Name	Rabindra Kumar Panda
Student number	643998
Supervisor	Dr. Xuan Vinh Nguyen
Total number of credit points	75
Type of project	Research
Subject Code	COMP60002
Project title	Large scale real-time traffic flow prediction using SCATS volume data

Department of Computing and Information Systems
The University of Melbourne

Masters Thesis

Large scale real-time traffic flow prediction using SCATS volume data

Author:
Rabindra Kumar Panda

Supervisor:
Dr. Xuan Vinh Nguyen

*A thesis submitted in fulfilment of the requirements
for the degree of Master of Science in Computer Science*

June 2016

Declaration of Authorship

I , Rabindra Kumar Panda, certify that

- this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.
- where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the Department.
- the thesis is approximately 17000 words in length (excluding text in images, tables, bibliographies and appendices).

Signed: Rabindra Kumar Panda

Date: 06/06/2016

Abstract

Road traffic congestion is a global issue that results in significant wastage of time and resources. Rising population, urbanisation, growing economies and affordable personal vehicles aggravate the issue. Many urban cities have been trying to mitigate this by expanding and modernising the transportation infrastructure. Even though increasing the road capacity accommodates the travel demands, studies have shown this does not eliminate the congestion problem. Hence, since 1970's advanced traffic management systems have been used to address the issue of congestion. But for these systems to increase their operational efficiencies and fully realise their effectiveness, they need to have the predictive capabilities in the short term, usually ranging between few seconds to few hours. The research in short term traffic prediction has been active since the 1970's. Numerous models have been proposed to use the traffic data collected by inductive loop detectors for short term traffic prediction. Most of the works have shown promising results through experiments at particular locations, however we are still to find a robust and globally adaptable solution. In last decade the attention have shifted from theoretically well established parametric methods to non parametric data driven algorithms. This work is an extension to that.

Neural networks have always been one of the most capable mathematical models that can model complex non-linear relations. Up to 2006, their use have been hindered by practical issues related to the training. But recent breakthroughs in new ways of training deep neural architectures have made them reemerged as victors by realising the capabilities they had promised. In this thesis we study and extend their applications to short term traffic predictions. We applied three deep recurrent neural networks (Simple RNN, LSTM and GRU) in predicting the short term traffic volumes. The goal was to use both the temporal and spatial relationships that are present in the traffic flow data. We used these networks at univariate and multivariate settings to make predictions at single location and multiple locations respectively. For this work we used the volume data collected by VicRoads in Melbourne. We compared the results of our work with several existing methods and found promising results.

Acknowledgements

I would like to thank my supervisor Dr. Xuan Vinh Nguyen for his invaluable support and advice. His understanding of my situation and guidance helped me to complete this work. I would also like to thank my friends and colleagues who provided numerous helpful discussions.

Contents

List of Figures

List of Tables

Dedicated to my parents and teachers

Chapter 1

Introduction

"As a reader I loathe introductions...Introductions inhibit pleasure, they kill the joy of anticipation, they frustrate curiosity."

Harper Lee, *To Kill a Mockingbird* (1960)

1.1 Background

Predicting the future has always been a fascinating topic throughout the history of mankind. Instances of predicting the future through unconventional means have been mentioned in various forms of literature such as mythologies, fantasies, science fictions etc. Even today, while the means of prediction have changed, we still try to predict almost everything in our day to day lives - from election polls to sports outcomes to financial results.

Road traffic congestion is a serious global issue, resulting in significant wastage of time and resources. Several factors such as growth in population, urbanisation and affordable personal vehicles have aggravated the issue. In Australia the number of personal vehicles have grown from 1.4 million to 13 million during the period from 1955 to 2013, an average annual growth of 4%[1]. In 2012, the majority of Australians used personal vehicles, 72% to work or study and 88% for other activities. When it comes to travel time, across Sydney and Melbourne, the overall travel time is 37% and 29% more than the normal because of traffic congestions[2]. Across the globe, this figure is far worse in many other cities like Mexico City (59%), Bangkok (57%), Istanbul (50%), Rio de Janiro (47%) and

[1] Australian Bureau of Statistics - `http://www.abs.gov.au/AUSSTATS/abs@.nsf/Lookup/4102.0Main+Features40July+2013`

[2] TomTom Traffic Index - `http://www.tomtom.com/en_au/trafficindex/list`

Moscow (44%). While improving and extending the road infrastructure has reduced the issue to some extent, this is time consuming and does not eliminate the issue. Thus in last few decades, for better planning and control of road traffic, advanced traffic management systems have been deployed around the world. Still the role of these systems is not fully realised without predictive capabilities in the short term, without which these systems only react to events at real time. While this is the desired objective, the performance of these systems could be significantly improved by making them proactive (Smith and Demetsky [45]). Short term traffic flow prediction is not only helpful for these advanced traffic control systems, it is also useful for advanced traveller information systems.

Research in short term traffic prediction had been active for more than three decades. This shows the strong interest in solving the growing problem of traffic congestion by providing accurate predictions that can be used in both advanced traffic management systems and advanced traveller information systems. However due to the complex nature of traffic conditions, a globally applicable short term prediction model that can easily be embedded into advanced traffic management and advanced traveller information systems is yet to be found.

Neural networks have the potential to solve such complex problems, and it has been realised long term ago. But due to the practical issue that arise in training, their use have been restricted. In last decade new ways of training deep neural networks have made them reemerged as victors. Also the use of fast graphical processing units have reduced the training time by 10 to 20 times. The capabilities that neural networks had initially promised have been realised in several fields such as machine vision, speech recognitions and language modelling. In this study we extend their applications to the field of short term traffic predictions. We used several variants of deep recurrent neural networks to model the traffic conditions and make predictions at multiple steps ahead at a wider multi-location level. Our study shows some promising results and encourages similar future works.

1.2 Motivations

There are two main factors that have motivated us to undertake this study. These are

- Short term traffic prediction has more than three decades of active research, making this a challenging task. This is mainly due to the complex nature of traffic data - temporal and spatial relationships, noise and missing values, effects of non-recurrent events (weather, accidents, public events etc.)

- Availability of huge amount of historical data and new breakthroughs in deep learning.

1.3 Objectives and scope

Research objective is to use the large amount of available historical traffic volume data and apply deep neural networks for future predictions at a wider multi-location level. More importantly this research tries to answer the following questions -

- How can we use the large amount of traffic data available for predicting short term traffic flow for better predictions?
- Can we use the ability of deep neural networks to capture the tempo-spatial dimension of traffic conditions and make predictions at a transportation network level?

Research scope - The scope of this research is to predict traffic volume in the short term. The use of the phrase 'short term' implies that we are only interested in the predictions within a very short time horizon which typically ranges between few seconds to few hours in practice. The traffic parameters that are of usually of interest to be predicted are volume, time, speed and density. The scope of this research is limited to the prediction of only traffic volume. While doing so, we are only taking the past data into consideration and not taking the non-recurrent phenomena such as traffic accidents, weather or public events into consideration.

1.4 Thesis outline

Chapters – This thesis is divided into six chapters.

- Chapter 1: Introduction - In this chapter we present the background and research context, research objectives and scope.
- Chapter 2: Traffic Prediction: Literature Review - In this chapter we provide a reasonably thorough review of existing literature on short term traffic prediction.
- Chapter 3: SCATS Traffic Volume Data - In this chapter we describe the traffic volume data collected by VicRoads using the SCATS systems. Methods to deal with missing data are presented in this chapter. Finally we present some exploratory data analysis on the traffic data.

- Chapter 4: Deep Neural Networks for Short Term Traffic Prediction - In this chapter we present the details of deep neural networks with emphasis on Long Short Term Memory (LSTM) neural network.

- Chapter 5: Experiments and Results - In this chapter we conduct experiments using several existing methods in short term traffic prediction and three variants of deep recurrent neural networks. The results of these experiments are presented. Three accuracy measures were used to draw a comparison among the models.

- Chapter 6: Conclusions and Future Directions - In this chapter we conclude our thesis and provide inputs for future work.

Chapter 2

Traffic Prediction: Literature Review

“There is no way that we can predict the weather six months ahead beyond giving the seasonal average”

Stephen Hawking, *Black Holes and Baby Universes* (1993)

2.1 Introduction

In this chapter we provide an account of various elements involved in short term traffic prediction as a process and a reasonably complete review of existing literature. Research on short term traffic prediction has been active since 1970’s. Yet many professionals around the world still show a strong interest in this field. The simplest reason being the complex non-linear nature of traffic data and the effects of non-recurrent events (weather, public events, accidents etc.) on it. Critical reviews of existing literature on short term traffic flow have been presented in detail by Smith and Demetsky [45], Vlahogianni et al. [58], Van Lint and Van Hinsbergen [57] and Vlahogianni et al. [60]. The use of the phrase ’short term’ limits the scope of traffic prediction in terms of the prediction horizon which usually varies between few seconds to few hours depending upon the approach and application.

The process of short term traffic prediction consists of determining the scope, formulating the conceptual output specifications and model selection (Vlahogianni et al. [58]) as shown in figure 2.1.

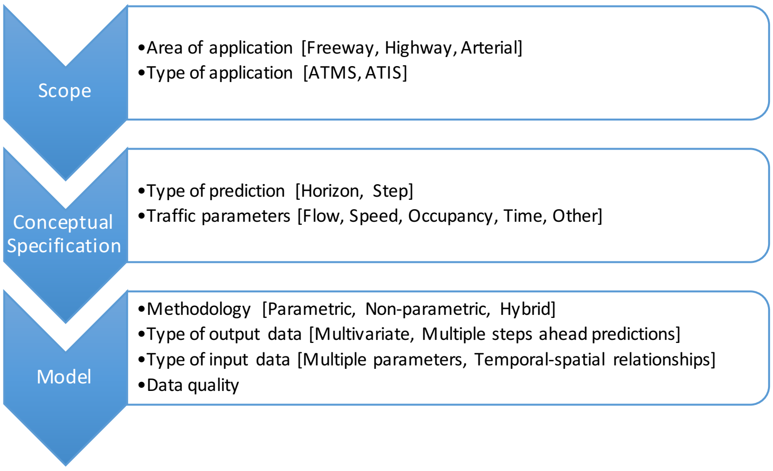

Scope
•Area of application [Freeway, Highway, Arterial]
•Type of application [ATMS, ATIS]
Conceptual Specification
•Type of prediction [Horizon, Step]
•Traffic parameters [Flow, Speed, Occupancy, Time, Other]
Model
•Methodology [Parametric, Non-parametric, Hybrid]
•Type of output data [Multivariate, Multiple steps ahead predictions]
•Type of input data [Multiple parameters, Temporal-spatial relationships]
•Data quality

Figure 2.1: Elements of short term traffic prediction

Determining whether the prediction model to be developed is going to be part of an advanced traffic management system or advanced traveller information system is important. This decision is influenced by other elements such as type of road and traffic parameters involved. However, a suitable forecasting method that can be easily integrated into these systems is still elusive.

The type of area influences the prediction process. Short term traffic predictions can be done for highway, freeway and urban arterial roads. Most of the existing works focus on either highway or freeway traffic. The reason being predicting traffic conditions at a urban setting is more complex. While predicting traffic conditions at highways and freeways are important for both advanced traffic management systems and advanced traveller information systems, for urban settings the need for short term traffic predictions is more relevant for signal control at intersections.

The traffic parameters that are predicted can be - flow (number of vehicles per hour), time (minutes to travel between two points), speed (mean speed in km/hour) and density (number of vehicles per km). Relevance of flow is more stable and important than other parameters as per Levin and Tsao [33]. However this is conflicting and many authors have argued otherwise. Dougherty and Cobbett [17] attempted to determine the parameter that best describes the traffic conditions and their findings suggested that flow and density are more relevant than speed. Predicting travel time has also been the focus of many works, especially in recent years. This is because of its importance when it comes to advanced traveller information systems, while flow and density are more important for advanced traffic management systems. While a lot of earlier works have gone into predicting one of the parameters, there have been few attempts to predict a combination of traffic parameters. There are also non-recurrent events such as accidents, severe weather, public events etc. that affect traffic conditions. Not many attempts have been made to include these as input parameters while creating a forecasting model. This is because the influence of these have been found to be ambiguous. Recently Tsirigotis et al. [53] studied the influence of weather on the performance of traffic forecasting models. The authors observed a very marginal improvement in the performance with the inclusion of weather data as an exogenous variable.

Forecasting horizon and step are also important and they show the stability of the method. It is worth noticing that forecast horizon is the total amount of time we are trying to predict ahead while the number of steps depends on the frequency of traffic data. For instance with data collected at 5 minutes interval, we may want to predict the traffic flow in next 15 minutes, where this is the forecast horizon, the forecast step is 3. By intuition we can understand that the accuracy will suffer as we expand the forecast horizon. The *Highway Capacity Manual* (2000) suggests that the best forecast horizon for short term traffic prediction is 15 minutes. Yue et al. [66] empirically examined the relationship among prediction horizon, the effectiveness of real-time data and traffic predictability. They concluded that the choice of a model can be affected by the understanding of this relationship.

One of the other aspects to consider is the importance of spatial relationships in traffic flow data along with its temporal characteristics. The reason being that this fully captures the dynamics of traffic conditions and is a more accurate representation of traffic flow characteristics. Several attempts have been made to incorporate traffic flow from upstream, downstream and adjacent locations to increase the prediction accuracy. Cheng et al. [11] provides an extensive examination of the spatiotemporal characteristics of traffic data and an exploratory autocorrelation analysis.

Selecting the right model for short term traffic prediction is a challenging task. A number of models have been suggested and yet there is no consensus on a globally acceptable method. The various methods that have been suggested for short term traffic prediction can be categorised into four groups - naïve, parametric, non-parametric and hybrid as shown in figure 2.2.

In the following sections, we review significant amount of earlier works in this field, grouped by the type of method.

2.2 Naïve methods

These are heuristics methods, and often used in practice because of their simplicity and the ease of implementations. In most cases these methods are used as baselines for

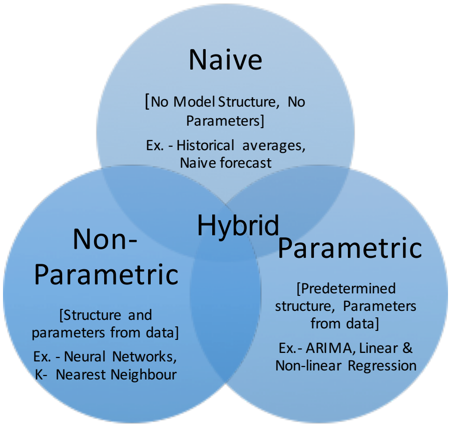

Naive
[No Model Structure, No Parameters]
Ex. - Historical averages, Naive forecast
Hybrid
Non-Parametric
[Structure and parameters from data]
Ex. - Neural Networks, K- Nearest Neighbour
Parametric
[Predetermined structure, Parameters from data]
Ex.- ARIMA, Linear & Non-linear Regression

Figure 2.2: Methods in short term traffic prediction

comparison while creating more advanced methods. We briefly present these methods here.

2.2.1 The Naïve method

The simplest naive approach in short term prediction would be to take the last observed value and this involves no computational effort. Formally, at any time t the prediction is given as

$$\hat{x}_t = x_{t-1} \tag{2.1}$$

Another variant of this method is known as seasonal Naïve method, where the estimate at any time t is last observed value from the same season of the year. This is mainly used for highly seasonal time series data.

2.2.2 Average method

Another simple heuristic method known as the historical averages uses the average of past observed values. We define this as

$$\hat{x}_t = (x_{t-1} + x_{t-2} ... + x_{t-n})/n \tag{2.2}$$

2.3 Parametric methods

In parametric models, we estimate the parameters from the training dataset to determine the function that make predictions for new unseen data. The number of parameters are fixed. The advantage of parametric models are that these perform quite well in situations where large amount of data is not available. Some of the typical examples of parametric models include Linear and nonlinear regressions, ARIMA models, Kalman filters etc.

2.3.1 Classical regression

In machine learning and statistical applications, the use of linear models are predominant. These models are also important in time series domains such as traffic flow prediction. The primary idea behind the regression is to express the output variable as a linear combination of input vectors. We can express the linear regression in time series as an output influenced by a collection of inputs, where the inputs could possibly be an independent series

$$x_t = \beta_1 z_{t1} + \beta_2 z_{t2} + ... + \beta_q z_{tq} + w_t \tag{2.3}$$

where $\beta_1, \beta_2, ..., \beta_q$ are unknown regression coefficients and w_t is a random error.

A very few attempts have been made to model the traffic conditions using linear and non-linear models of regression in the field of short term traffic prediction. The simplest reason being the inability of these statistical methods to capture the highly nonlinear and complex relationships that are present in the traffic data. Low [34] and Jensen and Nielsen [26] used linear models of regression for predicting traffic volumes while Högberg [23] used nonlinear regression for traffic prediction. Rice and Van Zwet [42] revisited the application of linear regression to short term traffic prediction. They used this approach to predict travel time on freeways. For their method, they created a matrix V with entries $V(d, l, t)$ which denotes the measured velocity on day d at loop l at time t. Using this matrix they calculated $TT_d(a, b, t)$, which is the time it took to travel from a to b at time t on a day d. They calculated another proxy value for these times, which would have been the travel time if traffic conditions were remained constant, this was denoted as $T_d^*(a, b, t)$. Using empirical analysis they observed a linear relationship between $T^*(t)$ and $TT(t + \delta)$ and proposed a linear regression model to capture this relationship.

Lan and Miaou [32] proposed a recursive algorithm by using a dynamic generalised linear model in this context to predict traffic flow. In this work a negative binomial probability distribution was chosen. The flow data used in this experiment was obtained using a video camera and counted manually and hence a very small set of data, containing only 139 observations at 20 seconds interval, was used for model evaluation.

2.3.2 ARIMA

ARIMA (Auto Regressive Integrated Moving Average) is a class of parametric regression models. In this section we will introduce ARIMA and related methods such as moving averages and autoregressive. For an in depth understanding of these methods, the reader is encouraged to refer to to Tong [52], Brockwell and Davis [6] and Box et al. [5]. It is important to understand that ARIMA modelling works only with stationary time series data. A stationary time series is one whose properties do not depend on the time it is being observed at. Trends and seasonality affect time series and hence make it non-stationary. Although this seems as a big restriction, in short term traffic prediction, ARIMA models have been very successful. Two basic models constitute ARIMA models - AR (autoregressive) and MA (moving average).

The main idea behind autoregressive models is that past values affect the present value, i.e. x_t can be expressed as a function of past p values $x_{t-1}, x_{t-2}, ..., x_{t-p}$, where p is the number of steps into the past. We can express an autoregressive model of order p as below

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + w_t \tag{2.4}$$

where x_t is stationary and $\phi_1, \phi_2, ..., \phi_p$ are constant parameters that are to be chosen. We have added the term w_t as a Gaussian white noise with zero mean and variance σ_w^2.

In the MA model, the current value is dependent on the last q one-step forecast errors $e_{t-1}, e_{t-2}, ..., e_{t-q}$ and the white noise w_t. The expression for moving average is

$$x_t = -\theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} + w_t \tag{2.5}$$

$\theta_1, \theta_2, ..., \theta_q$ are the parameters to be chosen.

Now proceeding to an ARMA (autoregressive moving average) model, we define an ARMA(p,q) model where the present value x_t is dependent on p past recent values and q past recent forecast errors and a white noise w_t.

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} + w_t \tag{2.6}$$

When q is 0, the model becomes an autoregressive model of order p, AR(p) and when p is 0 the model is a moving average of order q, MA(q). We can rewrite 2.6 by using the backshift operator B^α, which is defined as $B^\alpha z_t = z_{t-\alpha}$,

$$\phi(B)x_t = \theta(B)e_t \tag{2.7}$$

where

$$\phi(z) = 1 - \phi_1 z - ... - \phi_p z^p \tag{2.8}$$

$$\theta(z) = 1 - \theta_1 z - ... - \theta_q z^q \tag{2.9}$$

In practice, most time series data are non-stationary and so several approaches, for instance differencing, are used to make it stationary before applying the ARMA(p,q) model. By combining differencing with autoregressive and moving averages, we obtain the ARIMA model which is defined as below

$$x'_t = \phi_1 x'_{t-1} + \phi_2 x'_{t-2} + ... + \phi_p x'_{t-p} - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} + w_t \tag{2.10}$$

where x'_t is the differenced series. Formally the model is denoted as ARIMA(p,d,q) where p is the order of autoregressive part, d is the degree of differencing and q is the order of moving average. This is also known as a non-seasonal ARIMA model.

The common method used to determine the parameters in an ARIMA(p,d,q) model is known as the Box-Jenkins approach (Box et al. [5]) which is a three stage procedure. The three stages are identification, estimation and diagnostic checking. At the identification stage, the values p, d and q are determined by observing the autocorrelation and partial autocorrelation functions of the time series and its differences. At the estimation stage, the maximum likelihood estimates are determined for each model parameter. Finally in the diagnostics stage, the residuals are analysed and model comparisons are done. If the model fits well then the standardised residuals behave as an i.i.d. with mean zero and variance one.

Ahmed and Cook [2] used Box-Jenkins method for short-term traffic forecast. The input data used was 166 sets of time series traffic data collected by freeway traffic surveillance systems in three locations - Los Angeles, Minneapolis and Detroit. The authors concluded an ARIMA(0,1,3) model, based on the autocorrelation and partial autocorrelation functions, as a reasonable fit for short term predictions of both traffic volume and occupancy. The model performance was evaluated against a moving average, a double smoothing average and a Trigg and Leach adaptive model. The comparisons suggest that the ARIMA model had better accuracy than the others. The authors used this model in detecting traffic incidents by comparing the real-time flow occupancy with the predicted value. Nihan and Holmesland [40] used the Box-Jenkins technique on monthly data collected at 15 minutes interval on a freeway segment from 1968 to 1976 to forecast for the year 1977. After examining several models they finally selected an ARIMA(12,1,7) model. The forecast was done for average weekday volume with positive results. Hamed et al. [18] also studied the application of ARIMA model in short term traffic volume prediction. They found a simple ARIMA(0,1,1) model to be adequate for modelling the traffic data. The used a 1-min interval dataset collected in five urban arterials.

Williams [61] used an ARIMAX model to use upstream traffic data along with the predicting location's traffic data while estimating the parameters of the ARIMA model. This was done using ARIMAX model which is an extension of the ARIMA model where an exogenous variable is used. The data was collected form four locations near Beaune, France. The data from three upstream locations were used for forecasting at the fourth location in Beaune. The same data were used in the proposed ATHENA (Danech-Pajouh and Aron [14]) and KARIMA (Van Der Voort et al. [54]) models. The model was compared against a univariate ARIMA model. The results show that the ARIMAX model consistently outperformed the ARIMA model. However the complexity of the ARIMAX model was more than the ARIMA model with as many as twice the parameters to estimate. Also in case of missing values the ARIMAX model performance degraded more than the ARIMA model.

Min et al. [37] proposed a dynamic Space Time ARIMA (STARIMA) model for short term traffic prediction. Their argument for the new proposed model was based on the factor that most of the existed model failed to take the spatial information of the transportation system into account. The proposed dynamic STARIMA model combined STARIMA and Dynamic Turn Ratio Prediction (DTRP) models. Using DTRP they dynamically updated the static matrix W_k in STARIMA model that contains the structural information of the transportation network. The results of the study showed significant improvement in forecast accuracy. The authors later published another similar work (Min et al. [38]) that used the generalised STARIMA (GSTARIMA) model. The authors presented the results where this model has a small improvements over the STARIMA model. However the major drawbacks of the GSTARIMA model is the estimation of large number of parameters which significantly increases the computational time. It also suffers in performance if enough historical data is not available.

Williams and Hoel [62] proposed for the acceptance of seasonal ARIMA models for short term traffic prediction. A seasonal ARIMA $(p, d, q)(P, D, Q)_s$ for a time series x_t

is one where s is the period, d and D are nonnegative integers. The time series theorem known as the World decomposition is used as the theoretical justification of applying seasonal ARIMA model to univariate time series with stationarity. Data from two freeway locations, one each from the United States and the United Kingdom were used for evaluating the model. The performance of the models were compared against three heuristics approaches - historical averages, random walk and deviation from historical averages. The results show that for both the locations the seasonal ARIMA has better performance than the three methods mentioned earlier. However the authors did not present whether a non-seasonal ARIMA model would have similar performance. The only other model that was considered for comparison was the KARIMA model, which did not perform as good as the seasonal ARIMA model. Kumar and Vanajakshi [31] also used a seasonal ARIMA in the context of limited data for short term traffic prediction. They used data collected over three days from an arterial road in Chennai, India for the study. The model was validated on 24 hours ahead forecast. Their results were positive when compared with historical averages and naïve methods. They argued when availability of large traffic dataset is a constraint seasonal ARIMA method is a better choice.

The major deficiency of the ARIMA models is that they do not take the extremes into consideration and focus on the means. This is in contrast to the nature of the traffic data. ARIMA models are also have the inability to perform well with missing data, as pointed out by Smith and Demetsky [45].

2.3.3 Kalman filter

Kalman filter is a parametric regression technique usually used in the field of automatic control systems and signal processing. It was proposed by Kalman [28]. It can be used to model both stationary and non-stationary time series. We present a brief description of this theory, for detail understanding the reader should refer to some extensive literature (such as Harvey [19] and Haykin [20]). It is important to realise that Kalman filer and state space model refer to the same basic theory.

The Kalman filter solves the problem of sequential state estimation of a dynamic linear system, where in such a system the state evolution and the measurements are both linear and Gaussian. Let us consider a state space model of the form

$$x_n = P_n x_{n-1} + \tau_n \tag{2.11}$$

$$y_n = Q_n y_{n-1} + \upsilon_n \tag{2.12}$$

where, x_n and y_n are the state and measurement respectively at time step n. P_n is a $N \times N$ state transition matrix and τ_n is a $N \times 1$ Gaussian random state noise vector with zero mean and covariance matrix R_n. Q_n is a $M \times N$ measurement matrix and υ_n is a $M \times 1$ Gaussian random measurement noise vector with zero mean and covariance matrix S_n.

In this state-space setting, the two important tasks are - *filtering* and *prediction*. The filtering problem is to estimate the state x_n given the set of measurements $Y_n = y_1, y_2, ..., y_n$. And the prediction problem is to predict x_{n+t}, that is the state after t time steps, given the set of measurements Y_n. The Kalman filter algorithm can be described using the below equations.

1. Prediction step

$$m_{n|n-1} = P_n m_{n-1|n-1} \tag{2.13}$$

$$C_{n|n-1} = P_n C_{n-1|n-1} P_n^T + R_n \tag{2.14}$$

2. Update step to estimate $\hat{x}_n = m_{n|n}$

$$J_n = Q_n C_{n|n-1} Q_n^T + S_n \tag{2.15}$$

$$K_n = C_{n|n-1} Q_n^T J_n^{-1} \tag{2.16}$$

$$m_{n|n} = m_{n|n-1} + K_n(y_n - Q_n m_{n|n-1}) \tag{2.17}$$

$$C_{n|n} = C_{n|n-1} - K_n Q_n C_{n|n-1} \tag{2.18}$$

where $m_{n|n}$ and $C_{n|n}$ are the Gaussian mean and covariance of state x_n at time step n, in the posterior probability distributed function

$$p(x_n|Y_n) \equiv \mathcal{N}(x_n; m_{n|n}, C_{n|n}) \tag{2.19}$$

The subscript notation $n|n$ denotes the recursive computation of the pdf of the state x_n at step n using the measurements up to time step n.

Okutani and Stephanedes [41] proposed two models using Kalman filtering for short term traffic volume predictions. The predictions of volume on a link were done using not only the data from that link but also from adjacent links. They found, using these models, the average error rate to be around 9%. This shows the ability of Kalman filtering to predict in a multivariate setting which is difficult in other statistical regression models

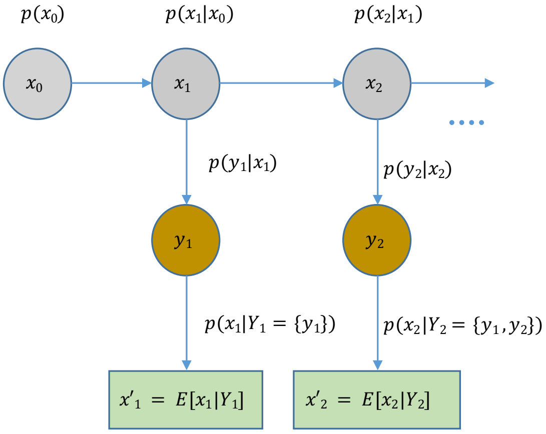


FIGURE 2.3: Sequential Bayesian estimation in Kalman filtering, recursively computes the posterior probability $p(x_n|Y_n)$

such as ARIMA. Stathopoulos and Karlaftis [49] used traffic data collected at 3-minutes interval from urban arterials to develop a multivariate time-series state space model. For prediction of traffic flow at one location, both the data at that location and from upstream were considered. The results of this model were compared against an ARIMA model. The model performance was found to be superior than the ARIMA model. For one of the locations, the mean absolute percentage error (MAPE) of the state space model was 12%, compared to the 20% MAPE value of the ARIMA model. For other locations the differences were not observed to be this large. The authors also concluded that short term traffic flow prediction at urban arterials is a very difficult task and can not be as accurate as the predictions at freeways.

2.4 Non-Parametric methods

In non-parametric methods the parameters are not fixed, and vary with the amount of data available. Usually more data is required for these models than parametric methods. However, the main advantage of these methods is that they can model complex non-linear data significantly better than the parametric methods. Some of the widely used non-parametric models are - k-Nearest Neighbour, Support Vector Machines and Neural Networks

2.4.1 K-nearest neighbour

K-nearest neighbour is a non-parametric regression method. In this approach the basic concept is based on the idea of a phase space embedding and finding a neighbourhood in that. A phase space or state space is a vector space that represents the state of a purely deterministic system. The points in this phase space show the dynamics of such a system. For a time series, the problem of constructing a phase space is known as phase space embedding. Given a time series $(x_1, ...x_t)$ with scalar values, in order to forecast x_{t+1}, a suitable value k is used which can be determined heuristically by using an error measure such as RMSE.. The forecast algorithm then predicts the value by taking either a simple arithmetic mean of k neighbours or a weighted average with respect to the distance. The distance functions can be Euclidean, Manhattan, Minkowski etc.

Davis and Nihan [15] were one of the first to show the application of a k-NN method in short term traffic flow prediction. The authors used a minute aggregated traffic flow and occupancy data collected at an intersection in an interstate highway. However a detail account of the results was missing in the work, this could be because of the lack of data, as the authors used only about one and half hour of data for this experiment.

The performance of a k-NN algorithm in short term traffic prediction was further demonstrated by Smith and Demetsky [44]. They used k-nearest neighbour with k value of 10. For comparison purpose they used a back propagation neural network model with one hidden layer. From their results, they showed that nearest neighbour was more effective than the neural network model. They also argued in favour of the usage of the nearest neighbour methods because nearest neighbour methods are simple to understand by practitioners and easy to implement. Another application of k-nearest neighbour was by Lv et al. [35] to predict highway traffic accidents. They used data collected from inductive loop detectors along with historical traffic accident data. As per the authors knowledge this was the first time this approach was used to identify traffic accidents. The changes between normal traffic conditions and hazardous conditions were represented as traffic accident precursors. Using these precursors, a 5-nearest neighbour method was proposed to identify hazardous conditions.

The advantages of using a non-parametric regression such as K-NN in short term traffic prediction are their simplicity in modelling multivariate data, independence of the assumption on the state transitions of the traffic conditions and intuitive model formulation (Vlahogianni et al. [58]).

2.4.2 Neural networks

Artificial Neural Networks (ANN) were mathematical models (McCulloch and Pitts [36], Rosenblatt [43]) designed to provide a representation of how the human brain works. It is now obvious that these mathematical models bear little resemblance to the structure of brain, yet they have been hugely successful, especially recently. Because they were initially inspired by the biological brain, the term neural is associated with such kind of mathematical models. A basic artificial neural network consists of a set of nodes connected by edges with weights. We can say that the nodes represent the biological neurons and the edges represent the synapses. The connections among the nodes can be cyclic or acyclic. The former is known as a feedforward neural network and the later as a recurrent network. We describe about these neural networks in more details in chapter 4. Several variations of artificial neural networks have been used in short term traffic prediction. Some well known examples include - *Multilayer perceptrons, Radial basis function networks, Kohnen maps* and *Hopfield networks.*

Clark et al. [13] made a comparison of neural networks and ARIMA models in an urban setting and found only a slight difference between their performances. Dougherty and Cobbett [17] applied a back propagation feedforward neural network to predict flow, occupancy and speed traffic parameters. They found the prediction of speed to be disappointing. The results for the predictions of flow and occupancy, although were not outstanding, showed some promise for further work in this area. Kirby et al. [30] extended the work of Clark et al. [13] and compared a neural network model with the ATHENA and ARIMA models. They concluded that the neural networks performed worse than the ARIMA model for 30 and 60 minutes prediction horizons. However they argued that the neural networks are by nature the most suited models to fit the traffic characteristics than the statistical time series methods. Yasdi [64] used a Jordan neural network for traffic volume predictions. The authors made forecasts for weekly, daily and hourly traffic volumes. For their work they used data collected from traffic loop inductors and aggregated at fifteen minutes interval. The data was then further classified based on events and stored in a knowledge base for reference. The results were exceptional with an mean squared error (MSE) of less than 0.003.

Dia [16] used a time-lag recurrent network (TLRN) to predict traffic speed for fifteen minutes horizon. They performed their experiment using data collected from a section of the Pacific Highway between Brisbane and Gold Coast in Queensland, Australia. Unlike previous mentioned studies the authors used an object-oriented dynamic neural network model. The dataset they used consisted of 5000 observations at 20 seconds of interval collected over five hour periods on two days. Their results show that the model had an accuracy of 90-94% for 5 minutes predictions. The accuracies dropped to 84% and 80% for 10 and 15 minutes prediction horizons respectively. Chen and Grant-Muller [9] applied a dynamic neural network model in this context, which was based on a resource allocating network (RAN). The RAN is a single hidden layer neural network with no initial hidden units. The hidden units were added dynamically and the number of hidden units corresponded to the complexity of the mapped function. A maximum number of hidden units were set to 30 in this study. They used their model on motorway data collected on normal and incident related conditions. Their results showed that the performance of the dynamic model was better than the static model. Using five hidden units the model was able to achive a MAPE of approximately 9.5% compared to the 11% MAPE of the static model.

Innamaa [25] applied a feedforward MLP to predict travel time in an interurban highway. They used data collected over a period of four months in a highway in southern Finland. The neural network implemented for the experiment was very simple with one hidden layer and at most 20 units in the hidden layer. Also they used separate neural networks for each sub-link to predict the average travel time, thus in practice these are unrealistic to be implemented due to increased complexity. On an average they achieved an accuracy of 90%. They suggested inclusion of flow information could have been beneficial. Jiang and Adeli [27] used a nonparametric dynamic time-delay recurrent wavelet neural network model for forecasting traffic flow. They suggested that this model can be used for both the short term and long term (from a day to a month) traffic flow forecasting. They used a limited dataset and showed the results to be within 10% error rate.

Van Lint et al. [56] studied the robustness of a state-space neural network (SSNN) to predict travel time under missing data. The hidden layer had 12 units. The data used for experiment was corrupted to contain both incidental and structural input failures. It was observed that the SSNN model was insensitive to these corruptions. The model was compared with the instantaneous predictor. The model outperformed the instantaneous predictor by a large margin. The authors concluded that, in order to be practically applicable, a prediction model should be robust by not being affected by missing or corrupt data. They also suggested simple imputation schemes such as spatial interpolation or exponential forecasting to be used to handle missing data.

Neural networks are very powerful not only because of their good predictive ability but also due to their robustness to missing and corrupt data and better modelling capability of the traffic conditions with good overall performance. They are also capable of making better multi-step ahead predictions than other mentioned methods.

2.4.3 Support vector machine

Since its inception, Support Vector Machines (SVM) have been very popular and widely used in a range of classification and regression applications. This is due to their greater generalisation ability. The basic idea of SVM is to map the data into a higher dimensional feature space and use a separating hyperplane to classify the data. For regression problems, a version of SVM was proposed by Smola and Vapnik [47] and is known as support vector regression. Similar to SVM, the SVR uses a small subset of training data, because the cost function for the model ignores any data that is close to the model prediction. Another version of SVM to solve regression problem is Least Square Support Vector Machine (LS-SVM) proposed by Suykens et al. [50]

In time series domain the use of SVR has been shown some promise, hence it was inevitable that its application to traffic forecasting problem would remain untested. Wu et al. [63] used an SVR model to predict travel time in a highway. They compared their model to a historical averages algorithm and found better accuracies. However the authors did not present any reason of not comparing the model with a stronger model such as neural networks. Zeng et al. [67] proposed an online accurate support vector regressor (AOSVR) to increase the time efficiency of traffic flow predictions. They first created an SVR model with Gaussian kernel and trained using an LS-SVM algorithm. With the availability of new data the model parameters were updated online. A comparison of this model with a simple neural network, with one hidden layer with 8 units, and a historical averages method was done. Their results showed that the AOSVR model performed better than the other methods.

2.4.4 Bayesian networks

A Bayesian network is a probabilistic graphical model, that uses a directed acyclic graph (DAG) to represent a set of random variables and their conditional probabilities. In such a DAG the nodes correspond to random variables and the edges correspond to the conditional probabilities. We can see that the edges in such a DAG represent a direct causal influence and the belief in such a causal structured network changes with new evidence. Formally we can define a Bayesian network as a pair (G, P), where G is a DAG with a set of nodes X and $P = \{p(x_1|\pi_1), ..., p(x_n|\pi_n)\}$ is a set of conditional probabilities with π_i is the set of parent nodes of node x_i. The joint probability of all nodes is defined as

$$p(X) = \prod_{i=1}^{n} p(x_i|\pi_i) \tag{2.20}$$

The joint probability distribution for a Gaussian Bayesian network is a multivariate normal distribution $N(\mu, \Sigma)$, which is defined as

$$f(x) = (2\pi)^{-n/2}|\Sigma|^{-1/2}exp\{-1/2(x-\mu)^T\Sigma^{-1}(x-\mu)\} \tag{2.21}$$

One of the advantage of using a Bayesian network for traffic flow prediction is that they can very easily model the multivariate traffic flow data to capture the tempo-spatial relations.

Castillo et al. [7] proposed a Gaussian Bayesian network for traffic prediction. The idea was to use the origin-destination (OD) and link flows as a multivariate random variable in this network. The OD flows were used as parent nodes and the link flows as the child nodes. The conditional probability of each link flow given the OD flows was defined as a normal distribution. Once the model was built the joint probability distribution was used to predict the traffic flow for the links, when new data were available. The authors argued that the Bayesian networks are natural tools for representing the random dependence structures of traffic flows in OD pairs and link traffic flows. However, the authors did not present any empirical comparison with other models that have been used in this field.

2.5 Hybrid Methods

In recent years many hybrid methods have been tried in short term traffic prediction with mixed results. Many of these methods take a combined approach, where a forecasts of more than one methods are combined to enhance the prediction accuracy.

One of the first hybrid approach used in short term traffic flow prediction was the ATHENA model. The ATHENA model (Danech-Pajouh and Aron [14]) employed a layered statistical approach. It used a clustering method to group the data and then a linear regression model was applied to each cluster. A hybrid method by combining Kohonen maps with ARIMA model was proposed by Van Der Voort et al. [54]. The model known as KARIMA, used the same data (collected near Beaune, France) that was used in the ATHENA model for an accurate comparison with the later. The authors used rectangular and hexagonal Kohonen maps to cluster the traffic volume data. Then each of the new data cluster was fitted using an ARIMA model. This layered approach was similar to the ATHENA model. The authors observed the superiority of the hexagonal Kohonen maps over the rectangular one, but unable to determine the reason of that. Overall the model showed improved performance than the ATHENA and a simple ARIMA model. Chen et al. [10] analysed the use of hybrid neural networks in the context of traffic prediction and the effect of missing data on those. The authors used two hybrid methods using the self-organising maps (SOM). In the first method they used four ARIMA models while in the second method two multi-layer perceptrons (MLP) were used. The SOM was used to classify the traffic data into different cluster that can then be used by a suitable ARIMA or MLP model. The SOM/ARIMA performed better than individual ARIMA models while the SOM/MLP method outperformed all other methods used in the study. They also observed that the ARIMA models were also the most sensitive to missing data, while neural networks were mostly unaffected. Szeto et al. [51] used a hybrid SARIMA model with cell transmission model for multivariate traffic prediction. The authors reasoned the use of multivariate models captured the spatial characteristics of the transportation network and hence are the natural and better choice over an univariate model. The model was validated against data collected form the city centre in Dublin, Ireland. The results at two junctions were compared against real observations and had MAPE of 4.45 and 10.6. The authors however did not provide comparison against other univariate models or multivariate models which could present the model's relative performance.

Both ARIMA and GARCH (Generalised Autoregressive Conditional Heteroscedasticity) models have received a lot of popularity in time series modelling, especially in the financial analysis. The ARIMA-GARCH model is a new hybrid method that has received some attention. Chen et al. [8] proposed an ARIMA-GARCH model for short term traffic prediction. The authors combines a linear ARIMA and a GARCH model to create a hybrid non-linear model. The authors argued that, for traffic flow prediction, in practice the assumption of ARIMA model for a constant variance is not met. Hence the combination with the GARCH model which has time-dependent variance can improve the prediction accuracy. The ARIMA model was fitted with preprocessed data. The prediction error series was then fitted with the GARCH model. The process is repeated until the model is accurate enough. They evaluated the hybrid model's performance by using traffic flow data collected from a freeway. The performance of the hybrid model did not show any improvements over the standard ARIMA model and reached similar accuracies. The author concluded that even though the model did not show improved performance, they are better choice over a simple ARIMA model as they capture the traffic characteristics more comprehensively.

Yin et al. [65] used a hybrid fuzzy-neural model (FNM) for this task. The FNM model consisted of two modules - a gate network (GN) and an expert network (EN). The role of the GN is to classify the input traffic data into a group of clusters using a fuzzy method. On the other hand the EN was a neural network that models the the clustered data for predictions. An online rolling training scheme was proposed to train the FNM model. The performance of this model was done with a very simple neural network model. It was observed clustering the input data beforehand helped improving the overall performance. It would have been worth comparing the model performance with other hybrid models such as the KARIMA model that takes a similar approach.Stathopoulos et al. [48] proposed a fuzzy rule-based system (FRBS) to combine traffic flow forecasts resulting from an online adaptive Kalman filter and an artificial neural network. The

FRBS is used to represent different forms of knowledge at hand and model the relationships among the variables. The authors used a hybrid FRBS with a meta-heuristic optimisation technique, to automate the tuning of its parameters. The combining of forecasts from the ANN and KF models were done using IF-THEN rules in the rule base. They evaluated this model using real traffic flow data from a urban signalised arterial aggregated at 3-minutes interval. The combines forecasts were better than both individual forecasts. Moreover, the accuracy of the combined model is directly related to the individual accuracies.

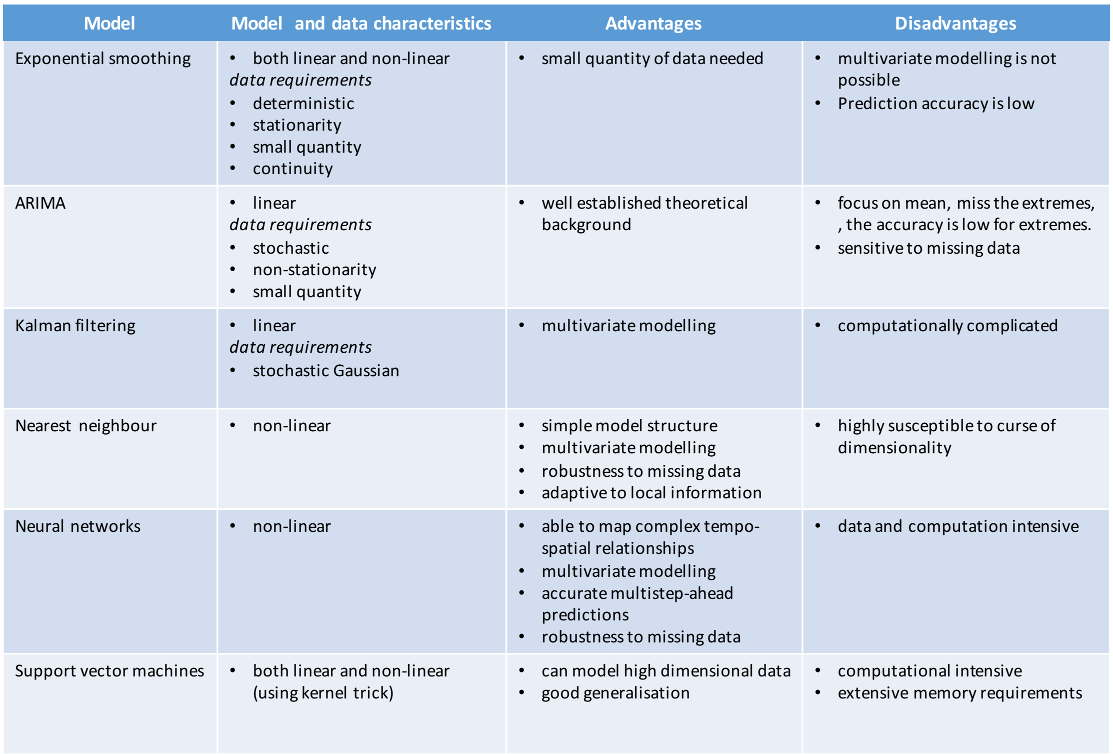
Abdulhai et al. [1] validated the used of neuro-genetic algorithms in short term traffic flow and occupancy predictions on a freeway. They used a time delayed neural network (TDNN) model, whose structure was synthesised by using a genetic algorithm. The model used tried to capture the tempo-spatial relationships in the traffic conditions by using the traffic data from upstream and downstream sections. A genetic algorithm was used to chose the neural network from a population. The model performance was validated using both simulated and real traffic data. The comparisons were made against a back propagation feedforward MLP. The model performed acceptably with an accuracy of 86% for 15 minutes predictions. The model performance was significantly affected for less spatial effects. Vlahogianni et al. [59] took a similar approach where the neural network structure was optimised using a genetic algorithm. The use of GA was to optimise the number of processing units in the hidden layer. When compared with a non-optimised MLP they found that the optimised version reached similar performance with less number of hidden units. They evaluated the model in first using univariate traffic flow data and compared the results with an ARIMA and a state space model. The results were better than the compared models, but on the other hand it was found that ARIMA models do not perform well for urban traffic flow as they concentrate on the means and miss the extreme; Thus this comparison seemed unfair. They also evaluated the model performance in a multivariate traffic flow data to capture the spatial relationships. The results show that using this multivariate approach, the model exhibited a better performance for longer predictive horizons.

2.6 Comparisons

A vast majority of the previous work present some form of comparison between different methods based on the empirical results. But as the results apply only to a specific area, we can not make these as the basis for a general comparison between these methods. Also in most cases the comparison is subjective to the proposed model by the author. Smith et al. [46] performed a comparison between non-parametric and parametric regression models for single point traffic flow forecasting based on their theoretical foundations. They found the parameter estimation and outlier detection using a seasonal ARIMA model is time consuming, hence in practical situations they may not be the best suited. While ARIMA models have their foundations in stochastic system theory, the non-parametric regression is founded on chaotic system theory. The argument that is in favour of using ARIMA models is that traffic conditions data are stochastic in nature. Although this is a valid assumption, the presence of chaotic nature in traffic data can not be dismissed, especially in a congested traffic environment. Hu et al. [24] explored this and applied phase space reconstruction theory to forecast traffic flow and found some positive results. In a similar context Karlaftis and Vlahogianni [29] performed a comparison between statistical methods and neural networks. In this work they outlines some similarities and fundamental differences between these methods. They suggested three areas where these two methods can act in synergy and compliment each other, these are - core model development, analysis of large data sets and causality investigation. They also argued that the comparison of these methods in several work as unfair, as they are solely based on model accuracies.

It is also worth mentioning here that the majority of earlier work focus on prediction of a traffic parameter such as flow at a single location. A very few attempts have been made to predict traffic flow at a network level. Because in practice this is more valuable in both urban and freeway environments. van Hinsbergen and Sanders [55] summarised a list of existing methods used in traffic predictions and mentioned the importance of a model that predicts at a larger network-wide scale.

We present a summary of the comparisons among the above mentioned methods in the table 2.1. The comparisons show the basic characteristics of the methods, the requirements on the input data and overall advantages and disadvantages of the methods. In chapter 5, we present an empirical comparison of these methods.



Model	Model and data characteristics	Advantages	Disadvantages
Exponential smoothing	• both linear and non-linear *data requirements* • deterministic • stationarity • small quantity • continuity	• small quantity of data needed	• multivariate modelling is not possible • Prediction accuracy is low
ARIMA	• linear *data requirements* • stochastic • non-stationarity • small quantity	• well established theoretical background	• focus on mean, miss the extremes, , the accuracy is low for extremes. • sensitive to missing data
Kalman filtering	• linear *data requirements* • stochastic Gaussian	• multivariate modelling	• computationally complicated
Nearest neighbour	• non-linear	• simple model structure • multivariate modelling • robustness to missing data • adaptive to local information	• highly susceptible to curse of dimensionality
Neural networks	• non-linear	• able to map complex tempo-spatial relationships • multivariate modelling • accurate multistep-ahead predictions • robustness to missing data	• data and computation intensive
Support vector machines	• both linear and non-linear (using kernel trick)	• can model high dimensional data • good generalisation	• computational intensive • extensive memory requirements

Table 2.1: Comparison of existing methods applied in short term traffic predictions.

Chapter 3

SCATS Volume Data

"There is no order in the world around us, we must adapt ourselves to the requirements of chaos instead."

Kurt Vonnegut, *Breakfast of Champions* (1973)

3.1 Introduction

SCATS (Sydney Coordinated Adaptive Traffic System) is an adaptive traffic control system. It was developed by the Department of Main Roads in the 1970's. SCATS operates in real-time by adjusting signal timings in response to changes in traffic demand and road capacity. All major and minor cities in Australia and New Zealand use SCATS. Few other cities around the world such as Hong Kong, Kuala Lumpur, Shanghai and Singapore also have adopted SCATS over other adaptive traffic control system. In Melbourne and surrounding cities, SCATS controls more than 3,900 sets of traffic signals

3.2 Traffic volume data

Traffic flow describes the rate at which vehicles pass through a fixed point. The volume is the number of vehicles that are measured for a time t. Even though the two terms flow and volume represent different measurements, often in the literature they have been used interchangeably. In this section we present how traffic volume data is acquired using vehicle loop detectors and the measurement errors that are normally found while dealing with this data.

3.2.1 Data acquisition

Traffic loop detectors are embedded in the road pavement and located in each lane near the stop line at traffic intersections. These detectors collect traffic volume and the time it takes a vehicle to clear the loop. A schematic diagram of a loop detector is shown in fig 3.1. The main components of a loop detector are the wire loops, the extension cables and the control unit. The control unit sends an electrical energy to the wire loops which creates a magnetic field. When a vehicle is stopped or passes over the wire loops, it induces an eddy current in the wire loops causing a decrease in inductance. This decrease in frequency is sensed by the control unit and a presence or passing of a vehicle is detected.

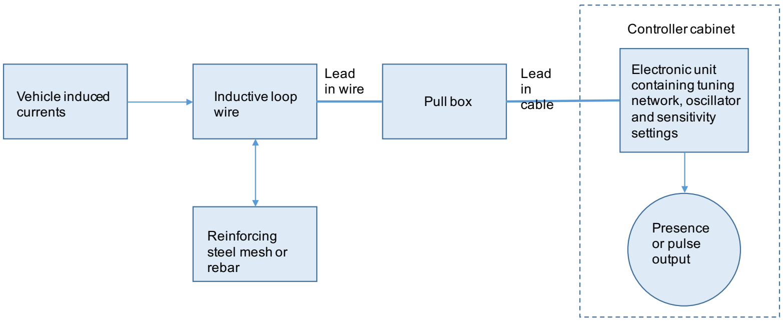


Figure 3.1: An inductive loop detector system used for detecting the presence or passing of a vehicle. (Source - *Traffic Detector Handbook: Third Edition—Volume I*)

For this study a relatively large data set was used. This data set contains loop detector data collected at several detection points in and around the Melbourne CBD. This dataset is a homogeneous set of 1084 road sections. A homogeneous section of the road is where the traffic flow normally remains unchanged during the measured time period. The data was aggregated to a 15 minutes interval over a period from 01/01/2007 to 25/07/2012, making a total 195168 observations.

3.2.2 Measurement errors

There are various factors that contribute to the measurement errors in traffic volume data. This can at the vehicle loop detection system or at the traffic control centre. The electronic unit that detects the change in frequency may not always be accurate leading to a false or missing reading. We present the data caused by errors in two groups - missing data and false or unreliable data.

3.2.2.1 Missing data

Missing data occurs often in the case where either the loop detection system or the computer system at the control centre goes down. The missing period could vary between minutes to days depending on the detection and resolution of the fault. Secondly the data could be missing for a certain location or a set of locations.

As the data we acquired has zero value for missing data, it is difficult to say if that a zero value is a result of missing data due to fault or a valid actual measurement. However we can aggregate the data at a daily level, which then gives us a more better view of the missing data phenomena. Figure 3.2 shows the number of days for which no measurement was recorded for each of the 1084 locations.

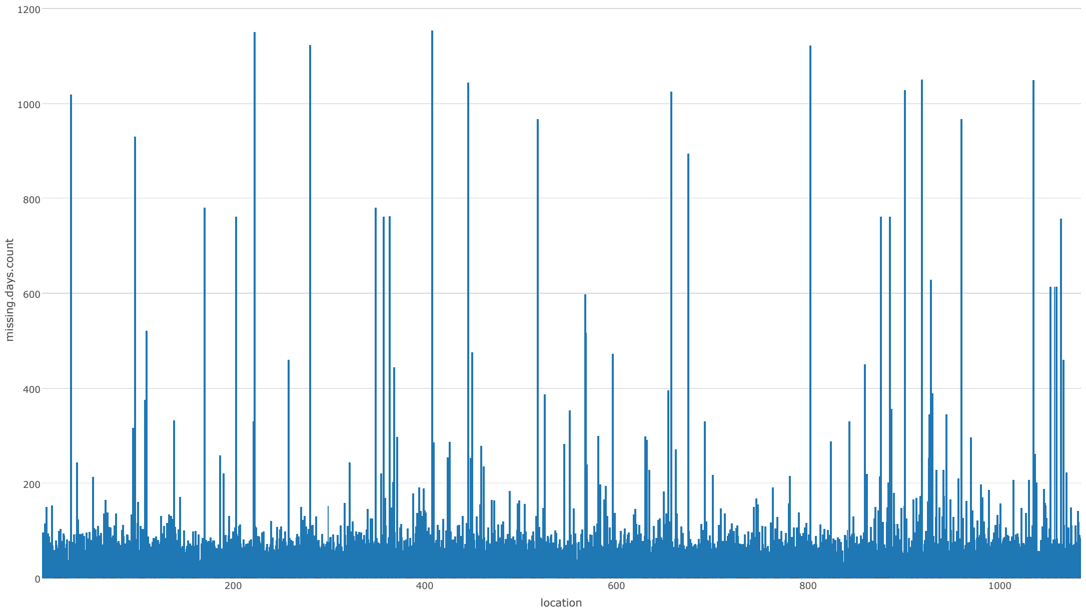


FIGURE 3.2: Missing data - number of days when no measurement was recorded at each of the 1084 locations in our data. The locations are shown as index numbers.

We can see the number of missing data it is quite significant and it can largely affect the model for short term traffic prediction. There are several imputation strategies that are usually used in practice to deal with missing data in traffic data. One of the methods is known as historical imputation. In this method previously collected data at similar time interval at the same location is used to fill in the missing value. The distance in time is often selected as minimum as possible. A variation of this is another simple approach that takes an average of past few recorded observations and uses that to fill in the missing data. A second method that is used is spline/linear regression regression that interpolates the missing value from neighbourhood data points.

3.2.2.2 Unreliable data

Unreliable data is observed when the measurements recorded by the loop detectors show unreasonably large deviations. This is due to some fault in the loop detector system. These are obvious to the naked eyes when looking at the plot of the measurements but hard to detect automatically. Such errors are usually detected during preprocessing by setting a maximum value that a measurement can have. The maximum value can be obtained using the frequency distributions of the measurements.

3.3 Analysis of the traffic volume data

In this section, we present a moderate analysis on the traffic volume data mentioned earlier in previous section. An in depth analysis on traffic flow, covering all the factors that influence the short and long term variations, extends beyond the scope of the work. For the purpose of our analysis we chose the location with the least number of missing data. This location is a road section(HF No. 16913) on Victoria street between Hoddle street and Church street, which is located on the north-east of Melbourne CBD. This is shown in the figure 3.3 annotated in red. The region in blue is the neighbourhood of this section road, which is used to find spatial correlations. This region is further used for experiments and evaluation of various traffic prediction models, presented in chapter 5.

3.3.1 Variations and trends

3.3.1.1 Systematic variations

Recurrent daily variations in traffic flow data exhibit similar behaviour at a location for a particular day. These variations are caused by travel demands of day to day activities. In general, for any particular day we can segment the traffic flow into following components - peak hours usually one in morning and one afternoon, off-peak hours, evening hours and night hours. These can be easily examined using simple visualisations. Let us inspect how average traffic volume changes over time during a day. We first group the volume data into two categories - weekdays and weekends and present the plots in figure 3.4,

We can see from the plot that, a significant difference in how traffic flow changes during the course of a week day and a weekend. On weekdays we can see a steep rise in the traffic flow in the morning between 7AM and 9AM. Another peak is seen at the evening

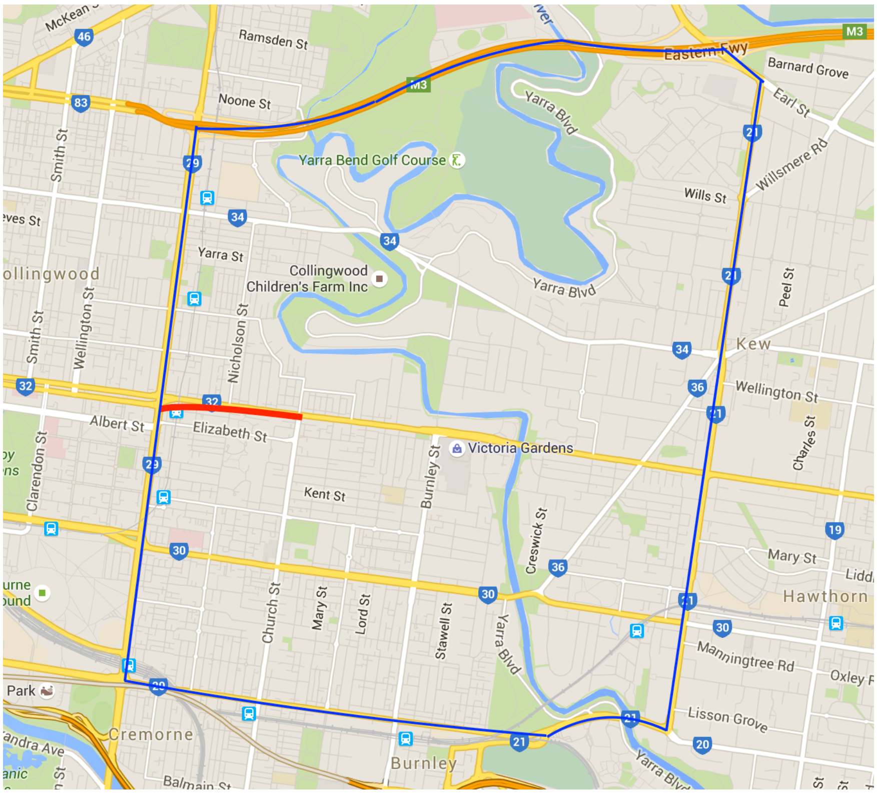


FIGURE 3.3: The traffic region used in this experiment. The boundary is dentoed by the red line. Source - *Google Maps*

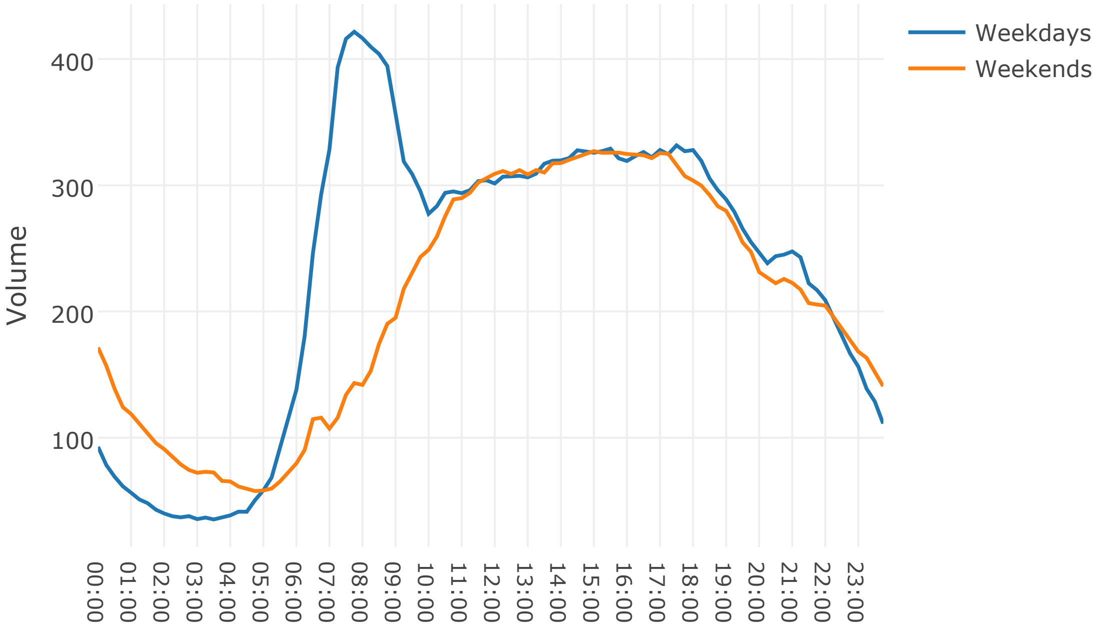


FIGURE 3.4: Variations in average traffic flow at 15 minutes interval grouped by weekdays and weekends.

time between 5PM and 6PM. These sudden changes in traffic flow are mainly because of the traffic caused by getting to and coming from work. There is also a steep drop in traffic that starts at 9PM. On weekends the variations do not have such steep rises and falls, rather the traffic slowly builds up and stays almost stable throughout the afternoon and then slowly starts to drop. These variation on weekends can mainly be attributed to personal activities.

However traffic flow on each day of the week is also different. In figure 3.5, we plot the daily variations in traffic. From this we can infer the morning peak is highest on Thursdays, while Mondays have the least of the morning peaks among all weekdays. During afternoon there is not a large difference in the peak among the days. We can also see on Friday and Saturday the drop in traffic flow happens much later than the other days and also the drop is less steeper.

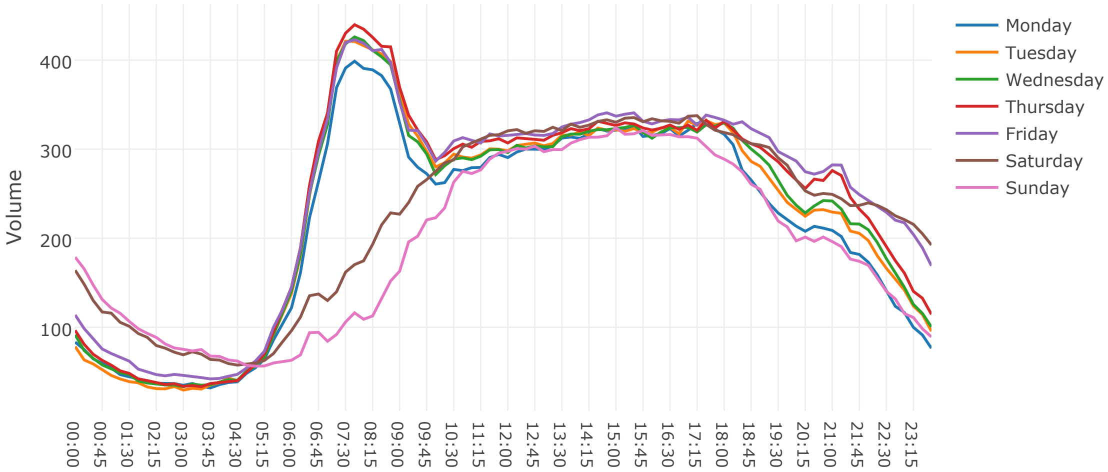
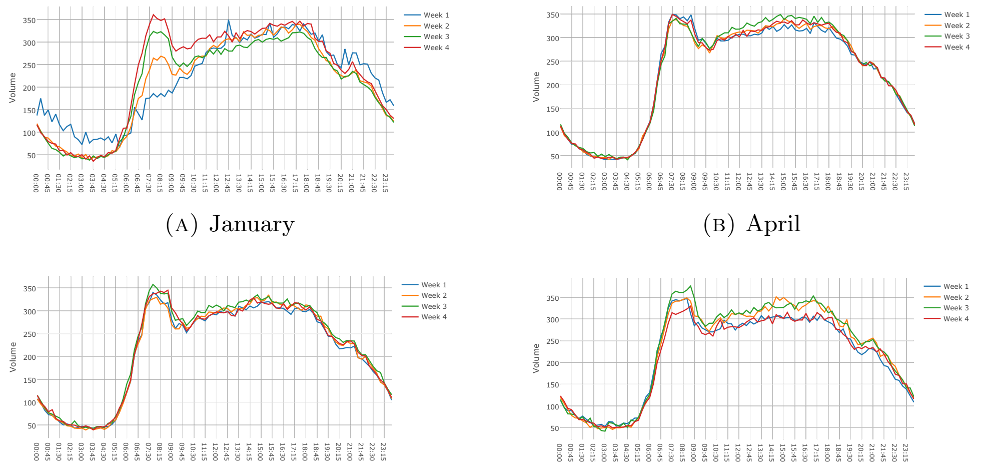


Figure 3.5: Daily variations in traffic flow, the above plot shows average 15 minute traffic flow grouped by day of the week

Similarly during the course of a month, traffic flow on an average day can be different for different weeks. Again to observe these variations we plot the average traffic volume for four weeks of the month of January, April, July and October. This is shown in figure 3.7. From this we can not tell any consistent difference in average traffic flow on each of these four weeks.

Finally to find out the changes in traffic flow on an average day of different months of a calendar year. We can see that the morning peak is lowest in the month of December. The overall traffic is lower in the months of June and December than the other months.



(A) January

(B) April

(C) July

(D) October

Figure 3.6: Average traffic flow during a day in a month grouped by weeks. We only show three months here. The plot shows average 15 minute traffic flow during the months of January, April and July.

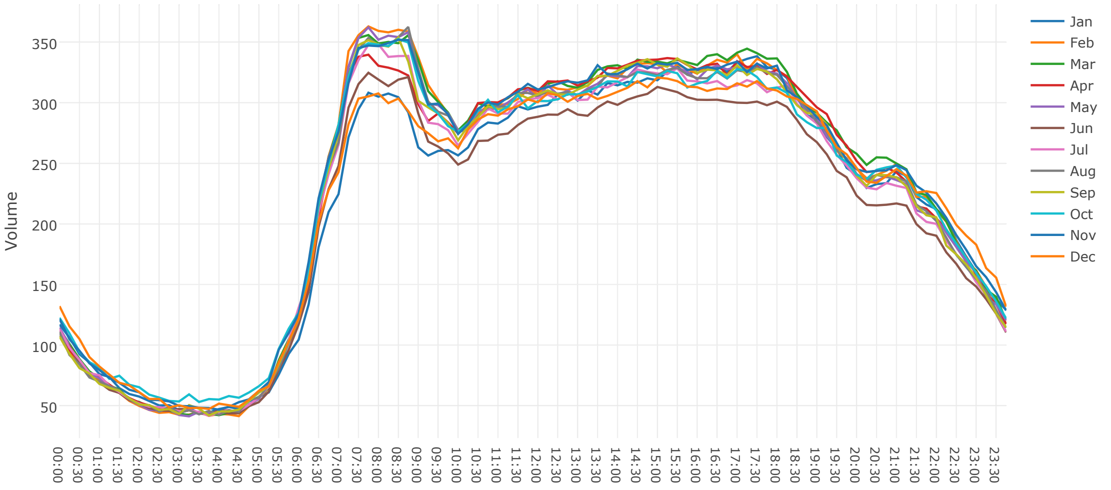
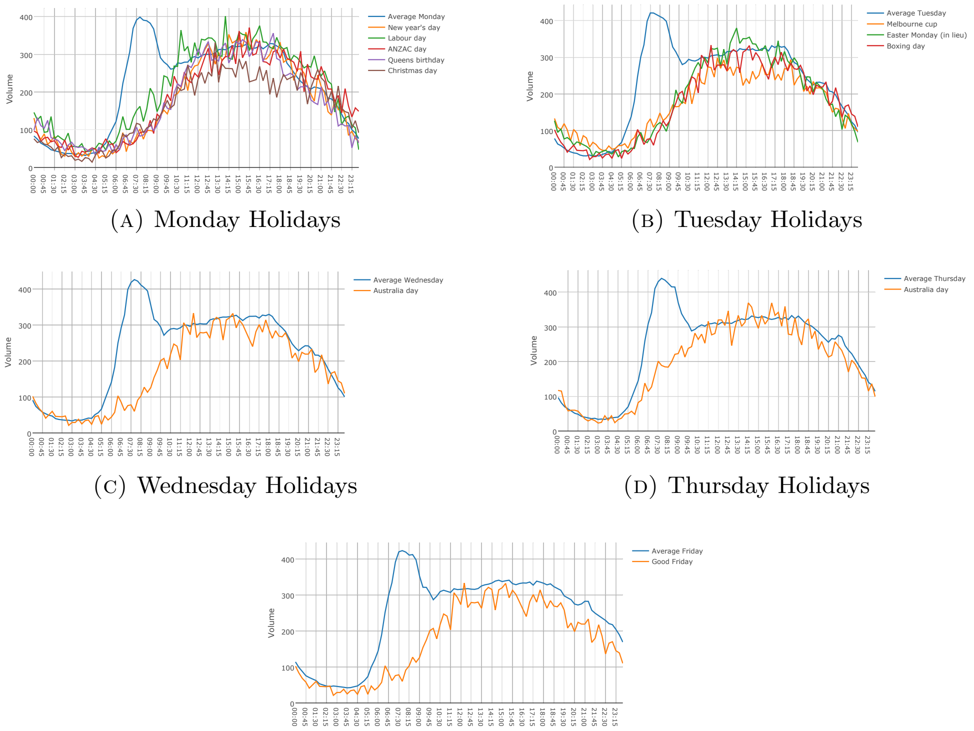


Figure 3.7: Average traffic flow during a day grouped by months.

3.3.1.2 Outliers

Outliers in traffic variations are caused by public events or non-recurrent events such as accidents, road work etc. In fig 3.8, we plot the traffic flow on public holidays. For comparison purpose we divided this into five groups based on the day of the week, that is from Monday to Friday. In each plot we also show the average traffic flow. From these plots we can see different behaviours in changes in traffic flow on different public holidays. On Easter Monday, Labour day, Melbourne Cup Day and ANZAC day there is a higher road traffic flow than the average, while on the rest of the public holidays the flow is significantly lower than the average. So the effect of public holidays on traffic flow is not obvious and does not follow a consistent pattern.



(A) Monday Holidays

(B) Tuesday Holidays

(C) Wednesday Holidays

(D) Thursday Holidays

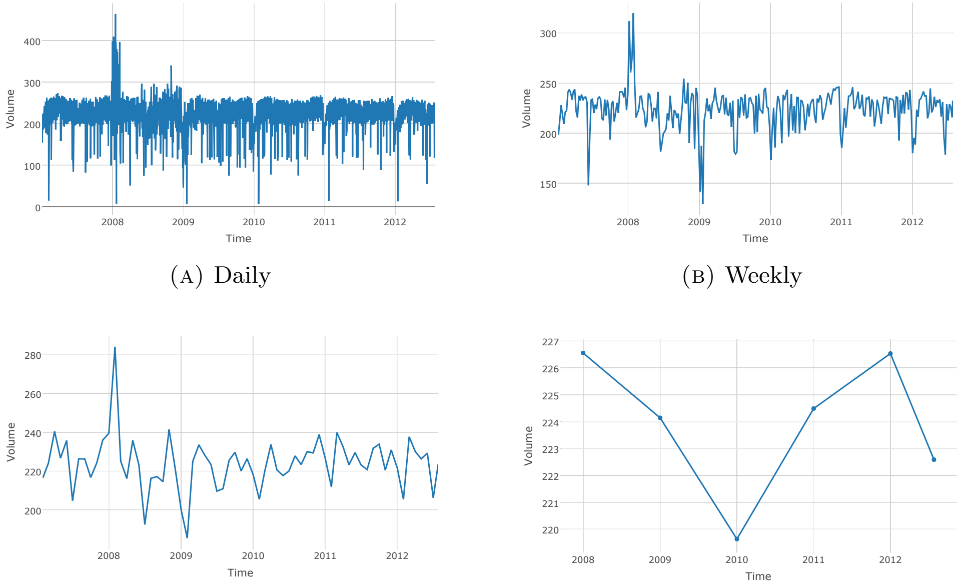
(E) Friday Holidays

FIGURE 3.8: Variations in traffic volume caused by public holidays

3.3.1.3 Trends

The secular variations in traffic volume data is important from infrastructure planning point of view. While long term forecasting of traffic data is out of the scope for this work it is interesting to find out the trends in the traffic volume, to see how it has evolved over the years. Figure 3.9 shows the average 15 minute flow on a daily, weekly, monthly and yearly basis.

Moreover, we can decompose the traffic volume data, and see the different components. The plots are shown in figure 3.10. We can see that the trend in traffic flow is not linear. There was a small upward trend between 2007 and 2008, then a small downward trend till 2009. After that was a slowly rising upward trend which started to flatten since the end of 2011.



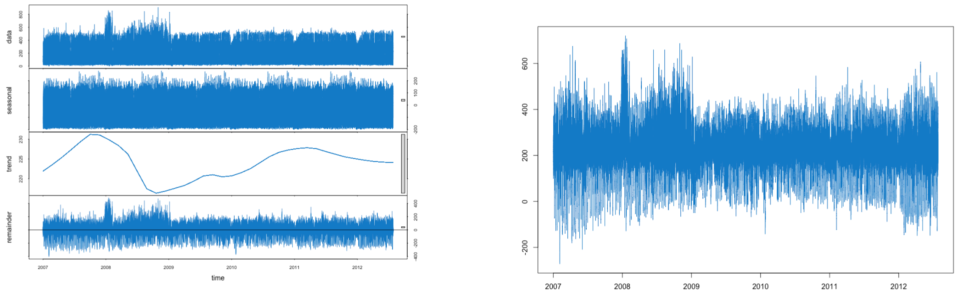
(A) Daily

(B) Weekly

(C) Monthly

(D) Yearly

FIGURE 3.9: (a) daily, (b) weekly, (c) monthly and (d) yearly average of traffic volume (15 mins interval)



(A) Time series components

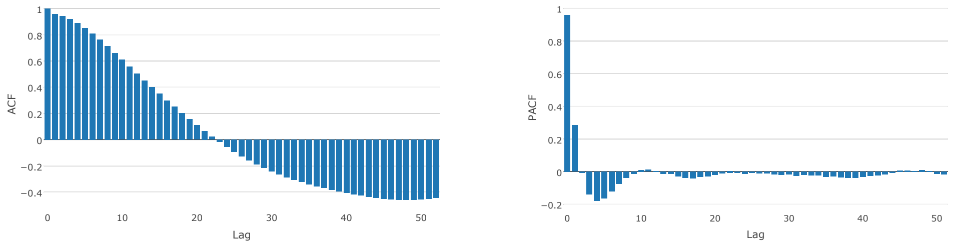
(B) Seasonality adjusted

FIGURE 3.10: Traffic volume time series components and seasonality adjusted

3.3.2 Temporal and Spatial correlations

We would also like to analyse the temporal relations between a series of traffic volume observations and its own lagged values. We can see these relation by using the Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF). The ACF gives a representation of which lag values have linear relationship with the current observed value. The plots of these are shown in 3.11. From the ACF, it is obvious that there is a strong correlation between the observation at time t with lags up to 10. It means that currently observed traffic volume is shows a strong linear relationship between past few observations.

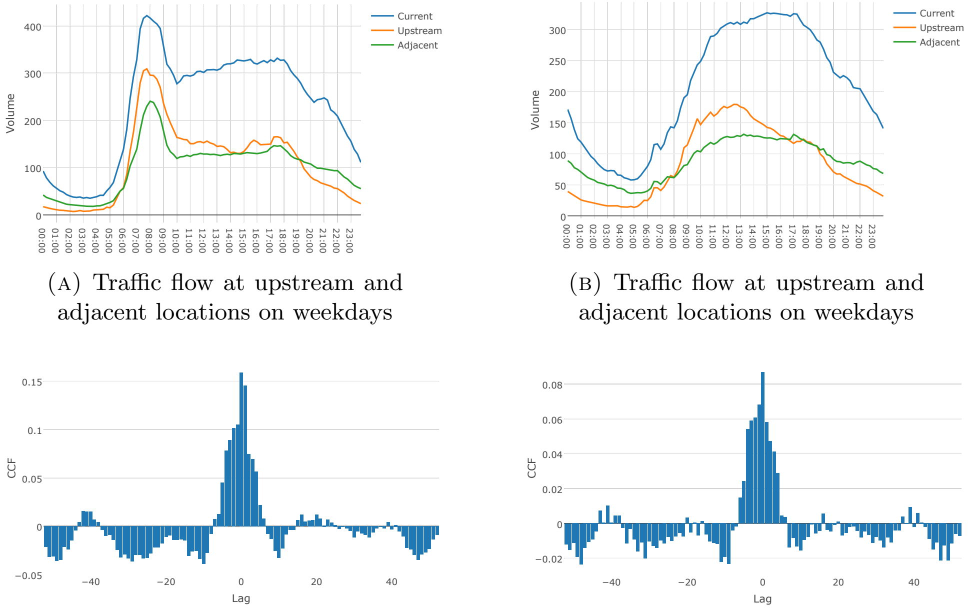
Traffic flow at upstream and adjacent locations correlate with the current location's traffic. This spatial relations are always present in a road network and can be easily



(A) ACF (B) PACF

Figure 3.11: Autocorrelation and partial autocorrelation of traffic volume time series data

be described by the nature of traffic movements. To analyse the spatial relations, we chose one upstream location and one adjacent location. In figure 3.12, we show the cross correlation between the current location's traffic volume with the upstream and adjacent location's traffic volume. We can see that there is strong correlation between the current traffic flow with the upstream traffic flow at lag 1, while the for correlation with the adjacent traffic flow is at lag 0.



(A) Traffic flow at upstream and adjacent locations on weekdays

(B) Traffic flow at upstream and adjacent locations on weekdays

(C) Upstream traffic

(D) Adjacent traffic

Figure 3.12: Spatial correlations between traffic volume data from different locations. (a) and (b) shows the average traffic flow on weekdays and weekends at the current, upstream and adjacent locations. Figure (c) and (d) shows the plots of cross-correlation functions between the current location and upstream and adjacent location.

Chapter 4

Deep Neural Networks for Short Term Traffic Prediction

"I am a brain, Watson. The rest of me is a mere appendix."

Arthur Conan Doyle, *The Adventure of the Mazarin Stone* (1921)

In section 2.4.2, we presented a brief introduction to artificial neural networks and reviewed existing literature in short term traffic prediction that used various types of neural networks. In the following sections we present a brief overview of deep learning. We then describe deep feedforward networks, deep recurrent networks with emphasis on the Long Short Term Memory (LSTM) networks which are a redesigned version of recurrent networks. Later we present how we can we can use these kind of networks for short term traffic prediction.

4.1 Introduction

We live in a world where almost every interaction of ours with the external world uses some form of computing. Computers have become an inseparable part of human lives. In the earlier days when computers were built, people began to ponder whether they could achieve human level of intelligence. Even though at that point the answers seemed optimistic, it has taken quite some time and understanding on our part to make significant achievements in the field of artificial intelligence. One of the approaches was to use knowledge base systems, where computers reason about real world concepts, that were defined in hard-coded formal languages, using logical inference rules. These systems led to little success. The difficulties faced in the knowledge based approach made us built computers to learn automatically from data, an approach we know as machine learning.

A large number of real world problems could easily be tackled using machine learning. However for the machine learning algorithms to perform well they need to be provided with proper representation of data. For example, in a problem where we would like to detect humans in images, it is difficult to represent various shapes of human body in terms of raw pixels. Finding a proper representation from data is a challenge and sometimes become very difficult. A class of machine learning algorithms called representation learning, tackles this problem by learning the representations as well. Autoencoders are such types of algorithms. Again the problem with representation learning is that it is not easy to find the representations due to the presence of various factors of influence (Bengio et al. [3]). Deep learning solves this problem in representation learning by taking a layered approach by expressing representations in terms of simpler representations. The mapping from the input to output is done through a series of hidden layers, where each layer is an abstraction on the previous layer. The depth of the model can be viewed as the depth of the computational graph, i.e. the number of sequential instructions that need to be executed to map an input to output.

4.2 Feedforward neural networks

Deep feedforward networks are the most important deep learning models. The main goal of a deep feedforward network is to approximate a function f^* that maps an input $\mathbf{x}$ to an output y. As the name implies, the information in these models flow in the forward direction. These are the basis of several models used in commercial applications such as the convolutional networks, which are extensions of the feedforward networks, have been very successful in image recognition. With the addition of feedback connections to feedforward networks, recurrent networks are created. Feedforward networks consist of a chain of layers, which is simply done by composing functions for instance we can compose three functions as to map an input $\mathbf{x}$ to an output y, $y = f(\mathbf{x}) = f^3(f^2(f^1))$. Function f^2 acts as the hidden layer that maps the output from the input layer f^1 to the input of the output layer f^3.

The diagram 4.1 illustrates a simple feedforward neural network with 3 nodes in the input layer, 3 and 4 nodes in two hidden layers and a single node output layer. Information propagates from the input layer through the hidden layer to the output layer, known as the forward pass of the network. This kind of feedforward network is called a multilayer perceptron. Multilayer perceptrons are good at classification.

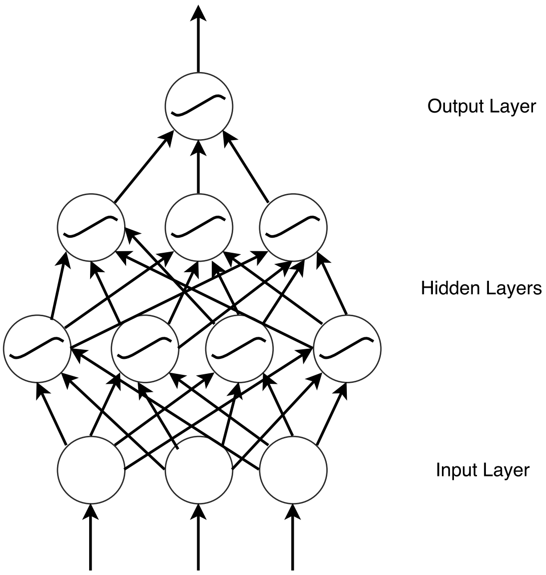


FIGURE 4.1: A feedforward neural network with two hidden layers, this network is also known as a multilayer perceptron. The S-shaped curves denote the sigmoidal function.

Let's consider a simple multilayer perceptron with I nodes in the input layer. For an input vector $\mathbf{x}$, where length of $\mathbf{x}$ is I. Each node in the hidden layer gets a weighted sum of the units in the input layer. The output of each hidden unit a_h is then applied to an activation function θ_h to produce the activation b_h

$$a_h = \sum_{i=1}^{I} w_{ih} x_i \tag{4.1}$$

$$b_h = \theta_h(a_h) \tag{4.2}$$

There are several choices for the activation functions with sigmoidal and hyperbolic tan functions are the most common choices. The reason of these choices is nonlinearity of these functions. Recently the recommended activation function for feedforward neural networks is the *rectified linear unit* or ReLU (Nair and Hinton [39]), defined as $f(x) = max0, x$, as they allow faster and efficient training of deep neural network architectures.

The activations flow through the rest of the hidden layers in similar fashion. For instance the l^{th} hidden unit in layer H_l

$$a_h = \sum_{h' \in H_{l-1}} w_{h'h} b_{h'} \tag{4.3}$$

$$b_h = \theta_h(a_h) \tag{4.4}$$

In the output layer, the activation function is applied on the output from the hidden layer to produce the output y. The input a_k to the output unit is given by

$$a_k = \sum_{h \in H_L} w_{hk} b_h \tag{4.5}$$

where L is the number of hidden layers in the network. The number of units in the output layer and the type of activation function are chosen based on the problem task at hand. For binary classification a single unit with logistic sigmoid activation function is primarily used. For classification with k ¿ 2 classes, k output units are used and the outputs are normalised using the *softmax* function. A very common example of this is the hand-written digits classification, where the output layer consists of 10 units.

4.2.1 Network training using gradient descent

Training neural networks is no different than any other machine learning models with a loss function and gradient descent algorithm. However the difficulty is that the non-linear characteristics of neural networks causes to the loss functions to become non-convex. So the training procedure usually involves small iterative gradient descent algorithm to get a very low value of the cost function. For feedforward networks the weights are initialised with very small random numbers and the biased may be initialised to zero or very small values as well.The choice of a cost function is somehow important and usually these are same as the linear models.

4.3 Recurrent neural networks

As mentioned earlier, we can create a recurrent neural network by adding feedback connections to a feedforward network. Several types of recurrent neural networks have been proposed over the years, some of which are - *echo state networks, time delay networks, Jordan networks.* At first the difference between a feedforward and a recurrent network may not be obvious and seem trivial but recurrent networks are very powerful in the sense that they can retain the history and thus forming a memory in their feedback connections. Similar to the MLP, the forward pass of the RNN the information propagates from the input layer to the output layer through the hidden layers. The only difference is that the input to the hidden layers consists of both the external input and the activations from the previous step. For a sequence of inputs $x = x_1, ...x_T$, the activation of the hidden unit j at time step t is calculated as

$$h_t^j = \begin{cases} 0 & \text{if } t = 0 \\ \phi(h_{t-1}^j, x_t) & \text{if } t > 0 \end{cases}$$

where ϕ is a non-linear function such as a logistic sigmoid function. In the backward pass, the gradients of the weights are calculated, using one of the two algorithms - Real Time Recurrent Learning (RTRL) or Back propagation Through Time (BPTT).

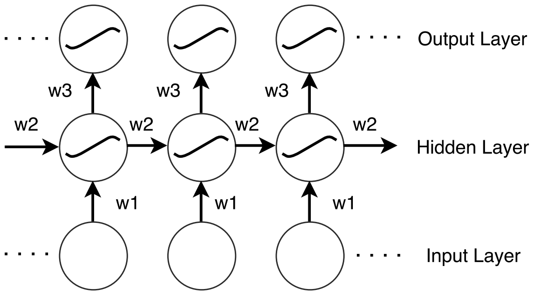


Figure 4.2: An unfolded recurrent neural network. w1, w2 and w3 are weighted connections.

4.3.1 LSTM networks

In previous section we learn that using a recurrent neural networks we can store information in form of activations in the feedback connections. The major disadvantage with recurrent neural networks is their inability to retain information for a long period of time. This is caused by an effect known as *vanishing gradient problem* (Bengio et al. [4], Hochreiter et al. [21]). The vanishing gradient problem is depicted in the figure 4.3. Number of attempts were made in the 1990's to resolve this issue. Hochreiter and Schmidhuber [22] proposed a redesigned network called Long Short Term Memory (LSTM) to address this problem.

An LSTM network is a set of recurrently connected LSTM blocks, also known as memory blocks, where each memory block has one or more memory cells and three units (input, output and forget gates) that perform the read, write and reset operations. A basic LSTM block with one memory cell is depicted in the figure 4.4. The multiplicative units allow the LSTM to store information for a long time and thus addresses the problem of vanishing gradient. An LSTM network is shown in figure 4.5, the hidden layers contains the LSTM blocks.

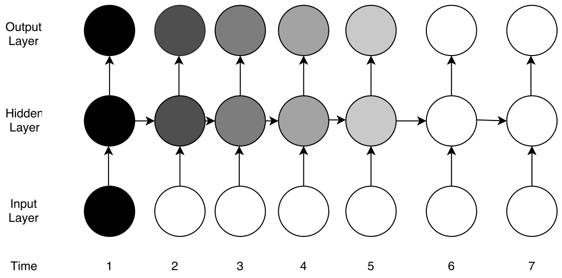


Figure 4.3: The problem of vanishing gradient in recurrent neural networks. The sensitivity, as indicated by the shading, gradually diminishes with time

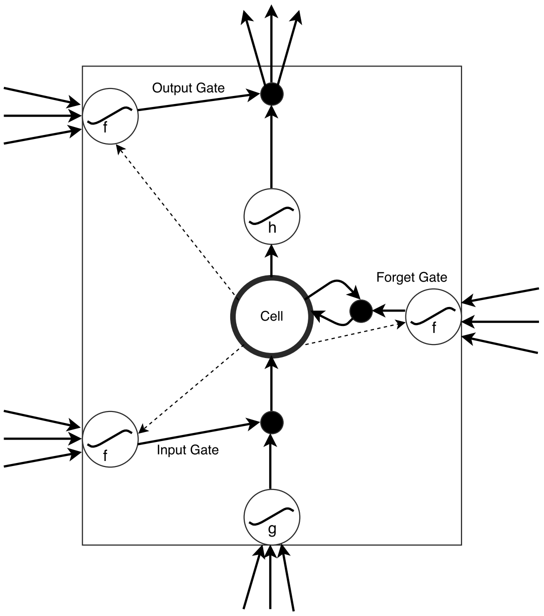


Figure 4.4: An LSTM block with one cell. The three units collect activations from both inside and outside of the block. The small black circles represents multiplications by which the gates control the memory cell. The gate activation function is f, usually a logistic sigmoid. The cell input and output functions are g and h, usually tanh or logistic sigmoid. The dashed lines represent the weighted peephole connections from the cell to the gates. All other connections are not weighted. The only outputs from the block to the rest of the network is from the output gate multiplication.

Like other neural networks, the LSTM is trained using the gradient descent. The training algorithms that was initially proposed used a variant of Real Time Recurrent Learning (RTRL). In the forward pass the hidden layer activations are recursively calculated starting at time step t=1 to t=T, where T is the length of the sequence. But unlike RNN the activation of a hidden unit is not a simple non-linear function. As the an LSTM unit has a memory cell c_t^j at time t, the activation of this unit at time t is calculated as

$$h_t^j = o_t^j tanh(c_t^j) \tag{4.6}$$

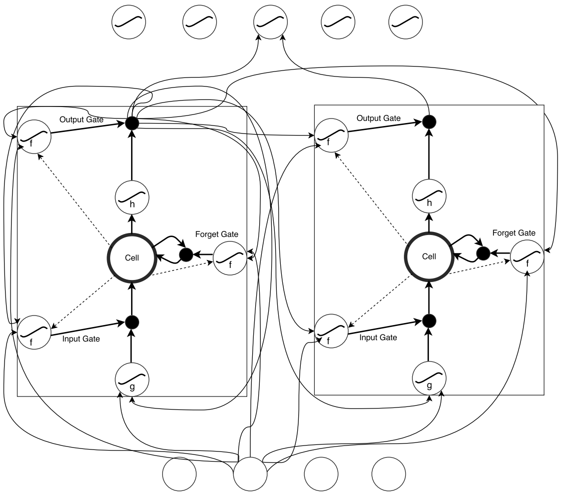


FIGURE 4.5: An LSTM network with one hidden layer with two memory blocks. The input layer consists of four input units and the output layer consists of five output units. Not all connections are shown in the figure. There is only one output from the block.

where o_t^j is the output gate of the unit j, which controls the amount of exposed memory. The memory c_t^j is updated by the forget gate f_t^j by forgetting some of its content and adding new memory contents. The amount of new memory content is controlled by the input gate i_t^j

In the backward pass the training algorithm is used to repeatedly apply the chain rules to calculate the gradients. The difference is that the activations from hidden layer not only influence the output layer but also the hidden layer in the next time step.

4.3.2 GRU networks

The Gated Recurrent Unit network was proposed by Cho et al. [12] is a simpler version of the LSTMs. Unlike LSTMs, in the GRUs the gated units did not require a memory cell. The output and forget gates are replaced with a single update gate. The activation at time step t is updated by interpolations between previous and current activations. The amount of update to the activation is decided by the update gate. The activation of the hidden unit j is given by

$$h_t^j = (1 - u_t^j)h_{t-1}^j + u_t^j h_t^j \tag{4.7}$$

where u_t^j is the update gate that controls how much update is done to the activation.

4.3.3 Applying LSTM to short term traffic prediction

LSTMs were designed to address the issue of vanishing gradient that is observed in RNNs. The design of the LSTM's allowed them to store information in the memory cell, and thus make able to remember long range information. Earlier in chapter 3 we analysed the traffic volume data and observed that traffic volume at any particular location and time has dependencies on the both the earlier observations at that location and observations at neighbourhood locations. These dependencies can be modelled using an LSTM network. In figure 4.6 we present an architecture for such a network.

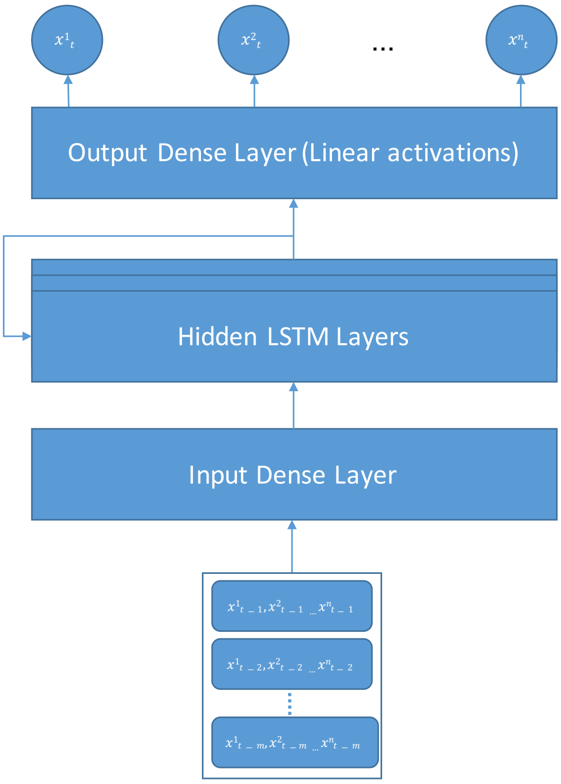

x^1_t
x^2_t
...
x^n_t
Output Dense Layer (Linear activations)
Hidden LSTM Layers
Input Dense Layer
$x^1_{t-1}, x^2_{t-1} \dots x^n_{t-1}$
$x^1_{t-2}, x^2_{t-2} \dots x^n_{t-2}$
$x^1_{t-m}, x^2_{t-m} \dots x^n_{t-m}$

Figure 4.6: Block diagram of an LSTM network for short term traffic prediction. Each input sequence has m vectors, where each vector contains the observations at n locations. The output is a sequence of length n, which contains prediction at time t for each of the n locations. The Dense input and output layers are fully connected regular NN layers. The output layer units have linear activation functions.

The network has one input layer and one output layers. The number of hidden layers and the number of units in them can be found by experiments. In chapter 5, we present these details. The input dataset consists of a set of sequences, where each sequence is $m \times n$; m is the number of time steps and n is the number of locations. The output layer has linear activations.

Chapter 5

Experiments and Results

"Science, my boy, is made up of mistakes, but they are mistakes which it is useful to make,because they lead little by little to the truth"

Jules Verne, *Journey to the Centre of the Earth* (1864)

For our experiment we used various existing methods in short term traffic prediction along with the deep neural network models. This is to perform a broad comparison among the methods. For baseline model purpose we chose the naive method. For comparison purpose we chose Linear regression, ARIMA, Exponential smoothing with state space model, Neural network autoregression with a single hidden layer, K nearest neighbour and Support vector regression. We used three variants of deep recurrent networks - simple RNN, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). For the purpose of our experiment we used the data, whose details were presented in chapter 3. All the models used the data from the homogeneous road section (HF No. 16913) on Victoria street between Hoddle street and Church street, as shown in the figure 3.3 annotated in red. We also used the data from neighbourhood locations (the region annotated in blue) for training the deep neural networks to capture the spatial relationships. For handling missing data, we used interpolation method. All the models were used for prediction horizons of 15 minutes, 30 minutes and 45 minutes.

5.1 Basenline and compared models

In this section, we present the details of the experiments and results of the chosen baseline method and the compared models. The naive method was used to set up a baseline, the naive method uses the last observed value as the prediction. The results of the naive method is shown in 5.1. The plot shows that the naive method perform very well in predicting short term traffic prediction. Also the main advantage is there is no significant computation is required for calculating the prediction values.

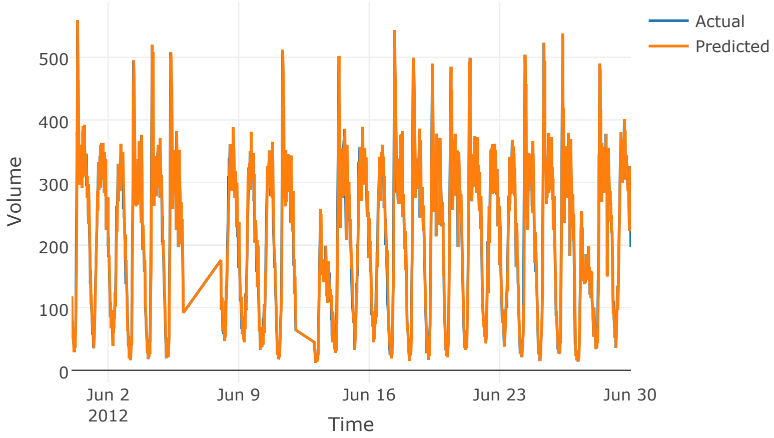


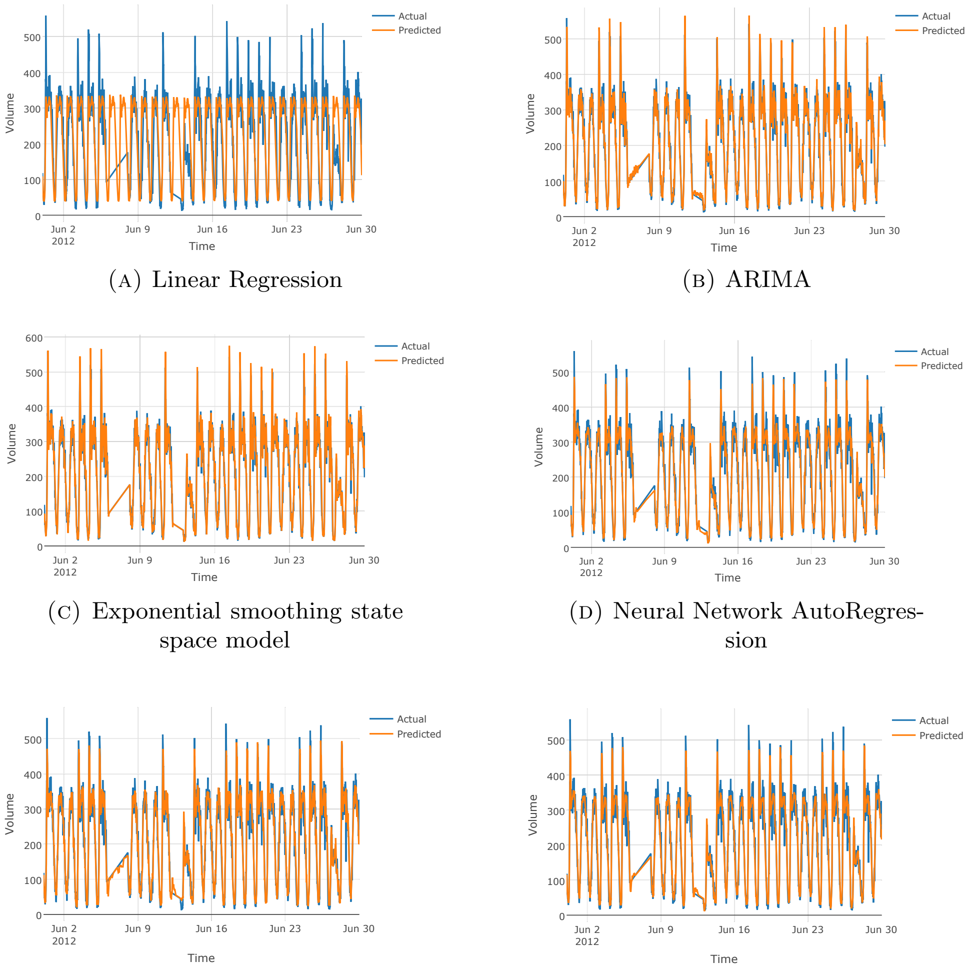
Figure 5.1: Results of the Naive method, which is used as the baseline method.

Once the baseline is set, we used the comparing models to find out their performance. The results of the comparing models are presented in figure 5.2.

- **Linear regression**: We used a linear model for time series regression. The model was created using the season component of the traffic volume time series data. The model did not perform well, in fact its performance was the worst among all the models used in this experiment.

- **ARIMA**: After handling missing data, the Box-Cox transformation was done to stabilise the variance. Then Using the Hyndman-Khandakar algorithm, the best ARIMA model was chosen. The selected model was a seasonal ARIMA(2,0,5)(1,0,0)[96]. The model was fit using data from January to May 2012 with frequency 96. Once the model was fit forecasts at steps 1 to 3 (15,30 and 45 minutes horizon) were done in a recursive manner, that is after every forecast the model was refit using the newly available observed data. The ARIMA model performed better than the baseline.

- **Exponential smoothing**: For exponential smoothing, a state space model was used. This model was a damped linear method with additive errors. Data preprocessing was done by handling missing data through interpolation and then applying Box-Cox transformation to stabilise the variance. The model used the same data used for the ARIMA model. The performance of the model was worse when compared to the ARIMA model.



(A) Linear Regression

(B) ARIMA

(C) Exponential smoothing state space model

(D) Neural Network AutoRegression

(E) K-Nearest Neighbour

(F) Support Vector Regression

FIGURE 5.2: Actual vs Predictions - naive, linear regression, ARIMA, simple feed forward neural network with one hidden layer, exponential smoothing using state space model, k-nearest neighbour and support vector regression. The models were trained on traffic data from one homogeneous road segment. The plots show the actual vs predictions (15 mins) for the month of June 2012.

- **Neural network autoregression**: A simple feedforward network with one hidden layer was used. The lagged values of traffic volume time series were used as inputs to this network. A 38-20-1 neural network with 801 weights was used. Again for training purpose the same data that was used for ARIMA and exponential smoothing methods were used. The performance of this model was very good, better than the ARIMA model.

- **K Nearest neighbour**: A neighbour based regression model was used, where the number of nearest neighbours, k is set to 5. This value was selected by using both the autocorrelations and cross validations. The distance measure was used to assign weights to each value in the local neighbourhood.

- **Support vector regression**: An epsilon-SVR with RBF kernel was used for this experiment. The input data was standardised to mean 0 and variance 1.

5.2 Deep neural network models

The experiment details of the three used models are presented below. We used these models on two sets of data. First dataset is from a single location as mentioned earlier. The second dataset consisted of the traffic volume data from the location of interest along with data from 10 other locations in the neighbourhood. The input data used were standardised to mean zero and variance 1. We used various setting for length of the sequence, number of hidden layers and number of units in those layers and optimisation algorithms used for optimising the gradients. We present the final settings of these models below. The results of these models are shown in the figure 5.3.

- **Simple RNN**: The simple RNN used was a fully connected RNN where the output was fed back to the input is used. For univariate modelling, that is using data from a single location and make predictions for that location, the RNN model with 2 hidden layers with 100 units in each layer was used. To avoid overfitting a dropout of 10% was used at both the hidden layers. The weights were initialised randomly using the uniform distribution. A linear activation function was used in the output layer. The loss function used for training was the mean squared error. Finally for optimising gradients, the RMSprop algorithm, which was proposed by Geoff Hinton, was used. The RMSprop is an adaptive learning rate algorithm. The input data was a set of sequences, where each sequence was created with 50 time steps. We trained the model for 30 epochs. For multivariate modelling three hidden layers with number of units [100,200,200] were used. The weight initialisations, optimising algorithm and error functions were same as the univariate model.

- **GRU**: Gated Recurrent Unit is a simple version of LSTM and was recently proposed. Similar to the simple RNN modelling, we used GRU to model both univariate and multivariate datasets. For univariate model, the network consisted of two hidden layers with 100 units in each of them. We used the RMSprop algorithm for gradient optimisation. The number of epochs used was 30. Similar to the RNN network input data was a set of sequences, where each sequence was created with 50 time steps. For multivariate modelling we used three hidden layers with number of units [100,200,200].

- **LSTM**: An LSTM model was also implemented for modelling both univariate and multivariate traffic volume datasets, similar to the above RNN and GRU networks. For univariate modelling number of combinations of hidden layers and units in those were used. The final univariate model consisted of two hidden layers with 100 and 200 units in them. The output layer had an linear activation function. Different sequence lengths were used and a final length of 50 was selected. For optimisation purpose we tried both the RMSprop and Adaptive Moment Estimation (ADAM) algorithms. We found that the ADAM algorithm outperformed the RMSprop. We trained the network for 20 epochs. For multivariate modelling we used three hidden layer with [100,200,200] units. Again we compared the RMSprop and ADAM optimisation algorithms and found the later outperformed the former again.

5.3 Comparisons

5.3.1 Prediction accuracies

For model comparison we used three accuracy measures - mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE). In below sections we briefly describe the accuracy measures.

For defining the accuracy measures let us denote x_i be the i^{th} observation and $\hat{x}_i$ be the prediction of x_i.

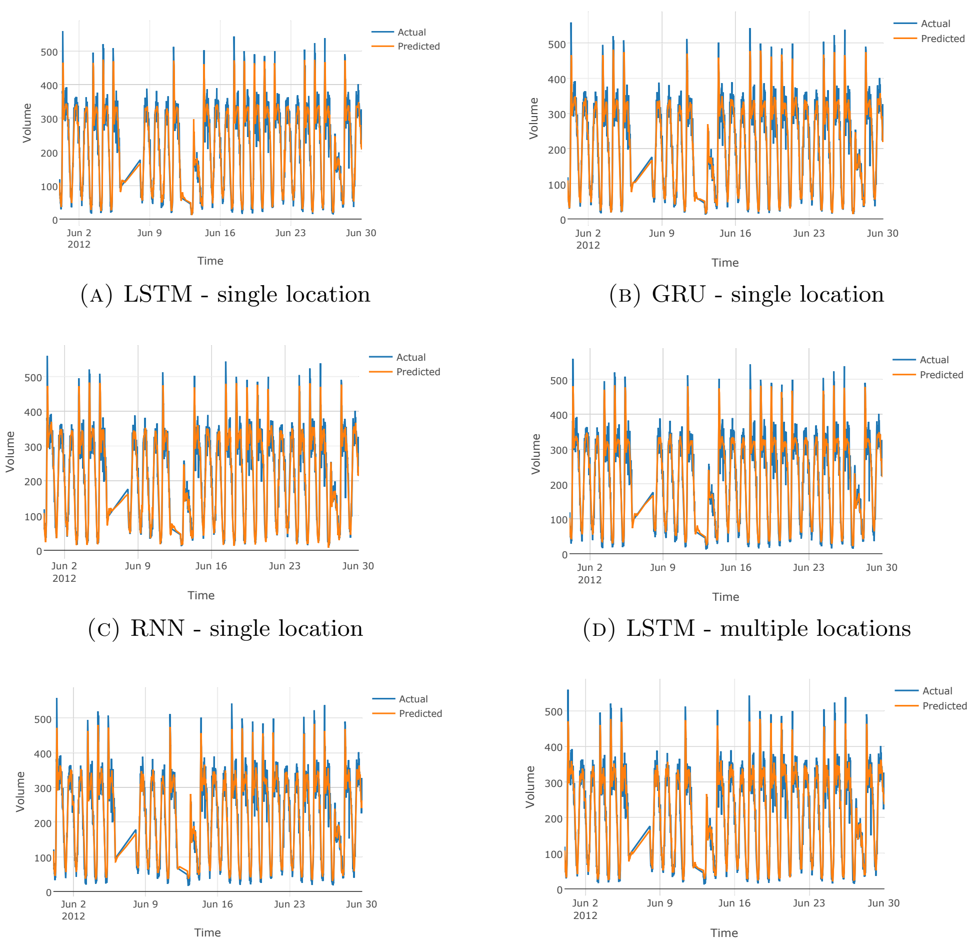
Scale-dependent errors The prediction error is simply given by $e_i = x_i - \hat{x}_i$, which is in the same scale as of the original data. So accuracy measures that depend on e_i are scale dependent and can not be used across multiple series on different scales. The two most used scale-dependent accuracy measures are mean absolute error and root mean squared error defined as below

$$MAE = mean(|e_i|) \tag{5.1}$$

$$RMSE = \sqrt{mean(e_i^2)} \tag{5.2}$$

MAE is easy to understand and popular in usage when using a single dataset.

Percentage errors Percentage errors are scale-independent and thus used across multiple datasets on different scales. The percentage error is given by $p_i = 100 * e_i/x_i$. The



(A) LSTM - single location

(B) GRU - single location

(C) RNN - single location

(D) LSTM - multiple locations

(E) GRU - multiple locations

(F) RNN - multiple locations

FIGURE 5.3: Actual vs Predictions - using three deep neural networks (LSTM, GRU and Simple RNN). The top three are results for the single homogeneous road segment when using data from that location only while the bottom three are results for the single homogeneous road segment when data from neighbouring locations (3.3) were also used . The plots show the actual vs predictions (15 mins) for the month of June 2012.

most commonly used percentage measure is Mean Absolute Percentage Error (MAPE) which is given by the below formula

$$MAPE = mean(|p_i|) \tag{5.3}$$

There are however few shortcomings of the MAPE, for instance when x_i is 0 or very large. Another shortcoming is that they put heavier penalty on negative error values than positive error values.

In table 5.1, the accuracies are presented in terms of the above mentioned error measures for the methods presented in this experiment. The error measures are presented for the

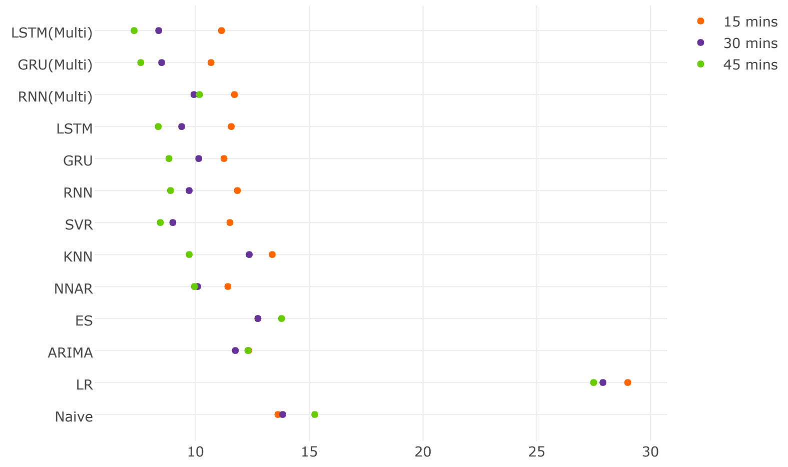


Figure 5.4: MAPE scores for the methods

predictions for 1 step-ahead (15-minutes), 2 steps-ahead (30-minutes) and 3-steps ahead (45-minutes).

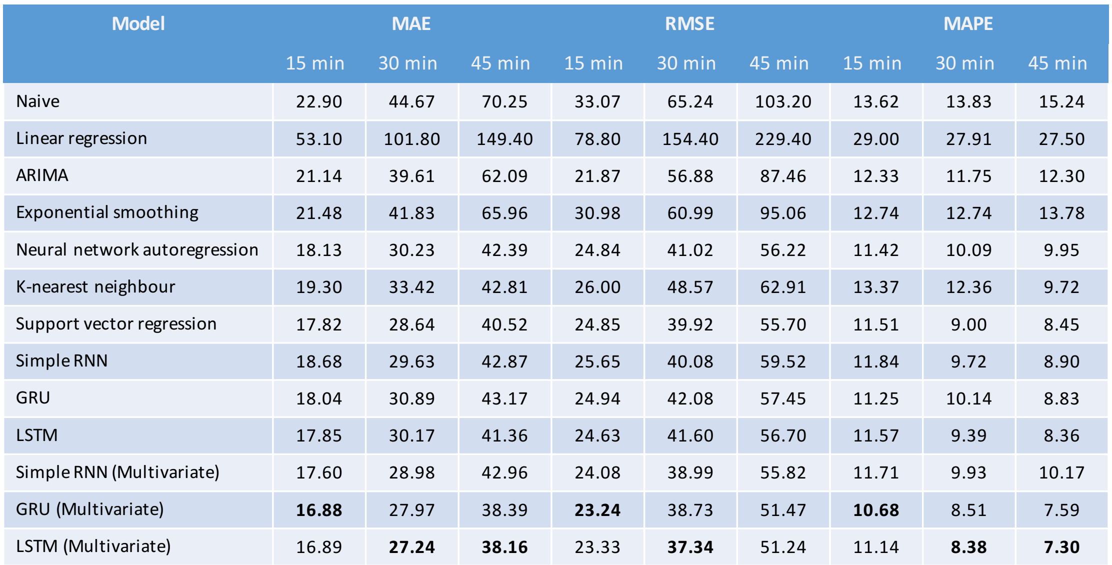
The accuracy of the naive method for 15-minutes predictions was about 86.4%, which is very good considering no computations are required. The linear regression has the worst performance among all the methods, with only 71% of accuracy for 15-minutes predictions, including the baseline naive method. All other methods had better performance than the naive method. Among the exisiting methods, neural network autoregression has the best performance with an accuracy of about 89.6%.

Overall the best accuracies in both one-step and multi-steps predictions were achieved by the deep LSTM and GRU networks with multivariate modelling. For 15-minutes prediction, the best performance was achieved by the deep GRU (multivariate) with about 89.3% accuracy, while for 30 and 45-minutes predictions the best performance was by the LSTM (multivariate) network with accuracies of about 91.6% and 92.7%. In figure 5.4, the mean absolute percentage errors of all the methods for 15, 30 and 45 minutes predictions are plotted.

The advantage of the deep networks used in the multivariate settings is that the predictions of all the 11 locations were done at the same time. In table 5.2, we present the error measures for these locations of the deep LSTM network.

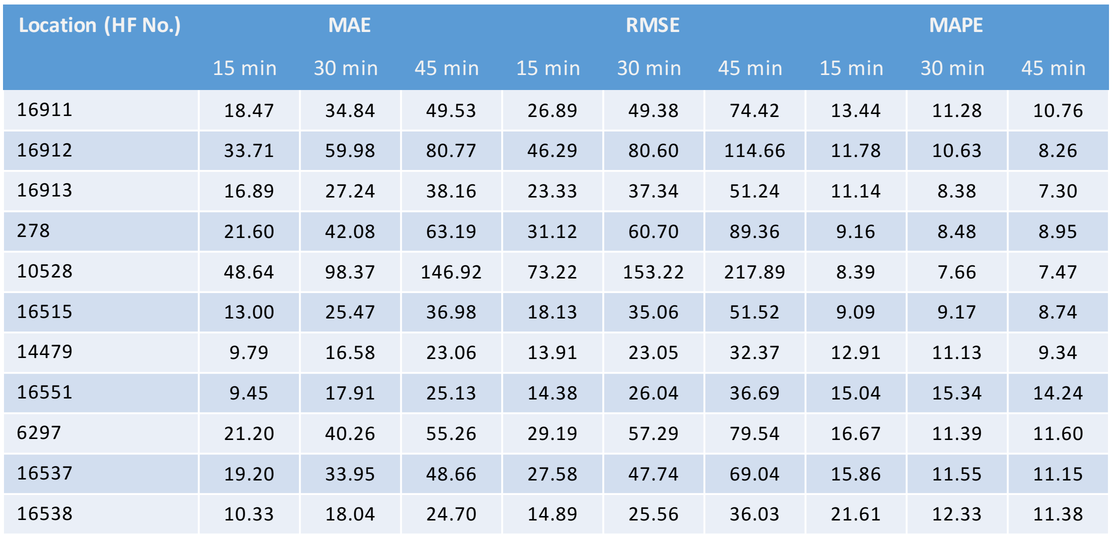
5.3.2 Computational time

For all these experiments we used a home computer with Intel i7-4770 CPU, 16GB RAM, GeForce GTX 960 GPU. The operating system used was Ubuntu 14.04 and for data processing and modelling we used R and Python languages. The training time of



Model	MAE			RMSE			MAPE		
	15 min	30 min	45 min	15 min	30 min	45 min	15 min	30 min	45 min
Naive	22.90	44.67	70.25	33.07	65.24	103.20	13.62	13.83	15.24
Linear regression	53.10	101.80	149.40	78.80	154.40	229.40	29.00	27.91	27.50
ARIMA	21.14	39.61	62.09	21.87	56.88	87.46	12.33	11.75	12.30
Exponential smoothing	21.48	41.83	65.96	30.98	60.99	95.06	12.74	12.74	13.78
Neural network autoregression	18.13	30.23	42.39	24.84	41.02	56.22	11.42	10.09	9.95
K-nearest neighbour	19.30	33.42	42.81	26.00	48.57	62.91	13.37	12.36	9.72
Support vector regression	17.82	28.64	40.52	24.85	39.92	55.70	11.51	9.00	8.45
Simple RNN	18.68	29.63	42.87	25.65	40.08	59.52	11.84	9.72	8.90
GRU	18.04	30.89	43.17	24.94	42.08	57.45	11.25	10.14	8.83
LSTM	17.85	30.17	41.36	24.63	41.60	56.70	11.57	9.39	8.36
Simple RNN (Multivariate)	17.60	28.98	42.96	24.08	38.99	55.82	11.71	9.93	10.17
GRU (Multivariate)	**16.88**	27.97	38.39	**23.24**	38.73	51.47	**10.68**	8.51	7.59
LSTM (Multivariate)	16.89	**27.24**	**38.16**	23.33	**37.34**	51.24	11.14	**8.38**	**7.30**

Table 5.1: Accuracy measures for the evaluated models. The scores are calculated for prediction horizon of 15, 30 and 45 minutes. Mean 15-minutes traffic volume is 224.10



Location (HF No.)	MAE			RMSE			MAPE		
	15 min	30 min	45 min	15 min	30 min	45 min	15 min	30 min	45 min
16911	18.47	34.84	49.53	26.89	49.38	74.42	13.44	11.28	10.76
16912	33.71	59.98	80.77	46.29	80.60	114.66	11.78	10.63	8.26
16913	16.89	27.24	38.16	23.33	37.34	51.24	11.14	8.38	7.30
278	21.60	42.08	63.19	31.12	60.70	89.36	9.16	8.48	8.95
10528	48.64	98.37	146.92	73.22	153.22	217.89	8.39	7.66	7.47
16515	13.00	25.47	36.98	18.13	35.06	51.52	9.09	9.17	8.74
14479	9.79	16.58	23.06	13.91	23.05	32.37	12.91	11.13	9.34
16551	9.45	17.91	25.13	14.38	26.04	36.69	15.04	15.34	14.24
6297	21.20	40.26	55.26	29.19	57.29	79.54	16.67	11.39	11.60
16537	19.20	33.95	48.66	27.58	47.74	69.04	15.86	11.55	11.15
16538	10.33	18.04	24.70	14.89	25.56	36.03	21.61	12.33	11.38

Table 5.2: Simultaneous predictions across all the locations using the deep LSTM network. The network models the spatio-temporal characteristics from the traffic volume data.

all the above mentioned models (except Naive) are shown in the figure 5.5. However not all of the algorithms were used optimised to reduce the training time, so the figures could vary in small amounts.

In this chapter we presented the results of our experiments in short term traffic prediction using several models. We compared the results and found that the performance of deep neural networks outperform all the existing methods.

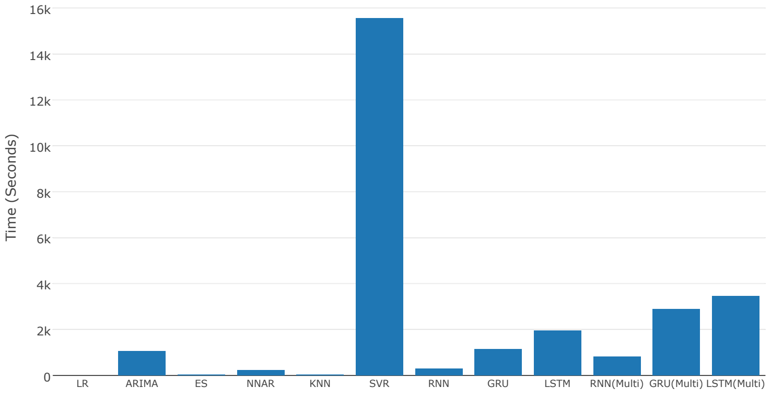


Figure 5.5: Comparison of training time of the models

Chapter 6

Conclusions and Future Directions

"Everything should be made as simple as possible but not simpler."

Albert Einstein

6.1 Conclusions

The objective of this work was two fold; first to find out whether we can use the large amount of available traffic volume data for short term traffic flow predictions. By reviewing existing literature, we understood that the data driven algorithms are capable of handling large amount of traffic volume data. Secondly we wanted to find out whether the application of deep neural networks can use the spatio-temporal relations in such data to provide better prediction accuracies. We have seen that deep neural networks have been used recently to solve various complex tasks in the field of computer vision, speech recognition and natural language modelling, however can they be used to solve other problems?

In chapter 2, We presented the state of the art of various traffic prediction methods and reviewed existing literature in short term traffic prediction. The complex and highly nonlinear nature of traffic volume data make it difficult to accurately predict the traffic in the short term. We found that most of the available methods can be categorised into four groups - naive, parametric, non-parametric and hybrid methodologies. We presented a high level theoretical comparison of these methods, however a fair comparison of these methods using the existing literature is difficult. This is mainly due to the significant differences in the experimental studies presented in these works, which used datasets from different locations and time.

We analysed the traffic volume data collected by VicRoads using vehicle loop detectors. A suitable traffic region was selected based on the number of missing days. We presented the variations in daily traffic flow at a homogeneous road segment in this selected region. We found that the traffic variations on a day is largely influenced by the location and day of the week. We also analysed the influence of public events on traffic flow data. Both the temporal and spatial relations in traffic flow data were analysed. The temporal characteristics are found in the traffic flow data by using autocorrelations. The presence of spatial relations of traffic flow data from current location with upstream locations and adjacent locations were detected using the cross correlations.

We briefly introduced the deep neural networks especially the Long Short Term Memory networks. We conducted the experiments to find out the performance of deep neural networks in predicting traffic flow. For comparison purpose we used a range of existing methods to set up benchmarks. The experiments were conducted in both univariate setting, where only data form one location was used for modelling, and multivariate setting where data from multiple locations were used for modelling the spatio-temporal characteristics. We found out from the results that the data driven algorithms outperformed the parametric methods. The application of deep neural networks showed promising results in both univariate and multivariate settings. The results of the GRU and LSTM networks were comparatively similar but better than the simple RNN networks. The GRU (multivariate) networks had the best accuracy in predicting 15-minutes traffic volume prediction, slightly better than LSTM (multivariate), while LSTM (multivariate) networks had best accuracies for 30-minutes and 45-minutes predictions. From these results we can positively conclude our second objective of this research.

6.2 Future directions

The application of deep neural networks to solve various problems are still at infancy. We believe in coming years their use will increase to solve a range of other problems. This work may provide a basis for further experiments of deep neural networks in short term traffic predictions. The network topologies used in this work can be modified for achieving better results. Implementation of such a network at much larger scale that spans hundreds of locations is computationally very expensive and may not generalise well as the spatial information will get lost in such large scale. In this work we manually handpicked a region and used traffic data from nearby locations. This can be automated by using clustering to partition the traffic volume data of the entire transportation network into several clusters based on strong spatial correlations. Then each of these clusters can be modelled using a deep neural network as presented in this work.

Bibliography

[1] Abdulhai, B., Porwal, H., and Recker, W. (2002). Short-term traffic flow prediction using neuro-genetic algorithms. *ITS Journal-Intelligent Transportation Systems Journal*, 7(1):3–41.

[2] Ahmed, M. S. and Cook, A. R. (1979). *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. Number 722.

[3] Bengio, Y., Goodfellow, I. J., and Courville, A. (2016). *Deep learning*. MIT Press.

[4] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

[5] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley &amp; Sons.

[6] Brockwell, P. J. and Davis, R. A. (2006). *Introduction to time series and forecasting*. Springer Science &amp; Business Media.

[7] Castillo, E., Menéndez, J. M., and Sánchez-Cambronero, S. (2008). Predicting traffic flow using bayesian networks. *Transportation Research Part B: Methodological*, 42(5):482–509.

[8] Chen, C., Hu, J., Meng, Q., and Zhang, Y. (2011). Short-time traffic flow prediction with arima-garch model. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 607–612. IEEE.

[9] Chen, H. and Grant-Muller, S. (2001). Use of sequential learning for short-term traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 9(5):319–336.

[10] Chen, H., Grant-Muller, S., Mussone, L., and Montgomery, F. (2001). A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. *Neural Computing &amp; Applications*, 10(3):277–286.

[11] Cheng, T., Haworth, J., and Wang, J. (2012). Spatio-temporal autocorrelation of road network data. *Journal of Geographical Systems*, 14(4):389–413.

[12] Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[13] Clark, S. D., Dougherty, M. S., and Kirby, H. R. (1993). The use of neural networks and time series models for short term traffic forecasting: a comparative study. In *Transportation Planning Methods. Proceedings Of Seminar D Held At The Ptrc European Transport, Highways And Planning 21st Summer Annual Meeting (September 13-17, 1993)*.

[14] Danech-Pajouh, M. and Aron, M. (1991). Athena: a method for short-term interurban motorway traffic forecasting. *Recherche Transports Sécurité*, (6).

[15] Davis, G. A. and Nihan, N. L. (1991). Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, 117(2):178–188.

[16] Dia, H. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, 131(2):253–261.

[17] Dougherty, M. S. and Cobbett, M. R. (1997). Short-term inter-urban traffic forecasts using neural networks. *International journal of forecasting*, 13(1):21–31.

[18] Hamed, M. M., Al-Masaeid, H. R., and Said, Z. M. B. (1995). Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering*, 121(3):249–254.

[19] Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press.

[20] Haykin, S. S. (2001). *Kalman filtering and neural networks*. Wiley Online Library.

[21] Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

[22] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[23] Högberg, P. (1976). Estimation of parameters in models for traffic prediction: a non-linear regression approach. *Transportation Research*, 10(4):263–265.

[24] Hu, J., Zong, C., Song, J., Zhang, Z., and Ren, J. (2003). An applicable short-term traffic flow forecasting method based on chaotic theory. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 1, pages 608–613. IEEE.

[25] Innamaa, S. (2005). Short-term prediction of travel time using neural networks on an interurban highway. *Transportation*, 32(6):649–669.

[26] Jensen, T. and Nielsen, S. (1973). Calibrating a gravity model and estimating its parameters using traffic volume counts. In *5th Conference of Universities' Transport Study Groups, University College, London.*

[27] Jiang, X. and Adeli, H. (2005). Dynamic wavelet neural network model for traffic flow forecasting. *Journal of transportation engineering*, 131(10):771–779.

[28] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

[29] Karlaftis, M. and Vlahogianni, E. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399.

[30] Kirby, H. R., Watson, S. M., and Dougherty, M. S. (1997). Should we use neural networks or statistical models for short-term motorway traffic forecasting? *International Journal of Forecasting*, 13(1):43–50.

[31] Kumar, S. V. and Vanajakshi, L. (2015). Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, 7(3):1–9.

[32] Lan, C.-J. and Miaou, S.-P. (1999). Real-time prediction of traffic flows using dynamic generalized linear models. *Transportation Research Record: Journal of the Transportation Research Board*, (1678):168–178.

[33] Levin, M. and Tsao, Y.-D. (1980). On forecasting freeway occupancies and volumes. *Transportation Research Record*, (773).

[34] Low, D. E. (1972). A new approach to transportation systems modeling. *Traffic quarterly*, 26(3).

[35] Lv, Y., Tang, S., and Zhao, H. (2009). Real-time highway traffic accident prediction based on the k-nearest neighbor method. In *Measuring Technology and Mechatronics Automation, 2009. ICMTMA'09. International Conference on*, volume 3, pages 547–550. IEEE.

[36] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[37] Min, X., Hu, J., Chen, Q., Zhang, T., and Zhang, Y. (2009). Short-term traffic flow forecasting of urban network based on dynamic starima model. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, pages 1–6. IEEE.

[38] Min, X., Hu, J., and Zhang, Z. (2010). Urban traffic network modeling and short-term traffic flow forecasting based on gstarima model. *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1535–1540.

[39] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

[40] Nihan, N. L. and Holmesland, K. O. (1980). Use of the box and jenkins time series technique in traffic forecasting. *Transportation*, 9(2):125–143.

[41] Okutani, I. and Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1):1–11.

[42] Rice, J. and Van Zwet, E. (2004). A simple and effective method for predicting travel times on freeways. *Intelligent Transportation Systems, IEEE Transactions on*, 5(3):200–207.

[43] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[44] Smith, B. L. and Demetsky, M. J. (1994). Short-term traffic flow prediction models - a comparison of neural network and nonparametric regression approaches. *IEEE INTERNATIONAL CONFERENCE ON SYSTEMS MAN AND CYBERNETICS*, 2:1706.

[45] Smith, B. L. and Demetsky, M. J. (1997). Traffic flow forecasting: comparison of modeling approaches. *Journal of transportation engineering.*

[46] Smith, B. L., Williams, B. M., and Oswald, R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4):303–321.

[47] Smola, A. and Vapnik, V. (1997). Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

[48] Stathopoulos, A., Dimitriou, L., and Tsekeris, T. (2008). Fuzzy modeling approach for combined forecasting of urban traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, 23(7):521–535.

[49] Stathopoulos, A. and Karlaftis, M. G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C: Emerging Technologies*, 11(2):121–135.

[50] Suykens, J. A., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J., Suykens, J., and Van Gestel, T. (2002). *Least squares support vector machines*, volume 4. World Scientific.

[51] Szeto, W., Ghosh, B., Basu, B., and O'Mahony, M. (2009). Multivariate traffic forecasting technique using cell transmission model and sarima model. *Journal of Transportation Engineering*, 135(9):658–667.

[52] Tong, H. (2002). *Non-linear time series: a dynamical system approach.* Oxford University Press.

[53] Tsirigotis, L., Vlahogianni, E. I., and Karlaftis, M. G. (2012). Does information on weather affect the performance of short-term traffic forecasting models? *International Journal of Intelligent Transportation Systems Research*, 10(1):1–10.

[54] Van Der Voort, M., Dougherty, M., and Watson, S. (1996). Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5):307–318.

[55] van Hinsbergen, J. and Sanders, F. (2007). Short term traffic prediction models.

[56] Van Lint, J., Hoogendoorn, S., and van Zuylen, H. J. (2005). Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5):347–369.

[57] Van Lint, J. and Van Hinsbergen, C. (2012). Short term traffic and travel time prediction models, in artificial intelligence applications to critical transportation issues. *Transportation Research Circular, National Academies Press, Washington DC.*

[58] Vlahogianni, E. I., Golias, J. C., and Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. *Transport reviews*, 24(5):533–557.

[59] Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. *Transportation Research Part C: Emerging Technologies*, 13(3):211–234.

[60] Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19.

[61] Williams, B. (2001). Multivariate vehicular traffic flow prediction: Evaluation of arimax modeling. *Transportation Research Record: Journal of the Transportation Research Board*, 1776(25):194–200.

[62] Williams, B. M. and Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672.

[63] Wu, C.-H., Ho, J.-M., and Lee, D.-T. (2004). Travel-time prediction with support vector regression. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):276–281.

[64] Yasdi, R. (1999). Prediction of road traffic using a neural network approach. *Neural computing &amp; applications*, 8(2):135–142.

[65] Yin, H., Wong, S., Xu, J., and Wong, C. (2002). Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies*, 10(2):85–98.

[66] Yue, Y., Yeh, A. G., and Zhuang, Y. (2007). Prediction time horizon and effectiveness of real-time data on short-term traffic predictability. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 962–967. IEEE.

[67] Zeng, D., Xu, J., Gu, J., Liu, L., and Xu, G. (2008). Short term traffic flow prediction based on online learning svr. In *Power Electronics and Intelligent Transportation System, 2008. PEITS'08. Workshop on*, pages 616–620. IEEE.