# FORECASTING OF URBAN TRAVEL TIMES

## Spatio-Temporal Analysis and Data Mining CEGEG076


Chun Siong Poh (Non-Parametric Approach Section 6.2)

Kah Siong Tan (Parametric Approach Section 6.1)

# 1 Table of Contents

## 2 Introduction

Smart City is a vision of many urban cities in the world. The smart city concept covers smarter transportation, a woe of many nations. Intelligent Transportation Systems (ITS) are advanced systems developed to handle urban traffic congestion. However, its success will depend on accurate real-time traffic information and predictive models to forecast the traffic pattern ahead in time. Predictive traffic model has been researched for decades and can be broadly categorised into parametric and non-parametric approaches. Improved predictive accuracy will provide vital information to travelers and city planners for long term intervention. The codes developed for this experiment are available in Github [1].

## 3 Literature Review

Parametric models include Kalman Filter, Markov model and classical time-series prediction [2]. Autoregressive Moving Average (ARMA), a time-series model, has been the cornerstone to spark development of Autoregressive Integrated Moving Average (ARIMA) and Space Time Autoregressive Integrated Moving Average (STARIMA) [3]. The key difference between ARIMA and STARIMA is the weight matrix to reflect configuration and topology of the spatial aspect.

Non-parametric models include Bayesian networks, Neural Networks (NN), Support Vector Machines (SVM) and distributed spatial-temporal weighted K-nearest neighbours (STW-KNN). These models are data driven and work by analysing the characteristics of a large dataset to predict with a certain accuracy [4]. In areas of short-term traffic prediction, neural network and evolutionary techniques has been gaining momentum [5]. Researchers demonstrated good success in using Long Short Term Memory (LSTM) [6], Feed Forward (FF), Non-linear Auto Regressive model with exogenous inputs (NARX) [7], Deep Belief Network (DBN) [8], Extreme Learning Machine (ELM) [2] and Multi-Layer Perceptron (MLP) [9] in the last 3 years. Notably, LSTM neural network was applied on Beijing traffic data and obtained favourable results over many other neural networks [6]. A more recent paper showed that Gated Recurrent Unit (GRU) achieved an edge over LSTM model [10].

# 4 Data Description

This experiment utilised data from Transport for London's (TFL) London Congestion Analysis Project (LCAP). The data contained 30 consecutive days of recording from 6am to 9pm between 2011 Jan 01 to 2011 Jan 30 in 5 minute intervals (180 observations per day) over 256 road links. The data was collected using Automatic Number Plate Recognition (ANPR) technology which operated in pairs to capture the travelling speed within each link. For this experiment, 17 links were selected as depict in the Figure 4-1 below.

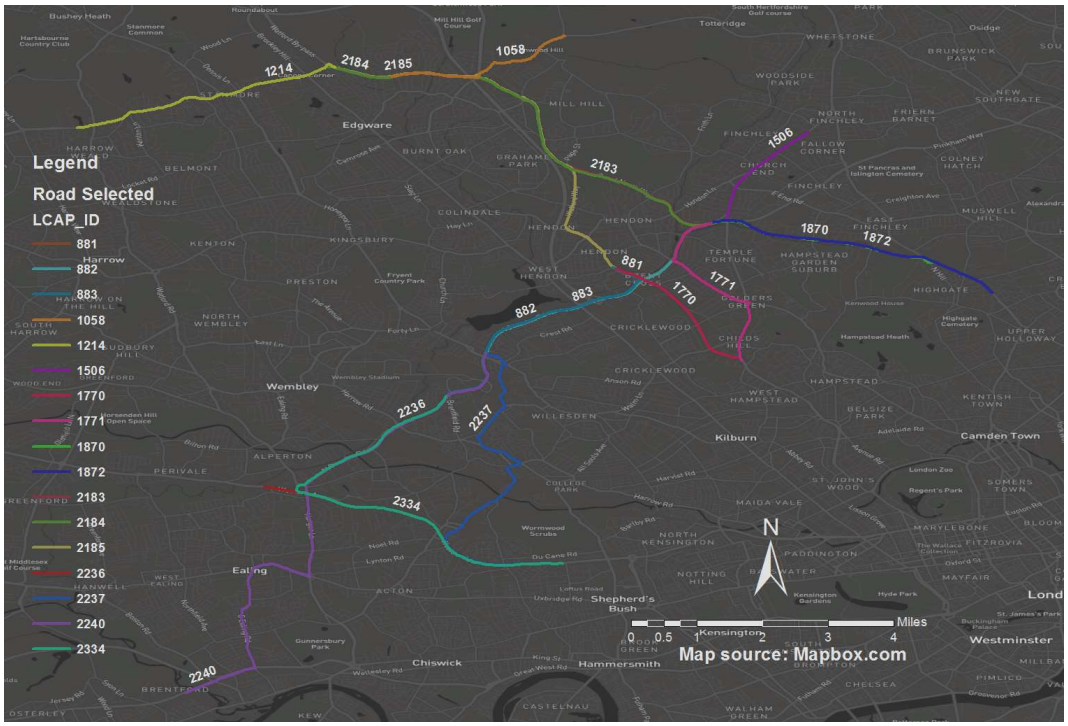



Figure 4-1: Selected Road Links

# 5 Exploratory Spatio-Temporal Data Analysis

## 5.1 Dynamic traffic conditions

Figure 5-1 shows the traffic flow of 17 links at 3 different time slots on a particular day, namely, 8am to 9am, 12pm to 1pm and 8pm to 9pm. Visual analysis showed a upward trend between links as the time shifted from morning to night The traffic flow across links varied differently as seen by the wide vertical variance.

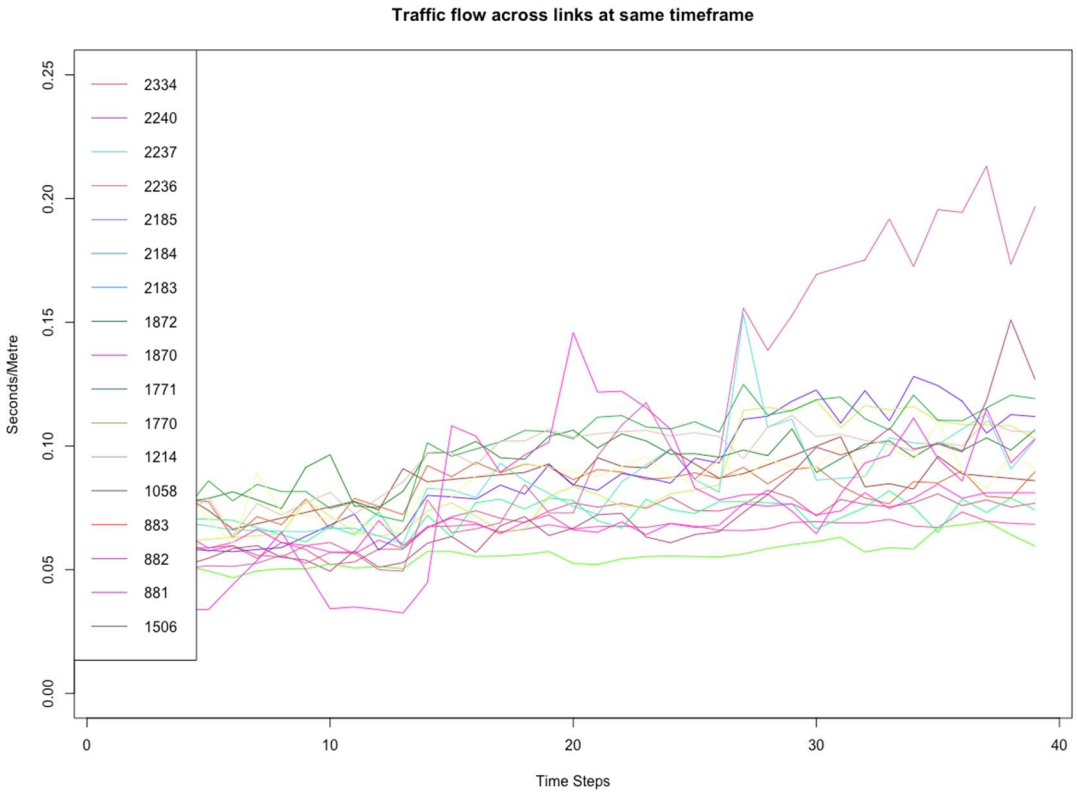



Figure 5-1: Traffic flows of all 17 links (Monday 8am-9am, 12pm-1pm, 8pm-9pm)

Figure 5-2 shows the traffic plot of 17 links across a 30-day observation period. The plots demonstrated that within each link, there could be out of ordinary traffic conditions that could result in uncommon variations such as spikes or troughs in traffic flow.

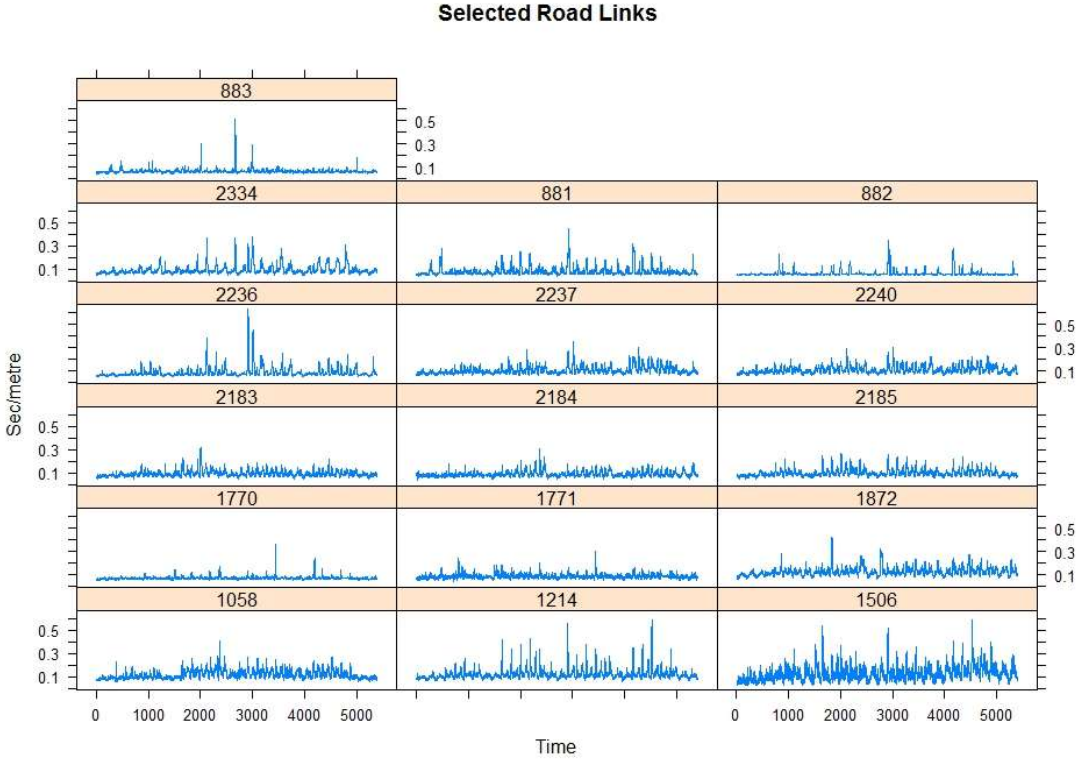



Figure 5-2: Traffic flow for 17 Links across 30 days

## 5.2 Non-random conditions

Was traffic flow in the city random or was there an underlying pattern that shaped it? Figure 5-3 shows the Quantile-Quantile (QQ) [11] plot of the traffic flow for the 17 links. If the distribution was normal, the blue line would have been in-line with the red line, which represents the theoretical normal distribution. It was clear that the distribution was not normal.

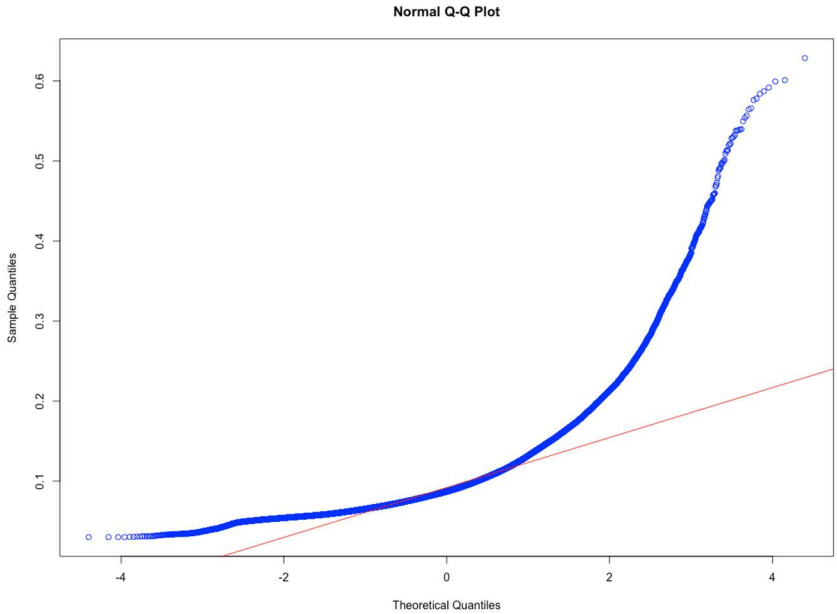



Figure 5-3: Quantile-Quantile plot of mean traffic of 17 links

The above analysis demonstrated the dynamism and non-random nature of this data set. These translate to potential challenges in building prediction models to handle such dynamic data.

## 5.3 Temporal characteristics

a. Was there an upward or downward trend in traffic flow over time?
b. Were there repeating patterns that in the traffic flow?

Logically, long term trends in traffic flow were not expected with only a month's worth of data. However, repeating patterns in traffic flow in a populated city would be expected.

### 5.3.1 Trends

In the interest of space, Figure 5-4 presents the trend analysis based on the traffic flows on only links 883 and 1770, and the mean traffic flow on all 17 links.

The red line represents a simple best-fit linear regression across the time-series. It was to be emphasized that a slope in regression would coarsely suggests a trend across 30 days, and not to be misunderstood to be a smooth

trend. The plot shows slight upward trend for links 883 and 1770 while the mean traffic plot across all links showed a more significant upward trend over 30 days.

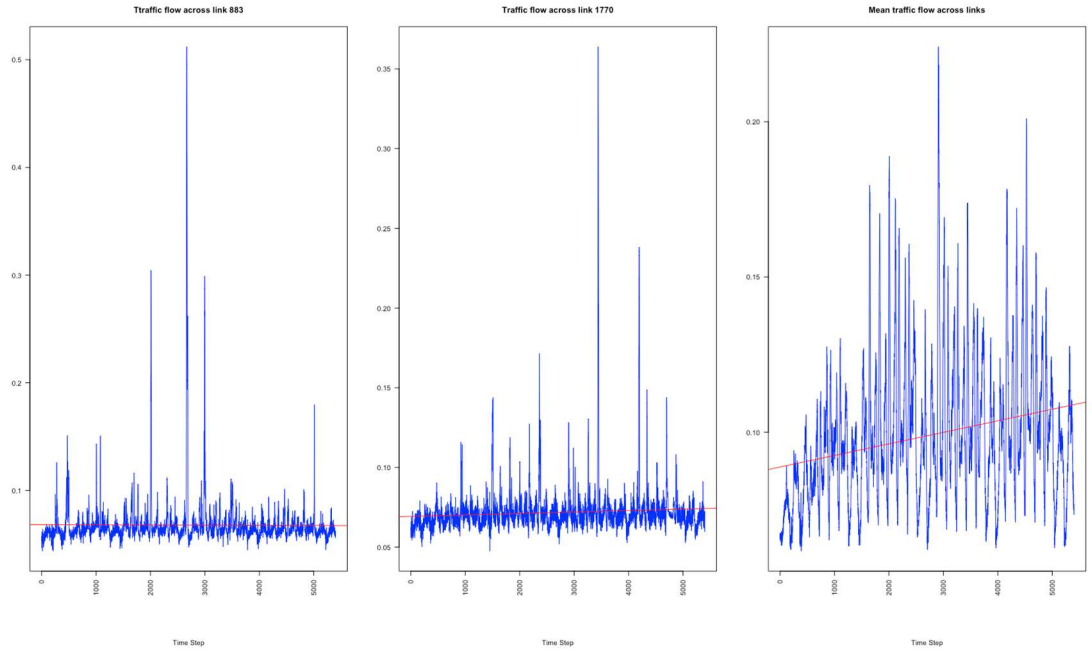



Figure 5-4: Trend analysis by regression

To confirm our observations, a statistical trend test was employed. Based on previous observations in Section 5.2, a non-normal distribution was noted in the dataset. As such, a non-parametric test that don't put assumptions on distributions was used. The Mann-Kendall Test [12] [13] was used and results tabulate in Table 5-1. With the rejection of the null hypothesis and a positive set of Z values, the results confirmed the trends observed above.

| Traffic flow | P-Value | Z-Value | Null Hypothesis Rejected? |
|---|---|---|---|
| Link 883 | 0.032748 | 2.1 | Yes |
| Link 1770 | < 2.22e-16 | 11.9 | Yes |
| All links | < 2.22e-16 | 21.4 | Yes |

Table 5-1: Mann-Kendall Trend Test results (Null hypothesis: No trend, Alternative hypothesis: Monotonic trend)

### 5.3.2 Seasonal patterns

Figure 5-5 presents a visual analysis of links 883 and 1770. For both links, a cyclic pattern at intervals of around 180 steps was observed and indicated with red vertical lines. For link 883, the green vertical line indicated a distinct repeating of pattern at intervals of around 1260 steps. This suggested more than one cyclic patterns on certain links.

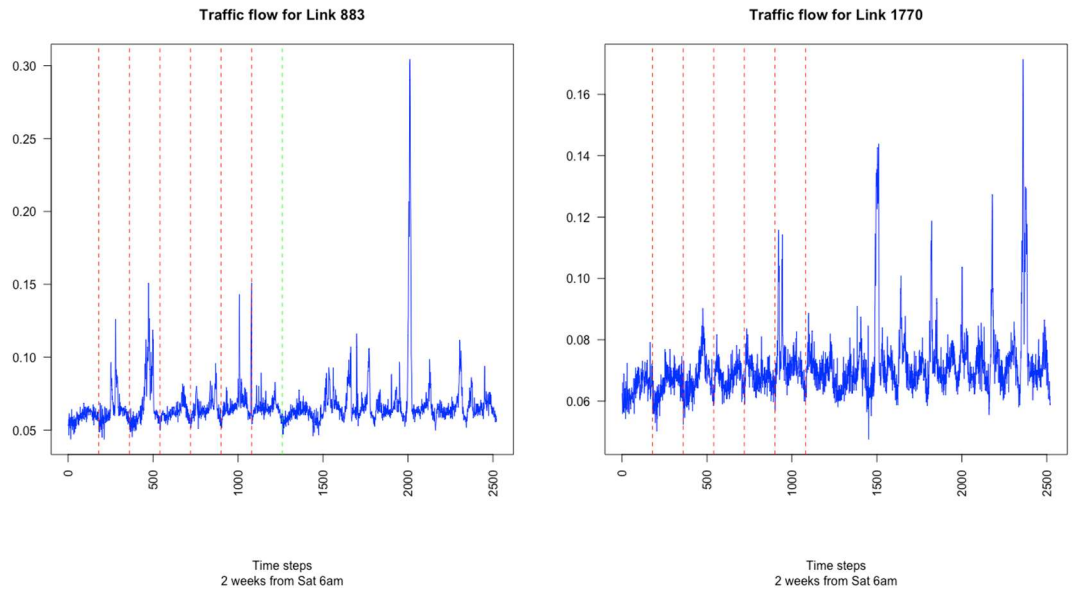



Figure 5-5: Pattern analysis on links 883 and 1770. Red and green dashed lines indicate 180 and 1260 cycles respectively.

Instead of error prone eyeballing, a periodogram [14] was employed to detect a maximum of two periods of a time-series. Table 5-2 lists periods detected on each link and showed that different links could demonstrate different periods of cyclic patterns and some appeared to overlap. This implied that every road link has its own traffic patterns and likely related to its location in the city. This observation supported the findings in Section 5.1. This information served as a heads-up for ARIMA modelling as it could mean a different seasonal model for each link.

| | Time steps | | Time steps in hours | |
|---|---|---|---|---|
| **Link** | **Seasons detected** | | **Seasons detected** | |
| 2334 | 182.86 | 89.30 | 15.24 | 7.44 |
| 2240 | 60.00 | 89.30 | 5.00 | 7.44 |
| 2237 | 60.00 | 89.30 | 5.00 | 7.44 |
| 2236 | 89.30 | 60.00 | 7.44 | 5.00 |
| 2185 | 89.30 | 1280.00 | 7.44 | 106.67 |
| 2184 | 89.30 | 1280.00 | 7.44 | 106.67 |
| 2183 | 89.30 | 3840.00 | 7.44 | 320.00 |
| 1872 | 3840.00 | 60.00 | 320.00 | 5.00 |
| 1870 | 182.86 | 89.30 | 15.24 | 7.44 |
| 1771 | 174.55 | 60.00 | 14.55 | 5.00 |
| 1770 | 89.30 | 174.55 | 7.44 | 14.55 |
| 1214 | 89.30 | 60.00 | 7.44 | 5.00 |
| 1058 | 3840.00 | 89.30 | 320.00 | 7.44 |
| 883 | 166.96 | 320.00 | 13.91 | 26.67 |
| 882 | 91.43 | 174.55 | 7.62 | 14.55 |

| | | | | |
|---|---|---|---|---|
| 881 | 174.55 | 153.60 | 14.55 | 12.80 |
| 1506 | 60.00 | 89.30 | 5.00 | 7.44 |

Table 5-2: Periodogram list periods detected on each of 17 links.

## 5.4 Spatial characteristics

In Figure 5-6, the relationship of different links was explored to see if there were any patterns. Due to space constraints, the scatterplot is only drawn using links on top left corner of Figure 4-1. Weak positive relationship was observed between the links. In another word, the links experienced congestion approximately the same period but the level of congestion were different. A stronger relationship can be seen between 881 and 882. This observation coincided with findings in Section 5.3.

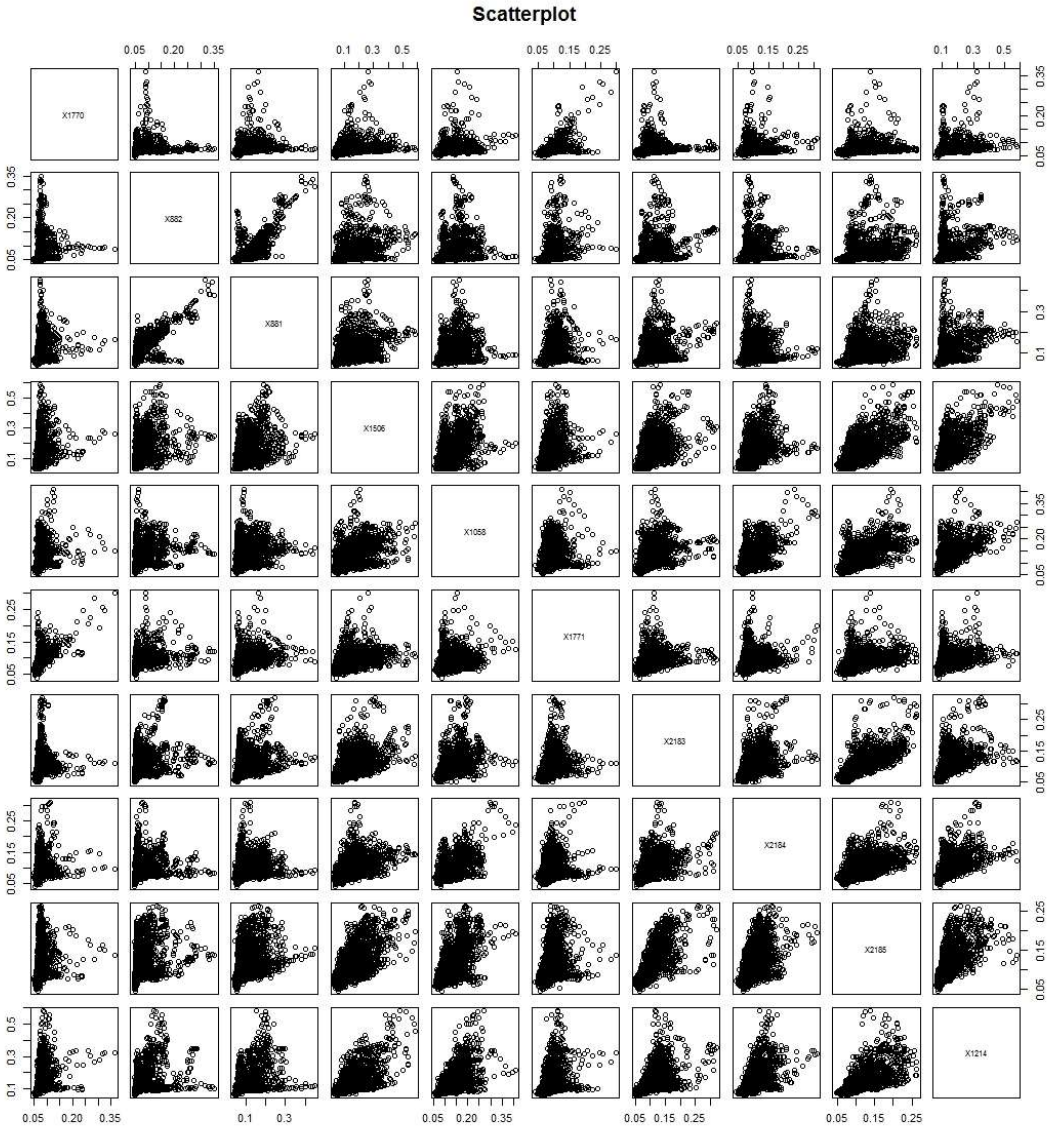



Figure 5-6: Scatterplot of Selected Road Links

# 6 Methodology & Results

The experiment was divided into parametric and non-parametric approach to make short term traffic prediction one step (i.e. 5 minute) ahead.

STARIMA was the chosen parametric approach and implemented in R. For the non-parametric approach, 4 variants of recurrent neural networks were implemented using TensorFlow(version 0.10rc). They were vanilla RNN, LSTM, GRU and stacked LSTM. TensorFlow was chosen for its extensive libraries and parallel processing capabilities to reduce development and training time.

Both approaches were trained using the first 23 days of data with last 7 days designated as test data. The models' predictions were evaluated based on Mean Absolute Percent Error (MAPE%) as shown below:

$$MAPE\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} X100\% \quad (1)$$

where $n$ is the number of observations, $y_i$ is the actual unit travel time and $\hat{y}_i$ is the predicted unit travel time.

## 6.1 Parametric (STARIMA)

### 6.1.1 Overview of STARIMA model development

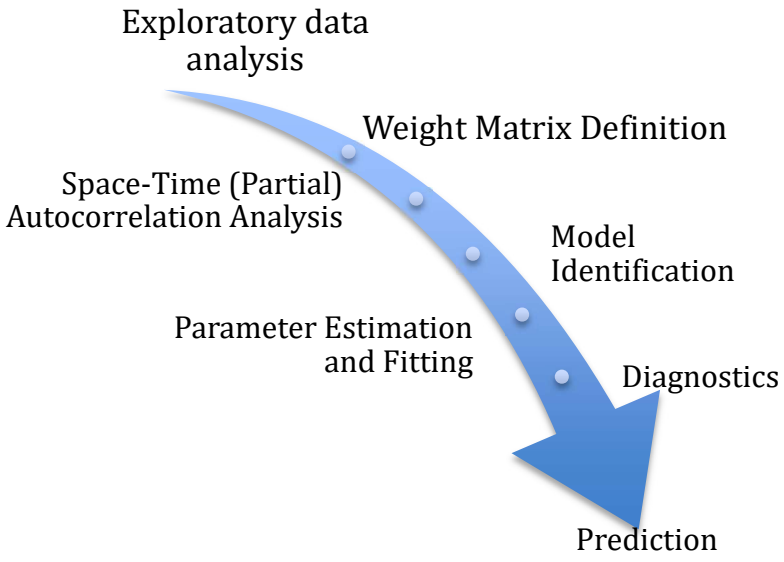



Figure 6-1: Overview of STARIMA model development

STARIMA was defined [15] as

$$z_i(t) = \sum_{k=1}^{p}\sum_{h=0}^{m_k}\phi_{kh}W^{(h)}z_i(t-k) - \sum_{l=1}^{q}\sum_{h=0}^{n_l}\theta_{lh}W^{(h)}\varepsilon_i(t-1) + \varepsilon_i(t) \quad (2)$$

where $z_i(t)$ is the space-time series variable at location $i$ and time $t$.
$p$ is the autoregressive order.
$q$ is the moving average order.
$k$ is the time lag.
$h$ is the space lag.
$W$ is the spatial weight matrix.
$m_k$ is the spatial order of the $k^{th}$ autoregressive term.
$n_l$ is the spatial order of the $l^{th}$ moving average term.
$\phi_{kh}$ is the autoregressive parameter at temporal lag $k$ and spatial lag $h$.
$\theta_{lh}$ is the moving average parameter at temporal lag $l$ and spatial lag $h$.
$\varepsilon_i(t)$ is the random normally distributed error at location $l$ and time $t$.

Figure 6-1 shows the steps taken in developing the STARIMA model.

### 6.1.2 Exploratory Data Analysis

In Section 5, an exploratory data analysis was performed on the LCAP data set. Insights received from the analysis were used in the model development.

### 6.1.3 Spatial Weight Matrix Definition

A spatial weight is best imagined as a structure of the spatial configuration and topology of data which describes the relationship between the entities within the dataset. The relationship could be in many forms, for example, distance between 2 points, shared boundaries between 2 sets of entities in an areal dataset or even a binary relationship between 2 entities. A Spatial Weight Matrix would be described as the pairwise representation of the relationship in a N by N matrix, where N represents the total number of entities being analysed. For the LCAP dataset, the spatial weights matrix was defined as the pairwise adjacency relationship between LCAP links. As an example, the following table described Link 2184 being adjacent to Links 2185 and 2183.

| | **Link 2184** | **Link 2185** | **Link 2183** |
|---|---|---|---|
| **Link 2184** | 0 | 1 | 1 |
| **Link 2185** | 1 | 0 | 0 |
| **Link 2183** | 1 | 0 | 0 |

Table 6-1: Unnormalised Spatial Matrix

Though not essential in this case, the Spatial Weighted Matrix defined above was row-normalised to remove effects on scale factors. The updated Spatial Weight Matrix for the above example is listed in Table 6-2.

| | Link 2334 | Link 883 | Link 882 |
|---|---|---|---|
| **Link 2334** | 0 | 0.5 | 0.5 |
| **Link 883** | 1 | 0 | 0 |
| **Link 882** | 1 | 0 | 0 |

Table 6-2: Normalised Spatial Matrix

The matrix was computed for the 17 links but is not printed for this report in the interest of space.

### 6.1.4 Space-Time (Partial) Autocorrelation Analysis and Model Identification

#### 6.1.4.1 Transformation for stationarity

To analyse the Space-Time Autocorrelation (STACF), the spatial weight matrix defined in Section 6.1.3 was used with the STACF function from the STARIMA_PACKAGE.R package to generate the following STACF plots.

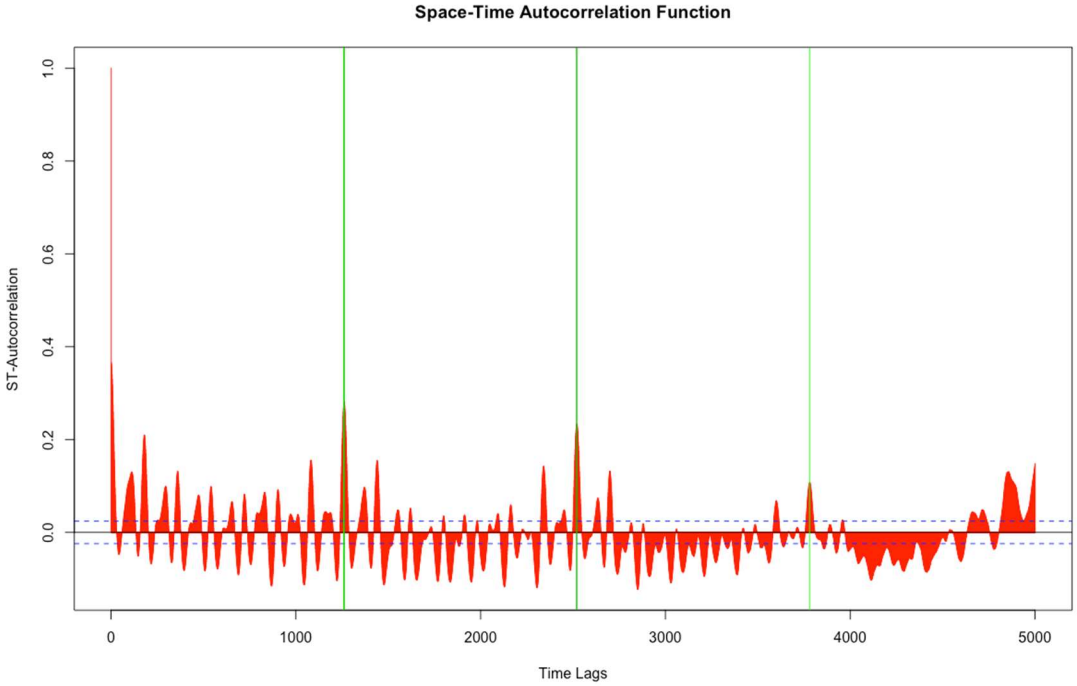



Figure 6-2: STACF of traffic flow at max lag of 5400 time steps (30 days)

Figure 6-2 shows no decay to zero, a clear indication that the space-time series was not stationary. Significant positive correlations were observed at lags 1260, 2520 and 3780, corresponding to exactly one week (E.g. Saturday

6am to following Friday 9pm). The waning correlation however implied a weak seasonal component.

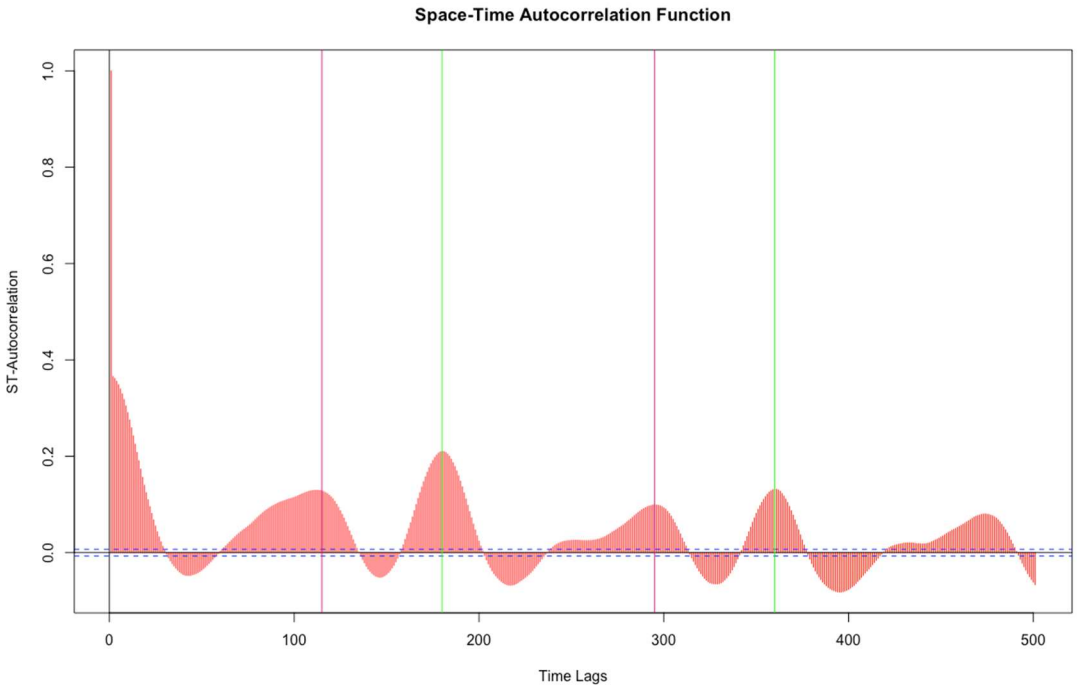
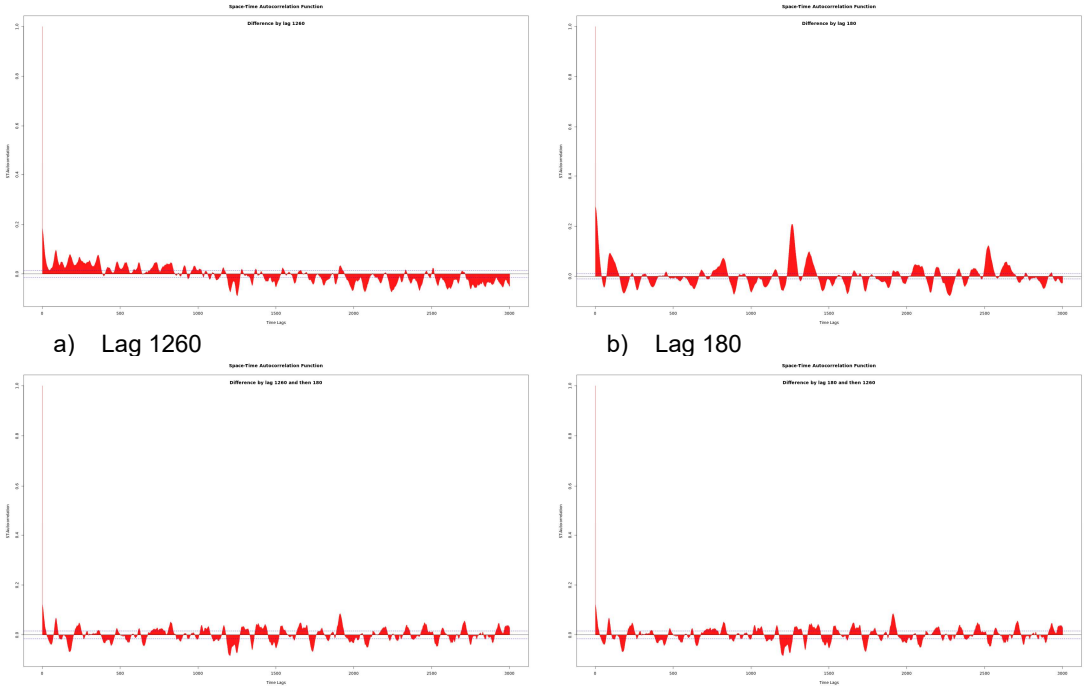



Figure 6-3: STACF of traffic flow at max lag of 500 time steps (Approx 1.5 days)

The STACF was replotted at a maximum lag of 500 for in-depth analysis. Figure 6-3 shows similar correlations at 180, 360 and 115, 295 respectively. The correlations corresponded to 3.35pm (pink vertical lines) and 9pm (green vertical lines) daily. The 9pm correlations could be explained by the daily observation cycles while the 3.35pm correlation explained by the beginning of evening peak at 4pm in London.

The above implied that an order of seasonal differencing of periods 180 and 1260 might be needed. Various lags differencing was performed and their STACF plots displayed in Figure 6-4.



a) Lag 1260 b) Lag 180

c) Lag 1260 then 180 d) Lag 180 then 1260

Figure 6-4: STACF of various differenced data

Although closer to stationarity, none of the seasonal differenced STACF showed signs of decaying to zero, thus none was in a stationary state. It also implied that the seasonality was not as strong as initially anticipated.

A differencing of 180 followed by 1260 resulted in an identical STACF plot if differenced the other way around. This showed that differencing operation was commutative by nature.

The remaining non-stationarity were removed with a further 1 lag difference (Decayed to zero).

Recall that a trend was observed in Section 5.3.1. As such, an order of difference of one lag was also applied to the original time series and resulting STACF plot in Figure 6-5. The plot cut off after one lag, stationarity was achieved.

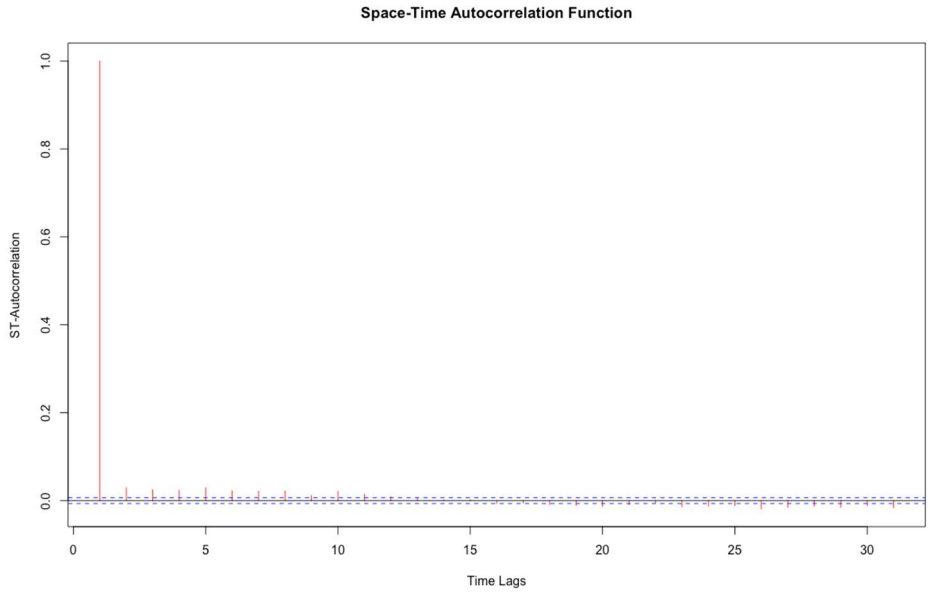



Figure 6-5: STACF of traffic data with a single order of differencing.

### 6.1.4.2 Model Identification

At this point, the following models were considered.

A. STARIMA (p,1,q)
B. STARIMA (p,1,q)$(P,1,Q)_{180}$
C. STARIMA (p,1,q)$(P,1,Q)_{1260}$
D. STARIMA (p,1,q)$(P,1,Q)_{180,1260}$

The STARIMA package did not support multiple seasonal components, as such, model D was dropped from consideration. The observations in Sections 5.3.2 and 6.1.4.1 implied that the seasonal components were not as strong as anticipated. A single order differencing of just one lag could achieve stationarity on the Space-Time series. Therefore, model A was selected to be further developed over models B and C.

The next task was to identify the order of Moving Average (MA) and Auto-correlation (AR) terms for STARIMA. The STACF of the differenced Space-Time series in Figure 6-5 was cut off after one lag, implying a MA order of 1. In Figure 6-6, the STPACF plot cuts off after lag implying an AR order of 1.

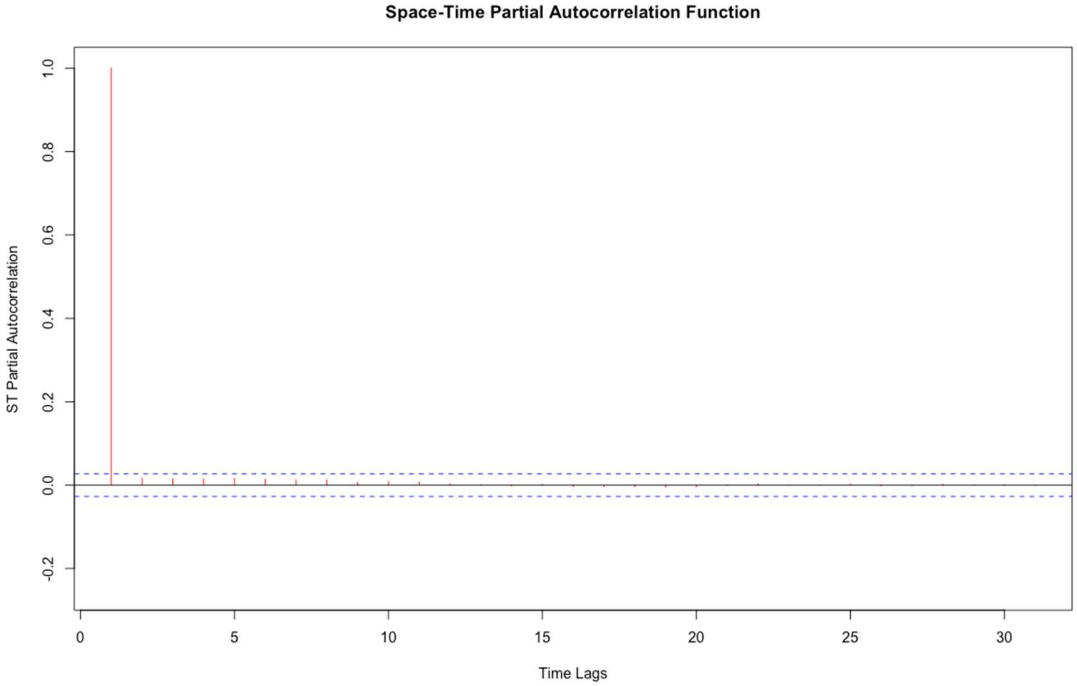



Figure 6-6: STPACF of traffic data with a single order of differencing.

The candidate model selected was STARIMA(1,1,1), however it cannot imply a best model for prediction.

### 6.1.5 Parameter estimation and fitting

Variants of the candidate model were fitted and the NRMSE for each link compared in Table 6-3.

| Model variant | Mean NRMSE across links | Lowest NRMSE |
|---|---|---|
| STARIMA(1,1,1) | 0.30988 | |
| STARIMA(0,1,1) | 0.31234 | |
| STARIMA(1,1,0) | 0.30987 | |
| STARIMA(1,1,2) | 0.30968 | |
| STARIMA(0,1,2) | 0.31218 | |
| STARIMA(2,1,2) | 0.30949 | ✓ |
| STARIMA(2,1,1) | 0.30982 | |
| STARIMA(2,1,0) | 0.30982 | |

Table 6-3: Mean NRMSE for various STARIMA models

STARIMA(2,1,2) scored the lowest mean NRMSE across all the links and was thus the best fit to the first 23 days of the data.

### 6.1.6 Diagnostics

The fitted model was checked for residuals which might not be handled by the model. If there were no significant correlations in STACF of residual data in fitted model, the residual would be random. Figure 6-7 shows that the residual STACF plot cuts off after 1 lag, thus the residual was random. The residual distribution is plotted in Figure 6-8 for all the 17 links. They all showed a normal

distribution, confirming the STACF observation of random residuals. The model is ready for prediction

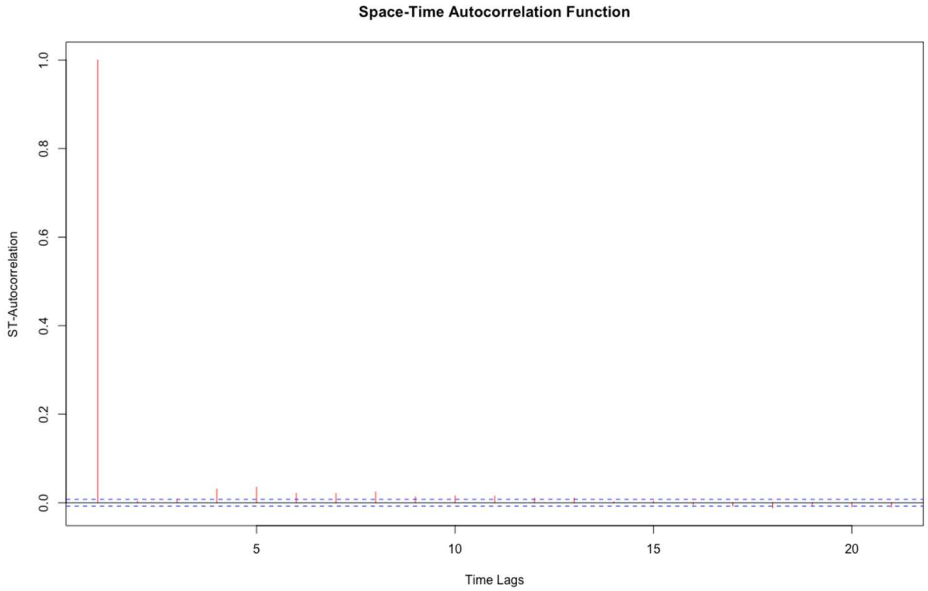



Figure 6-7: STACF of residuals

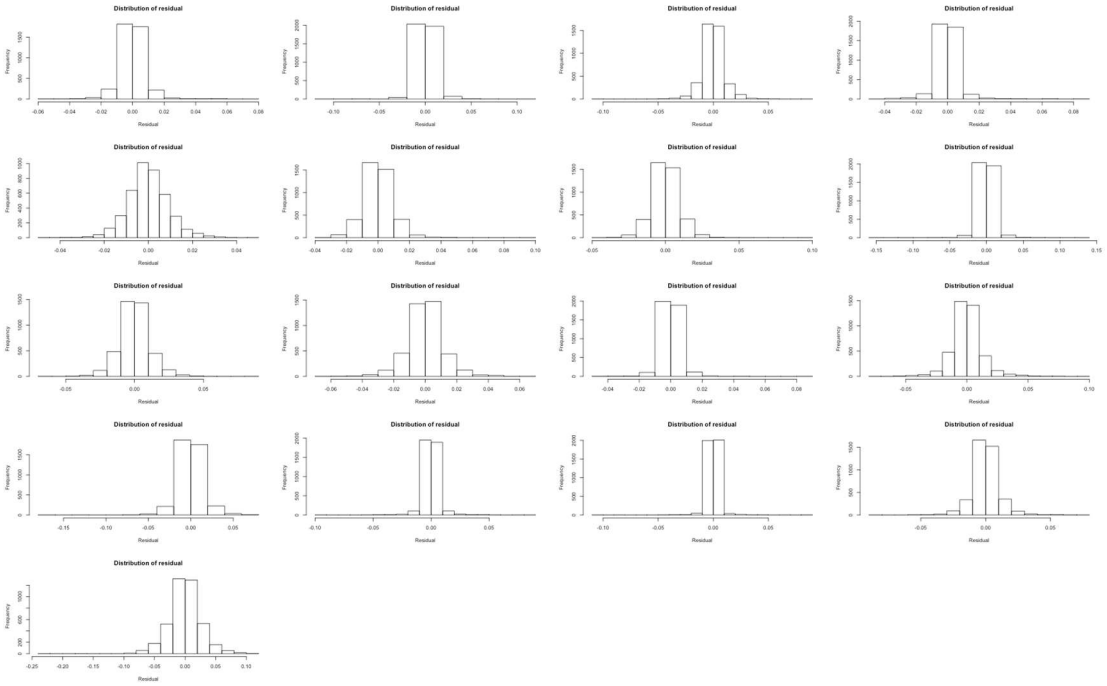



Figure 6-8: Histograms of residuals of each link

### 6.1.7 Prediction

The final step of STARIMA modelling was to perform prediction and determine the errors by comparing it with the test set.

| Link | MAPE |
|---|---|
| 2334 | 5.946917676 |
| 2240 | 6.130845496 |
| 2237 | 6.705016818 |
| 2236 | 4.533279811 |
| 2185 | 6.936152838 |
| 2184 | 7.613549313 |
| 2183 | 7.111850793 |
| 1872 | 5.561338027 |
| 1870 | 6.132586914 |
| 1771 | 8.961211573 |
| 1770 | 5.349565767 |
| 1214 | 7.076183412 |
| 1058 | 8.388946257 |
| 883 | 5.226658245 |
| 882 | 4.112017269 |
| 881 | 8.322006152 |
| 1506 | 13.87146103 |

Table 6-4: Prediction results over all links in MAPE.

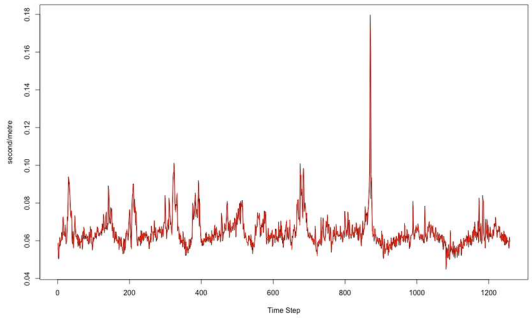



Figure 6-9: Prediction of link 883 over 7 days

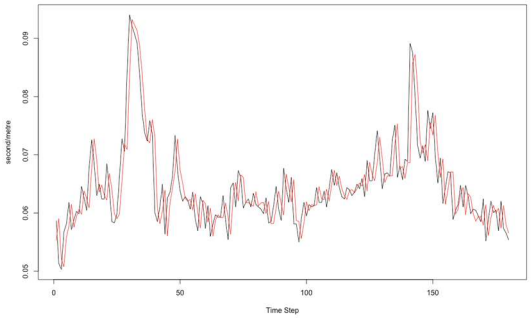



Figure 6-10: Prediction of link 883 over one day

A prediction of last 7 days based on one-step ahead was made. For uncluttered visual analysis, the results of one link was plotted. Figure 6-9 shows the prediction of Link 883 over 7 days. Figure 6-10 showed the same prediction but only plotted the first day of prediction for closer visual analysis. It was observed that although the model did well in predicting the general movement of the data, its prediction suffered from a time lag. Figure 6-9 also showed a slight reduction in prediction capability when put against spikes not so obvious in training data.

## 6.2 Non-Parametric (Recurrent Neural Network)

Inspired by the successful novel application of LSTM and GRU neural network to predict traffic speed [6] [10], the non-parametric experiment implemented 4 variants of recurrent neural network to model 17 road link as a single model. The recurrent neural networks along with brief description are listed in the table below.

| | **Recurrent Neural Network** | **Description** |
|---|---|---|
| 1. | Vanilla RNN [16] | Classic RNN |
| 2. | LSTM [17] | Extension of classic RNN to solve learning of long term dependencies. |
| 3. | GRU [18] | A simpler variant of LSTM |
| 4. | Stacked LSTM | Multi-layer LSTM to give the model more expressive powers |

Table 6-5: Four RNN Implemented for Experiment

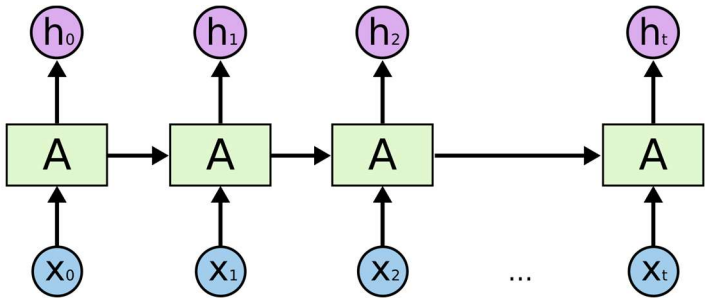
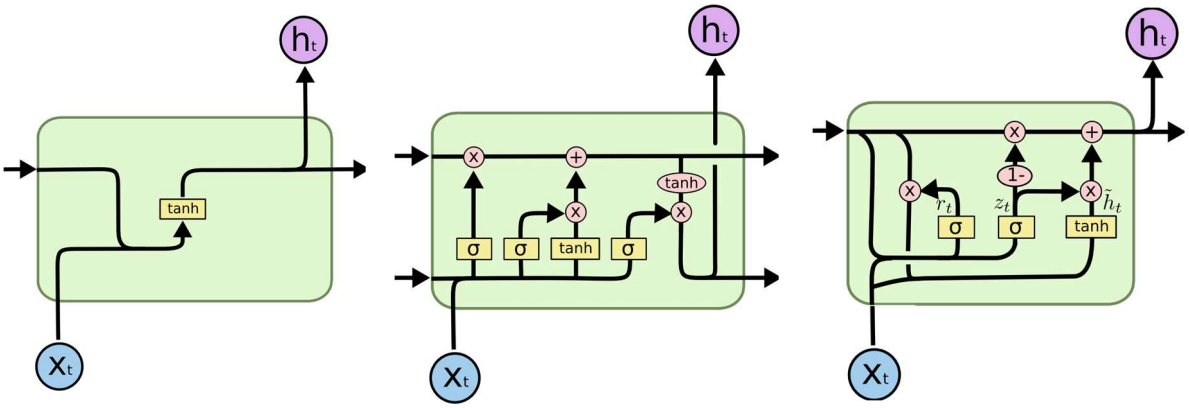



Figure 6-11: Unrolled Recurrent Neural Network [19]

The networks were implemented according to reference architecture as shown in Figure 6-11 above. In a chain like manner, $X_t$ takes in the cell state from the previous cell. The green box A is substituted with RNN, LSTM and GRU for the respectively network. The details of variants are shown in Figure 6-12 below.




Vanilla RNN LSTM GRU

Figure 6-12: RNN Cell, LSTM Cell and GRU Cell (Left to Right) [19]

The final output of the network was passed into a linear regression function to predict the traffic speed for the next step ahead.

$$\hat{y}_t = wh_t + b \tag{3}$$

where w is weight, b is bias, $h_t$ is the LSTM output and $\hat{y}_t$ is the predicted output.

The neural network error function was based on sum of squared defined as (4) where $y_t$ is the actual unit travel time and $\hat{y}_t$ is the predicted unit travel time.

$$Err \qquad \sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \tag{4}$$

Using the error function, the network was optimised using ADAM optimiser with independent parameters (i.e. learning rate, hidden dimension, layers) tuned for each network described in Table 6-5. ADAM is a first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [20].

Neural networks are susceptible to overfitting. Regularisation is a popular method that covers 3 main techniques namely, L1, L2 and dropout to generalise the model for unseen data. Dropout has shown to outperform other regularisation techniques in many situation [21]. In this experiment, dropout was implemented to randomly drop synapses based on a probability (i.e. 10%) to prevent overfitting.

Activation function such as sigmoid, hyperbolic tangent, elu, softplus and softsign are provided by Tensorflow. Through experiment, it was observed there was no significant difference for this set of data and hyperbolic tangent was selected as the activator.

The data used were described in Section 4. For non-parametric experiment, the neural networks were experimented using different combination of data:

1. Different window size (i.e. historical data points) was used for the prediction of the next point
   i.e. $Y_{t+1}$ predicted using $X_t$, $X_{t-1}$, $X_{t-2}$,..., $X_{t-n}$
2. Data differencing. The derivate of historical data were used instead of the actual value.
   i.e. $Y_{t+1}$ predicted using $X_t$, $X_t$-$X_{t-1}$, $X_{t-1} – X_{t-2}$,…..

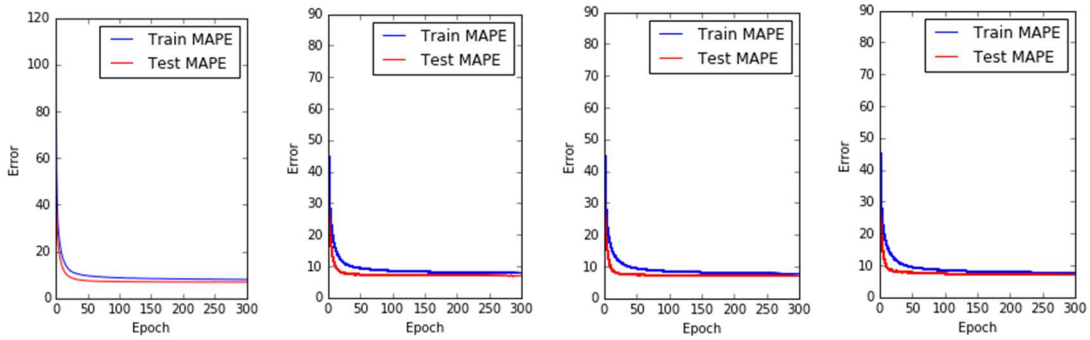
For the experiment, the neural networks were trained using window size of 1 to 5 historical points with data differencing (DD) set to true or false. Window size 1 means the network is trying to predict $Y_{t+1}$ using only $X_t$ and the previous network state. Table 6-6 shows the average error rate in MAPE for 17 road links against the test set. The networks collectively train all road links to produce a single model.

| **Window Size** | **RNN** | | **LSTM** | | **GRU** | | **Stacked LSTM (2 Layers)** | |
|---|---|---|---|---|---|---|---|---|
| | **No DD** | **DD** | **No DD** | **DD** | **No DD** | **DD** | **No DD** | **DD** |
| 1 | 7.13 | - | 7.08 | **-** | **7.00** | - | 7.45 | - |
| 2 | 7.17 | 7.19 | 6.98 | **6.85** | 6.96 | 6.93 | 7.24 | 7.22 |
| 3 | 7.37 | 7.17 | 7.03 | **6.91** | 7.13 | 7.01 | 7.42 | 7.28 |
| 4 | 7.83 | 7.28 | 7.10 | **6.95** | 7.35 | 7.10 | 7.59 | 7.30 |
| 5 | 8.15 | 7.39 | 7.20 | **7.00** | 7.65 | 7.24 | 7.88 | 7.48 |

Table 6-6: MAPE Result on Test Set

Comparatively, LSTM with data differencing had the best performance. It was observed that using more historical points worsen the results. Between LSTM and GRU, the difference was relatively small which was also similarly observed by [10]. The 2-layer LSTM did not perform as expected which could be due to the complexity of the layering effect.

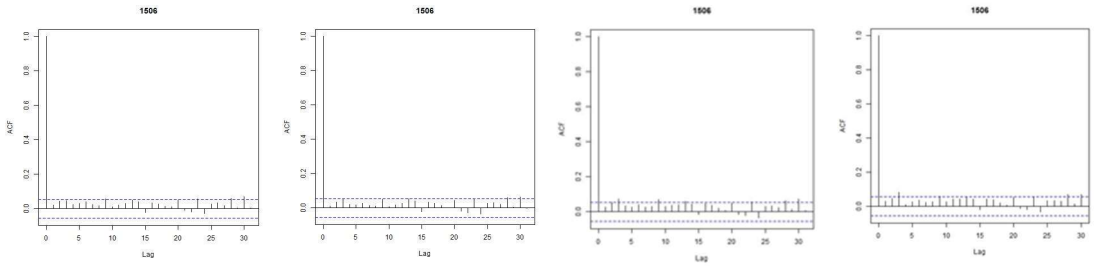
Focusing on LSTM with data differencing, the experiment examined the training and test epoch. It was observed the test set MAPE falls rapidly to below 10% error within the first 10-20 epochs and gradually stabilise with marginal improvement between 250 to 300 epochs. Further training beyond 300 epochs yield insignificant improvement. The hidden dimension for each network were 68, 102, 136 and 170 respectively to account for the larger number of input vectors due to window size. It was a coincident that the test set performs better than training set.




Window 2 Window 3 Window 4 Window 5

Figure 6-13: Train and Test Epoch for LSTM with Data Differencing

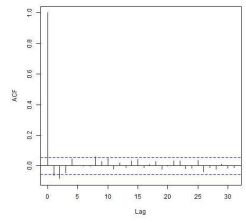
To further analyse the performance of LSTM, the autocorrelation frequency (ACF) of the predicted result were compared. In Figure 6-14, the ACF of LSTM across various window size were observed to be similar as shown using road link 1506. The remaining road links also exhibited similar behaviour across different window size and was not shown in the interest of space.



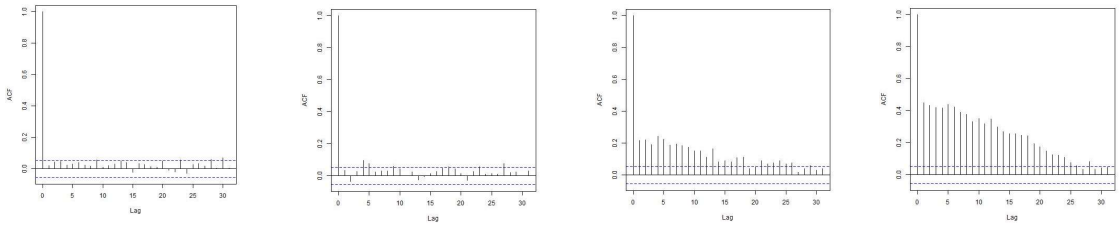

Window 2 Window 3 Window 4 Window 5

Figure 6-14: ACF for LSTM with different window size

Next, the experiment investigated the residual ACF across the 16 road links using LSTM with window size 2. From Figure 6-15, 50% of the road links had residual autocorrelation close to the 95% confidence interval while the rest exhibited autocorrelation.



883



1506 1771 881 882

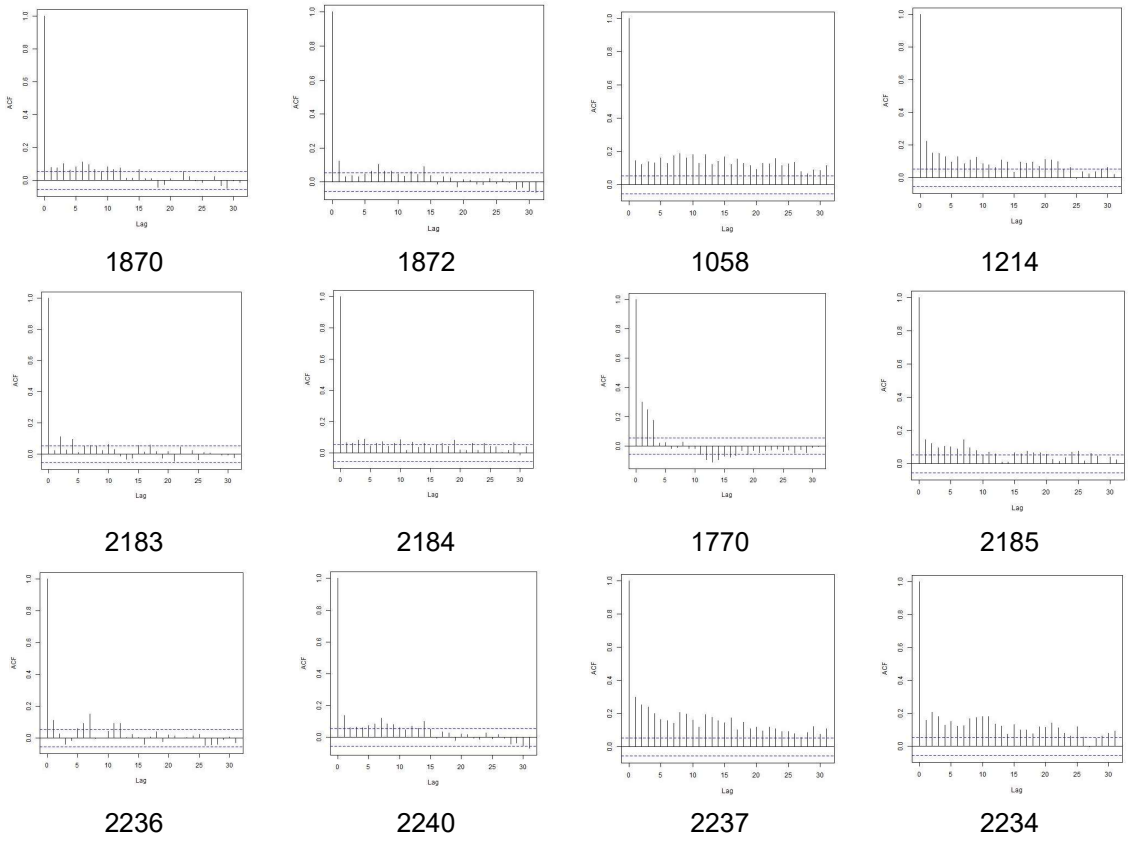
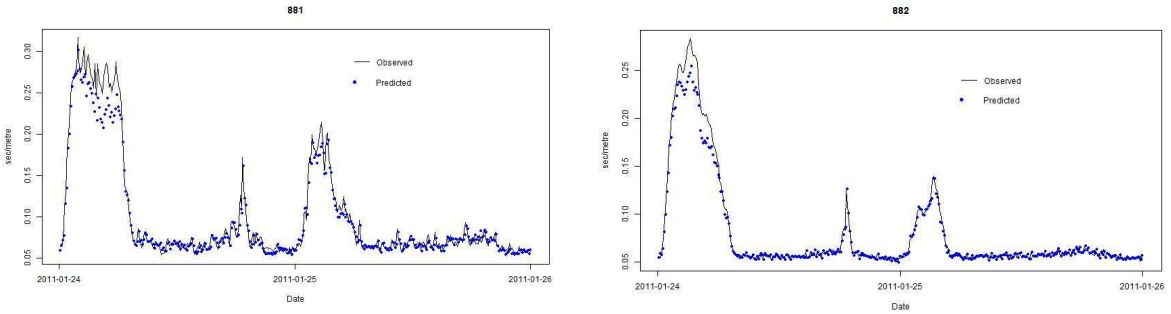



Figure 6-15: ACF for LSTM with 2 history points

Using the predicted and true value, the first 4 road links (i.e. 881, 882, 883 and 1506) were selected and plotted in Figure 6-16. The time was reduced to 2 days (i.e lag 360 = 180 x 2) to provide better visual comparison. It was observed that LSTM's prediction reflected the observed value with subtle differences in some areas. In 881 and 882, it was observed that the LSTM failed to predict the massive peak very well.



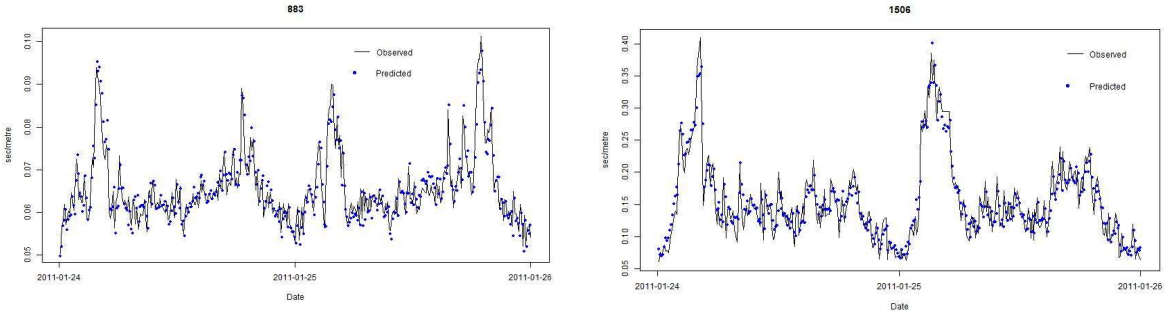



Figure 6-16: Predicted vs Observed for 4 Selected Road Links

# 7 Discussion and Conclusions

## 7.1 Performance

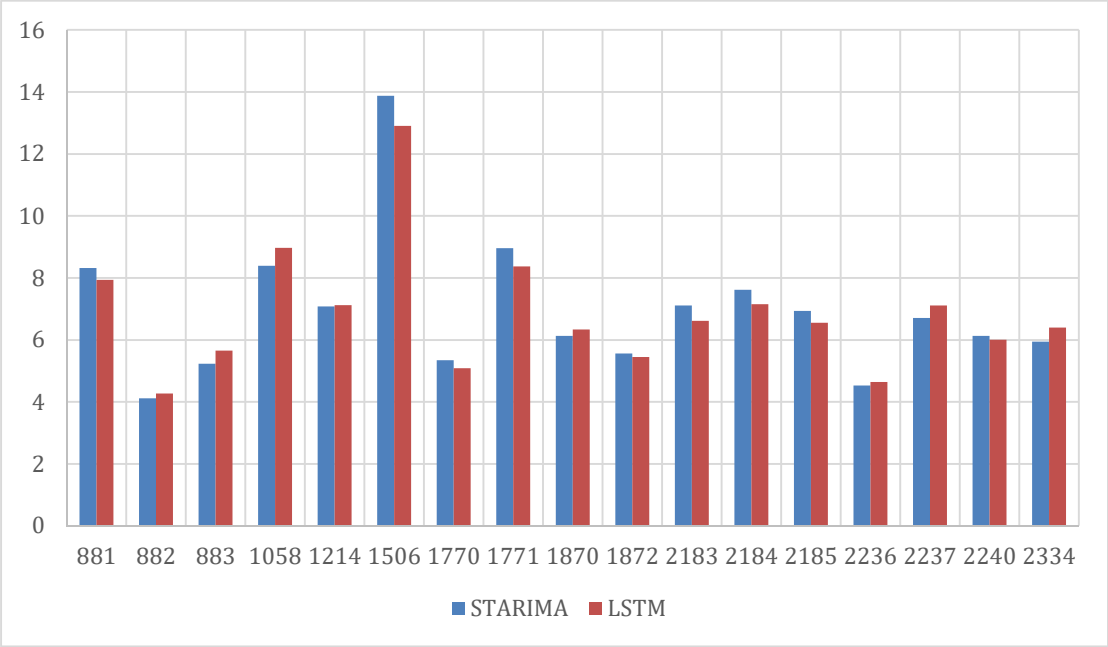



Figure 7-1: Performance across the study area

STARIMA and LSTM achieved the best performance of 6.93% and 6.85% MAPE respectively. Figure 7-1 shows the performance of the models across the study area. It can be observed that the difference between the two are insignificant. The LSTM did not perform as good as expected which could be due to modelling of 17 road link as a single model.

Figure 7-2 compares the prediction of 2 approaches based on road link 2334. Both approaches closely predicted the traffic condition including the peaks.

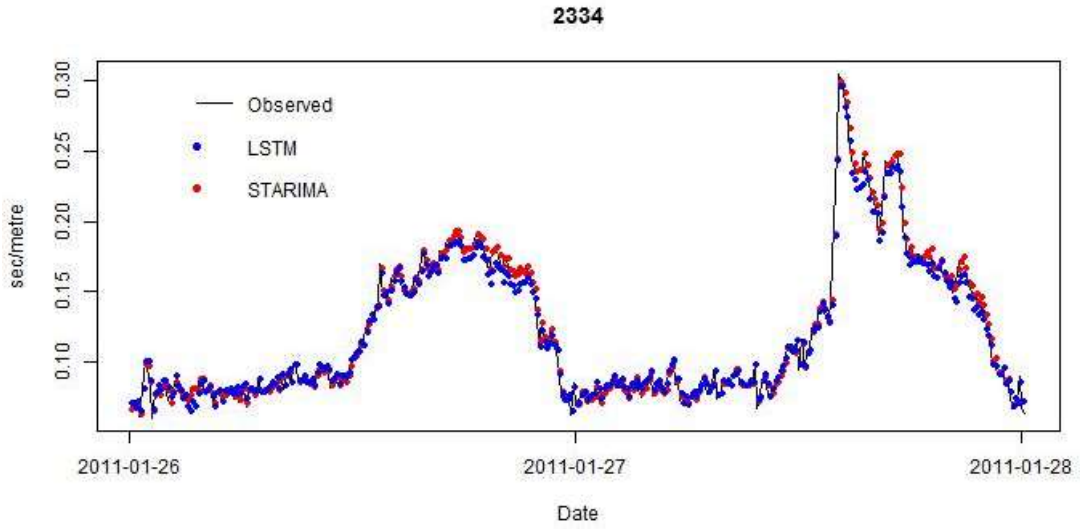



Figure 7-2: Observed vs Predicted STARIMA and LSTM

In term of interpretability, STARIMA is a well-defined statistical approach and provides scientific basis on its predicative capabilities. This also meant that in-depth analysis and interpretation were required to achieve a good model, making it slightly more involved to implement. On the other hand, neural network is akin to black box which makes it difficult to explain its capabilities in prediction despite achieving results on par with well-defined approaches.

In terms of running time, STARIMA's training time is highly dependent on the eventual model. In this experiment, it is fast compared to LSTM which took 30 second vs 20 minute to train respectively. If seasonal components are to be taken into consideration, it would result in a much longer training time for STARIMA.

## 7.2 Other findings

This section briefly discussed other findings from experiments not elaborated in the report due to reporting constraints.

### 7.2.1 Data Segregation

It was initially hypothesized that a better model could be achieved if the data is segregated into individual day of week or group of days. For example, a training set of Mondays to train and predict only Mondays, or a training set of Mondays through Thursdays. However, the results are not too different from final result used in this report. On top of this, this requires several models for ARIMA or STARIMA approaches.

### 7.2.2 Data Aggregation

In situations where ARIMA model had to deal with very high periods of seasonal components, there is a substantial increase in memory demand and training

time. One solution to this problem is to aggregate the data into 1 or even 6 hour intervals. However, such aggregation reduced the prediction precision of the model and made it harder to fairly compare against approaches that thrived on the availability of data points such as Neural Networks.

### 7.3 Future Works

In developing the STARIMA model, the weight spatial matrix was defined through a simple thought process. More work could be done in exploring the relationship between the links to achieve a Spatial Weight Matrix that would better handle the inter-dynamics between links and achieve a stronger predicative capability across links.

In the non-parametric arena, RNN has been gaining a lot of traction and is applied in domain ranging from natural language processing to image processing. Attention is first proposed by [22] to allow the model to search for parts of a source sentence that are relevant to predict a target word. The same concept could be extended to time series prediction to allow the model to focus on relevant historical points for its prediction. Secondly, the RNN could be extended into a bidirectional RNN where the network would be able to look into the past and future state [23]. This would provide more context for the network and should achieve better result.

References

[1] Kah Siong, Tan and Chun Siong, Poh, "STDM2017," 24 March 2017. [Online]. Available: https://github.com/jax79sg/stdm2017.

[2] J. X. Z. X. X. L. Dong Wang, "Short-term Traffic Flow Prediction based on Ensemble Real-time Sequential Extreme Learning Machine under Non-stationary Condition," *IEEE,* 2016.

[3] P. E. Pfeifer and S. J. Deutsch, "A Three-Stage Iterative Procedure for Space-Time Modeling," *TECHNOMETRICS,* pp. 35-47, 1980.

[4] G. Chang, S. Wang and X. Xiao, "Review of Spatio-temporal Models for Short-term Traffic Forecasting," *IEEE,* pp. 8-12, 2016.

[5] E. I. Vlahogianni, M. G. Karlaftis and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," 2014.

[6] X. Ma, Z. Tao, Y. Wang, H. Yu and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," 2015.

[7] G. Fusco, C. Colombaroni, L. Comelli and N. Isaenko, "Short-term traffic predictions on large urban traffic networks: applications of network-based machine learning models and dynamic traffic assignment models," 2015.

[8] C. Siripanpornchana, S. Panichpapiboon and P. Chaovalit, "Travel-Time Prediction With Deep Learning," 2016.

[9] M.-K. Arezou and Mo Jamshidi, "Leveraging Machine Learning Algorithms to Perform Online and Offline Highway Traffic Flow Predictions," 2014.

[10] R. Fu, Z. Zhang and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," 2017.

[11] ETH zurich Department of Mathematics, "R: Quantile-Quantile Plots," [Online]. Available: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/qqnorm.html.

[12] Pacific Northwest National Laboratory, "Mann-Kendall Test For Monotonic Trend," [Online]. Available: http://vsp.pnnl.gov/help/Vsample/Design_Trend_Mann_Kendall.htm.

[13] T. Pohlert, "Non-Parametric Trend Tests and Change-Point Detection," 14 May 2016. [Online]. Available: https://cran.r-project.org/web/packages/trend/trend.pdf. [Accessed 9 Feb 2017].

[14] PennState Eberly College of Science, "PennState Eberly College of Science: The Periodogram," 2017. [Online]. Available: https://onlinecourses.science.psu.edu/stat510/node/71. [Accessed 9 Feb 2017].

[15] D. J. Haworth, "Space-Time Exploratory Data Analysis (STEDA)," London.

[16] J. L. ELMAN , "Finding Structure in Time," *Cognitive Science,* vol. 14, p. 179–211, 1990.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[18] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," 2014.

[19] C. Olah, "Understanding LSTM Networks," 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 02 Feb 2017].

[20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," San Diego, 2015.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2014.

[22] D. Bahdanau, K. Cho and Y. Bengio, "NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE," 2015.

[23] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE TRANSACTIONS ON SIGNAL PROCESSING,* vol. 45, no. 11, pp. 2673-2681, 1997.

[24] X. Min, Q. Chen, T. Zhang and Y. Zhang, "Short-Term Traffic Flow Forecasting of Urban Network Based on Dynamic STARIMA Model," St. Louis, MO, USA, 2009.

[25] S.-L. Lin, H.-Q. Huang, D.-Q. Zhu and T.-Z. Wang, "The Application of Space-Time ARIMA Model on Traffic Flow," Baoding, China, 2009.

[26] D. Wang, J. Xiong and X. Li, "Short-term Traffic Flow Prediction based on Ensemble Real-time Sequential Extreme Learning Machine under Non-stationary Condition," *IEEE,* 2016.

[27] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014.