Analysing Travel Time Patterns in London

CEGE0042: Spatial-temporal Data Analysis and Data Mining

Ng Jia Wen, Junju Ng, Ju Young Park & Xulan Huang

Contents

1 Introduction

Understanding and predicting urban traffic flows is vital, especially in London, a global financial hub that is home to many international headquarters (HQs). 33% of European HQs of Global Fortune 500 are in London and over 40% of the world's foreign equities are traded in London (Global Alliance of SMEs 2016). London is also a transport hub to many other countries and cities, with its airport terminals seeing more than 100,000 flights per month (Global Alliance of SMEs 2016). As such, it is of utmost importance for London to maintain its infrastructure and ensure the highest of standards.

The strains and stresses of being a global city has made London the second most congested city in Europe (Williams 2018), behind Moscow, and the sixth most gridlocked city in the world (ITV 2019), with commuters spending an average of 227 hours a year stuck in traffic (Adams 2019). According to Inrix's 2018 Global Traffic Scorecard, traffic congestion has costed UK drivers 7.9 billion pounds, with London drivers losing up to Â£1,680 a year (Adams 2019). This highlights an urgency to improve urban traffic flows and one way to do so is to accurately predict and forecast travel times which can help aid the government's tranport policies and also inform road users of the estimated travel times.

As such, this project will be looking at forecasting travel time using data from Transport for London. In particular, the project will focus on the Westminster borough.

1.1 Data Description

Travel time data collected using automatic number plate recognition technology on road networks in London in January 2011 was obtained from the UJTWorkspace provided. A study area, the Borough of Westminster, was selected for further analysis given its location in the heart of London. All 25 road segments in Westminster were included in the analysis. The study area and road segments studied are shown below.

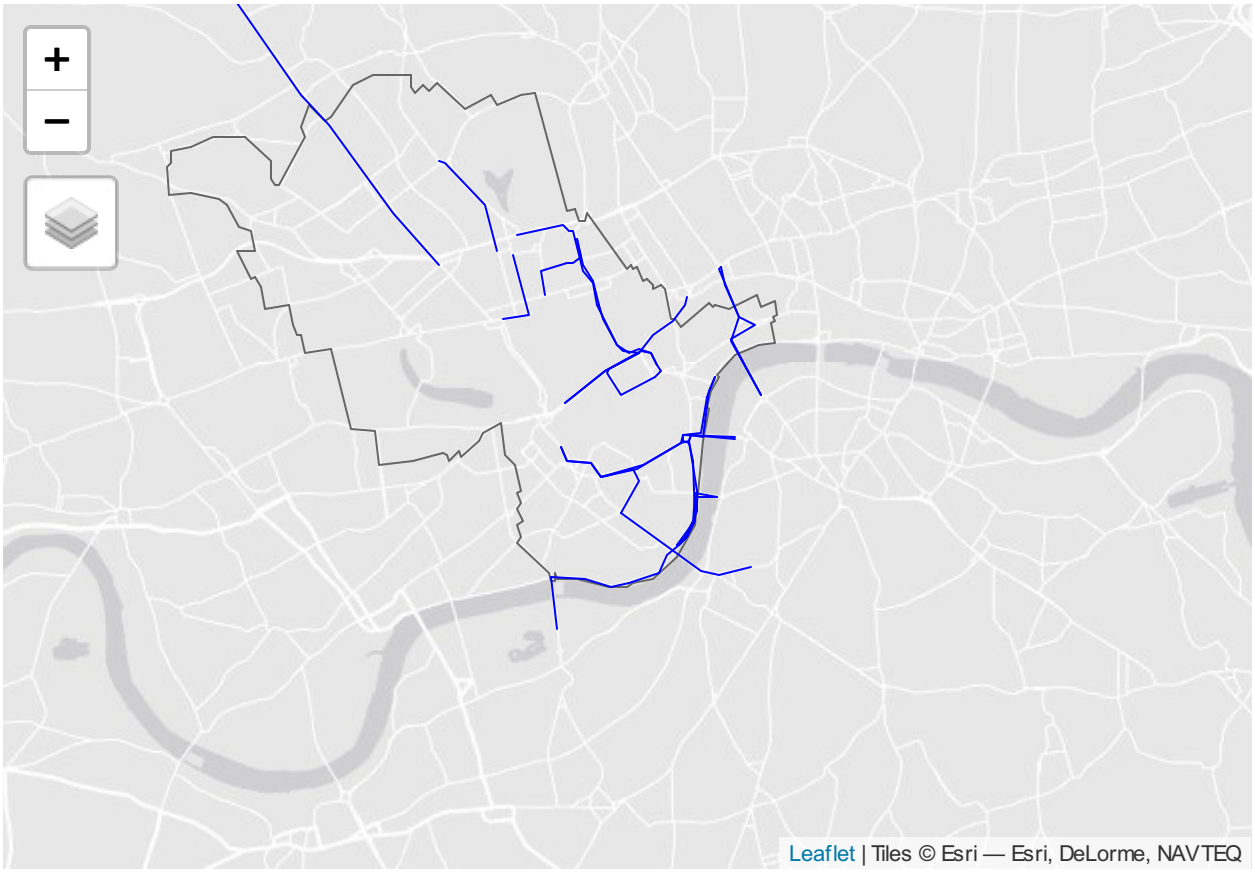


Figure 1: Plot of road segments in Westminster. The black polygon marks the boundaries of the Borough of Westminster while the blue lines represent the road segments that will be examined in this study.

1.1.1 Spatial Weight Matrix

A first-order adjacency matrix is provided in the UJTWorkspace. Only road links in the study area, Westminster, were extracted from the adjacency matrix provided. The provided adjacency matrix was then modified such that the sum of each row is equal to 1 (or 0) for use in later analysis. The modified adjacency matrix shown below will used to represent spatial characteristics in the spatial-temporal approaches which will be implemented (e.g. STARIMA, and ST-SVR modelling approaches).

```
##      1883 1884 420 423 2468 1576 1593 1613 1616 1460 1412 1413 2112 2364
## 1883    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 1884    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 420     0    0 0.0   0  0.0    0  0.0    0  0.5    0    0    0    0    0
## 423     0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 2468    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 1576    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 1593    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 1613    0    0 0.0   0  0.5    0  0.0    0  0.0    0    0    0    0    0
## 1616    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 1460    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 1412    0    0 0.5   0  0.0    0  0.5    0  0.0    0    0    0    0    0
## 1413    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
```

```
## 2112    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 2364    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 2363    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 2173    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 2318    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 2433    0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 435     0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 524     0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 437     0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
## 469     0    0 0.0   0  0.0    0  0.0    0  0.0    0    1    0    0    0
## 425     0    0 0.0   0  0.5    0  0.0    0  0.0    0    0    0    0    0
## 463     0    0 0.0   0  0.0    0  1.0    0  0.0    0    0    0    0    0
## 897     0    0 0.0   0  0.0    0  0.0    0  0.0    0    0    0    0    0
##       2363 2173 2318 2433 435 524 437 469 425 463 897
## 1883    0    0  0.0  0.0 0.0   0   1   0   0   0   0
## 1884    0    1  0.0  0.0 0.0   0   0   0   0   0   0
## 420     0    0  0.0  0.5 0.0   0   0   0   0   0   0
## 423     0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 2468    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 1576    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 1593    0    0  0.0  0.0 0.0   0   0   1   0   0   0
## 1613    0    0  0.0  0.0 0.5   0   0   0   0   0   0
## 1616    0    0  0.0  0.0 0.0   1   0   0   0   0   0
## 1460    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 1412    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 1413    0    0  0.0  0.0 0.0   0   0   1   0   0   0
## 2112    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 2364    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 2363    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 2173    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 2318    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 2433    0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 435     0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 524     0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 437     0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 469     0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 425     0    0  0.5  0.0 0.0   0   0   0   0   0   0
## 463     0    0  0.0  0.0 0.0   0   0   0   0   0   0
## 897     0    0  0.0  0.0 0.0   0   0   0   0   0   0
```

2 Exploratory Analysis

An exploratory analysis of the travel time data will be conducted in this section to better understand the characteristics of the data.

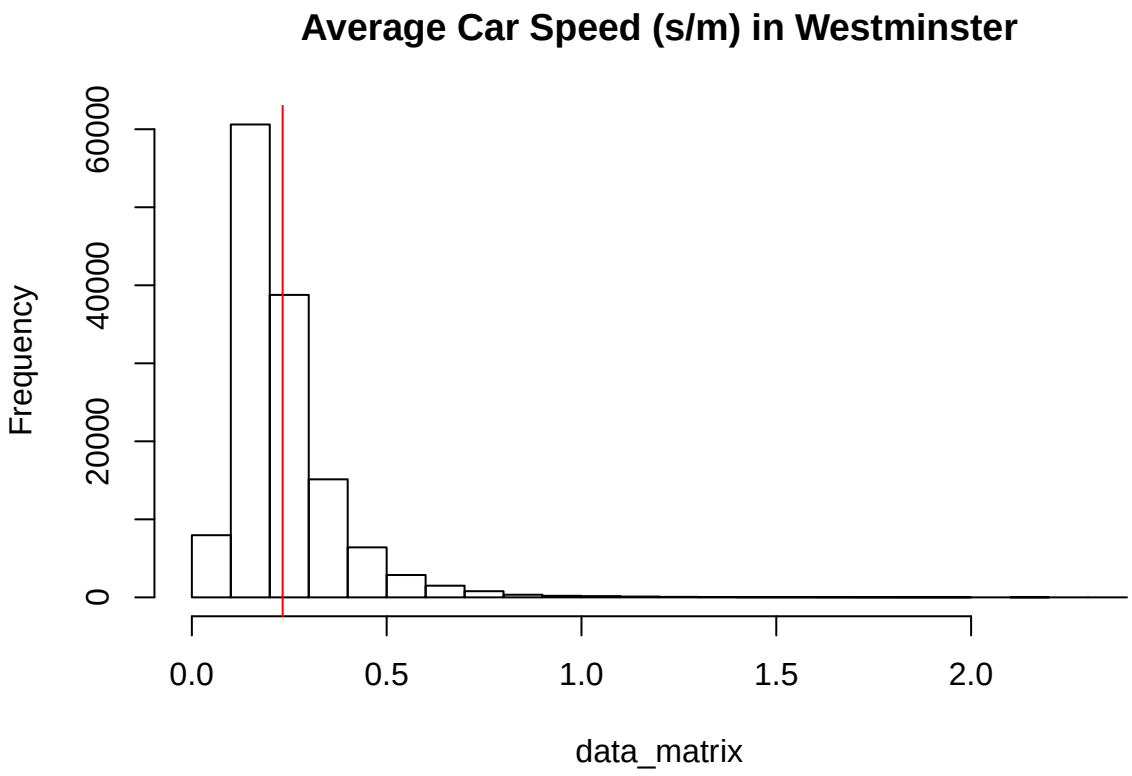


Figure 2: Histogram of car speeds in seconds per metre in Westminster.

A histogram of travel times in Westminster is plotted in Figure 2. Total average travel time: 0.2 sec/m which is 15.4km/h, with a standard deviation of 26.2 across 30 days of data. This shows that there is a huge variation of travel speeds in westminster. The histogram reveals a right-skewed distrbution.

To understand more about the underlying travel time patterns, a time series plot across the 30 days in January is used:

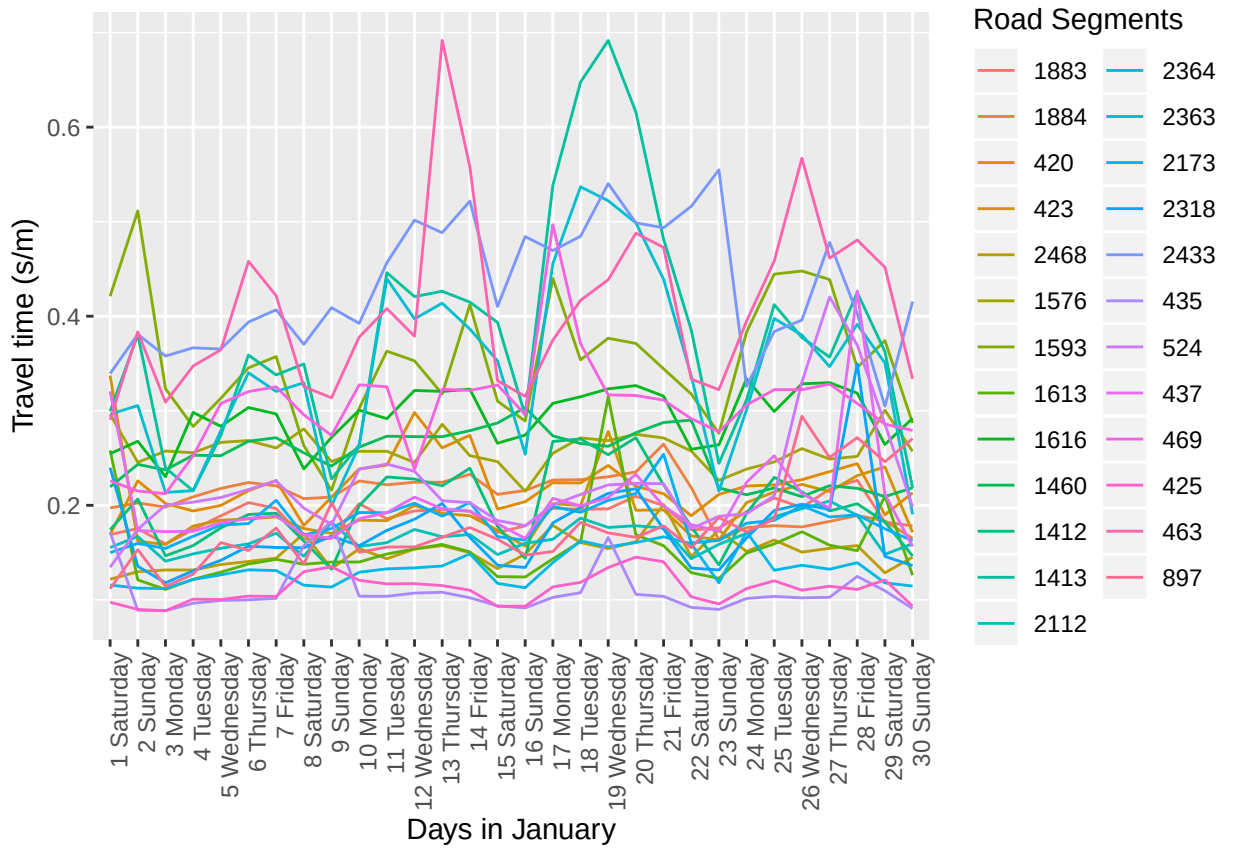


Figure 3: Average daily travel time in s/m across all road segments in Westminster.

The line graph (Figure 3) shows travel time in seconds per metres of 25 road segments in westminster in the month of January 2011. There seems to be 2 observable patterns:

1. Spatio-relationship between travel time and road segment

Magnitude of travel times is strongly correlated to the road segment they are on, where travel times for each road segment is almost always either consistently higher or lower than the ther road segments.

2. Temporal-relationship between travel time and day of the week

Peaks and dips in travel time seem to occur on the same day in all road segments. For example, there are obvious peaks in travel time on all road segments on 2nd January (Sunday) and 19th January (Wednesday).

An aggregated time series plot may be able to show a clearer pattern in temporal variation across the days:

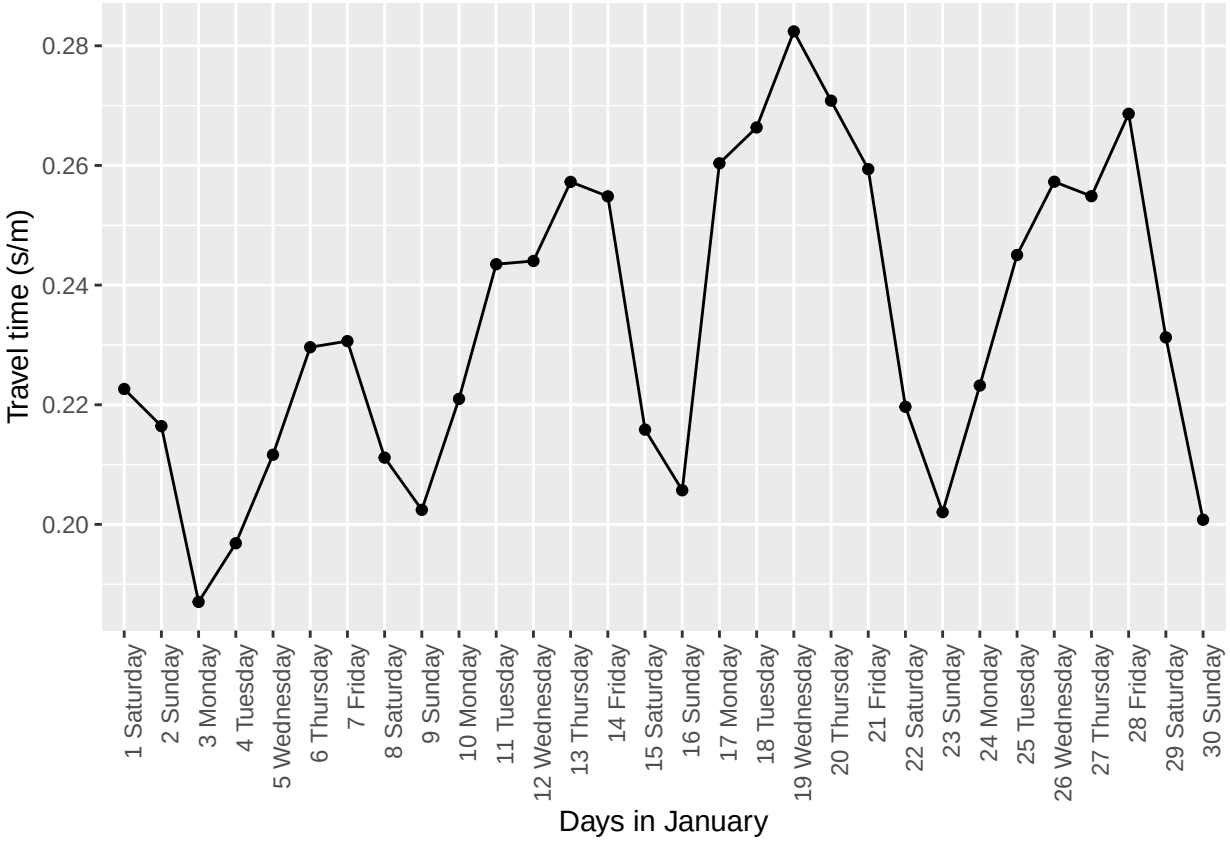


Figure 4: Average daily travel time in s/m across in Westminster.

From Figure 4, it is observed that travel time generally drastically drops from Friday to the Saturday, before reaching an all-time low on Sundays, and increases almost in a linear fashion from Monday to Friday (although there seems to be a small sharp increase in the middle of the week). This aligns to our expectations as fewer cars are on the roads during the weekends as compared to the weekdays where people drive to work.

The only outlier in the data is the travel time on 3rd January (Monday). This is likely due to it being a public holiday (in lieu of New Year's Day). Hence, travel time is expected to behave as if it was a weekend. This data point may have to be removed or be taken into account when building the forecasting model.

A heatmap of travel time across days in January and road segments is produced (Figure 5). The spatial relationship between travel time and road segment is clearly visible. Certain road segments like road segments 463, 2433, 2364, 1413 and 1593 experience consistent high travel times (denoted in yellow stripes).

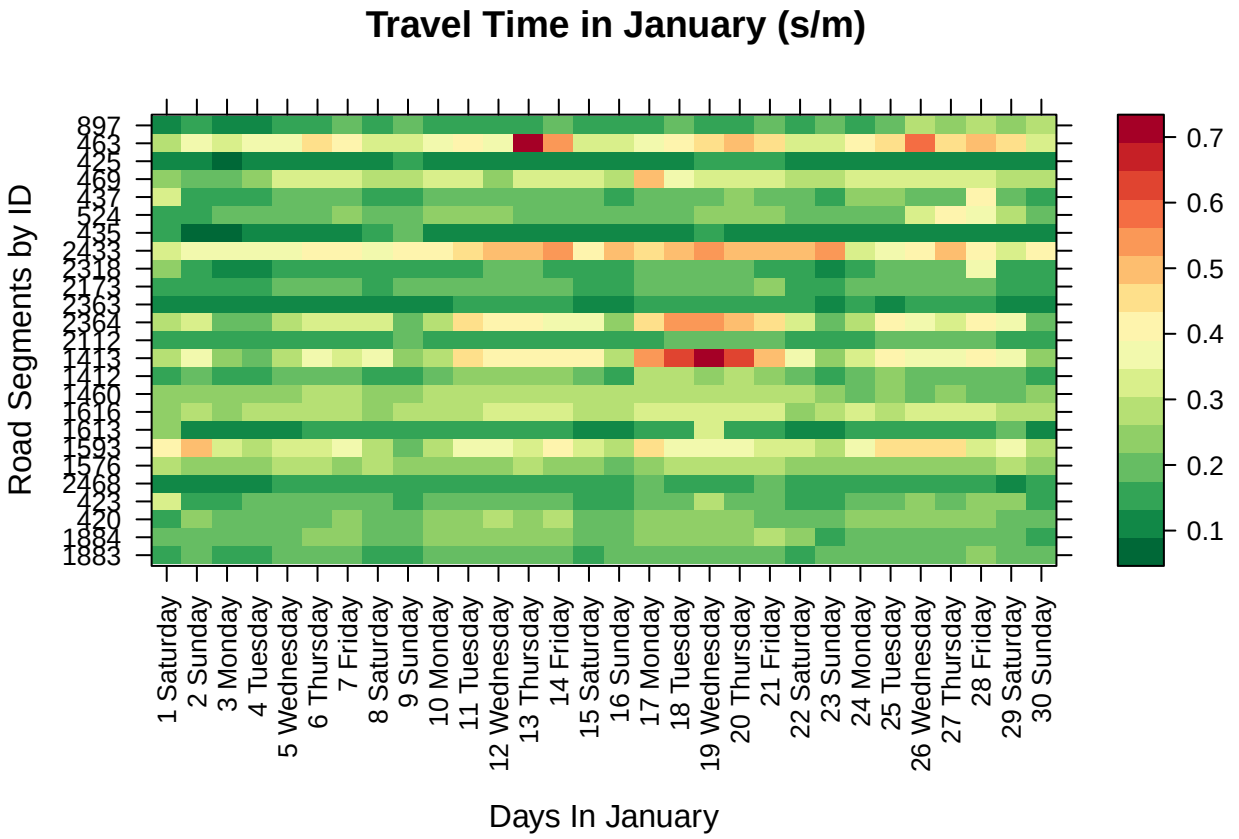


Figure 5: Heat map of average daily travel time in s/m across all road segments in Westminster.

The location of road segments may explain their high travel times (Figure 6). The Waterloo Bridge for instance, in a popular route for South Londoners who wish to travel across the river Thames to get to their workplace in the business district. Road segments in Central London near the St. James area are widely used and experience a heavy vehicle flow, as compared to road segments like road segment 897 that leads vehicles away from central london.

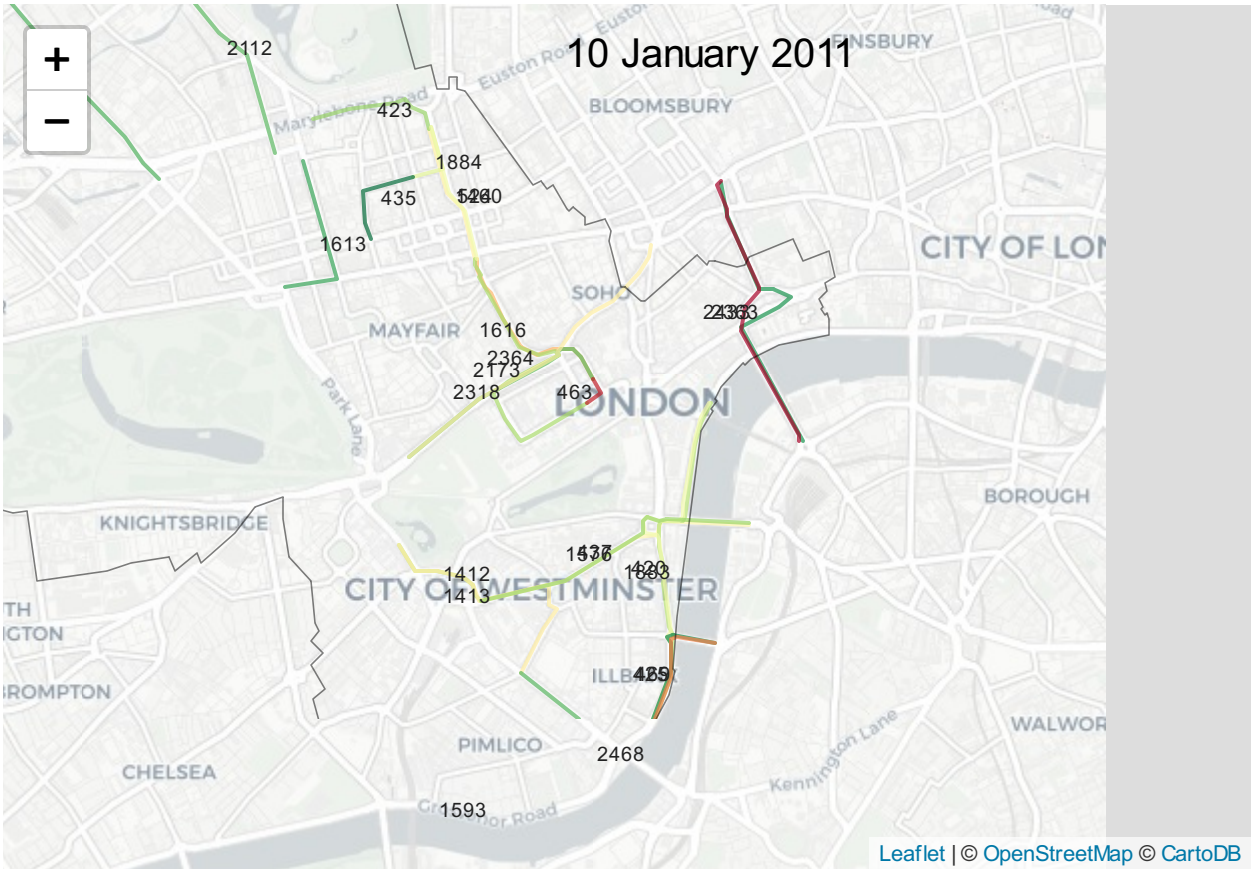


Figure 6: Travel time map in Westminster on 10 January 2011.

To better observe the temporal effects in travel time patterns, travel times across the road segments was standardised to exaggerate these effects. 5 vertical green blocks that represent low travel time during the weekends can be seen in the standardised heatmap (Figure 7). These findings support the earlier observations that travel times are spatially and temporally correlated.

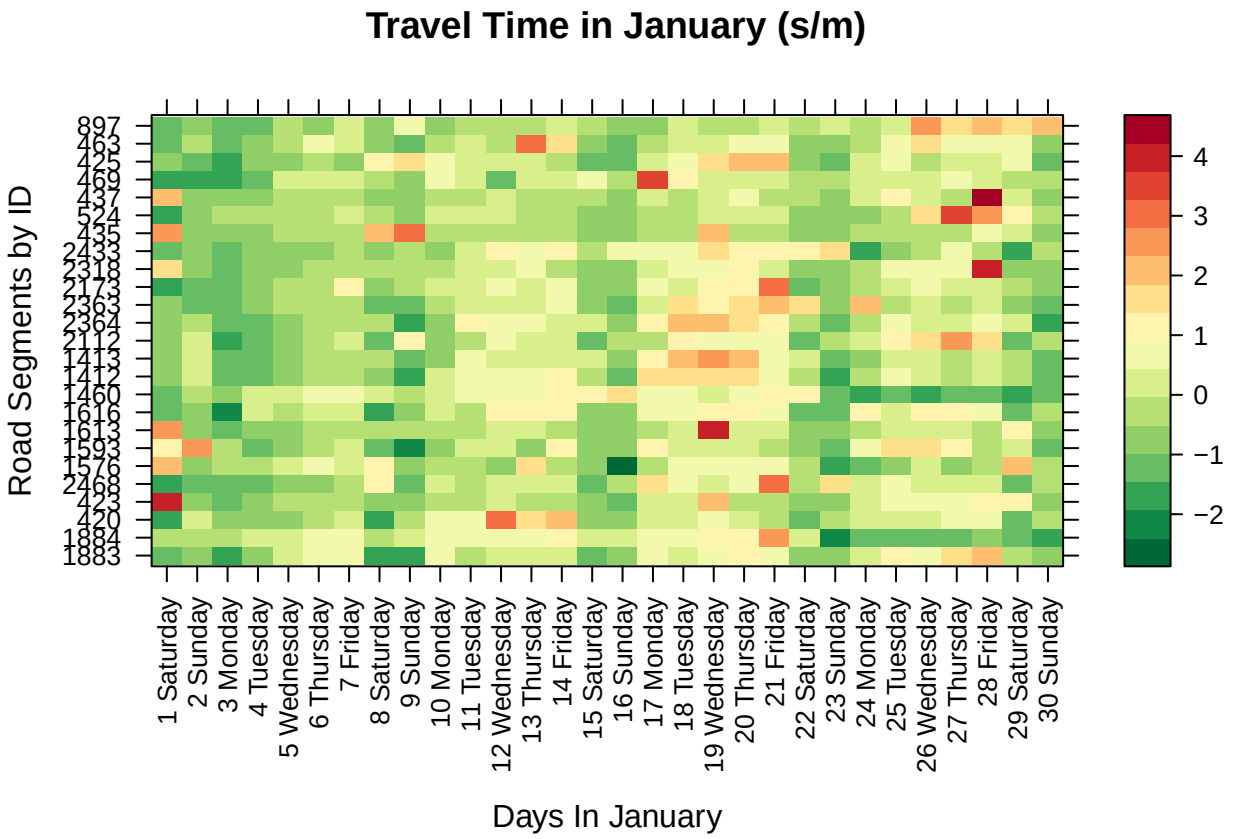


Figure 7: Heat map of standardised average daily travel time in s/m across all road segments in Westminster.

To further observe and break down any trends and seasonal patterns in the travel times, the averaged daily travel times were decomposed. Decomposing the daily travel times in January gives the following plots below (Figure 8). The strong seasonality pattern is further emphasised by the seasonality plot. There seems to be no clear trend however.

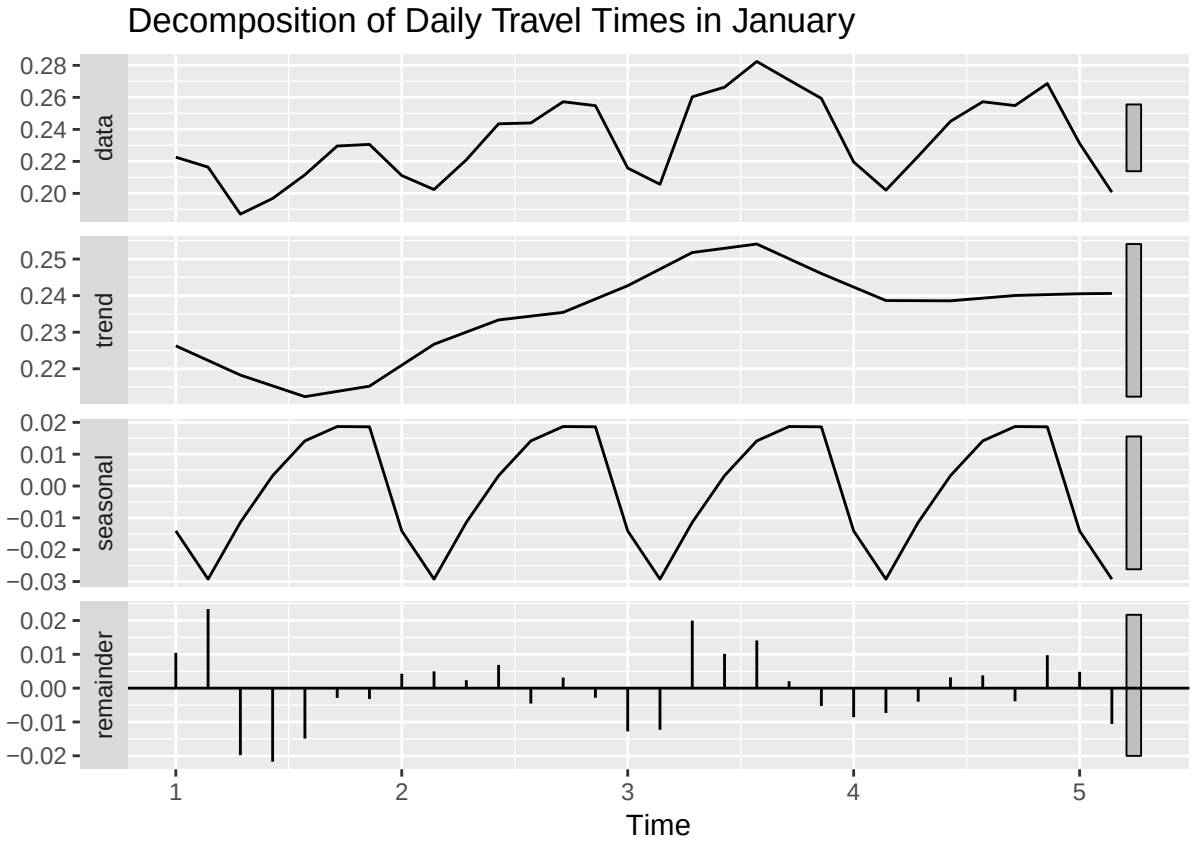


Figure 8: Decomposed plots of daily travel times in January. The topmost plot represents the orignal daily travel time averaged across all road segments in Westminster. The second plot shows the general trend in travel times, the third plot shows the seasonal pattern while the last plot shows the residuals of the decomposition.

3 Accounting for Outliers

It was found that 3rd January 2011 was an outlier as it was a holiday. To ensure unbiased analysis, the average travel time on that day was replaced with the mean travel time of the remaining mondays in January. As such, the original value of 0.1870441 s/m was replaced with the average travel time of the remaining mondays: 0.2348545 s/m.

4 Overall Methodology

There are 2 areas of interest we want to explore:

1. Modelling parts vs Modelling whole
2. Forecasting via statistical models vs Forecasting via machine learning

The following subsections will explain what these mean.

4.1 Modelling Parts vs Modelling Whole

Based on the earlier exploratory analysis, travel time is spatially and temporally correlated. Given that our task is to forecast the last 7 days of travel time, this means that we could model the aggregated daily travel time directly (without regard for its spatial relationships) or we could model the travel time for each road segment separately first then aggregate it. However, we hypothesise that modelling the travel time based on the aggregated data may be more accurate. Modelling individual road segments first may cause the model to overfit overall. This gives our first and second hypotheses:

H1. Modelling aggregated daily travel time by ARIMA is more accurate than modelling travel time by road segments by STARIMA and then aggregating it

H2. Modelling aggregated daily travel time by SVR is more accurate than modelling travel time by road segments by ST-SVR and then aggregating it

4.2 Forecasting via Statistical Models vs Forecasting via Machine Learning Models

Several methods have been used for traffic flow predictions. These can be broadly classified into two main categories: statistical methods and machine learning methods.

1. Forecasting via Statistical Models

Traditionally, statistical methods such as autoregressive integrated moving average (ARIMA) (e.g. Kumar and Vanajakshi 2015) and space-time autoregressive integrated moving average (STARIMA) models (e.g. Kamarianakis and Prastacos 2005) have been applied to model travel flows over time and space. These time-series models take into account the temporal sequence of a dataset. However, these methods usually require stationarity of the data.

2. Forecasting via Machine Learning Models

Machine learning methods such as support vector regression (SVR) (e.g. Hong et al. 2011) have also been used for urban traffic flow forecasting. Machine learning methods can be used for time-series modelling but they usually assume that the variables are independent. However, time-series data would be required to be re-framed into a supervised learning format, in order to feed them as inputs into the machine learning models.

Although both approaches have been successful, Hong et al. (2011) argue that statistical methods like ARIMA, unlike machine learning methods like SVR, face difficulty in capturing rapid variational changes in traffic flow processes. However, given that there is only 30 days' worth of data, we hypothesise that the low dimensionality and volume of the dataset may make it difficult for machine learning models to generalise well. However, Kumar and Vanajakshi (2015) report that techniques such as ARIMA are also one of the

most precise methods for forecasting traffic flows. As such, we will be employing ARIMA/STARIMA for the classical time-series modelling and SVR/ST-SVR for the machine learning model to examine which of the 2 methods would perform better with our 30 days worth of dataset.

To this end, this gives us our third and fourth hypotheses:

H3. Forecasting time-series with ARIMA gives better accuracy than with SVR

H4. Forecasting time-series with STARIMA gives better accuracy than with ST-SVR

5 Experiments

To test out both our hypotheses, we shall be using 4 modelling approaches:

1. Model aggregated daily travel time with ARIMA (Jia Wen)
2. Model daily travel time data on each of the road segments with STARIMA *then* aggregate the results (Junju)
3. Model aggregated daily travel time with SVR (Ju Yong)
4. Model daily travel time data on each of the road segments with ST-SVR *then* aggregate the results (Xulan)

5.1 Model Aggregated Daily Travel Time with ARIMA: Ng Jia Wen

ARIMA stands for auto-regressive integrated moving average, a set of statistical models that use past values of a time series to forecast future values of the series (Haworth 2018). It comprises of three components:

1. AR: The autoregressive component

$$\hat{y_t} = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t$$

2. I: The integration component

This component refers to the differencing procedure where an order of 1 would mean differencing the series by lag 1.

3. MA: The moving average component

This component uses past forecast errors to forecast future values of the series. A general MA model can be defined as:

$$\hat{y_t} = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q}$$

An ARIMA model can be non-seasonal or seasonal in which case their notations will be as defined as follows: ARIMA(p,d,q) for non-seasonal and ARIMA(pdq)(PDQ)m for seasonal where p/P is the AR order, d/D is the order of differencing, q/Q is the MA order and m is the lag of the seasonal component.

Data is aggregated to take the average of all travel time for all road segments for each day in January.

5.1.1 Stationarity

As ARIMA requires the time series to be stationary, we shall test for both trend and difference stationarity with Kwiatkowski-Phillips-Schmidt-Shin (KPSS) and Augmented Dickey-Fuller (ADF) test respectively. If both tests conclude that the series is stationary (non-stationary) then the series is stationary (non-stationary). If KPSS concludes stationary and the ADF concludes non-stationary then the series is trend stationary and the series can be detrended to make it stationary. If KPSS concludes non-stationary and ADF concludes stationary, then the series is difference stationary and the series can be differenced to make it stationary.

The null hypothesis of the KPSS test is that the data is trend stationary and since the test statistic of 0.2882 is smaller than the critical value at the 95% confidence interval – 0.463, the null hypothesis is not rejected, so the series is trend-stationary (see Appendix section 8.1).

The null hypothesis of the ADF test is that the data is non-stationary (see below) and since the p-value is 0.2055 which is bigger than 0.05 at the 95% confidence interval, the null hypothesis is not rejected – the series is non-stationary (see Appendix section 8.2). This indicates the presence of a unit root.

The implication of these results is that there may be an underlying trend so the series needs to be detrended.

Plotting the ACF and PACF plots gives the following results:

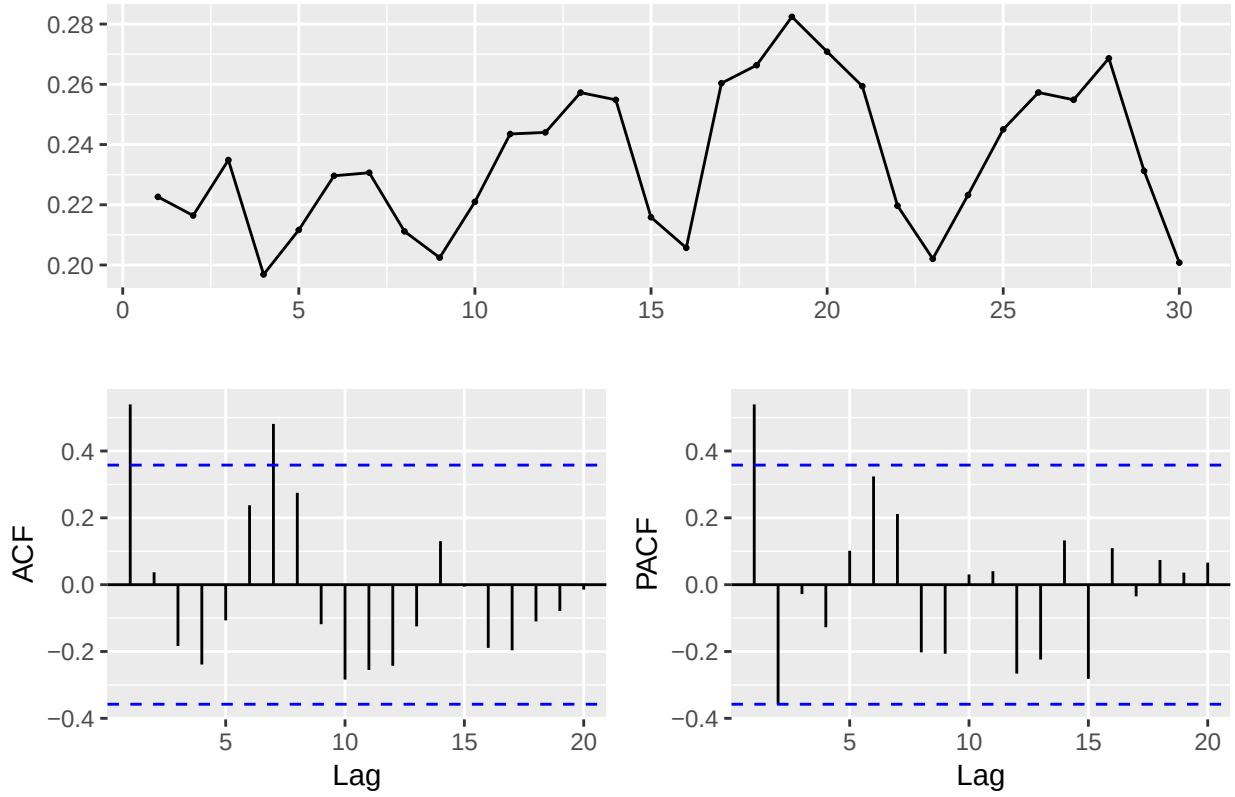


Figure 9: Time series, ACF and PACF plots for travel time data

The ACF plot shows significant lags at lags 1 and 7 (see Figure 9). A lag of 7 indicates a weekly seasonality, which is what we expect (travel time on mondays are likely to be similar to each other, same for tuesdays, wednesdays and so on). A lag of 1 implies that travel time today is correlated to travel time yesterday. (travel time on wednesday is more likely to be similar to travel time on tuesday than on monday).

This means that we can attempt to stationarise the data by differencing at lag 1, lag 7 or both. This yields 4 possible scenarios:

1. No differencing (benchmark)
2. Do a non-seasonal difference with lag of 1
3. Do a seasonal difference with lag of 7
4. Do both differencing above

Results indicate that differencing at lag 1 does not help to stationarise data (see Appendix section 8.3).

Results indicate that differencing at lag 7 stationarises the series as the null hypothesis of the KPSS test that the series is trend-stationary is not rejected and the null hypothesis of the ADF test that the series is not difference-stationary is rejected (see Appendix section 8.4).

Results indicate that differencing at both lag 1 and 7 does not help to stationarise data (see Appendix section 8.5).

Following the analyses from above, we can make several hypothesis:

1. Models that have been differenced by lag 7 will perform better than models that have been differenced by lag 1.
2. Models that have been differenced by lag 7 will perform better than models that have not been differenced.
3. Models that takes into account travel time of the previous day and seasonality, will offer the best predictions.

In the following section, we will build models for all scenarios to explore the validity of our hypotheses.

5.1.2 Finding optimal parameters

Optimal paramters will be determined by looking at ACF and PACF plots.

1. No Differencing (benchmark)

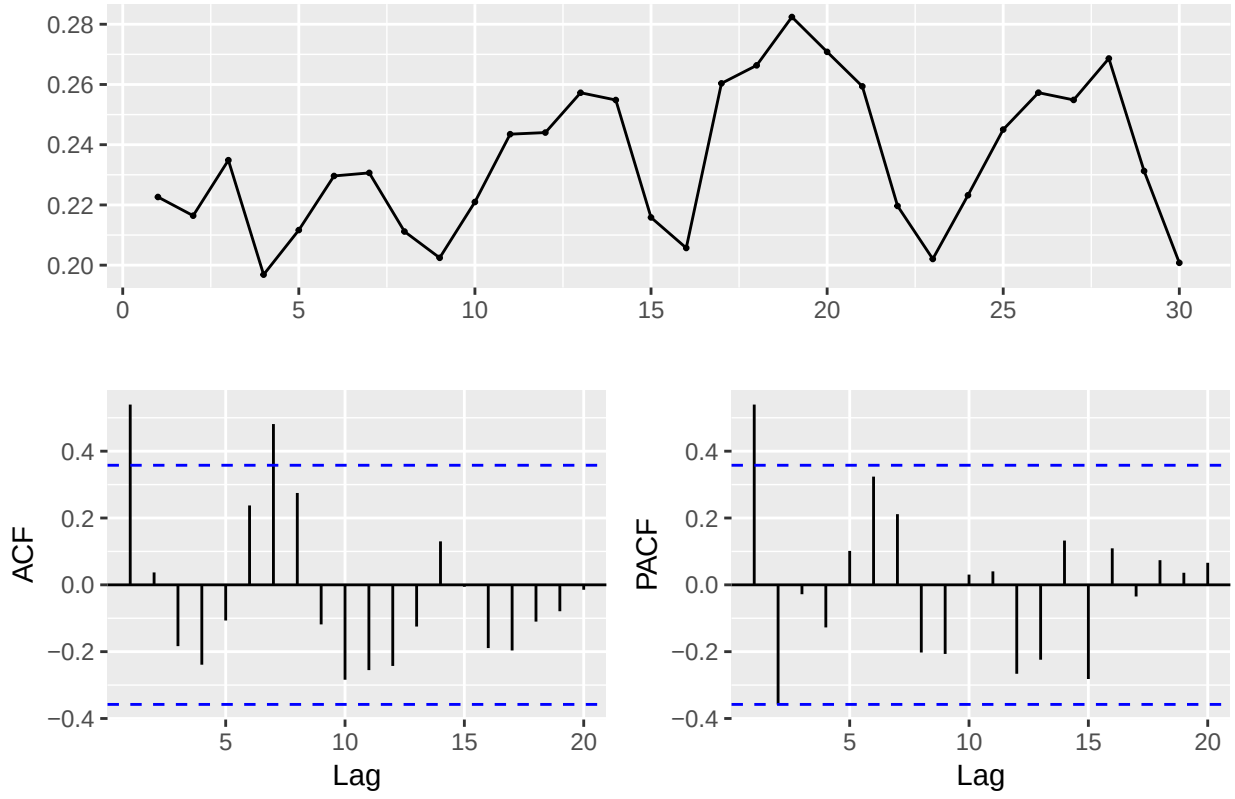


Figure 10: Time series, ACF and PACF plots of undifferenced time series

The ACF graph (see Figure 10) is described to have one or more spikes hence it is a MA model with order 2 (2 significant lags in ACF). This suggests a MA(2) model.

2. Non-seasonal difference with lag of 1

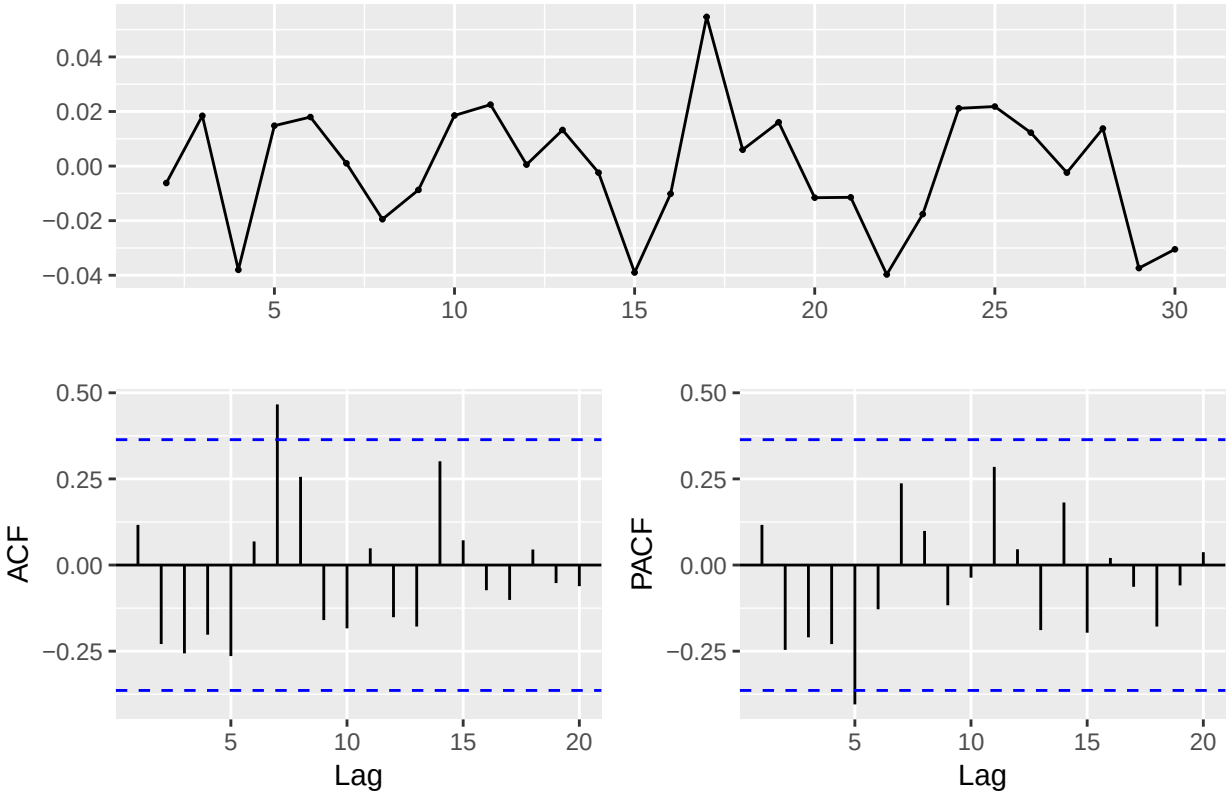


Figure 11: Time series, ACF and PACF plots of time series differenced at lag 1

The ACF graph (see Figure 11) is described to have one or more spikes hence it is a MA model with order 1 (1 significant lag in ACF). This suggests a MA(1) model with difference at lag 1.

3. Seasonal difference with lag of 7

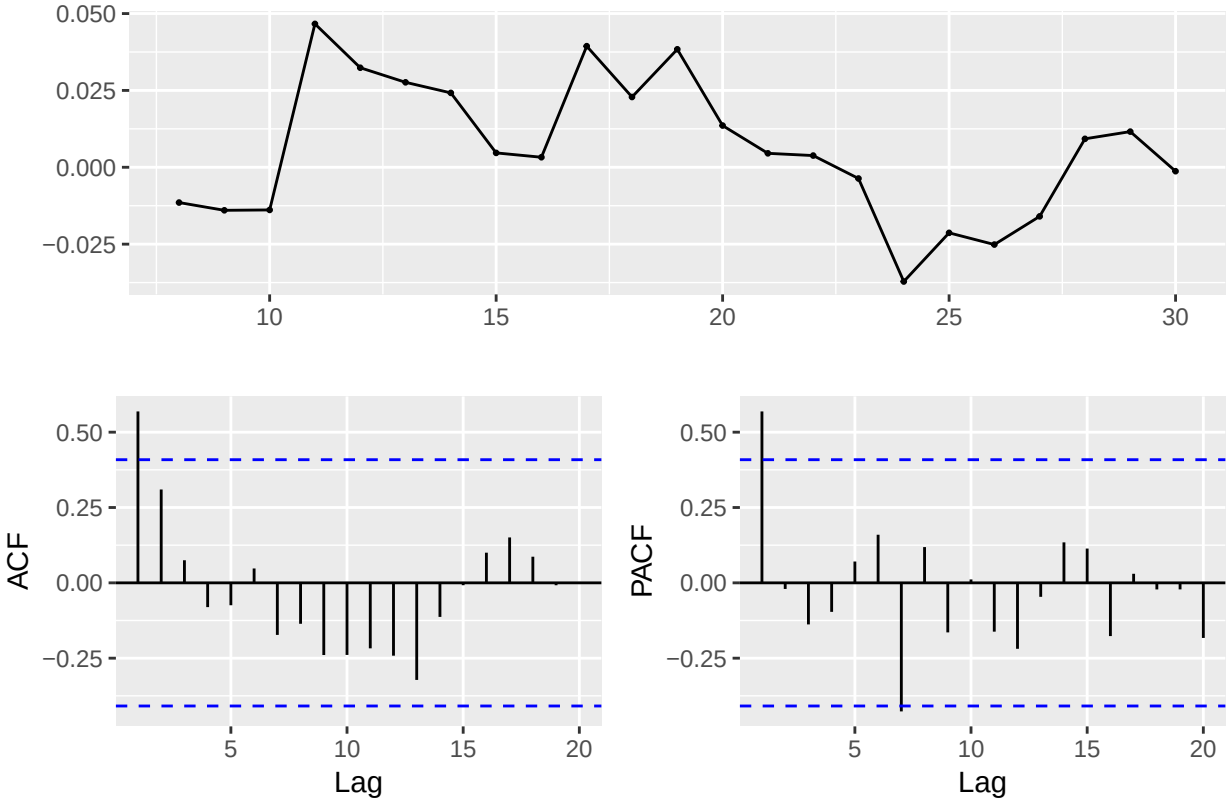


Figure 12: Time series, ACF and PACF plots of time series differenced at lag 7

The ACF graph (see Figure 12) is described to have one or more spikes hence it is a MA seasonal model with order 1 (1 significant lag in ACF). This suggests a MA(1) model with difference at lag 7.

4. Do both differencing above

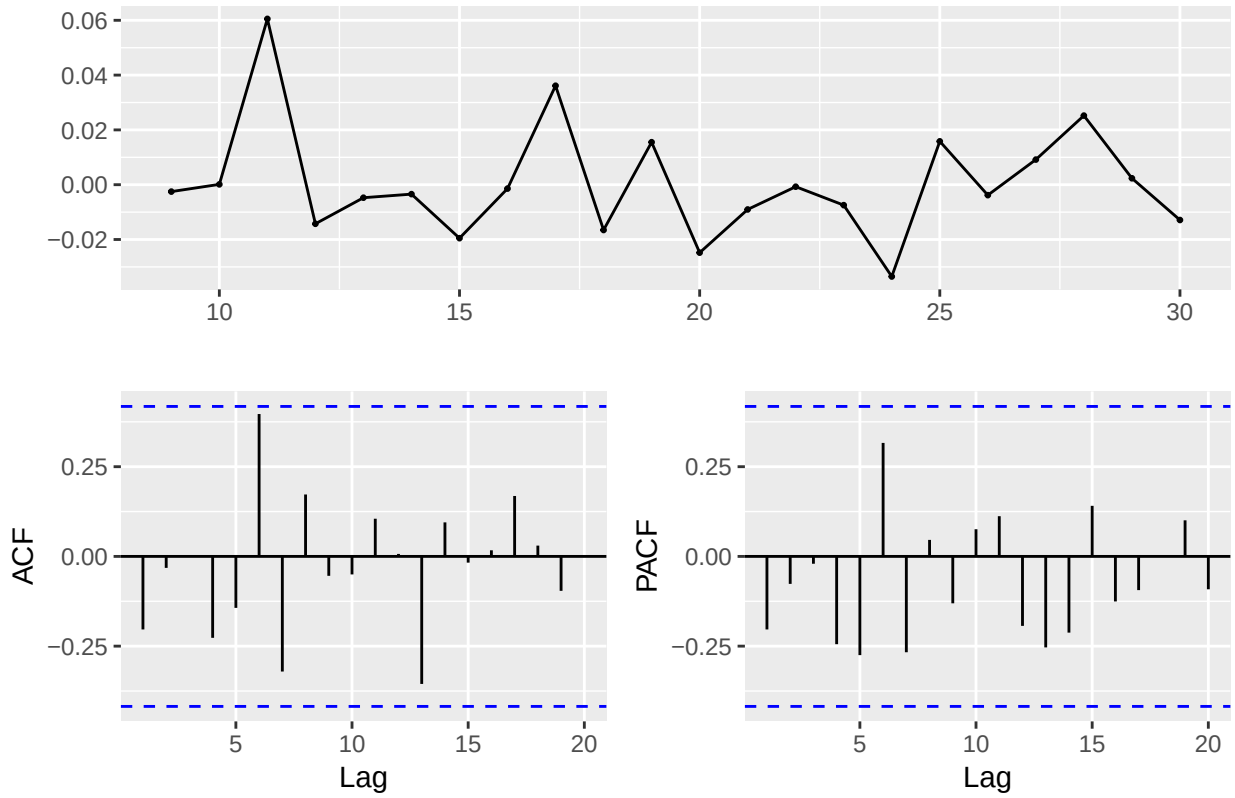


Figure 13: Time series, ACF and PACF plots of time series differenced at lag 1 and 7

Differencing with respect to the 1st and 7th lag yields a non-significant ACF and PACF plot (see Figure 13) – the data is now essentially random. Hence, this means that the data has been overdifferenced.

While the ACF plots suggest MA models to be used, it makes sense to consider autoregressive terms too, based on our initial hypothesis that the best model would account for both the effects of the previous day, and seasonality. Hence, we can also consider mixed AR and MA models, with p = 1.

This leads us to the following candidate models:

A1) No Differencing (MA model): ARIMA(0,0,2)

A2) No Differencing (mixed MA, AR model): ARIMA(1,0,2)

B1) Non-seasonal difference with lag of 1 (MA model): ARIMA(0,1,1)

B2) Non-seasonal difference with lag of 1(mixed MA, AR model): ARIMA(1,1,1)

C1) Seasonal difference with lag of 7 (MA model): ARIMA(0,0,0)(0,1,1)7

C2) Seasonal difference with lag of 7 (mixed MA, AR model): ARIMA(1,0,0)(0,1,1)7

5.1.3 Results

Diagnostic checking was conducted for each of the 6 candidate models and results show that in all cases, the residuals were normally distributed and uncorrelated (see Appendix section 8.6).

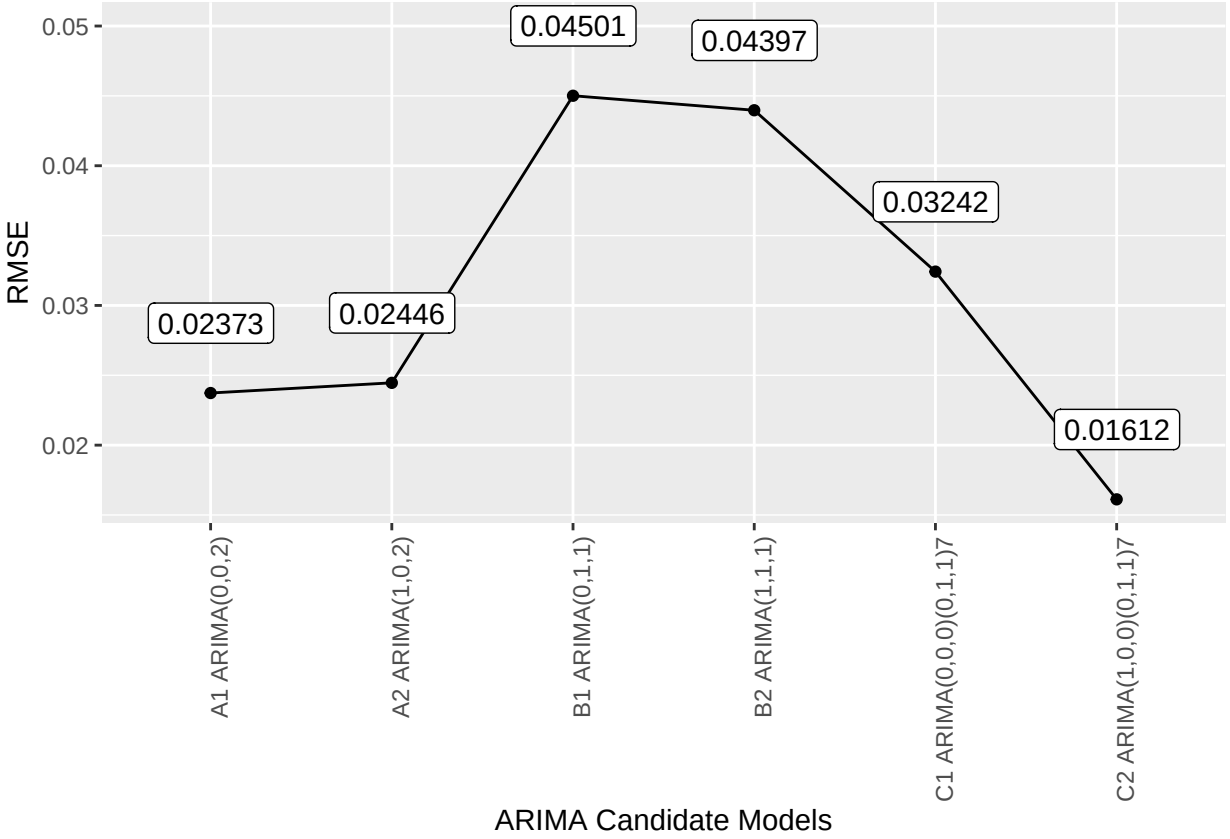


Figure 14: RMSE scores of all 6 candidate ARIMA models

Results indicate that the model C2: ARIMA(1,0,0)(0,1,1)7 yields the lowest Root Mean Squared Error (RMSE) (see Figure 14). This validates our third hypothesis: Models that takes into account travel time of the previous day and seasonality, offers the best predictions.

However, if we do not account for an autoregressive term (comparing models A1, B1 and C1), model A1: ARIMA(0,0,2) performs the best. This is surprising because we would expect model C1 to outperform model A1 as model C1 accounts for seasonality. Perhaps, with only 23 days worth of data to fit the model on, seasonality effects are not that significant. This denies our second hypothesis that models that have been differenced by lag 7 will perform better than models that have not been differenced.

Another interesting observation that we can see is that the RMSE drops significantly in models B1 to B2 and from C1 to C2. This however, is not observed from model A1 to A2. This means that the effects of including an autoregressive term in a model is more significant for a series that has been differenced than a series that has not been differenced. Perhaps, differencing a series helps to remove underlying seasonality and trends which helps the model to pick out autoregressive effects more easily.

Lastly, we observe that models that have been differenced by lag 7 performed better in both cases, as compared to models that have been differenced by lag 1 (C1 > B1 & C2 > B2). This validates our first

hypothesis that models that have been differenced by lag 7 will perform better than models that have been differenced by lag 1.

Predicting the last 7 days in the month of January with the best model, model C2, produces the following graph below (see Figure 15):

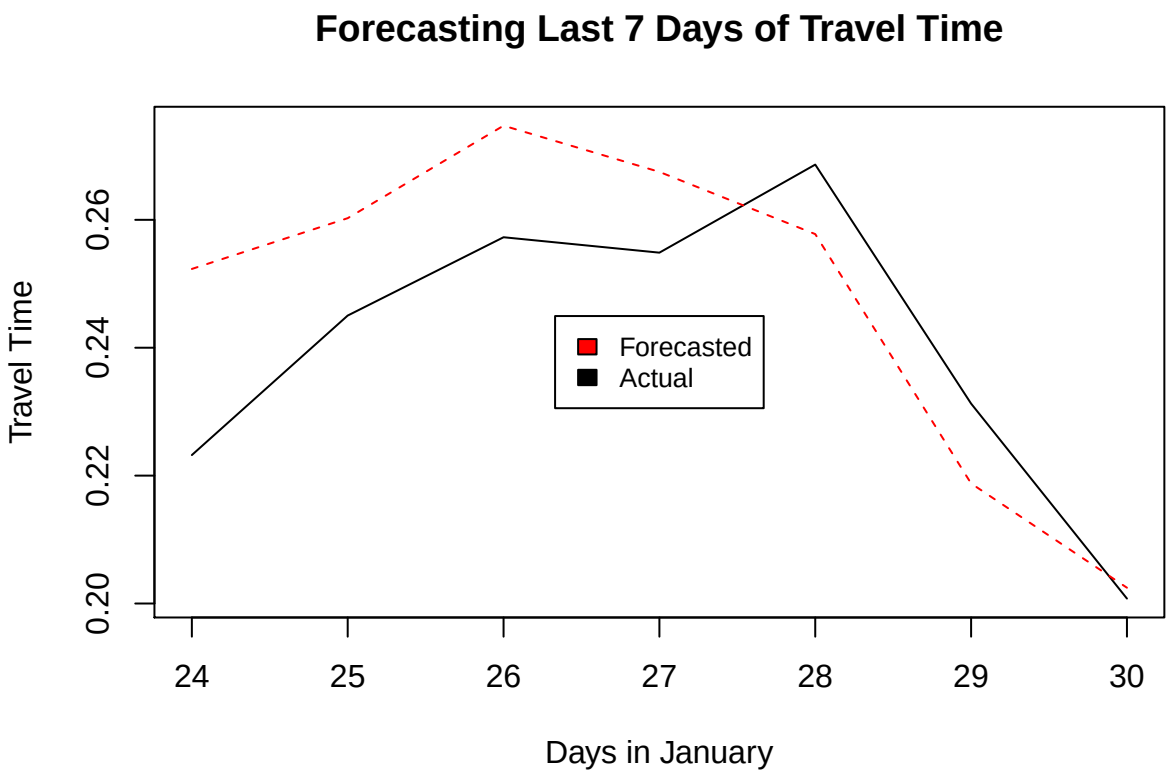


Figure 15: Forecasted time series with model C2 against actual travel time

5.2 Model daily travel time data on each of the road segments with STARIMA: Junju Ng

Note: There seems to be a labelling error in the STARIMA package. Specifically, the ACF lags are labelled m+1, where m is the lag in question. For instance, lag 1 on the ACF graph is in reality lag 0. This is also evidenced by how the ACF at lag = 1 is 1.0. Thus, all lag labels on the ACF plots need to be subtracted by 1.

5.2.1 STARIMA

A space-time autoregressive integrated moving average (STARIMA) model was implemented to forecast travel times for each individual road segment, taking into account first-order spatial characteristics. It presents each observation at time t and location i as a weighted combination of previous observations lagged across space and time (Kamarianakis and Prastacos 2005). The first-order spatial topological relationships across space are represented by means of an N x N spatial weight matrix, where N is the number of road segments studied.

The daily travel time on each road segment is **then** aggregated to give a modelled aggregated daily travel time.

Similar to the ARIMA model, the STARIMA model has three parameters: 1) autoregressive component, 2) integration component, 3) moving average component. These components are elaborated in the earlier section on ARIMA.

For a STARIMA model with no seasonal component, the vector of observations at time t at N locations is represented using the equation:

$Z_t = \sum_{k=1}^{p} \sum_{l=0}^{\lambda_k} \phi_{kl} W_l Z_{t-k} - \sum_{k=1}^{q} \sum_{l=0}^{m_k} \theta_{kl} W_l a_{t-k} + a_t$

where p is the autoregressive order, q is the moving average order, λ_k is the spatial order of the k^{th} autoregressive term, m_k is the spatial order of the k^{th} moving average term, ϕ_{kl} and θ_{kl} are parameters to be estimated, W_l is the $N \times N$ matrix for spatial order l, and a_t is the random normally distributed disturbance vector at time t (Kamarianakis and Prastacos 2005).

For a STARIMA model with a seasonal component, this is represented using the equation:

$\Phi_{P,\Lambda}(B^S)\Phi_{p,\lambda}(B) \bigtriangledown_S^D \bigtriangledown^d Z_t = \Theta_{Q,M}(B^S)\theta_{q,m}(B)a_t$

where

$\Phi_{P,\Lambda}(B^S) = I - \sum_{k=1}^{p} \sum_{l=0}^{\Lambda_k} \Phi_{kl} W_l B^{kS}$

$\phi_{P,\lambda}(B) = I - \sum_{k=1}^{p} \sum_{l=0}^{\lambda_k} \phi_{kl} W_l B^{k}$

$\Theta_{Q,M}(B^S) = I - \sum_{k=1}^{p} \sum_{l=0}^{\lambda_k} \Theta_{k,l} W_l B^{kS}$

$\theta_{q,m}(B) = I - \sum_{k=1}^{q} \sum_{l=0}^{m_k} \theta_{kl} W_l B^{k}$

where Φ_{kl} and ϕ_{kl} are the seasonal and nonseasonal autoregressive parameters at temporal lag k and spatial lag l, Θ_{kl} and θ_{kl} are the seasonal and nonseasonal moving average parameters at lags k and l, P and p are the seasonal and nonseasonal autoregressive orders, Q and q are the seasonal and nonseasonal moving average orders, Λ_k and λ_k are the seasonal and nonseasonal spatial orders for the k^{th} autoregressive term, M_k and m_k are the seasonal and nonseasonal spatial orders for the moving average term. In addition, D and d are the number of seasonal and nonseasonal differences required, where $\bigtriangledown_S^D$ and $\bigtriangledown^d$ are the seasonal and nonseasonal difference operators, such that i.e., $\bigtriangledown_S^D = (I - B^S)^D$ and $\bigtriangledown^d = (I - B)^d$ with seasonal lag S. Lastly, a_t is the random normally distributed error vector at time t (Kamarianakis and Prastacos 2005).

5.2.2 Space-time PACF and ACF Analysis

Space-time PACF and ACF analysis is conducted to identify model parameters and determine if the series is stationary.

5.2.2.1 Space-time ACF Analysis

A space-time autocorrelation function plot was plotted to check for stationarity (Figure 16). As most lags are insignificant, the series is likely to be stationary. This is corroborated with the KPSS test (test-statistic $= 0.1424 < 0.463$ (critical value for 95% confidence interval)). Thus, no additional differencing is needed to make the series stationary and a model based on the undifferenced series can be built.

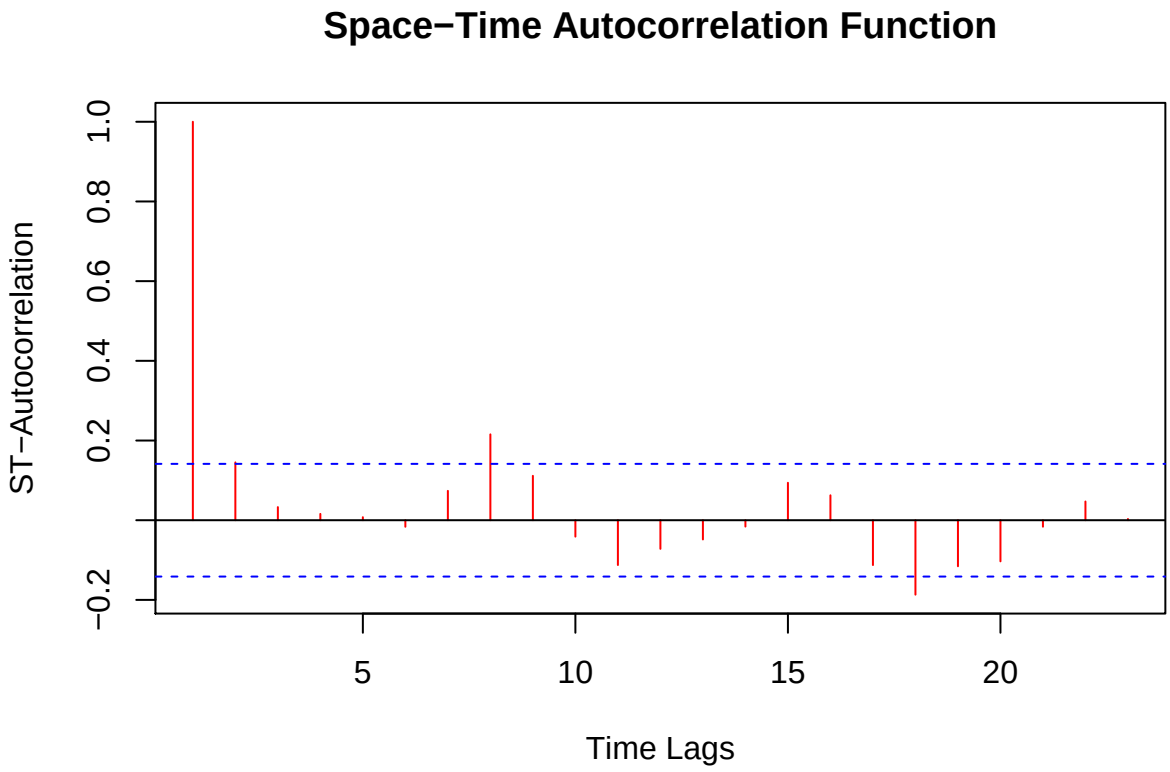


Figure 16: STACF plot of daily travel times across road segments in Westminster. The dashed lines approximate 95% confidence interval for the autocorrelation.

However, the undifferenced STACF plot depicts a spike at lag 8 (in reality lag 7), suggesting that travel times may be correlated with the day of the week. Thus, an additional model with weekly seasonal differencing will also be built. Consequently, an STACF plot with seasonal differencing at lag = 8 (Figure 17) is also plotted to estimate model parameters for the weekly differenced model.

The seasonally differenced STACF plot has only one significant lag at lag = 1, indicating that seasonal differencing can remove the potential weekly pattern. As the ACF plot only has a single spike, an MA(1) model is estimated. As most lags are insignificant, the series is likely stationary. This is corroborated with the KPSS test (test-statistic $= 0.075 < 0.463$ (critical value for 95% confidence interval)). Thus, no additional differencing is required.

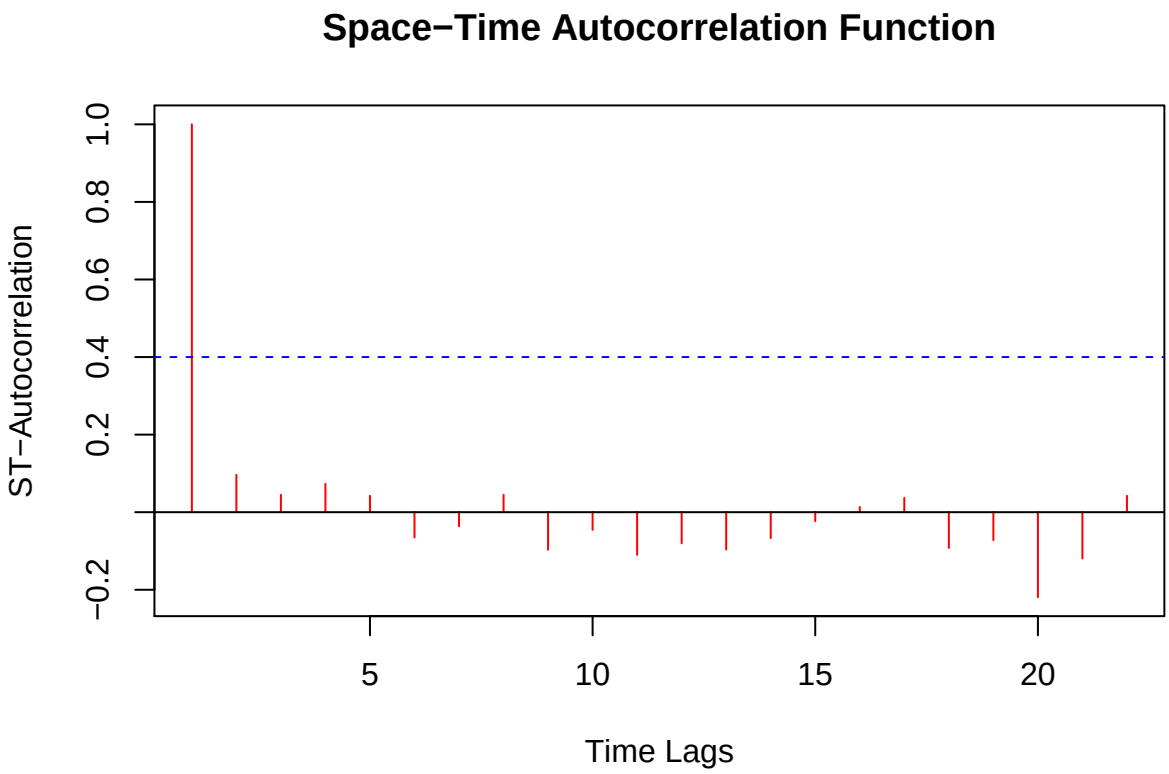


Figure 17: STACF plot of daily travel times across road segments in Westminster after weekly seasonal differencing. The dashed lines approximate 95% confidence interval for the autocorrelation.

5.2.2.2 Space-time PACF analysis

PACF plots are then plotted to determine the AR orders for undifferenced and weekly differenced travel time data for input into the STARIMA model. None of the lags of the undifferenced and differenced STPACF plots are significant (Figure 18 and 19), thus the autoregressive order for the undifferenced and weekly differenced models is 0, giving rise to pure MA models.

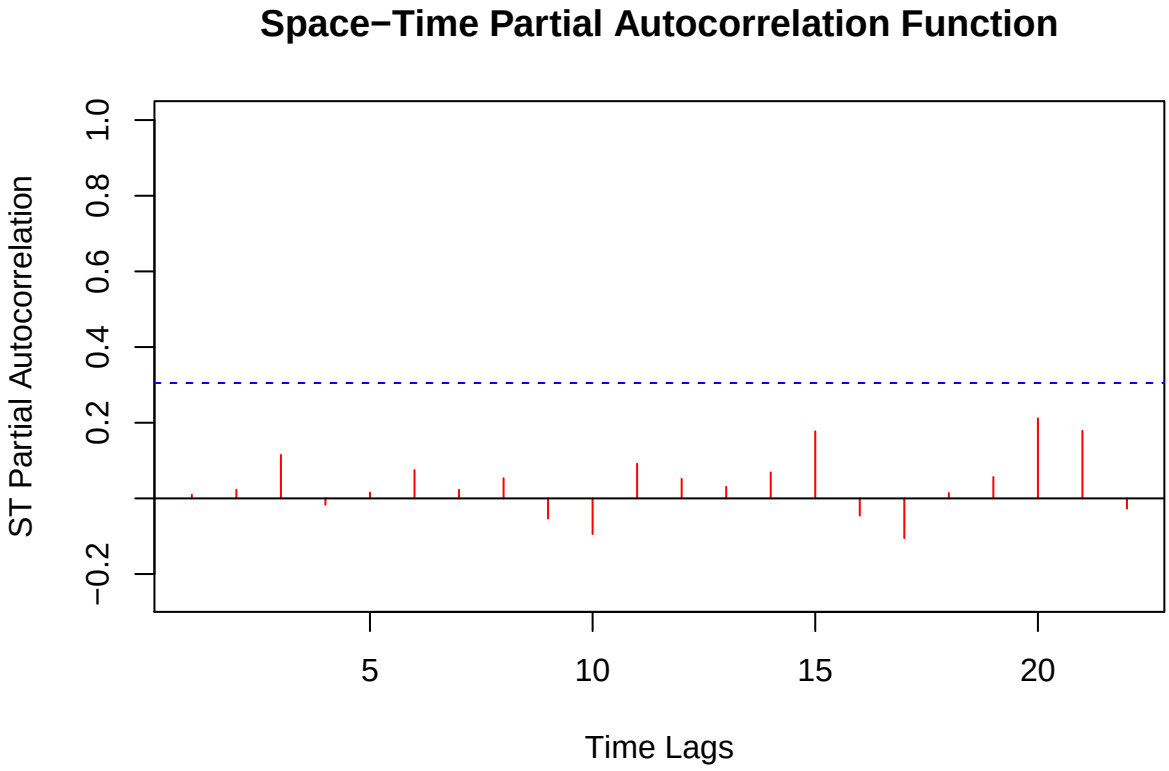


Figure 18: STPACF plot of daily averaged travel times in Westminster without differencing. The dashed lines approximate 95% confidence interval for the autocorrelation.

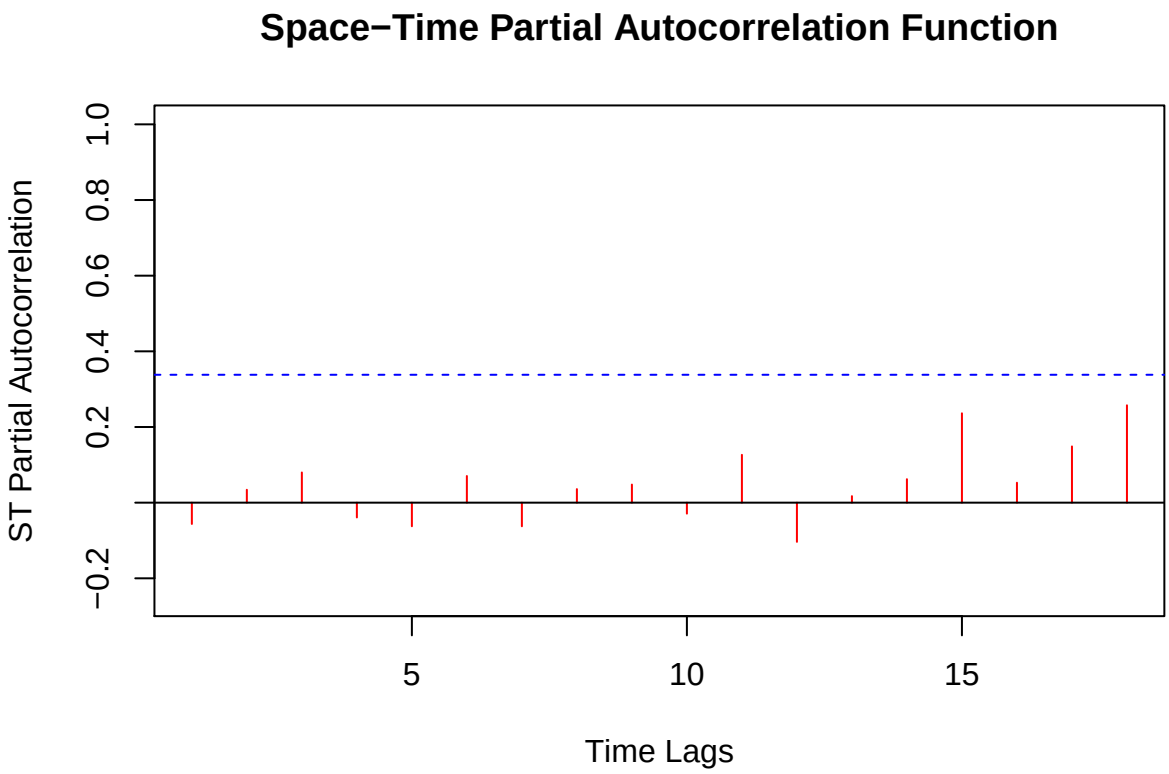


Figure 19: STPACF plot of daily averaged travel times in Westminster after seasonal differencing weekly. The dashed lines approximate 95% confidence interval for the autocorrelation.

5.2.3 Model Parameters

Based on the ST-ACF and ST-PACF analyses, four main models are tested. The details of the models are given below.

D1) No Differencing (MA model): STARIMA(0,0,2) – in reality a STARIMA(0,0,1) model due to the labelling error

This model uses the forecast errors of previous day's travel time to forecast the next day's travel time.

D2) No Differencing (mixed MA, AR model): STARIMA(1,0,2) – in reality a STARIMA(1,0,1) model due to the labelling error

This model uses the previous day's travel time and its forecast errors to forecast the next day's travel time.

E1) Seasonal differencing with lag of 7 (MA model): STARIMA(0,0,1)(0,1,1)8 – in reality a STARIMA(0,0,1)(0,1,1)7 model due to the labelling error

The forecast errors of previous week's travel time on a given day of the week is used to forecast travel times on the same day of the following week.

E2) Seasonal differencing with lag of 7 (mixed MA, AR model): STARIMA(1,0,1)(0,1,1)8 – in reality a STARIMA(1,0,1)(0,1,1)7 model due to the labelling error

The previous week's travel time on a given day of the week and its forecast errors is used to forecast travel times on the same day of the following week.

5.2.4 Model Testing

To create the model, travel time data for the first 23 days in January (i.e. 1 January to 23 January) was used to fit the model, and estimate model parameters. Thereafter, model parameters are used to predict travel times for the last 7 days in January (i.e. 24 January to 31 January).

STACF plots of the residuals of the fitted models were then plotted to determine if the residuals are random.

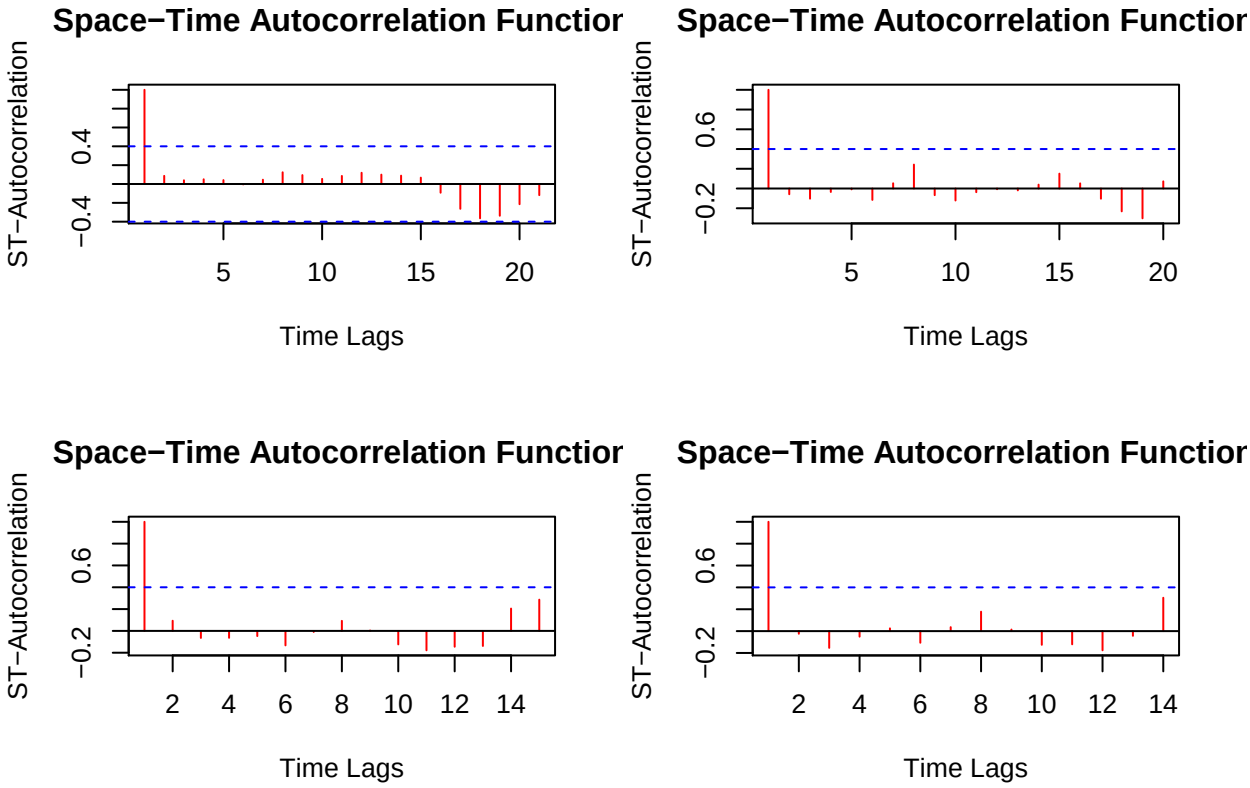


Figure 20: STACF plot for residuals of: D1 (top left), D2 (top right), E1 (bottom left), E2 (bottom right)

As all of the STACF plots (Figure 20) cut off after one time lag, the residuals of all these models are random. Thus, forecasting of the travel times for the last seven days using the various model parameters can be carried out.

5.2.5 Model Results

This section presents results of the 4 different STARIMA models.

5.2.5.1 NRMSE

The NRSMEs for each road segment for each model is calculated and presented in Figure 21. In general, NRMSEs are larger for the undifferenced models (e.g. D1 NRMSE ranges from 1.23-23.5, D2 NRMSE ranges from 0.822-1.97) compared to weekly differenced models (e.g. E1 NRMSE ranges from 0.756-13.8, E2 NRMSE ranges from 0.802-7.90).

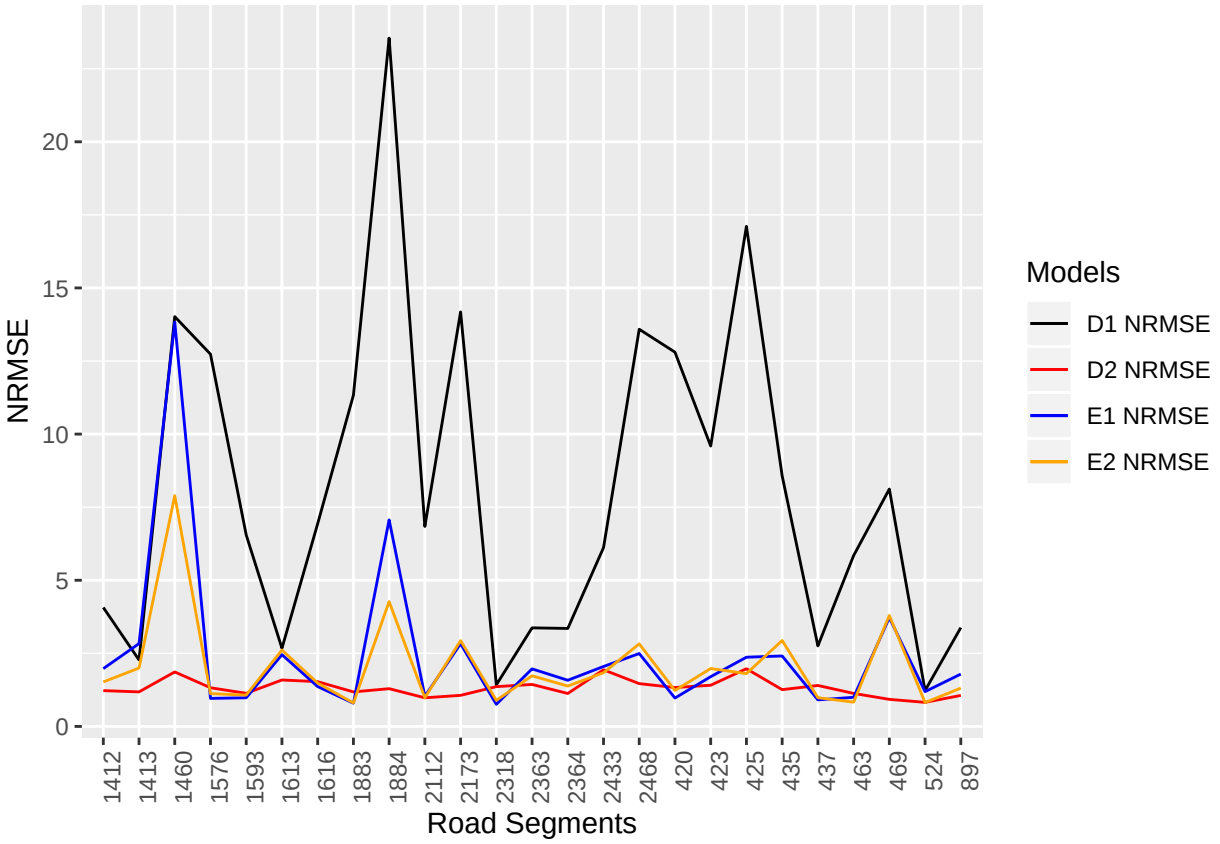


Figure 21: NRMSEs of daily travel times for each road segment for all 4 STARIMA models

Road segment 1460 has anomalously large NRMSEs (14.0, 1.86, 13.8, 7.90 for D1, D2, E1, E2 respectively). Further investigation suggests that the large error seen in the data for road segment 1460 is because travel times in the last week of January are anomalously shorter compared to the rest of the month (Figure 22). As such, the STARIMA model is not accurate when the travel times deviate from the long-term trend. Although it was possible to exclude road segment 1460 from the prediction, a decision was made to include it in the model, for comparison with other approaches.

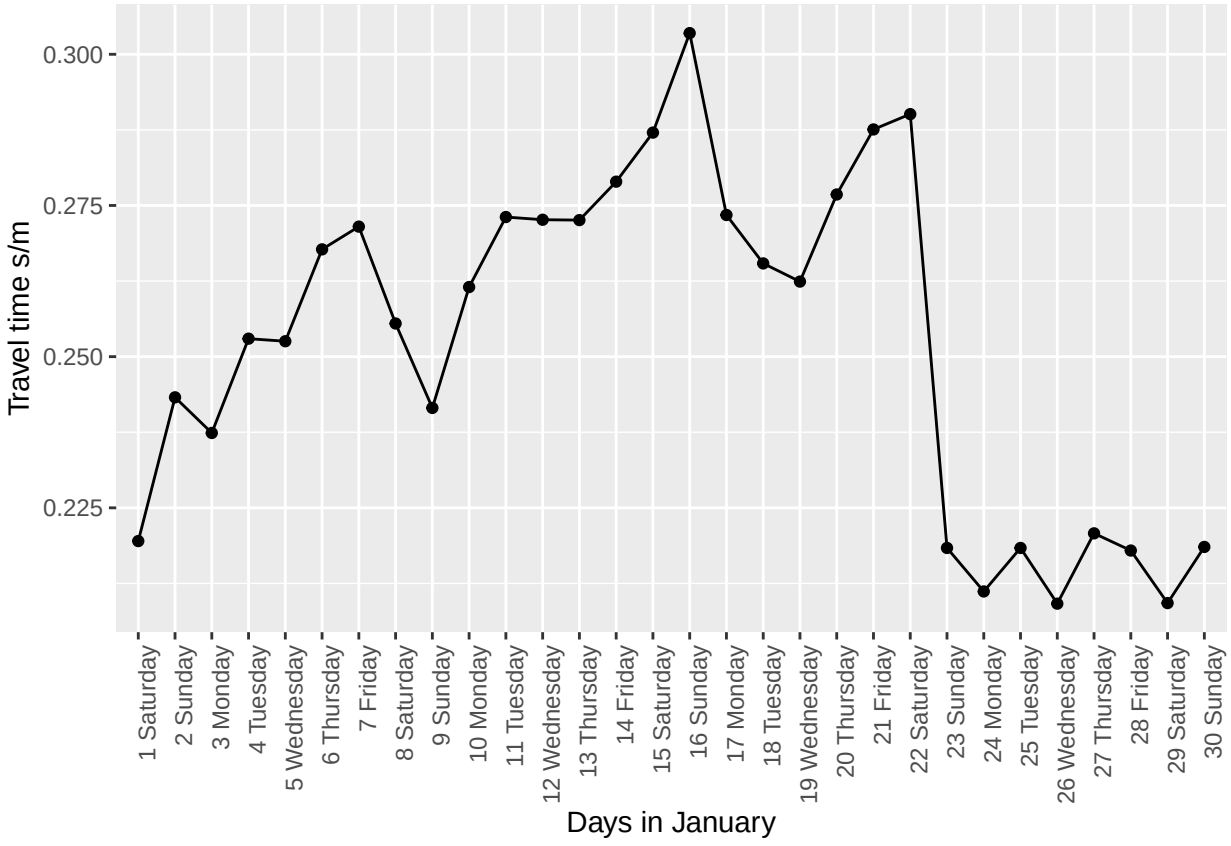


Figure 22: Plot of travel times in January for road segment 1460

5.2.5.2 Aggregated RMSE

To facilitate comparison with the other methods, the forecasted travel times for each road segment are aggregated and the forecasted mean travel time for each day is compared against the actual mean overall travel time for each day. The forecasted travel times are plotted below (Figure 23).

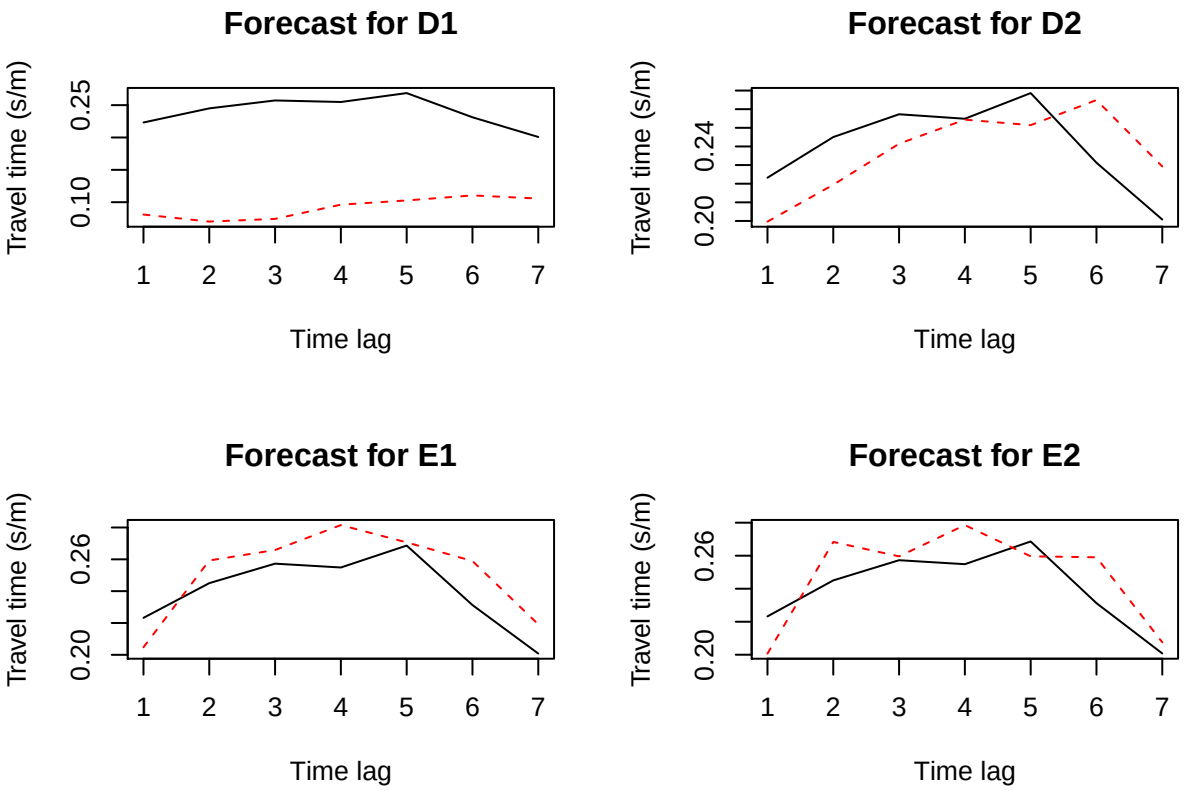


Figure 23: Forecasted aggregated daily travel times for the last seven days in January. The red dashed lines represent forecasted values, while the black solid lines represent actual values.

RMSEs are calculated for each of these models to quantitatively evaluate model fit. The RMSEs for each of the models are shown in Figure 24.

Several observations can be drawn from the RMSE values.

1. Model E1 (weekly differenced MA model) performs the best, and has the lowest RMSE of 0.0186. Thus, it will be used for comparison against other approaches.
2. Inclusion of an autoregressive term causes the weekly differenced mixed AR, MA model (E2) to perform slightly worse (RMSE of 0.0190) compared to a weekly differenced pure MA model (E1). This is unexpected as inclusion of the previous day's values is predicted to facilitate forecasting. Consequently, this suggests that weekly differencing is sufficient to account for the patterns in travel times.
3. Weekly differenced models (E1 and E2) perform better than undifferenced models (D1 and D2).This indicates that travel times are highly dependent on the day of the week. This might be because Westminster is located in Central London, and is therefore highly influenced by commuting patterns of workers.

4. A sharp improvement in model fit is observed with the inclusion of an AR term in the undifferenced data. For the undifferenced data, the pure MA model (D1) has high RMSE of 0.151, compared to the mixed AR, MA model (D2) which has an RMSE of 0.0230. This indicates that the previous day's travel times are very significant in forecasting the following day's travel times when weekly patterns are not taken into consideration. However, the opposite is observed when weekly patterns are considered – model E2 performs slightly worse than E1, indicating that weekly differencing is perhaps more important compared to the previous day's travel times and forecast errors.

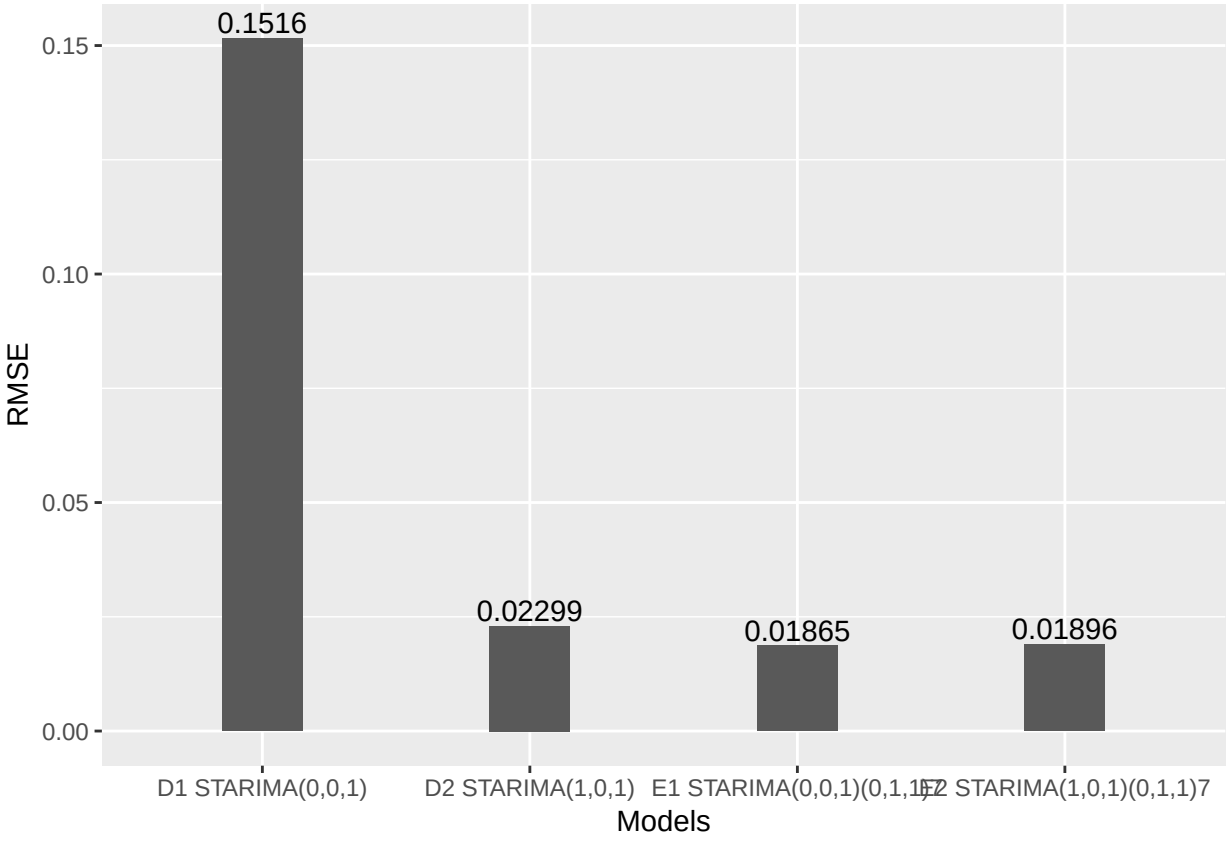


Figure 24: RMSEs of aggregated daily travel times for all 4 STARIMA models

5.3 Model Aggregated Daily Travel Time with SVR: Ju Young Park

5.3.1 SVM

Support Vector Machine (SVM) is an example of a non-parametric method. Non-parametric methods use a sample of the data to derive a model to predict values. The main distinction between the two methods is that parametric methods have a finite number of parameters whilst this number grows (potentially infinitely) with the amount of data available for non-parametric models. They aim to determine a model without having to estimate the parameters.

SVM is a type of supervised learning method under Machine Learning which aims to analyze data and recognize the underlying patterns. It is most commonly used for classification purposes (Support Vector Classification / SVC), but it can and has been adapted to support regression in the form of Support Vector Regression (SVR). It has a distinct advantage over other methods such as Artificial Neural Networks as it produces an optimal global solution instead of suffering from multiple local minima.

Within a dataset, SVC operates by maximizes the separation between the classes using quadratic optimization. Within Machine Learning, this 'line' of separation is referred to as the hyperplane. SVM aims to maximize the distance between points and the hyperplane with the largest maximum margin possible. The support vectors are the closest vectors on the margins, which help to define the hyperplane (Figure 25, adapted from (Gandhi 2018)).

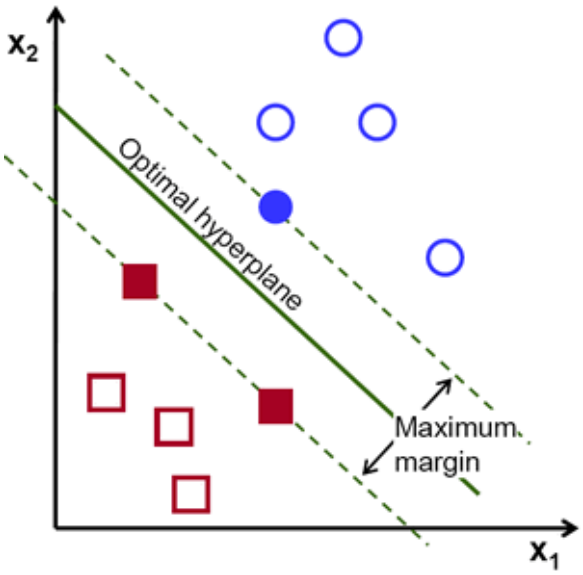


Figure 25: SVM Diagram (Source: Gandhi 2018)

There are two main parameters within SVM: the kernel parameters (I) and the constant C. The kernel parameter is important as it determines the complexity of the solution. The constant 'C' (also referred to as the cost parameter) is the amount of allowable errors in the solution. It is a tradeoff between training error and strictness of the margins. The larger C is, the stricter it is with allowable errors.

5.3.2 SVR

Within SVR, an epsilon parameter becomes important as it determines the width of the 'tube' the regression tries to contain the errors in. This contrasts with SVC as it seeks to keep values separated as far as possible from the hyperplane. Figure 26 illustrates this difference below.

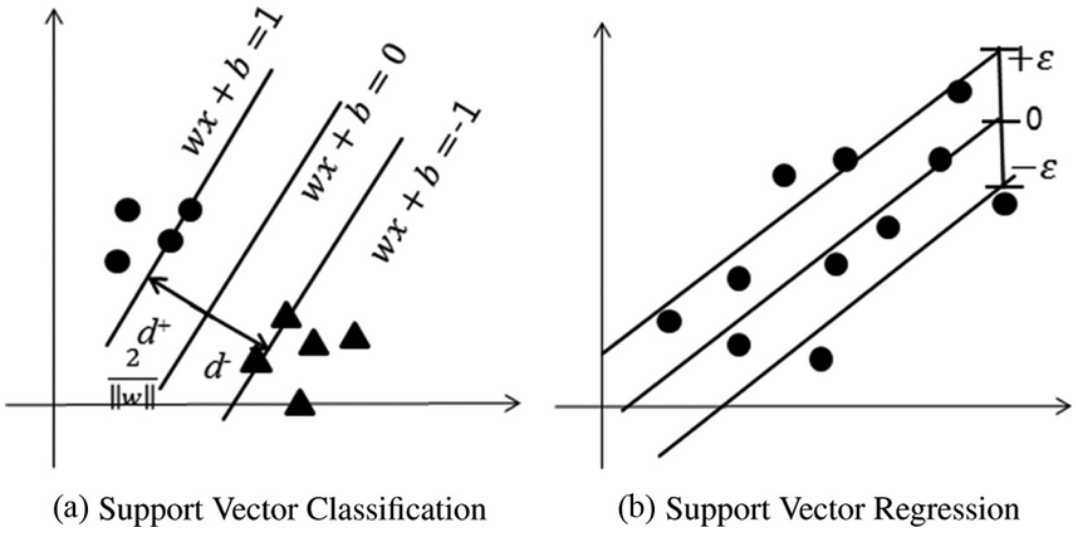


Figure 26: Difference between SVC and SVR (Source: Lee and Choo 2016)

SVR can be linear or non-linear (parametric or non-parametric) depending on the kernel function used. Some kernel functions such as radial basis kernels help to map data from an input space to a higher dimensional feature space, which help to solve non-linear problems.

5.3.3 Experimental set up

First, a time series model was built using a temporal autoregressive structure as there are no information on the predictors of travel time and traffic flow. This allows for forecasting future travel time as a function of previous travel times. The data was embedded to create a matrix that can conduct one-step ahead forecasting using the first m columns as the independent variable and the last (i.e. the original values) as the dependent variable. Different values of m (m = 3, 5, 6, 7) were tested. In other words, m refers to the number of previous days values used for forecasting. As SVR relies on the training set to exhibit all the various situations, this value would consequently impact the forecasting quality.

The data was then divided up into training and testing data. As data frame sizes varied based on the embedding dimension value, the exact numbers of the training dataset differed. However, testing data was set to the last 7 days of January. For example, if m=3, the length of the dataset would be 27 instead of 30. The training dataset would have 20 values, and the rest would be allocated for testing.

Afterwards, the model was trained using a k-fold cross-validation, using k = 5. A k-fold cross-validation helps to prevent overfitting the model on the particular subset of the training set by partitioning the data randomly into k folds. Each of these folds are then left out and the remainder are used to train the model and predict values. Grids of parameters were created to test the model, apart from epsilon values as the `caret` package does not offer this feature. Instead, epsilon values were locked at 0.1. Radial Basis Function kernel was used for this project as it has achieved better performance in other works (Dibike et al. 2001; Keerthi and Lin 2003).

Although k-fold cross-validation can be more computationally expensive compared to other validation methods such as holdout method, it is less biased on how the training and testing datasets were divided. Furthermore, as the k-fold cross-validation runs through each fold once and then subsequently used to train the model k-1 times, it results in a lower variance.

From the model chosen by the k-fold cross-validation, characteristics such as the proportion of points used as support vectors, the training error, and the prediction training error were investigated. The temporal autocorrelation of the residuals of these models were tested. Finally, all the models were collated and

compared to find the optimal model with the lowest prediction RMSE value. RMSE values were used instead of other similar metrics such as Mean Absolute Error as it places a high weight on large errors thus being more appropriate when large errors are not desirable (Wesner 2016).

5.3.4 Results

Example: M = 3

Two models of m=3 were created with various parameters and underwent k-fold cross-validation (Figure 27).

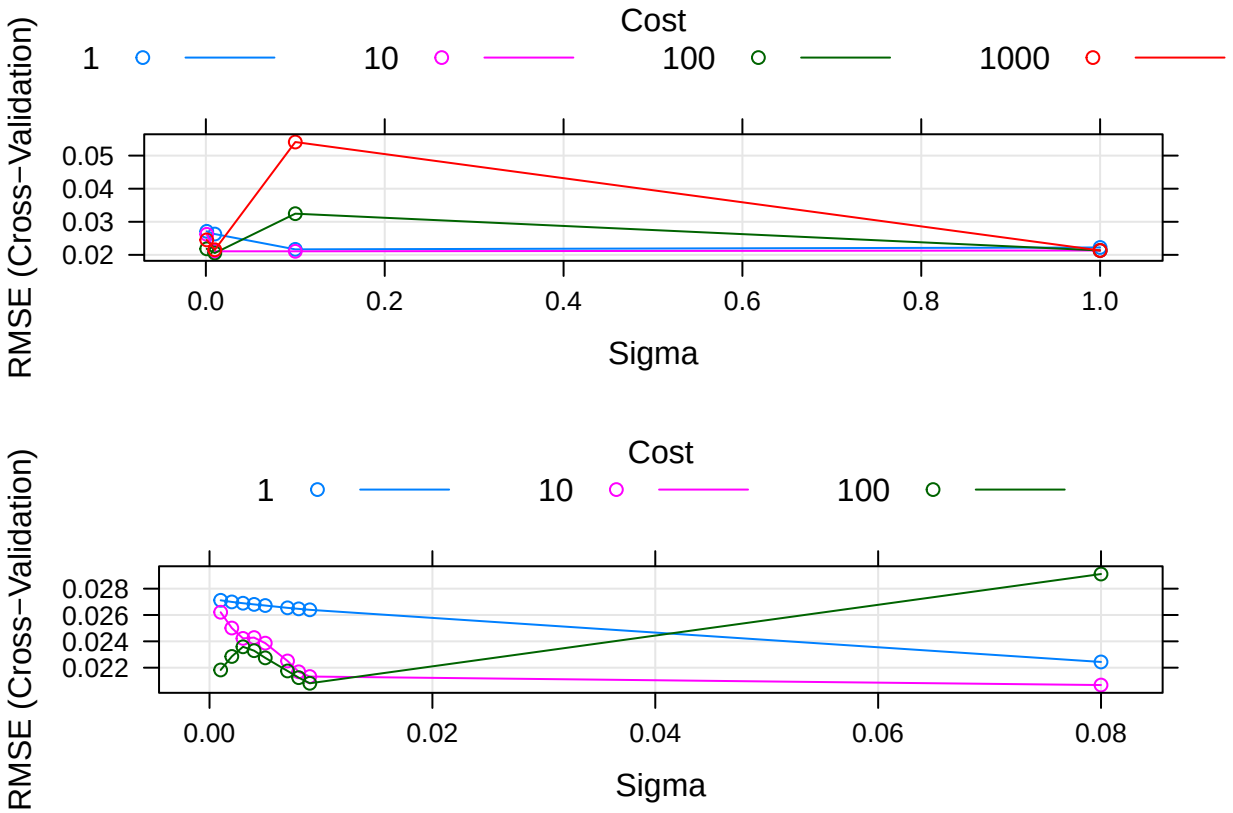


Figure 27: k-fold cross validation

The performance results show that both models had low RMSE for both training and testing in Table 1

Table 1: RMSE Results for M=3

Model	m	Sigma	C	Epsilon	Error	RMSE_Training	P_SV	RMSE_Test	MAE	Time
F1	3	0.010	100	0.1	0.4259454	0.0205495	95	0.0162042	0.0145334	1.00
F2	3	0.009	100	0.1	0.4332898	0.0196287	90	0.0162521	0.0144950	1.42

A comparison between the predicted and observed values show that the models both overestimate day 5 and underestimate day 6. This is shown clearer by comparing the residuals. In general, the models slightly overestimate the values (Figure 28, Figure 29)

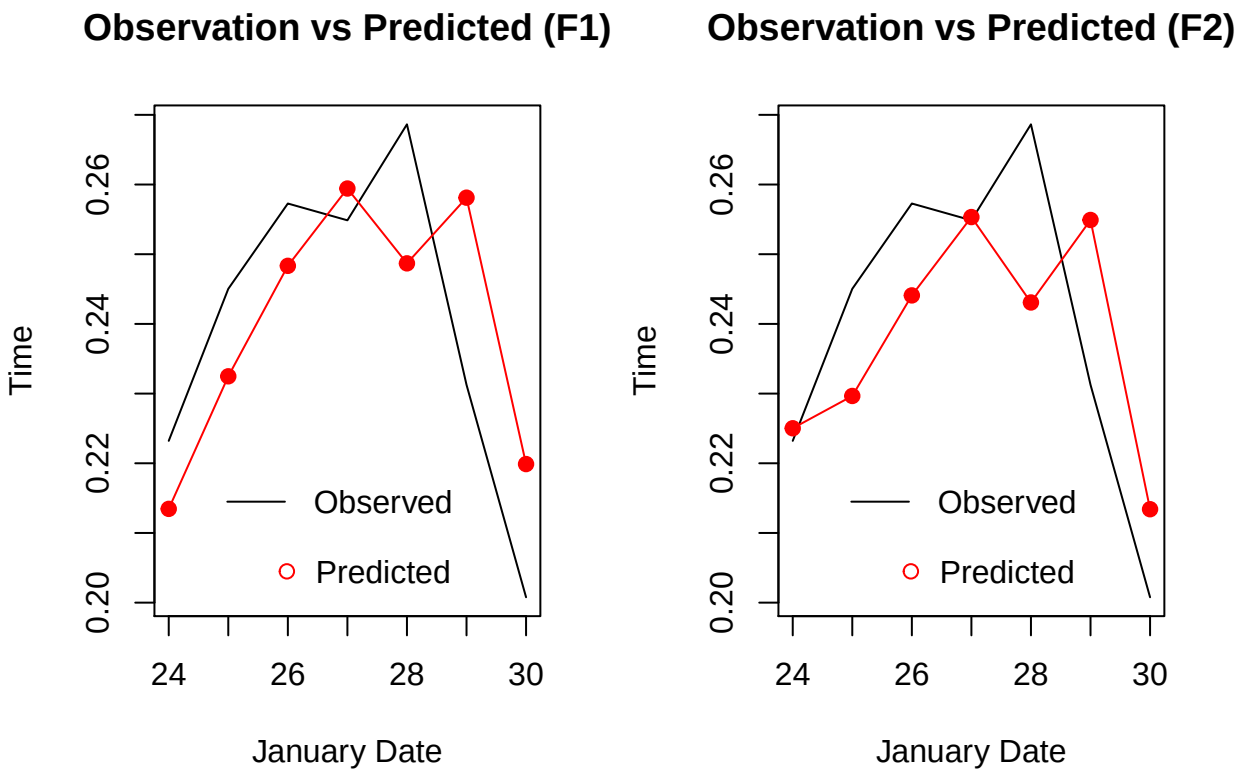


Figure 28: Comparison between predicted and observed values

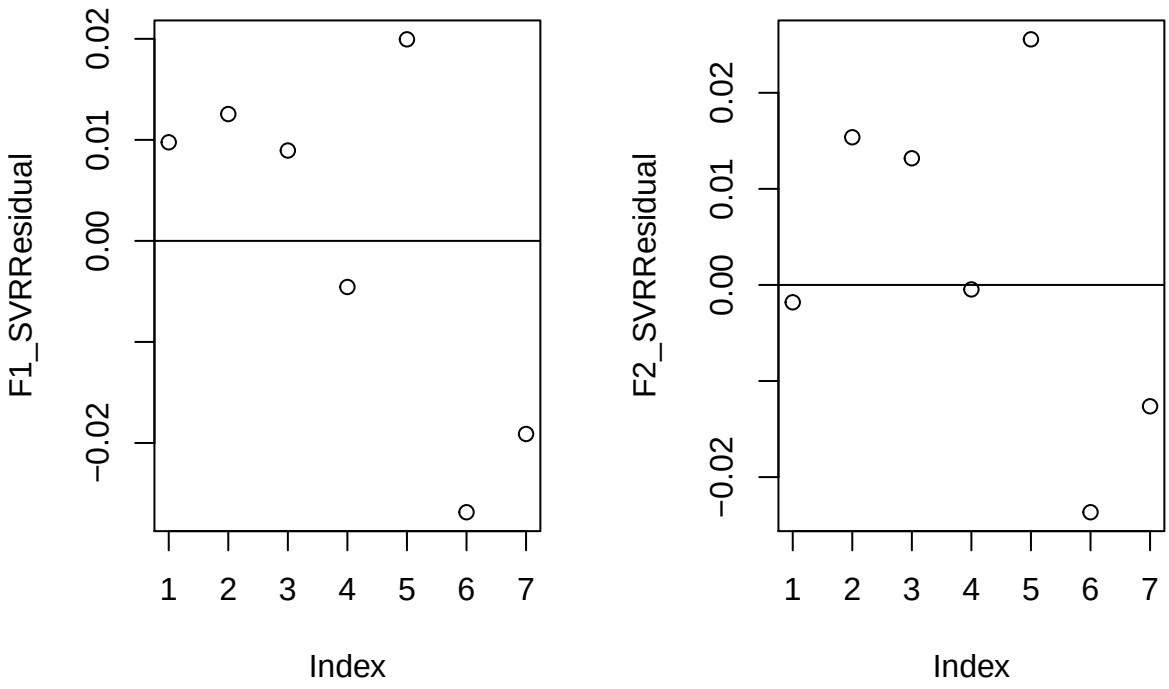


Figure 29: Residual comparison between models

The residual autocorrelation plots for both models suggest there are no significant temporal autocorrelation present (Figure 30)

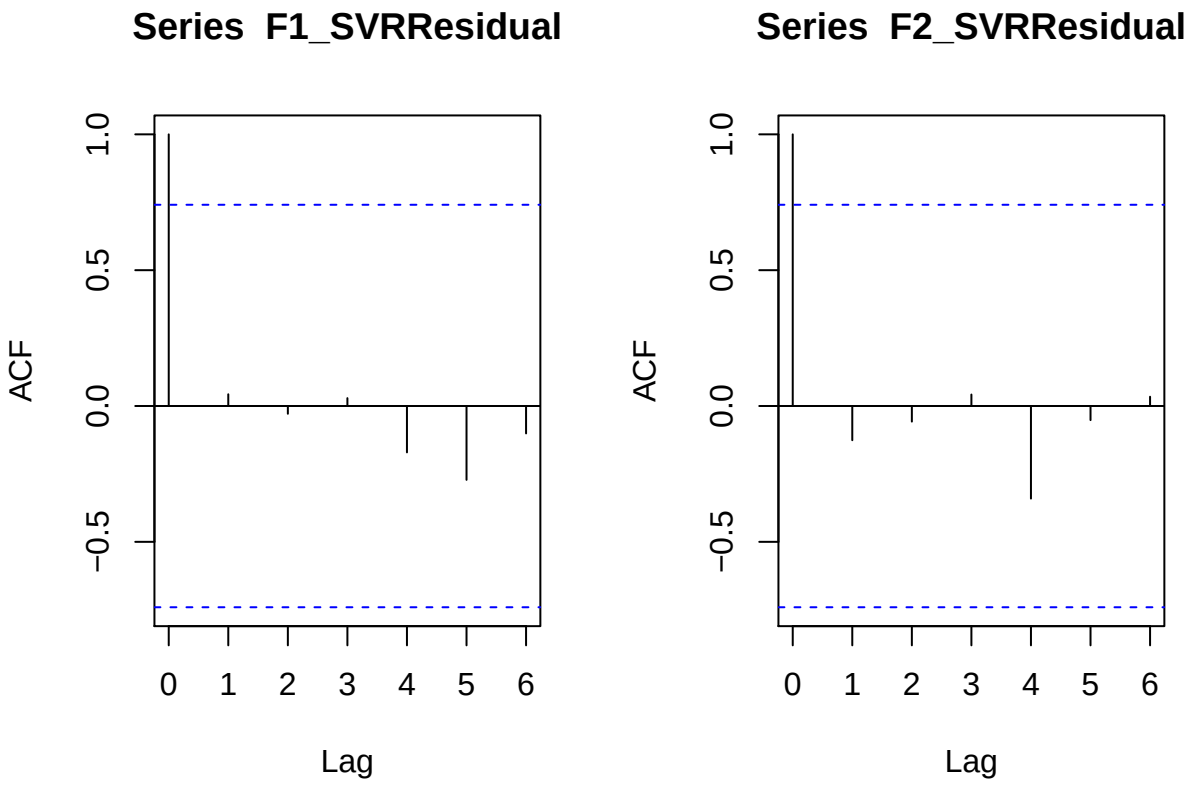


Figure 30: Residual autocorrelation plots for models F1 and F2

Looking at which values were support vectors from the testing data, F1 has more data included as support vectors. Interestingly, both models did not include Day 3 as a support vector (Figure 31)

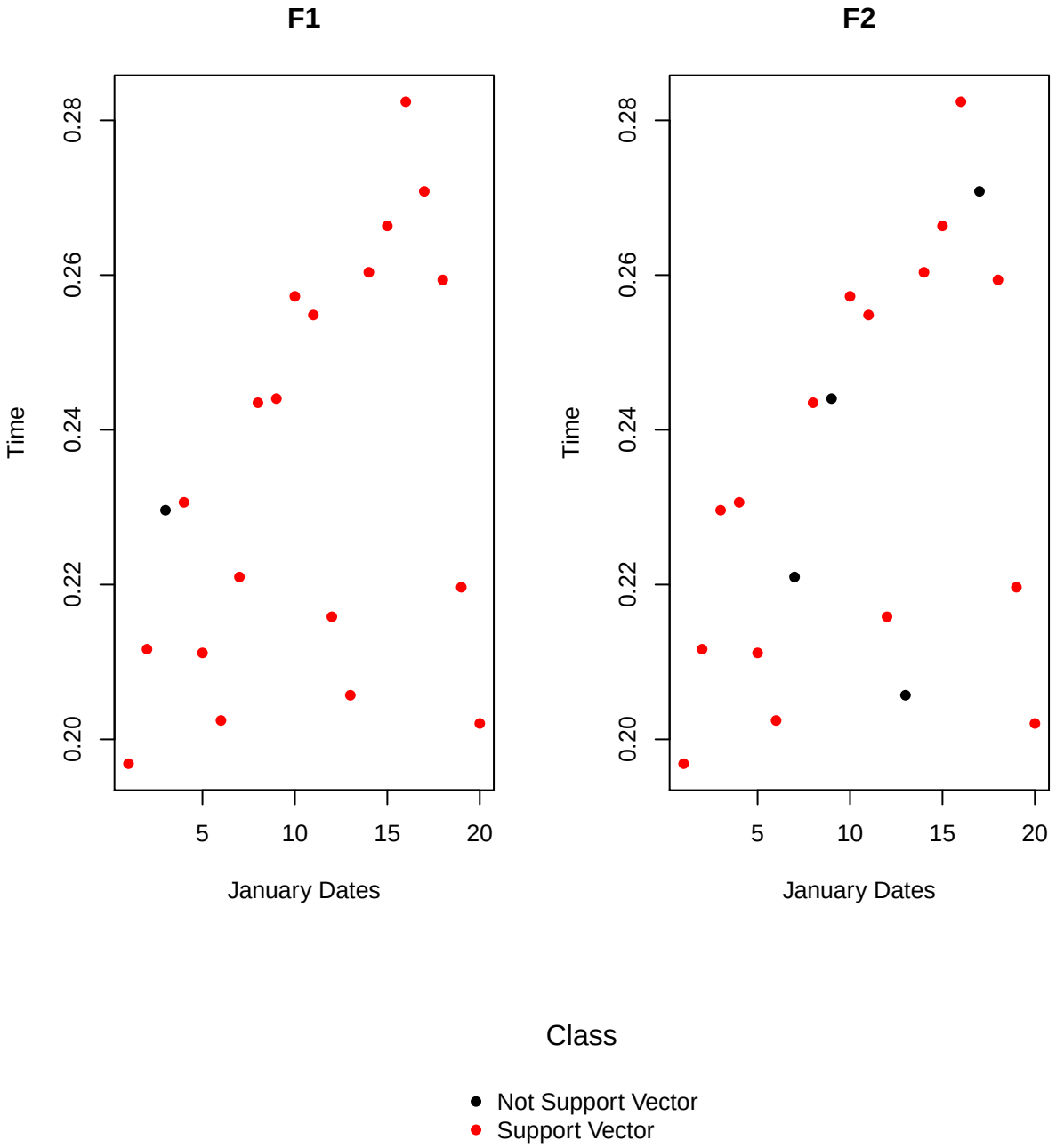


Figure 31: Support vectors and Non-support vectors for F1 and F2

When determining which was the better model, there was an interesting dilemma while looking at the model

performance. (Table 1)

Whilst F2 had a lower RMSE value for the training set, F1 ultimately had the better model for testing with a lower RMSE score. F1 also incorporated more of the data as support vectors (95%) compared to F2 (90%). This result suggests that F2 may have been overfitted as it performs better on data it has already seen compared to on data it has not seen. Both testing RMSE scores were closer to the MAE value, which implies that the models both make many relatively weak errors compared to few but larger errors. Additionally, in terms of duration, F2 had a longer elapsed time of `1.42` compared to F1's `1`.

This process was repeated for the rest of the models (G-J) with m = 4:7. Only the best models of each embedding dimension value are shown below, and the rest are available in the Appendix.

The best models of each embedding dimension value are shown below in Table 2, as well as their k-fold cross-validation result (Figure 32)

Table 2: . Results of best models for each embedding dimension

m	Sigma	C	Epsilon	Error	RMSE_Training	P_SV	RMSE_Test	MAE	Time
3	0.010	100	0.1	0.4259454	0.0205495	95.00	0.0162042	0.0145334	1.00
4	0.010	10	0.1	0.2731022	0.0214496	94.74	0.0137807	0.0107534	1.01
5	0.015	10	0.1	0.3855910	0.0187729	94.44	0.0138845	0.0118091	1.75
6	0.100	1	0.1	0.2823805	0.0248625	100.00	0.0140089	0.0132240	1.00
7	0.090	1	0.1	0.2217186	0.0224526	93.75	0.0119100	0.0100702	1.85

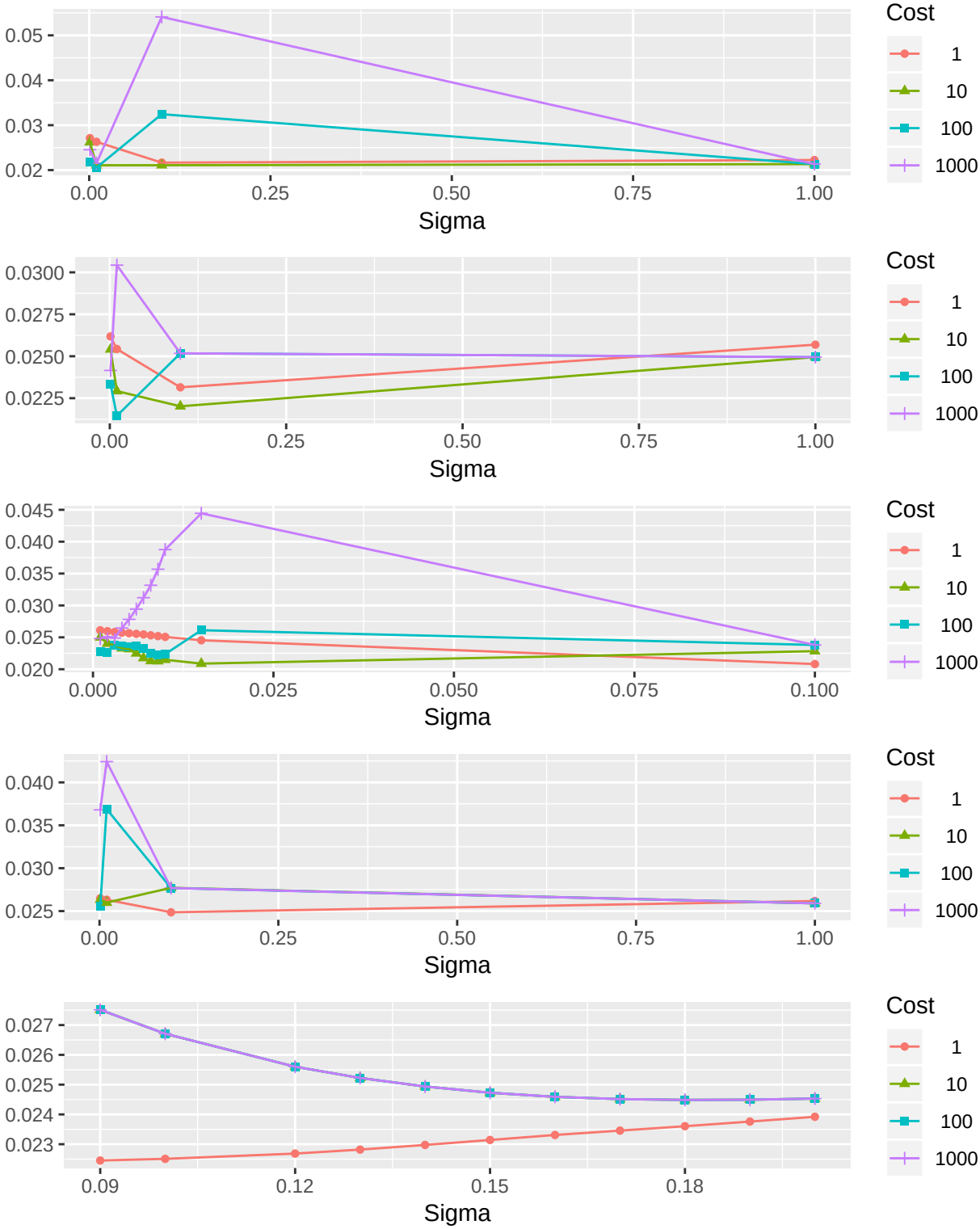


Figure 32: K fold validation results (F to J)

When comparing the observed and predicted values, it is clear that all the models followed the general

pattern where travel flow tends to increase from Monday and reaches a high on Friday followed by a sharp decrease during the weekend (Figure 33). No one model encapsulates the pattern completely. For example, G1 encapsulates the pattern in the early days, but fails to do so after Day 28. Although only I1 and J2 exhibit a decrease between Days 28-29, the other models soon follow suit between days 29-30 to arrive at a similar travel time prediction. Models F1,G1,and H2 all reach their peak on day 29, lagging a day behind.

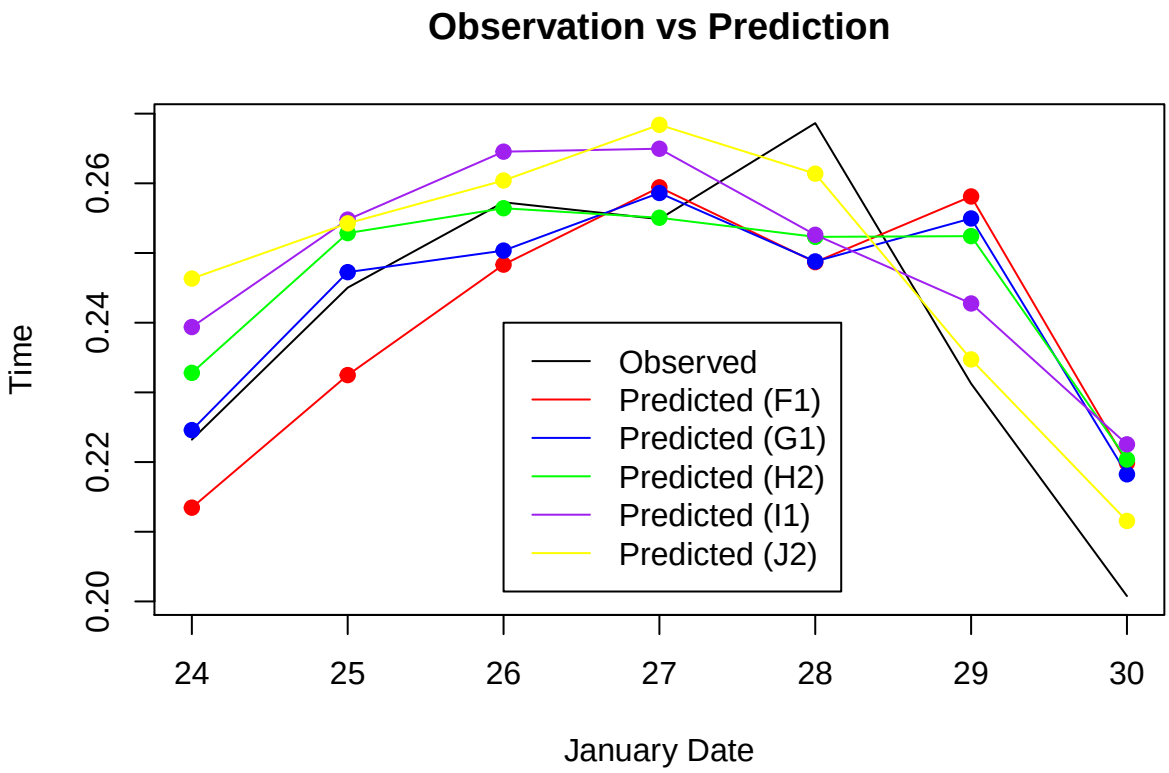


Figure 33: Observed vs Predicted

Residual plots indicate that all models at 5 all overestimate the prediction value. Most of the models apart from two (F1, H2) underpredict most of the values, whilst the rest overpredicts them (Figure 34)

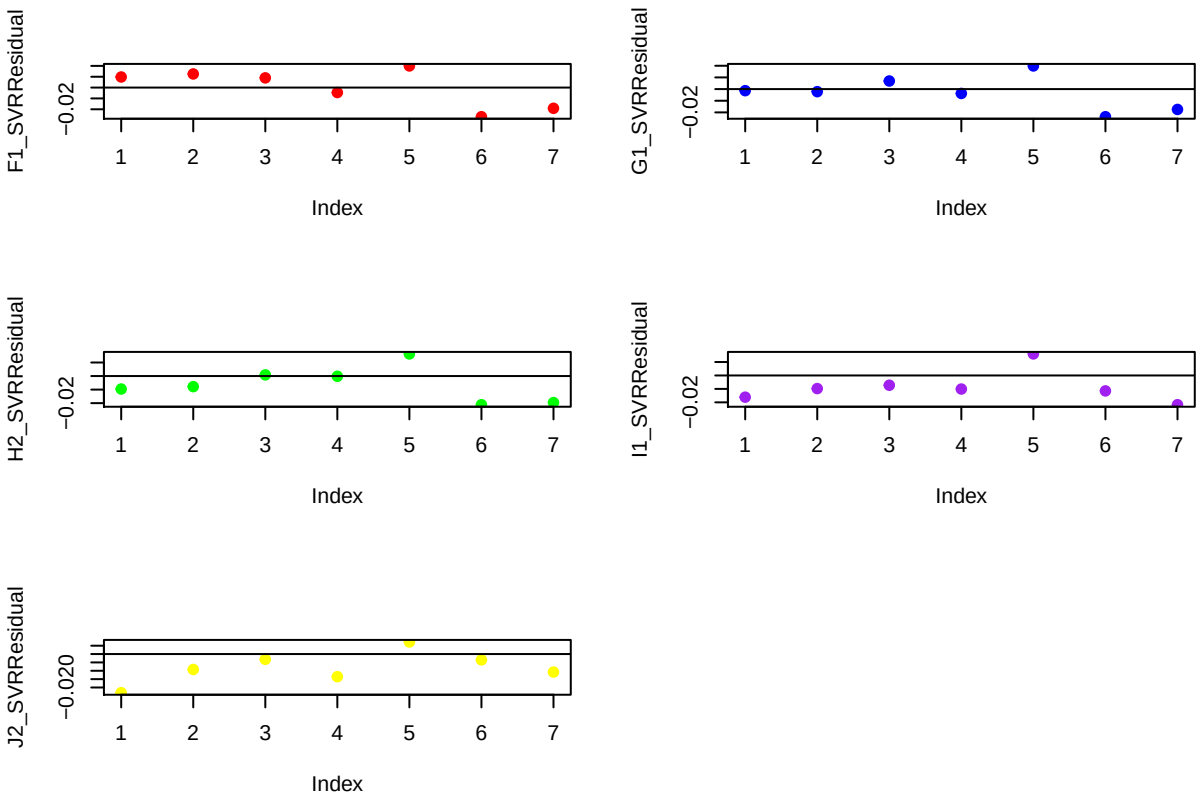


Figure 34: Residuals of all models at 5

None of the model exhibited significant residual autocorrelation (Figure 35).

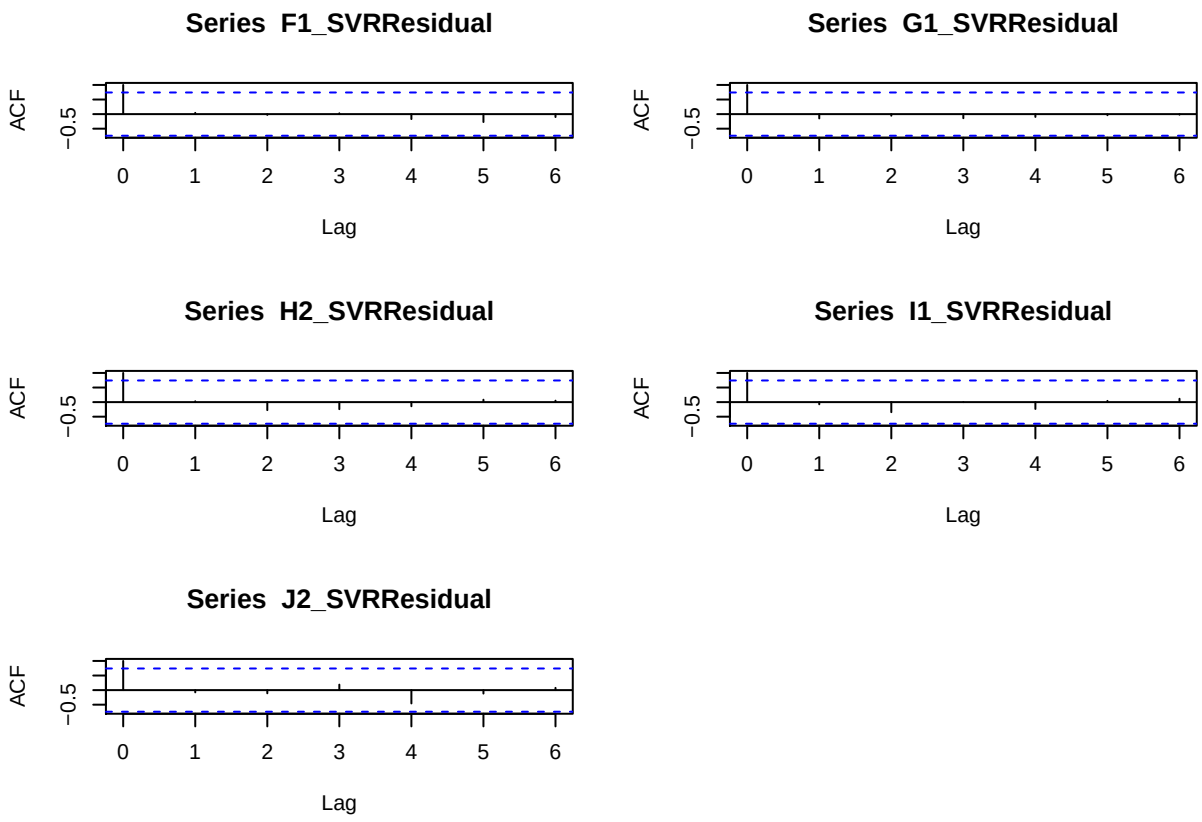


Figure 35: Residual autocorrelation after model fit

Figure 36 shows that the models incorporated most of the data as support vectors. Although I1 incorporated all the data as support vectors within the model, it could also suggest potential overfitting.

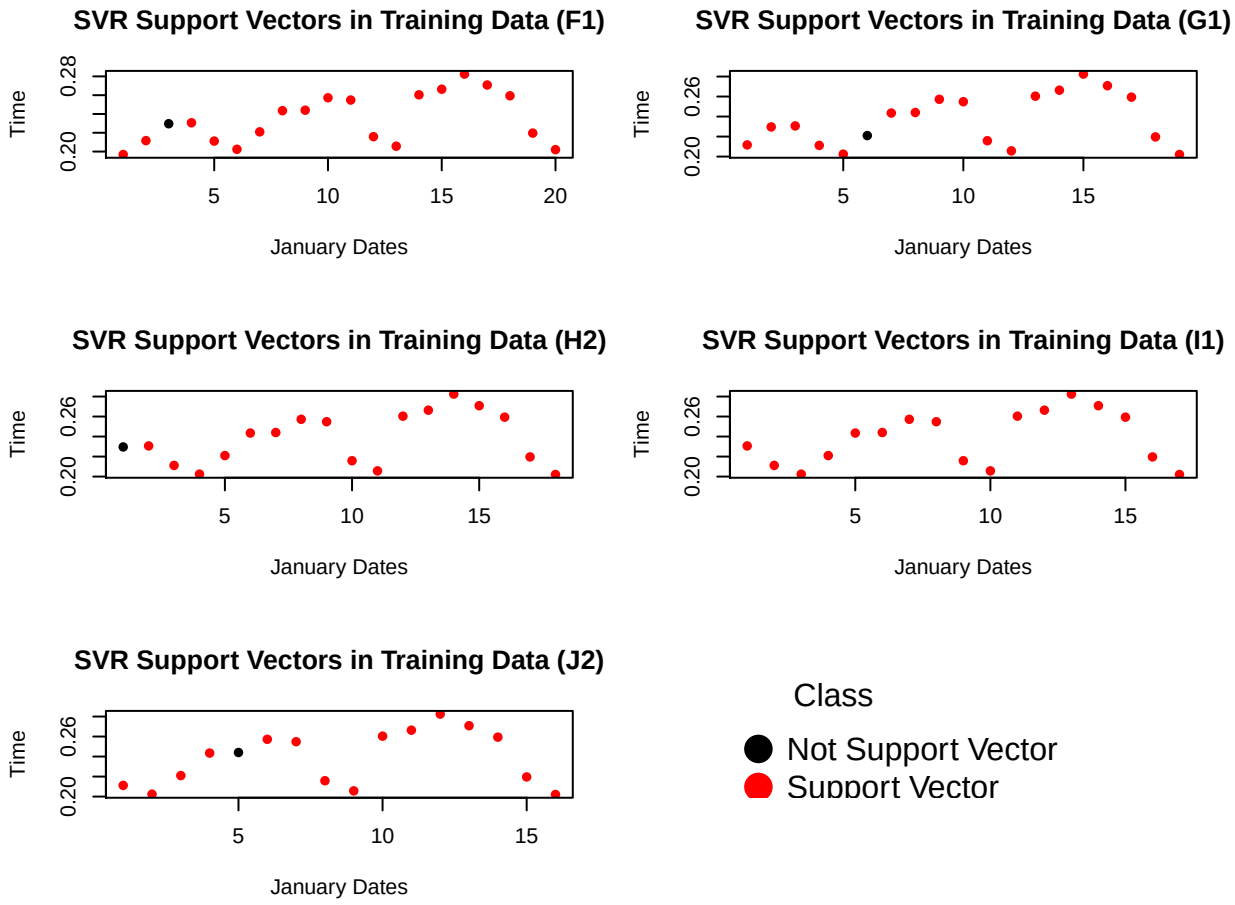


Figure 36: Support Vectors in Training Data

5.3.4.1 Best Model

From the figure below, the best model for time-series SVR in this project would be J2, which uses `m=7`, as it has the lowest `RMSE_Test` value. Additionally, it also has the lowest Training error value of 0.2217186. As the `RMSE_Test` value is closer to `MAE` value than the `MAE^2` value, it also suggests that although the model may make many errors, these errors would be relatively small. Although F2 had the longest elapsed time of `1.85`, it portrays the trend as close as possible to the testing data, which is intuitive as the time series showed there was a weekly seasonality.

5.4 Model daily travel time data on each of the road segments with ST-SVR: Xulan Huang

5.4.1 SVM

In machine learning, Support Vector Machine (SVM) is a supervised learning model and related learning algorithms for analysing data in classification and regression analysis. A vector represents a sample point and each sample is a row of data. The sample points on the decision plane are the support vectors (SV) and these vectors fall out the boundary of margin (see figure below). Given a set of training entities, each of which is marked as belonging to one or the other of the two categories, the SVM training algorithm creates a model that assigns the new entity to one of the two categories, making it non-probabilistic binary linear classification. Furthermore, a method for creating nonlinear classification by applying kernel techniques to the maximum margin hyperplane was proposed by Boser, Guyon, and Vapnik (1992). SVM is used below to classify the specific training values and tested values of last seven days.

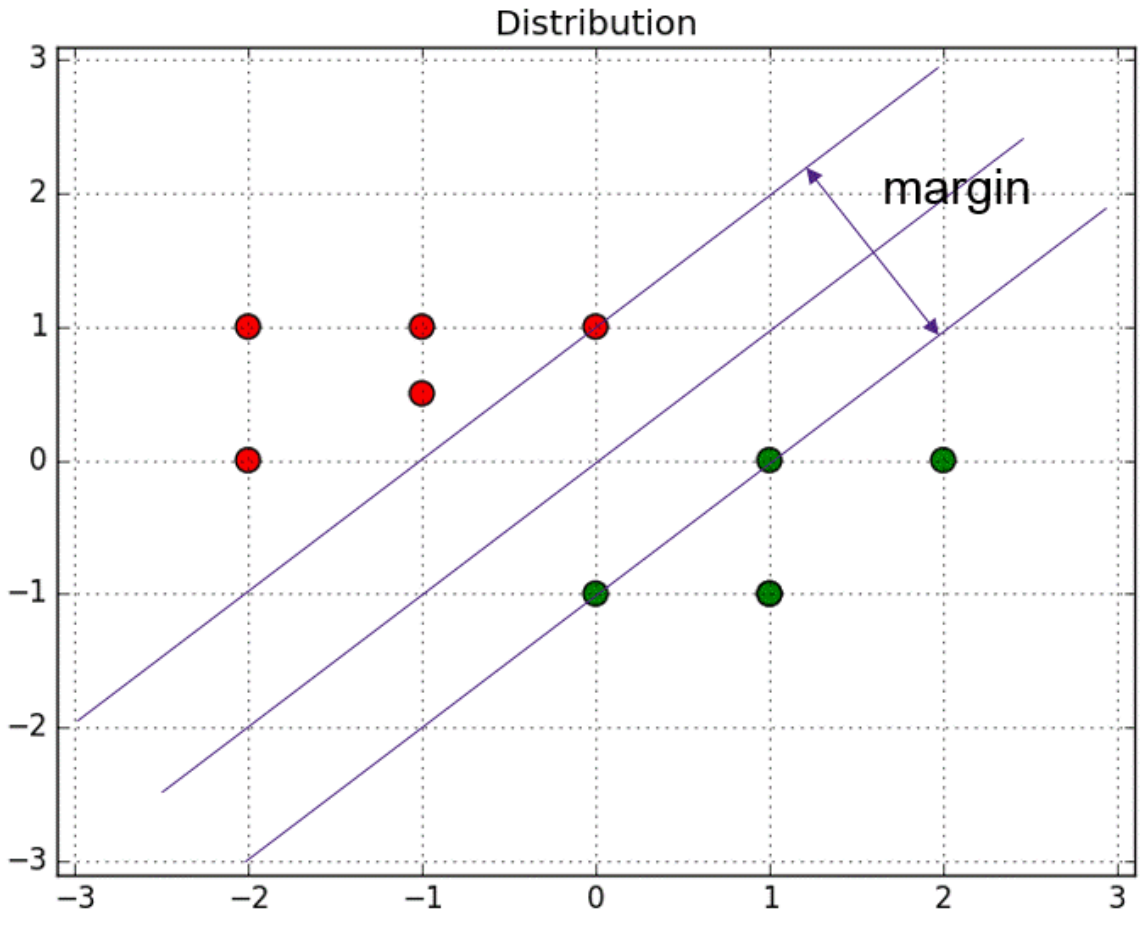


Figure 37: Support Vectors (Alpaydin, 2014)

5.4.2 SVR

Support Vector Regression (SVR) uses SV to complete the regression, which is essentially similar to SVM. In SVR, there are two significant parameters, constant C which is used for limited the errors producing in the data training session and epsilon which describes the width of the specific region to define the point loss

in this region as 0. Within the constraints, the model is looking for a strip instead of a simple line, also the model can be non-linear. SVR provides an optimal model with rather lower root mean square error (RMSE).

5.4.3 Preparation for setting up the space time models

5.4.3.1 Fitting an SVR model

Setting different m values for testing and m = 3, 4, 5, 6, 7, 8 have been used here. The m value means creating a new matrix with how much columns will be created based on the data frame with a time series (ts), which is an important parameter for data embeding. The matrix is used for one step ahead forecasting. The data training method used here is K-fold cross-validation with parameter k = 5 to separate K subsamples, a single subsample is retained as data for the validation model, and the other K-1 samples are used for training (Park et al. 2015). Afterwards, a grid of parameters is created to train and test the model with unchanged epsilon value but changeable sigma and C. Hence, suitable sigma and C can be found in the random values by trying set range of sigma and C in order to show the best model performance.

5.4.3.2 SVM Classification

SVM classification is used for classifying support vectors and non-support vectors. Setting the breaking points to separate the data is vital to decide which value falls to which class. The points are set as (-1, 1, 2) mainly because the points in the initial data fall into this range.

As for the SVM classification, when m=6, all the values are support vectors (see figure below). One more essential point is worth to be mentioned that the data of day three has not been taken into consideration for the reason that it is undefined, even it was replaced with average values of neighbour values in the initial data processing.

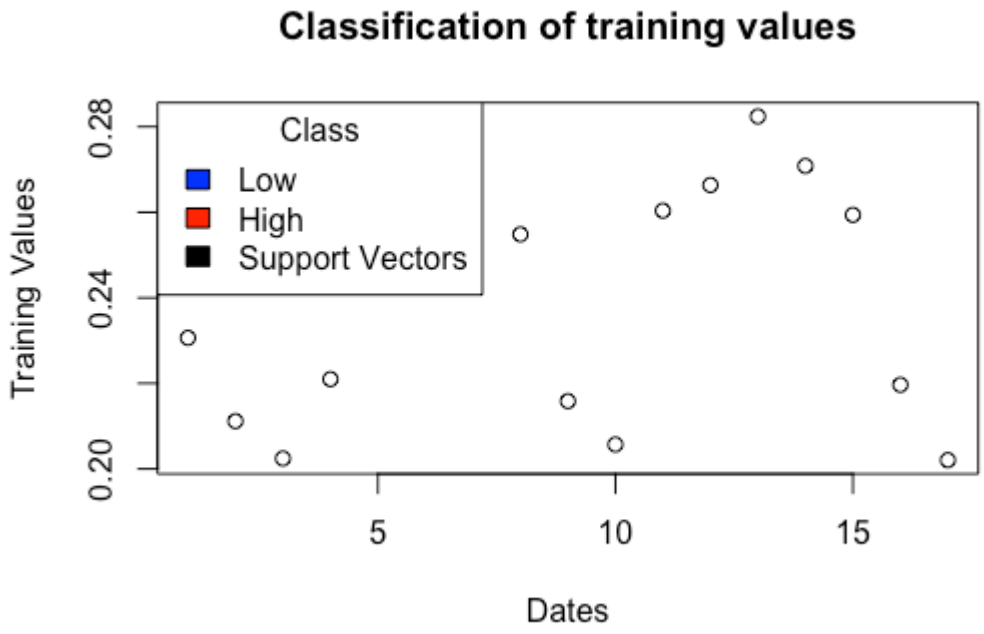


Figure 38: Classification of training values

5.4.4 Result: Comparison of accuracy of different models based on different m values

Whilst finishing the preparation above, the model can be used for one step ahead prediction and plot the results for each embeding dataset independently. From the results showed in the figures, when m = 5 and m = 6, the training data provide well matched predicted models. For m = 5, predicted values fall into the same range of tested one but with slight difference especially after the date of 27, the difference has slightly expanding trend (see left figure below). As for m = 6, the predicted model has the similar pattern with the tested one but rather larger difference with tested one (see right figure below). Interestingly, when after the date of 28, the predicted model gradually merges with the tested data. Hence, based on these results, it is still difficult to know which set of embeding data to be used in the space-time model building. Further verification is probably needed.

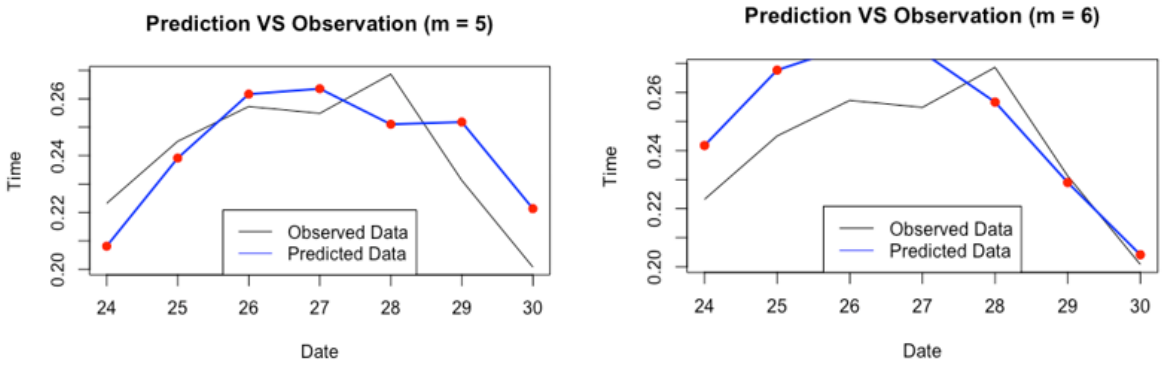


Figure 39: Left: Prediction VS Observation (m = 5), Right: Prediction VS Observation (m = 6)

According to the table below, when m = 6, there is the smallest RMSE also with other errors within the allowable range.

m	Mean Error	Root Mean Squared Error	Mean Absolute Error	Mean Percentage Error	Mean Absolute Percentage Error
3	-0.002662854	0.01489188	0.0126887	-1.530875	5.336162
4	-0.009396224	0.02002137	0.0171747	-4.316203	7.249228
5	-0.009721303	0.01570115	0.01377333	-4.074952	5.621864
6	-0.002221431	0.01469883	0.01325011	-1.215122	5.707408
7	-0.01153873	0.01692162	0.01270851	-4.780671	5.309917
8	-0.03158122	0.03785321	0.03158122	-12.83276	12.83276

Figure 40: The accuracy of different models based on different m values

5.4.5 Building up space-time model

The space-time model built up is for the purpose of prediction that forecasts the last seven days situation of traffic flows for each road in Westminster Borough on January of 2011. The model is set up with its own

previous data, ts, the embeding data which means it uses the matrix created when m = 6 and spatial weight matrix which defines from the dataset of roads in Westminster. All the undefined data in the spatial weight matrix is replaced by 0.

The ts is vital for data training, which is set up by coding mathematic functions to convert real time values to numbers which can used as time series in training session. Also, the space-time series for each specific road is needed, which embeds the series of a particular location along with its adjacent neighbours. According to the embeding data, the data for training and testing are separated.

Afterwards, the kernel function is used to train the data so as to set up the space-time model. As mentioned above, space-time models for each location will be set up to conduct the prediction. After completing these models, the last seven days prediction can be carried out.

In the results of these predicted space-time models, specific conclusion can be drawn. In the last seven days, the models of location 2364 (see left figure below) and location 435 (see right figure below) show the busiest traffic situation and with smallest traffic flows among 25 locations respectively.

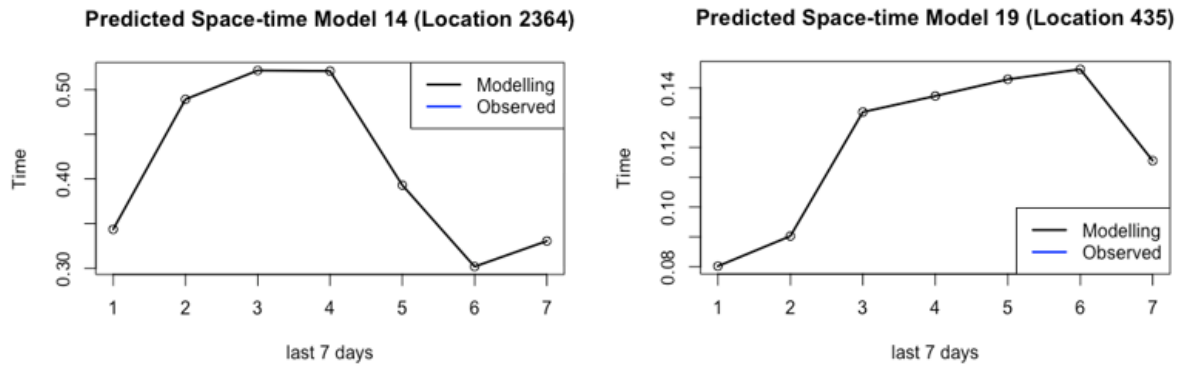


Figure 41: Left: Predicted Space-time Model (Location 2364), Right: Predicted Space-time Model (Location 435)

As for the model of location 1576 (see figure below), the predicted pattern is most similar with the tested one over all the predictions.

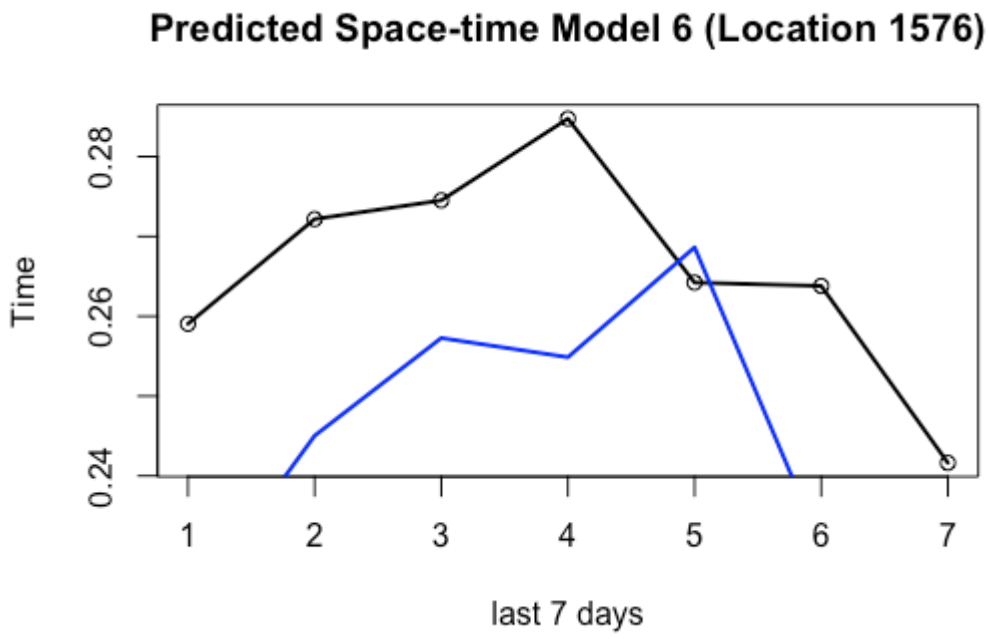


Figure 42: Predicted Space-time Model (Location 2364)

By contrast, space-time models for m = 5 are also produced for the comparison with models of m = 6 for the reason that well fitted model has been carried out above when m = 5. However, those space time models are out of range with the tested one, having similar results with when m = 6. It seems closer to the tested one when m = 6. The model of the location 1576 (see figure below) is rather match with tested one over 25 models, same with m = 6.

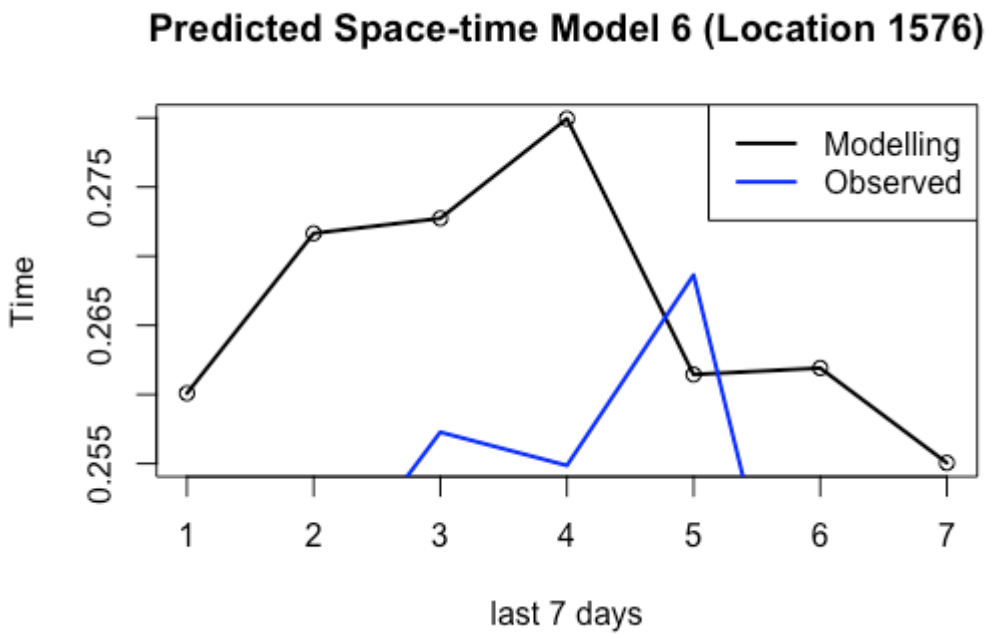


Figure 43: Predicted Space-time Model (Location 2364)

Interestingly, the prediction of busiest case and with smallest traffic flows are still location 2364 (see left figure below) and 435 (see right figure below) respectively.

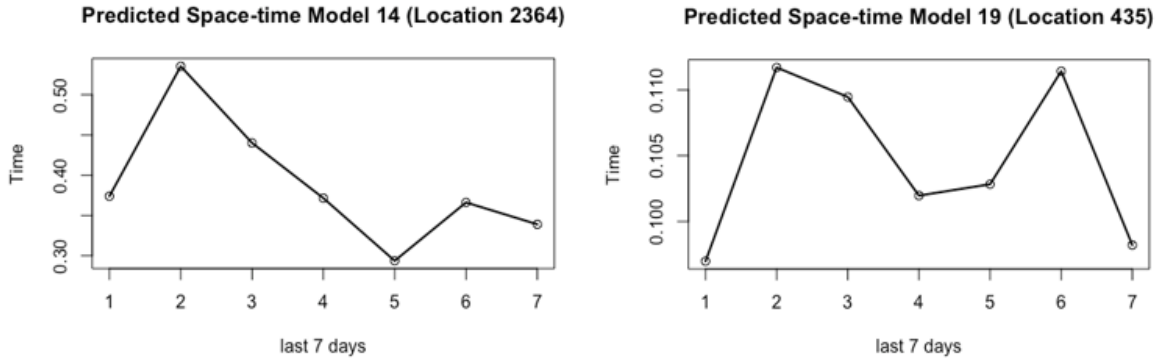


Figure 44: Left: Predicted Space-time Model (Location 2364), RightPredicted Space-time Model (Location 435)

As can be seen in these figures, there is still a rather obvious difference between predicted model and tested one even for the most similar one. Also, comparing with the prediction carried out by temporal autoregressive model, space-time model shows lesser accuracy, though the temporal autoregressive one is not completely coincided with the tested one. The reasons cause the rather lower accuracy are complicated based on insufficient information available. However, several significantly related reasons and corresponding resolutions could be summarized below.

(1) The training data is inadequate. Technically, data stability refers to the stability of time series. However, fundamentally, the stability of the data depends mainly on its variance, especially for uncomplex dataset. The training data is used in modelling is the mean value of each location each day. Additionally, the data training method used here is K-fold cross-validation which is an effective and efficient method but with only first 17 values being trained while building up the model for m = 6. To improve its accuracy, it is worth trying to use the raw data rebuild space-time models again for more usable data. More training data is likely to reduce the contingency when training and reduce the variance of data (Wang, Li, and Xu 2017).

(2) Random seeds are set to be able to generate a series of random numbers to train the data. However, random numbers generated by the random seeds are actually pseudo-random numbers, which are with regulation. Although it ensures that the results obtained each time are the same, it is not random enough to avoid causing dependency with the same random numbers (H.-h. Dong et al. 2012). Hence, multiple random seeds should be set for multiple training to carry out a better result.

(3) Weight matrix includes a large amount of invalid values. Although all the invalid values are replaced by 0, it is still lack of reliance to produce a reliable space-time series. In fact, the weight matrix is adopted from the raw dataset which means it is with flaws so that it is probably unavoidable.

6 Discussion

This section will compare results of the best models produced using the different modelling approaches.

6.1 Comparing RMSE scores

In general, from the exploratory analysis, it is known that the only observable pattern across all the weeks is that the travel time is significantly shorter on the weekends (Figure 45).

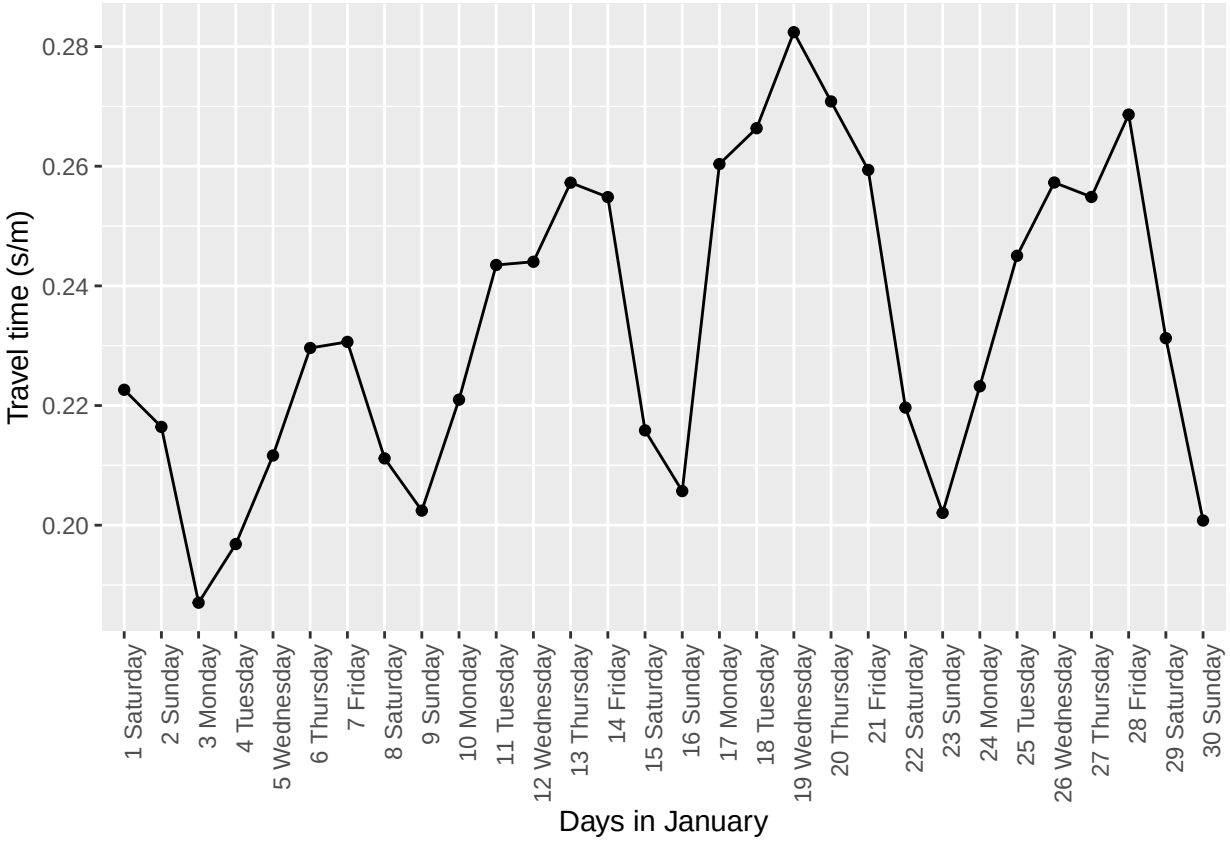


Figure 45: Average daily travel time in s/m across in Westminster.

Looking at the predictions for the last 7 days across the models, the following plot is obtained (Figure 46).

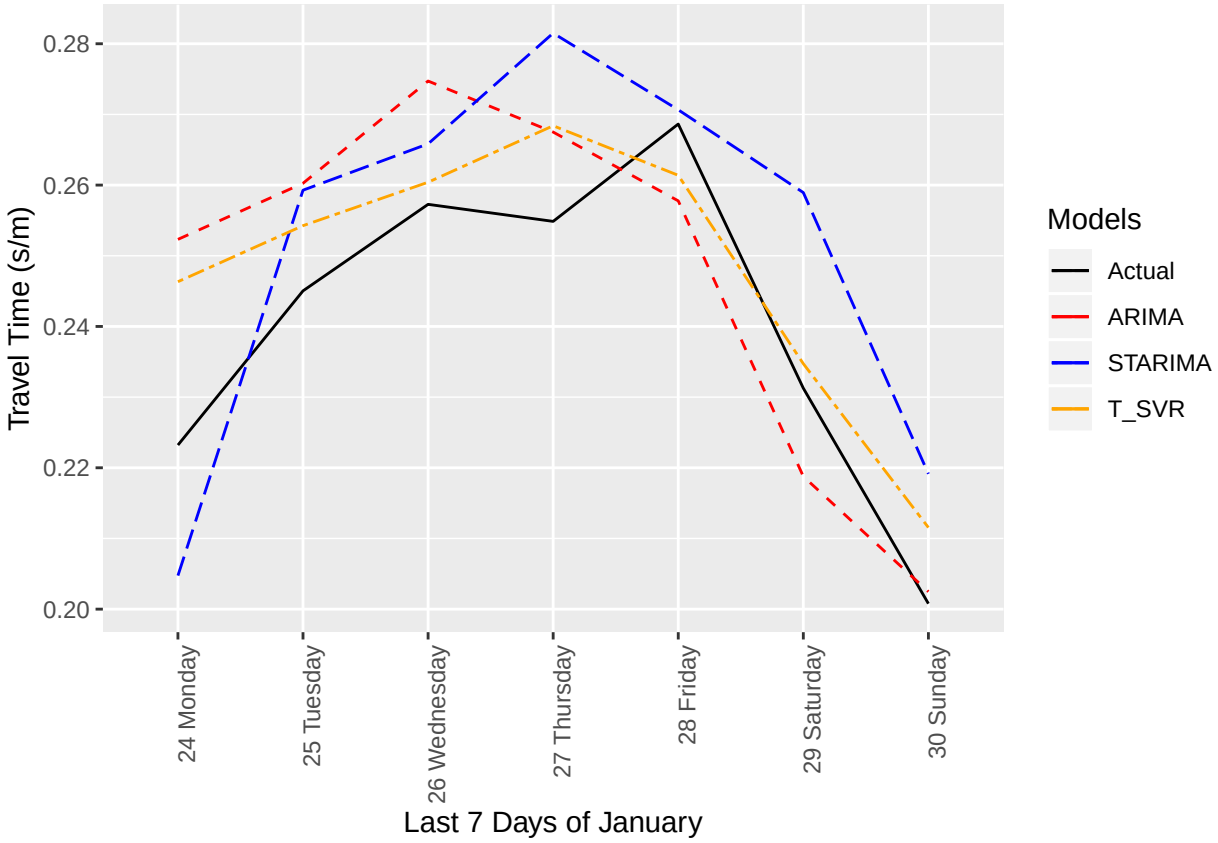


Figure 46: Comparison of Predicted Travel Times Among the Models for the Last 7 Days in January

From the plot, it is observed that actual travel times increases from Monday and decreases at the end of the week. This trend is successfully captured in all models.

However, the peaks and troughs of the actual travel time graph are not successfully predicted in all the models. This is expected because there had been no clear weekly patterns for the peaks and troughs and the models were only trained on limited data (23 days).

To quantitatively evaluate the models, RMSE (Figure 47) was used.

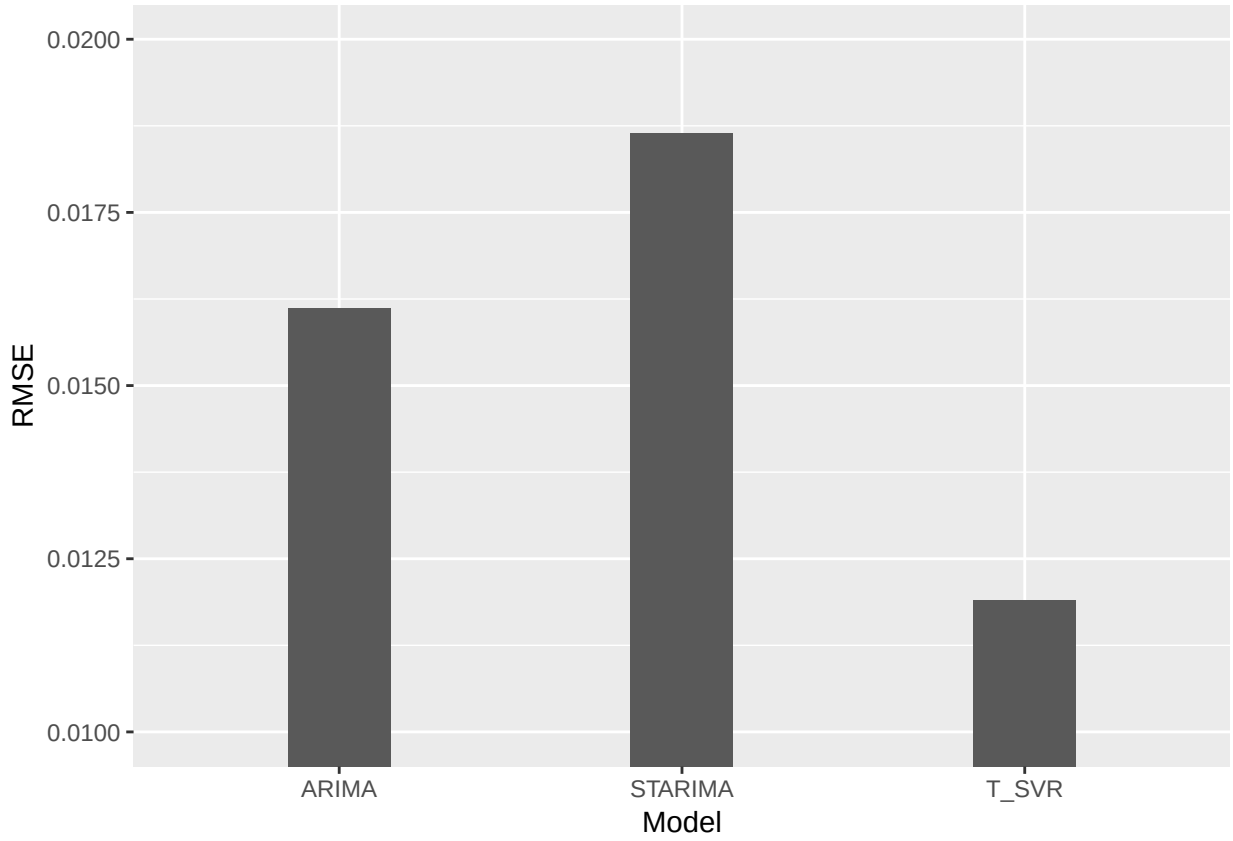


Figure 47: Comparison of RMSEs Among the Models

Overall, the SVR model has the lowest RMSE. This denies our third hypothesis:

H3. Forecasting time-series with ARIMA gives better accuracy than with SVR

It was realised a one-step forecast was implemented in SVR whereas a multi-step forecast was implemented in ARIMA. This means that the SVR model in our case, was privy to the actual last t-1 days of data. A fairer comparison may be to implement a multi-step forecast in SVR – use predicted values for the last t-1 days to train instead. We hypothesise that a one-step forecast was a significant factor in improving the RMSE score.

The ARIMA model performed better than the STARIMA model, which confirms our first hypothesis: H1. Modelling aggregated daily travel time by ARIMA is more accurate than modelling travel time by road segments by STARIMA and then aggregating it

For H1, this may confirm our hypothesis that given a small amount of data, STARIMA may overfit to each road segment. Also, ARIMA directly models what is to be predicted – the daily travel time itself – whereas STARIMA estimates the daily travel time by modelling the daily travel time of each road segment. Estimating travel time for each road segment introduces 25 times more errors as there are 25 road segments.

For hypothesis 2 and 4:

H2. Modelling aggregated daily travel time by SVR is more accurate than modelling travel time by road segments by ST-SVR and then aggregating it

H4. Forecasting time-series with STARIMA gives better accuracy than ST-SVR

These hypotheses remain inconclusive as the ST-SVR model was not successfully implemented.

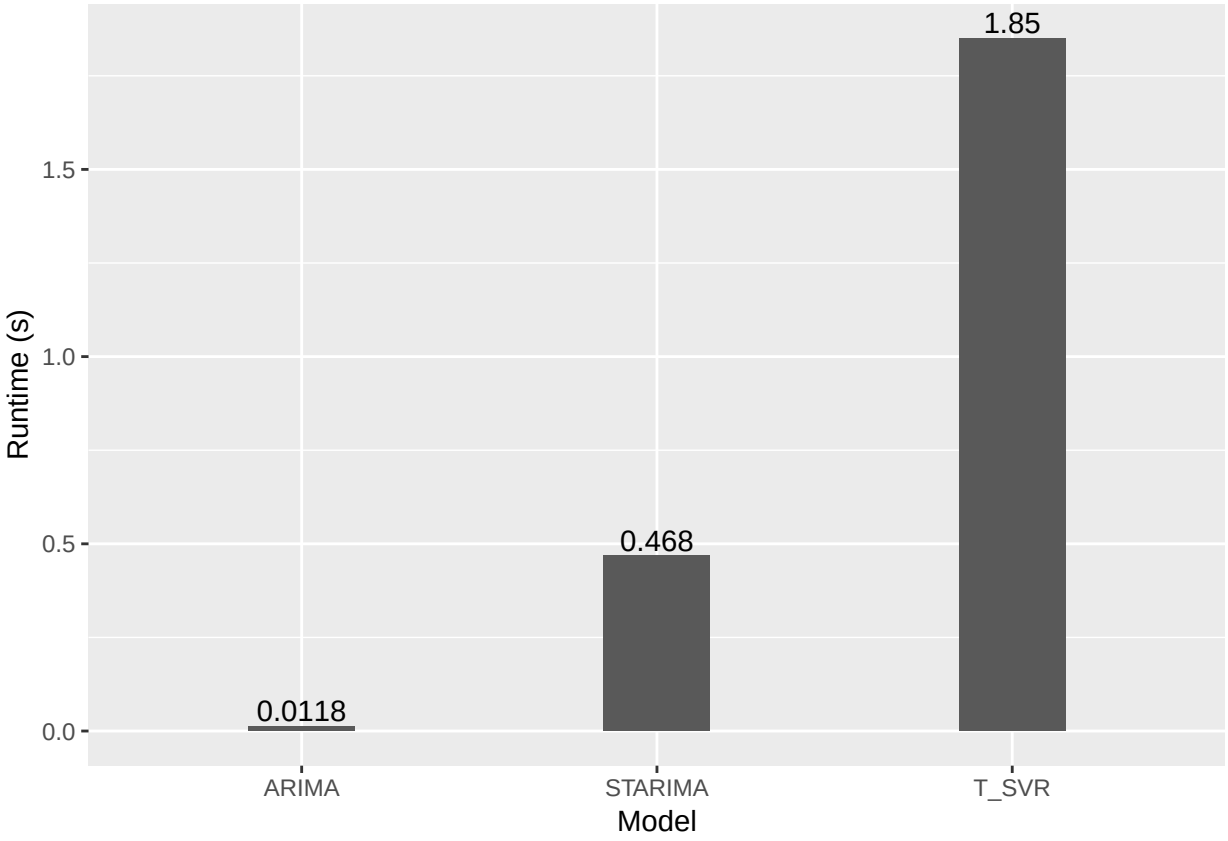
However, as comparisons between ARIMA and STARIMA were successful, this may give some insight of the likely relative performance of ST-SVR. We hypothesise SVR would outperform ST-SVR, given our findings above that ARIMA outperformed STARIMA.

6.2 Comparing General Model Performance

This section will compare the performance of the 3 models in terms of: runtime, ease of implementation and interpretability.

6.2.1 On Runtime

The amount of time taken to train the model and predict the results was recorded for each of the 3 models on the same computer and shown below:



The results show that SVR has a significantly longer runtime as compared to ARIMA and STARIMA (it is around 156 times slower than ARIMA). This is due to the fact that it is a machine learning model so it requires time to train and optimise over the parameters (sigma, c, epsilon, etc.). STARIMA is about 40 times slower ARIMA and this is due to the incorporation of the adjacency matrix and that it has to do a prediction for each of the 25 road segments. Hence, ARIMA performs the best in terms of runtime.

6.2.2 On Ease of Implementation

For the ease implementation, we think that ARIMA is the easiest to implement.

As the ARIMA model is from the *forecast* package which open-source, widely used and has properly undergone multiple revisions and improvements, it is likely to be the most user-friendly and reliable. It also has many resources online to guide users through the use of the package.

While the SVR model was trained and predicted using the *caret* package, which is open-source and well-documented, using it for time-series forecasting requires the data to be prepared in the correct format beforehand. The dimensions of the embedded data was also an additional parameter to be considered. This makes the implementation of the SVR model more tedious that the ARIMA model.

The STARIMA package is not open-source. This means that there are fewer contributors and users and thus is not as well-developed.

6.2.3 On Interpretability

For interpretability, ARIMA and STARIMA would be easier to interpret than SVR as they are parametric models where one defines the variables. In contrast, SVR, being a machine learning model, is a black box. Hence, it is also more challenging to interpret results derived from SVR than from ARIMA and STARIMA.

7 Conclusion

Overall, in terms of RMSE, the SVR model outperforms the ARIMA and STARIMA model, but this is keeping in mind that the SVR model was a one-step forecast whereas the ARIMA and STARIMA models were multi-step forecasts. In terms of runtime, ease of implementation and interpretability, ARIMA outperforms all the other methods.

We think that the ease of implementation is a significant factor to consider because having a high ease of implementation allows us to focus on the task at hand, instead of trying to figure out how to get it working. Also, depending on the project, it may be more advantangeous to obtain fairly accurate results quickly than to achieve extremely accurate results slowly. Getting results quickly enables us to update, change or pivot our strategies early on, which helps to minimise the chances of failure for the project. In most cases, we would favour R packages that are well-supported and documented. However, there are some projects that require specific and most R packages are too general in nature. Hence, these conclusions can change depending on the project requirements, nature of the dataset and the size of the dataset.

Future work on modelling travel time can look towards incorporating more data, perhaps over several months or years. We could also look at modelling time periods. Travel patterns vary across the day. For example, we might expect the rush hours (9am and 6pm) to have the highest travel time, or even model weekdays separately from weekends.

8 Appendix

8.1 Appendix A

```
##
## ######################
## # KPSS Unit Root Test #
## ######################
##
## Test is of type: mu with 2 lags.
##
## Value of test-statistic is: 0.2882
##
```

```
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

8.2 Appendix B

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_df
## Dickey-Fuller = -2.9557, Lag order = 3, p-value = 0.2055
## alternative hypothesis: stationary
```

8.3 Appendix C

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 2 lags.
##
## Value of test-statistic is: 0.0888
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

##
##  Augmented Dickey-Fuller Test
##
## data:  .
## Dickey-Fuller = -3.6224, Lag order = 3, p-value = 0.0475
## alternative hypothesis: stationary
```

8.4 Appendix D

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 2 lags.
##
## Value of test-statistic is: 0.2466
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  .
## Dickey-Fuller = -3.7087, Lag order = 2, p-value = 0.04223
## alternative hypothesis: stationary
```

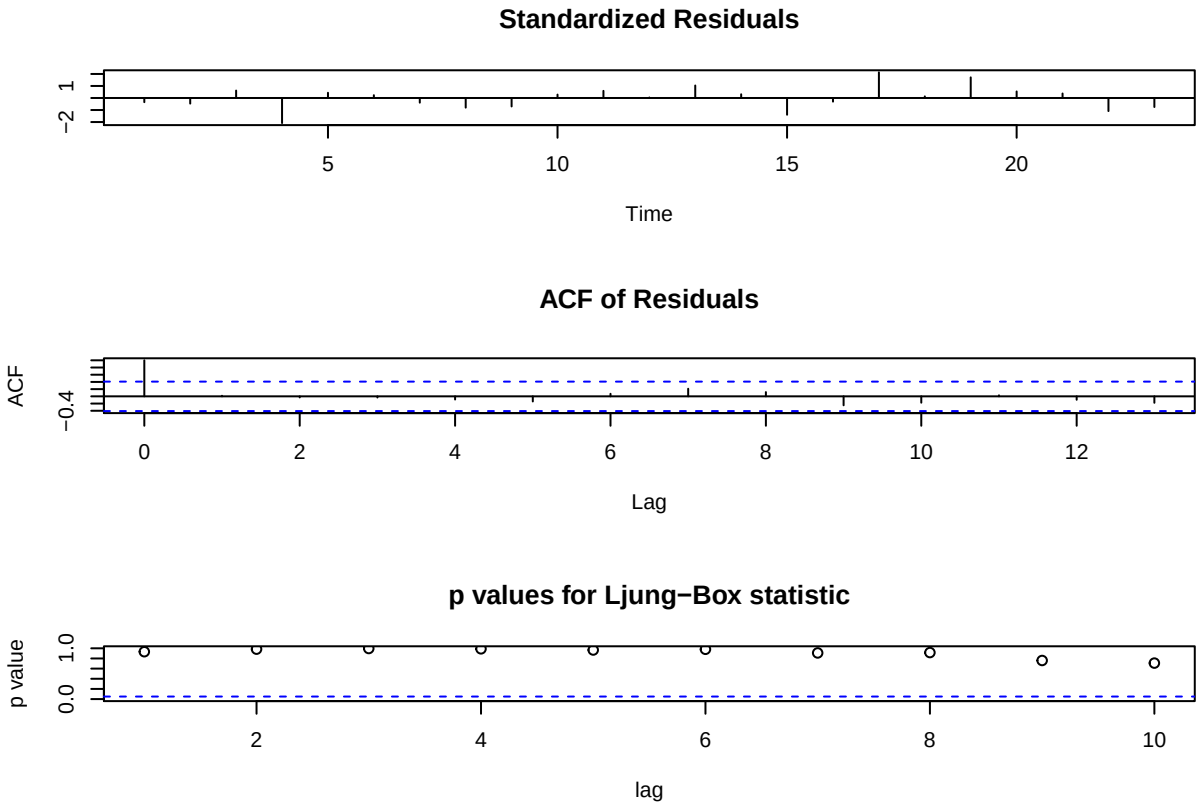
8.5 Appendix E

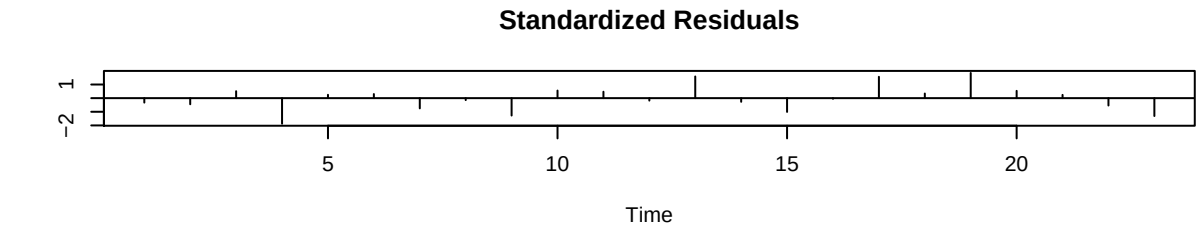
```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 2 lags.
##
## Value of test-statistic is: 0.1106
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

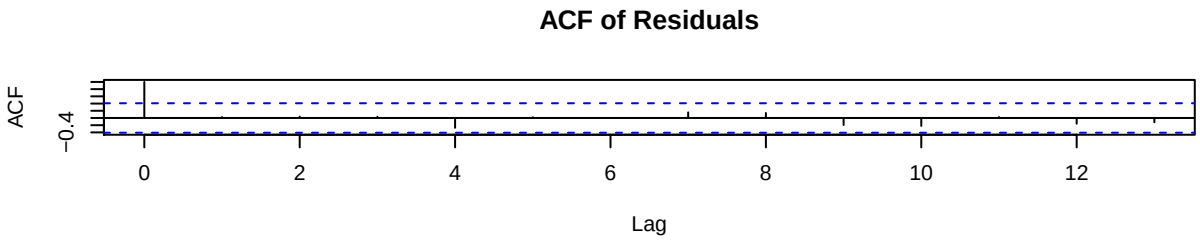
```
##
##  Augmented Dickey-Fuller Test
##
## data:  .
## Dickey-Fuller = -2.9116, Lag order = 2, p-value = 0.2251
## alternative hypothesis: stationary
```

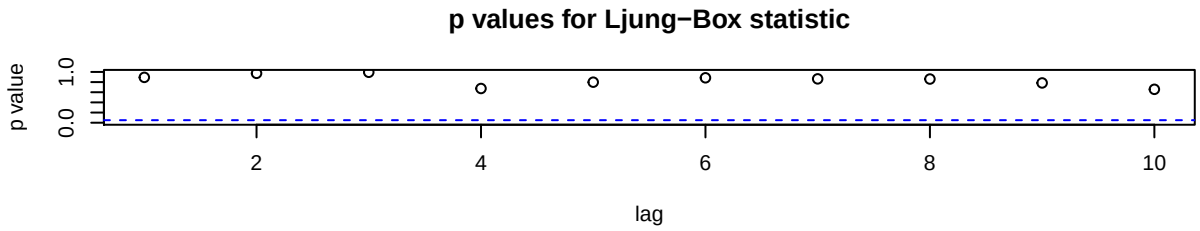
8.6 Appendix F

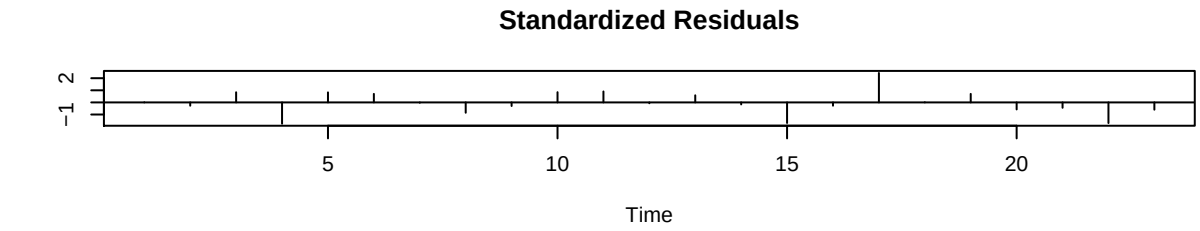
Diagnostic checking: ACF and Ljung-Box test results for models A1, A2, B1, B2, C1 and C2 respectively.

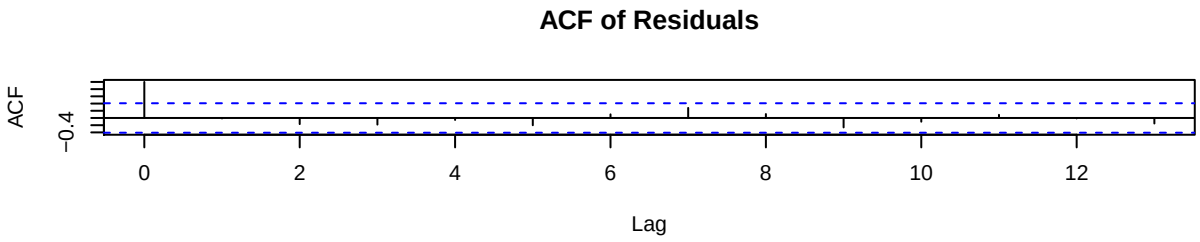


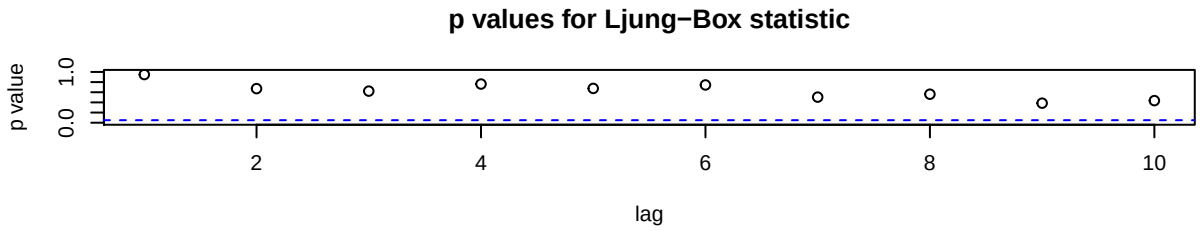

Standardized Residuals
1
−2
5
10
15
20
Time

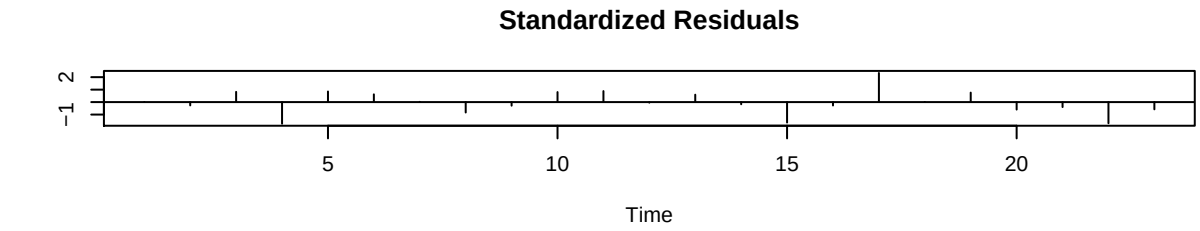

ACF of Residuals
ACF
−0.4
0
2
4
6
8
10
12
Lag

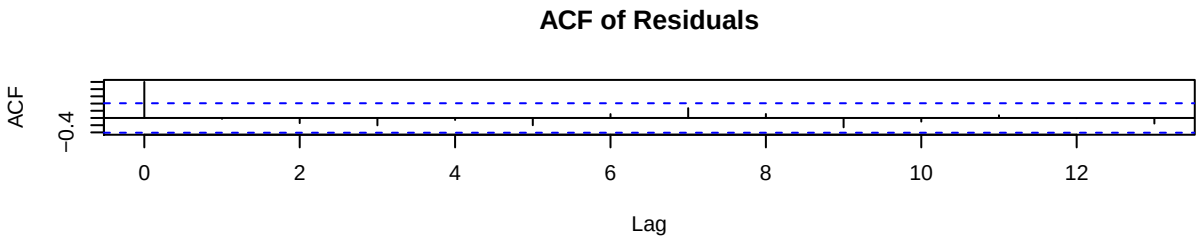

p values for Ljung−Box statistic
p value
1.0
0.0
2
4
6
8
10
lag

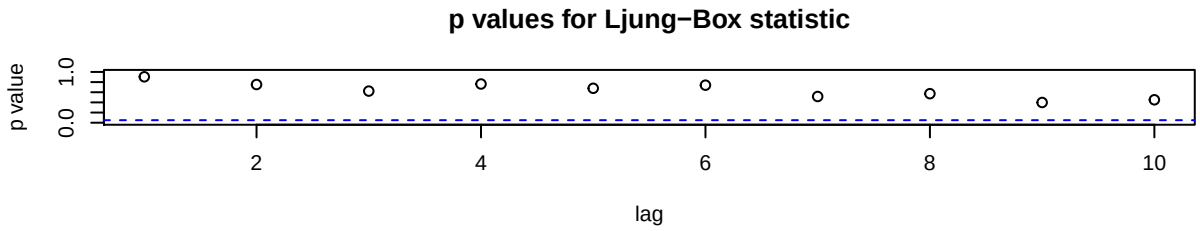

Standardized Residuals
−1
2
5
10
15
20
Time

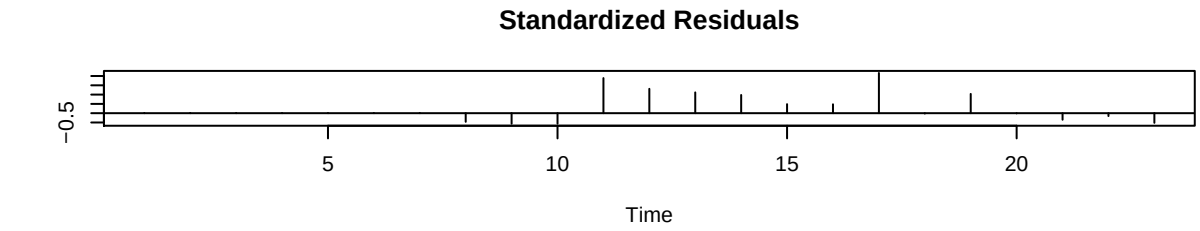

ACF of Residuals
ACF
−0.4
0
2
4
6
8
10
12
Lag

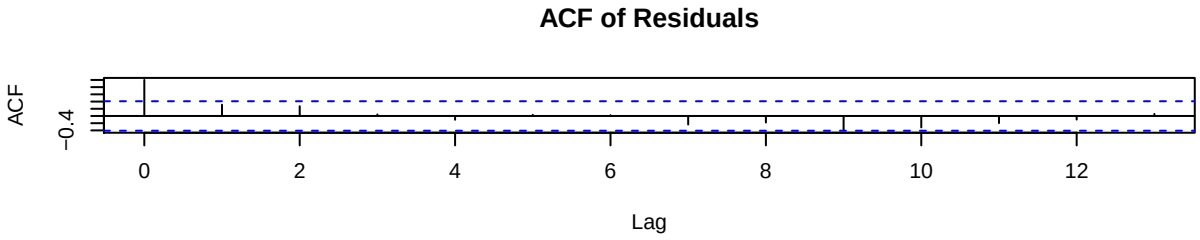

p values for Ljung−Box statistic
p value
0.0
1.0
2
4
6
8
10
lag

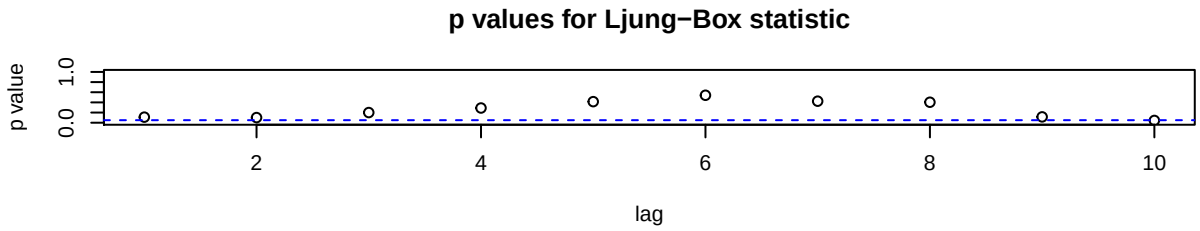

Standardized Residuals
−1
2
5
10
15
20
Time

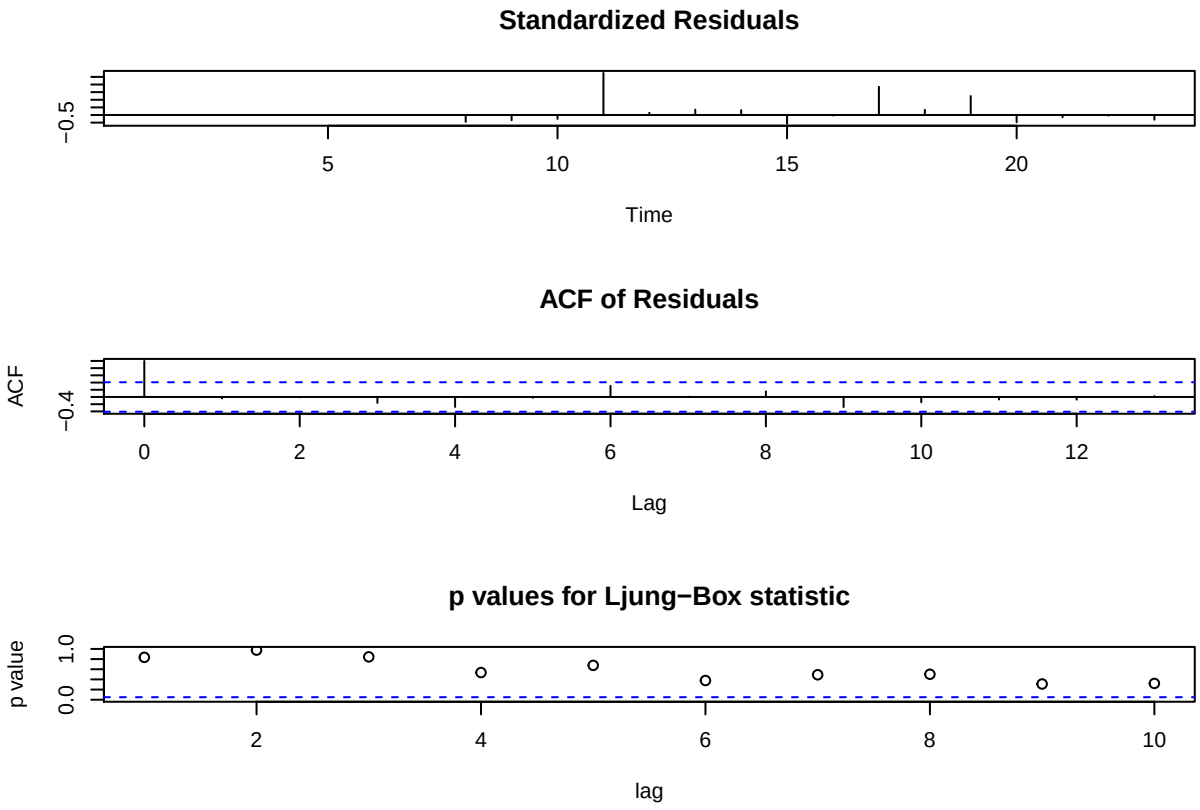

ACF of Residuals
ACF
−0.4
0
2
4
6
8
10
12
Lag


p values for Ljung−Box statistic
p value
0.0
1.0
2
4
6
8
10
lag


Standardized Residuals
−0.5
5
10
15
20
Time


ACF of Residuals
ACF
−0.4
0
2
4
6
8
10
12
Lag


p values for Ljung−Box statistic
p value
0.0
1.0
2
4
6
8
10
lag

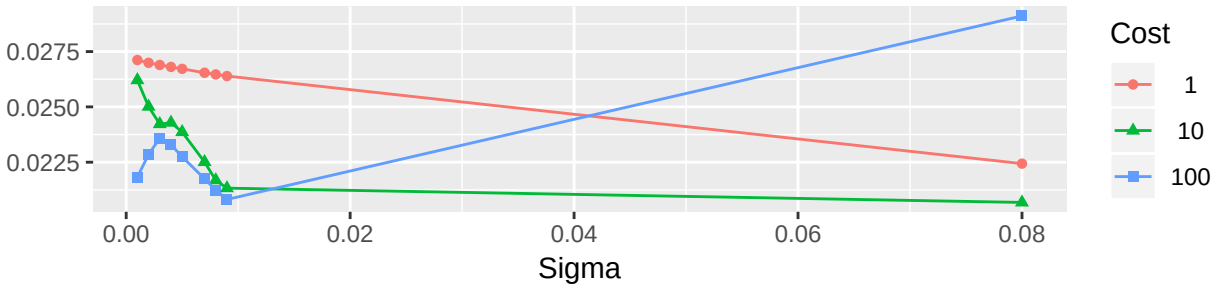


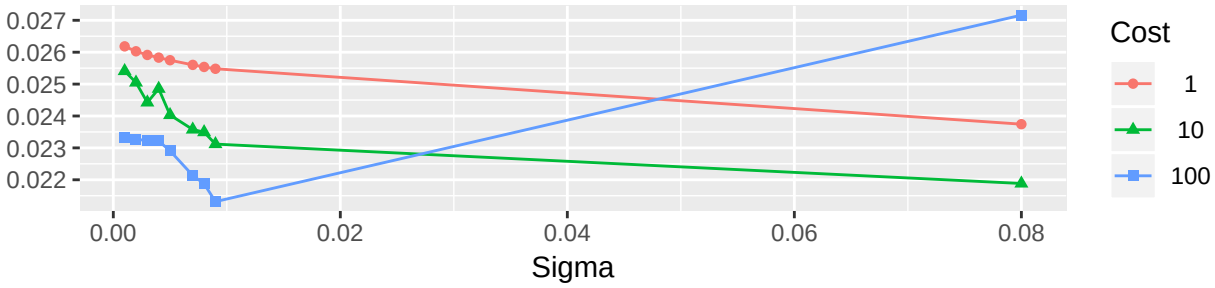
8.7 Appendix G

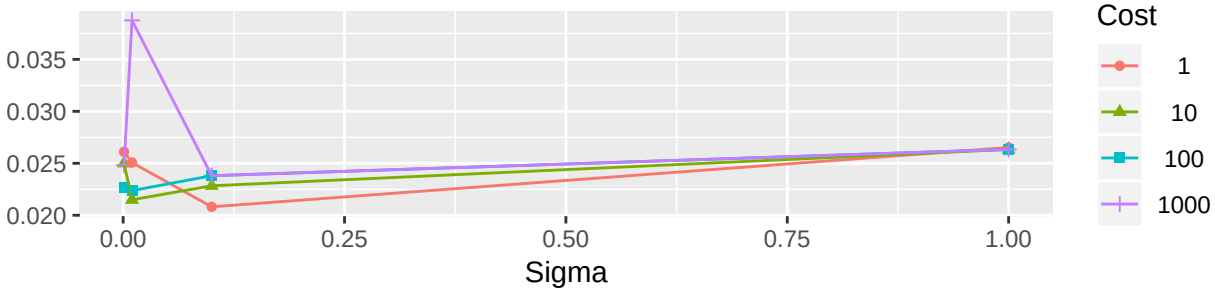
Results from other models

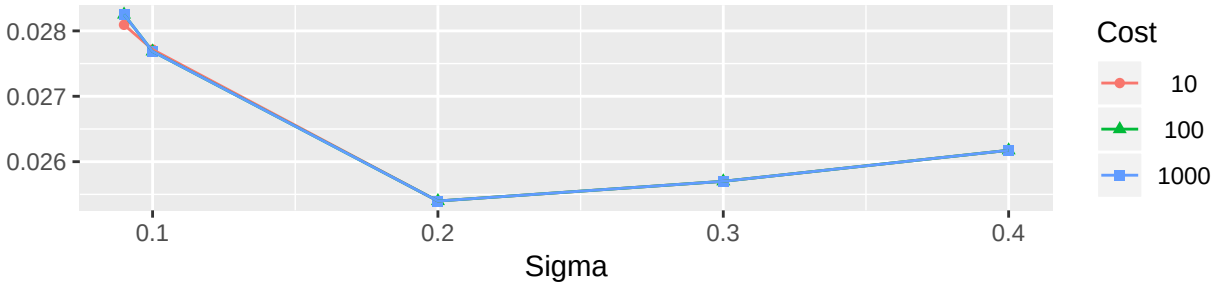
Table 3: . Results of other models for each embedding dimension

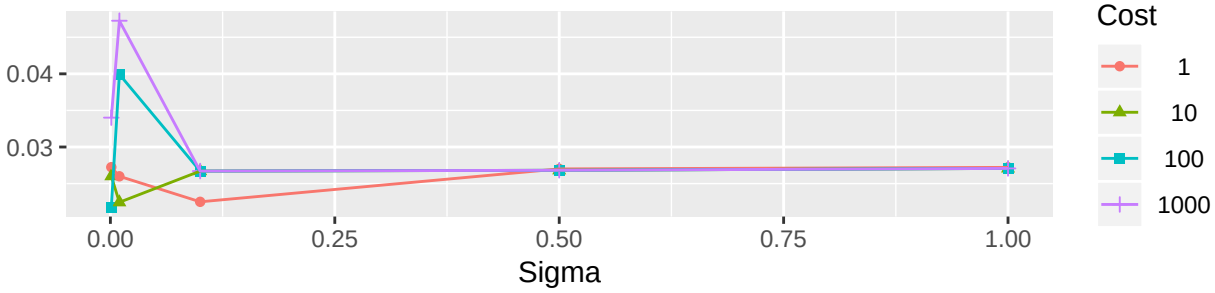
M	Sigma	C	Epsilon	Error	RMSE_Training	P_SV	RMSE_Test	MAE	Time
F2	0.009	100	0.1	0.4332898	0.0196287	90.00	0.0162521	0.0144950	1.42
G2	0.009	100	0.1	0.2892902	0.0213152	89.47	0.0138699	0.0105713	1.27
H1	0.100	1	0.1	0.3345778	0.0208098	94.44	0.0133728	0.0107891	0.98
I2	0.200	10	0.1	0.0100157	0.0249241	100.00	0.0110335	0.0092811	0.96
J1	0.100	1	0.1	0.3630428	0.0224470	87.50	0.0191821	0.0170332	1.14

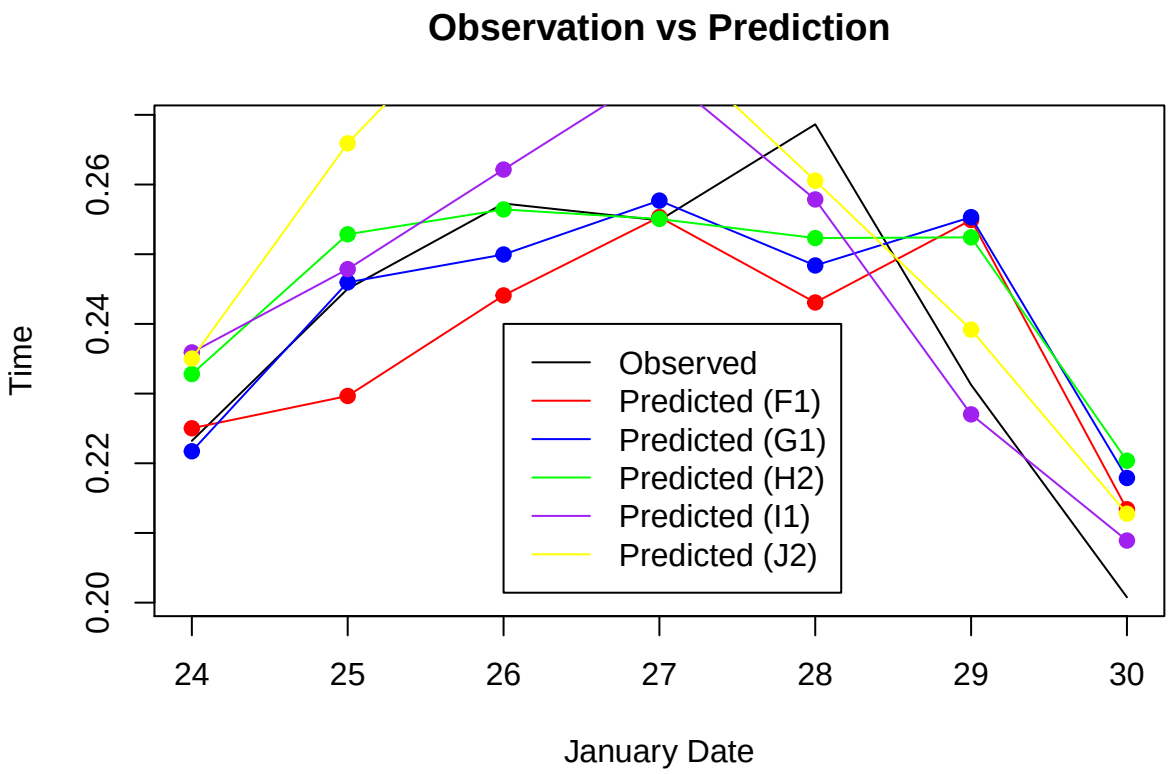

0.0275
0.0250
0.0225
0.00
0.02
0.04
0.06
0.08
Sigma
Cost
1
10
100

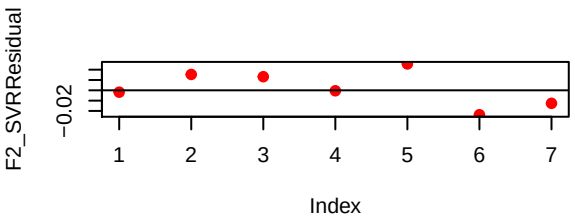

0.027
0.026
0.025
0.024
0.023
0.022
0.00
0.02
0.04
0.06
0.08
Sigma
Cost
1
10
100

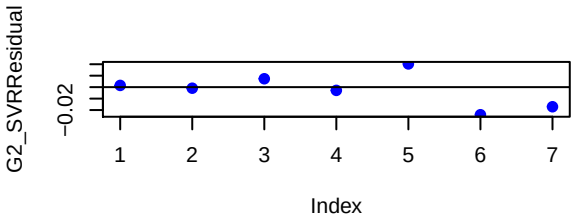

0.035
0.030
0.025
0.020
0.00
0.25
0.50
0.75
1.00
Sigma
Cost
1
10
100
1000

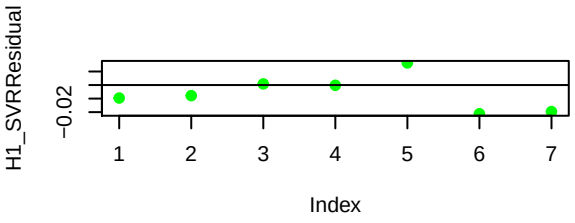

0.028
0.027
0.026
0.1
0.2
0.3
0.4
Sigma
Cost
10
100
1000

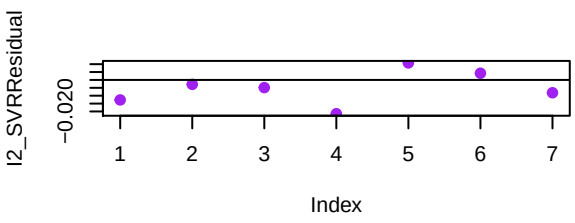

0.04
0.03
0.00
0.25
0.50
0.75
1.00
Sigma
Cost
1
10
100
1000

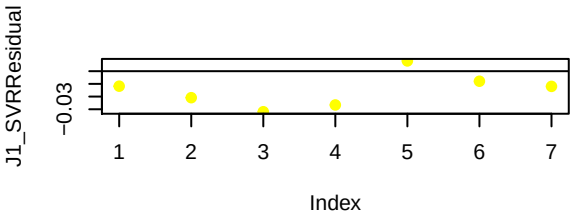

Observation vs Prediction
Time
0.20
0.22
0.24
0.26
24
25
26
27
28
29
30
January Date
Observed
Predicted (F1)
Predicted (G1)
Predicted (H2)
Predicted (I1)
Predicted (J2)

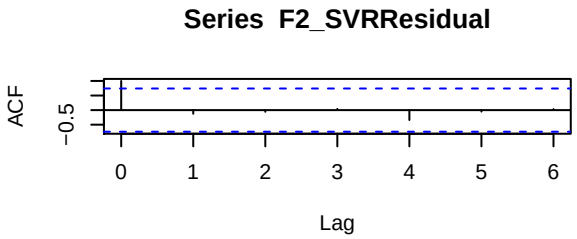

F2_SVRResidual
−0.02
1
2
3
4
5
6
7
Index

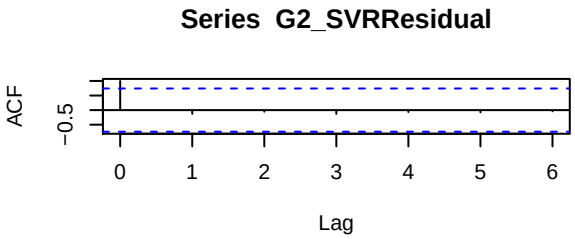

G2_SVRResidual
−0.02
1
2
3
4
5
6
7
Index

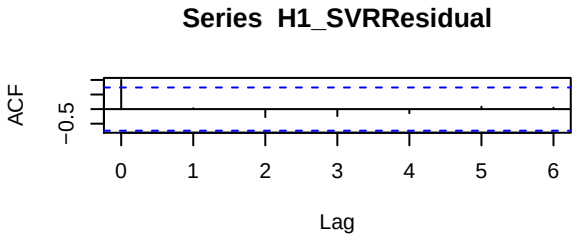

H1_SVRResidual
−0.02
1
2
3
4
5
6
7
Index

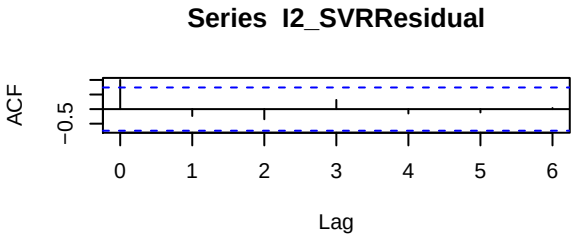

I2_SVRResidual
−0.020
1
2
3
4
5
6
7
Index

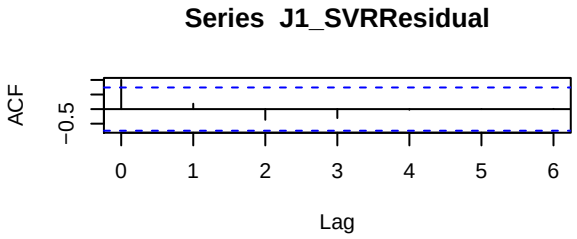

J1_SVRResidual
−0.03
1
2
3
4
5
6
7
Index

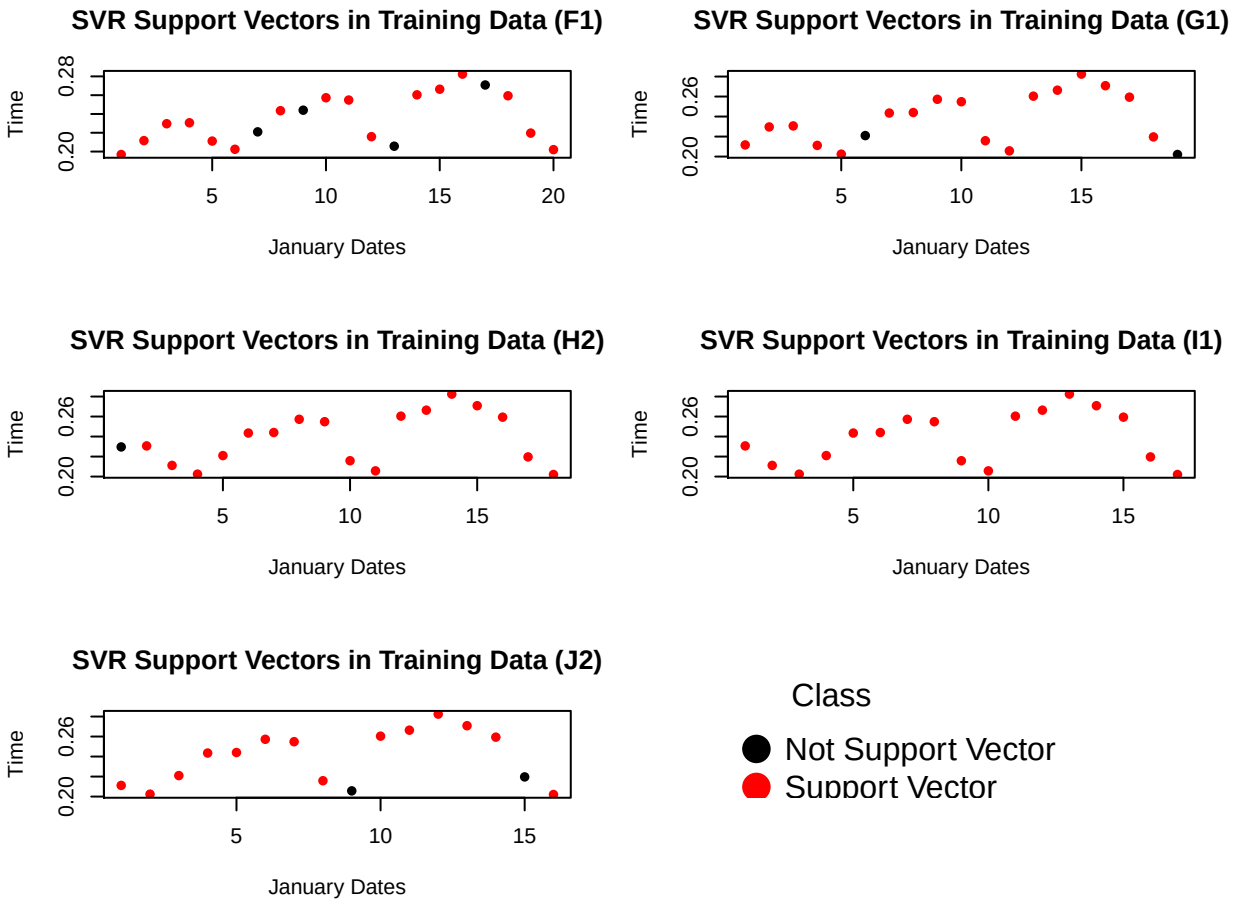

Series F2_SVRResidual
ACF
−0.5
0
1
2
3
4
5
6
Lag


Series G2_SVRResidual
ACF
−0.5
0
1
2
3
4
5
6
Lag


Series H1_SVRResidual
ACF
−0.5
0
1
2
3
4
5
6
Lag


Series I2_SVRResidual
ACF
−0.5
0
1
2
3
4
5
6
Lag


Series J1_SVRResidual
ACF
−0.5
0
1
2
3
4
5
6
Lag


SVR Support Vectors in Training Data (F1)
Time
0.20
0.28
5
10
15
20
January Dates
SVR Support Vectors in Training Data (G1)
Time
0.20
0.26
5
10
15
January Dates
SVR Support Vectors in Training Data (H2)
Time
0.20
0.26
5
10
15
January Dates
SVR Support Vectors in Training Data (I1)
Time
0.20
0.26
5
10
15
January Dates
SVR Support Vectors in Training Data (J2)
Time
0.20
0.26
5
10
15
January Dates
Class
Not Support Vector
Support Vector

References

Adams, Cathy. 2019. "London Drivers Spend 227 Hours Each Year in Traffic."

Boser, B., I. Guyon, and V. Vapnik. 1992. "A training algorithm for optimal margin classifiers." *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* 92: 144–52.

Dibike, Y, S Velickov, D Solomatine, and M Abbott. 2001. "Model Induction with Support Vector Machines: Introduction and Applications." *Journal of Computing in Civil Engineering* 15 (3): 208–16.

Dong, Hong-hui, Xiao-liang Sun, Li-min Jia, Haijian Li, and Yong Qin. 2012. "Traffic condition estimation with pre-selection space time model." *Journal of Central South University* 12 (19): 206–12. doi:10.1007/s11771-012-0993-6.

Gandhi, R. 2018. "Support Vector Machine: Introduction to Machine Learning Algorithms." https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

Global Alliance of SMEs. 2016. "What Makes London a Global City?"

Haworth, James. 2018. "Tutorial 3: Temporal and spatio-temporal modelling using ARIMA and STARIMA." In *Spatio-Temporal Analysis and Data Mining*.

Hong, Wei Chiang, Yucheng Dong, Feifeng Zheng, and Chien Yuan Lai. 2011. "Forecasting urban traffic flow by SVR with continuous ACO." *Applied Mathematical Modelling* 35 (3). Elsevier Inc.: 1282–91. doi:10.1016/j.apm.2010.09.005.

ITV. 2019. "Annual cost of traffic jams 'reaches £8 billion'." ITV. https://www.itv.com/news/2019-02-12/annual-cost-of-traffic-jams-reaches-8-billion/.

Kamarianakis, Yiannis, and Poulicos Prastacos. 2005. "Space-time modeling of traffic flow." *Computers and Geosciences* 31 (2): 119–33. doi:10.1016/j.cageo.2004.05.012.

Keerthi, S, and C Lin. 2003. "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel." *Neural Computation* 15 (7): 1667–89.

Kumar, S. Vasantha, and Lelitha Vanajakshi. 2015. "Short-term traffic flow prediction using seasonal ARIMA model with limited input data." *European Transport Research Review* 7 (3): 1–9. doi:10.1007/s12544-015-0170-8.

Park, Jinwoo, Syed M Raza, Pankaj Thorat, Dongsoo S Kim, and Hyunseung Choo. 2015. "Network Traffic Prediction Model Based on Training Data." In *Computational Science and Its Applications – Iccsa 2015*, edited by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Marina L Gavrilova, Ana Maria Alves Coutinho Rocha, Carmelo Torre, David Taniar, and Bernady O Apduhan, 117–27. Cham: Springer International Publishing.

Wang, Y., L. Li, and X. Xu. 2017. "A Piecewise Hybrid of ARIMA and SVMs for Short-Term Traffic Flow Prediction." *Neural Information Processing* 17 (10): 493–502.

Wesner, J. 2016. "MAE and RMSE - Which Metric is Better?" https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d.

Williams, Sophie. 2018. "London is second most congested city in Europe... and these are the roads motorists should avoid."