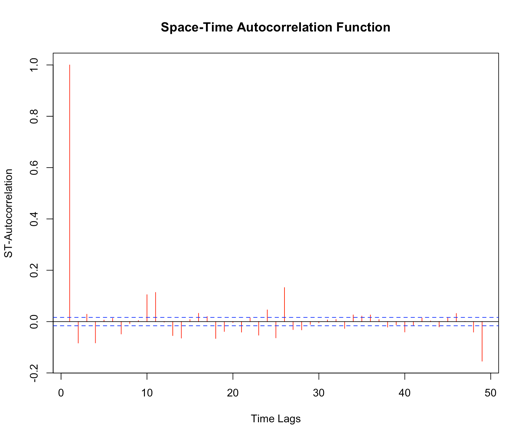
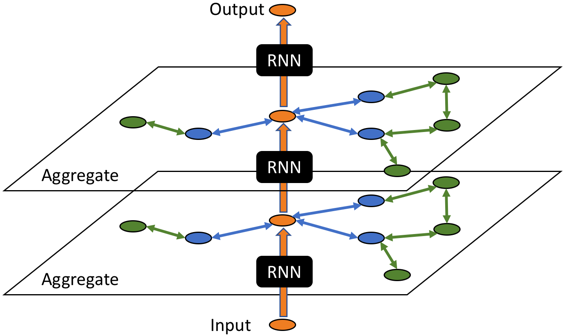
 

Fig.[NUM] a. histogram of fitted residual in first 48 hrs. b. STPACF of fitted residual in first 48 hrs.

**3.GNN LSTM**

GNN with LSTM embeddings was adapted from an open sourced example on Keras by Khodadadi(2021). The 2 files containing spatio-temporal data and weight matrix were imported and converted into matrices as the input of the model. A validation set was separated out by the last 2/7 in the training set. In the data preparation process, arrays in all datasets were normalized by the mean and std of the training set. The normalized arrays of time-series then went through time\_series\_dataset\_from\_array() and create\_tf\_dataset() in Keras to create tf.data.Dataset that helps to load samples (as tensors) in batches. The tensor input after processing should have 4 dimensions: batch size, input sequence, number of stations and an integer 1 as a placeholder for potential extra feature inputs.

The GNN model was then built from Graph convolution layers, where an initialized weight was given to each node. The weights at each node were updated repeatedly by aggregating weighted values from neighbouring nodes during the training process. Combined with the pre-defined weight matrix between stations, spatial correlation can be well considered. Each time after the graph updates, the nodes on was passed through a LSTM layer and outputs a tensor of the same shape (4,) as the nodes’ values. The tensors are then past through a dense layer to extract features and generate the final output. The structure of the GNN with LSTM embedded can be shown as the fig[NUM] below:



Fig[NUM] caption

After several rounds of tests, the parameters ‘input\_sequence\_length’, ‘forecast\_horizon’ are decided as 12 and 1 respectively, meaning that the measurement from previous 24 hours were used to predict the traffic count in the future 1 hour. Batch size was set as 16 to allow more data to be loaded in one training process. All batches were trained in 20 epochs.

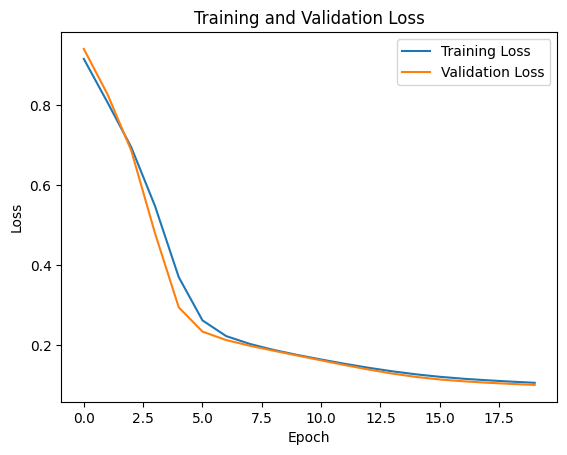


Fig. training and validation loss for 20 epochs

**Results and Discussions**

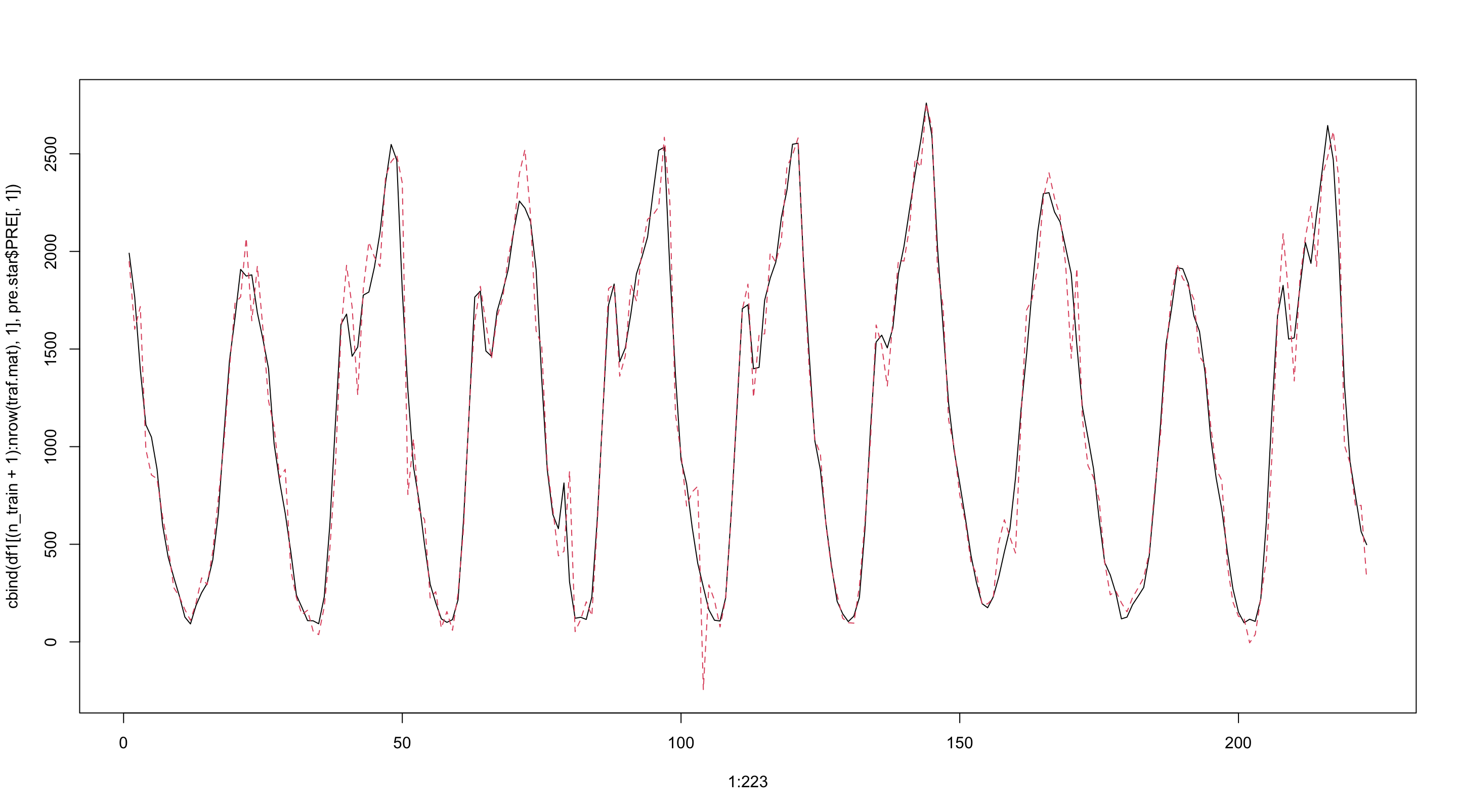
****

Fig.[NUM] prediction on station no.354 by STARIMA

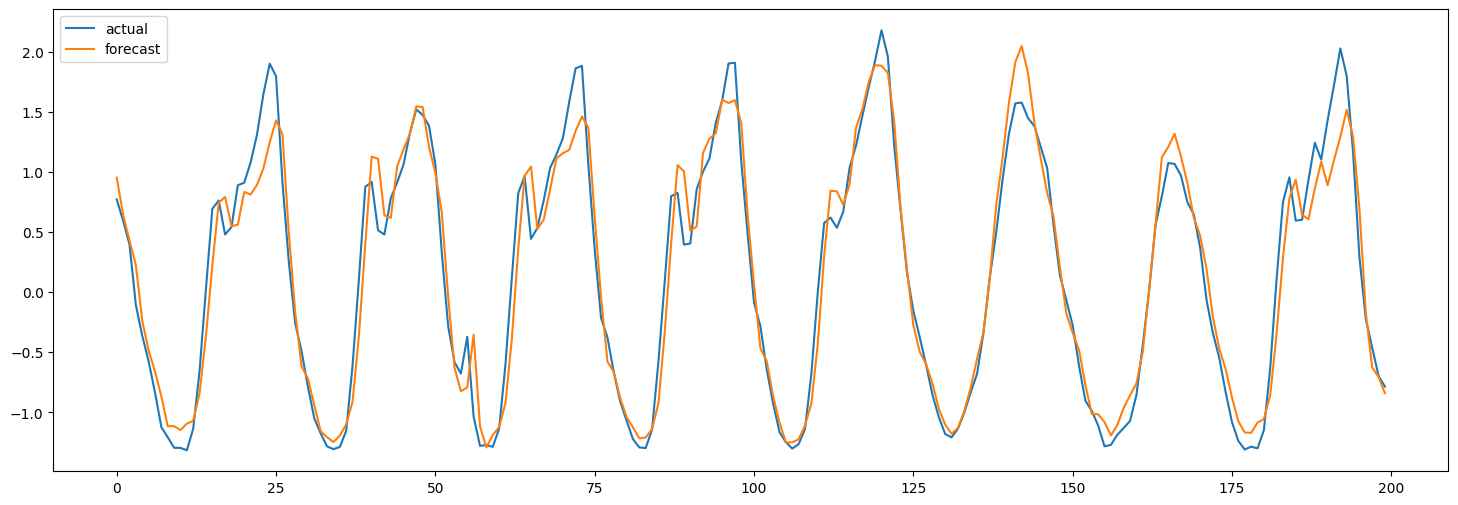


Fig.[NUM] prediction on station no.354 by GNN-LSTM

Fig[NUM] and fig[NUM] show the predictions made by the 2 different models over the time span of test dataset (223 timestamps in total) at an example station (no. 354). Both models have produced relatively accurate predictions on the measurements according to time. The STARIMA appears to have modelled the intensity of traffic count better while GNN-LSTM frequently. It is notable that STARIMA have better predictions on the intensity of traffic count on weekends (between ticks 150 and 200 in fig[NUM]) while GNN-LSTM seems to have exaggerated it. However, there are also several obvious divergences within the STARIMA predictions such as the prediction at around tick 100 in fig[NUM], and GNN-LSTM appears to predict better in the overall trend of data. Therefore, evaluation by statistical indexes may be required to compare the performances of the models.

Normalized rooted mean squared (NRMSE) error are calculated for both models to compare their performances. In predicting the next hour traffic flow at all stations, the STARMA produced an averaged NRMSE of 0.256 while GNN-LSTM have produced 0.066. It may suggest that GNN-LSTM is approximately 4 times accurate in making predictions in this task.

**Conclusion:**

There are limitations regarding data and methodology which could be improved. First, the construction of weight matrix in STARIMA and graph representation in GNN-LSTM takes a relatively crude approach. The attribute of the road numbers of the locations does not provide sufficient information to build

**References:**

<https://www.udot.utah.gov/connect/business/traffic-data/traffic-statistics/>

<https://www.kaggle.com/datasets/johnyoungsorensen/salt-lake-city-traffic>

<https://keras.io/examples/timeseries/timeseries_traffic_forecasting/>

<https://gis.utah.gov/data/transportation/roads-system/>

<https://spatialanalysis.github.io/lab_tutorials/Distance_Based_Spatial_Weights.html#distance-band-weights>

https://arxiv.org/abs/2103.07719

<https://www.researchgate.net/publication/358840548_Sentiment_Analysis_of_Weibo_Comments_Based_on_Graph_Neural_Network>

<https://www.arxiv-vanity.com/papers/1904.08035/>

Ma, Y., Zhang, H., Guo, J., & Yu, B. (2020). A comprehensive review on traffic flow prediction with big data. IEEE Transactions on Intelligent Transportation Systems, 21(1), 386-403.

Zhao, Y., Wang, X., & Yan, Y. (2021). Predicting traffic flow in urban road networks using deep spatio-temporal residual networks. Transportation Research Part C: Emerging Technologies, 124, 103173.

Goh, J., & Sayed, T. (2018). A comparison of regression and machine learning models for real-time urban traffic prediction. Transportation Research Part C: Emerging Technologies, 91, 296-314.