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.

Autoregressive integrated moving average (ARIMA) is a statistical method widely used modelling and forecasting time series data. ARIMA models use the past values of a time series to forecast future values. STARIMA further incorporates spatial dimensions of the observations by introducing a weight matrix measurement. Ding et al.(2010) Have suggested that STARMA outperforms ARIMA model at some longer forecasting intervals.

Graph neural network-long short-term memory (GNN-LSTM) is a spatial-temporal model that uses a graph neural network (GNN) to capture the spatial dependencies between observations and a long short-term memory (LSTM) network to model the temporal dependencies. One of the example structure of of GNN-LSTM proposed by Yu, Yin & Zhu.(2018) is STGCN which formulates traffic data on graph representations instead of convolutional units.

This report will follow these 2 approaches and aim to evaluate the performance of these 2 models in dealing with spatial temporal data using the example of constructing a traffic flow network for sampled traffic monitors in Salt Lake City.

Experiment outlines

The data (d) will be first be split into a training set d_{train} and a test set d_{test} by first 70% and latter 30% respectively in a time-sequence order. The process of making predictions on traffic flow from an observation ($x \in d_{train}$) at a certain time (t) and location (g) can be denoted as the formula: $y = f(x_{t,g})$. Given a set of input observations where $x_{t,g}$ is the observation at time t and location g:

$$x = \begin{bmatrix} x_{1,1}x_{1,2}\dots x_{1,g} \\ x_{2,1}x_{2,2}\dots x_{1,g} \\ \dots \\ x_{t,1}x_{t,2}\dots x_{t,g} \end{bmatrix}$$

and a time lag of $k \leq t$, the function f is to be defined to map input $x_{t-k,g} \in \{x\}$, to the output $y_{t+f,g} \in y$, which denotes the future measurement:

$$y = \begin{bmatrix} y_{t+1,1}y_{t+1,2}\dots y_{t+1,g} \\ y_{t+2,1}y_{t+2,2}\dots y_{t+2,g} \\ \dots \\ y_{t+f+,1}y_{t+f,2}\dots y_{t+f,g} \end{bmatrix}$$

The task of this project is to model x with functions f, 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 $y_{t+f,g}$ and a real measurement $y_{t,g} \in d_{test}$. The performance of the 2 models will be evaluated and compared on NRMSE (normalized Root Mean Squared Error).

Data Descriptions:

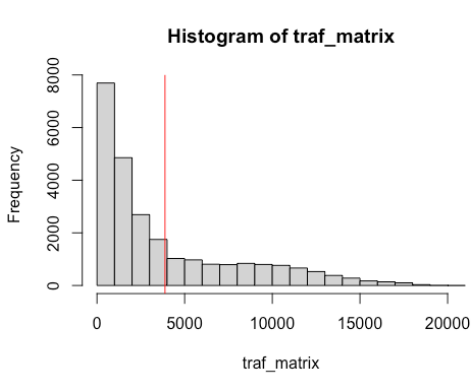
The data modelled in this project is downloaded from a Kaggle dataset created by Sorensen, J.Y.(2022) in 2 .csv files: 'Utah_Traffic.csv' and 'Utah_Traffic_Meta.csv'. The data was sourced from the Utah department of transportation and collected by sensors available on major highways and freeways(UDOT). 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.

Fig.1.monitor stations to calculate traffic flow (UDOT)

Exploratory Data Analysis

1.Descriptive Analysis

The overall mean (μ) and standard deviation (std) of traffic amount was calculated by averaging all the readings from data regardless of location and timestamps, which have given results of $\mu = 3879.591$ and $std = 4157.406$. The distribution is shown in fig.2.a below, where a high positive skewness can be observed, suggesting a poisson-like distribution. This pattern can also be seen in fig.2.b, 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.



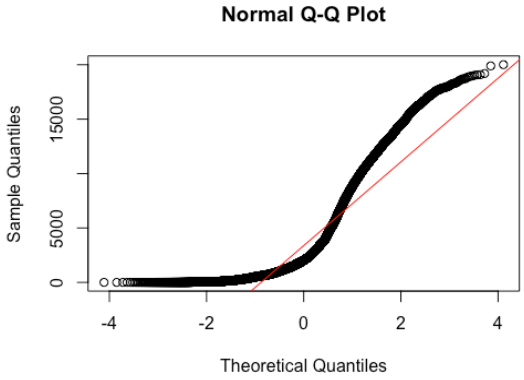


Fig.2.a. histogram of averaged traffic count over time and locations; b. QQ Plot of averaged traffic count over time and locations

Next, the data was explored according to its spatial and temporal dimensions. As fig.3.a. 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.4.

Regarding the temporal dimension, fig.3.b. 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 probably the morning and evening rush hours where the evening rush hours appear to have more traffic counts than in the morning. In addition, Fig.4. provides an illustration that shows observations in both spatial and temporal dimensions.

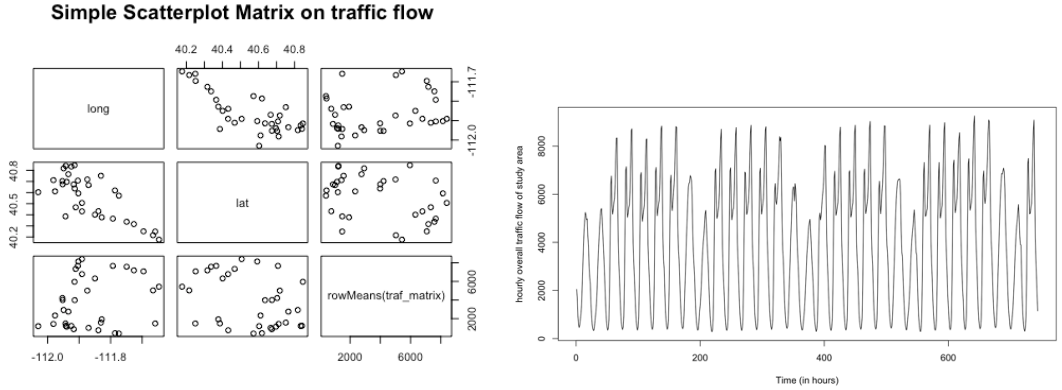


Fig.3. a.scatterplot matrix on traffic flow; b. time series of averaged traffic flow of all locations

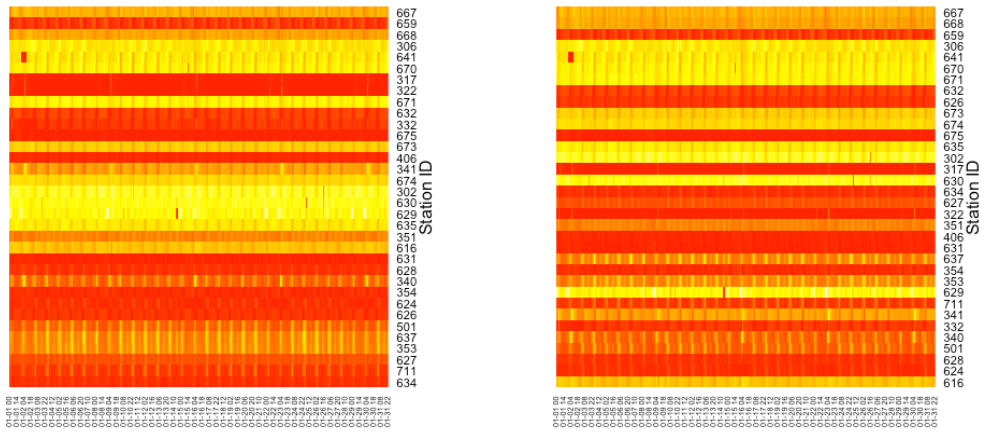


Fig.4. heat map showing spatio-temporal distribution of traffic flow.

a. ordered by longitude on y-axis; b. ordered by latitude on y-axis

2.Statistical tests on Autocorrelations

The patterns derived from the illustrations may need to be further verified by statistical tests on spatial and temporal autocorrelation. On the temporal dimension, autocorrelation was measured on the time-series data averaged over all stations. As shown on fig.5, a cyclic pattern of 24 hours can be observed. To eliminate the

influence from this seasonal change, first differencing at lag 24 was applied over the time series and the results shown in fig.6.

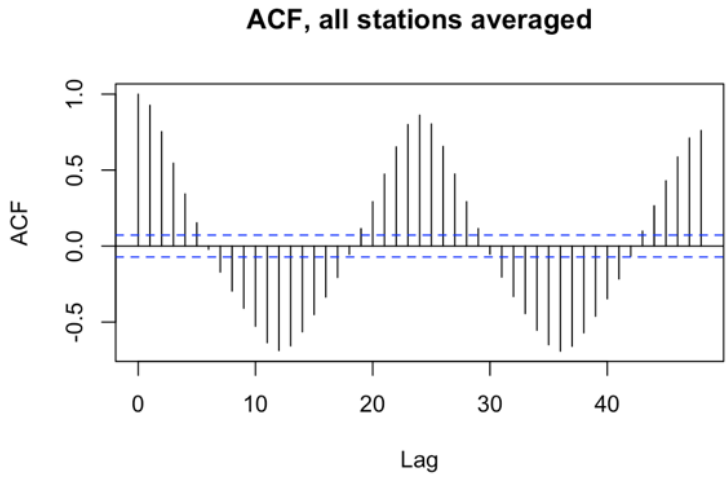


Fig.5. Autocorrelation functions (ACF) on time-series of averaged traffic flow of all locations

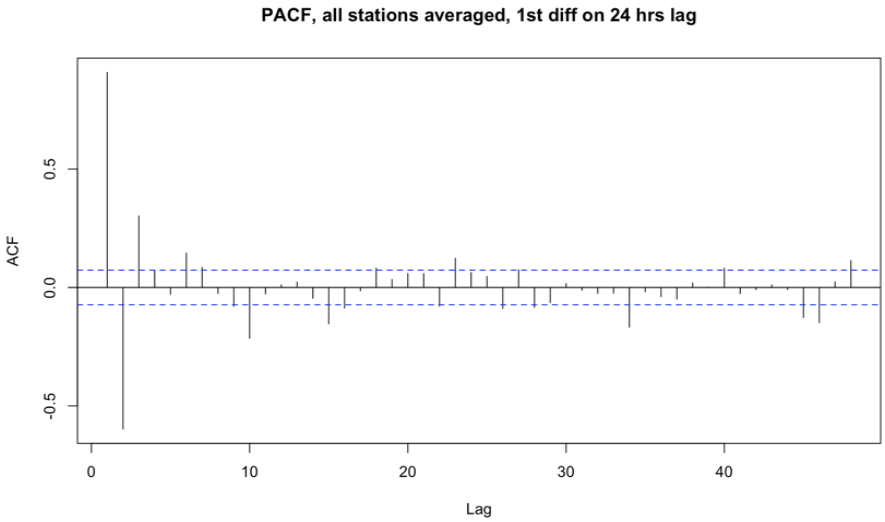


Fig.6. Partial Autocorrelation Functions (PACF) on time-series of averaged traffic flow of all locations

For the spatial dimensions, Moran's I test and Monte Carlo simulation are performed on hourly averaged traffic counts of stations. Both tests have suggested a spatial autocorrelation between sampled locations at a confidence level of over 90%.

```
        Moran I test under randomisation

data:  traf_avg
weights: W

Moran I statistic standard deviate = 1.3378, p-value = 0.09049
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
       0.11928762       -0.03030303        0.01250396
```

```
        Monte-Carlo simulation of Moran I

data:  traf_avg
weights: W
number of simulations + 1: 10000

statistic = 0.11929, observed rank = 9068, p-value = 0.0932
alternative hypothesis: greater
```

Fig.7 a. Moran I test on averaged hourly traffic count readings at monitor stations; b. Monte Calo simulation on averaged hourly traffic count readings at monitor stations

The level of spatial autocorrelation can be plotted on the semi-variogram. Fig.9. shows spatial autocorrelations in 4 directions, amongst which a stronger autocorrelation is observable at the angle of 45°. This reflects the shape of the sampled area shown in fig.8., where around 1/3 of the stations stretches to the south-eastern area to the city along US Highway I15.

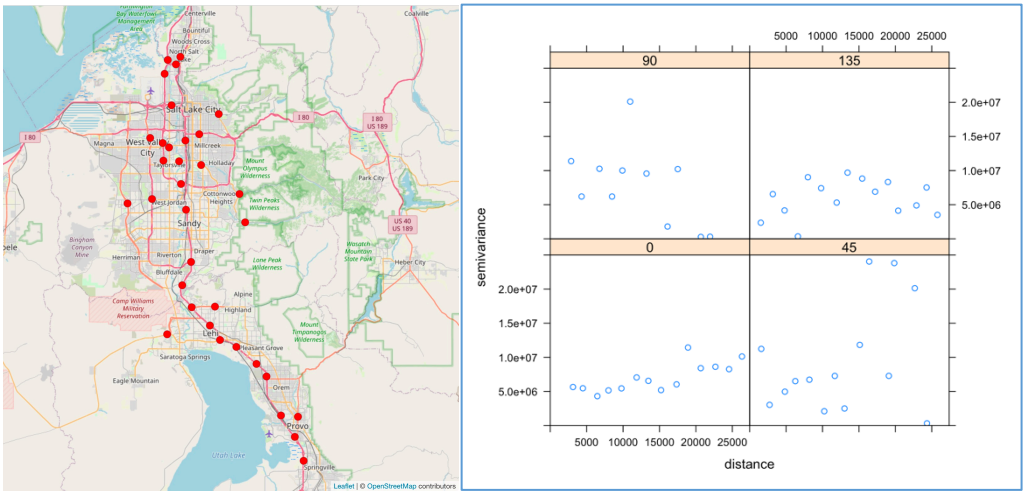


Fig.8. Locations of monitor stations, Fig.9.semi-variogram of traffic count readings at monitor stations on 4 directions

Methodology

1.Data preprocessing

Data preprocessing is required to create a weight matrix regarding the spatial dimension. The same weight matrix is used in both models and calculated by a distance-based spatial weights method provided by Anselin & Morrison(2019). During 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’.

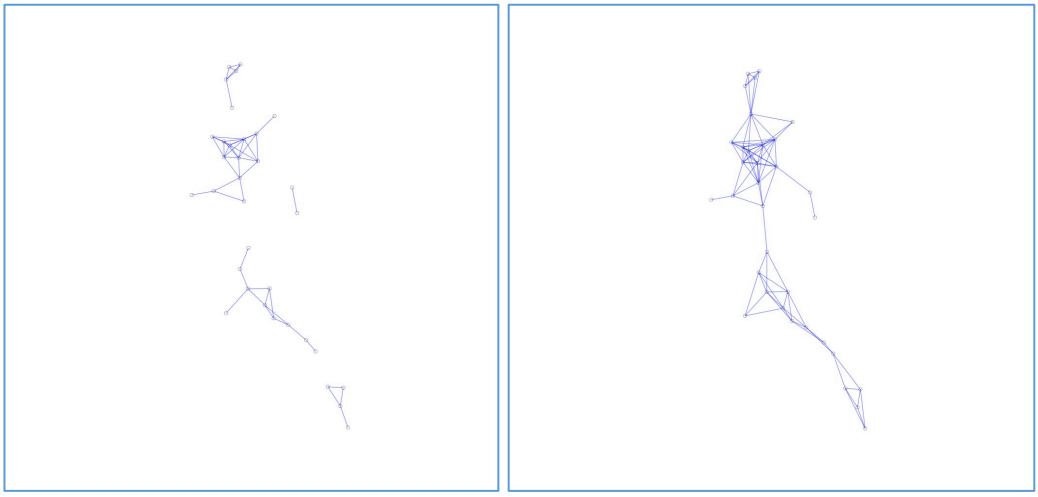


Fig.10. a. graph constructed with threshold as max distance between 2 neighbouring nodes, b. graph constructed with extra 3000 as threshold.

2.STARIMA

STARIMA was performed in R with the package 'STARIMA'. As shown in fig.10.a, the autocorrelation in time series have shown a cyclic pattern of 24 hours (which also reflects back to fig.10.b). The spatio-temporal PACF plot suggests a parameter p (autocorrelation lag) of 3 and q (moving average) of range 0 to 4. The sets of parameters $(p = 3, d = 24,\ q = 1)$ produces a good result with a relatively normal and stationary distribution (fig.11.), suggesting that most both spatial and temporal dimensions are well considered in STARIMA model.

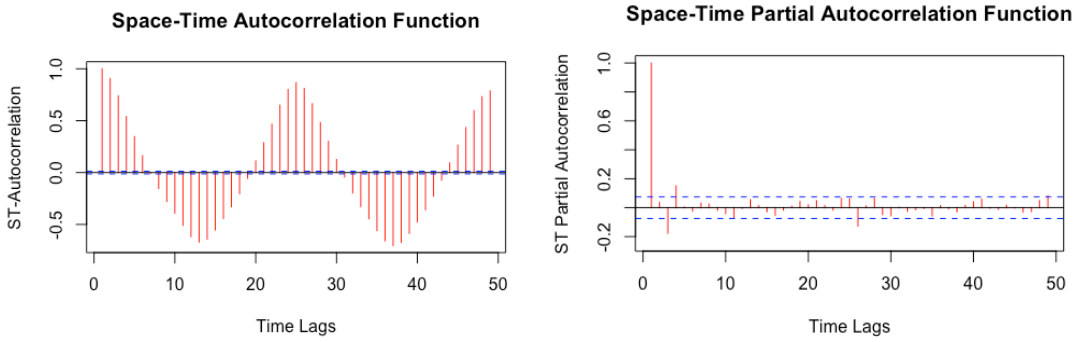


Fig a. ST-ACF covering 48 hours before differencing, b. ST-PACF covering 48 hours after 1st differencing.

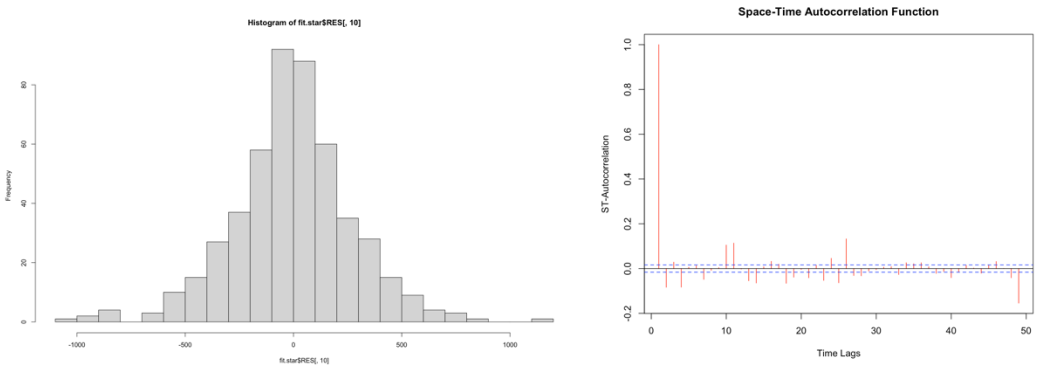


Fig.11 a. histogram of fitted residual in first 48 hrs; b. STPACF of fitted residual in first 48 hrs.

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 one Graph convolution layer, 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 could be well incorporated. 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. A brief illustration on the structure of GNN with LSTM embedded can be shown as the fig.12. below:

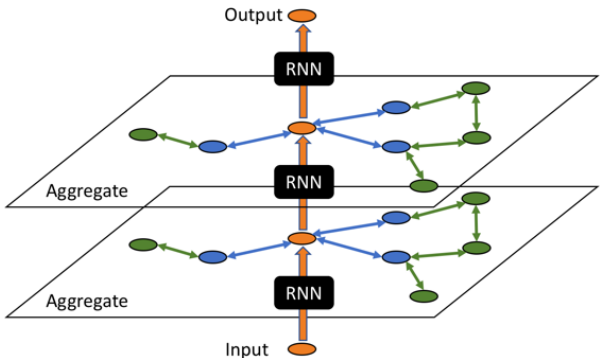


Fig.12. A brief illustration of GNN-LSTM structure (Huang & Carley, 2019).

To be in accordance with STARIMA, the parameters 'input_sequence_length' and 'forecast_horizon' are decided as 3 (equal to p in STARIMA) and 1 (equal to the prediction interval) respectively, meaning that the measurement from previous 3 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 first

trained in 50 epochs and the model started to overfit after 22 epochs (fig.13.). Model was then trained in 22 epochs to produce the final result.

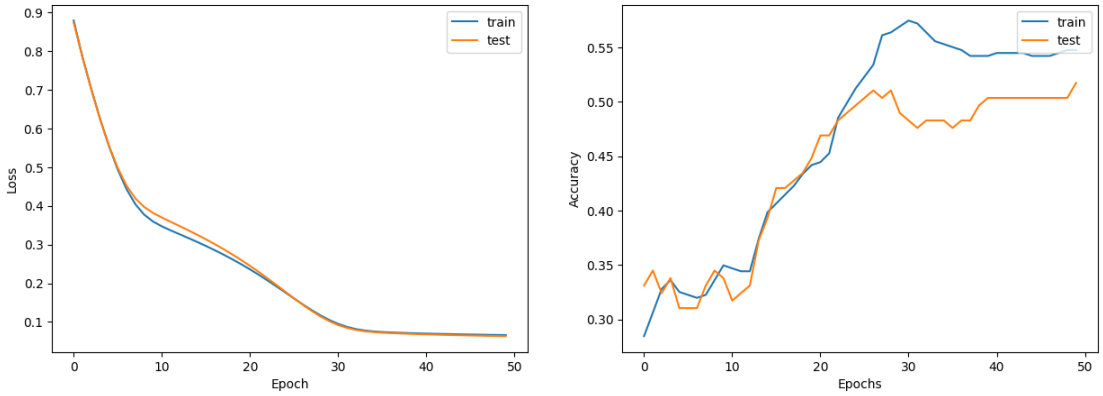


Fig.13. training and validation loss for 50 epochs

Results and Discussions

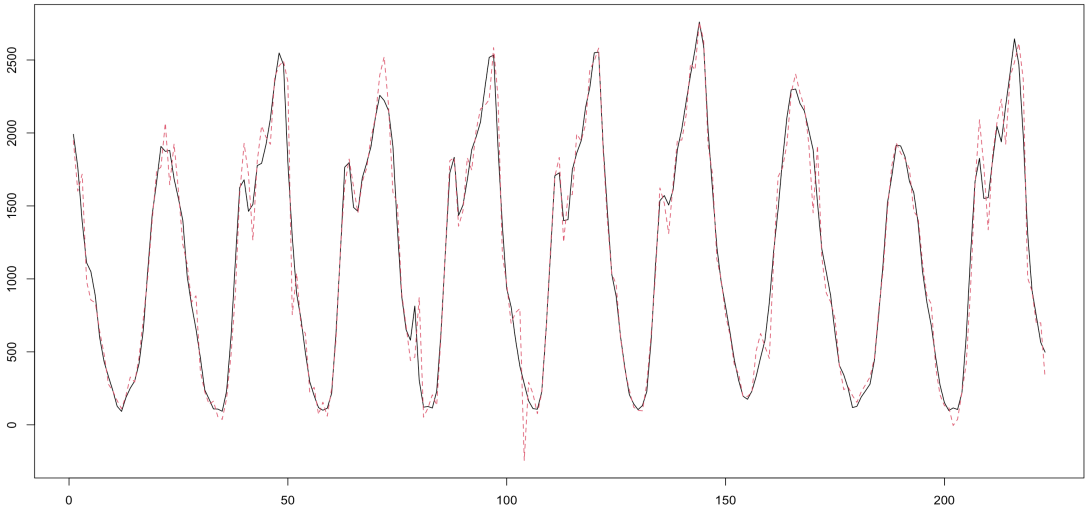


Fig.14. prediction on station no.354 by STARIMA

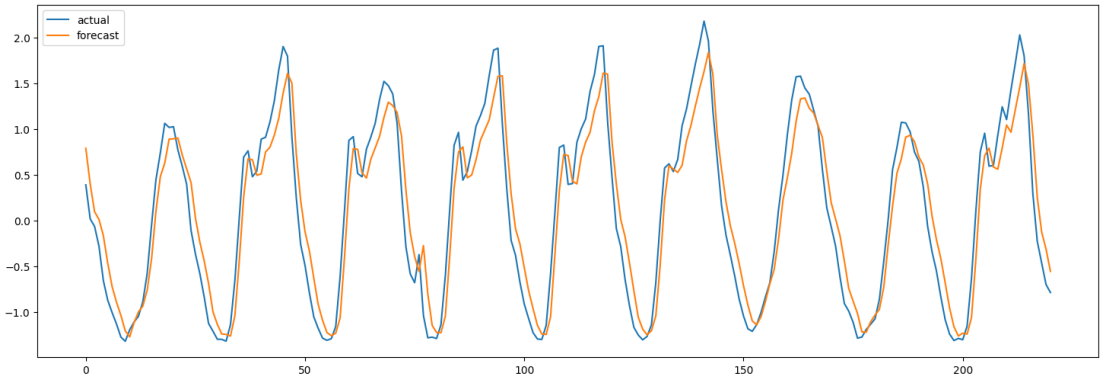


Fig.15. prediction on station no.354 by GNN-LSTM

Fig.14 and fig.15 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 weekdays (e.g. between ticks 100 and 150 in fig.14. & fig.15.) while GNN-LSTM seems to have underestimated the numbers. However, there are also several obvious divergences within the STARIMA predictions such as the prediction at around tick 100 in fig.14, 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 0.102. It may suggest that GNN-LSTM is approximately 2.5 times accurate in making predictions in this task.

Conclusions

The two models both produced predictions with relatively high accuracy, which may suggest the suitability of these 2 types of models to be applied on network traffic prediction tasks. GNN-LSTM have outperformed the STARIMA in terms of NMRSE. However, it tends to generalize the differences in measurement between weekdays and weekends, while STARIMA could distinguish these differences better.

There are might be limitations in this project that 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 a network according to the real-world geometries. Datasets with more detailed information on such fields could provide more accurate simulation of the geometries in modelling.

Secondly, when measuring autocorrelations of daily averaged time-series, a cyclic pattern of 7 days can be observed on fig.16, which may suggest that a weekly pattern exists in data and could be further considered within the STARIMA model.

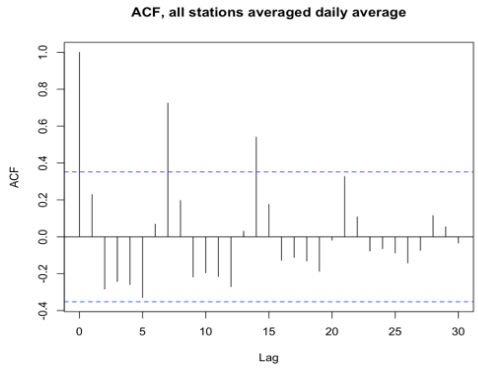


Fig.16. PACF after 1st differencing with seasonal lag 24

Furthermore, GNN-LSTM could be further improved by methods such as getting the weight matrix automatically updated (Cao et al. 2021), or incorporate self-attention matrices into the network structure (Guo et al. 2019).

References:

Anselin. L & Morrison. G. (2019). Distance-based spatial weights. [online] Spatial Data Analysis and Visualization Lab, University of Connecticut. Available at: https://spatialanalysis.github.io/lab_tutorials/Distance_Based_Spatial_Weights.html#distance-band-weights [Accessed 31 March 2023].

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J. and Zhang, Q. (2021). Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. arXiv:2103.07719 [cs]. [online] Available at: https://arxiv.org/abs/2103.07719.

Ding, Q.Y., Wang, X.F., Zhang, X.Y. and Sun, Z.Q. (2010). Forecasting Traffic Volume with Space-Time ARIMA Model. Advanced Materials Research, 156-157, pp.979–983. doi:https://doi.org/10.4028/www.scientific.net/amr.156-157.979.

Guo, S., Lin, Y., Feng, N., Song, C. and Wan, H. (2019). Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 33, pp.922–929. doi:https://doi.org/10.1609/aaai.v33i01.3301922.

Huang, B. and Carley, K.M. (2019). Residual or Gate? Towards Deeper Graph Neural Networks for Inductive Graph Representation Learning. arXiv:1904.08035 [cs, stat]. [online] Available at: https://arxiv.org/abs/1904.08035 [Accessed 31 Mar. 2023].

Khodadadi, A (2021). Time series forecasting for traffic. [online] Available at: https://keras.io/examples/timeseries/timeseries_traffic_forecasting/ [Accessed 31 March 2023].

Sorensen, J.Y. Salt Lake City traffic.(2022) [online] Available at: https://www.kaggle.com/datasets/johnyoungsorensen/salt-lake-city-traffic [Accessed 31 March 2023].

Utah Department of Transportation. Traffic statistics (UDOT). [online] Available at: https://www.udot.utah.gov/connect/business/traffic-data/traffic-statistics/ [Accessed 31 March 2023].

Yu, B., Yin, H. and Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, [online] pp.3634–3640. doi:https://doi.org/10.24963/ijcai.2018/505.