Short-term traffic prediction under normal and abnormal conditions

Fangce Guo

A thesis submitted for the degree of Doctor of Philosophy of
Imperial College London

Centre for Transport Studies
Department of Civil and Environmental Engineering
Imperial College London, United Kingdom

July 2013

Abstract

Intelligent Transport Systems (ITS) is a field that has developed rapidly over the last two decades, driven by the growing need for better transport network management strategies and by continuing improvements in computing power. However, a number of ITS applications, such as Advanced Traveller Information Systems (ATIS), Dynamic Route Guidance (DRG) and Urban Traffic Control (UTC) need to be proactive rather than reactive, and consequently require the prediction of traffic state variables into the short-term future. Similarly, individual travellers can use this predictive information to plan their mobility more efficiently. This PhD thesis develops models that are able to accurately predict short-term traffic variables such as link travel time and traffic flow on urban arterial roads under both normal and abnormal traffic conditions.

This research first reviews the state of the art in data prediction applications in engineering domains especially traffic engineering and presents existing statistical and machine learning methods and their applications in relation to short-term traffic prediction. This review establishes that most existing work has focused on the apparent superiority of one individual statistical or machine learning method over another. Little attention has been paid, however, to the issues surrounding the overall structure of prediction models, in particular in relation to data smoothing and error feedback. In developing a short-term traffic prediction model, therefore, a 3-stage framework including a data smoothing step and an error feedback mechanism is

proposed. This proposed framework is applied in conjunction with five different machine learning methods to develop a range of short-term traffic prediction methods.

The proposed prediction framework is then tested under different traffic conditions using traffic data generated from a traffic simulation model of a corridor in Southampton. The prediction results show that the proposed 3-stage prediction framework can improve the accuracy of traffic prediction, regardless of the machine learning method used under both normal and abnormal traffic conditions. After demonstrating the effectiveness of predicting traffic variables using simulated data, the proposed methodology is then applied to real-world traffic data collected from different sites in London and Maidstone. These results also show that the framework can improve the accuracy of prediction regardless of the machine learning tool used. The prediction accuracy comparison shows that the proposed 3-stage prediction framework can improve the prediction accuracy for either travel time or traffic flow data under both normal and abnormal traffic conditions. In addition, the results indicate that the kNN based prediction method, when applied through the proposed framework, outperforms other selected machine learning methods under abnormal traffic conditions on urban roads. The findings suggest that, in order to arrive at a robust and accurate prediction model, attention should be paid to combining data smoothing, model structure and error feedback elements.

Declaration of Originality

At various stages during this PhD, I have been involved in collaborative efforts with both academic and industrial colleagues. In certain cases, the output of this collaboration is included in this thesis to better explain and support the research presented. In particular, my research has built upon collaborative work with my supervisors and other colleagues, working on several collaborative research papers that were presented at various conferences and submitted for journal publication. These are listed in the reference section and are all my own work.

I hereby declare that besides the collaboration referred to above I have personally carried out the work described in this dissertation.

…….

Fangce Guo

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgements

First and foremost I would like to thank my supervisors Professor John Polak and Dr Rajesh Krishnan for offering me the opportunity to study in Intelligent Transport Systems (ITS). Without their inspirational guidance, excellent supervision and financial support, this thesis would not have been accomplished.

I am very grateful to Martin Wylie of Southampton City Council, who provided me with the AIMSUN micro-simulation model of Southampton. I would like to thank Chunkin Cheung and Andy Emmonds of Transport for London for providing me with travel time data for the A40 road in London. I must also thank John Murdoch of Kent County Council and Malcolm Kersey of Jacobs for providing the traffic data for Kent used within this thesis.

I would like to thank Dr Robin North and Dr Tzu-Chang (Joe) Lee for the many useful suggestions in the early stage of my PhD research, Dr Simon Hu in explaining simulation related issues and Dr Jack Han for the many useful discussions on ITS related topics from traffic data collection to traffic estimation.

I would also like to express my gratitude to my colleagues and officemates in Room 609 and 613 and to Mrs Jackie Sime for her administrative help during the past four years.

Special thanks go to my friends, Siyi Li and Ada Hao in China, who are always there to encourage me via Skype and facetime when I need them.

Last but not least, I dedicate this work to my parents and other family members in Shenyang for their continuous support and encouragement, and to my husband Hongda for his patience and sacrifices. Without your love this thesis would never have been finished.

Contents

Abstract **2**

Declaration of Originality **4**

Copyright Declaration **5**

Acknowledgements **6**

Contents **8**

List of Figures **15**

List of Tables **21**

Chapter 1 Introduction **24**

1.1 Background 25

1.1.1 Short-term traffic prediction problem statement 26

1.1.2 Factors influencing traffic conditions 28

1.2 Research scope and objectives 29

1.2.1 Research scope 29

1.2.2 Research objectives 30

1.2.3 Research considerations 30

1.2.3.1 Prediction accuracy 30

1.2.3.2 Model robustness 31

1.2.3.3 Ease of implementation and transferability 31

1.3 Structure of this thesis 31

Chapter 2 Review of Short-term Data Prediction Methods 33

2.1 Introduction 33

2.2 Short-term traffic prediction methods 33

2.3 Factors influencing short-term traffic prediction models 35

2.3.1 Implementation context for short-term traffic prediction 35

2.3.2 Input variables in short-term traffic prediction 36

2.3.3 Input data resolution in short-term traffic prediction 38

2.3.4 Prediction steps in short-term traffic prediction 38

2.3.5 Seasonal temporal and spatial patterns in short-term traffic prediction 39

2.3.6 Traffic conditions in short-term traffic prediction 39

2.3.6.1 Traffic prediction under normal traffic conditions 40

2.3.6.2 Traffic prediction under abnormal traffic conditions 41

2.3.7 Summary 43

2.4 Short-term data prediction in other domains 43

2.4.1 Short-term data prediction in finance 44

2.4.2 Short-term data prediction in hydrology 46

2.4.3 Short-term data prediction in energy 47

2.4.4 Summary 49

2.5 Review of statistical and machine learning methods in traffic prediction 50

2.5.1 Historical average 51

2.5.2 Statistical methods 51

2.5.3 Grey System Model (GM) 54

2.5.4 Kalman filter (KF) 58

2.5.5 Neural Network (NN) 60

2.5.6 K-Nearest Neighbour method (kNN) 65

2.5.7 Kernel Smoothing (KS) 70

2.5.8 Spinning Network (SPN) 71

2.5.9 Support Vector Regression (SVR) 73

2.5.10 Random Forests (RF) 76

2.6 Summary of existing traffic prediction methods 80

2.7 Conclusions 89

Chapter 3 Short-term Traffic Prediction Frameworks 90

3.1 Background 90

3.2 Data smoothing 92

3.2.1 Overview of formal data smoothing approaches 93

3.2.2 The SSA method 94

3.2.3 Prediction framework with data smoothing 99

3.3 Machine learning methods 102

3.3.1 Introduction 102

3.3.2 kNN 102

3.3.3 GM 106

3.3.4 NN 107

3.3.5 RF 110

3.3.6 SVR 111

3.4 Additional input variables 112

3.4.1 Background 112

3.4.2 Error feedback structure 115

3.5 Quantification of prediction accuracy 119

3.6 Summary 120

Chapter 4 Evaluation of Proposed Traffic Prediction Frameworks Based on Simulation Experiments 121

4.1 Background 121

4.2 Microscopic traffic simulation 122

4.2.1 Selection of traffic simulator 122

4.2.2 Benefits and challenges of AIMSUN simulator 126

4.2.2.1 Benefits of simulation 126

4.2.2.2 Weaknesses of simulation 127

4.3 Description of the simulation setup used 129

4.3.1 Scenario design in simulation experiments 129

4.3.2 Simulation model settings 131

4.3.2.1 Road network layout in simulation 131

4.3.2.2 Traffic demand 133
4.3.2.3 Signal control 135
4.3.2.4 Model calibration and validation 136
4.3.3 Outputs of simulation 140
4.4 Prediction accuracy under normal traffic conditions - Scenario 1 143
4.5 Prediction accuracy under abnormal traffic conditions 147
4.5.1 Scenario 2: One-lane closure in simulation 149
4.5.1.1 One lane closure during the off-peak period 149
4.5.1.2 One lane closure during the peak period 152
4.5.2 Scenario 3: Two-lane closure in simulation 156
4.5.2.1 Two-lane closure during the off-peak period 156
4.5.2.2 Two-lane closure during the peak period 157
4.5.3 Further analysis under abnormal traffic conditions 159
4.5.3.1 Different data resolution 159
4.5.3.2 Comparison with the Kalman filter based method 160
4.6 Summary 162

Chapter 5 Short-term Traffic Prediction Using Real-world Traffic Data..... 164

5.1 Introduction 164
5.2 Real-world traffic data 164
5.2.1 Link travel time data 165
5.2.1.1 Travel time data in London 165

5.2.1.2 Travel time data in Maidstone 167

5.2.2 Traffic flow data in London 168

5.3 Short-term traffic prediction under normal traffic conditions 172

5.3.1 Short-term travel time prediction using data from the A40 road in London under normal traffic conditions 172

5.3.2 Short-term traffic flow prediction using data from the Russell Square corridor in London under normal traffic conditions 177

5.3.3 Short-term traffic flow prediction using data from the Marylebone corridor in London under normal traffic conditions 181

5.4 Short-term traffic prediction under abnormal traffic conditions 185

5.4.1 Short-term travel time prediction using data from the A40 road in London under abnormal traffic conditions 186

5.4.2 Short-term travel time prediction using data from Maidstone under abnormal traffic conditions 192

5.4.3 Short-term traffic flow prediction using data from London Marylebone corridor under abnormal traffic conditions 195

5.5 Conclusions 199

Chapter 6 Conclusions and Future Research 201

6.1 Revisiting the objectives 201

6.2 Contributions 204

6.3 A note on practical implementation 205

6.4 Future research 206

Appendiex A Conceptual Impacts of Traffic Variables Caused by Abnormal Traffic Conditions 210

A.1 Basic queuing theory 210

A.2 Queuing theory in traffic modelling interrupted by abnormal conditions 212

Appendiex B Traffic Data Cleaning Methods 216

B.1 LCAP data cleaning methods 216

B.2 ANPR data cleaning methods used in Maidstone 217

Appendiex C Main Traffic Modelling in AIMSUN 219

C.1 Car-following model 219

C.2 Lane changing model 220

C.3 Gap Acceptance Model 222

References 223

List of Figures

Figure 1.1: Illustration of the travel time prediction problem as a time-space diagram (adapted from Van Lint (2004))....27

Figure 2.1: Example of intraday order arrival rates in the foreign exchange market (Source: Bollerslev & Domowitz (1993))....45

Figure 2.2: Example of rainfall series (solid line without marker) (Source: Hong (2008))....46

Figure 2.3: Example of daily load data pattern within a week (Source: Espinoza *et al.* (2007))....48

Figure 2.4: Algorithmic loop of the KF (Source: Thacker & Lacey (1996)) 59

Figure 2.5: Process of a single neuron (Source: Rokach (2010)) 62

Figure 2.6: General architectures of feed-forward networks (Source: Mitchell (1997)) 63

Figure 2.7: General structure of kNN based prediction method....66

Figure 2.8: Structure of spinning rings (Source: Huang & Sadek (2009)) 72

Figure 2.9: A general architecture of RF (Source: Verikas *et al*., (2011)) 78

Figure 2.10: Flow-chart of the RF process 79

Figure 3.1: General 3-stage framework for traffic prediction....92

Figure 3.2: Flow-chart of a basic SSA method (adapted from Golyandina *et al.* (2001)) 98

Figure 3.3: Traffic data, smoothed series and residuals....99

Figure 3.4: Flow-chart for the prediction framework using data smoothing....101

Figure 3.5: Process of NN based method for prediction problems 108
Figure 3.6: Flow-chart of the proposed 3-stage short-term traffic prediction framework 118
Figure 4.1: Scenario design of simulation experiments 131
Figure 4.2: The selected link in Southampton AIMSUN network 132
Figure 4.3: A representation of the interaction between traffic simulation and signal control 136
Figure 4.4: Plots of averaged, maximum and minimum values of traffic profiles in the training dataset 142
Figure 4.5: (a) An example of a travel time profile during 05:00 – 22:00 under normal traffic conditions and (b) An example of a travel time profile during 05:00 – 22:00 under abnormal traffic conditions in Scenario 2 143
Figure 4.6: MAPE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks under normal traffic conditions in Scenario 1 146
Figure 4.7: RMSE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks under normal traffic conditions in Scenario 1 147
Figure 4.8: Location of the lane closure in simulation 148
Figure 4.9: One-lane closure in area A 149
Figure 4.10: MAPE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the off-peak period 150
Figure 4.11: RMSE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the off-peak period 152

Figure 4.12: MAPE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the peak period 155
Figure 4.13: RMSE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the peak period 155
Figure 4.14: Two-lane closure in area A 156
Figure 5.1: Link 1309 on the A40 road in London (Source: Google Earth)..... 167
Figure 5.2: Selected Link 99AL0005D in Maidstone (Source: Google Earth) 168
Figure 5.3: The Russell Square corridor (Source: Google Maps) 170
Figure 5.4: The Marylebone Road corridor (Source: Google Maps)..... 170
Figure 5.5: MAPE for five machine learning methods and three prediction frameworks of one-step ahead prediction under normal traffic conditions on link 1309 of the A40 road in London..... 174
Figure 5.6: MAPE for five machine learning methods and three prediction frameworks of multi-step ahead prediction under normal traffic conditions on link 1309 of the A40 road in London..... 174
Figure 5.7: Travel time prediction performance using RF with the 1-stage framework on the A40 road in London under normal traffic conditions 175
Figure 5.8: Travel time prediction performance using RF with the 2-stage framework on the A40 road in London under normal traffic conditions 176
Figure 5.9: Travel time prediction performance using RF with the 3-stage framework on the A40 road in London under normal traffic conditions 176

Figure 5.10: MAPE for five machine learning methods and three prediction frameworks of one-step ahead prediction under normal traffic conditions on the Russell Square corridor 179
Figure 5.11: MAPE for five machine learning methods and three prediction frameworks of multi-step ahead prediction under normal traffic conditions on the Russell Square corridor 179
Figure 5.12: Traffic flow prediction performance using NN with the 1-stage framework on the Russell Square corridor in London under normal traffic conditions 180
Figure 5.13: Traffic flow prediction performance using NN with the 2-stage framework on the Russell Square corridor in London under normal traffic conditions 180
Figure 5.14: Traffic flow prediction performance using NN with the 3-stage framework on the Russell Square corridor in London under normal traffic conditions 181
Figure 5.15: MAPE for five machine learning methods and three prediction frameworks for one-step ahead prediction under normal traffic conditions on the Marylebone corridor in London 183
Figure 5.16: MAPE for five machine learning methods and three prediction frameworks for multi-step ahead prediction under normal traffic conditions on the Marylebone corridor 183
Figure 5.17: Traffic flow prediction performance using RF with the 1-stage framework on the Marylebone corridor in London under normal traffic conditions 184
Figure 5.18: Traffic flow prediction performance using RF with the 2-stage framework on the Marylebone corridor in London under normal traffic conditions 184

Figure 5.19: Traffic flow prediction performance using RF with the 3-stage framework on the Marylebone corridor in London under normal traffic conditions 185
Figure 5.20: Location of the abnormal event on 21st December 2010 on link 1309 of the A40 road in central London (Source: Google Maps).....187
Figure 5.21: MAPE for five machine learning methods and three prediction frameworks during the abnormal period using data from link 1309 on the A40 road 189
Figure 5.22: Comparison of observed and predicted travel time using three prediction frameworks with the kNN based method (a) Prediction comparison during the day when the abnormal event occurred and (b) Prediction comparison during the abnormal period on the testing day.....190
Figure 5.23: Travel time prediction performance using kNN with the 1-stage framework on the A40 road in London under abnormal traffic conditions.....190
Figure 5.24: Travel time prediction performance using kNN with the 2-stage framework on the A40 road in London under abnormal traffic conditions.....191
Figure 5.25: Travel time prediction performance using kNN with the 3-stage framework on the A40 road in London under abnormal traffic conditions.....191
Figure 5.26: Location of abnormal event on 26th August 2011 on Link 99AL0005D in the Maidstone area of Kent (Source: Google Maps) 192
Figure 5.27: MAPE for five machine learning methods and three prediction frameworks during the abnormal period using data from Link 99AL0005D in Maidstone.....194
Figure 5.28: Comparison of observed and predicted travel time using three prediction frameworks with the kNN based method during the abnormal period.....194

Figure 5.29: Time-series plot between the profiles under normal traffic conditions and abnormal traffic conditions 196
Figure 5.30: MAPE for five machine learning methods and three prediction frameworks during the abnormal period using data from the Marylebone corridor.. 197
Figure 5.31: Comparison of observed and predicted traffic flow using three prediction frameworks with the kNN based method during the abnormal period 198
Figure 5.32: Traffic flow prediction performance using kNN with the 3-stage framework on the Marylebone corridor under abnormal traffic conditions 198
Figure 6.1: Summary of prediction implementation 205
Figure A.1**:** A general queuing system 211
Figure A.2. Vehicle queuing-capacity-time diagram 214
Figure C.1: Lane changing zones 221

List of Tables

Table 2.1: Summary of the KF recursive algorithm (Source: Thacker & Lacey (1996)) 59

Table 2.2: Equations of distance metrics (Source: Robinson (2005)) 67

Table 2.3: Categorisation of available literature in existing traffic prediction models 82

Table 2.4: Characteristics of reviewed statistical and machine learning methods in short-term traffic prediction 85

Table 2.5: Comparison of reviewed statistical/machine learning methods in traffic prediction 88

Table 4.1: Main features of three simulators 124

Table 4.2: Attributes and levels used in Scenario 2 and Scenario 3 130

Table 4.3: An example of the O-D matrix in the simulation model from during 07:00 to 07:15 (all values are in vheicles/hour) 134

Table 4.4: Important parameters in AIMSUN 138

Table 4.5: Traffic data in scenarios used for framework evaluation 142

Table 4.6: Prediction accuracy of link travel time using three different frameworks with five machine learning methods under normal traffic conditions in Scenario 1 . 145

Table 4.7: Averaged prediction accuracy of link travel time using three different frameworks with five machine learning methods under normal traffic conditions ... 146

Table 4.8: Comparison of prediction accuracy when one lane blocked during the off-peak period 151

Table 4.9: Comparison of prediction accuracy when one lane blocked during the peak period 154
Table 4.10: Comparison of prediction accuracy when two lanes were blocked during the off-peak period 157
Table 4.11: Comparison of prediction accuracy when two lanes were blocked during the peak period 158
Table 4.12: Comparison of prediction using data at 15-minute granularity using kNN with three different prediction frameworks 160
Table 4.13: Comparison of prediction accuracy between the kNN and Kalman filter based methods under abnormal traffic conditions 161
Table 5.1: Comparison of prediction accuracy of link travel time on the A40 road in London using three different frameworks with five machine learning methods under normal traffic conditions 173
Table 5.2: Comparison of prediction accuracy of traffic flow on the Russell Square corridor using three different frameworks with five machine learning methods under normal traffic conditions 178
Table 5.3: Comparison of prediction accuracy of traffic flow on the Marylebone corridor using three different frameworks with five machine learning methods under normal traffic conditions 182
Table 5.4: Comparison of prediction accuracy of travel time from link 1309 on the A40 road using three different frameworks with five machine learning methods during the abnormal period 188
Table 5.5: Comparison of prediction accuracy of travel time from Link 99AL0005D in Maidstone using three different frameworks with five machine learning methods during the abnormal period 193

Table 5.6: Comparison of prediction accuracy of traffic flow on the Marylebone corridor using three different frameworks with five machine learning methods during the abnormal period 196
Table A.1: Description of parameters in Figure A.2 215
Table A.2: Estimated traffic characteristics using queuing theory (Source: Qin & Smith (2001)) 215
Table B.1: Methods to patch missing data in LCAP 217
Table C.1: Algorithm used in gap acceptance model (Source: TSS (2004)) 222

Chapter 1 Introduction

Intelligent Transport Systems (ITS) is a field that has developed rapidly over the last two decades. It applies Information and Communications Technology (ICT), such as data processing and advanced data mining methods, and advances in computer hardware to the operation and management of transport networks. The overall function of ITS is "*to improve decision making, often in real time, by transport network controllers and other users, thereby improving the operation of the entire transport system*" (Miles & Chen, 2004).

There are a number of ITS applications such as Advanced Traffic Management Systems (ATMS), Advanced Traveller Information Systems (ATIS), Dynamic Route Guidance (DRG) and Urban Traffic Control (UTC) (Miles & Chen, 2004). One of the key uses of these systems is to ensure optimal efficiency of the transport network and to relieve traffic congestion. From a system point of view, a system is reactive when real-time data about current traffic conditions are utilised; a control system is proactive when predictive information about near future traffic conditions is utilised. ITS applications need to be proactive rather than reactive in order to help network managers develop strategies to mitigate network problems and avoid undesirable effects. The main purpose of short-term traffic prediction is to help transport network managers develop more sophisticated strategies to anticipate and mitigate network problems so as to alleviate network congestion. Similarly, individual travellers can also use predictive information to choose the most efficient transport option (e.g., route, mode or time of day) to avoid traffic congestion and reduce the travel time of

their journey. These applications therefore require the prediction of traffic state variables into the short-term future.

1.1 Background

Accurate and robust short-term traffic prediction is one of the key components for ITS applications. Short-term traffic prediction can be defined as the process of estimating the anticipated traffic conditions in the short-term future given historical and current traffic information (Vlahogianni *et al.*, 2004). In this thesis, the phrase *short-term* is used to refer to a time horizon of up to one hour. Real-time or near-real-time data in combination with historical information usually form the basis for the prediction of future traffic variables.

A large amount of research has been concerned with the problem of urban traffic congestion and its economic and environmental impacts. Congestion usually occurs when traffic demand exceeds road capacity. In many cities, it is difficult to construct more roads due to economic, environmental and physical constraints. Because of this limitation, ITS applications are becoming an increasingly important alternative to avoid or mitigate traffic congestion. According to the ITS Handbook (Miles & Chen, 2004), an important benefit of ITS is "*to relieve congestion using traffic management tools to ensure maximum efficiency of the road networking*" (Miles & Chen, 2004). Managing Urban Traffic Congestion – Summary Document (OECD/ECMT, 2007) states that "*urban regions will never be free of congestion. Road transport policies, however, should seek to manage congestion on a cost-effective basis with the aim of reducing the burden that excessive congestion imposes upon travellers and urban dwellers throughout the urban road network.*"

Traffic congestion can be categorised into recurring congestion and non-recurring congestion. Recurrent congestion is easy to predict by traffic managers because it is caused by high volumes of vehicles at specific locations during the same time period. Non-recurrent and unexpected congestion caused by abnormal occurrences is a significant problem for transport network managers. Although it is not possible to predict the occurrence of non-recurrent congestion, it is possible, and helpful, to predict traffic variables during unexpected congestion after congestion has occurred.

1.1.1 Short-term traffic prediction problem statement

Traffic prediction focuses on estimating the value of variables such as traffic flow and travel time in the future based on known data and information. Short-term traffic prediction can then be defined as the process of estimating the anticipated traffic conditions in the short-term future given historical and current traffic information (Vlahogianni *et al.*, 2004). Real-time or near-real-time data in combination with historical information usually form the basis for the prediction of future traffic variables. Therefore, a formal mathematic statement of one-step ahead short-term traffic prediction at time t, is given as follows:

$$\hat{x}(t+T) = f(x(t), x(t-T), \dots, x(t-mT)) \tag{1.1}$$

where,

T: time interval for prediction

$x(t)$: real-time or near-real-time traffic variable measured at time *t*

m: a positive integer

$\hat{x}(t+T)$: predicted traffic variable at $(t+T)$.

A time-space diagram can also be used to describe the short-term traffic prediction problem. Figure 1.1 is an example of an illustration of the travel time prediction problem by means of a time-space diagram.

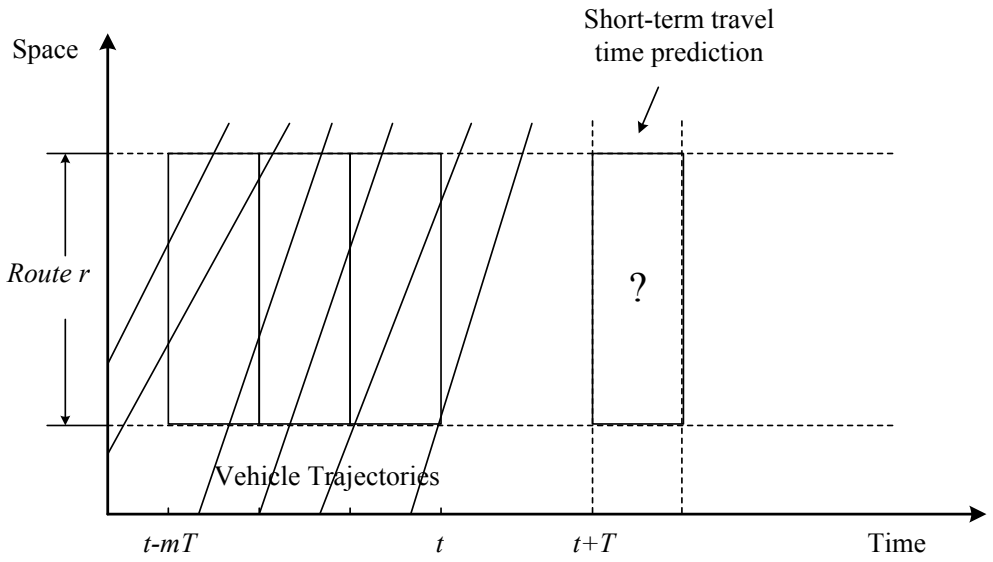


Figure 1.1: Illustration of the travel time prediction problem as a time-space diagram (adapted from Van Lint (2004))

In the short-term traffic prediction problem, traffic data sources are the most critical and basic component. The prediction accuracy of a traffic prediction model depends on the level of consistency or agreement between the characteristics of the traffic datasets used for model development and those of the new data observed in the real world. Traffic variables are subject to occasional, abrupt disturbances that can change the underlying dynamics and the stability of the data generation process. Abnormal events such as traffic incidents and accidents may lead to sudden changes in traffic speed, a reduction in the road capacity in a traffic network and an increase in the travel time between two locations. Traffic conditions and traffic patterns may change not only as a consequence of planned and unplanned events such as roadworks, incidents and accidents, but also due to seasonal effects such as holidays, and each of

these circumstances may affect the performance of short-term traffic prediction. The conceptual impact of traffic variables caused by abnormal traffic conditions is introduced in Appendix A and the factors that influence traffic conditions and traffic variables are categorised and discussed in the next subsection.

1.1.2 Factors influencing traffic conditions

Traffic conditions are strongly related to traffic variables such as travel time and traffic flow and, therefore, awareness of traffic conditions is essential to enable traffic engineers to monitor current traffic networks and the operational performance of traffic facilities. The factors influencing traffic conditions can be categorised into two groups: traffic demand and traffic supply. Traffic demand influences the amount of vehicles or travellers using a road network; traffic supply reflects the available capacity of the road facility and infrastructure.

Traffic demand factors are expressed in terms of seasonal effects, network effects, population characteristics and traffic information (Van Lint, 2004). Traffic supply reflects the capacity of the road facility and is affected by planned and unplanned events, weather conditions, road geometry and dynamic traffic management (Van Lint *et al.*, 2008). In practice, most of these factors overlap and are dependent on each other (Van Lint *et al.*, 2008). For example, some factors affect not only traffic supply but also traffic demand and vice versa; adverse weather conditions may change travellers' routes, modes and departure times as well as reduce the road capacity.

1.2 Research scope and objectives

1.2.1 Research scope

The previous section provided the background for this PhD research. In this section, the main research scope is summarised. Firstly, this research will focus on short-term not long-term traffic prediction. Short-term and long-term traffic prediction entail differences in both the nature of the problem, the requirements of prediction models and the application context (Van Lint, 2004). Short-term traffic prediction makes use of current and near-past traffic conditions. Long-term traffic prediction relies on model assumptions such as historical patterns. Secondly, this research will address short-term traffic prediction on urban (signal controlled) arterial roads, where congestion has more influence on people, the environment and the economy. The choice of urban networks is also motivated by the importance and the limitation of applications on urban roads and will be discussed in more detail in Chapter 2. Thirdly, the aim is to develop traffic prediction models capable of rendering good prediction outcomes under both normal and abnormal traffic conditions. Finally, this research will focus on advanced machine learning methods to tackle the short-term traffic prediction problem. This choice is motivated by the complexity of the prediction problem under normal and abnormal traffic conditions and a more detailed discussion of this choice will be presented in Chapter 2. Hence, this research will focus on short-term traffic prediction on urban arterial roads under both normal and abnormal conditions.

1.2.2 Research objectives

The overall aim of this thesis is to develop models that are able accurately to predict short-term traffic variables such as travel time and traffic flow on urban arterial roads under both normal and abnormal traffic conditions. The specific research objectives to fulfil this scope are:

- Develop traffic prediction models to improve machine learning methods based on a more comprehensive prediction framework;
- Develop robust models to accurately predict traffic during both normal and abnormal traffic conditions on urban arterial roads;
- Develop traffic prediction models that can be easily implemented without laborious calibration and maintenance, and that have the quality of location transferability; and
- Develop methods to provide both one-step and multi-step ahead traffic prediction.

1.2.3 Research considerations

1.2.3.1 Prediction accuracy

The predicted values generated by a forecasting method should be as close to actual values as possible. Good traffic prediction models should have a small error bias and variance. Inaccurately predicted traffic information will cause drivers to be less confident about the provided information, and therefore the accuracy of predicted information is a key factor that determines the impact of ITS applications.

1.2.3.2 Model robustness

Robustness is another important consideration for a traffic prediction model. Most short-term traffic prediction models learn from past data and make use of historical patterns to make their predictions. Traffic patterns can deviate from historic trends when planned events, such as road works, and unplanned events, such as incidents and accidents, occur on the road network. Such scenarios are commonly referred to as abnormal traffic conditions. A robust prediction model should be accurate during both normal and abnormal conditions of traffic. In this PhD study, model robustness will be evaluated using traffic data collected under a wide range of traffic conditions.

1.2.3.3 Ease of implementation and transferability

Implementation efficiency means that a short-term traffic prediction model should be easily implementable at different locations. In other words, the developed models should have modest data and computational requirements, and are as far as possible transferable across locations. Transferability in this context refers to the ability of a model to work well without extensive site-specific calibration and without the need for the use of detailed data on for example, the geometrical properties of the road system, the number of lanes, the locations of the installed detection sensors and without different information on the characteristics of the implemented signal control plans. In summary, prediction models should be easily implementable and transferrable. Model implementation and transferability across locations will be investigated using traffic data collected from different road layouts under traffic signal control systems.

1.3 Structure of this thesis

This PhD thesis is organised as follows:

Chapter 1 presents the background of this research and describes its scope and objectives.

Chapter 2 provides a review of the literature on short-term traffic prediction and a discussion of the state-of-the-art techniques in short-term prediction in transport and other comparable fields.

Chapter 3 presents the proposed traffic prediction frameworks. The prediction frameworks are generic and can work using a number of machine learning methods.

Chapter 4 uses traffic data generated from a series of simulation experiments to test the proposed traffic prediction frameworks described in Chapter 3 under different traffic conditions. The three proposed frameworks are comprehensively tested using simulated traffic data with five different machine learning methods.

After testing the proposed prediction frameworks in a controlled setting, the frameworks are evaluated in Chapter 5 using real-world traffic data from London and Maidstone.

Chapter 6 summarises the findings of this research and suggests avenues for further research.

Chapter 2 Review of Short-term Data Prediction Methods

2.1 Introduction

In the previous chapter the conceptual basis of the traffic prediction problem was explained. This chapter moves on to review the application of data prediction in the engineering domain, especially traffic engineering. The literature on existing statistical and machine learning methods and their application of traffic variable prediction is reviewed and discussed in this chapter.

The chapter starts with an overview of existing short-term traffic prediction models with different aspects of these being reviewed and discussed in Section 2.2. In Section 2.3 short-term data prediction in the domains of finance, hydrology and energy is reviewed in order to identify similarities and differences with transport. Then, in Section 2.4 ten statistical and machine learning methods and their applications in traffic prediction are presented. A concise discussion of the advantages and weaknesses of these methods and the potential areas, in which improvements might be realised, is provided in Section 2.5.

2.2 Short-term traffic prediction methods

Generally, traffic prediction methods are divided into two distinct strands, namely traffic process based and statistical/machine learning tool based prediction methods.

Traffic process based prediction methods use simulation of the traffic system itself, including the traffic flow, road network and signal control plan. This approach

considers the detailed simulation of the activities and decision making of drivers on the road network. Microscopic traffic models focus on the prediction of individual vehicle trajectories based on assumptions of driver-behaviour (e.g., Ben-Akiva *et al.*, 1998). Macroscopic traffic prediction models centre on the prediction of a stream of traffic based on analogies of vehicular traffic flow with fluid and gas-dynamics (Van Lint, 2004). The main advantages of these traffic process based methods are that *'they allow inclusion of traffic control measures (ramp metering, routing, traffic lights, and even traffic information) in the prediction, and that they provide full insight into the locations and causes of possible delays on the road network of interest'* (Van Lint, 2004). On the other hand, the disadvantages of traffic process based methods include the computational complexity of parameter calibration and the maintenance of simulated traffic models. Furthermore, the predictive quality of traffic process based methods is strongly influenced by the quality of estimated traffic demands in the real-time application. Fuller discussions of the advantages and disadvantages of traffic process based methods in short-term traffic prediction can be found in Algers *et al.* (1997); Van Lint (2004); Tam & Lam (2009).

This PhD research, however, focuses on the use of statistical and computational machine learning methods to provide short-term traffic prediction. The main difference between machine learning methods and traffic process based methods is that machine learning methods can consider traffic processes as black boxes and learn the relationship between inputs in order to predict traffic variables directly. Not only are these methods less complicated and burdensome to implement, they may also potentially enable the prediction process to adapt more easily to normal or abnormal traffic regimes. A detailed description of statistical and machine learning methods can be found in Section 2.5.

Ideally, in order to investigate the relative performance for short-term traffic prediction of traffic process based methods and statistical and computational machine learning methods the same real traffic data should be used. Such a systematic comparison of the two methods cannot be found in existing literature, however.

2.3 Factors influencing short-term traffic prediction models

The field of short-term traffic prediction is complicated, because there are many factors that may influence the performance of short-term traffic prediction models. These include the implementation context, input traffic variables, input traffic data resolution (min), traffic prediction steps, input traffic data, spatio-temporal patterns and traffic conditions. Each of these factors are reviewed and discussed in the following subsections.

2.3.1 Implementation context for short-term traffic prediction

Generally, the implementation context for short-term traffic prediction is categorised into two groups, highway (freeway and motorway) and urban arterial road. The principal difference between these two categories is that on urban arterial roads the traffic is interrupted by controlled or uncontrolled intersections (Van Lint, 2004). Another difference is that the spatio-temporal characteristics of traffic on urban arterial roads are more complex than those of highway networks (Van Lint, 2004). Most examples of prediction models are developed for highways and are aimed at operating as traveller information systems in ITS applications (Vlahogianni *et al.*, 2004). Some examples of short-term traffic prediction on highways can be found in Ahmed & Cook (1979), Levin & Tsao (1980), Davis & Nihan (1991), Kirby *et al.* (1997), Smith & Demetsky (1997), Park *et al.* (1998), Abdulhai *et al.* (1999), Park &

Rilett (1999), Smith et al. (2002), Clark (2003), Sun *et al.* (2003), Williams & Hoel (2003), Ishak & Alecsandru (2004), Wu *et al.* (2004), Turochy (2006), Zheng *et al.* (2006) and Castro-Neto *et al.* (2009).

Most previous research into short-term traffic prediction has been focused on the highway context. The application of traffic prediction on urban arterial roads is both more uncertain and more complex (Vlahogianni *et al.*, 2004), and urban networks are also not as comprehensively covered by measurement equipment as freeway networks (Hu, 2011). Some examples of short-term traffic prediction in urban networks include Innamaa (2000), Huang & Ran (2002), Krishnan (2008), Stathopoulos & Karlaftis (2003), Ghosh *et al.* (2007) and Tam & Lam (2009). Van Lint (2004) has stated that there should be more research focusing on the development of traffic models for urban traffic prediction, which can be used in both traveller information systems and urban traffic control systems for transport network managers.

According to statistics from Transport for London (2007), the length of motorways in London is 60 km, while the length of principal/minor roads is 14,866 km. It would therefore, evidently, be useful to develop prediction models for urban networks but, unfortunately, there are only a few studies concerned with traffic prediction on urban arterial roads (Krishnan, 2008; Hu, 2011). One aim of this thesis, therefore, is to develop a short-term traffic prediction model for use in urban networks.

2.3.2 Input variables in short-term traffic prediction

The most commonly used traffic variables in prediction models include traffic flow, occupancy, speed and travel time. Variables of traffic flow, occupancy and speed are point-measurement data collected by point-detection devices such as loop detectors

and laser detectors. Compared to traffic occupancy and speed, traffic flow is more suitable for the description of traffic state and dominates the field of traffic prediction using point-measurement data (Vlahogianni *et al.*, 2004). Levin & Tsao (1980) demonstrated that traffic state prediction using traffic flow is more stable than those using occupancy. However, traffic occupancy as an indicator of traffic states was suggested in the short-term traffic prediction model of Lin *et al.* (2002), while Innamaa (2000) attempted to use both traffic flow and mean speed as model inputs to predict 15-minute ahead traffic flow. The raw traffic flow and mean speed data used in Innamaa's study was measured by ILDs.

Travel time prediction is another popular research area, since the concept of travel time is easily understood by both transportation engineers and travellers. Travel time is used to describe the journey time between two fixed points along roads. Nevertheless, short-term prediction of travel time is strongly connected to the availability of appropriate data. In traffic surveillance networks, travel time can be directly measured by advanced sensing and vehicle identification techniques using active test vehicles, passive probe vehicles, or number plate matching. In these cases, models can directly predict travel time using link-time measurements (Park *et al.*, 1998). Sometimes, travel time may be estimated or indirectly inferred from traffic variables such as traffic flow, occupancy and spot speed, measured by loop detectors, laser detectors or other point based sensors. In this case, the performance of travel time prediction is based on the capability of predicting space mean speed. Generally speaking, link-measurement approaches can collect more accurate travel-time data, but point-measurement approaches can be deployed more cost effectively (Wu *et al.*, 2004).

2.3.3 Input data resolution in short-term traffic prediction

Dougherty & Cobbett (1997) stated that input data resolution is an important element in short-term traffic prediction because it may affect the quality of information describing traffic states. Traffic data should be available with a sampling frequency sufficient to capture the dynamics of traffic in prediction models. Abdulhai *et al.* (2002) examined the accuracy of short-term traffic prediction using different data sampling frequencies. Their results showed that data at a high resolution is very noisy, and this may decrease the prediction accuracy. Although traffic data with low temporal resolution may miss valuable traffic information that could potentially influence traffic states, it can improve the prediction computational efficiency. Ideally, the granularity of traffic data should be dynamically controlled based on the current traffic state. For example, a low sampling frequency should be used under normal free-flow traffic conditions in which the recurrent traffic pattern does not suddenly change; a high sampling rate should be applied during traffic accident and incident conditions to quickly monitor current traffic states and the change in traffic patterns. Because of the limitations of the measurement equipment used for traffic data collection, the sampling frequency tends to be configured at a fixed rate with most traffic data, such as traffic flow, obtained from loop detectors being aggregated into 15-minute periods (e.g., Williams & Hoel, 2003). The data resolution of travel time is commonly 5 minutes in the literature (e.g., Park & Rilett, 1999; Tam & Lam, 2009).

2.3.4 Prediction steps in short-term traffic prediction

Prediction step is related to prediction horizon. Vlahogianni *et al.* (2004) stated that '*the prediction step represents the time interval upon which the forecasts are made and indicates the frequency of predictions in the forecasting horizon'*. Specifically,

the prediction accuracy may decrease with an increase of prediction steps. Most attempts at short-term traffic prediction are one-step ahead prediction (Castro-Neto *et al.*, 2009; Ghosh *et al.*, 2007; Smith *et al*., 2002; Smith & Demetsky, 1997; Stathopoulos & Karlaftis, 2003; Williams & Hoel, 2003). Multi-step ahead prediction is less common (Abdulhai *et al*., 1999; Huang & Sadek, 2009; Kamarianakis & Prastacos, 2003; Krishnan, 2008). In this research, the intention is to develop a short-term traffic prediction model that covers both one-step and multi-step ahead prediction.

2.3.5 Seasonal temporal and spatial patterns in short-term traffic prediction

Theoretically, the temporal and spatial relationship of input data is used in short-term traffic prediction models to improve predictive accuracy. Many researchers have discussed this issue, such as Clark (2003), Vlahogianni *et al.* (2004) and Krishnan & Polak (2008). Traffic variables such as traffic flow and travel time have a seasonal trend at a daily and weekly level (Krishnan, 2008). Traffic data from upstream locations can also provide additional information to predict traffic data downstream. Various traffic prediction models under free-flow traffic conditions make use of this relationship.

2.3.6 Traffic conditions in short-term traffic prediction

From a traffic condition point of view, short-term traffic data prediction models can be principally divided into two categories: traffic prediction concerned with normal traffic conditions and traffic prediction under abnormal traffic conditions.

In this research, normal traffic conditions mean that there are no special occurrences which might significantly change the recurrent traffic pattern (Castro-Neto *et al.*, 2009). Traffic conditions are sometimes affected by planned events, such as road works and holidays, and unplanned incidents and accidents, resulting in abnormal traffic conditions (Abbas *et al.*, 2005; Venkatanarayana & Smith, 2008; Castro-Neto *et al.*, 2009). Representative models of short-term traffic prediction grouped based on the type of traffic states on the testing day, and on normal and abnormal traffic conditions are presented below.

2.3.6.1 Traffic prediction under normal traffic conditions

Most of work related to traffic prediction during normal conditions makes use of recurrent traffic information. The main assumption of these studies is that the recorded traffic data that had been influenced by abnormal conditions such as planned and unplanned events should be detected and separated from the database used in traffic prediction applications. Only traffic data during normal traffic conditions are used in experiments.

From the 1970s, ARIMA time series models (Box & Jenkins, 1970) have been widely used for short-term traffic prediction under typical normal conditions (e.g., Ahmed & Cook, 1970; Hamed *et al.*, 1995; Williams *et al.*, 1998; Williams & Hoel, 2003). This model applies a statistical approach to obtain information from the past and current data of a series. It then uses this information to predict the future values. The Kalman filter has also been used for traffic prediction for more than two decades (e.g., Okutani & Stephanedes, 1984) with the assumption of no occurrence of planned or unplanned events. Stathopoulos & Karlaftis (2003) used 3-minute interval traffic flow data collected from upstream urban arterial streets in Athens to predict traffic flow at the downstream locations during normal traffic conditions. More recently,

more machine learning methods such as Neural Network (NN) (e.g., Dougherty, 1995) and the k-Nearest Neighbour (kNN) (e.g., Davis & Nihan, 1991) approaches have been used for prediction under normal traffic conditions. The most recent generation of machine learning tools applied to short term traffic forecasting include Support Vector Regression (e.g., Kim, 2003; Sapankevych & Sankar, 2009), Grey System Model (e.g., Guo *et al.*, 2012a) and Random Forests (e.g., Leshem & Ritov, 2007).

2.3.6.2 Traffic prediction under abnormal traffic conditions

Much effort has been put into addressing traffic prediction under normal traffic conditions. Short-term traffic prediction problems are relatively simplistic, however, since they avoid prediction under abnormal conditions. Abnormal traffic conditions such as non-recurrent traffic congestion, which might be caused by planned events such as road works or unplanned events such as incidents or accidents, cannot be neglected. In order to expand the scope of the potential application of traffic prediction models, an increasing amount of research focused on prediction under abnormal traffic conditions for real-world ITS applications. Short-term prediction is arguably more important during abnormal conditions because of uncertainty about how the traffic state will evolve into the future. Over the past ten years, therefore, there have been a number of attempts to develop prediction models for abnormal conditions.

Weather may cause abnormality in traffic, and weather information such as weather condition, visibility, temperature and moisture can be easily collected. Some researchers, therefore, have included weather factors into short-term traffic prediction models under abnormal traffic conditions. For example, Huang & Ran (2002) used a traffic prediction model based on a neural network that was developed for a link in Chicago to forecast the impact of weather, using weather data gathered each hour and

traffic speed data gathered every five minutes. Both traffic variables and weather information are used as the inputs of their prediction model. Samoili & Dumont (2012) also directly added weather information to their prediction model as an explanatory variable to the traffic forecasting framework. These traffic prediction models will not work, however, when data about the cause of traffic abnormality is not available.

Increasingly, studies are focusing on traffic-related factors that may cause abnormal traffic conditions in traffic prediction models. For example, Tao *et al.* (2005) used three different topology types of neural network based models, namely multilayer perceptron, a modular neural network and a principal component analysis network to predict travel times on a highway corridor in Northern Virginia when an incident happened during the testing day. Castro-Neto *et al.* (2009) used an Online-Support Vector Regression (OL-SVR) model for traffic flow prediction on normal days, holidays and days with traffic incidents on a freeway in the United States. Although the above studies developed models to predict traffic variables during abnormal traffic conditions, it should be noted that the implementation context of most of these attempts is freeways and motorways.

As a part of the current PhD research, Guo *et al.* (2010) used the kNN based algorithm to predict traffic flow recorded by ILDs in central London during abnormal traffic conditions. The authors compared the prediction results of the kNN models against Recurrent Neural Network (RNN) (Mitchell, 1997) and Time Delay Neural Network (TDNN) (Saad *et al.*, 1998), with three different input structures (Krishnan & Polak, 2008). All the models were tested during both normal and traffic incident conditions. The results showed that the kNN method outperforms the recurrent neural network and time delay neural network based methods under traffic incident conditions.

Because travellers do not like this non-recurrent traffic congestion and delay due to its unexpectedness, it is necessary for network managers to know the future traffic situation when an atypical event occurs so that they can take appropriate actions to mitigate traffic congestion and delay. Traffic prediction under abnormal conditions, whether caused by planned events or unexpected incidents, is scientifically challenging, especially since. The available research literature on traffic prediction under abnormal traffic conditions on urban arterial roads is limited. One objective of this research, therefore, is to develop prediction models under abnormal traffic conditions in urban areas.

2.3.7 Summary

An overview of key aspects in the literature on short-term traffic prediction models was presented in this section. Based on this literature, the main objective of this PhD thesis is to develop a short-term traffic prediction model under both normal and abnormal traffic conditions on urban arterial roads. A detailed introduction of the development of a short-term traffic prediction model to achieve this main objective will be presented in the following chapter.

2.4 Short-term data prediction in other domains

Short-term data prediction has been used in other domains including financial data prediction, rainfall forecasting and energy utility load prediction. Because there are similarities and differences between data prediction in transport and these other domains, this section provides a brief review of data prediction in finance, hydrology and energy.

2.4.1 Short-term data prediction in finance

Although financial data is complex and difficult to understand and predict, short-term financial data prediction is a key element in financial and managerial decision making. The main objective of financial data prediction is to reduce the risk in decision making, this being of critical importance for financial organisations, firms and private investors. Financial variables that require short-term prediction include stock market prices (e.g., Lawrence, 1997; Saad *et al.*, 1998; Wang, 2002; Hassan *et al.*, 2007) , foreign exchange rates (Tenti, 1996; Majhi *et al.*, 2009) and interest rates (Ju *et al.*, 1997).

Financial data is difficult to predict because of the characteristics of non-stationarity and high noise (Abu-Mostafa & Atiya, 1996). The non-stationary characteristics implies that financial data series might change over time (Cao & Tay, 2003). Therefore, it is difficult to understand the short-term trends in this noisy data (Cao & Tay, 2003). Figure 2.1 is an example of one day intraday order arrival rates in a foreign exchange market (Bollerslev & Domowitz, 1993).

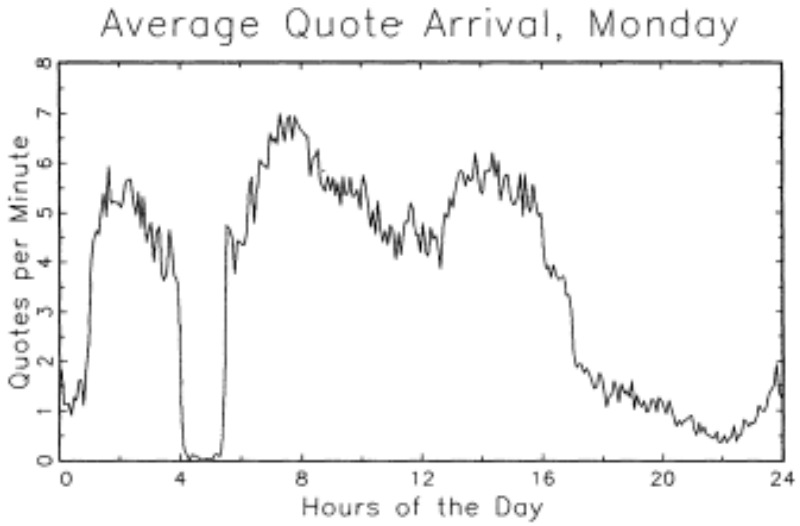


Figure 2.1: Example of intraday order arrival rates in the foreign exchange market (Source: Bollerslev & Domowitz (1993))

Moreover, in non-stationary financial data series the recent data points provide more important information than the distant historical data points; thus, information provided by the recently observed points is given more weight than that provided by the distant data points. Financial data such as stock prices can sometimes be influenced by factors such as macro-economic and political events (Kuo *et al.*, 2001) and, therefore, short-term financial data prediction models also take these characteristics into account in the model development.

Existing literature provides a wide range of prediction models for financial data from statistical methods such as the Auto-Regressive Integrated Moving Average (ARIMA) model (Pai & Lin, 2005) to machine learning based methods such as support vector regression (SVR) (Cao & Tay, 2003; Kim, 2003; Lu *et al.*, 2009), grey system model (GM) (Wang, 2002), k-nearest-neighbour (kNN) (Sharma & Sharma, 2012) and neural networks (NN) (Saad *et al.*, 1998; Majhi *et al.*, 2009).

2.4.2 Short-term data prediction in hydrology

In hydrological research, accurate and reliable short-term rainfall forecasting is a key component in flood warning systems so as to provide proactive information with sufficient lead time to the public. Compared to the financial domain, measurement and measurement errors are important in this domain. Accurately forecasting rainfall is a challenging issue due to the physical characteristics of urban catchments. The hydraulic outputs are sensitive to both rainfall volumes and spatial and temporal information (Simoes, 2012). Moreover, rainfall data is non-linear and noisy because the data collection devices are not error free. The solid line without marker in Figure 2.2 is an example of a rainfall series. Hence, prediction models should consider rainfall data characteristics.

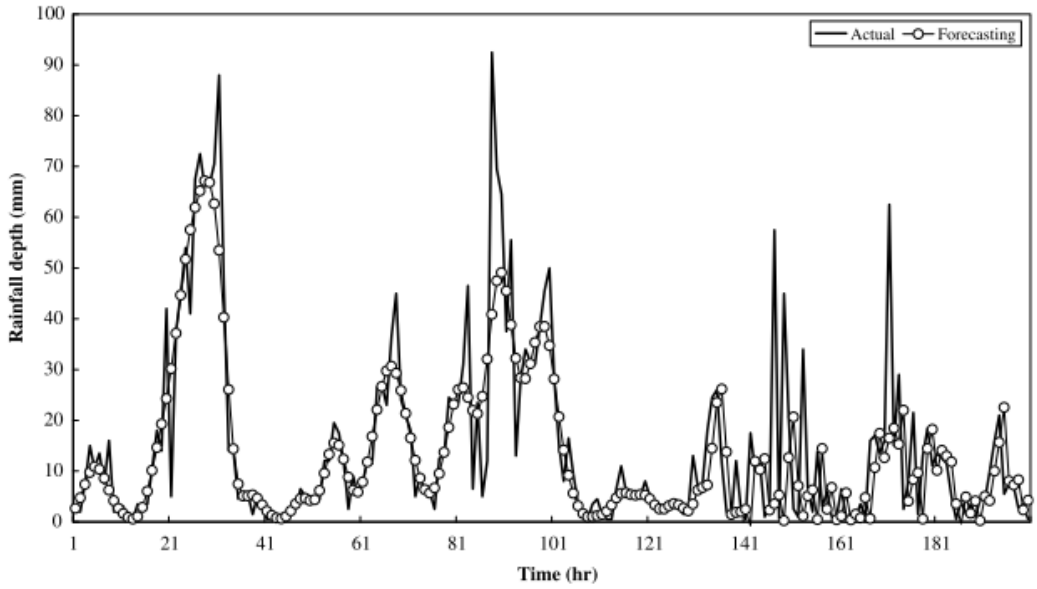


Figure 2.2: Example of rainfall series (solid line without marker) (Source: Hong (2008))

Traditional rainfall prediction models in hydrology are designed using physical mechanisms in hydrologic processes based on the characteristics and knowledge of a specific catchment (Hong, 2008). It is not feasible, however, to predict rainfall series

using this physical approach both because it requires additional calibration data that is not easily collected (Yapo *et al.*, 1996), and because the calculation of the volumes of rainfall requires sophisticated mathematical process (Duan *et al.*, 1994). Statistical models such as ARIMA are also used to formulate the relationships between inputs and outputs without consideration of the physical structure. A statistical time series prediction model has a limitation, however, in its ability to predict sudden changes in the pattern of rainfall related variables, because parameters are estimated offline. For the past two decades, therefore, machine learning methods have been used by researchers for the prediction of rainfall data. For example, French *et al.* (1992) studied the use of neural networks for predicting one-hour ahead accurate rainfall information. Trafalis *et al.* (2003) proposed an SVR approach to forecast rainfall series using radar data. Simoes (2012) developed a stochastic SVR method with a data smoothing technique using singular spectrum analysis to improve the prediction accuracy. Hong (2008) combined machine learning approaches including neural network and SVR to predict rainfall series.

2.4.3 Short-term data prediction in energy

Short-term utility load prediction is used in power systems to estimate future load requirements. The prediction horizon range is from less than one hour to one week. It is necessary to predict hourly loads as well as daily peak loads from both the power generation side and an economic perspective. Accurate load prediction can help electrical engineers in the power industry to optimise the operational state of the power systems and to set up contingency strategies for various time intervals. Power supply at any time must be sufficient to meet the demand from consumers and grid losses.

One of the most significant aspects of utility load data is its daily and weekly recurrent pattern. An example of a daily pattern plot in utility load data is shown in Figure 2.3. Many utility load prediction models make use of seasonal recurrent pattern characteristics to improve prediction accuracy. The utility load data used in prediction models is not as noisy as the one in the financial domain. However, the utility load pattern is heavily dependent on weather fluctuations. Hence, another feature of utility load prediction model is the multivariate inputs. The predicted demand is dependent upon the required loads and weather factors such as temperature, humidity in winter and wind speed in the summer. Because of the complex relationship in inputs, most energy forecasting models use machine learning methods that can efficiently learn the complex relationship among multivariable inputs to achieve sufficiently accurate predictive results.

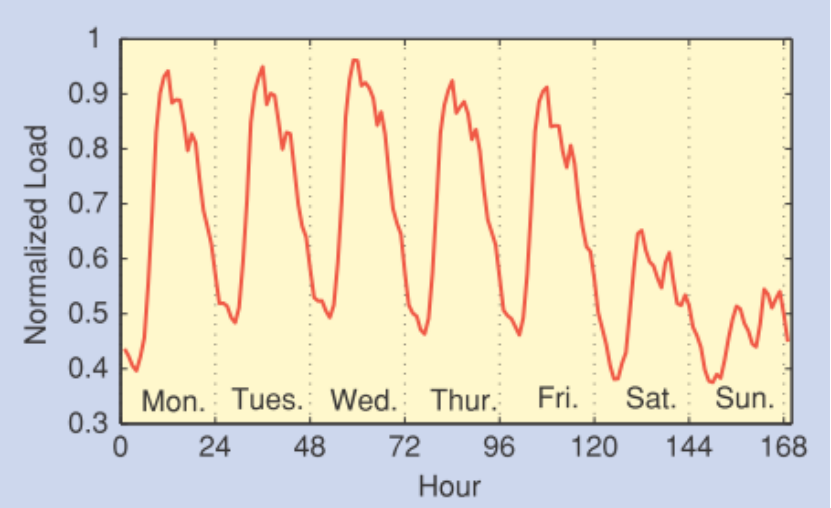


Figure 2.3: Example of daily load data pattern within a week (Source: Espinoza *et al.* (2007))

For example, Park *et al.* (1991) applied an artificial neural network model using historical load data and weather information to electricity load forecasting. In another instance, Charytoniuk *et al.* (1998) forecasted the energy load using a neural network

model where the input data was composed of residential, industrial and commercial customers. More recently, Espinoza *et al.* (2007) proposed a support vector machine approach to predict 1-hour ahead and 24-hour ahead electricity loads. A k-nearest neighbour model is also used in energy data prediction by Kusiak *et al.* (2009) over horizons ranging from 10 minutes to 4 hours into the future.

2.4.4 Summary

This section reviewed short-term data prediction in the financial, hydrological and energy domains. The similarity between data prediction in transport and these above-mentioned engineering domains lies in the modelling of complex relationships between input data and target data to predict future values. Two of the techniques used in financial forecasting and rainfall prediction models, the grey system model and singular spectrum analysis, are less commonly used in the context of traffic prediction. The predictor of the grey system model (GM) is computationally efficient without the processes of method training and parameter optimisation. Moreover, the GM predictor can dynamically update parameters with inputs so as to reduce the dependency on historical data patterns in a way that can be used to detect the change of traffic pattern during abnormal traffic conditions. Singular spectrum analysis is a data smoothing technique used with machine learning methods to improve prediction performance when input data is noisy. Traffic data is indeed very noisy, especially in urban areas, because of sampling and non-sampling errors (Robinson, 2005). Therefore, these two techniques will be investigated further in subsections 2.4.3 and 3.2.2 for their potential use in short-term traffic prediction.

It is noted that one of the key elements in short-term data prediction models in the domains of transport, finance, hydrology and energy is the selection of an appropriate

prediction method. The commonly used prediction methods, including statistical and machine learning methods, are presented and discussed in the following section.

2.5 Review of statistical and machine learning methods in traffic prediction

There is not a well-accepted definition of what a machine learning method is. An early definition of machine learning method stated by Samuel (1959) is:

'a field of study that gives computers the ability to learn without being explicitly programmed'.

A more recent definition by Mitchell (1997) is that:

'*a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*'.

The term *machine learning* is used in this thesis for any algorithms that can learn the relationship between two related data series by fast computers.

Machine learning methods are quite popular in traffic prediction because of their structural flexibility, accuracy and reliability (Vlahogianni, 2009). They have proved to provide better predictive performance than traffic process based models in complex congestion conditions (Qiao *et al.*, 2001; Smith *et al.*, 2002). Moreover, such data-driven machine learning tools do not require extensive expertise on physical rules in traffic flow modelling (Van Lint, 2004).

Machine learning based methods used in traffic prediction range from parametric models such as Kalman filter to nonparametric methods that can capture the patterns or relationships between data series without modelling the physical traffic process, such as neural networks (NN) (Mitchell, 1997) and the k-nearest neighbour (kNN) (Krishnan & Polak, 2008) method. A large number of short-term traffic prediction models based on such statistical/machine learning based methods can be found in the literature. An overview of statistical/machine learning based methods is presented in this section.

2.5.1 Historical average

The simplest traffic prediction methods use the historical average of the variables for each time interval at each site as the predictor. Jeffery *et al.* (1987) used this method in the demonstration project of AUTOGUIDE ATIS in London. This approach is computationally efficient but simplistic. It may provide reasonably accurate results when daily traffic patterns do not dynamically change. However, this approach does not use real-time data and is inaccurate during abnormal conditions.

2.5.2 Statistical methods

The Auto-Regressive Integrated Moving Average (ARIMA) time-series model, which is also known as the Box-Jenkins approach (Box & Jenkins, 1970), is one of the most commonly used statistical models in time series analysis. It uses a statistical approach to identify recurring patterns of different periodicity from historical observations of a time series. Then it uses this information in combination with current observation as a basis for prediction.

Given a time series $\{X_t\}$, white noise series $\{e_t\}$ and the backshift operator B (that is, $B^k X_t = X_{t-k}$), the ARIMA(p, d, q) structure is defined as

$$\phi_p(B)(1-B)^d X_t = \theta_q(B)e_t \tag{2.1}$$

where d is the order of differencing; ϕ_p, $\theta_q(B)$ are polynomials of order p and q respectively, such that

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 -, \dots, \phi_p B^p \tag{2.2}$$

and

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 -, \dots, \theta_q B^q \tag{2.3}$$

The first explicit use of the ARIMA model in the transport prediction literature was by Ahmed & Cook (1979) in which an ARIMA*(0,1,3)* model was proposed to predict freeway traffic flow and occupancy series. All the data sets were collected from three different monitoring systems in Los Angeles, Minneapolis and Detroit at intervals of 20, 30 and 40 seconds respectively. The results showed that the proposed ARIMA model was more accurate in predicting in traffic flow in the freeway system compared to other simple smoothing approaches. Levin & Tsao (1980) compared the traffic volume prediction accuracy of ARIMA*(0,1,1)* and ARIMA*(0,1,0)* using data collected from a Chicago expressway. Later, Hamed *et al.* (1995) used an ARIMA*(0, 1, 1)* model to predict 1-minute ahead traffic flow in urban arterials.

A variation of the basic ARIMA model can accommodate seasonal series. When seasonal terms are included, a seasonal ARIMA model SARIMA*(p, d, q)(P, D, Q)$_S$* is defined as

$$\phi_p(B)\Phi_P(B^S)(1-B)^d X_t = \theta_q(B)\Theta_Q(B^S)e_t \tag{2.4}$$

where Φ, Θ, P, D and Q are the seasonal counterparts of ϕ, θ, p, d and q respectively; S denotes the seasonality.

Williams and Hoel (2003) exploited the recurrence of traffic data using a SARIMA model to improve the accuracy of traffic flow prediction. The authors used a SARIMA$(1, 0, 1)(0, 1, 1)_{672}$ model to predict 15-minute future traffic flows based on data sets collected from two highway locations, one in the United States and the other in the United Kingdom. In this model, the weekly periodicity is considered as the main seasonal factor in traffic prediction.

A Bayesian SARIMA$(1, 0, 0)(0, 1, 1)_{96}$ model was also used by Ghosh *et al.*(2007) in short-term traffic flow prediction. Traffic flow data with a 15-minute interval was collected from a four-legged junction in Dublin. The seasonality of this study was 96, which is one full day. In their study, a total of 1,920 observations, excluding weekends, were used. To demonstrate the quality of the proposed model, however, only time-series prediction models were compared.

The advantage of the above ARIMA models is that they have a well-estimated theoretical background and their implementation is computationally efficient when the input data is stationary. A drawback of ARIMA, however, is the difficulty of determining the optimal model structure. Additionally, all the parameters of ARIMA based models are estimated offline and fixed during prediction.

In the application of short-term traffic prediction, ARIMA models linearly depend on previous traffic observations and capture historical traffic patterns. They therefore use currently identified traffic patterns to predict future values. Under free-flow traffic conditions, or recurrent congestion circumstances, because traffic patterns are not significantly changed, ARIMA models can provide acceptable prediction results by

using their structure to capture the latest patterns based on previous data. Given that the ARIMA model is generally used for stable traffic prediction, it finds prediction more difficult when a traffic incident or accident occurs. Hence, the accuracy of ARIMA based prediction models can deteriorate when long term patterns in data are disrupted during abnormal traffic conditions.

As a part of this PhD research, Guo *et al.* (2012a) tested a SARIMA based model to predict one-step ahead traffic flow using data from central London during both normal and abnormal traffic conditions. The results demonstrated that the pre-calibrated SARIMA was not very accurate in traffic flow prediction during abnormal traffic conditions. Consequently, ARIMA based methods are not further used in the development of short-term traffic prediction models in this PhD research.

2.5.3 Grey System Model (GM)

In the process of time series prediction, a pre-defined mathematical model is sometimes used to provide accurate prediction results. Statistical models such as ARIMA models have been reviewed above. Grey system theory, which is a time series prediction method used in financial, industrial and medicinal areas, was first proposed by Deng (1982). Later research by the same author (Deng, 1989) stated that only a limited amount of data was required to identify the behaviour of unknown systems.

The GM based method predicts the future values of a time series based only on a set of the recently observed data, where the observation process depends on the window size of the predictor (Kayacan *et al.*, 2010). It is assumed that all data values to be used in grey models are positive, and the sampling frequency of the time series

is fixed. From the simplest point of view, grey models which will be formulated below, can be viewed as curve fitting approaches. Kayacan *et al.* (2010) summarised the theory and applications of GM based models and applied them to the short-term prediction of the foreign currency exchange rates. Chang and Tsai (2008) used a grey system model trained by the Support Vector Regression (SVR) method to forecast an equity volume index.

In a generic GM(*n,m*) model, *n* is the order of the differential equation and *m* is the number of variables. In the literature, the most widely used grey prediction model, because of its computational efficiency, is GM(*1, 1*). The GM(*1,1*) model can only be used in non-negative data sequences (Deng, 1989). $\{x^{(0)}(k), k = 1,2, \dots, n \; and \; n \geq 4\}$ is the original positive sampling data. In order to reduce the randomness and improve the regularity, positive data sequences are first transferred to monotonically increasing sequences using an Accumulating Generation Operator (AGO) (Deng, 1989). This method is described as follows:

$$x^{(1)}(1) = x^{(0)}(1) \tag{2.5}$$

$$x^{(1)}(2) = x^{(0)}(1) + x^{(0)}(2) \tag{2.6}$$

$$\vdots$$

$$x^{(1)}(T) = x^{(0)}(1) + x^{(0)}(2) + \cdots + x^{(0)}(t) \tag{2.7}$$

or, the above equations can be summarised as:

$$x^{(1)}(k) = \sum_{i=0}^{k} x^{(0)}(i), k = 1,2, \dots, t \tag{2.8}$$

where the superscript (0) of $x^{(0)}$ represents that this is an original element and the superscript (1) of $x^{(1)}$ means this element is newly formed using AGO.

It is clear that the new sequence $X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(t), t \geq 4\}$ is monotonically increasing, that is:

$$x^{(1)}(1) \leq x^{(1)}(2) \leq \cdots \leq x^{(1)}(t) \tag{2.9}$$

GM(*1,1*) is easily defined as follows (Deng, 1989):

$$x^{(0)}(k) + az^{(1)}(k) = u, k = 2,3, \dots, n \tag{2.10}$$

where:

$$z^{(1)}(k) = c * x^{(1)}(k) + (1 - c) * x^{(1)}(k - 1), 0 \leq c \leq 1 \tag{2.11}$$

c is a coefficient usually set as 0.5 ; *Z* is the mean value of adjacent data (Deng, 1989).

$$\frac{dx^{(1)}(t)}{dt} + ax^{(1)}(t) = u \tag{2.12}$$

$[a, u]^T$ is a sequence of parameters that can be found as follows:

$$[a, u]^T = (B^T B)^{-1} B^T Y \tag{2.13}$$

where $Y = \left[x^{(0)}(2), x^{(0)}(3), x^{(0)}(4), \dots, x^{(0)}(n)\right]^T$

and $B = \left[\left(-z^{(0)}(2), 1\right), \left(-z^{(0)}(3), 1\right), \left(-z^{(0)}(4), 1\right), \dots, \left(-z^{(0)}(n), 1\right)\right]^T$.

The solution of $x^{(1)}$ at time k in the above differential equation is:

$$\hat{x}^{(1)}(k + 1) = \left[x^{(0)}(1) - \frac{u}{a}\right] e^{-ak} + \frac{u}{a}. \tag{2.14}$$

Initially the AGO method is used to generate an increasing sequence. Then the Inverse Accumulating Generation Operator (IAGO) method is applied to find the predicted value of the original data (Deng, 1989).

$$\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k)$$

$$= \left[x^{(0)}(1) - \frac{u}{a}\right] e^{-ak} - \left[x^{(0)}(1) - \frac{u}{a}\right] e^{-a(k-1)} \qquad (2.15)$$

$$= \left[x^{(0)}(1) - \frac{u}{a}\right] e^{-ak}(1 - e^{a}).$$

In summary, grey models, as formulated above, can be simplistically viewed as a curve fitting approach. Kayacan *et al.* (2010) summarised the theory and applications of GM based models and applied them to the short-term prediction of foreign currency exchange rates. Guo *et al.* (2012a) compared GM(*1,1*) and SARIMA*(1,0,1)(0,1,1)*$_{96}$ models in short-term traffic prediction under normal and incident traffic conditions on urban roads as a part of this PhD research. The results show that the GM based method (MAPE of 10.00%) has slightly better prediction accuracy than SARIMA (MAPE of 10.62%) under normal conditions. Under abnormal traffic conditions, the GM based method (MAPE of 22.97%) produced more accurate traffic prediction results than the SARIMA based method (MAPE of 37.47%), because GM has better ability to detect and respond to the sudden change of traffic patterns which are caused by an unplanned event such as a traffic incident. In contrast, the SARIMA based method is less accurate in predicting traffic during incidents.

Since the GM predictor can model non-linear relationships of input traffic data and can dynamically update parameters with inputs to reduce the dependency on historical data patterns, it should predict acceptable traffic variables under abnormal traffic conditions.

2.5.4 Kalman filter (KF)

The Kalman filter algorithm, which is widely used by statisticians in time series modelling, was first introduced by Kalman (1960). Maybeck (1979) stated that '*a Kalman filter is simply an optimal recursive data processing algorithm*'. The Kalman filter is a parametric method which is able continuously to update the prediction of selected variables based on explicit models of measurement and the physical processes of a system. This method successively updates its parameters at different time periods and uses two equations, namely the state transition equation and the observation equation (also known as the measurement equation), to estimate the state of a process. A KF model assumes that '*the state of a system at a time* t *evolved from the prior state at time* $t-1$' using the state equation (Faragher, 2012). The output of the system is calculated using the observation equation.

The definitions of the state and observation equations are as follows:

- *state equation:*

$$x(t) = Ax(t-1) + Bu(t-1) + w(t-1) \tag{2.16}$$

- *observation equation:*

$$y(t) = Hx(t) + n(t) \tag{2.17}$$

where $x(t)$ is the state variable, A is the state transition matrix of the process from the state at $t-1$ to the state at t; B is the control-input matrix which is applied to the control vector $u(t)$; $y(t)$ is observations from sensor sources; H is the observation model which maps the true state space into the observed space, and $w(t)$ and $n(t)$ are the process and measurement noise respectively, which are assumed to be independent of each other, white and with a normal probability distribution with

covariance Q and R. The state transition equation is denoted as a first-order Markov process of the state vector (Han, 2012); the observation equation estimates the unknown state with the observable measurements. The algorithmic loop of the KF is shown in Figure 2.4. KF recursive algorithm is summarised in Table 2.1.

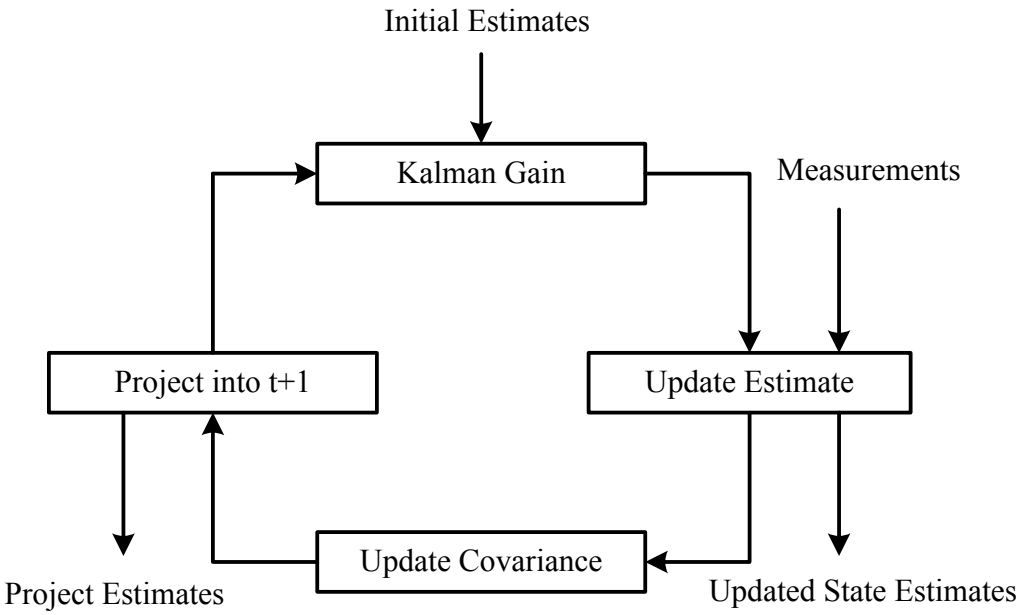


Figure 2.4: Algorithmic loop of the KF (Source: Thacker & Lacey (1996))

Table 2.1: Summary of the KF recursive algorithm (Source: Thacker & Lacey (1996))

Description	Equation
Kalman Gain	$K_t = P_t' H^T (H P_t' H^T + R)^{-1}$
Update Estimate	$\hat{x}_t = \hat{x}_t' + K_t(y_t - H\hat{x}_t')$
Update Covariance	$P_t = (I - K_t H P_t')$
Project into $t+1$	$\hat{x}_{t+1} = A\hat{x}_t$ $P_{t+1} = AP_tA^T + Q$

Okutani & Stephanedes (1984) showed that a Kalman filter was an efficient method to predict traffic flow on urban roads. Kalman filter based models were used on weekly and daily differenced traffic flow data. In their study, the results showed that both weekly and daily models performed substantially better than the prediction model of the Urban Traffic Control System (UTCS) (FHWA, 1973; Stephanedes *et al.*, 1981). Similarity, Stathopoulos & Karlaftis (2003) used 3-minute interval traffic flow data collected from upstream urban arterial streets in Athens to predict traffic flow at the downstream locations. The results showed that a Kalman filter based model yielded a MAPE of 12% compared to a 20% MAPE value from a simple ARIMA prediction model.

The Kalman filter is a state-space model with multivariate inputs. The multivariate nature of the KF allows traffic data from multiple sensors with known physical relationships so as to increase the prediction accuracy. Physical relationships are difficult to determine in some cases, however. Another drawback of the Kalman filter method is its reliance on some fundamental assumptions such as that the system and measurement noises are white and Gaussian distributed. These assumptions limit the real applications of a Kalman filter based system (Maybeck, 1979).

2.5.5 Neural Network (NN)

Non-parametric methods such as Neural Network (NN), which has its basis in Artificial Intelligence, have also been used for short-term traffic prediction. Neural network methods were originally motivated by the goal of having machines that can mimic the brain. As stated by Mitchell (1997),

'*neural network learning methods provide a robust approach to approximating real-valued, discrete-valued and vector-valued target functions*'.

Complex non-linear relationships between multiple inputs and outputs can be modelled by neural networks in order to capture or learn patterns within data.

A basic framework of a neural network model includes four main elements, namely nodes, connection, layers and transfer function. Nodes, also known as neurons, are simple processing units. Two interconnected nodes are connected by weighted connections which represent the nature of their interaction. Optimal weights for each connection can be calculated during the training process which is used to calibrate the model using patterns in data. Layers are the topology of a neural network, where nodes[1] and connections are assigned. Transfer function determines the state of each neuron. The mathematical process at a single neuron is shown in Figure 2.5. A single neuron includes a set of *synapses* connected to the inputs. Each of them is characterised by a weight. This process includes two steps:

- Calculate a linear combination of inputs, and
- Transfer the weighted sum into output using an activation function.

[1] The terms *neuron* and *node* are interchangeably used throughout this chapter.

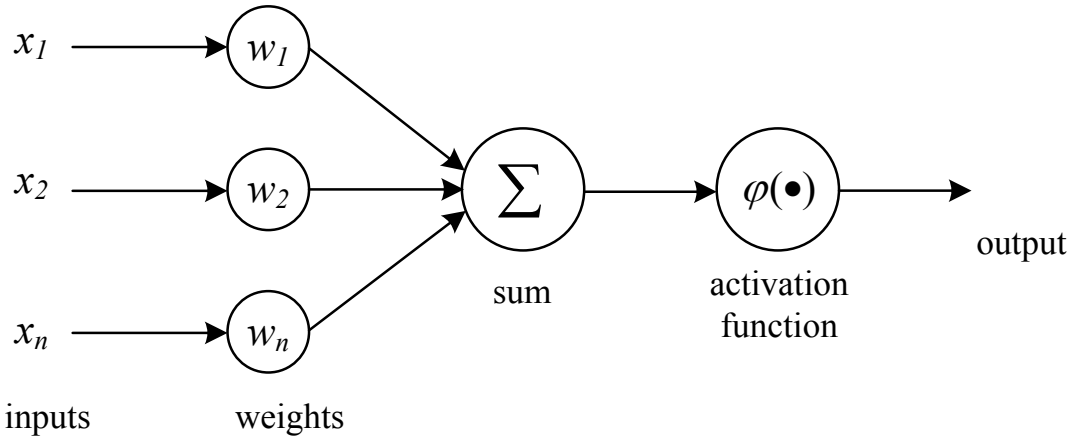


Figure 2.5: Process of a single neuron (Source: Rokach (2010))

Let x_i be the i^{th} input and w_i be the corresponding weight. The sum of a linear combination of inputs is given by $Z = \sum w_i x_i$. Then, a nonlinear activation function $\varphi(\cdot)$ is applied to the weighted sum. The output of this neuron is $y = \varphi(Z)$. Generally, the most commonly used activation functions include sign functions, piecewise-linear functions and sigmoid functions.

A variety of different neural network models have been applied to traffic engineering applications (Dougherty, 1995). In short-term traffic prediction, *feed-forward neural network* (FFNN) (Kriesel, 2007) is a simple and widely used network, the structure of which is shown in Figure 2.6. There are three layers in this model structure, which does not have any cycles and loops, namely an input layer, some hidden layers and an output layer. In FFNN, each neuron in one layer strictly feeds forward to the output units of the next layer.

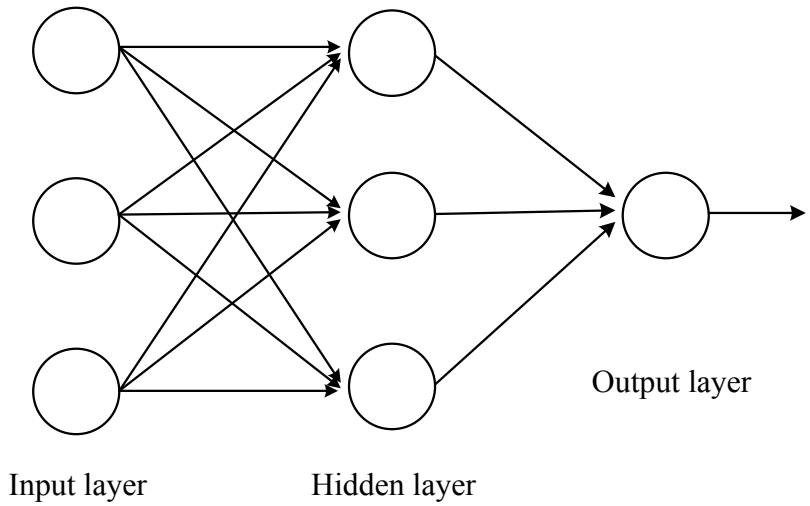


Figure 2.6: General architectures of feed-forward networks (Source: Mitchell (1997))

Many studies have investigated the applications of neural networks in short-term traffic prediction. For instance, Park & Rilett (1999) used a FFNN model to predict 5-minute future traffic time on a freeway in the United States. In their model, data from 79 days was used for training and 1 day for testing. The results showed that the prediction of the FFNN based model was less accurate under traffic congestion conditions than under free-flow conditions. Huang and Ran (2002) used a FFNN model to forecast future traffic speed using both weather data gathered each hour and traffic speed data gathered every five minutes. In this study, weather, which caused the abnormality in traffic, was directly included in the model as an explanatory variable. Only weather and normal traffic flows were considered in the experiments, however, with no attempt being made to run the model when traffic conditions were abnormal due to traffic incidents or accidents.

A growing number of developments to the basic neural networks have been proposed for traffic prediction. For example, Park *et al.* (1998) used radial basis function neural networks (Park & Sandberg, 1991), where the transfer function is a Gaussian function rather than a sigmoid function, for freeway traffic volume

forecasting. Abdulhai *et al.* (1999) predicted traffic flow on freeways using a time delay neural network (Saad *et al.*, 1998) that can learn seasonal patterns. Van Lint (2004) presented a model used for travel time prediction on freeways with state-space neural networks that can learn both spatial and temporal patterns from traffic data. The prediction results showed that the prediction accuracy was acceptable using travel time data from both the simulation and the real world. Ishak & Alecsandru (2004) used four different architecture of neural networks to optimise the performance of short-term traffic speed prediction. Zheng *et al.* (2006) combined a Bayesian method with a neural network model in 15-minute ahead freeway traffic flow prediction. Innamaa (2000) used a multi-layer perceptron neural network model (Kriesel, 2007) with more than one layer of trainable weighted connections to predict both traffic flow and speed. However, the accuracy difference between the proposed model and other machine learning methods were not compared in their test.

The advantage of neural network based methods is their learning ability in capturing the traffic patterns from large historical traffic datasets. They can make full use of historical traffic datasets to address complex problems where the relationships between data series are not clear and well-defined. However, a lack of robustness is a main drawback of neural network based methods, since they require a large amount of accurate historical traffic data for their training and learning processes. Neural network based methods also suffer from difficulty in selecting a large number of controlling parameters in their implementation, which makes them less practical in ITS applications. Moreover, when data is highly noisy and dimensional, neural network methods often exhibit inconsistent and unpredictable performance (Kim, 2003).

2.5.6 K-Nearest Neighbour method (kNN)

The kNN method is a typical lazy learning method that does not involve any model construction before it is required for testing (Hastie *et al.*, 2001). This model-free method is highly unstructured and does not require an understanding of the nature of the relationship between the features and the outcomes. The kNN method, therefore, is more robust than the parametric time series models.

The basic assumption of the kNN method is '*that observations which are close together in feature-space are likely to belong to the same class or to have the same a posterior distributions of their respective classes*' (Devijver & Kittler, 1982). In its application of data prediction, the kNN method can locate a number k of observations (also termed as nearest neighbours) from a historical dataset and then predict future variables based on the nearest neighbour set. The nearest neighbour set can reflect the historical traffic data that are similar to the current traffic state during congestion.

The kNN based prediction method can be deconstructed into three fundamental components: a database of observations, a neighbourhood search procedure and a prediction process. Figure 2.7 depicts the general flow of data through the kNN based prediction method.

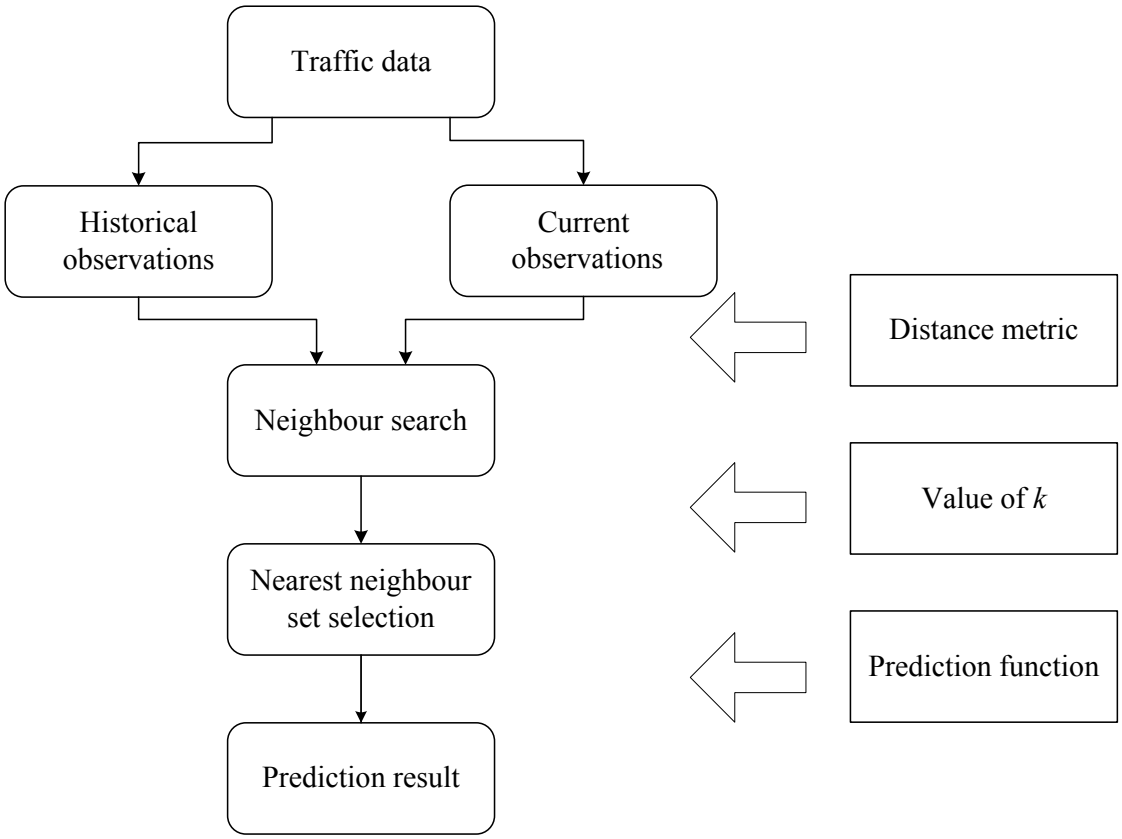


Figure 2.7: General structure of kNN based prediction method

The search procedure finds the nearest neighbours, which are the historical observations that are most similar to the current condition. The nearest neighbours then become the inputs to the prediction step so that it may calculate a predictive value. During these three procedures, three key design parameters are the definition of a distance metric to determine the nearness of historical data to the current conditions, the choice of k and the selection of a prediction function given a collection of nearest neighbours.

- *Distance metric:* this is used to determine the distance between the current input feature vector and historical observations. The most commonly used metrics include Euclidean distance, weighted Euclidian distance, the Mahalanobis distance metric and the Minkowski distance metric (Kruskal, 1964).

Let $dist_{p,q}$ be the distance between two feature vectors x_p and x_q with dimension n. The equations of the three above mentioned distance metrics are shown in Table 2.2.

Table 2.2: Equations of distance metrics (Source: Robinson (2005))

Distance Metric	Equation
Euclidian distance	$dist_{p,q}{}^2 = (x_p - x_q) \cdot (x_p - x_q)^T$
Mahalanobis distance	$dist_{p,q}{}^2 = (x_p - x_q) \cdot S^{-1} \cdot (x_p - x_q)^T$ S: the variance covariance matrix
Minkowski distance	$dist_{p,q}{}^2 = [\sum_{i=1}^{n} \lvert(x_{pi} - x_{qi})\rvert^c]^{\frac{1}{c}}$ c: norms of distance metric

- *Choice of k:* determines how many nearest neighbours are chosen from the historical dataset. For example, if k is chosen to be 10, then the 10 historical observations that have the nearest distances to the input feature vector will be used in the prediction process (Robinson, 2005).
- *Prediction function:* is the core of the kNN based method. Let J_y represent the set of k nearest neighbours corresponding to the current input x_c. The predictive output y_c is given by:

$$y_c = f(J_y). \tag{2.18}$$

The function $f()$ is called *the prediction function* (Smith *et al.*, 2002) or *local estimation method* (Robinson, 2005).

In the early 1990s, a kNN based method, which was first explicitly used in traffic prediction literature by Davis & Nihan (1991), provided an alternative method for short-term traffic prediction. This study used the real-time data of traffic flow and occupancy at the time interval $t-1$ to predict flow and occupancy at time interval t for a freeway network. In addition, three different settings for the value of k with three different distance metrics were used and compared with a statistical ARIMA approach. The results showed that the kNN method did not conclusively outperform a time-series model. One obvious reason for this outcome is an insufficient historical dataset for training in their proposed model.

In Smith & Demetsky (1997), the prediction results of a kNN model were compared with those using historical averages, ARIMA model, and neural networks. They used data collected from two sites on a freeway in North Virginia to predict traffic volume at the interval of 15 minutes. Large data sets were collected from two independent sites and the training data from the first site included a few incidents. The results suggested that kNN models achieved the greatest accuracy and robustness. Later, Smith *et al.* (2002) demonstrated that kNN based methods can be used to predict traffic variables on highways under different traffic conditions.

Clark (2003) used a multivariate nonparametric regression model based on the kNN method, in which three types of traffic variables were selected as inputs, namely speed, traffic flow and occupancy. Specifically, all these traffic variables were used in the calculation part of the distance metric to search nearest neighbours. Clark found that the proposed multivariate models (MAPE of 10.52%), were more slightly more accurate than univariate models (MAPE of 10.98%) for traffic flow prediction on motorways. The accuracy of this approach in relation to other prediction methods was not compared in this study, however.

Turochy (2006) developed short-term traffic prediction models that employed the kNN based method with normalcy information for the characterisation of network conditions. Prediction procedures with normal traffic conditions performed more accurately than the one with abnormal traffic condition. In this study, there were a total of 500 observations, collected from five different locations, including of mean speed, traffic flow, and occupancy. Turochy concluded that the use of traffic condition information in kNN models improved the accuracy of traffic prediction systems. However, only normalcy information was considered in this test.

Krishnan & Polak (2008) also proposed the use of the kNN method ($k = 100$) to predict traffic flow. In addition, three types of traffic prediction models were introduced by Krishnan & Polak (2008) based on information used to model the recurrent traffic process. However, only the kNN method was used to compare different prediction model structures. A comparison with other machine learning methods for prediction performance was not carried out.

Tam & Lam (2009) used an improved kNN method to predict the travel times in the next five minute interval during incidents in Hong Kong. The prediction function of their proposed method incorporated temporal variances and co-variances of travel times between time intervals for improving the predicted travel times with the most recent information, which showed that the proposed kNN based method (4.6% MAPE) produced more accurate travel time predictions than a historical average model (14.27% MAPE) and a simpler kNN based method ($k = 2$) (6.82% MAPE). They did not, however, consider the effect of varying the parameter k in the kNN prediction model. Moreover, only one single link was used as explanatory variables in their study, and whether this method can be generally used has not been tested.

Errors in prediction outputs are also a function of variation in the underlying traffic datasets. The same training and testing datasets are required to compare prediction accuracy of different machine learning methods. Guo *et al.* (2010) compared the prediction accuracy of kNN and NN based methods for traffic flow prediction using traffic data from the Russell Square and the Marylebone corridors. More recently, as a part of this PhD research, Guo *et al.* (2012c) used the same traffic datasets to compare kNN with GM and SVR methods for accuracy of short-term traffic prediction under normal and incident conditions on urban roads. The results showed that the kNN based method that can exploit information by choosing past traffic patterns is more accurate than other methods. Given the large variation because of traffic incidents or accident under abnormal traffic conditions, kNN is more effective than pre-determined methods that attempt to develop a single mapping function.

The kNN based method is a well-established non-parametric method that does not involve any model development before requiring prediction. It can find similar input patterns from historical datasets and uses a certain combination of the outputs with input patterns as its final prediction. The literature reviewed above shows that a kNN predictor can be used in traffic prediction problems in different traffic conditions. Hence, the kNN method should be considered as a candidate for traffic prediction during abnormal traffic conditions.

2.5.7 Kernel Smoothing (KS)

Kernel Smoothing (KS), which is similar to the kNN method, uses all the observations in the historical data to identify the next output. Not all the records have the same influence on the output value, however; hence the Kernel Smoothing method

uses a weighted combination of an appropriately determined smoothing parameter for the observations (Wand & Jones, 1995; Hastie *et al.*, 2001). Moreover, the weight of each input observation is related to its Euclidean distance from the current observation.

El Faouzi (1996) predicted the traffic count at a station using a Kernel smoothing approach. Sun *et al.* (2003), meanwhile, proposed a local linear based traffic prediction method based on a traditional KS method to predict 5-minute future traffic speed on a freeway in Houston. A KS based method, however, since it uses all data in the historical database to estimate the future output value, requires large amounts of computational time and memory space. Another disadvantage, discussed by Hjort & Walker (2001), is '*kernel density estimator with optimal bandwidth lies outside any confidence interval, around the empirical distribution function, with probability tending to 1 as the sample size increases*', which means that sometimes observations that are arbitrarily chosen will influence the final accuracy.

2.5.8 Spinning Network (SPN)

University of Vermont (2008) developed a forecasting method, named a Spinning Network (SPN), which has applicability beyond traffic flow prediction and which uses several features of human memory to enhance artificial intelligence. These features are '*the imprecise nature of information received, the association of ideas, and the improvement of information retrieval through an investment of time and effort*' (University of Vermont, 2008). The main idea of SPN is to keep observing continuous data while at the same time organising and storing all the data into its memory. There are two basic concepts in the SPN, namely a data item and a spinning ring. The data item is the element that can be collected, stored and processed. In the SPN, all the data items are stored in spinning rings, as shown is Figure 2.8. All the rings will spin

at a predefined speed. In the process of data input, each ring has a fixed window to receive new items; in the process of data output, another fixed window can send the merged data items to the inner ring of the network.

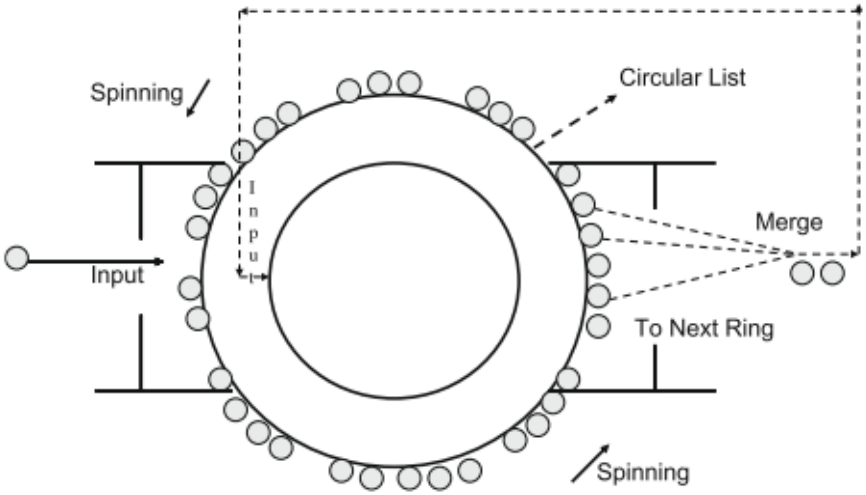


Figure 2.8: Structure of spinning rings (Source: Huang & Sadek (2009))

Huang & Sadek (2009) used a SPN method to predict traffic flow on urban roads in Virginia. In their study, four rings were chosen, spinning at different speeds. The outmost ring had the fastest speed, while the innermost ring spun the slowest. The authors tested their proposed SPN model for the prediction of 30-minute future travel time. They compared the prediction results of the SPN based method with those using the back-propagation neural network and the kNN method. Their experiments confirmed that the proposed SPN gave the smallest error amongst all the methods. The corresponding Absolute Percentage Errors (APE) were 7.57% for SPN, 17.3% for the 3 nearest neighbour algorithm, and 68.57% for the back propagation neural network. The model structure is extremely complex (Samoili & Dumont, 2012), however. Moreover, in the test which compared the SPN method with the kNN method, they only chose a very small value of k $(k = 3)$: the value of k is an important factor that affects the final prediction results.

SPN is a type of memory-based learning methods that selects output data based on the closest training vectors. Similar to a kNN based prediction method, SPN does not require a data training process, but it has a complex structure and requires more parameters to be pre-determined.

2.5.9 Support Vector Regression (SVR)

Support Vector Regression (SVR) was introduced by Vapnik (1995) based on statistical learning theory and implements the structural risk minimisation principle from computational learning theory. The basic idea of SVR is to '*map the data* x *into a high dimensional feature space F via a nonlinear mapping Φ and to do linear regression in this space; thus, linear regression in a high dimensional (feature) space corresponds to non-linear regression in the low dimensional input space*' (Müller *et al.*, 1997). The regression function can be written as:

$$f(x) = \langle w \cdot \Phi(x) \rangle + b \text{ with } \Phi: R^n \rightarrow F, w \in F \qquad (2.19)$$

where w is a vector in the feature space, $\Phi(x)$ is a function which maps the input x to a vector in the feature space and b is a threshold.

The dot product in Equation 2.19 can be replaced using a kernel function. The advantage of using a kernel function is that this can enable the dot product to be calculated in a higher-dimensional feature space without explicitly mapping $\Phi(x)$ into the feature space (Al-Anazi & Gates, 2010). There are several kernel functions, such as a linear function, polynomial function, radial basic function and multi-layer perception functions, which are introduced by Gunn (1998). Among these, the Radial Basic Function (RBF) is the most popular for use in non-linear classification problems (Tay & Cao, 2001; Sapankevych & Sankar, 2009) and is defined as:

$$k(x, y) = exp(-\gamma \|x - y\|^2) \quad (2.20)$$

The parameter γ is the bandwidth of the Gaussian kernel.

The goal is to find an optimal value of b and weights w. When the mapping function $\Phi(x)$ is fixed, two parts should be considered to determine w. One is '*the flatness of the weights* w'; the other is '*the error generated by the estimation process of the value, also known as the empirical risk*' (Sapankevych & Sankar, 2009). The value w should be determined by minimising the sum of empirical risk $R_{emp}(f)$ and a complexity term $\|w\|^2$:

$$R_{reg}(f) = R_{emp}(f) + \frac{\lambda}{2}\|w\|^2 = C\sum_{i=1}^{N} \Gamma\left(f(x_i) - y_i\right) + \lambda\|w\|^2 \quad (2.21)$$

where C is a pre-specified value, N is the sample size, Γ is a cost function (also known as loss function) and the scale factor λ is a regularisation constant. Vapnik's ε-insensitivity loss function is used in non-linear regression (Tay & Cao, 2001). The definition of ε-insensitivity loss function is given by:

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon & if |f(x) - y| \geq \varepsilon \\ 0 & otherwise. \end{cases} \quad (2.22)$$

Equation 2.21 can be minimised as a quadratic programming problem that is defined as:

$$minmise: \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)\left(\alpha_j - \alpha_j^*\right)k\left(x_i, x_j\right) - \sum_{i=1}^{N}\alpha_i^*(y_i - \varepsilon) - \alpha_i(y_i + \varepsilon) \quad (2.23)$$

$$subject\ to: \sum_{i=1}^{N}\alpha_i - \alpha_i^* = 0,\ \alpha_i, \alpha_i^* \in [0, C] \quad (2.24)$$

By solving the Equation 2.24 with the constraint of Equation 2.25, the Lagrange multipliers α_i and α_i^* can be found. The vector w can be written in terms of data combination as:

$$w = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\Phi(x_i). \quad (2.25)$$

Hence, Equation 2.19 can be rewritten as:

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)k(x_i, x). \quad (2.26)$$

The SVR based method is widely used to solve data prediction problems. Sapankevych & Sankar (2009) presented a general survey of time series prediction applications using the SVR method, including financial markets, telecommunications, electrical loading or price prediction, as well as other fields. Wu *et al.* (2004) successfully used SVR to predict 3-minute future traffic time using the traffic data collected from a freeway in Taiwan. They compared the prediction results of the SVR model with those using historical averages and current-time predictors. The results showed that the SVR method had the smallest errors of all the methods and it was also successful when the prediction experiments were transferred to a different site. Wu *et al.* (2004) failed, however, to consider the effect of varying the key design parameters of their SVR model and to compare their model with other machine learning methods. Moreover, during the period in which test data was collected, there were no planned or unplanned traffic incidents, and the data loss rate was under the threshold value. In other words, only normal traffic conditions were dealt within their experiments.

Castro-Neto *et al.* (2009) applied a development of the basic SVR approach, an Online-Support Vector Regression (OL-SVR) model, to traffic flow prediction on a freeway in the United States under both normal and abnormal traffic conditions. They compared the prediction accuracy of their SVR model against Gaussian maximum likelihood (GML) (De Lurgio, 1998), Holt exponential smoothing (HES) (Lin, 2002) and neural networks (NN). A total of 107,520 observations were used in their test and they found that the SVR model had the best overall prediction performance under

abnormal traffic conditions, with an average MAPE of 13.1%, compared to a 40.9% MAPE value for GML, a 14.8% MAPE value for HES and a 14.7% MAPE value for NN. This study also found that using historical information does not improve prediction accuracy during incidents. Once again, however, the study was limited by a failure to consider the effect of varying the key design parameters and function of their SVR model. In addition, they did not compare their proposed OL-SVR method with other widely used non-parametric algorithms such as kNN.

SVR can deal with data prediction problems in non-linear systems using a regression function by fitting a curve to a set of data points. SVR methods have been successfully applied to a number of data prediction applications ranging from financial data (Tay & Cao, 2001) to traffic data. SVR is a non-parametric method and can be applied without any prior knowledge. Hence, this method might be able to predict traffic variables during abnormal traffic conditions when traffic patterns suddenly change.

2.5.10 Random Forests (RF)

The random forests based method was first introduced by Breiman (2001) as a statistical learning method for use with high-dimensional classification and regression problems, where classification is used to model categorical variables and regression is used to predict continuous variables. Random Forests (RF) are tree-based ensemble learning methods using bootstrap samples and randomness in the procedure of tree building (Breiman, 2001). Breiman (2001) defined a random forest as:

'a random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x,\Theta_k), k=1,2,...\}$ *where the* Θ_k *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input* x*.'*

RF grows an ensemble of trees using training data. Let us assume a set of training data $\{(X_n, y_n), n = 1, \dots, N\}$, where $X = (X^1, \dots, X^P), X \in R^P$ contains independent variables and y_n is a predictor output. P is the dimensionality of the independent variables. Unlike standard trees, RF employs randomness when selecting a variable to split each tree and each node. For each tree in a random forest, the training data uses a bagged version (Meinshausen, 2006). Each node is split using the best split-point among a subset of predictors randomly chosen at that node, rather than choosing the best one among all predictors (Liaw & Wiener, 2002). The random parameter vector, called θ, is used to determine the growth of the trees and to calculate the split-points at each node. The corresponding tree is denoted by $T(\theta)$. The output of the ensemble of trees is $\{T_b\}_1^B$, where B is the number of the trees. For regression, the prediction of random forests at a new point x is the average of all corresponding trees $\frac{1}{B}\sum_{b=1}^{B} T_b(x)$. Figure 2.9 shows the general architecture of RF.

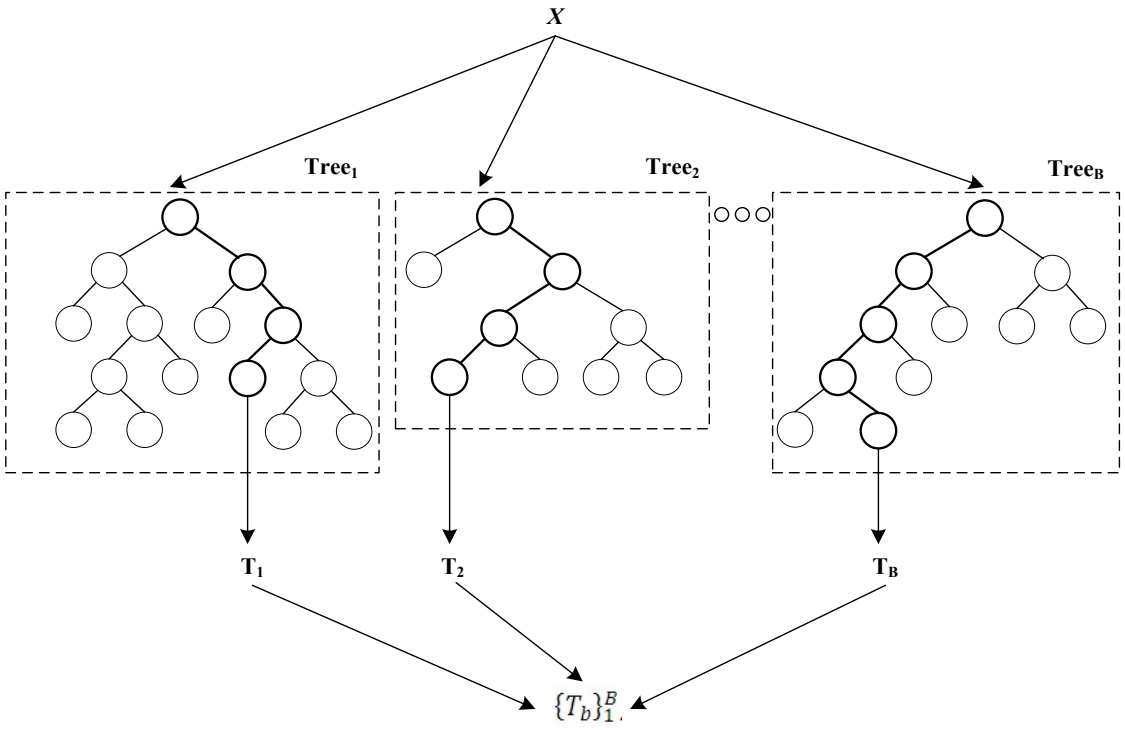


Figure 2.9: A general architecture of RF (Source: Verikas *et al*., (2011))

The RF algorithm discussed above can be summarised into three main steps:

1). Draw bootstrap samples from the training data $\{(X_n, y_n), n = 1, \ldots, N\}$. N is the sample size.

2). For each of the bootstrap samples θ, grow an un-pruned regression tree $T(\theta)$.

3). Predict new data x by aggregating the predictions of the corresponding trees $\{T_b\}_1^B$using an averaging algorithm for regression, that is $\frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Figure 2.10 shows a flow chart of the RF method to illustrate this process.

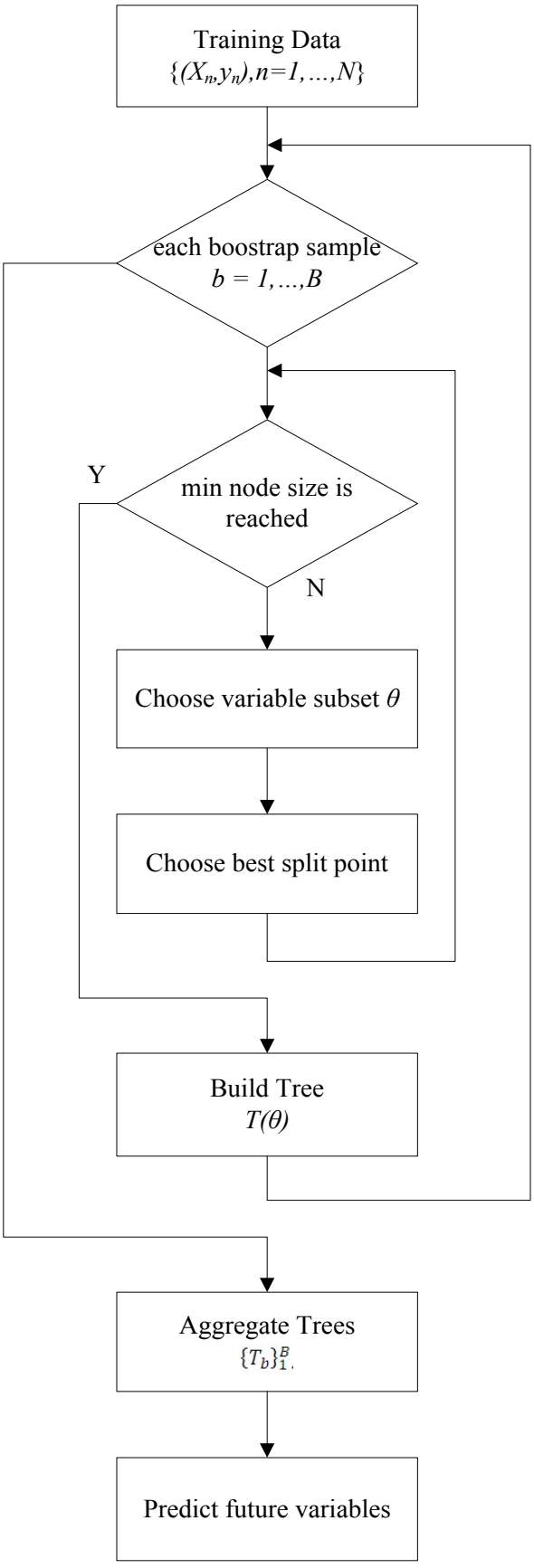


Figure 2.10: Flow-chart of the RF process

RF is becoming a popular technique in a variety of fields for classification, prediction (e.g., Prasad *et al.*, 2006) , variable selection (e.g., Genuer *et al.*, 2010) and outlier detection (e.g., Zhang & Zulkernine, 2006). Verikas *et al.* (2011) summarises the RF method and its applications in the fields of engineering. The RF method has not been widely used in traffic prediction, however, with only Leshem & Ritov (2007) having used RF in Traffic Management and Information Systems to predict traffic flow under normal traffic conditions. The prediction horizon in their study is 30 minutes, but factors such as prediction step, data sampling frequency and traffic conditions are not discussed.

The main advantages of RF, as summarised by Hastie *et al.* (2008) and Saffari *et al.* (2009), include its ability to capture interactions, handle missing data, scale well for a large sample size and to deal robustly with both irrelevant inputs and outliers. Hastie *et al.* (2008) also states that neural networks and support vector machine methods demonstrate a lack of the above characteristics. RF, however, requires an extensive training dataset to build trees. In the context of RF regression, RF can solve traffic variable prediction problems by searching similar patterns from the training dataset.

2.6 Summary of existing traffic prediction methods

Section 2.3 discussed the main factors that may influence the development of prediction models for short-term traffic prediction. Section 2.4 reviewed the basic concepts and algorithms of statistical and machine learning methods in the application of short-term traffic variable prediction. Table 2.3 shows a comparison summarising the key features of literature reviewed in this chapter, under a number of headings covering the characteristics of the prediction context (urban vs freeway, nature and

temporal resolution of input data, prediction horizon), the characteristics of the prediction method used (parametric model based or non-parametric, nature of the training process and data requirements) and the nature of the traffic conditions (normal or abnormal) within which the models were implemented. Table 2.4, meanwhile, summarises the key characteristics, advantages and weaknesses of the existing prediction methods as reviewed in Section 2.4 (Hastie *et al.*, 2001; Vlahogianni *et al.*, 2004; Samoili & Dumont, 2012). In addition, Table 2.5 compares the statistical and machine learning methods reviewed in Section 2.4 in terms of data utilisation, prediction accuracy, model robustness, calibration, ease of implementation and transferability. There are three levels for each characteristic, namely Good, Fair and Poor.

Table 2.3: Categorisation of available literature in existing traffic prediction models

Author	Context	Input data resolution (min)	Prediction step	Input variables	Input data pattern		Training		Traffic Condition	Method	Structure
					Seasonal temporal pattern	Spatial pattern	Process	Dataset			
Ahmed & Cook (1979)	Freeway	0.5	1	Flow and occupancy			Offline (Calibration)	Yes	Normal	ARIMA	Parametric
Levin & Tsao (1980)	Freeway	20	1	Flow and occupancy			Offline (Calibration)	Yes	Normal	ARIMA	Parametric
Hamed *et al.* (1995)	Urban road	1	1	Flow			Offline (Calibration)	Yes	Normal	ARIMA	Parametric
Williams & Hoel (2003)	Freeway	15	1	Flow	Yes		Offline (Calibration)	Yes	Normal	SARIMA	Parametric
Ghosh *et al.* (2007)	Urban road	15	1	Flow	Yes		Offline (Calibration)	Yes	Normal	SARIMA	Parametric
Guo *et al.* (2012a)	Urban road	15	1	Flow	Yes	Yes	No	No	Normal & abnormal	GM	Non-parametric
Okutani & Stephanedes (1984)	Urban road	5	1 & 6	Flow	Yes		Offline (Calibration)	Yes	Normal	KF	Parametric
Stathopoulos & Karlaftis (2003)	Urban road	3	1	Flow		Yes	Offline (Calibration)	Yes	Normal	KF	Parametric
Park & Rilett (1999)	Freeway	5	1 & 5	Time	Yes	Yes	Offline (Calibration)	Yes	Normal	NN	Non-parametric

Table 2.3: Categorisation of available literature in existing traffic prediction models (Continued)

Author	Context	Input data resolution (min)	Prediction step	Input variables	Input data pattern		Training		Traffic Condition	Method	Structure
					Seasonal temporal pattern	Spatial pattern	Process	Dataset			
Huang & Ran (2002)	Urban road	5	3	Flow, speed and weather	Yes		Offline (Calibration)	Yes	Normal & abnormal	NN	Non-parametric
Park *et al.* (1998)	Freeway	5	1	Flow			Offline (Calibration)	Yes	Normal	NN	Non-parametric
Abdulhai *et al.* (1999)	Freeway	0.5	1 & 30	Flow and occupancy		Yes	Offline (Calibration)	Yes		NN	Non-parametric
Ishak & Alecsandru (2004)	Freeway	5, 10,15,20	1, 2, 3 & 4	Speed	Yes	Yes	Offline (Calibration)	Yes		NN	Non-parametric
Zheng *et al.* (2006)	Freeway	15	1	Flow			Offline (Calibration)	Yes	Normal	NN	Non-parametric
Davis & Nihan (1991)	Freeway	1	1	Flow and occupancy			Online (Lazy)	Yes	Normal	kNN	Non-parametric
Smith & Demetsky (1997)	Freeway	15	1	Flow			Online (Lazy)	Yes	Normal & abnormal	kNN	Non-parametric
Smith et al. (2002)	Freeway	15	1	Flow			Online (Lazy)	Yes	Normal	kNN	Non-parametric
Clark (2003)	Highway	10	1	Speed, flow &occupancy			Online (Lazy)	Yes	Normal	kNN	Non-parametric

Table 2.3: Categorisation of available literature in existing traffic prediction models (Continued)

Author	Context	Input data resolution (min)	Prediction step	Input variables	Input data pattern		Training		Traffic Condition	Method	Structure
					Seasonal temporal pattern	Spatial pattern	Process	Dataset			
Turochy (2006)	Freeway	15	1	Speed, flow &occupancy	Yes	Yes	Online (Lazy)	Yes	Normal	kNN	Non-parametric
Krishnan & Polak (2008)	Urban road	15	1 & 4	Flow	Yes	Yes	Online (Lazy)	Yes	Normal	kNN	Non-parametric
Tam & Lam (2009)	Urban road	5	1	Time	Yes		Online (Lazy)	Yes	Abnormal	kNN	Non-parametric
Guo *et al.* (2010)	Urban road	15	1	Flow	Yes	Yes	Online (Lazy)	Yes	Normal & abnormal	kNN	Non-parametric
Guo *et al.* (2012b)	Urban road	15	1	Flow	Yes	Yes	Online (Lazy)	Yes	Normal & abnormal	kNN	Non-parametric
Sun *et al.* (2003)	Freeway	5	Yes	Speed	Yes		Online (Lazy)	Yes	Normal	KS	Non-parametric
Huang & Sadek (2009)	Freeway	5	1	Flow	Yes		Online (Lazy)	Yes	Normal & abnormal	SPN	Non-parametric
Wu *et al.* (2004)	Highway	3	1	Time			Offline (Calibration)	Yes	Normal	SVR	Non-parametric
Castro-Neto *et al.* (2009)	Freeway	5	1	Flow	Yes		Offline (Calibration)	Yes	Normal & abnormal	SVR	Non-parametric
(Leshem & Ritov, 2007)	Urban road	30	1	Flow			Offline (Calibration)	Yes	Normal	RF	Non-parametric

Table 2.4: Characteristics of reviewed statistical and machine learning methods in short-term traffic prediction

Methods	Characteristics	Advantages	Weaknesses
Historical average (e.g. Jeffery *et al.* (1987))	• Use the historical average as the predictor • Values are pre-determined	• Computationally efficient • Simple structure	• Inaccurate during abnormal conditions
ARIMA/SARIMA (e.g. Ahmed and Cook (1979); Levin & Tsao (1980); Hamed *et al.* (1995); Williams & Hoel (2003); Ghosh *et al.* (2007))	• Statistic parametric method • Linear or non-linear • Stochastic • Seasonal temporal structure	• Simple structure • Well-established theoretical background • Computationally efficient	• Weak stationarity • Weak transferability • Inaccurate prediction during abnormal traffic conditions
GM (e.g. Guo *et al.* (2012a))	• Non-linear • Successively updates parameters with input feature vector	• Easily detects the change of traffic pattern during abnormal traffic conditions • No training procedure • Better prediction performance than ARIMA	• Requires high quality traffic data

Table 2.4: Characteristics of reviewed statistical and machine learning methods in short-term traffic prediction (Continued)

Methods	Characteristics	Advantages	Weaknesses
KF (e.g. Okutani and Stephanedes (1984); Stathopoulos & Karlaftis (2003))	• Linear or non-linear • Stochastic Gaussian nature of initial conditions • Continuously updates parameters	• Multivariate input • Flexible model structure	• Gaussian hypothesis • Requires knowledge of system's dynamics model • System must be controllable
NN (e.g. Park & Rilett (1999); Huang & Ran (2002); Park *et al.* 1998; Abdulhai *et al.* (1999); Van Lint (2004); Ishak & Alecsandru (2004); Zheng *et al.* (2006))	• Non-linear • Non-parametric • No requirements of hypothesis on the statistical nature of data	• Multivariate model • High prediction accuracy • Acceptable prediction accuracy during abnormal traffic conditions	• Requires extensive training dataset • Complex selection of model parameters
SPN (e.g. Huang & Sadek (2009))	• Using historical average data • Data merge and comparison process	• No training procedure • Transferability	• Complex model structure

Table 2.4: Characteristics of reviewed statistical and machine learning methods in short-term traffic prediction (Continued)

Methods	Characteristics	Advantages	Weaknesses
kNN (e.g. Davis & Nihan (1991); Smith & Demetsky (1997); Smith et al. (2002); Clark (2003); Turochy (2006); Krishnan & Polak (2008); Tam & Lam (2009); Guo *et al.* (2010); Guo *et al.* (2012b))	• Non-linear • Non-parametric • Pattern matching • Model free	• Simple structure • High prediction accuracy • Transferability • Robustness • Easy implementation • Acceptable prediction accuracy during abnormal traffic conditions	• Requires extensive historical dataset
KS (e.g. Sun *et al.* (2003))	• Similar to kNN method	• Simple structure	• Requires more computing time than kNN
SVR (e.g. Wu *et al.* (2004); Castro-Neto *et al.* (2009))	• Non-parametric • Map input feature vector into a high dimensional feature space	• Transferability • Acceptable prediction accuracy during abnormal traffic conditions	• Requires extensive training dataset
RF	• Non-parametric • Based on the decision tree method	• Simple structure • Acceptable prediction accuracy during abnormal traffic conditions	• Requires extensive training dataset

Table 2.5: Comparison of reviewed statistical/machine learning methods in traffic prediction

Method	**Historical data utilisation**	**Real-time data utilisation**	**Prediction accuracy**	**Robustness**	**Ease of implementation**	**Computational efficiency**
Historical average	Good	Poor	Poor	Poor	Good	Good
ARIMA	Good	Good	Fair	Poor	Poor	Fair
GM	Fair	Good	Good	Fair	Good	Good
KF	Fair	Good	Good	Poor	Fair	Fair
kNN	Good	Good	Good	Fair	Good	Fair
KS	Good	Good	Fair	Fair	Good	Fair
SVR	Good	Good	Good	Fair	Fair	Fair
NN	Good	Good	Good	Fair	Fair	Fair
SPN	Good	Good	Fair	Fair	Poor	Fair
RF	Good	Good	Good	Fair	Fair	Fair

2.7 Conclusions

This chapter has presented an overview of data prediction models in the literature, especially traffic prediction models, focusing on those based on statistical and machine learning tools. The advantages and limitations of widely used statistical/machine learning methods were compared and discussed. None of these methods can accurately and robustly predict traffic variables as well as being easy to implement during both normal and abnormal traffic conditions. Therefore, one of the objectives of this research is to develop a traffic prediction framework in conjunction with machine learning methods that will be able to address the weakness just listed. Based on the discussion in Section 2.6, five advanced machine learning tools were selected for future evaluation, investigation and application of the proposed prediction frameworks. These are the k-nearest neighbour, neural network, support vector regression, grey system and random forest methods. The proposed traffic prediction frameworks used for both normal and abnormal traffic conditions are presented in the next chapter.

Chapter 3 Short-term Traffic Prediction Frameworks

In the previous chapters, we discussed the nature of short-term traffic prediction problems and reviewed the existing literature that applies a wide range of statistical and machine learning methods to these problems. The strengths and weaknesses of the various models were also discussed. Although a wide variety of different traffic prediction methods have been published in the literature, accurate, robust and reliable traffic prediction models for practical use are still not readily available. In this chapter a novel traffic prediction framework is proposed for the short-term traffic prediction.

3.1 Background

Most studies of short-term traffic prediction focus on statistical and machine learning methods and the apparent superiority of one prediction method over others when applied to a specific short-term prediction problem. However, few studies have attempted to develop a general framework of short-term traffic prediction. Increasingly more complex machine learning methods are used and more datasets and computational power are required in traffic prediction implementation. However, the accuracy of the traffic prediction using a given model depends not only on the choice of the statistical or machine learning prediction tool, but also on the overall model structure (Krishnan & Polak, 2008).

Short-term prediction problems arise in many fields, and in some of these fields, wider prediction frameworks have been developed. Such frameworks typically

address not only the prediction step but also questions such as data cleaning and smoothing and prediction feedback (Krishnan, 2008; Simoes *et al.*, 2011). For example, in the field of hydrology, a data smoothing technique based on SVM methods was used in a rainfall prediction framework to improve the ultimate prediction accuracy (Simoes *et al.*, 2011). Similar to rainfall data in hydrology, traffic data are typically noisy because of sampling and non-sampling errors. Hence, the element of data smoothing in hydrology can be adopted and amended in short-term traffic prediction to improve prediction accuracy. As an early part of this PhD research, Guo *et al.* (2012c) demonstrated that a data smoothing structure can improve traffic prediction accuracy using the kNN method. This is because this data smoothing step can help kNN easily and accurately extract the main trends in noisy traffic data. Therefore, the stage of data smoothing is introduced to generate a 2-stage framework for short-term traffic prediction.

One of the objectives of this research is to develop a traffic prediction model to predict traffic variables under abnormal traffic conditions, and this objective raises particular challenges. Compared with the historical average traffic patterns, traffic patterns suddenly change during abnormal periods. An error feedback mechanism was demonstrated to improve short-term traffic prediction accuracy with a machine learning method during abnormal traffic conditions in an early part of this PhD research (Guo *et al.*, 2010; Guo *et al.*, 2012b). A strong correlation exists between the current prediction error and previous errors during the given time interval. The error feedback elements can use this relationship to improve prediction accuracy. Hence, a mechanism of error feedback is added to the 2-stage prediction framework to create the 3-stage framework.

In summary, this chapter focuses on a general short-term traffic prediction framework rather than a specific machine learning method. A novel 3-stage traffic prediction framework is proposed for short-term traffic prediction. This chapter is logically divided into three parts. Each part represents one stage in the prediction framework, shown in Figure 3.1. In this 3-stage framework, the first stage is to smooth traffic data in order to extract the main patterns and trends in traffic data and improve prediction accuracy. The second stage uses a machine learning method that can learn the relationship between input and output datasets which can be used for prediction calculation. The third stage adds an error feedback mechanism to the second stage. Each stage of the proposed prediction framework is introduced and discussed in the following sections.

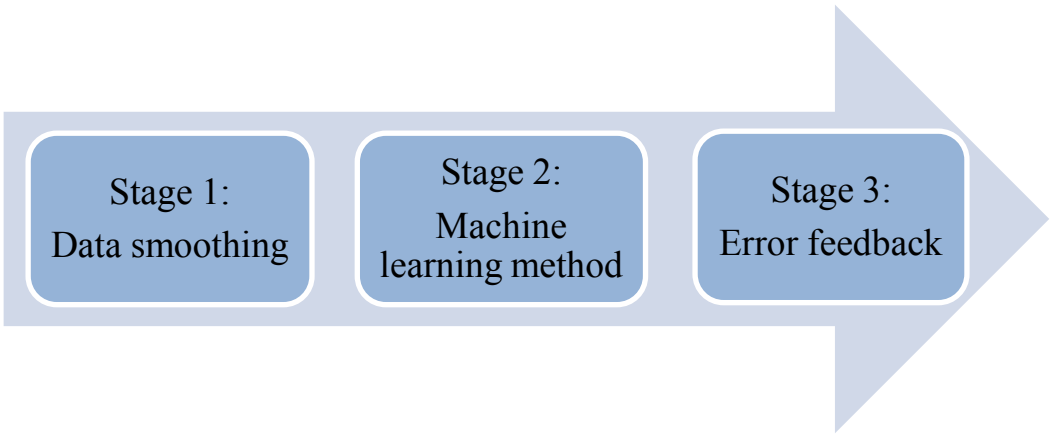


Figure 3.1: General 3-stage framework for traffic prediction

3.2 Data smoothing

This section focuses on the first stage of the proposed framework for traffic variables prediction in short-term future. The first stage using a data smoothing technique in the proposed framework is a data pre-processing step. Its aim is to smooth/de-noising input traffic data to help machine learning tool to reduce volatility and easily extract

the main real patterns and trends from the noisy traffic data. Technically, data smoothing can be considered as a form of low pass filter that can remove the high frequency noise and emphasise the low frequency components representing temporal traffic patterns (Golyandina *et al.*, 2001).

3.2.1 Overview of formal data smoothing approaches

Notwithstanding the extensive literature on alternative prediction model approaches, it is rather surprising that relatively little attention has been paid to issues surrounding the pre-processing of traffic sensor data. These data are subject to a wide range of types of sampling and non-sampling errors (Robinson, 2005) and hence are typically very noisy, especially in urban areas. In several other fields in which a prediction model using very noisy inputs is required (e.g. hydrology), it has been shown that appropriate smoothing/de-noising data pre-processing treatments can improve the ultimate prediction accuracy (Sivapragasam *et al.*, 2001; Simoes *et al.*, 2011). Robinson & Polak (2006) demonstrated that simple data cleaning processes can significantly improve the accuracy of traffic estimation models. However, the value of formal data smoothing/de-nosing techniques has not systematically explored to date in the context of traffic prediction. Hence, a data smoothing/de-noising step is introduced to short-term traffic prediction framework in this research.

Data smoothing and de-noising are two similar concepts and cannot be well distinguished in practice. The term *data smoothing* means that extraction of the smoothed component of the series; *data de-noising* is used to remove the noise from the series. Barclay *et al.* (1997) defined data smoothing and de-noising in signal processing terms as:

'smoothing removes high-frequency components of the transformed signal regardless of amplitude, whereas denoising removes small-amplitude components of the transformed signal regardless of frequency.'

If a series is considered as the sum of two components only, the smoothed part and the noise, there is no distinct border between data smoothing and de-noising in practice (Golyandina *et al.*, 2001). Hence, the terms '*data smoothing*' and '*data de-noising*' are interchangeably used in this research.

This research uses one of the most widely used techniques, SSA, since it has been shown to be one of the most effective data smoothing techniques in a variety of different applications and is a good representative of the current state of the art in data smoothing (Golyandina & Zhigljavsky, 2013). However, the proposed framework is generic and any suitable data smoothing method can be used within the framework. The introduction of SSA method is outlined in Section 3.2.2.

3.2.2 The SSA method

Singular Spectrum Analysis (SSA) is a data smoothing and de-noising method used in the analysis of time series (Broomhead & King, 1986). It is widely used in many fields such as hydrology (e.g., Sivapragasam *et al.* (2001); Simões *et al*. (2011)) and atmospheric and geophysical research (e.g., Ghil & Vautard (1991)) but has not been applied to short-term traffic prediction.

SSA that is a model-free, adaptive noise-reduction algorithm based on the Karhunen-Loeve transform (Sivapragasam *et al.*, 2001) was first published by Broomhead & King (1986). It can be used as a data de-noising method by decomposing an original time series to a smoothed trend curve and a noise series

(Hassani, 2007). Mineva & Popivanov (1996) present a comprehensive description and discussion of the SSA method and identify a number of advantages of SSA compared to other data smoothing techniques. These advantages include the ability to characterise both trend and oscillatory components, the capability to reduce local noise, enhance pattern recognition and computational efficiently. Therefore, SSA is chosen as an example of data smoothing and de-noising methods in this research.

A detailed explanation of the SSA method can be found in Chapter 1 of Golyandina *et al.* (2001). Only one-dimensional real-valued time series is considered in the basic SSA algorithm. SSA is based on the singular-value decomposition of a specific matrix constructed upon time series (Zhigljavsky, 2010). The SSA methods can be summarised in the following four steps:

Step 1: Embedding

This step is an embedding step that transfers the original one-dimensional time series to a multi-dimensional series, which can form the *trajectory matrix*.

Let $X = \{x_{1,}, \dots, x_{N,}\}$ be an original real nonzero series, where N is the length of a time series. The embedding procedure forms the L $(1 < L < N)$ lagged vectors $X_i = [x_{i,} \dots, x_{i+L-1}], 1 \leq i \leq K, K = N - L + 1$, where the value of L is the embedding dimension or called window length. This step uses embedding method in order to transfer an original series X to a *trajectory matrix,* $T_X = [X_1, \dots, X_K]^T$ with the size of $K * L$. A *trajectory matrix* is a Hankel matrix where all the elements along the diagonal $i + j = const$ are equal. Obviously, the newly-formed lagged vector is the row vector of this matrix. In other words, the trajectory matrix T_X is written as

$$T_X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{pmatrix} = (x_{i,j=1}^{K,L}) = \begin{pmatrix} x_1 & x_2 & \cdots & x_L \\ x_2 & x_3 & \cdots & x_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_K & x_{K+1} & \cdots & x_{K+L-1} \end{pmatrix}. \quad (3.1)$$

Step 2: Singular Value Decomposition (SVD)

This step uses Singular Value Decomposition (SVD) to change the trajectory matrix formed in the Step 1 into a decomposed trajectory matrix.

Applying SVD to the trajectory matrix, the matrix T_X is decomposed into $T_X = UDV^T$, where U is a $K * L$ orthonormal matrix, V is a $L * L$ square orthonormal matrix, and $D = diag(\sigma_1, \sigma_2, \dots, \sigma_r)$ is a diagonal matrix. In this step, $\lambda_1, \dots, \lambda_r$ denotes the non-zero eigenvalues of $T_X {T_X}^T$ in a decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. The corresponding singular value of the trajectory matrix T_X is $\sigma_i = \sqrt{\lambda_i}$ $(i = 1,2, \dots, r)$ and r is the rank of T_X. The diagonal matrix D can be rewritten as

$$\begin{aligned} D &= \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} + \begin{bmatrix} 0 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix} + \cdots + \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \end{aligned} \quad (3.2)$$

Therefore, the trajectory matrix T_X can be written as

$$T_X = T_{X_1} + T_{X_2} + \cdots + T_{X_r} = \sum_{i=1}^{r} u_i \sigma_i {v_i}^T, \quad (3.3)$$

where u_i and v_i are the left and right eigenvectors of the trajectory matrix. The element $u_i \sigma_i {v_i}^T$ is called the ith eigentriple of the SVD.

Step 3: Grouping

The decomposed trajectory matrix will be reconstructed in this step.

This step is a grouping step and corresponds to splitting the matrices, computed at the SVD step, into several groups and summing the matrices within each group. The grouping procedure turns a partition of the set $\{1,2,\dots,r\}$ into the collection of m disjoined subsets of $= \{I_1, I_2, \dots, I_m\}$, which is called eigentriple grouping. $T_{I_i}$is a sum of $T_j, j \in I_i$.Thus, the expansion of T_X can be written as

$$\begin{aligned} T_X &= f_{SVD}(T_1, T_2, \dots, T_r) \\ &= f_{Grouping}(T_{I_1}, T_{I_2}, \dots, T_{I_m}). \end{aligned} \tag{3.4}$$

Assume that there are only two groups of the eigentriples of the trajectory matrix, namely T_L and T_R, and $L \cup R = I$, where I is the entire set $I = 1,2,\dots,r$. Therefore, $T_L + T_R = T_I$, $T_L = \sum_{i\in L} u_i \sigma_i {v_i}^T$ and $T_R = \sum_{i\in R} u_i \sigma_i {v_i}^T$.

Step 4: Reconstruction using diagonal averaging

A new time series of length N is created by the grouped matrices T_{I_i} in Step 3.

The corresponding operation in this step uses diagonal averaging for recovery. It is a linear operation and maps the trajectory matrix of the initial series into the original series itself. In this way, a decomposition of the initial series into several additive components can be obtained.

The basic SSA algorithm can be summarised in two main stages: decomposition and reconstruction. The basic idea of the SSA approach is to undertake a spectral analysis of the raw input data in order to separate out high frequency "noisy" components thus allowing the remaining components to be reconstructed into a smoothed version of the original series. Step 1 and Step 2 are in the decomposition

stage; reconstruction stage includes Step 3 and Step 4. Figure 3.2 shows the outline and procedural steps of the SSA method described above (Golyandina *et al.*, 2001).

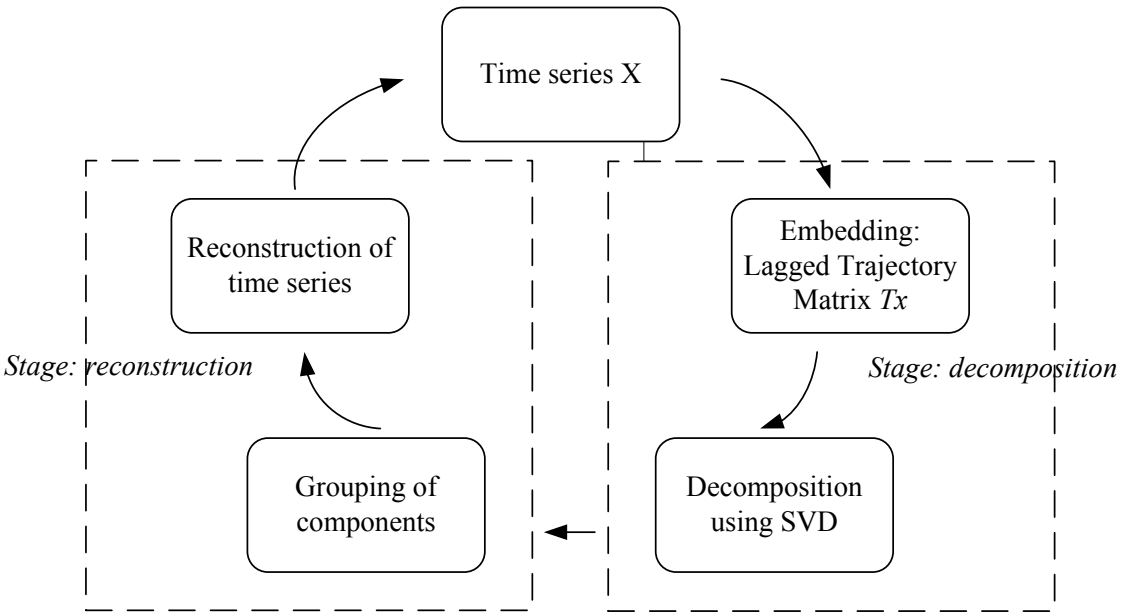


Figure 3.2: Flow-chart of a basic SSA method (adapted from Golyandina *et al.* (2001))

In this research, SSA introduced above is used in data smoothing step before a machine learning tool is applied to prediction. The original traffic time series can be divided into two parts: the smoothed series and the residuals. In Figure 3.3 it is shown a plot example of 24-hour time series traffic flow data, the smoothed part and its residuals using SSA.

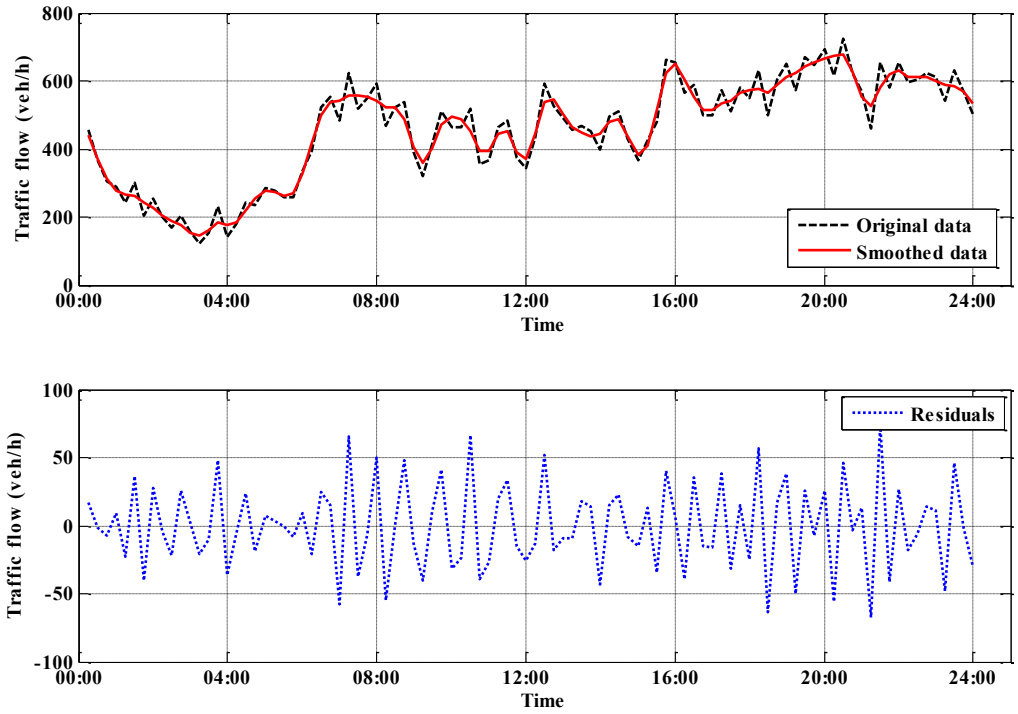


Figure 3.3: Traffic data, smoothed series and residuals

3.2.3 Prediction framework with data smoothing

The proposed framework uses the data smoothing technique of the initial traffic data. Figure 3.4 shows the flow-chart of the proposed 2-stage framework for traffic prediction. An initial time series can be decomposed into two parts by data smoothing: a smoothed series and its residual. In the application of traffic prediction, two types of data series inputs are assumed – historical (x_h) and currently observed (x_c) traffic data. Historical traffic data is used for training process; currently observed traffic data informs on the current traffic states. In the first data smoothing step, the historical traffic data x_h is decomposed into a smoothed series x_{h_s} and its residual x_{h_r} in an offline process. At the same time, the estimated residual series $\hat{x}_{h_r}$ is defined using the historical average value of the residual x_{h_r}. In the online process, the data

smoothing extracts a smoothed component x_{c_s} from the observed traffic data (and discards the residual). The final prediction result is the sum of the predicted smoothed $\hat{x}_{c_s}$ and the estimated residual $\hat{x}_{h_r}$, based on the historical data.

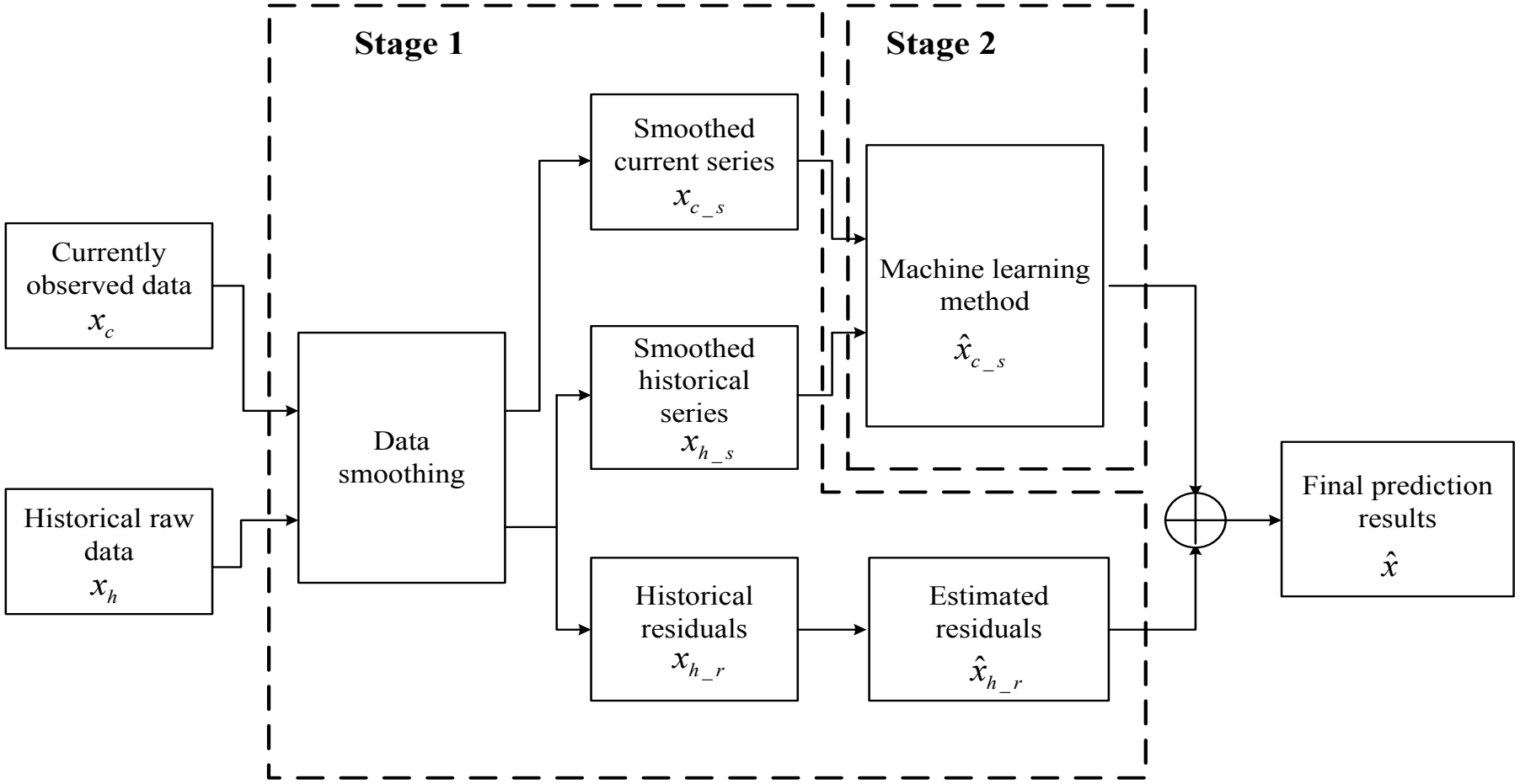


Figure 3.4: Flow-chart for the prediction framework using data smoothing

3.3 Machine learning methods

3.3.1 Introduction

The previous introduces the first stage of the prediction framework using data smoothing technique. Short-term traffic prediction is a complex dynamic problem; hence, it requires machine learning methods to deal with dynamic process. The second stage of the proposed prediction framework involves the use of any of a member of machine learning techniques to extract the relationship between input and output data datasets in a form that is useful for prediction. Five different machine learning methods are examined in this research. They are k-Nearest Neighbour (kNN), Grey system Model (GM), Neural Network (NN), Random Forests (RF) and Support Vector Regression (SVR). These are commonly used but quite different techniques for short-term traffic prediction. The proposed prediction framework is generic and any suitable statistical and machine learning method can be used within the framework. The following subsections introduce the implementation of these selected five examples of machine learning methods in the proposed prediction framework.

3.3.2 kNN

Given the structure of the kNN method, information about known traffic states and unexpected traffic conditions, including incidents and accidents, can be easily incorporated into its framework. Moreover, it has good capability of utilising available historical and current traffic data, good prediction competency, ease of implementation and computational efficiency. Therefore, the kNN based approach is

selected as one of the machine learning methods for traffic prediction under both normal and abnormal conditions in this study.

As introduced in Chapter 2, there are three key design parameters applying the kNN method to prediction. They are an appropriate definition of a distance metric to determine nearness of historical data to the current conditions, choice of k and the selection of a prediction function given a collection of nearest neighbours.

Distance metric

The distance metric is used to calculate the distance between two feature vectors. Short & Fukunaga (1981) demonstrates that distance metric can influence the results of data classification when data in the training dataset is not sufficient. However, distance metric in time series prediction with large training dataset is not the most significant component (Smith *et al.*, 2002). Euclidean distance $dist_{p,q}$ is a common method in data prediction to calculate the distance between two feature vectors x_p and x_q, given in

$$dist_{p,q}{}^2 = (x_p - x_q) \cdot (x_p - x_q)^T \tag{3.5}$$

Using calculation results of the Euclidean distance to identify k neighbours, the selected neighbour set can be defined as $x_{i,t}^{kNN}$, where $i = 1,2, \dots, k$. Its corresponding Euclidean distance is d_i and $\{i = 1,2, \dots, k\}$.

Choice of k

The value of k determines the number of nearest neighbours that are selected from the historical dataset. Too small a value of k will filter out relevant neighbours; too big a value of k will introduce noise and weaken the prediction. Stone (1977) found that the optimal value of k is data-dependent, and it usually depends on the sample size and

variability in data. In academic literature of short-term traffic prediction, the value of k is selected from 10 to 100 (Davis & Nihan, 1991; Smith *et al.*, 2002; Krishnan & Polak, 2008). Guo *et al.* (2012c) tested the sensitivity of the value of k that was chosen between 10 and 100 in increments of 10 for the same dataset used in this PhD research. The results showed that the prediction model had the most accurate results using the value of $k = 30$.

Prediction function

The purpose of a prediction function is to estimate the future values. This is the most important aspect in the design of kNN method. Smith *et al.* (2002) demonstrated that traffic prediction accuracy varies according to the type of prediction function used. The overview of the most commonly used prediction functions are presented as follows.

- *Arithmetic average:*

  This is the simplest and straightforward function to calculate future traffic variables. It computes a straight average of the dependent variable values of the neighbours in the identified neighbourhood. The prediction result $\hat{x}$ can be calculated by the equation below

$$\hat{x} = \left(\frac{1}{k}\right) \sum_{i=1}^{k} x_{i,t+h}^{kNN} \tag{3.6}$$

  where h is the prediction horizon. However, it ignores that models should not put the same weight on the data in the "nearest" neighbour dataset. In other words, the "nearer" dataset plays a more important role in prediction (Smith et al., 2002). Moreover, Robinson (2005) states that another criticism of the use of arithmetic average is that *'these estimators are susceptible to boundary*

bias'. To avoid the predicted bias caused by using same weights, a weighted averaged by inverse of distance function is proposed.

- *Weighted average by inverse of distance*:

  This function assumes that the data from closer neighbours will provide better prediction information. Therefore, it uses the inverse Euclidean distance as the weight of each point in the "nearest" neighbour dataset. The equations are given by

$$\hat{x} = \sum_{i=1}^{k} \frac{x_{i,t+h}^{kNN} * w_i}{\sum_{i=1}^{k} w_i} \tag{3.7}$$

$$w_i = \frac{1}{d_i} \tag{3.8}$$

  where d_i is the Euclidean distance.

- *Adjusted by current variable*:

  This function assumes that the value of predicted variable is strongly related to the current variable. The equation is given by

$$\hat{x} = \left(\frac{1}{k}\right) \sum_{i=1}^{k} \frac{x_{i,t+h}^{kNN} * x_{c,t}}{x_{i,t}^{kNN}} \tag{3.9}$$

  where $x_{c,t}$ is the current traffic variable at time t.

- *Regression:*

  A linear regression method can be used to estimate future values for input feature vector using k nearest neighbours. An Ordinary Least Squares (OLS) (Amemiya, 1985) regression method is used by Mulhern & Caprara (1994) in market forecasting in a kNN method. The definition of OLS in kNN prediction method is given below

$$\hat{x} = X_{i,t+h}^{kNN}\beta + \varepsilon \quad (3.10)$$

where β is the unknown parameter and ε is the error term. Robinson (2005) compared the OLS based regression function against arithmetic average method and found that the regression function gave a better estimation result.

To investigate the predictive performance using different prediction functions, Smith *et al.* (2002) and Guo *et al.* (2012c) compared the prediction functions of *arithmetic average*, *weighted average by inverse of distance* and *adjusted by current variable*. Smith *et al.* (2002) observed that prediction based on kNN using the function *adjusted by current variable* had the best performance in terms of Mean Absolute Percentage Error (MAPE) in short-term traffic flow prediction on motorways under normal traffic conditions. In the early part of this research, Guo *et al.* (2012c) demonstrated the above results using traffic flows in the central of London under normal traffic conditions and found that prediction using the function *adjusted by current variable* also had the best performance during incident conditions. Therefore, *adjusted by current variable* function is selected for the implementation of kNN method in the proposed traffic prediction framework.

3.3.3 GM

GM is selected as a machine learning method in the traffic prediction model because of its reduction of the dependency on method training and parameter optimisation. In grey system theory, a grey system GM(*n,m*) can dynamically update parameters based on the relationship between feature vectors. The GM model constructs a differential equation to describe the unknown system. The output of GM can be calculated by solving the differential equation.

In GM(n,m), n is the order of the differential equation and m is the number of variables. Various types of grey system models can be found in the literature; however, a GM($1,1$) model is most commonly used because of its performance and computational efficiency, which are important design parameters in practice. Trivedi & Singh (2005) presented the reasons why first-order differential equation was selected in the context of mathematics and practical implementations. Therefore, a GM($1,1$) model is selected in this research for future investigation. The model parameters $[a, u]$ of GM($1,1$) in Equation 2.10 are updated with new observations and are not required to pre-determine.

3.3.4 NN

The topology of feed-forward neural network (FFNN) is implemented using neural network toolbox in MATLAB (Beale *et al.*, 2012). There are two stages in the process to implement NN: training and predicting. Before these two stages, the network inputs are required to scale so that data might fall approximately in the range $[-1,1]$. In the training stage, a Levenberg-Marquardt backpropagation algorithm (Zurada, 1992) is used as the training function in the toolbox (Beale *et al.*, 2012). The function of tan-sigmoid is used as transfer function for hidden layers; a linear transfer function is used as transfer function for output layer (Beale *et al.*, 2012). To achieve an optimised network performance, the neural networks need to be trained by adjusting the weight values and reducing network bias. The criterion of Mean Square Error (MSE), which is the average squared error between the network outputs and the target outputs, is used to evaluate the training performance. The definition of MSE is given by Equation 3.11.

$$MSE = \frac{1}{n}\sum_i^n (e_i)^2 = \frac{1}{n}\sum_i^n (output_i - target_i)^2 \qquad (3.11)$$

where n is the sample size, output_i the predicted results for i and $target_i$ the real value for i. The training process will be stopped when MSE is less than a pre-determined value.

In the stage of prediction, the model that is created and trained in the training stage can be used to calculate the network output of new input data for testing. Unlike kNN and GM methods, a NN method creates an acceptable model off-line and applies it to new input data to calculate prediction. Figure 3.5 shows the stages of training and predicting using a NN method.

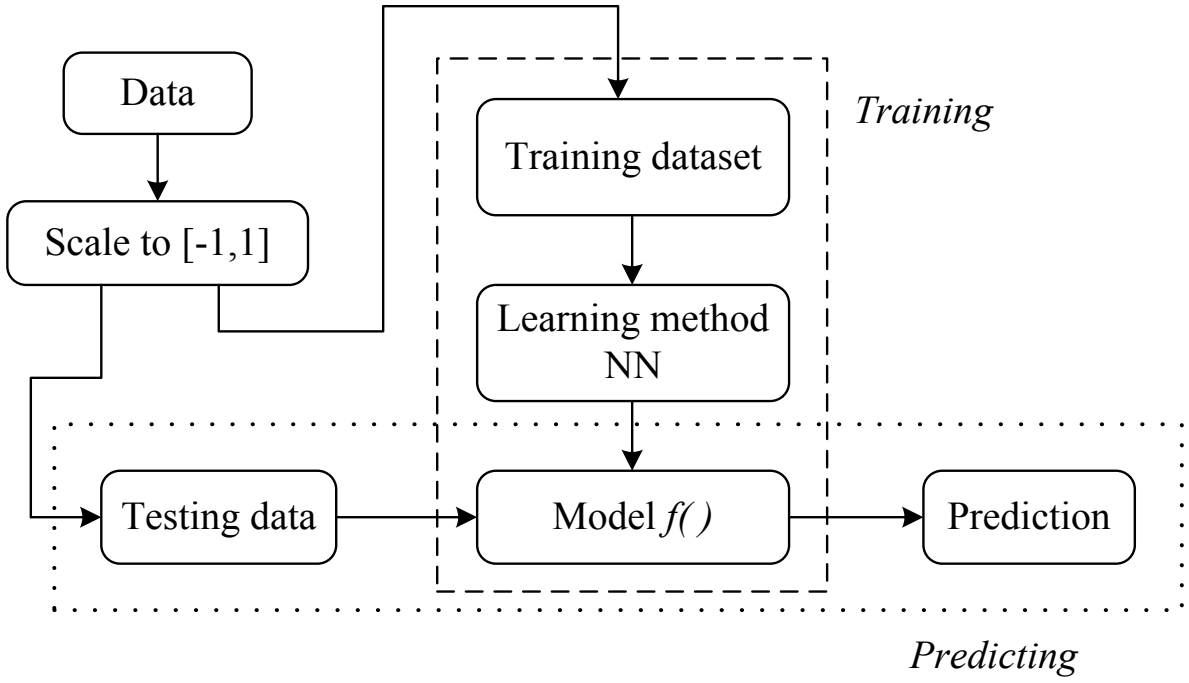


Figure 3.5: Process of NN based method for prediction problems

In the neural network toolbox of MATLAB, some additional settings that need to be determined before training process include learning rate and momentum.

- *Learning rate* (η)*:*

In the training process, the network can optimise the bias and link value of each direction to compute a more accurate output. The rate of improvement can be known.

A learning rate (η) is pre-designated to determine how much the link weights and node biases can be changed in each epoch. A small learning rate can reduce the network's computational efficiency. The network may become unstable such as oscillatory using a larger value of learning rate. The parameter of learning rate is often set as a small positive value less than 1.

- Momentum (α)*:*

  A back-propagation algorithm is used in neural networks to avoid the risk of instability. The term of momentum (α) used in a back-propagation algorithm can help learning rate to stabilise the weight change. The value of momentum α is commonly set [0, 0.9]. This process can be describe using the Equation 3.12

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_i(n) y_i(n) \qquad (3.12)$$

  where

  α: the momentum value

  δ_i: the local gradient of neuron j

  $w_{ji}(n)$: the weight between neuron j and i at iteration n

  y_i: the output of neuron i.

3.3.5 RF

A free software package of random forest that includes an interface of the R statistical software (Venables *et al.*, 2011) to the Fortran programming language (Adams *et al.*, 2008) is used in this research and available at:

http://www.stat.berkeley.edu/users/breiman/RandomForests/.

The original code of random forest method was created by Breiman and Cutler in the Fortran programming language (Press *et al.*, 1992).

Liaw & Wiener (2002) introduced the usage and features of this R function of random forests method in the applications of classification and regression problems. There are only two parameters that need to be pre-determined in the implementation, namely m_{try}, the number of variables in the random subset at each node and n_{tree}, the number of trees in the forest (Liaw & Wiener, 2002). The default value of m_{try} is the dimension of features in the package. The default value of n_{tree} is 500. Based on the suggestion of Breiman & Cutler (2005), Liaw & Wiener (2002) used the default m_{try}, half of the default m_{try} and twice the default in their experiments. The results did not dramatically change according to different values of m_{try} and the random forest function is not sensitive to the value of m_{try}. Genuer *et al.* (2010) investigated the selection of n_{tree} in random forest method. They used two values of n_{tree}, the default 500 and 2000. Genuer *et al.* (2010) suggested that the default value of n_{tree} should be used at first and only change the value when prediction result is not acceptable, because the results showed that the effect of n_{tree} is less visible. Therefore, it seems that the random forest package is user-friendly regarding the selection of parameters and the default value of n_{tree} is used in this PhD research.

3.3.6 SVR

A free software package named mySVM (Ruping, 2000) is used in this prediction framework to carry out short-term traffic prediction. The core of mySVM is based on the optimisation of SVM^{light} a free software package developed by Joachims (1999) in the C programming language (Kernighan & Ritchie, 1988). SVM^{light} is an implementation of the Support Vector Machine (SVM) introduced by Vapnik (1998) used for both classification and regression. Because SVR is originated from basic SVM theory, the technique of SVM used for regression is named SVR. The mySVM package is used in pattern recognition, regression and distribution estimation and can be found at:

http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html.

The parameters of mySVM that need to specify are the choice of parameter C, ε and the choice of kernel function.

- *choice of parameter C in Equation 2.21*

  The user-defined constant C is the capacity parameter of the SVR and must be positive (Ruping, 2000). It determines the trade-off between the model complexity and estimation errors (Hastie *et al.*, 2001; Cherkassky & Ma, 2004; Wu *et al.*, 2004). For example, when the value of *C* goes to infinity, the SVR model would not allow any estimation errors without considering the model complexity in the training process. Cherkassky & Ma (2004) proposed the equation to determine the value of C that is

$$C = max(|\overline{y} + 3\sigma_y|, |\overline{y} - 3\sigma_y|) \quad (3.13)$$

where $\overline{y}$ is the average of the outputs in the training dataset and σ_y is the standard deviation of the training output y.

- *choice of parameter ε in Equation 2.22*

  Parameter ε represents the width of ε-intsensitivity area and is used to fit the training dataset (Cherkassky & Mulier, 2007). Cherkassky & Ma (2004) suggested that the choice of ε should be $\varepsilon \in (0,0.1]$.

- *kernel function*

  The selection of kernel function is usually based on a priori knowledge of application domains (Schölkopf *et al.*, 1998; Chapelle & Vapnik, 1999; Cherkassky & Ma, 2004). A Radial Basis Function (RBF) kernel is most commonly used is the SVR models for classification and regression problems.

3.4 Additional input variables

3.4.1 Background

The nature of input information in traffic prediction models is of importance to predictive accuracy. Most academic studies use two types of additional input explanatory variables, namely multi-input variable and the temporal and spatial relationship.

From the first perspective, more than one input parameter can be chosen in the models to improve prediction accuracy. For example, Florio & Mussone (1996) used traffic flow, density, speed and other additional normalised information such as the percentage of heavy vehicles, brightness, weather conditions, visibility and the presence of information on variable message signs (VMS) to predict traffic variables.

In a more recent instance, Huang & Ran (2002) predicted traffic speed using weather information such as visibility, temperature and moisture as additional explanatory variables. However, the limitation of additional information collection will restrict the application of this method in practice. Moreover, the quality of additional information will also affect the accuracy of the prediction of traffic variables in the models.

More research has concentrated on the temporal-spatial relationship of the input traffic data. Krishnan (2008) discussed the recurrent nature of traffic flow and travel time patterns at a daily and weekly level. Various traffic prediction models in the literature make use of this recurrent nature of traffic patterns to improve traffic prediction accuracy. For example, Williams & Hoel (2003) exploited the recurrence of traffic data using a SARIMA model with a seasonal lag of one week to improve the accuracy of traffic flow prediction. In their prediction model, weekly periodicity information is added to the currently collected traffic data.

Moreover, the historical average traffic data at the given time-of-the-day and the given location is also used in prediction models to improve accuracy. For example, Zhang & Rice (2003) predicted travel time using both historical average travel time data and currently observed traffic data in a regression framework. Krishnan & Polak (2008) used historical average traffic flows as an additional explanatory variable to predict one-step and multi-step ahead traffic flows under normal traffic conditions. In this model, the historical average data are obtained by calculating the mean traffic flow in the training dataset for every 15-minute period for the same day of the week and time of the day. Adding historical average traffic flow profile helps prediction model improve the prediction accuracy for both one-step and multi-step ahead prediction. More recently, Guo *et al.* (2010) and Guo *et al.* (2012b) used the same model adding historical average data as an explanatory variable with different

machine learning tools in traffic flow prediction under normal and abnormal traffic conditions. The prediction results show that the historical average improves the prediction performance under normal traffic conditions; however, historical average information does not improve prediction accuracy during incidents. When traffic regime suddenly changes, prediction models should not put much weight on historical information.

In the literature, some prediction models use additional data from upstream locations to predict downstream traffic variables. For example, Kamarianakis & Prastacos (2005) used Space-Time Autoregressive Integrated Moving Average (STARIMA) model to predict traffic flows using spatial information. This model considered both the recurrent nature of traffic patterns and additional data from upstream locations and assumed that the spatio-temporal autocorrelation in input data can be described by fixed parameters. However, this assumption is difficult to be satisfied in practice. Cheng *et al.* (2011) discussed the temporal and spatial relationship using real-world data from urban network. A traffic network, especially urban network, is dynamic not stationary. Hence, the fixed parameter used to represent the temporal and spatial relationship of traffic data might affect prediction performance, particularly in the presence of non-recurrent abnormal traffic conditions (Cheng *et al.*, 2011). They found that it is not easy to build complex spatio-temporal autocorrelation structure using data in real time. Thus, space-time models such as STARIMA that rely on the fixed parameter assumption are likely to have low prediction power.

In summary, there are many types of additional explanatory variables in short-term traffic prediction models that may affect prediction performance. Two typical types of additional explanatory variables are discussed above. The applications of

these explanatory variables discussed above are based on the accessibility of additional traffic information and less-dynamic traffic pattern. The selection of these different additional explanatory variables should depend on the data and the context where prediction models are implemented. One of the objectives of the PhD research is to improve prediction accuracy during abnormal traffic conditions. Traffic patterns would suddenly change during abnormal traffic conditions. Moreover, information about abnormal events such as duration and the degree of severity is only accessible in the offline process rather online applications. Therefore, the above variables are not used in the short-term traffic prediction framework.

3.4.2 Error feedback structure

The recurrent nature of traffic process might change with the occurrence of abnormal events. A traffic prediction model should be dynamically self-adaptive in response to such change to carry out an accurate prediction result. Krishnan (2008) found a strong correlation between current prediction error and the previous errors during the given time interval. An error feedback mechanism that makes use of this error relationship was added to models to improve short-term traffic prediction accuracy during normal traffic conditions. In the prediction model, the prediction errors calculated from the previous time intervals are used to update the prediction generated for the current time interval. Guo *et al.* (2010) used the same model structure to predict traffic variables during abnormal traffic conditions. They found that for prediction during normal conditions prediction models with error feedback are slightly more accurate than the models without error feedback, of which the results are similar as the ones demonstrated in Krishnan & Polak (2008) and Krishnan (2008); however, the significant advantage of error feedback is the improvement of prediction accuracy

during abnormal conditions. Therefore, this research uses error feedback structure that has been proven helpful in short-term traffic prediction during abnormal conditions. This proposed 3-stage traffic prediction framework is summarised in Figure 3.6.

The structure of error feedback is described below.

$$\hat{\varepsilon}_t = \sum_{k=h}^{t_{lag}} \beta_k * \varepsilon_{t-k} + \varepsilon'_t \tag{3.14}$$

$$\hat{x} = \hat{x}_t - \hat{\varepsilon}_t \tag{3.15}$$

where

$\hat{\varepsilon}_t$: estimated prediction error (or called the feedback term)

h: prediction horizon

t_{lag}: maximum lag of the error model

β: constant parameter

ε_{t-k}: prediction error using machine learning method

ε'_t: error term in regression

$\hat{x}$: final prediction result

$\hat{x}_t$: prediction result using machine learning method.

This error feedback mechanism can deal will the difference between current traffic patterns and the historical average traffic patterns. The estimated prediction error calculated using errors from the previous time intervals can correct prediction result for the current time interval. Some methods such as ARIMA and the Kalman filter use the similar concept and principle to update the prediction results. However,

ARIMA is required to predetermine the model parameters. As a part of this research, Guo *et al.* (2013) demonstrated that the pre-calibrated ARIMA was not accurate in short-term traffic prediction under abnormal traffic conditions. As discussed in Section 2.4.4, Kalman filter is a state-space model with multivariate inputs. One of the main weaknesses of Kalman filter is that the previous knowledge about the system and measuring devices is required. Another disadvantage is its reliance on some fundamental assumptions such as measurement noises are white and Gaussian distributed. These assumptions limit the real applications of a Kalman filter based system (Maybeck, 1979). The prediction of Kalman filter and the proposed error feedback structure will be compared in next chapter.

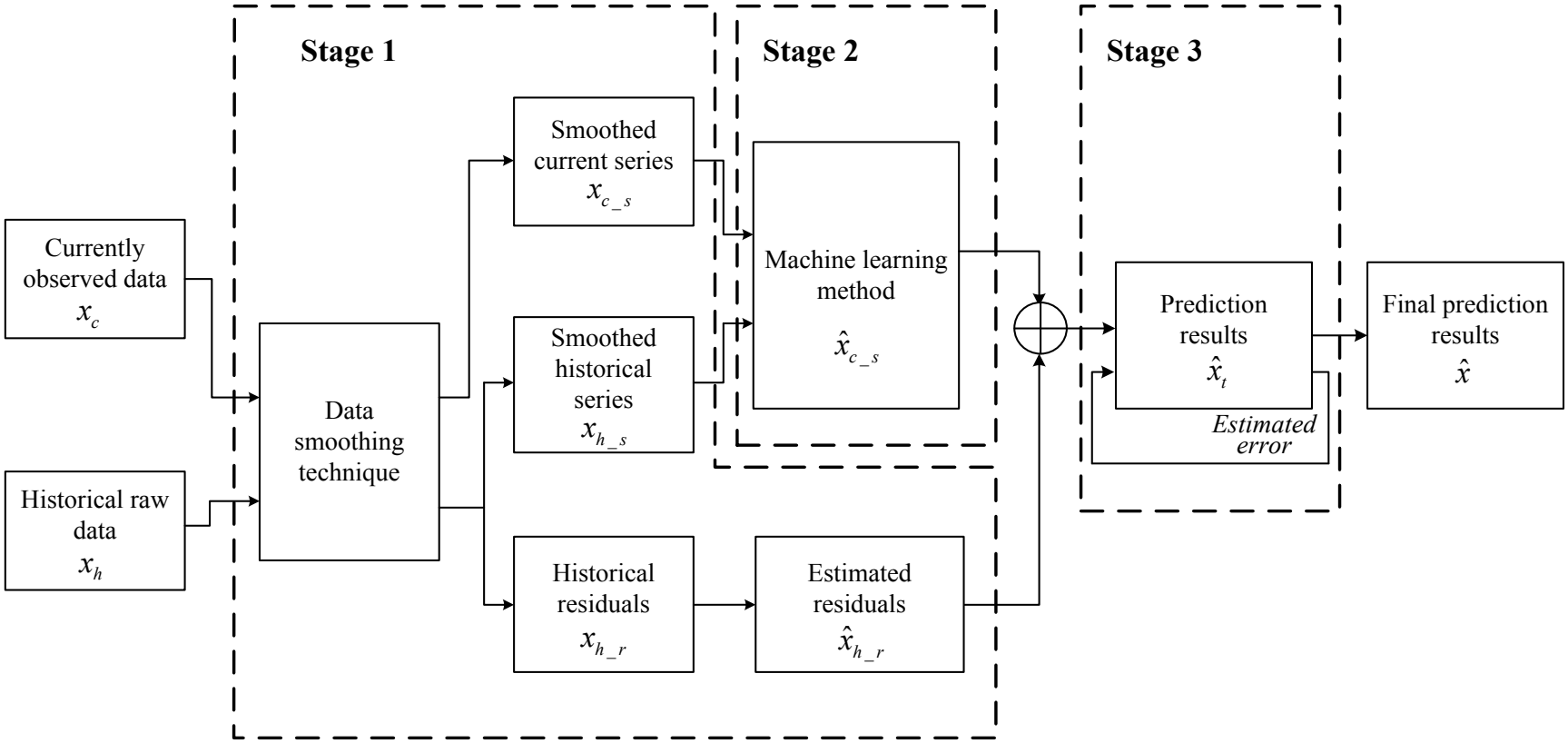


Figure 3.6: Flow-chart of the proposed 3-stage short-term traffic prediction framework

3.5 Quantification of prediction accuracy

The prediction accuracy is evaluated using three criteria, namely Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). MPE is the average of the percentage errors for a given dataset during a specific period and is used to calculate prediction bias. MAPE calculates the average of the absolute difference between predicted and actual values. Both positive and negative predictive errors can be considered in the accuracy measurement. Compared to MAPE, RMSE gives additional weight to larger absolute errors. Taken together, these three measures evaluate an assessment to be made of accuracy and precision of prediction reference. These measures are defined as follows:

Mean Percentage Error (MPE):

$$MPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{x}_i - x_i}{x_i}\right) * 100\% \tag{3.16}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|x_i - \hat{x}_i|}{x_i}\right) * 100\% \tag{3.17}$$

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\sum_{i=1}^{n}\left(\frac{|x_i - \hat{x}_i|^2}{n}\right)} \tag{3.18}$$

where,

x_i: observed traffic variable

$\hat{x}_i$: predicted traffic variable

n: number of predicted time intervals.

3.6 Summary

This chapter focused on the presentation of the proposed traffic prediction framework. The short-term traffic prediction model is developed through mainly three stages, including data pre-processing stage using data smoothing technique (Section 3.2), prediction stage using machine learning method (Section 3.3) and error feedback mechanism (Section 3.4). The quantification of prediction model is then introduced in Section 3.5. In Chapter 4 and Chapter 5, this proposed prediction framework is implemented and tested using both simulation data and real-world traffic data.

Chapter 4 Evaluation of Proposed Traffic Prediction Frameworks Based on Simulation Experiments

Three short-term traffic prediction frameworks were presented in the previous chapter. This chapter presents a traffic simulation model of a corridor in Southampton to simulate a number of abnormal traffic conditions and uses link travel time data generated from this simulation model to evaluate the accuracy of the proposed frameworks for short-term traffic prediction. The results of the prediction process under different traffic conditions for each of the three prediction frameworks and the five machine learning methods are comprehensively presented in this chapter.

4.1 Background

It is not easy to obtain traffic data covering a wide range of traffic conditions in the real world. Therefore, the main purpose of the simulation experiments in this PhD study was to generate link travel time data under a range of different traffic conditions in order to examine the impacts of data smoothing and error feedback structures on the accuracy of short-term traffic prediction.

A simulation system may be defined as *'a dynamic representation of some part of the real world, achieved by building a computer model and moving it through time'* by Drew (1968). Simulation modelling tools are used by transportation engineers to examine traffic states and to understand the interaction of vehicles, infrastructure and traffic management and control. Based on the level of detail which the simulation

seeks to represent, traffic simulation systems can be categorised into three types: macroscopic, mesoscopic and microscopic (Huang & Pan, 2007).

4.2 Microscopic traffic simulation

A traffic simulation tool provides an environment where different scenarios can be introduced and evaluated in a controlled setting without disrupting traffic conditions on the roads in the real world. It can also provide facilities for modelling the effects of vehicle detectors, traffic signal control systems and static and dynamic route guidance. There are three commonly used commercial traffic simulation tools: AIMSUN (TSS, 2010), PARAMICS/S-PARAMICS (Quadstone, 2003; SIAS, 2005) and VISSIM (PTV, 2009). These are developed based on different theories of microscopic traffic behaviour, such as car-following, lane-changing and driver behaviour (Dia & Cottman, 2006).

4.2.1 Selection of traffic simulator

Many studies have attempted to evaluate the effectiveness AIMSUN, PARAMICS and VISSIM. Perales Roehrs (2001) compared the performance of VISSIM and PARAMICS in respect to traffic incident modelling and recommended both for incident simulation. Xiao *et al.* (2005), meanwhile, presented a simulation model selection process between AIMSUN and VISSIM, taking into account quantitative and qualitative evaluation criteria. They found that AIMSUN and VISSIM can incorporate most standard features used in traffic modelling and that the accuracy of both simulators was similar. In their evaluation, each criterion was assigned a grade based on the simulator's performance. The qualitative evaluation criteria included functional capabilities, input/output features, ease of use and the quality of service

provided by the developers. Goodness-of-fit measures and completion efforts for calibration were used as quantitative criteria. To model the impact of incidents, the functions of lane blocking and capacity reduction were compared. AIMSUN was given a slightly higher score in the evaluation of the simulation of abnormal traffic conditions compared with PARAMICS.

The main features of these simulators are summarised in Table 4.1. In a simulator, both car-following models and lane-changing models are the key components of a traffic-flow model. The algorithms used in these models have been developed based on a variety of theoretical backgrounds. The first explicit comparison of car-following models in AIMSUN, VISSIM and PARAMICS was by Panwai & Dia (2005). Their results showed that the Gipp's car-following model used in AIMSUN has the lowest error. The findings of a qualitative test to calculate the distance between leader and follower vehicles also confirmed the above results. Very few studies, however, have evaluated the performance of the underlying algorithms in lane-changing models using the same dataset in commonly used traffic simulators.

Overall, the literature would tend to suggest that the differences between AIMSUN, VISSIM and PARAMICS in terms of their simulation of traffic networks under abnormal traffic conditions are not obvious. As Zhang & Hounsell (2010) suggest, therefore, the selection of a simulator should depend on the specific requirements and objectives of an experimental study.

Table 4.1: Main features of three simulators

Basic features	AIMSUN	VISSIM	PARAMICS
Vehicle types	Vehicle types include car, taxi, private-bus, public-bus, HGV, truck, ambulance, police-car and HOV-car	Car, LGV, HGV, bus, articulated bus, tram and new types can be added by user	The characteristics of the vehicles such as length, width, height and speed are adjustable.
Car following	Based on Gipp's car-following model. Vehicles are classified as free or constrained by the vehicle in front. When constrained by the vehicle in front, the follower tries to adjust its speed to obtain safe space headway to its leader.	Based on Weidermann's model. The basic concept is that the driver of a faster moving vehicle starts to decelerate as he reaches his individual perception threshold to a slower moving vehicle.	Based on the Psycho-physical model developed by Fritzsche (1994). The difference between the published Fritzsche model and the model implemented in Paramics are not publicly known.
Lane changing	Based on Gipp's lane-changing model. Lane change is a decision model that approximates the driver's behaviour.	Based on Weidermann's model (Wiedemann, 1974). Vehicle dynamics combine a mixture of driver behaviour and some limitations based on vehicles' physical type and kinematics.	Change occurs when a gap in the target lane is available and adjacent to the simulated vehicle (gap acceptance).
Route choice	The default Route Choice models available are: Proportional, Multinomial Logit and C-Logit, but the user can also define his/her own user-defined route choice model using the function editor.	In VISSIM there are basically two methods to model automobile routing information. These are static routes using routing or direction decisions, and dynamic assignment.	Route choice is based on route cost tables and allows vehicles to dynamically re-route as costs vary.
Abnormal condition	Abnormal conditions such as traffic incidents and events can be simulated in AIMSUN by lanes blockage over a certain time period. Incidents include: a heavy goods vehicle loading/unloading, a taxi picking up/dropping off a passenger, a broken down vehicle, road works and etc.	The effects of temporary lane blockages have been modelled to simulate abnormal traffic conditions in VISSIM. These conditions are modelled either by time dependent and lane specific speed reductions or by stopping "dummy" PT vehicles for specified amounts of time.	PARAMICS can simulate abnormal traffic conditions such as breakdowns or accidents. The incidents can be modelled in two ways. Either incidents occur at a specific time or can occur at a specified rate. The duration of each incident must be coded by the user.
Outputs	Outputs include flows, speeds, travel times, etc.	Traffic data include traffic volume, queues, and delays.	PARAMICS outputs provide the statistical variables of all links.

Taking the above into account, the selection of the simulator for use in this research was based on the ability to simulate urban networks, and to model abnormal traffic conditions, the availability to connect to traffic signal control systems such as a UTC emulator, the convenience of storing output files in a database and the accessibility of the simulator.

Based on these criteria, the AIMSUN simulation model, developed by the Department of Statistics and Operational Research, Universitat Poletecnica de Catalunya (UPC), was selected for the generation of link travel time data. AIMSUN is readily available and has been demonstrated to simulate urban road networks under both normal and abnormal traffic conditions (Barceló *et al.*, 2002; Barcelo *et al.*, 2005; Hadi *et al.*, 2007). For example, Barceló *et al.* (2002) used AIMSUN to simulate the impacts of traffic incidents and road works by blocking roads in the simulator so as to evaluate the effectiveness of designed incident management strategies in urban and interurban areas. Hadi *et al.* (2007), meanwhile, examined the modelling of incidents in AIMSUN and related link capacity reduction because of incidents. Model parameters in AIMSUN were calibrated to achieve target link capacity values for both normal and incident traffic conditions.

Most micro-simulation models use a Graphical User Interface (GUI) to record outputs of simulation experiments. AIMSUN, however, has more flexible facilities for storing simulation outputs than other simulators since it generates an output ASCII file that can be read in an Excel or Access database. In AIMSUN, in contrast to PARAMICS, simulation results can also be directly reported in user-defined time intervals.

In addition, AIMSUN is easier to connect to UTC than PARAMICS and VISSIM. More detailed information on the communication between AIMSUN and UTC will be presented in subsection 4.3.2.3.

It was concluded, therefore, that AIMSUN was best placed to meet the specific simulation requirements of this research. The theories of microscopic traffic behaviour in AIMSUN are described in Appendix C.

4.2.2 Benefits and challenges of AIMSUN simulator

4.2.2.1 Benefits of simulation

As briefly presented above, the main purpose of using a simulation model in this study is that it can generate link travel time data for urban areas, which in turn enables the evaluation of the proposed traffic prediction frameworks under both normal and abnormal traffic conditions. The use of a simulation model offers a more feasible approach to model traffic states and provides an environment where different scenarios can be introduced.

At the end of each simulation cycle the outputs provided by a simulation model include traffic data generated by simulated data collection devices, such as link travel time, traffic flow and occupancy. This integrated traffic data is another advantage of using a simulation model to generate traffic data for framework evaluation. A simulation model can also simulate traffic states during different abnormal traffic conditions, which might not be easily observable on-street. Hence, simulation experiments have been carried out to provide an initial test of the proposed prediction frameworks before re-testing using real-world data.

4.2.2.2 Weaknesses of simulation

Although there are many benefits of using simulation models, the weaknesses should not be neglected. Road traffic networks are complex systems involving many factors, such as the human factors involved in driving behaviour, infrastructure characteristics, in-vehicle technologies (e.g. navigation system), traffic control/management strategies and detector errors. Most simulation packages can replicate the real traffic situation in some aspects but have limitations in other aspects (Zhang *et al.*, 2012).

AIMSUN uses models of car-following, lane changing and gap acceptance to represent vehicles' behaviour. The car-following model of AIMSUN is evolved from Gipps (1981). The Gipps car-following model is a safety distance model to avoid collision using pre-determined parameters. Lee (2007), however, found that, in real-life, car-following behaviour may vary with traffic conditions and the characteristics of drivers; thus, car-following models such as the Gipps model may have specific limitations under some traffic situations. In other words, the car-following model in simulations cannot always accurately model real-world phenomena.

The lane changing model of AIMSUN is based on Gipps' model (Gipps, 1986). The AIMSUN simulation model can simulate traffic incidents and accidents; however, no information is provided about lane changing under incident and accident conditions (Hidas, 2002).

The simulation experiments are carried out on the assumption that there are no errors in detector measurements. In real-life however, collected traffic data is not error free. Simulation results should add measurement errors to create a more realistic environment; however, perfect models of measurement errors in traffic sensor

systems are not currently available and, hence, this research will not introduce measurement errors.

In reality, a traffic control system is required to optimise traffic networks. Manual operation by a skilled and knowledgeable traffic engineer or manager is sometime needed to adjust traffic signal timing plans. AIMSUN cannot fully simulate this manual operation of traffic control.

Another weakness of simulation is that when simulating abnormal events the number of lanes blocked must be an integer number. In reality, however, most incidents, such as vehicle breakdowns and crashes, block part of a lane rather than the whole lane. AIMSUN is unable to simulate such abnormal events fully because of this phenomenon. In addition, the immediate disappearance of a lane blockage represents the clearance of abnormal events in AIMSUN. In reality, however, the process of removing broken vehicles, wreckage and other items from the roads needs time and gradually shrinks the incident scenes rather than causing them to disappear immediately.

In simulation experiments, incidents and abnormal events can also be simulated using public transport vehicles such as buses to stop at designated bus stops during a pre-defined period of time. It is not possible to model incidents that block more than one lane.

4.3 Description of the simulation setup used

4.3.1 Scenario design in simulation experiments

In this chapter, a simulation model was used to model traffic states under both normal and abnormal traffic conditions. Abnormal traffic conditions are non-recurrent and usually caused by a temporary and unexpected reduction in capacity because of incident or accidents. AIMSUN does not provide a function to simulate abnormal traffic conditions directly but such conditions can be modelled by using the function of lane closure at a given location for a specific time period.

It is possible for abnormal traffic conditions to be caused by extreme weather conditions and by planned event such as sports or cultural activities. For the purposes of this research, however, these are considered to be atypical and of less interest because of the ability to plan strategies ahead of their occurrences.

Some possible attributes that can describe lane closure include:

- Number of lanes closed – number of lanes blocked to simulate abnormal events
- Traffic demand level during closure – the total number of vehicles in the traffic network during lane closure
- Duration of closure – the total time from the start of lane closure to the clearance of the closure

Based on these possible attributes, the prediction accuracy of the proposed traffic prediction framework was assessed under three simulated scenarios, which are described below:

- **Scenario 1 -** The basic scenario models normal traffic conditions without incidents or accidents.
- **Scenario 2 -** In this scenario, one lane of two is blocked at a specific location to simulate abnormal traffic conditions during a given period. The level of lane closure is 50%.
- **Scenario 3 -** In this scenario, all lanes (two lanes in this simulation model) are blocked at a specific location to simulate abnormal traffic conditions during a given period. The level of closure is 100%.

In Scenarios 2 and 3, the selected duration of lane closure is predefined to 30 minutes, 60 minutes and 90 minutes. These different duration are also combined with two different traffic demand levels (off-peak and peak periods). Table 4.2 gives the levels of each attribute used in Scenario 2 and Scenario 3. Hence, in Scenario 2 and Scenario 3, a total twelve simulation runs are required to simulate the full range of abnormal traffic conditions. Figure 4.1 shows the scenario design of the simulation experiments in this chapter.

Table 4.2: Attributes and levels used in Scenario 2 and Scenario 3

	Level 1	**Level 2**	**Level 3**
Duration of closure	30 minutes,	60 minutes	90 minutes
Traffic demand level	Low: off-peak period	High: peak period	

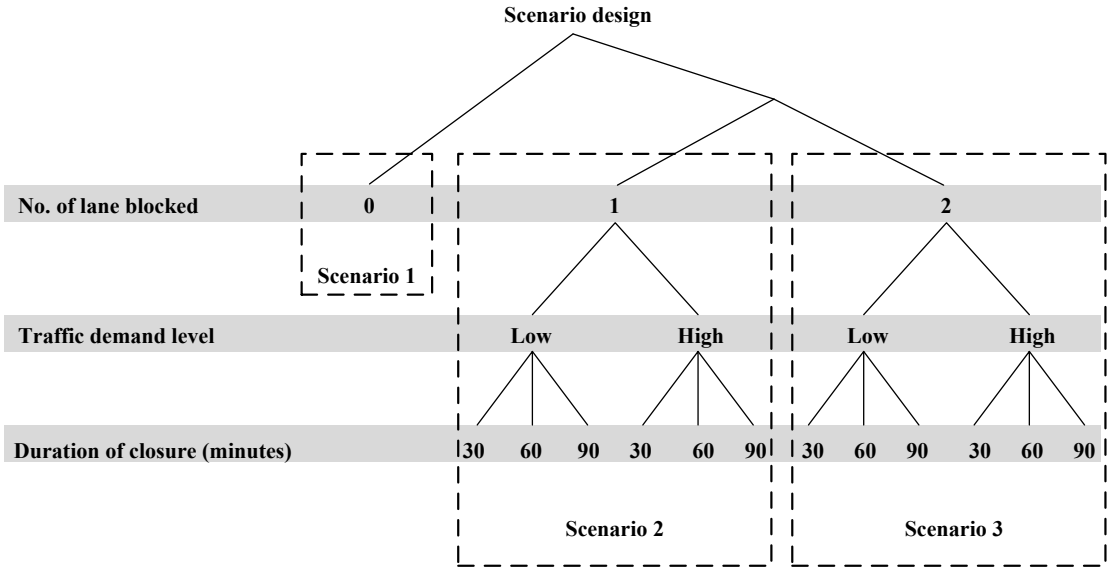


Figure 4.1: Scenario design of simulation experiments

4.3.2 Simulation model settings

4.3.2.1 Road network layout in simulation

A simulation model developed for Southampton City Council (SCC) is used in this PhD research. The road network in Southampton is a typical arterial road network in the UK. Figure 4.2 shows the corridor selected in Southampton for the generation of link travel time data. Characteristic of many similar corridors in the UK, the Southampton corridor is frequently congested, and is used as an example for study by the team at ROMANSE (Road Management System for Europe) in order to evaluate the impact of the Urban Traffic Control emulator. The selected link consists of eight sections and seven signalised junctions with a length of 1.5km, which connects the Southampton Ferry Dock to the city centre. The direction of traffic is from southeast to northwest.

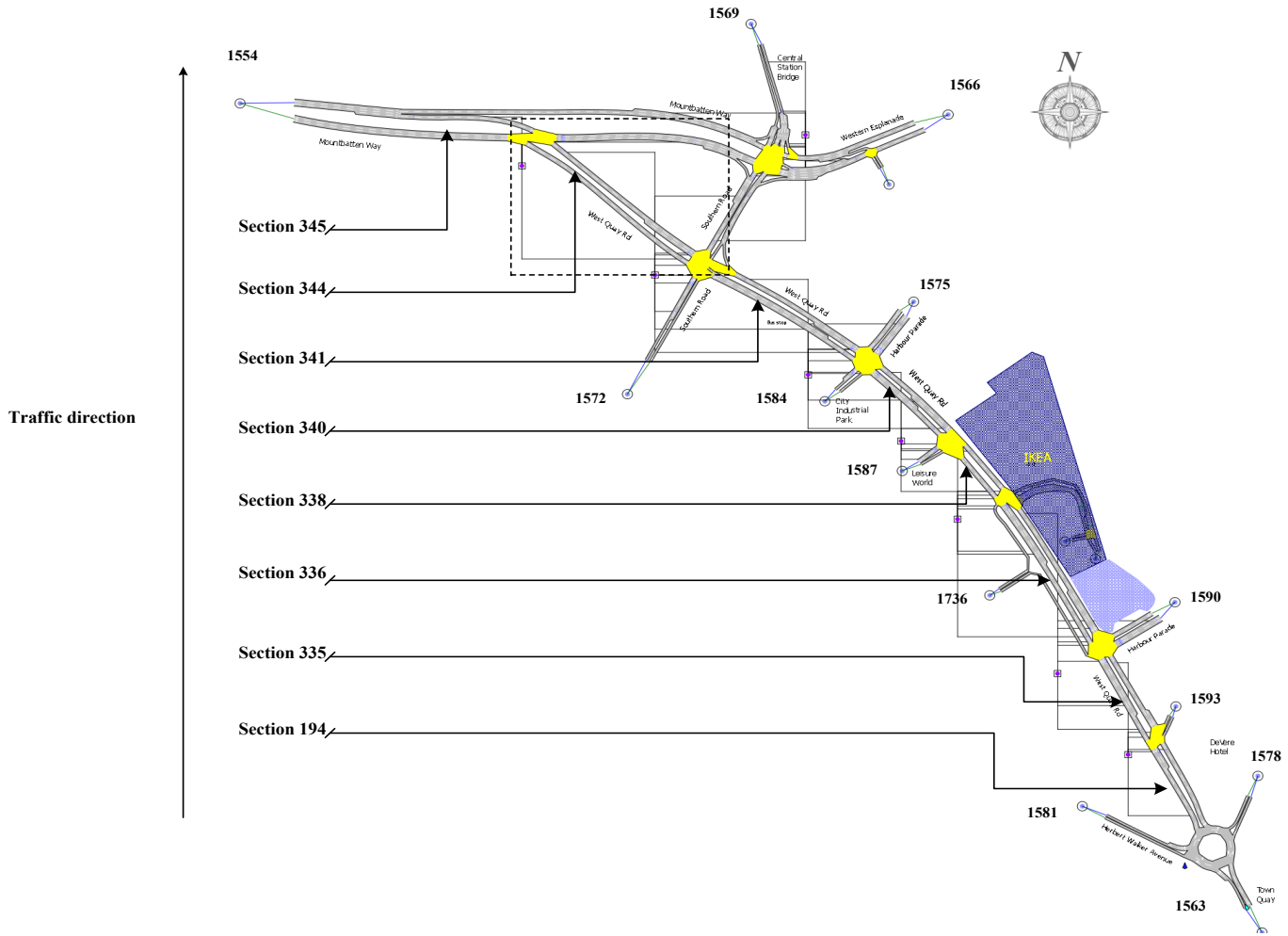


Figure 4.2: The selected link in Southampton AIMSUN network

4.3.2.2 Traffic demand

The traffic demand data used in the simulation model was defined by Origin-Destination (O-D) matrices provided by SCC. The number of trips between each O-D pair for both car and HGV (Heavy Goods Vehicle) were calculated from an existing traffic database held by SCC. A time period of 24 hours was chosen as the period for simulation modelling. The network consists of 15 zones, the ID numbers of which are also shown in Figure 4.2. Table 4.3 gives an example of the O-D matrix for the vehicle type of car used in simulation experiments between 07:00 to 07:15. The first row of the table is the ID of origin zones and the first column is the ID of destination zones.

Table 4.3: An example of the O-D matrix in the simulation model from during 07:00 to 07:15 (all values are in vheicles/hour)

O / D	1554	1563	1566	1569	1572	1575	1578	1581	1584	1587	1590	1593	1609	1612	1736
1554	0	77.0394	155.17	73.8977	19.8927	61.1237	4.4461	2.21457	0.899494	16.7905	25.8404	0.758961	11.9217	11.9217	5.32554
1563	68.546	0	0.876326	22.1633	6.30421	18.8431	6.84719	5.18113	0.574861	6.82057	21.8578	0	4.88221	4.88221	0
1566	80.8493	6.13471	0	12.6871	5.7402	2.43844	0.242797	0.094676	0.024838	0.811216	1.54611	0.040679	0.652323	0.652323	0.296784
1569	19.09	11.943	21.3394	0	16.2903	5.37022	0.520462	0.215741	0.061384	1.73449	3.18266	0.087668	1.37169	1.37169	0.625761
1572	12.4179	4.7257	0.000578	0.018472	0	1.16187	0.177285	0.069783	0.007365	0.504982	1.12728	0.029629	0.454373	0.454373	0.213887
1575	13.1435	26.1893	0.03619	1.12727	0.138562	0	0.962007	0.408303	32.0806	0.997934	5.50653	0.160935	1.90171	1.90171	1.04992
1578	1.38123	2.47145	0.004947	0.151298	0.054144	0.16374	0	0.007997	0.00446	0.060739	0.159889	0	0.035948	0.035948	0
1581	14.5905	16.1079	0.11097	2.97851	0.883756	2.48812	0.145997	0	0.07297	0.81043	1.60703	0	0.4719	0.4719	0
1584	2.98247	4.29913	0.006554	0.218037	0.022472	3.56454	0.143386	0.057635	0	0.109533	0.852933	0.023752	0.280868	0.280868	0.157591
1587	4.15308	4.30055	0.020561	0.583256	0.142005	0.081192	0.124117	0.057955	0.001549	0	0.611373	0.020389	0.084726	0.084726	0.107672
1590	18.44	49.9152	0.151004	3.96621	1.08411	2.59324	0.639319	0.769555	0.076196	0.599382	0	0.095125	0.119391	0.119391	0
1593	0.473018	1.49415	0.001274	0.041478	0.015957	0.04852	0.008367	0.013237	0.001273	0.018898	0.05667	0	0.011259	0.011259	0
1609	4.26317	3.84639	0.021125	0.604291	0.178089	0.352043	0.077447	0.048248	0.009731	0.022564	0.206454	0.012371	0	0	0.023825
1612	4.26317	3.84639	0.021125	0.604291	0.178089	0.352043	0.077447	0.048248	0.009731	0.022564	0.206454	0.012371	0	0	0.023825
1736	10.18	0.187747	0.011261	0.473095	0.344566	0.711115	0.003717	0.002365	0.018277	0.049015	0.00937	0.000589	0.039366	0.039366	0

4.3.2.3 Signal control

The AIMSUN simulation model was connected to a SIEMENS off-line Urban Traffic Control (UTC) emulator including the SCOOT (Split Cycle Offset Optimisation Technique) optimisation algorithm. The SIEMENS SCOOT UTC signal control emulator adjusts the traffic signal timings based on current traffic states. SCOOT UTC has been deployed in more than 200 towns and cities around the world, and it has been shown to reduce congestion and delays (DfT, 1999). Compared with Vehicle Actuation (i.e. non co-ordinated) signal operation, SCOOT UTC has achieved a reduction in network delay of approximately 30% in Southampton (DfT, 1999). Similarly, the reduction in delay was 14-20% compared with fixed-time plans in Surrey.

SCOOTLink, a model interface developed using CORBA (Common Object Request Broker Architecture), was used to communicate data between AIMSUN and UTC. Traffic flow and occupancy data from loop detectors in the AIMSUN simulation model are sent to UTC SCOOT via SCOOTLink. SCOOT processes this traffic data, determines the optimal traffic signal timings and sends the results back to AIMSUN using SCOOTLink. The traffic signal timings include green times, inter-greens and offsets. Both systems work in an asynchronous way. Traffic data from loop detectors are passed to SCOOT every second as the AIMSUN model responds to changing signal timings. This advanced control interface is shown in Figure 4.3. More information about the SIEMENS SCOOT UTC signal control emulator can be found in Wylie (2012).

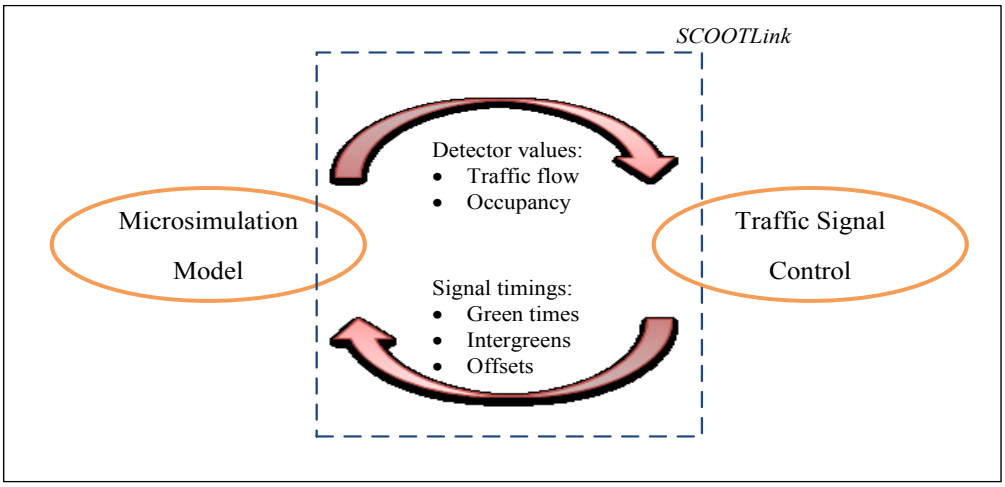


Figure 4.3: A representation of the interaction between traffic simulation and signal control

4.3.2.4 Model calibration and validation

The credibility of a simulation model depends on its ability to replicate field conditions accurately. A large number of parameters in a simulation model may influence the performance of the model. Hence, calibration and validation procedures are required to simulate the real-world traffic networks by adjusting model parameters through trial-and-error. In a data-rich situation, two independent datasets are generally used in this procedure. The first dataset is used to calibrate the model parameters to represent local traffic conditions. The second is for validating the model by comparing the outputs generated from the calibrated model and the field observed data. Many publications have discussed the general requirement for a calibration procedure with the goodness-of-fit tests for model validation, for example Hourdakis *et al.* (2003); Jha *et al.* (2004) and Toledo & Koutsopoulos (2004).

The simulation parameters to be calibrated in AIMSUN can be classified into three main categories: global parameters, local parameters and vehicle attributes (TSS, 2004). Global parameters, including a driver's reaction time, response time at stop, queuing-up and queuing-leaving speeds, are used for all vehicles and affect the

performance of the entire simulation network. Local section parameters, such as section speed limit, lane speed limit, turning speed and visibility distance at junctions, affect only a specific section of the network regardless of vehicle types. Vehicle parameters, such as maximum desired speed, maximum acceleration, normal deceleration and maximum deceleration, influence all vehicles of a determinate type in the simulation network.

As summarised by Hourdakis *et al.* (2003), an ideal method for model calibration has three stages. The first stage is volume-based calibration, the objective of which is to obtain simulated traffic volumes which are as close as possible to the real measured volumes. The global parameters and vehicle characteristics are modified in this stage. In the second stage, speed-based calibration, most local parameters and global parameters need further modification to accurately simulate real-world traffic networks. The third stage, objective-based calibration (e.g. queue lengths), is an optional stage that depends on the specific objective of the simulation model.

The main purpose of model calibration and validation is to ensure that the simulated network replicates the real traffic network as closely as possible by comparing the simulated outputs with measured data. There are various approaches to simulation validation available in the literature. These include statistics (such as correlation efficient, root mean squared percentage error, Theil's inequality coefficient, error mean relative positive and error mean relative negative, as summarised in Vilarinho & Tavares (2012)), other statistical analyses (Student's-t test and hypothesis test, e.g. Barcelo & Casas (2004)) and graphical representation (band comparison and scatter-grams, e.g. Haas (2001); Barcelo & Casas (2004)).

The values of some important parameters used in AIMSUN are provided in Table 4.4.

Table 4.4: Important parameters in AIMSUN

Parameter Name	Value	Unit
Driver's reaction time	0.75	sec
Reaction time at stop	1.35	sec
Reaction time at traffic light	1.35	sec
Car		
Length	2.5-5.16	metre
Width	1.4-2.08	metre
Maximum desired speed	95-160	km/h
Maximum acceleration	2.8	m/s^2
Normal deceleration	4-6	m/s^2
Maximum deceleration	8-11	m/s^2
Speed acceptance[2]	1-1.4	
Minimum distance between vehicles	1-2	metre
Give way time	10-50	sec
Guidance acceptance	100	%
HGV		
Length	12	metre
Width	2.3	metre
Maximum desired speed	80-100	km/h
Maximum acceleration	1.4-1.6	m/s^2
Normal deceleration	3.5	m/s^2
Maximum deceleration	8	m/s^2
Speed acceptance[3]	0.9-1.2	
Minimum distance between vehicles	1	metre
Give way time	5-60	sec
Guidance acceptance[4]	100	%
Section 338		
Distance on ramp[5]	5	sec
Visibility distance	25	metre
Yellow box speed	10	km/h
Maximum speed	50	km/h

[2] A parameter measures the driver's degree of accomplishment of the speed limits on the section.
[3] This parameter can be interpreted as the 'level of goodness' of the drivers or the degree of acceptance of speed limits.
[4] This parameter gives the level of compliance of this vehicle type with the guidance indications, such as information given through Variable Messages Signs or particular Vehicle Guidance Systems.
[5] The distance on ramp in AIMSUN were set as a time and internally converted to a distance using the desired speed of each vehicle.

The main purpose of model calibration and validation is described above. There are some 'features' of the simulation model, however, such as vehicle crashes, disappearing vehicles and vehicles stopped in appropriately on the link which also . need to be examined during the procedure of model calibration and validation. In this simulation experiment, the following actions were taken to check the 'features' mentioned above:

- To visually monitor links and junctions when a model is running on the frontend;
- To monitor the dialogue of *Simulating Replication* written by AIMSUN that records the number of 'lost' vehicles when a model is running; and
- To check the output file that records the number of input and output vehicles into the road network.

These non-standard 'features' should be checked during every single run of the model; in practice, however, this process is time consuming. Hence, two simulated days were randomly selected. One was under normal traffic conditions; the other was under heavily congested condition. Under normal traffic condition, the 'features' of vehicle crashes and inappropriately stopped vehicles did not happen when monitoring the running simulation model on the frontend. In this simulation experiment, 68,287 vehicles entered the road network; 15 vehicles stopped inside the network when the simulation was finished and 68,098 vehicles exited the network, resulting in 174 'lost' vehicles. The percentage of 'lost' vehicle is therefore 0.25%. Under heavily congested conditions, the problems of vehicle crashes and inappropriately stopped vehicles did not happen when monitoring the running model. During the simulation, 68,428 vehicles entered the network; 16 vehicles stopped inside the network when the

simulation was finished and 68,254 vehicles exited the network, resulting in 158 'lost' vehicles. The percentage of 'lost' vehicles is therefore 0.23%.

4.3.3 Outputs of simulation

Outputs used in the simulation experiments are link travel time data. Krishnan (2008) summarised three commonly used definitions of link travel time. Link travel time data can be generated using 1) vehicles that enter the link during a given period; 2) vehicles that exit the link during a given period and 3) vehicles that enter and exit the link during a given time period. Common traffic simulators such as AIMSUN and VISSIM and ITS system deployed in the real-world use the second definition, which is able to calculate link travel time in real-time without any delays and does not account for vehicles that do not finish their journey during a given time period.

Thus, in AIMSUN data about only those vehicles that exit the selected link are used to generate the link travel time, given in Equation 4.1 (TSS, 2010). This equation is also used in the real world to define link travel time.

$$TT = \frac{1}{N_{link}} \sum_{i=1}^{N_{link}} TEX_i - TEN_i \quad (4.1)$$

where

- TT = link travel time during the given period
- N_{link} = number of vehicles that exit the selected link during the given time period
- TEX_i = Exit time of vehicle (i) from the selected link
- TEN_i = Entry time of vehicle (i) to the selected link

Under highly congested conditions, vehicles entering the network may not exit the selected link during a given period, such as 5 minutes. In such circumstances the link travel time will be calculated when these vehicles do exit the selected link and thus no vehicles in the network will be ignored even if they do not finish their journey during a single time interval.

When all lanes are blocked during the given interval because of an abnormal traffic event, such as an incident occurring on the link, none of the vehicles will exit the link. The output of the above equation is then set to a negative value '-1' in AIMSUN.

The simulation was configured to run for a 24-hour period. Simulation runs were started from 00:00 and ended at 23:59. A simulation always starts with an empty network. Thus a warm-up period is required to get realistic traffic data. The AIMSUN simulation model has a standard 15-minute warm-up period from 00:00 to 00:15. The output data are aggregated at 5-minute interval. Random seed numbers were used in the simulation of normal traffic conditions in order to model different weekday traffic states in the training dataset. This simulation experiment did not, however, consider the variability in traffic demand. Figure 4.4 shows the plots of averaged, maximum and minimum traffic profiles in the training dataset. These plots indicate the day-to-day variability in traffic patterns using random seed numbers, which simulate the variability in traffic demands caused by the randomness in daily travel activities.

Table 4.5 summarises the number of days used in experiments. Although simulation runs were set to 24 hours, only data from 05:00 to 22:00 were used, since traffic engineers are more interested in traffic states during this period. Three scenarios in this experiment use the same training dataset, which includes 40

individual days to avoid inaccurate prediction due to insufficient training data. Five testing days are used in Scenario 1 to simulate weekday traffic patterns in one week. The number of testing days in Scenarios 2 and 3 depends on the combinations of possible attributes used to describe lane closure, such as number of lanes closed, traffic demand level during closure and closure duration. In this simulation experiment, there are six cases under each scenario to simulate abnormal events. Figure 4.5(a) is a time-series example of testing days in Scenario 1 and Figure 4.5(b) is an example plot in Scenario 2.

Table 4.5: Traffic data in scenarios used for framework evaluation

	Scenario 1	**Scenario 2**	**Scenario 3**
Training days	40	40	40
Testing days	5	6	6

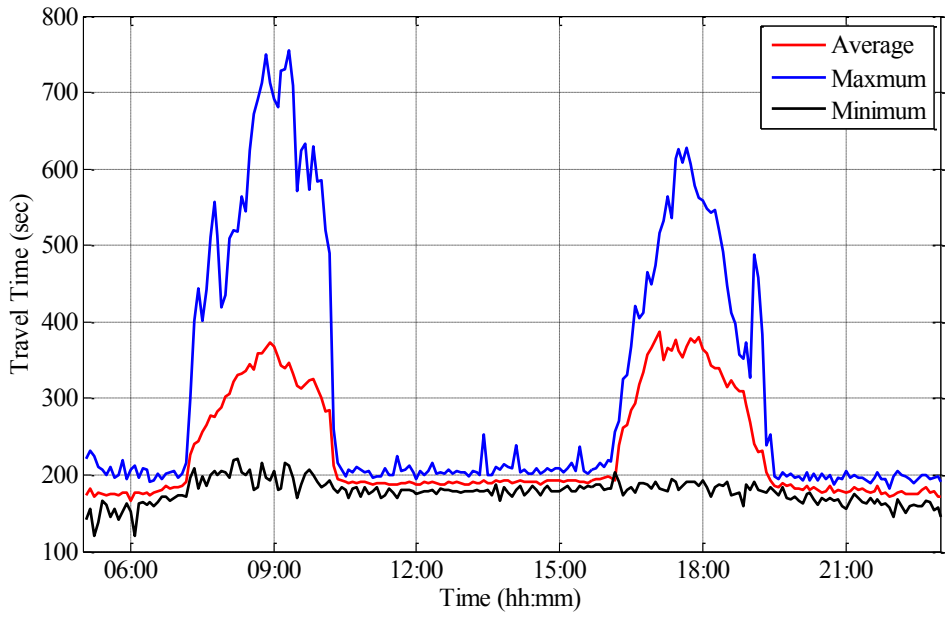
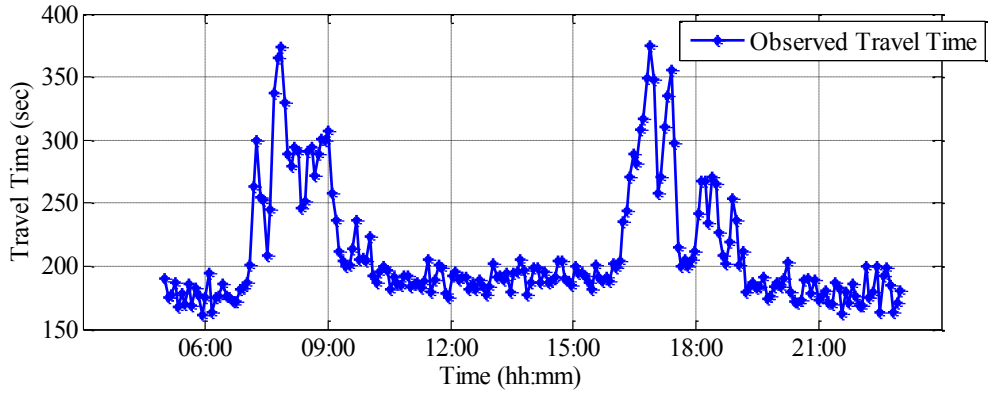
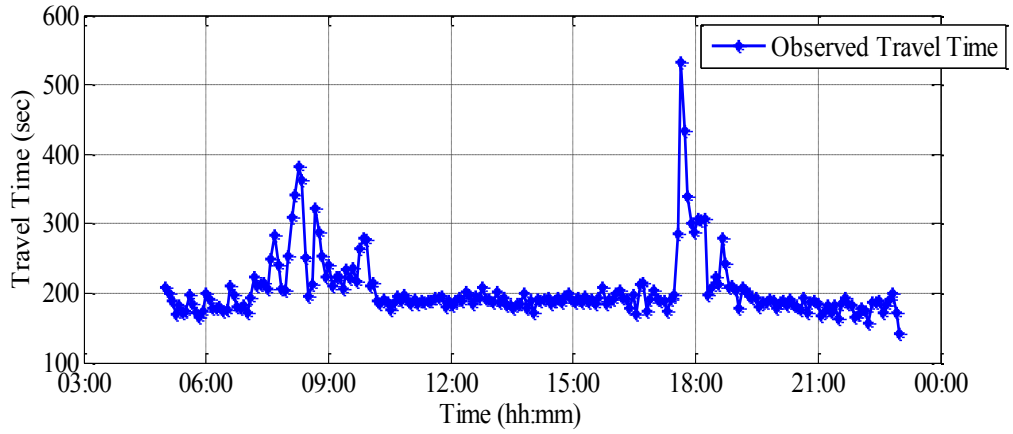


Figure 4.4: Plots of averaged, maximum and minimum values of traffic profiles in the training dataset



(a)



(b)

Figure 4.5: (a) An example of a travel time profile during 05:00 – 22:00 under normal traffic conditions and (b) An example of a travel time profile during 05:00 – 22:00 under abnormal traffic conditions in Scenario 2

4.4 Prediction accuracy under normal traffic conditions - Scenario 1

The prediction accuracy of the proposed prediction frameworks under normal traffic conditions is discussed in this section. The integrated 5-min link travel time data were used to predict one-step ahead travel time. The values of MPE, MAPE and RMSE

were calculated throughout the entire period of prediction, which is from 05:00 to 22:00.

Table 4.6 compares the prediction accuracy for five different machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks for each of the five testing days under normal traffic conditions and Table 4.7 gives the average prediction accuracy of the five normal traffic days. Figure 4.6 and Figure 4.7 depict the RMSE and MAPE scores of these frameworks.

It is clear that, under normal traffic conditions, prediction accuracy in terms of MPE, MAPE and RMSE increases when the 2-stage and 3-stage frameworks are applied, regardless of the machine learning method used. For example, the value of MPE by kNN using the 1-stage traffic prediction framework is 0.83%, with the prediction bias being reduced to 0.68% using the 2-stage framework and 0.30% using the 3-stage framework. The MAPE by kNN using the 1-stage traffic prediction framework is 6.47% while MAPE values of 4.69% and 4.19% are yielded when the 2-stage and 3-stage frameworks are used. This equates an improvement of 27.5% and 35.2%, respectively. The average MAPE metric of five machine learning methods shows an improvement from 6.56% using the 1-stage framework to 4.08% using the 3-stage framework, a 37.8% increase in accuracy. Similarly, the RMSE metric improves from 20.12 seconds for the 1-stage framework to 13.09 seconds using the 3-stage framework, a 34.9% improvement.

The MAPE is quite similar across the five machine learning methods, although the NN based method using the 3-stage prediction framework has the best overall prediction accuracy among all methods under normal traffic conditions, with an average MPE of -0.07%, MAPE of 3.31% and RMSE of 11.27 seconds.

Table 4.6: Prediction accuracy of link travel time using three different frameworks with five machine learning methods under normal traffic conditions in Scenario 1

	MPE (%)			MAPE (%)			RMSE (seconds)		
Testing Day 1									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.69	0.55	0.30	5.90	4.23	3.73	15.36	11.50	9.96
GM	0.50	0.25	0.14	7.69	4.76	4.37	20.30	12.95	11.75
NN	0.73	-0.11	-0.04	5.28	3.18	2.30	13.94	8.32	7.75
RF	0.59	0.42	0.24	5.21	3.68	3.35	14.37	10.36	9.19
SVR	0.56	0.47	0.16	5.39	3.69	3.18	14.32	11.84	10.32
Testing Day 2									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.65	0.58	0.32	6.29	4.28	4.19	18.59	13.54	13.11
GM	0.60	0.32	0.25	7.72	5.09	4.99	24.51	15.75	16.90
NN	0.72	-0.09	-0.05	5.40	3.88	3.98	16.16	11.68	11.99
RF	0.90	0.71	0.36	5.33	4.38	4.48	15.93	13.49	14.71
SVR	0.06	0.24	0.20	6.40	4.31	4.19	17.26	14.25	13.74
Testing Day 3									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	1.08	0.74	0.32	6.52	5.14	4.37	20.77	17.64	14.47
GM	0.58	0.33	0.06	8.89	5.95	5.45	28.03	20.89	18.07
NN	0.65	-0.20	-0.14	6.15	3.57	3.34	20.73	14.05	13.03
RF	0.57	0.65	0.23	5.96	4.39	3.89	20.45	15.40	13.15
SVR	0.76	0.65	0.36	6.28	4.34	3.69	21.06	14.61	13.17
Testing Day 4									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.94	0.79	0.36	7.20	5.11	4.46	21.21	15.37	13.24
GM	0.65	0.36	0.17	9.37	5.85	5.30	27.29	17.38	15.73
NN	0.73	-0.10	-0.03	6.38	3.80	3.72	19.50	13.22	10.79
RF	0.84	0.41	0.21	6.15	4.23	3.98	19.62	12.67	11.45
SVR	0.79	0.53	0.32	7.35	5.51	4.16	21.14	14.66	13.08
Testing Day 5									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.81	0.76	0.22	6.42	4.69	4.21	21.18	17.54	14.58
GM	0.60	0.37	0.08	8.48	5.86	5.29	28.30	21.39	18.48
NN	0.55	-0.13	-0.08	5.85	3.42	3.21	20.48	16.71	12.79
RF	0.70	0.49	0.19	6.07	4.12	3.37	21.24	14.91	12.69
SVR	0.89	0.61	0.32	6.96	5.80	4.84	21.24	17.90	13.23

Table 4.7: Averaged prediction accuracy of link travel time using three different frameworks with five machine learning methods under normal traffic conditions

	MPE (%)			MAPE (%)			RMSE (seconds)		
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.83	0.68	0.30	6.47	4.69	4.19	19.42	15.12	13.07
GM	0.59	0.33	0.14	8.43	5.50	5.08	25.69	17.67	16.19
NN	0.68	-0.13	-0.07	5.81	3.57	3.31	18.16	12.40	11.27
RF	0.72	0.536	0.25	5.74	4.16	3.81	18.32	13.37	12.24
SVR	0.61	0.50	0.27	6.48	4.73	4.01	19.00	14.65	12.71
Mean	0.69	0.38	0.18	6.56	4.53	4.08	20.12	14.24	13.09

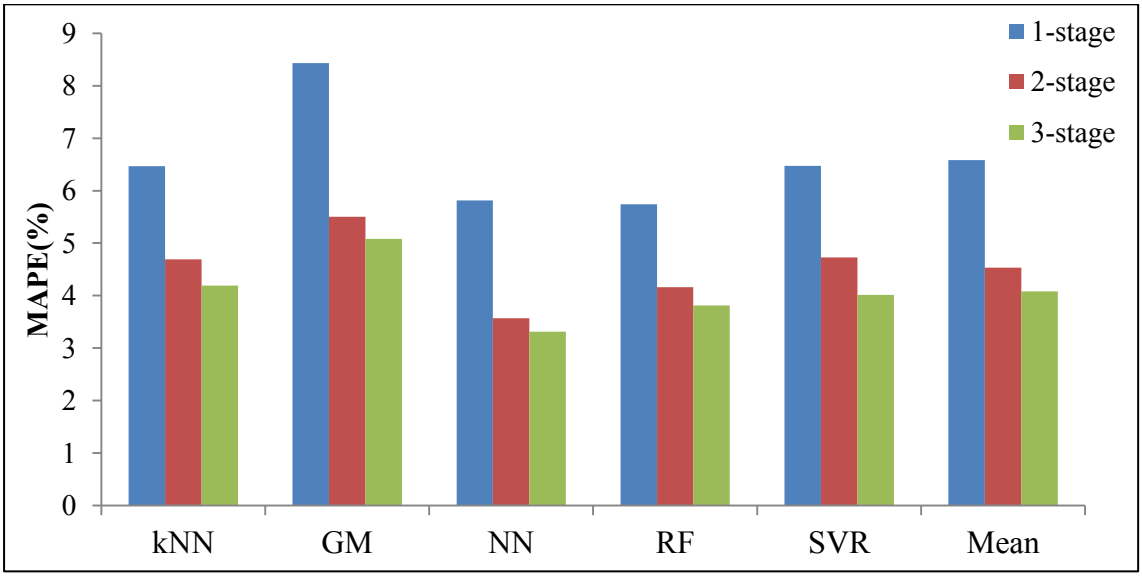


Figure 4.6: MAPE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks under normal traffic conditions in Scenario 1

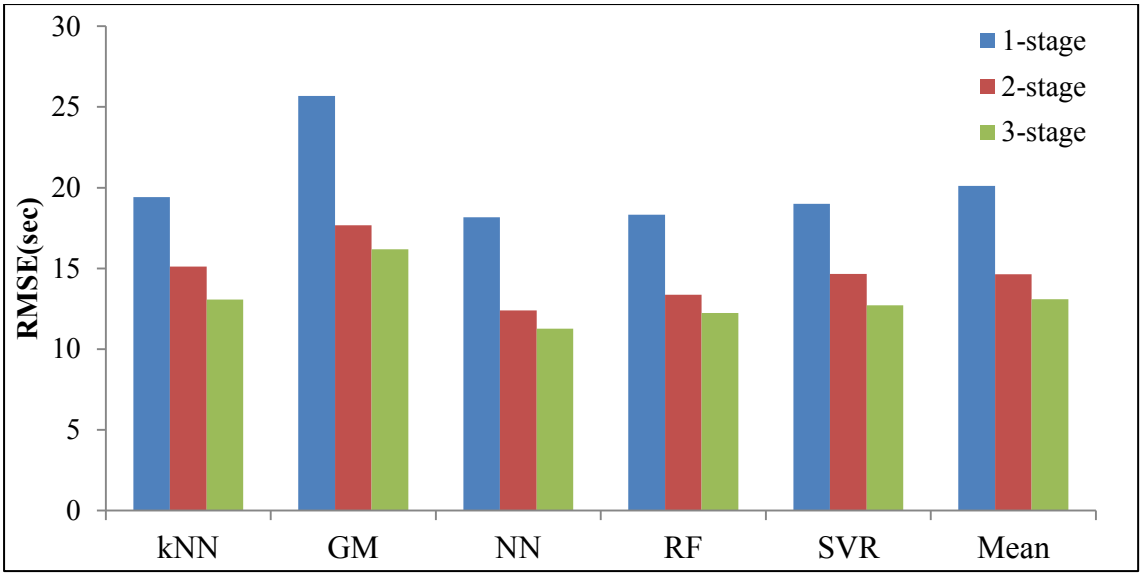


Figure 4.7: RMSE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks under normal traffic conditions in Scenario 1

4.5 Prediction accuracy under abnormal traffic conditions

Most links in the simulation network comprise a two-lane carriageway; the number of lanes increases from two to three only in the proximity of merges and roundabout. In order to prevent any other effects rather the abnormal event itself influencing the results, careful consideration needs to be given the selection of the location for the lane closure. Given the nature of the corridor in Southampton used by the simulation model in this research, when two lanes are blocked, vehicles must wait rather than re-route. For these reasons, the location of the lane closure was selected to be far from the entrance to the link and near to the exit so as to avoid the queue extending beyond the entrance of the simulation model. The location of the lane closure is shown in Figure 4.8 (area A).

Under abnormal traffic conditions, there are two scenarios based on the different levels of lane closure. The following subsections discuss the prediction accuracy

using three traffic prediction frameworks under abnormal traffic periods. The MPE, MAPE and RMSE values in Scenarios 2 and 3 were calculated based on the period starting 30 minutes before the occurrence of the abnormal event and finishing around 30 minutes after the clearance of the event. The assumption is that the traffic state might recover to normal conditions within 30 minutes.

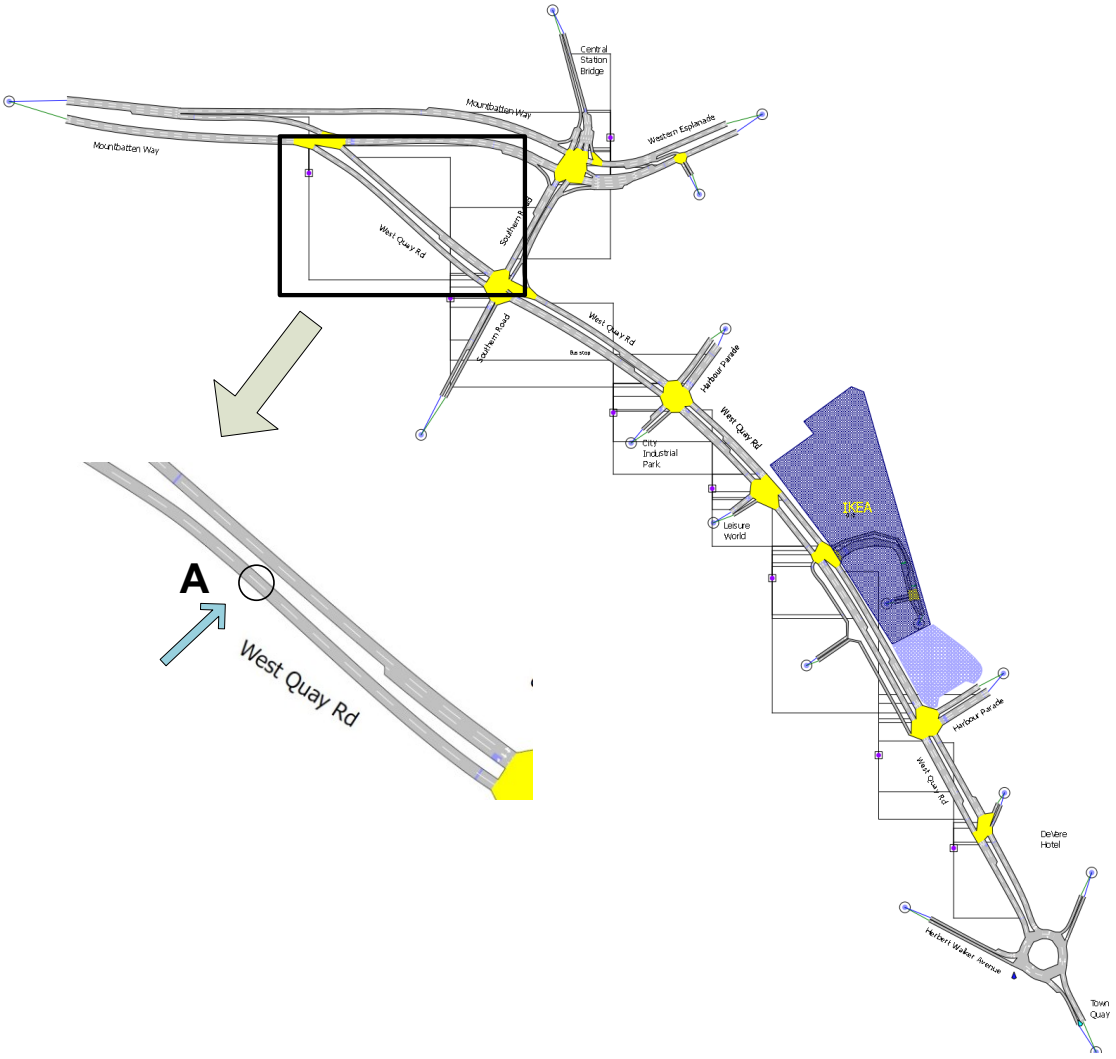


Figure 4.8: Location of the lane closure in simulation

4.5.1 Scenario 2: One-lane closure in simulation

In scenario 2, six experiments were undertaken by changing the start time (off-peak and peak period) and the duration (30, 60 and 90 minutes) of lane closure. Figure 4.9 shows the blocked area in Scenario 2. In this scenario, when one lane was blocked during an off-peak period, traffic demand is at a low level. Vehicles may use the other lane to pass the selected link. Traffic profiles do not change a lot compared with historical traffic patterns. When two lanes were blocked during a peak period, however, traffic demand is at a high level. Non-recurrent traffic congestion may occur under this traffic conditions and the lane closure may cause a sudden change in the traffic profiles of link travel time. As a result, short-term traffic prediction is more challenging under these abnormal traffic conditions.

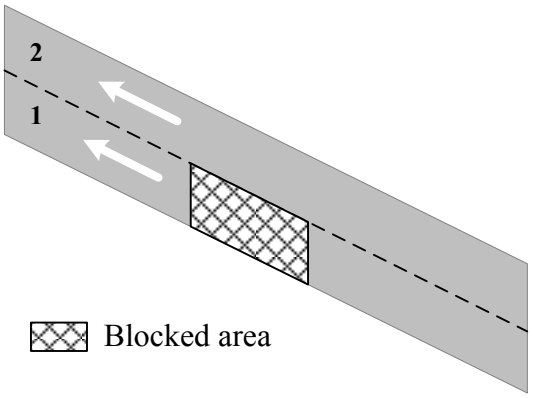


Figure 4.9: One-lane closure in area A

4.5.1.1 One lane closure during the off-peak period

Three experiments were tested by varying lane closure duration during the off-peak period. In the experiments described in this subsection the closure of a single lane started from 12:00 and one-step ahead link travel time was predicted.

Table 4.8 gives the prediction accuracy for the three prediction frameworks with five machine learning methods during the closure period. Figure 4.10 and Figure 4.11

illustrate the decrease of RMSE and MAPE scores using different prediction frameworks. As expected, the prediction accuracy is significantly improved by using the 2-stage and the 3-stage frameworks. The average MPE, MAPE and RMSE of the five machine learning methods when the 1-stage framework was used is -0.18%, 4.04% and 10.03 seconds respectively. With the introduction of data smoothing and error feedback, the average MPE, MAPE and RMSE when the 1-stage framework was used is 0.10%, 2.63% and 6.58 seconds respectively. The prediction accuracy was again quite similar across the five machine learning methods. The NN-base method with the 3-stage prediction framework has best prediction accuracy.

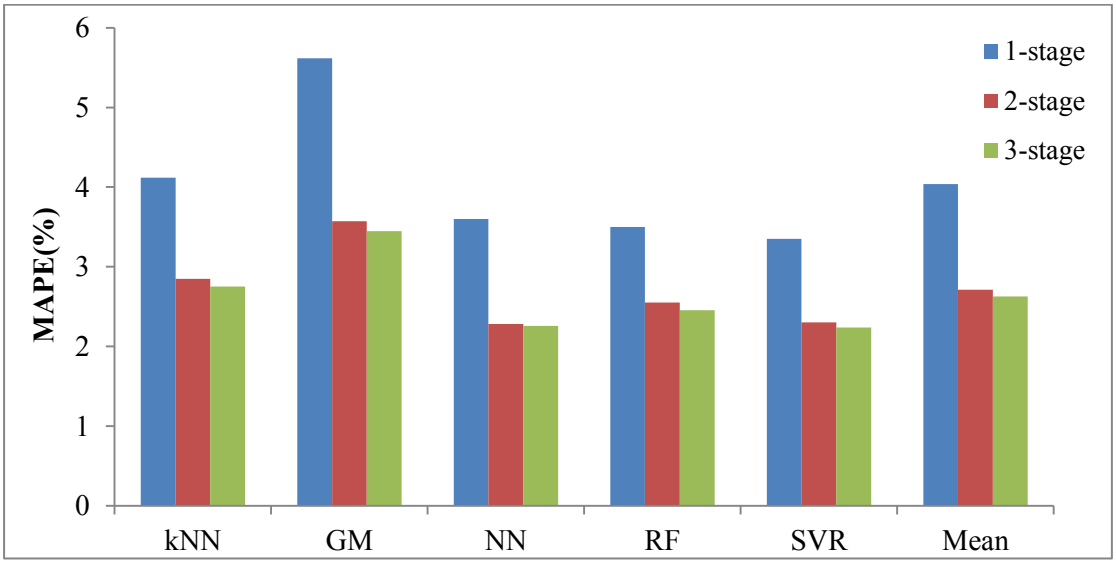


Figure 4.10: MAPE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the off-peak period

Table 4.8: Comparison of prediction accuracy when one lane blocked during the off-peak period

	MPE (%)			MAPE (%)			RMSE (seconds)		
Duration of closure: 30 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-0.06	-0.41	-0.31	4.10	2.68	2.66	10.45	6.68	6.61
GM	-0.04	-0.19	-0.15	5.35	3.16	3.11	12.87	8.19	7.72
NN	-0.58	-0.55	-0.01	3.44	2.19	2.18	8.87	5.63	5.59
RF	-0.78	-0.34	-0.22	3.09	2.26	2.25	7.89	5.70	5.69
SVR	-0.27	-0.21	-0.17	3.27	2.23	2.21	8.11	5.54	5.42
Duration of closure: 60 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.20	0.08	0.17	3.85	2.73	2.58	9.16	6.61	6.34
GM	0.76	0.17	0.49	4.68	3.62	3.41	11.04	8.35	7.96
NN	-0.16	0.13	0.21	3.50	2.35	2.31	8.39	5.99	5.96
RF	-0.32	0.19	0.27	3.59	2.28	2.23	8.36	5.57	5.69
SVR	-1.21	0.68	0.36	3.77	2.58	2.49	8.83	6.42	6.09
Duration of closure: 90 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-0.17	0.04	0.06	4.40	3.13	3.01	11.12	7.84	7.51
GM	0.55	0.32	0.31	6.82	3.93	3.82	16.03	9.66	9.31
NN	-0.28	-0.06	-0.01	3.86	2.31	2.28	10.08	6.00	5.75
RF	0.22	0.15	0.14	3.82	3.11	2.88	10.22	7.61	7.10
SVR	-0.57	-0.23	0.31	3.01	2.09	2.01	8.98	6.33	5.97
Average of three above cases									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-0.01	-0.10	-0.03	4.12	2.85	2.75	10.24	7.04	6.82
GM	0.42	0.10	0.22	5.62	3.57	3.45	13.31	8.73	8.33
NN	-0.34	-0.16	0.06	3.60	2.28	2.26	9.11	5.87	5.77
RF	-0.29	-0.03	0.06	3.50	2.55	2.45	8.82	6.29	6.16
SVR	-0.68	0.08	0.17	3.35	2.30	2.24	8.64	6.10	5.83
Mean	-0.18	-0.02	0.10	4.04	2.71	2.63	10.03	6.81	6.58

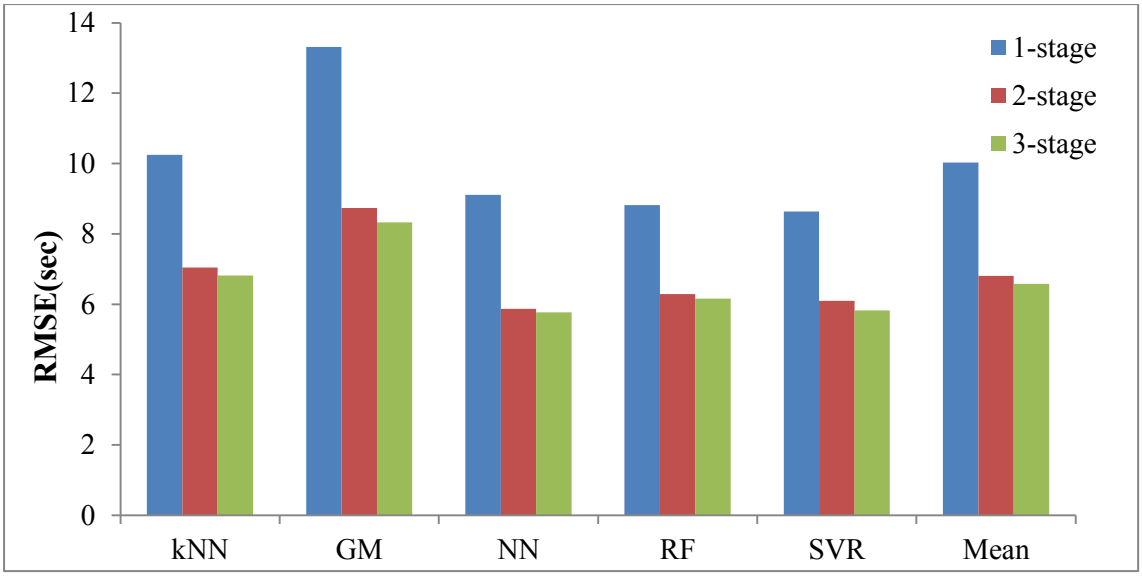


Figure 4.11: RMSE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the off-peak period

4.5.1.2 One lane closure during the peak period

Three experiments were conducted to evaluate the proposed framework when abnormal events happened during the peak period. The lane closure started from 16:30. Three proposed prediction frameworks with five different machine learning methods were used to predict one-step ahead link travel time.

Table 4.9 shows the comparison of prediction accuracy using the three prediction frameworks and the five machine learning methods during this closure period. Figure 4.12 and Figure 4.13 show the change of RMSE and MAPE scores using the three different prediction frameworks. It can be seen that here the 2-stage prediction framework with SSA improves the prediction accuracy in terms of MAPE and RMSE when compared with the prediction using the 1-stage framework for the kNN, GM, NN, RF and SVR methods. The 3-stage prediction framework with both SSA and error feedback has the best prediction accuracy of the three frameworks. Of the five

different machine learning methods, the kNN based method with the 3-stage prediction framework has the best prediction accuracy. Using the kNN based method, the average MAPE metric reduces from 11.66% with the 1-stage framework to 6.56% with the 2-stage SSA framework, and even to 6.02% with the 3-stage framework with the introduction of SSA and error feedback mechanisms. Similarly, the average RMSE metric using NN reduces from 55.63 seconds with the 1-stage framework to 23.79 seconds with the 3-stage framework.

It can also be seen that with the increase of the closure duration, the prediction accuracy decreases. This is not surprising since an increase in the closure duration during a peak period may cause traffic patterns and congestion to change significantly.

Table 4.9: Comparison of prediction accuracy when one lane blocked during the peak period

	MPE (%)			MAPE (%)			RMSE (seconds)		
Duration of closure: 30 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.75	0.87	0.71	7.89	6.68	5.91	27.16	23.39	19.99
GM	1.64	1.03	0.93	10.03	7.23	6.66	32.53	22.91	21.04
NN	-1.24	-0.95	-0.57	8.36	5.12	4.94	28.86	21.99	18.10
RF	0.35	0.34	0.45	8.81	5.70	5.15	29.63	21.90	18.32
SVR	1.32	0.75	0.48	8.23	5.14	4.96	30.29	22.75	19.87
Duration of closure: 60 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-0.09	0.31	0.72	13.37	7.89	6.91	77.40	43.54	38.51
GM	2.78	1.62	1.45	16.94	10.48	9.40	87.33	58.39	46.93
NN	-1.54	1.31	0.27	12.06	9.00	7.84	70.37	56.15	46.33
RF	-1.39	-1.60	0.67	13.21	9.25	7.81	75.11	54.84	42.31
SVR	2.14	0.93	0.89	14.91	9.90	8.82	75.15	55.64	43.12
Duration of closure: 90 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	1.28	-0.65	0.12	13.66	7.85	7.82	59.28	23.00	21.37
GM	3.09	2.36	1.69	19.02	15.38	11.98	95.03	66.28	48.83
NN	-0.28	1.34	-1.08	11.21	10.51	9.97	64.63	53.58	41.43
RF	1.01	1.13	0.22	12.03	11.14	10.80	60.52	50.84	42.75
SVR	2.43	1.06	1.15	12.16	11.49	10.77	65.29	54.53	46.61
Average of three above cases									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.46	-0.24	0.34	11.66	6.56	6.02	55.63	26.63	23.79
GM	2.50	1.67	1.22	15.33	11.03	9.12	71.63	49.19	37.75
NN	-1.02	0.57	-0.16	10.54	8.21	7.39	54.62	43.91	33.43
RF	-0.01	-0.04	0.42	11.35	8.70	7.62	55.09	42.53	32.75
SVR	1.96	0.91	0.84	11.77	8.84	8.18	56.91	44.31	36.53
Mean	0.78	0.57	0.53	12.13	8.67	7.67	58.77	41.31	32.85

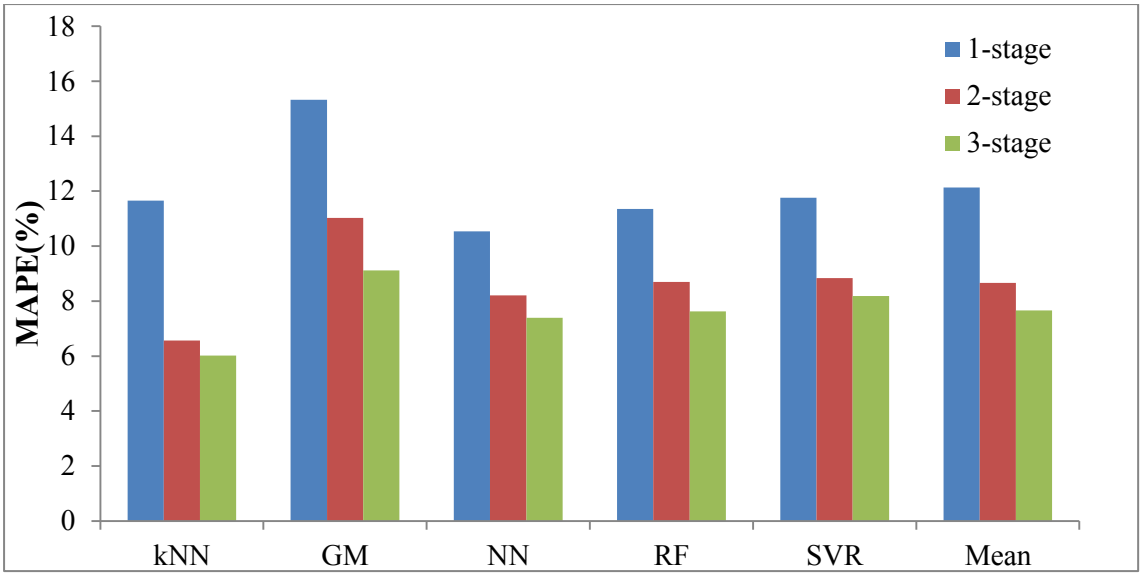


Figure 4.12: MAPE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the peak period

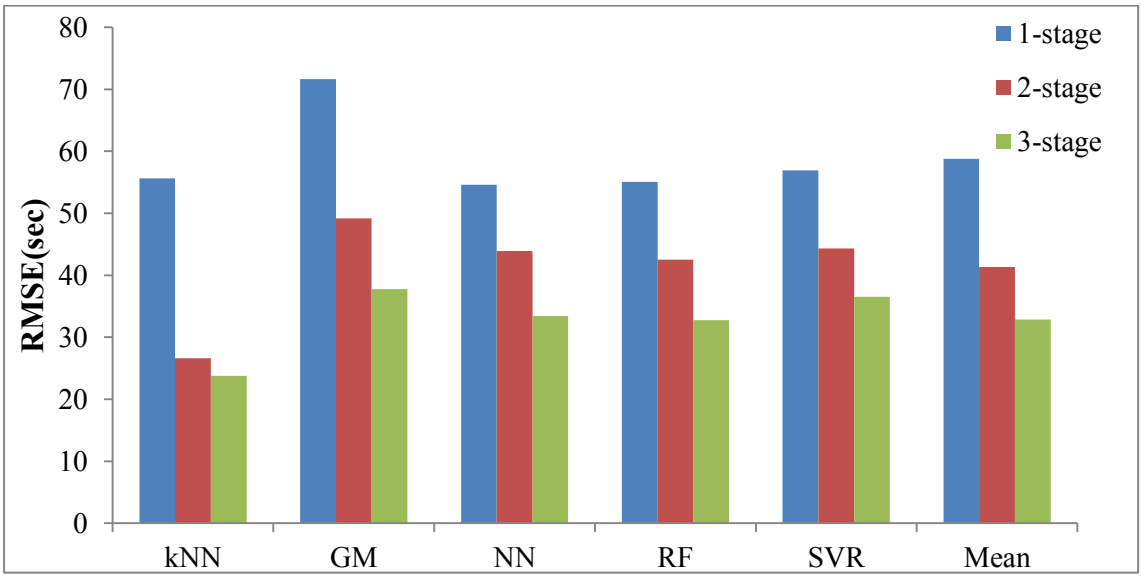


Figure 4.13: RMSE for five machine learning methods using the 1-stage, 2-stage and 3-stage traffic prediction frameworks when one lane was blocked during the peak period

4.5.2 Scenario 3: Two-lane closure in simulation

In Scenario 3, each of the two lanes of the selected section was blocked at a given location. Figure 4.14 shows the blocked area in this scenario. Six experiments were produced by varying the start time of the lane closure (off-peak and peak period) and the duration of the blockage (30, 60 and 90 minutes).

In Scenario 3, during the period in which both the lanes are blocked, all the vehicles in the selected section cannot pass until the abnormal event is cleared. The profile of link travel time may therefore quickly change. After the lane closure is cleared, the waiting queue dissipates and the traffic condition recovers to its normal state. An extremely large value for link travel time, including the waiting time to pass the blocked area, can be found in the traffic profile.

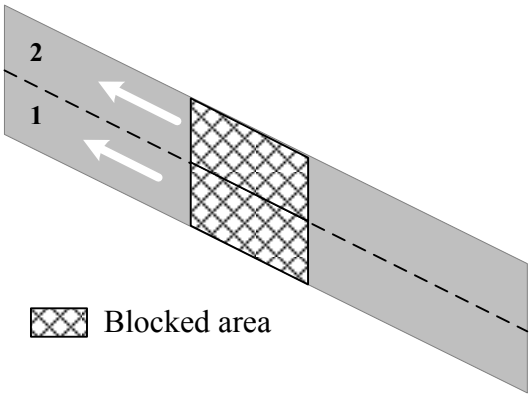


Figure 4.14: Two-lane closure in area A

4.5.2.1 Two-lane closure during the off-peak period

All the experiments were run with different lane closure durations starting from 12:00. Table 4.10 gives the comparison of one-step ahead prediction accuracy using the three prediction frameworks with five machine learning methods during the specific lane closure period. The prediction results show that the difference between the observed

data and predicted data is significant regardless of the prediction frameworks and machine learning methods used.

Table 4.10: Comparison of prediction accuracy when two lanes were blocked during the off-peak period

	MPE (%)			MAPE (%)			RMSE (seconds)		
Duration of closure: 30 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-11.03	-11.18	-10.48	19.92	19.43	17.90	347.04	136.40	120.62
GM	-7.29	-11.34	-7.83	24.10	24.71	20.29	392.72	391.45	362.12
NN	-8.22	-9.85	-3.55	20.75	18.87	17.99	315.20	186.93	167.43
RF	-6.87	-9.30	-5.51	22.05	20.49	17.50	338.78	288.95	251.16
SVR	-18.04	-14.26	-9.37	32.89	28.69	22.12	340.26	267.98	287.64
Duration of closure: 60 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-11.76	-17.96	-17.97	31.84	29.42	28.86	649.11	352.70	329.27
GM	-12.80	-26.00	-15.67	30.08	33.47	30.48	767.40	498.59	597.35
NN	-6.34	-9.71	-3.87	21.27	23.66	22.91	689.53	432.10	397.81
RF	-2.07	-3.93	-3.96	20.50	22.32	21.09	740.80	689.56	570.33
SVR	-4.46	-6.05	-8.27	20.71	23.98	22.83	739.22	456.52	422.87
Duration of closure: 90 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-18.71	-17.47	-18.39	31.97	31.18	31.10	740.94	530.56	502.74
GM	-12.26	-29.49	-18.16	27.55	36.94	34.08	1050.79	703.94	901.03
NN	-3.51	-8.53	-4.32	18.68	22.58	21.78	1057.95	793.47	691.42
RF	-5.63	-1.82	-4.58	18.24	26.49	25.99	1302.05	1243.63	870.22
SVR	-3.40	-5.02	-6.64	18.46	27.45	25.72	1109.33	782.46	861.99

4.5.2.2 Two-lane closure during the peak period

The two-lane closure started when the traffic state became congested at 16:30. Three tests were carried out with closure durations of 30, 60 and 90 minutes. Table 4.11 gives the comparison of one-step ahead prediction accuracy during the specific lane

closure periods. Similar to the results in 4.5.2.1, the prediction errors are large no matter which frameworks and machine learning methods are used. The prediction results in this subsection are clearly worse than those under two-lane closure during the off-peak period. The proposed prediction frameworks therefore cannot help machine learning methods to predict traffic variables under extremely severe traffic conditions. The error feedback mechanism, however, can reduce predictive errors in terms of MAPE and RMSE under these traffic conditions.

Table 4.11: Comparison of prediction accuracy when two lanes were blocked during the peak period

	MPE (%)			MAPE (%)			RMSE (seconds)		
Duration of closure: 30 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-3.42	-8.04	-9.62	15.85	17.49	19.69	330.65	152.52	157.49
GM	-2.54	-10.71	-2.49	25.26	35.00	23.48	360.69	293.24	361.14
NN	-5.06	-3.90	-0.18	19.86	16.39	18.46	341.27	169.53	152.07
RF	-0.69	-1.97	-1.45	21.22	20.24	18.03	386.48	331.74	270.55
SVR	-8.62	-7.30	-3.17	32.22	26.33	19.83	388.42	341.95	207.66
Duration of closure: 60 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-10.45	-12.10	-15.47	27.08	29.31	27.37	736.78	248.94	243.03
GM	-10.56	-51.75	-10.62	28.63	88.53	27.54	736.20	468.63	635.59
NN	-1.76	-4.87	-2.46	19.41	20.88	25.61	713.19	440.73	393.69
RF	-6.64	-1.46	-0.33	19.02	22.64	22.60	777.71	721.72	584.08
SVR	-9.92	-6.58	-6.01	24.21	26.84	26.23	781.89	463.54	451.96
Duration of closure: 90 minutes									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-22.06	-29.31	-32.34	37.34	43.57	47.06	1143.80	647.86	676.81
GM	-19.07	-30.11	-11.98	31.59	57.67	28.26	1042.01	488.47	935.46
NN	-3.44	-4.87	-2.46	20.02	32.74	40.10	1471.14	1158.12	902.29
RF	-3.45	-0.85	-1.80	24.10	26.12	25.46	1553.82	1500.18	1080.83
SVR	-6.13	-3.20	-6.82	45.98	51.16	39.57	1819.31	899.36	766.82

4.5.3 Further analysis under abnormal traffic conditions

4.5.3.1 Different data resolution

Both 5-minute and 15-minute traffic data are commonly used in short-term traffic prediction for ITS applications. Traffic data from the Urban Traffic Management and Control (UTMC) (Cheese *et al.*, 1998) system is collected, stored and integrated in 5 minutes, whereas a time resolution of 15 minutes is used in the ASTRID (Hounsell & McLeod, 1990) database to receive traffic data from the SCOOT traffic control system.

Some existing literature, such as Dougherty & Cobbett (1997) and Abdulhai *et al.* (2002), demonstrates that the traffic data time resolution plays an important role in traffic prediction under normal traffic conditions. The previous sections predicted one-step ahead link travel time using 5-minute granularity. Here, one-step ahead traffic prediction accuracy is tested using aggregated 15-minute traffic data for Scenario 2 under abnormal traffic conditions when one lane was blocked during the peak period. The results are given in Table 4.12.

Table 4.12: Comparison of prediction using data at 15-minute granularity using kNN with three different prediction frameworks

	MPE(%)	MAPE(%)	RMSE (sec)
Duration of closure: 30 minutes			
1-stage	4.76	14.03	47.80
2-stage	5.19	13.41	39.36
3-stage	3.10	9.40	30.44
Duration of closure: 60 minutes			
1-stage	0.13	17.81	87.13
2-stage	0.12	12.89	62.98
3-stage	-0.38	12.87	60.11
Duration of closure: 90 minutes			
1-stage	0.33	21.21	88.50
2-stage	1.80	17.96	71.31
3-stage	0.89	17.14	64.21

4.5.3.2 Comparison with the Kalman filter based method

In the proposed 3-stage traffic prediction framework, a mechanism for feedback correction using prediction errors from the past is added to the prediction result of the 2-stage framework in order to improve the prediction accuracy under both normal and abnormal traffic conditions. The above prediction results in Scenario 1 and Scenario 2 demonstrate this improvement regardless of the machine learning method used. This subsection compares two types of feedback, the proposed error correction structure and the feedback in a Kalman filter. The Kalman filter uses a feedback system in each iteration, the purpose of which is to update the estimate of the state vector of a system based upon information in a new observation.

The linear Kalman filter applied in this research is derived from existing studies which estimated the short-term traffic prediction problem in linear systems (Okutani

& Stephanedes, 1984; Chien & Kuchipudi, 2003; Yang, 2005; Xie *et al.*, 2007; Zhu *et al.*, 2009). A non-linear Kalman filter and an Extended Kalman filter are used in prediction when there is a need to estimate more complex relationships of multivariate inputs needs (Antoniou *et al.*, 2007). This test only uses travel time data from one link as an input and therefore, a linear Kalman filter is selected for prediction. The state and observation equations used in a liner Kalman filter are $x(t) = Ax(t-1) + w(t-1)$ and $y(t) = Hx(t) + n(t)$. The state transition matrix A is defined as $\frac{1}{n}\sum_{k=1}^{n}\frac{h_k}{h_{k-1}}$, where h_k is the historical traffic data in the training dataset; the transformation matrix H is an identity matrix. The assumption of process noise w is $p(w) \sim N(0, Q)$ and the measurement noise n is $p(v) \sim N(0, R)$, where Q is predetermined to 10,000 and R is the average variance of the historical travel time.

The comparison of prediction accuracy between the kNN based model using the 3-stage prediction framework and the Kalman filter based traffic prediction model is given in Table 4.13. The test was carried out using link travel time data generated in Scenario 2 when one lane was blocked during the peak period for a duration of 60 minutes.

Table 4.13: Comparison of prediction accuracy between the kNN and Kalman filter based methods under abnormal traffic conditions

	MPE (%)	**MAPE (%)**	**RMSE (seconds)**
kNN:3-stage	0.72	6.91	38.51
Kalman filter	-1.89	18.31	101.18

It is clear that the kNN based traffic prediction model using the 3-stage framework is superior to the Kalman filter based model for traffic prediction under abnormal

traffic conditions. The Kalman filter based model is less able to deal with abnormal traffic conditions and the predicted traffic variables are underestimated.

4.6 Summary

In this chapter, three short-term traffic prediction frameworks with five machine learning methods were evaluated using traffic data from simulation under both normal and abnormal traffic conditions. The prediction results show that the proposed 3-stage prediction framework can improve the accuracy of traffic prediction regardless of the machine learning method used under Scenario 1 – normal traffic conditions and in Scenario 2 where only one lane was blocked during testing. The average improvement in prediction quantified using the MAPE metric is 37.8% in Scenario 1 – normal traffic conditions and 31.75% in Scenario 2 during abnormal traffic conditions when one lane was blocked. Similarly, the average improvement of RMSE is 34.9% and 32.8% in Scenario 1 and Scenario 2 respectively.

In Scenario 3 – where both lanes are blocked, the 2-stage and 3-stage frameworks cannot accurately predict link travel time. Under these extreme traffic conditions, it is difficult for machine learning methods to model the relationship between inputs and outputs. SSA, however, may eliminate the peak values contained within the testing dataset when the value of traffic variables is extremely large.

This chapter has demonstrated that the multi-stage traffic prediction framework can improve traffic prediction accuracy regardless of the machine learning method used. The data smoothing stage can help machine learning methods accurately extract the main trends. The error feedback mechanism can be used to correct prediction error

bias because of the existence of the strong relationship between the current prediction error and previous errors.

The next chapter will build on this work by evaluating the prediction accuracy of the proposed frameworks by applying them to a real-world traffic environment.

Chapter 5 Short-term Traffic Prediction Using Real-world Traffic Data

5.1 Introduction

The previous chapter evaluated the proposed traffic prediction frameworks based on five different machine learning methods under both normal and abnormal traffic conditions using link travel time data from simulation experiments. It was shown that, in this simulated environment and with data smoothing and error feedback, the proposed frameworks can improve the accuracy of short-term traffic prediction.

This chapter further demonstrates the effectiveness of the proposed short-term traffic prediction frameworks using real traffic data. The prediction accuracy is evaluated in a range of traffic conditions in urban areas in UK. The robustness of the model is examined by applying the proposed frameworks to both normal and abnormal traffic conditions. The proposed traffic prediction frameworks can be used to predict not only the link travel time but also the traffic flow. Real traffic data including link travel time and traffic flow collected from different cities under different traffic management systems was used to examine the transferability of the proposed frameworks. The metrics used for the quantitative evaluation of accuracy are MPE, MAPE and RMSE.

5.2 Real-world traffic data

The context of this research for short-term traffic prediction is the urban road network in the UK. At least two testing locations in urban areas are required to examine the

location transferability of the proposed traffic prediction models. The selected locations need to suffer from traffic congestion during both morning and evening periods and they have to be well-equipped with devices for traffic data collection. In addition, information about abnormal traffic in the area of the selected sites is needed.

Two different types of real-world traffic variables, link travel time and traffic flow were used in this chapter. Link travel time data was collected from London and Maidstone using ANPR cameras. Traffic flow data was collected inside London from two corridors using ILDs. An overview of the traffic variables used in this chapter is presented in the following subsections.

5.2.1 Link travel time data

5.2.1.1 Travel time data in London

All link travel time data from London was obtained from the London Congestion Analysis Project (LCAP) (TfL, 2010). LCAP is operated by TfL to capture and store link travel time data in London based on ANPR camera data. ANPR cameras record an image of the license plate and the corresponding time of passing vehicles. Image processing techniques extract the vehicle registration number from these images and by matching the license plates from pairs of ANPR cameras the travel time of vehicles between the two camera locations can be measured.

LCAP exports the cleaned averaged link travel time data at 5 minute intervals. The strategies used in LCAP to clean the raw travel time data are presented in Appendix B.1. The main objective of this data cleaning includes patching missing travel time data and removing data relating to vehicles that are not subject to normal traffic rules. For example, emergency vehicles such as police cars and ambulances can

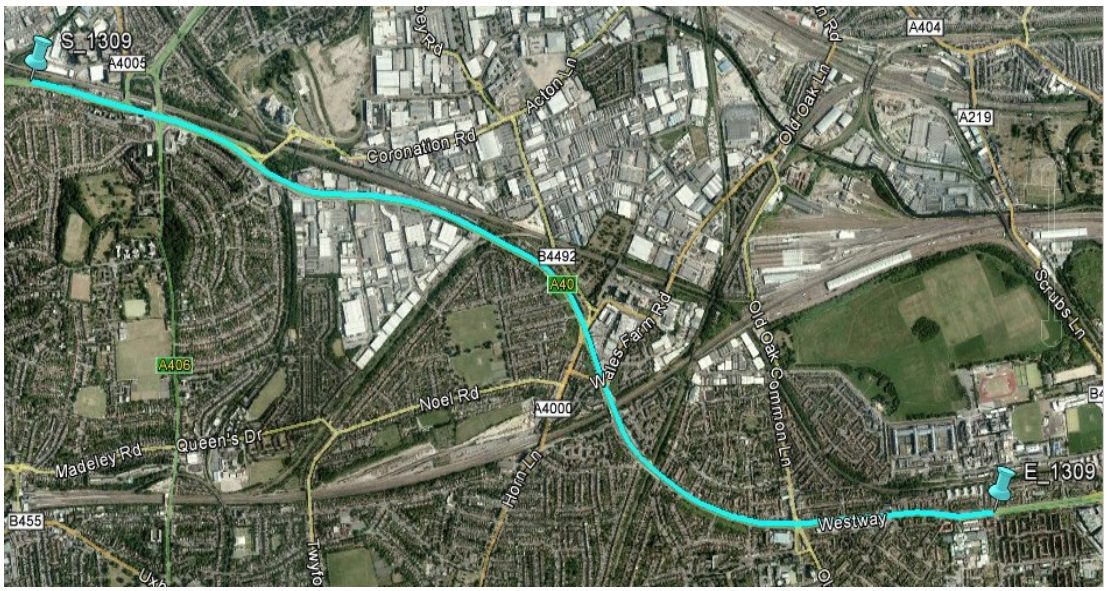
travel faster than normal vehicles. Similarly, the records of some vehicles, such as taxis, which may travel excessively slowly due to picking up or dropping off passengers, or taking a detour between two camera sites may also be removed. The essential idea underlying the data cleaning strategy used in LCAP, therefore, is to remove excessively slow and fast travel time data. This strategy in general is reasonable but cannot always reliably discriminate between illegitimate trajectories (e.g., taxi and emergency vehicles) and legitimate but extreme trajectories (e.g., excessively slow travel times arising from abnormal traffic conditions). This limitation must be born in mind in interpreting the empirical results presented in this chapter.

The main objective of this research is to develop robust and accurate models for short-term traffic prediction under both normal and abnormal traffic conditions on urban roads. Hence, the selected link should satisfy the following requirements:

- An urban arterial road context;
- Availability of detailed abnormal event information, such as location, start/end time, duration and severity;
- Presence of signalised junctions that are controlled by advanced signal timing plans;
- Presence of traffic congestion during both morning and evening periods;
- Availability of at least continuous three month's continuous travel time data.

Considering the above requirements, link 1309 of the A40 in London with a length of 5.63 km was selected for this application. The topology of this road link in

LCAP systems is shown in Figure 5.1. As presented above, the selected link 1309 on the A40 road in London is monitored by a pair of ANPR cameras. The direction of travel is from west to east.



S_1309: Start Point of Link 1309 (equipped with ANPR)

E_1309: End Point of Link 1309 (equipped with ANPR)

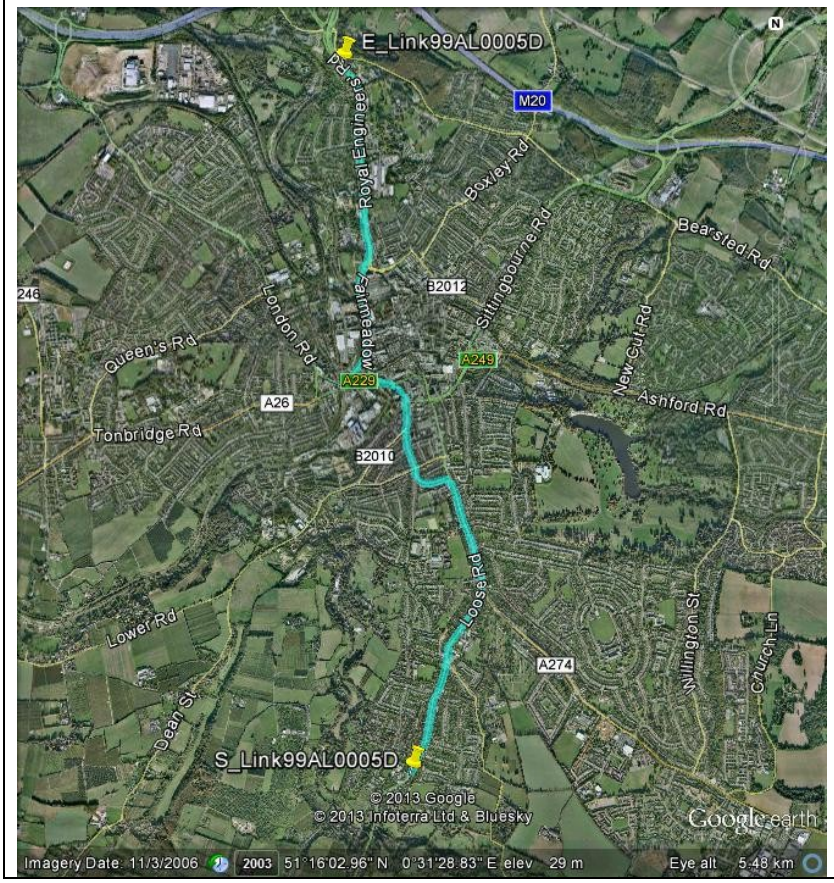
Figure 5.1: Link 1309 on the A40 road in London (Source: Google Earth)

5.2.1.2 Travel time data in Maidstone

The aggregated 5-minute link travel time data used in this PhD thesis from Maidstone was directly provided by Kent County Council (KCC). The link travel time data in Maidstone was monitored by ANPR. Only the cleaned aggregated link travel time data, not the raw matched data, was provided because of privacy concerns. The data cleaning method used by KCC is presented in Appendix B.2. Similar to the cleaning methods in LCAP system, the excessively slow travel times that may happen under abnormal traffic conditions.

Considering the requirements presented in Section 5.2.1.1, the Link 99AL0005D in the Maidstone area, with a length of 6.4 was selected for this research. The selected

link connects the A229, Loose Road, to the A229, Royal Engineers Road, Maidstone. The topology of this road link is shown in Figure 5.2.



S_Link99AL0005D: Start Point of Link 99AL0005D (equipped with ANPR)

E_Link99AL0005D: End Point of Link 99AL0005D (equipped with ANPR)

Figure 5.2: Selected Link 99AL0005D in Maidstone (Source: Google Earth)

5.2.2 Traffic flow data in London

All the traffic flow data used in this study was obtained from Inductive Loop Detectors (ILDs), which form part of the SCOOT (Hunt *et al.*, 1981) traffic control system in Central London. The outputs of the ASTRID system (Hounsell & McLeod, 1990) associated with SCOOT are the aggregated 15-minute traffic flow and occupancy data. There are over 6000 ILDs in London that provide near real-time traffic flow data for all the major links and, due to this comprehensive spatial and temporal coverage, SCOOT ILD data can be widely used in the application of traffic

estimation and prediction for arterial roads in London (Krishnan, 2008). ILDs deployed under the road are connected to a power source, which applies an oscillating voltage. The oscillating current causes a magnetic field in the loop area. When a large metallic objective such as a vehicle passes over the ILD, the inductance around the ILD is reduced and the oscillator frequency is increased. A vehicle's presence is determined when frequency change exceeds the pre-determined threshold. Single ILDs are widely used to collect traffic flow and occupancy data at a fixed location. The ILD data used in this research are from two separate corridors in central London; the Russell Square corridor and the Marylebone Road corridor. The corridor characteristics are described below.

- *Russell Square corridor*

  The Russell Square corridor is a frequently congested corridor in Central London. It connects the junction between Euston road and Upper Woburn Place to the junction between Southampton Row and Theobalds Road and runs in both directions with two lanes along most of the corridor, except for the part from Guilford Street to Russell Square adjacent to Bloomsbury Square, which has only one southbound lane (Krishnan, 2008). A map of the Russell Square corridor is shown in Figure 5.3.

- *Marylebone corridor*

  The Marylebone corridor is more heavily congested than the Russell Square corridor. Marylebone Road is an important thoroughfare in the centre of London from Euston Road at Regent's Park to the A40 Westway at Paddington. This corridor has two directions with three lanes. A map of the Marylebone corridor is shown in Figure 5.4.

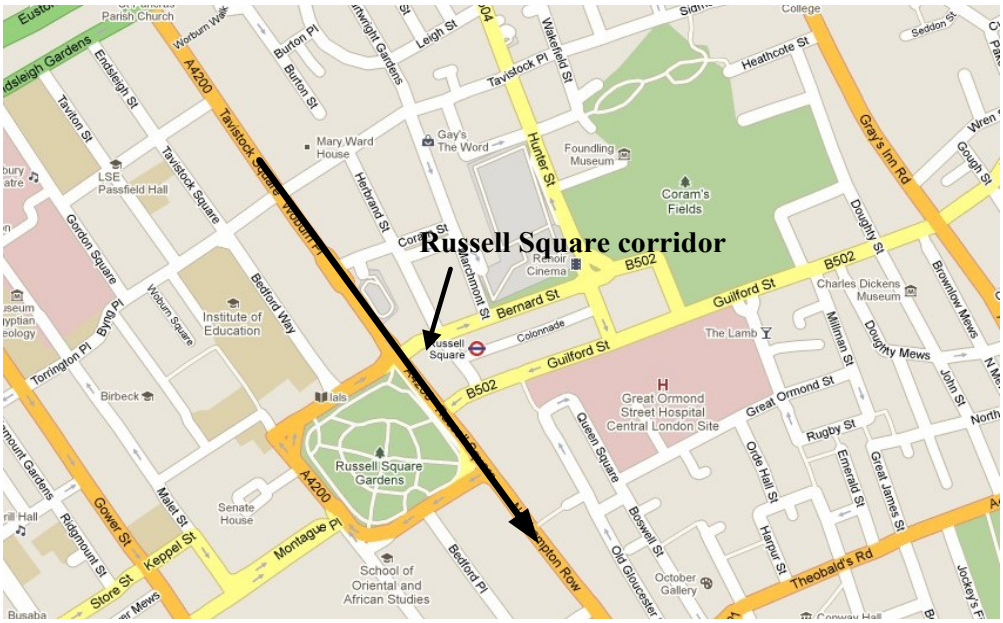


Figure 5.3: The Russell Square corridor (Source: Google Maps)

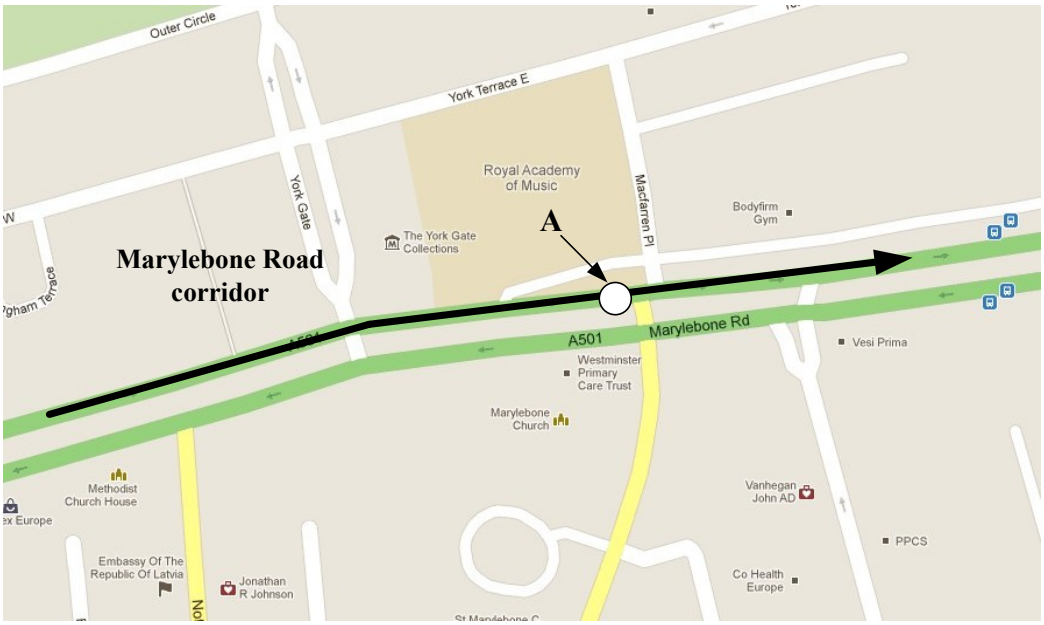


Figure 5.4: The Marylebone Road corridor (Source: Google Maps)

Both the Russell Square corridor and the Marylebone Road corridor are frequently congested corridors that are characteristic of many similar corridors in Central

London (Guo *et al.*, 2013). Moreover, these are two corridors that we have studied in detail in the past and for which we therefore have relevant readily accessible data. Therefore, traffic flow data from these two corridors is used in these experiments.

- *Abnormal traffic condition information for the Marylebone Road corridor*

  The proposed framework needs to be evaluated during both normal traffic conditions and traffic incidents. Information about abnormal events such as accidents and incidents within the duration of the above ILD dataset is obtained from a data feed in TpegML (Transport protocol expert group in Extensible Markup Language) format disseminated by the British Broadcasting Corporation (BBC). The BBC obtains information of abnormal events from Trafficlink, which is a company providing real-time or near real-time traffic information to public and private agencies. Traffic information is aggregated from a number of sources such as the London Traffic Information System (LTIS). This feed consists of information on planned events and unplanned incidents (Hu *et al.*, 2008). Planned event information is provided by organisations such as local authorities, the police, utility companies and event organisers. Information about unplanned incidents and accidents is mainly obtained from Transport for London staff, who can monitor Closed Circuit television (CCTV) cameras, and the police who are informed by the public about accidents and other disruptions (Hu *et al.*, 2008). The traffic information service from the BBC can therefore be used to identify the location, duration and the degree of severity of each incident.

5.3 Short-term traffic prediction under normal traffic conditions

This section presents experiments that were undertaken to compare the prediction performance of the three different frameworks with five machine learning methods described in Chapter 3 using traffic data under normal traffic conditions. Two types of traffic variables were used in the experiments, namely link travel time data and traffic flow data. Link travel time data was extracted from the A40 road in London in the UTMC system (Cheese *et al.*, 1998) that can collect travel time data and integrate them into 5 minute intervals. Traffic flow data was collected from the Russell Square corridor and Marylebone corridor in central London in the ASTRID database (Hounsell & McLeod, 1990) that receives traffic data aggregated at 15-minute intervals from the SCOOT traffic control system.

5.3.1 Short-term travel time prediction using data from the A40 road in London under normal traffic conditions

In this subsection, only travel time data collected from link 1309 of the A40 road in the London LCAP system under normal traffic conditions is tested. Travel time data for a period of three months between January and March is divided into two datasets. Training data is from 3rd January 2011 to 17th March 2011; while testing data is from 18th March 2011 to 31st March 2011. Since the focus in on weekdays, weekend data is eliminated. The travel time prediction accuracy is compared using MPE, MAPE and RMSE metrics in Table 5.1. Figure 5.5 and Figure 5.6 present the values of MAPE for one-step and multi-step ahead prediction for five machine learning methods using three traffic prediction frameworks.

Table 5.1: Comparison of prediction accuracy of link travel time on the A40 road in London using three different frameworks with five machine learning methods under normal traffic conditions

	MPE (%)			MAPE (%)			RMSE (sec)		
One-step ahead (5-min ahead)									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.68	0.56	0.45	5.04	3.76	3.66	38.26	32.86	29.46
GM	-0.36	-0.13	-0.11	6.34	3.82	3.78	43.66	28.82	27.26
NN	-3.33	-2.39	-2.23	7.80	6.27	5.32	44.97	37.39	32.34
RF	-0.40	-0.13	-0.12	4.72	3.32	3.31	40.33	34.68	32.60
SVR	-3.45	-3.36	-2.08	7.09	4.51	3.96	67.98	56.19	40.43
Mean	-1.37	-1.09	-0.82	6.20	4.34	4.01	47.04	37.99	32.49
Multi-step ahead (15-min ahead)									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	1.53	1.55	1.25	8.13	7.71	6.74	79.41	74.50	70.79
GM	-0.77	-0.61	-0.47	8.76	7.56	6.68	81.37	76.14	64.03
NN	-6.77	-4.06	-3.17	10.87	8.88	7.45	84.18	83.06	68.43
RF	-0.97	-0.73	-0.20	7.35	6.89	5.49	81.45	80.08	49.77
SVR	-6.52	-3.21	-2.84	10.98	8.91	6.35	84.38	81.69	80.32
Mean	-2.70	-1.42	-1.09	9.22	7.99	6.54	82.16	79.09	66.67

It is clear that both a data smoothing structure and a feedback mechanism can in general improve prediction accuracy. The 2-stage traffic prediction framework can improve prediction accuracy for both one-step and multi-step ahead prediction under normal traffic conditions. The feedback mechanism can significantly improve multi-step ahead prediction accuracy under normal traffic conditions; however, it does not help prediction models to improve one-step ahead traffic prediction significantly. For example, the value of MAPE by kNN using the 2-stage framework for one-step ahead prediction is 3.76%, which is reduced to 3.66% using the feedback mechanism, in other words, an improvement of 2.7%. For multi-step ahead traffic prediction, the MAPE metric improves from 7.71% for kNN using the 2-stage framework to 6.74% using the 3-stage framework, a 12.6% improvement.

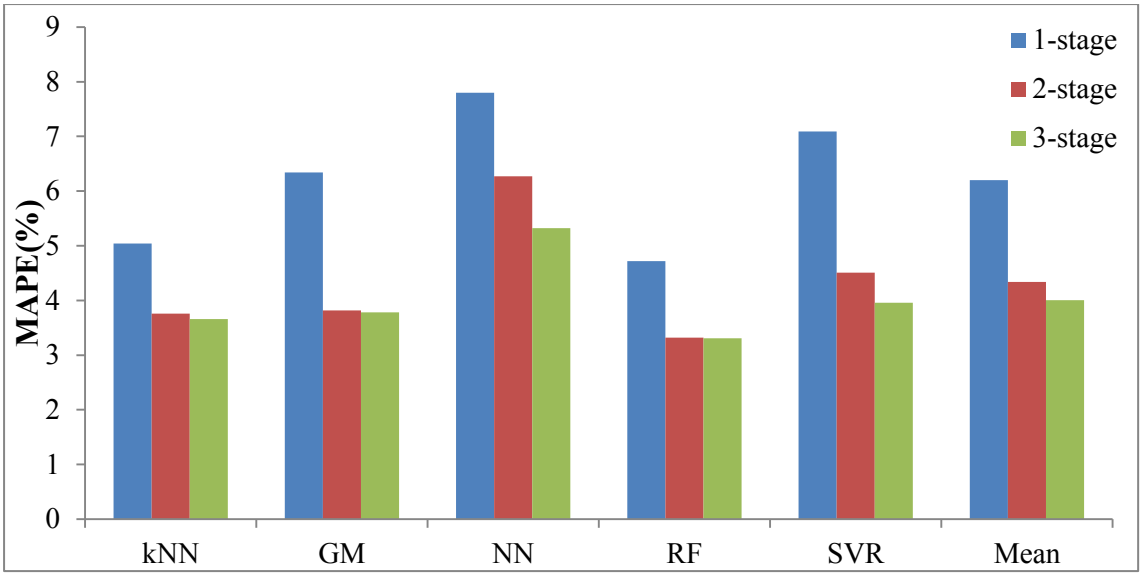


Figure 5.5: MAPE for five machine learning methods and three prediction frameworks of one-step ahead prediction under normal traffic conditions on link 1309 of the A40 road in London

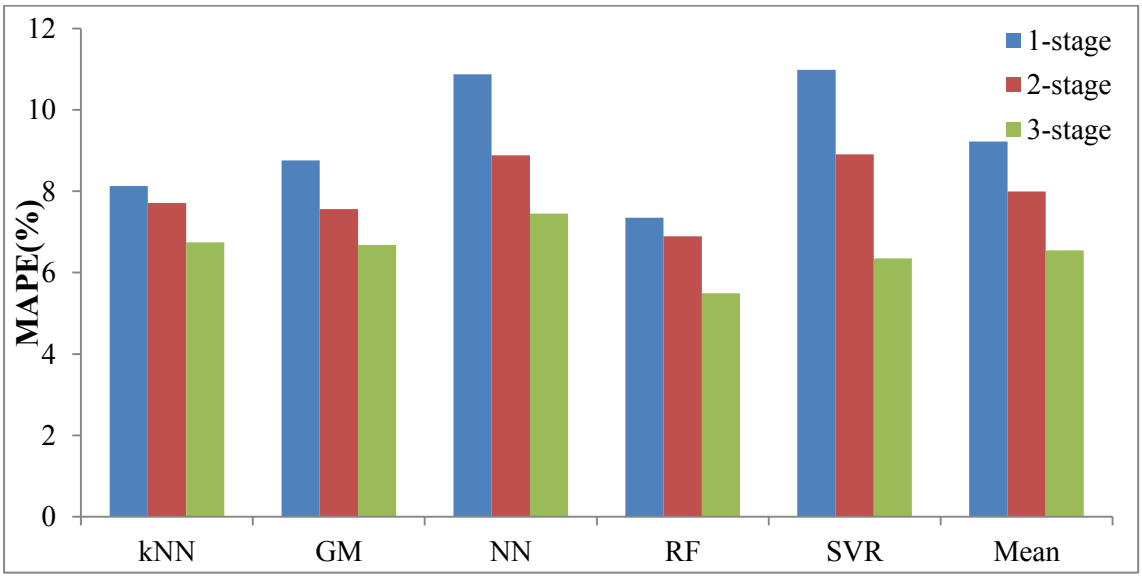


Figure 5.6: MAPE for five machine learning methods and three prediction frameworks of multi-step ahead prediction under normal traffic conditions on link 1309 of the A40 road in London

The results in Table 5.1 show that the RF based method with the 3-stage prediction framework has the most accurate prediction. Figures 5.7, 5.8 and 5.9 show the scatter-plot of predicted and observed travel time data, the error auto-correlation plot of predictions, the histogram of error distribution and the sample time-series plot between predicted and observed travel time of the RF method with the 1-stage, 2-stage and 3-stage prediction frameworks. It can be seen from the scatter-plots of predicted and observed travel time data that the three prediction models tended slightly to underestimate the observed travel times for the higher travel times under highly congested conditions.

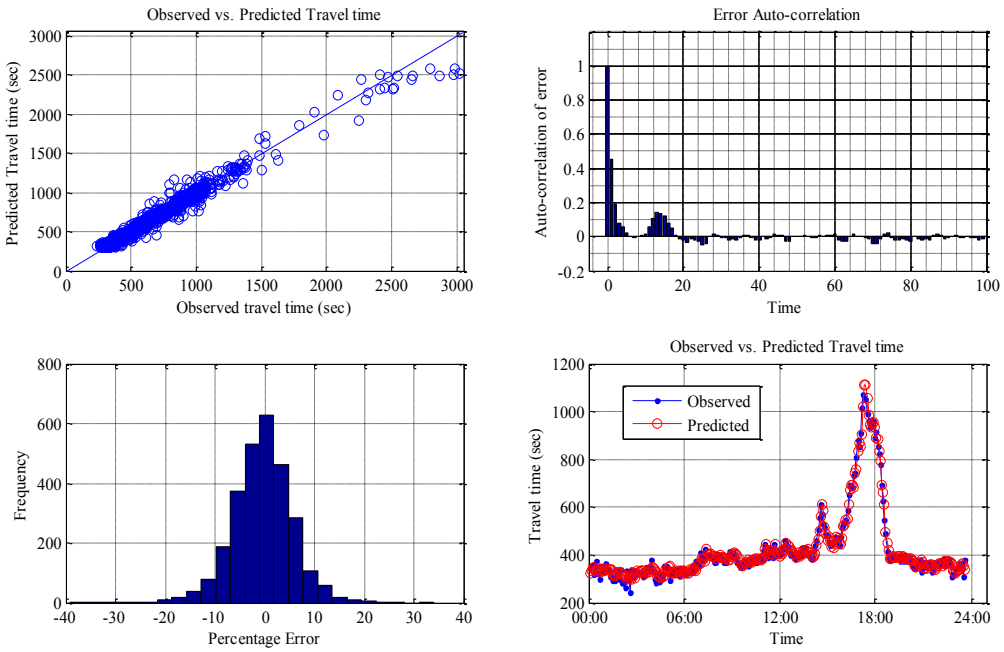


Figure 5.7: Travel time prediction performance using RF with the 1-stage framework on the A40 road in London under normal traffic conditions

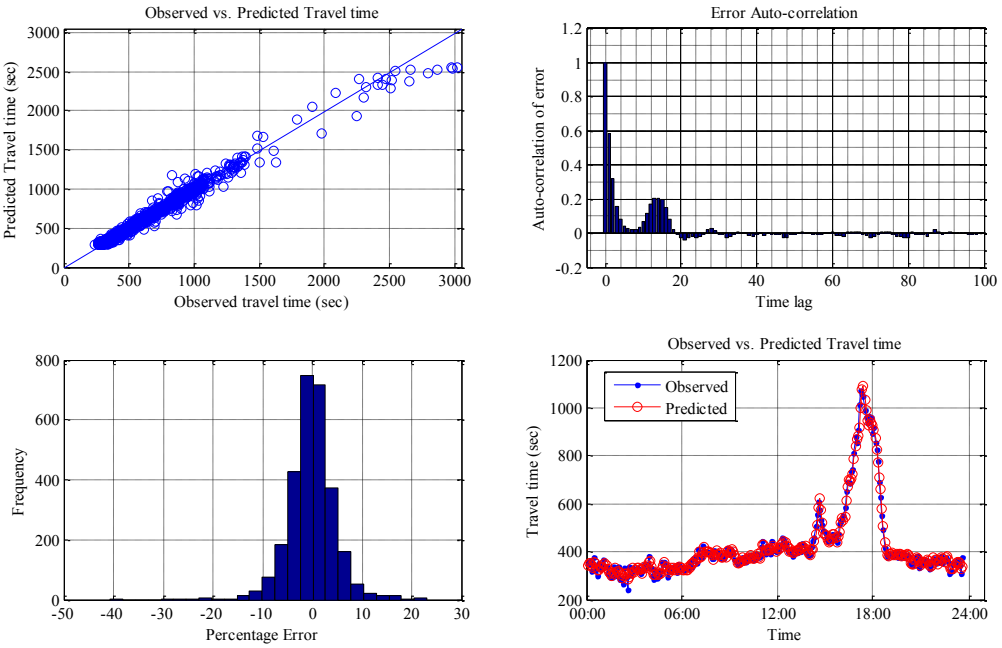


Figure 5.8: Travel time prediction performance using RF with the 2-stage framework on the A40 road in London under normal traffic conditions

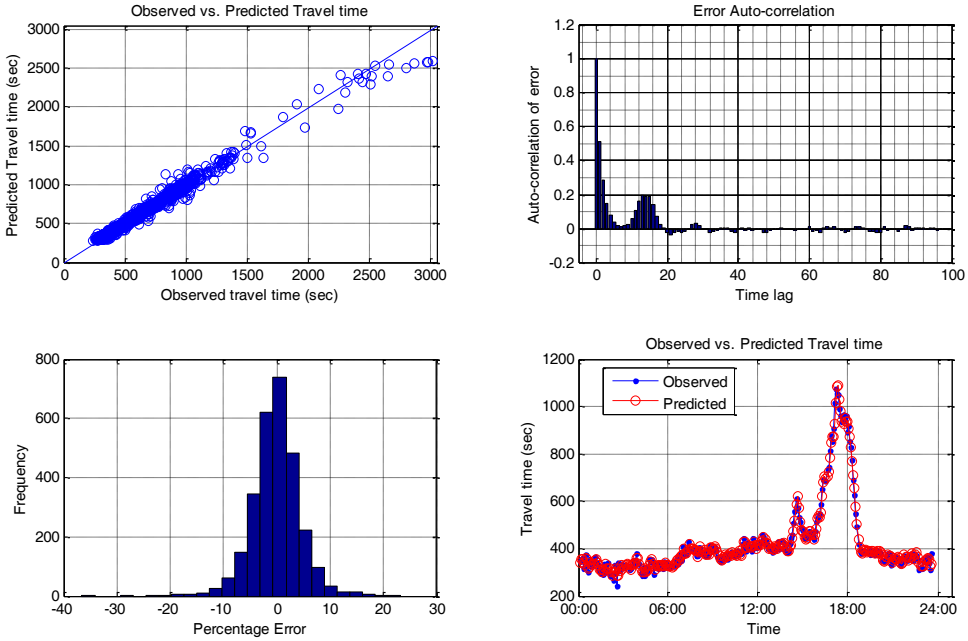


Figure 5.9: Travel time prediction performance using RF with the 3-stage framework on the A40 road in London under normal traffic conditions

5.3.2 Short-term traffic flow prediction using data from the Russell Square corridor in London under normal traffic conditions

In the dataset, the aggregated 15-minute traffic flow and occupancy data from the ASTRID system (Hounsell & McLeod, 1990) was obtained from the Russell Square corridor (Figure 5.3). Typical link lengths varied between 90 m and 150 m in the Russell Square corridor. Erroneous data due to detector faults and data caused by abnormal events were filtered out in this dataset using Univariate tests (Robinson, 2005) combined with the Modified Adjacent Detector Test (MADT) given in Krishnan (2008) so as to obtain error free traffic data. Hence, only normal and non-incident traffic conditions are included in this dataset. The data recorded from the Russell Square corridor covers a three-month period between June and August 2007. Traffic data was divided into two groups, with 26 days of training data from June and July and 6 days of testing data from August. Only traffic data from weekdays was used.

Three traffic prediction frameworks, 1-stage, 2-stage and 3-stage frameworks with five different machine learning methods, were evaluated using this traffic flow data. The results comparing the one-step and multi-step ahead prediction accuracy are given in Table 5.2. From this table it can be seen that the 3-stage traffic prediction framework outperforms other frameworks when MAPE and RMSE metrics are considered.

Table 5.2: Comparison of prediction accuracy of traffic flow on the Russell Square corridor using three different frameworks with five machine learning methods under normal traffic conditions

	MPE (%)			MAPE (%)			RMSE (vehicles/hour)		
One-step ahead (15-min ahead)									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-0.36	-0.51	-0.26	12.67	9.61	9.36	73.20	55.96	54.34
GM	1.40	1.24	1.01	13.66	12.98	11.73	77.49	73.59	67.02
NN	2.15	0.71	-0.72	12.19	8.70	8.45	72.48	50.57	49.51
RF	1.52	0.84	0.80	11.68	8.91	8.77	67.04	52.12	51.55
SVR	0.44	0.49	0.44	13.95	9.72	9.63	78.16	58.84	58.78
Mean	1.03	0.554	0.254	12.83	9.98	9.59	73.67	58.22	56.24
Multi-step ahead (1-hour ahead)									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-2.36	-2.67	-0.30	17.65	16.70	13.08	104.35	99.82	77.30
GM	0.92	0.88	0.86	19.10	19.08	13.53	104.46	103.72	77.42
NN	-3.96	8.65	-4.72	19.57	17.51	13.70	102.29	90.55	75.28
RF	2.51	1.71	1.42	15.58	14.23	12.08	84.82	79.78	69.80
SVR	2.78	1.89	1.37	18.54	17.67	15.16	110.63	103.26	92.29
Mean	-0.02	2.09	-0.27	18.09	17.04	13.51	101.31	95.43	78.42

Figure 5.10 and Figure 5.11 illustrate the decrease of MAPE values for one-step and multi-step ahead traffic prediction using traffic flow data from the Russell Square corridor under normal traffic conditions. The results show that the traffic prediction framework with a data smoothing technique is better for one-step ahead prediction; however, this structure does not significantly improve multi-step ahead prediction accuracy under normal traffic conditions. On the other hand, the 3-stage framework with feedback is slightly more accurate than the 2-stage framework without a feedback structure; indeed, a significant advantage of error feedback is the improvement of multi-step ahead prediction accuracy under normal traffic conditions.

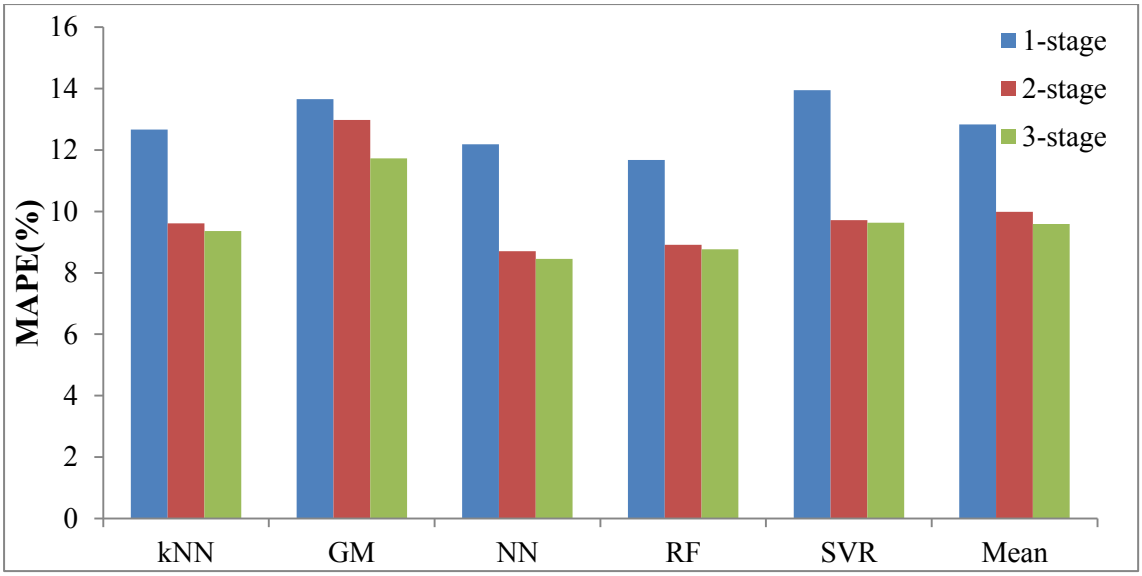


Figure 5.10: MAPE for five machine learning methods and three prediction frameworks of one-step ahead prediction under normal traffic conditions on the Russell Square corridor

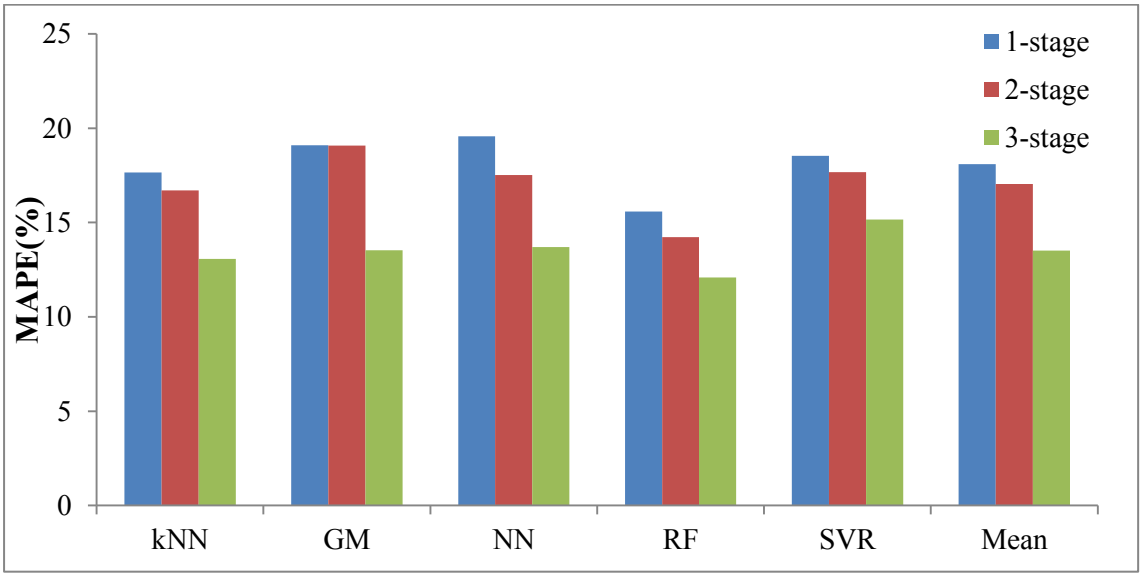


Figure 5.11: MAPE for five machine learning methods and three prediction frameworks of multi-step ahead prediction under normal traffic conditions on the Russell Square corridor

Figures 5.12, 5.13 and 5.14 show the scatter-plot of predicted and observed travel time data, the error auto-correlation plot of predictions, the histogram of error

distribution and the sample time-series plot between predicted and observed travel time in the NN method for each of the 1-stage, 2-stage and 3-stage prediction frameworks.

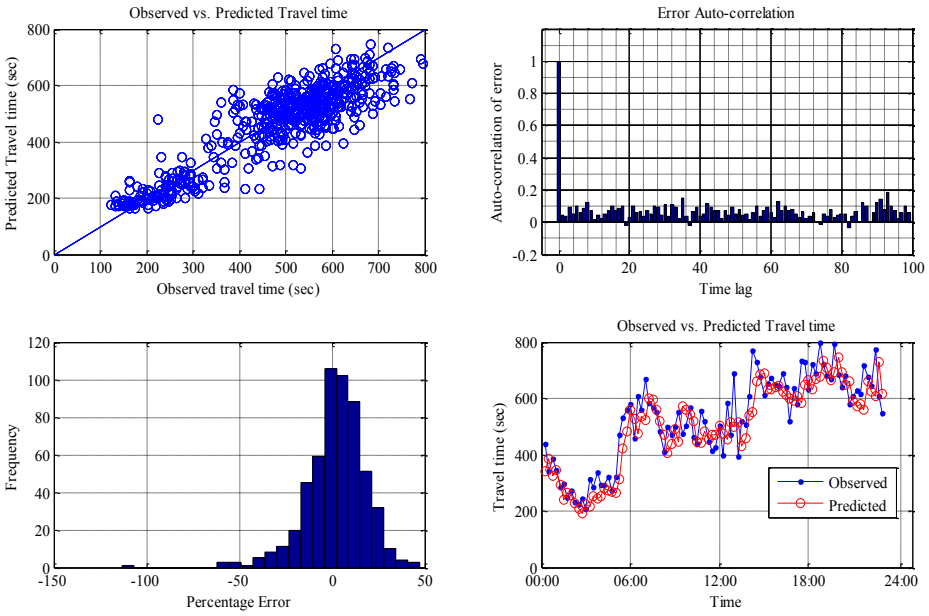


Figure 5.12: Traffic flow prediction performance using NN with the 1-stage framework on the Russell Square corridor in London under normal traffic conditions

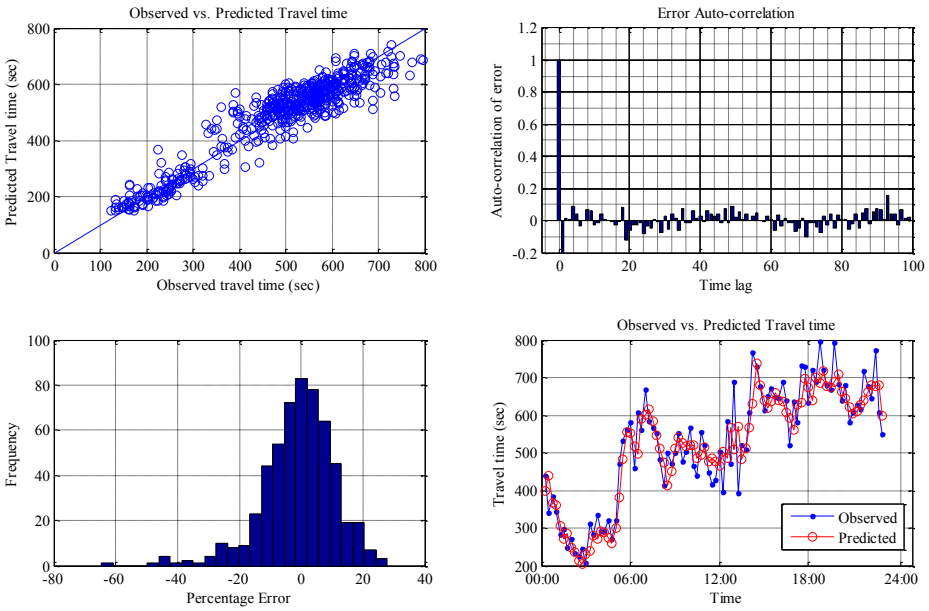


Figure 5.13: Traffic flow prediction performance using NN with the 2-stage framework on the Russell Square corridor in London under normal traffic conditions

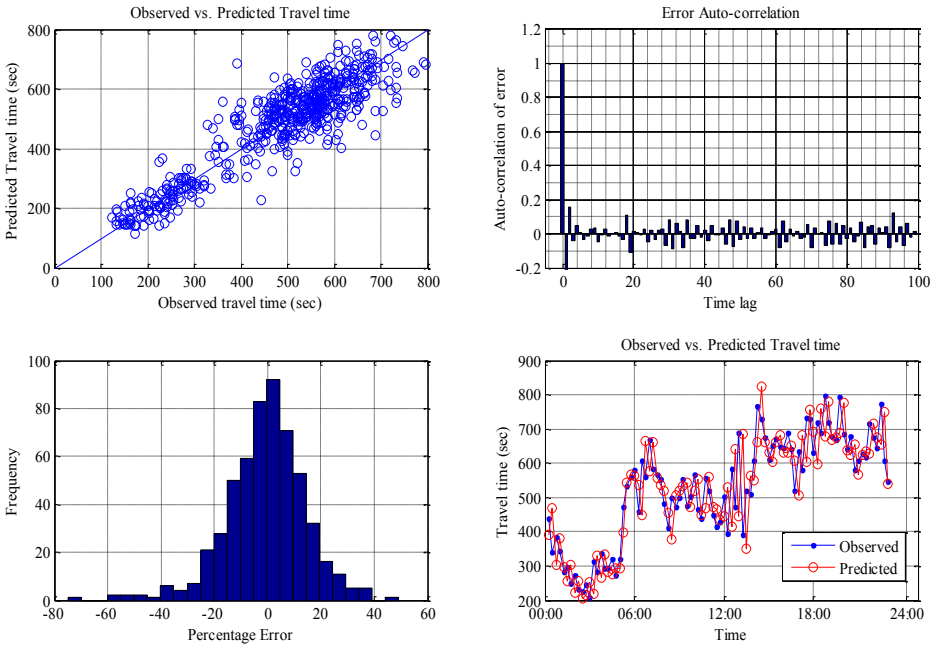


Figure 5.14: Traffic flow prediction performance using NN with the 3-stage framework on the Russell Square corridor in London under normal traffic conditions

5.3.3 Short-term traffic flow prediction using data from the Marylebone corridor in London under normal traffic conditions

Traffic flow data in this subsection is from the Marylebone corridor in central London. The aggregated 15-minute traffic data is from the ASTRID system (Hounsell & McLeod, 1990). In the training dataset, 3,936 records from 41 weekdays during April and May 2008 were selected. Independent testing data is from the 5th June to 19th June, 2008. There are 11 days of testing data left after filtering out weekend data and erroneous data due to detector device faults using the Daily Statistics Algorithm (DSA) (Chen *et al.*, 2003). DSA was demonstrated by Robinson (2005) to be successful in removing erroneous data caused by the failure of detector devices. Traffic data in the training and testing datasets is normal traffic data without incidents and other abnormal events.

The results showing the accuracy of three traffic prediction frameworks with five machine learning methods in one-step and multi-step ahead traffic prediction using data from the Marylebone corridor under normal traffic conditions are given in Table 5.3. The MAPE values in Table 5.3 are presented in Figure 5.15 and Figure 5.16 for 15-minute and 1-hour ahead prediction under normal traffic conditions. Similar to the results in Section 5.3.1 and Section 5.3.2, under normal traffic conditions the prediction accuracy is improved using the data smoothing framework and feedback structure.

Table 5.3: Comparison of prediction accuracy of traffic flow on the Marylebone corridor using three different frameworks with five machine learning methods under normal traffic conditions

	MPE (%)			**MAPE (%)**			**RMSE (vehicles/hour)**		
One-step ahead (15-min ahead)									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	1.46	-0.39	-0.35	8.44	7.04	6.83	119.79	97.33	96.00
GM	3.07	3.18	2.46	10.01	9.57	8.30	134.62	125.47	112.25
NN	3.16	1.96	-0.57	9.92	7.68	6.21	120.78	110.52	92.54
RF	-1.11	-0.82	-0.94	7.68	5.78	5.71	103.50	80.74	78.88
SVR	0.97	0.51	0.52	10.27	6.85	6.80	150.61	111.12	100.21
Mean	1.51	0.89	0.22	9.26	6.98	6.37	125.86	97.04	89.98
Multi-step ahead (1-hour ahead)									
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	7.15	5.89	4.59	15.17	14.17	11.91	222.18	201.04	169.15
GM	-0.13	-3.58	-3.47	15.29	12.41	11.81	222.45	178.97	160.17
NN	-2.98	-1.53	-1.27	14.23	10.89	9.91	189.25	141.35	129.60
RF	-1.80	-1.68	-1.16	10.91	9.89	8.43	140.96	133.96	116.89
SVR	-1.08	-1.75	-1.15	11.37	10.42	9.35	152.87	143.63	129.44
Mean	0.23	-0.53	-0.49	13.39	11.56	10.28	185.54	159.79	141.05

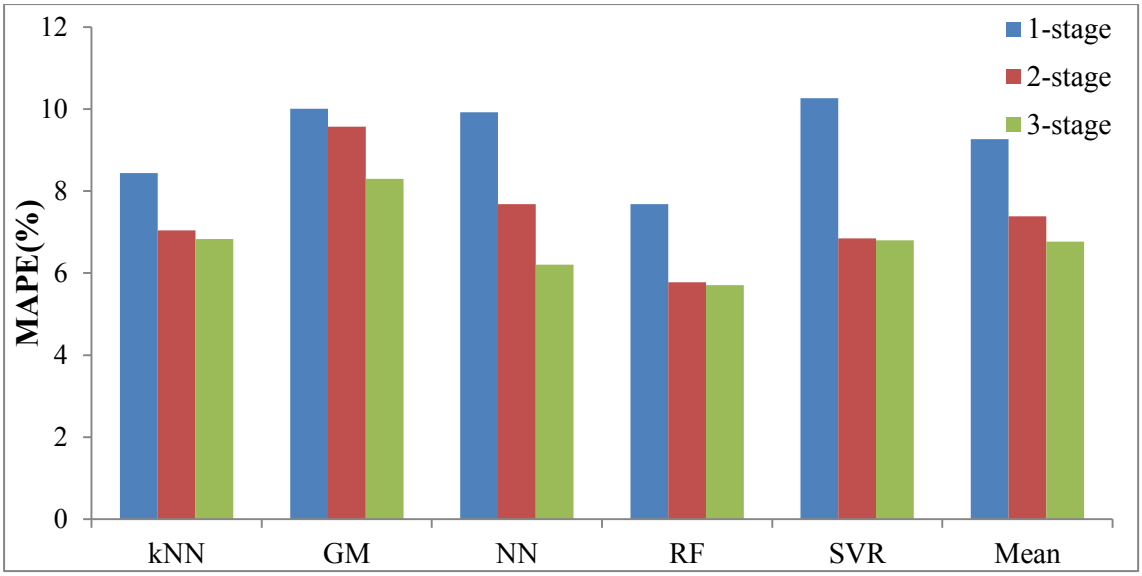


Figure 5.15: MAPE for five machine learning methods and three prediction frameworks for one-step ahead prediction under normal traffic conditions on the Marylebone corridor in London

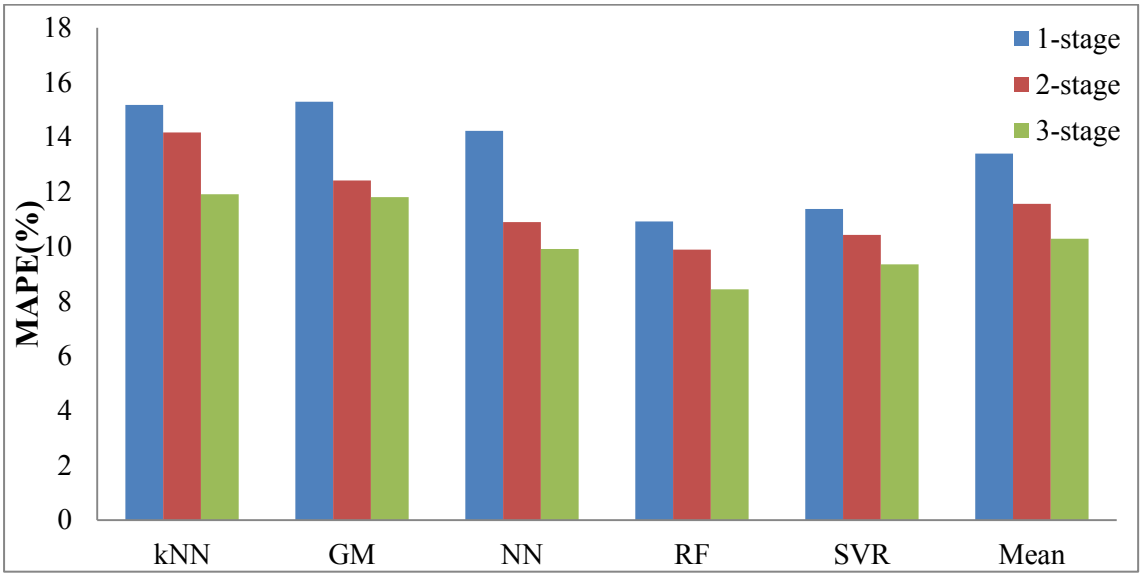


Figure 5.16: MAPE for five machine learning methods and three prediction frameworks for multi-step ahead prediction under normal traffic conditions on the Marylebone corridor

Figures 5.17, 5.18 and 5.19 are the scatter-plot of predicted and observed travel time data, the error auto-correlation plot of predictions, the histogram of error

distribution and the sample time-series plot between predicted and observed travel time of the NN method for each of the 1-stage, 2-stage and 3-stage prediction frameworks.

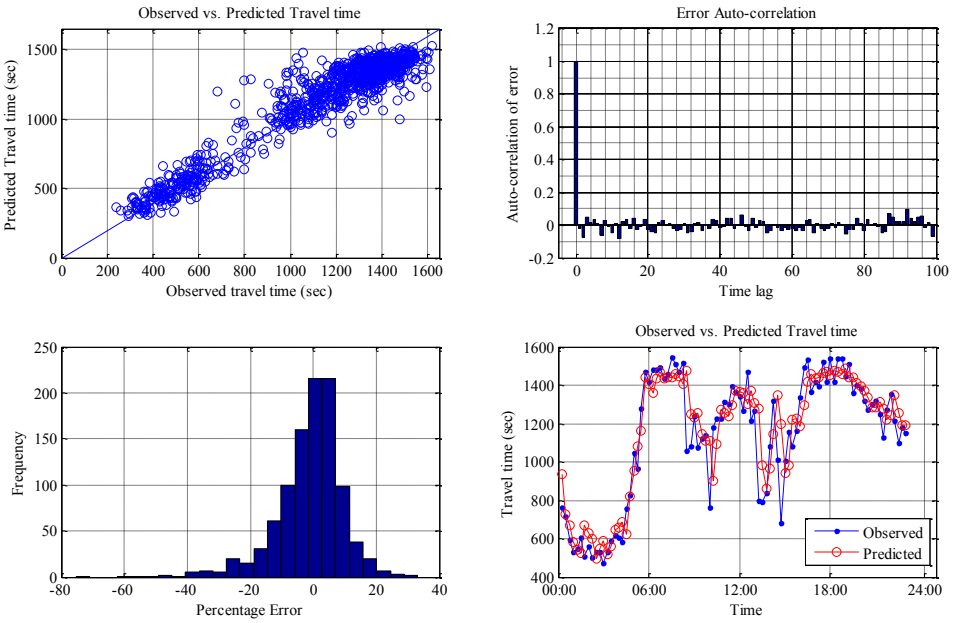


Figure 5.17: Traffic flow prediction performance using RF with the 1-stage framework on the Marylebone corridor in London under normal traffic conditions

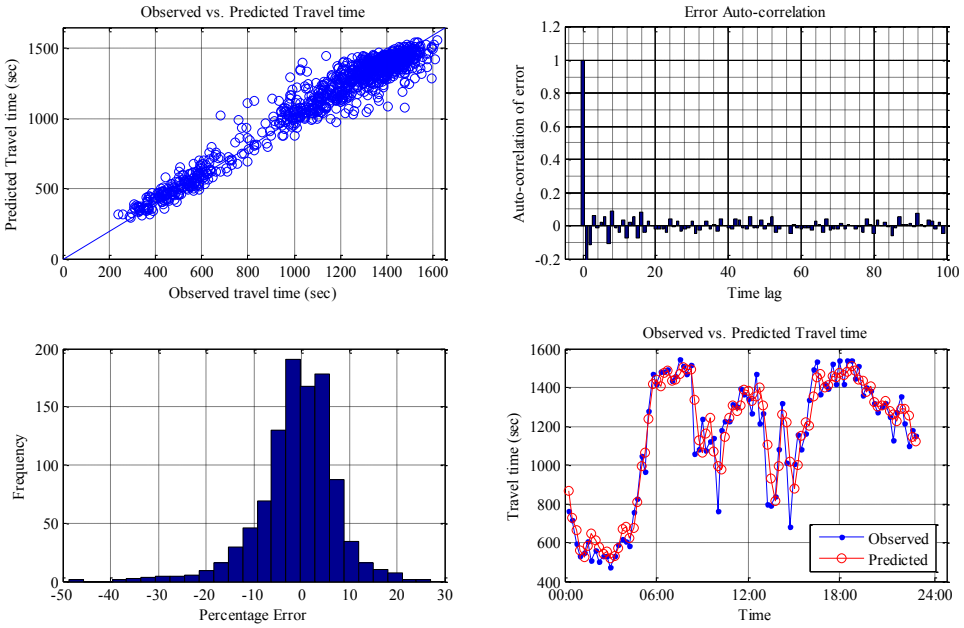


Figure 5.18: Traffic flow prediction performance using RF with the 2-stage framework on the Marylebone corridor in London under normal traffic conditions

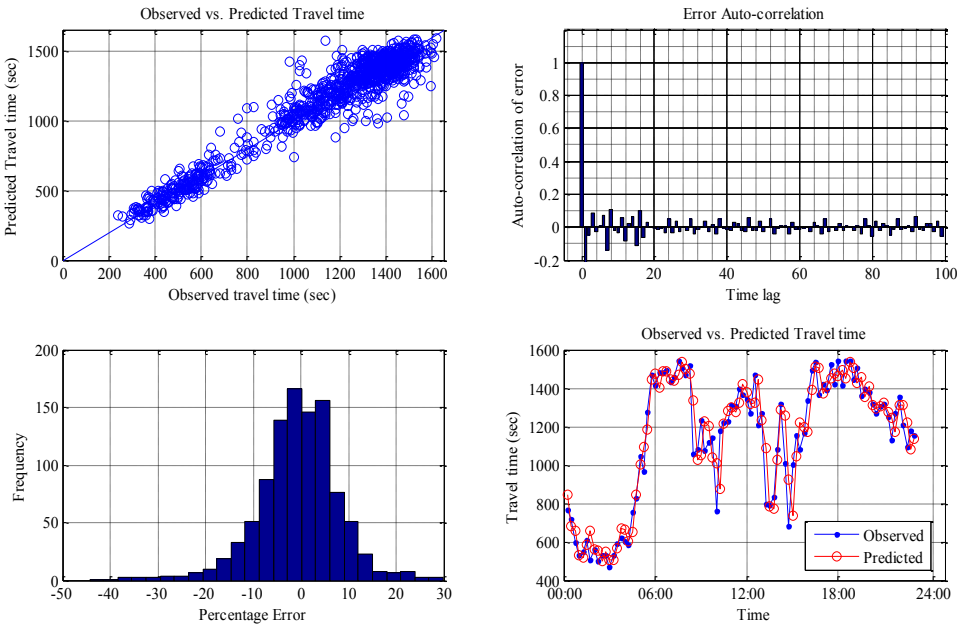


Figure 5.19: Traffic flow prediction performance using RF with the 3-stage framework on the Marylebone corridor in London under normal traffic conditions

5.4 Short-term traffic prediction under abnormal traffic conditions

This section describes experiments that were undertaken to evaluate the data from London and Maidstone for short-term traffic prediction under abnormal traffic conditions. Only one-step ahead traffic prediction was tested under abnormal traffic conditions. When abnormal traffic events happen, traffic patterns may suddenly change. Current traffic predictors are unable to predict this sudden change unless additional information such as detailed information about the abnormal events is provided. Since most information can usually be obtained offline rather than online, however, the results of multi-step ahead prediction are less important than one-step

ahead prediction under abnormal traffic conditions and are beyond the scope of this research.

5.4.1 Short-term travel time prediction using data from the A40 road in London under abnormal traffic conditions

Link travel time data from link 1309 of the A40 road in central London was extracted in this experiment. Traffic data under normal traffic conditions is used in the training dataset while the testing dataset consists of known traffic incidents. Training data is from weekdays during October, November and December 2010; testing data is from 21st December 2010. One lane was blocked on Western Avenue eastbound on the testing day because of a broken down vehicle. Information about abnormal traffic conditions was also directly provided by TfL, including event date, start time, end time, category, location and severity. Figure 5.20 shows the location of the abnormal event. Detailed information of this abnormal event is as below:

- Abnormal event date: 21st Dec 2010
- Abnormal event period: about 45 minutes from 17:57 to 18:40
- Abnormal event category: broken down vehicle
- Abnormal event location: Western Avenue (point A in Figure 5.20)

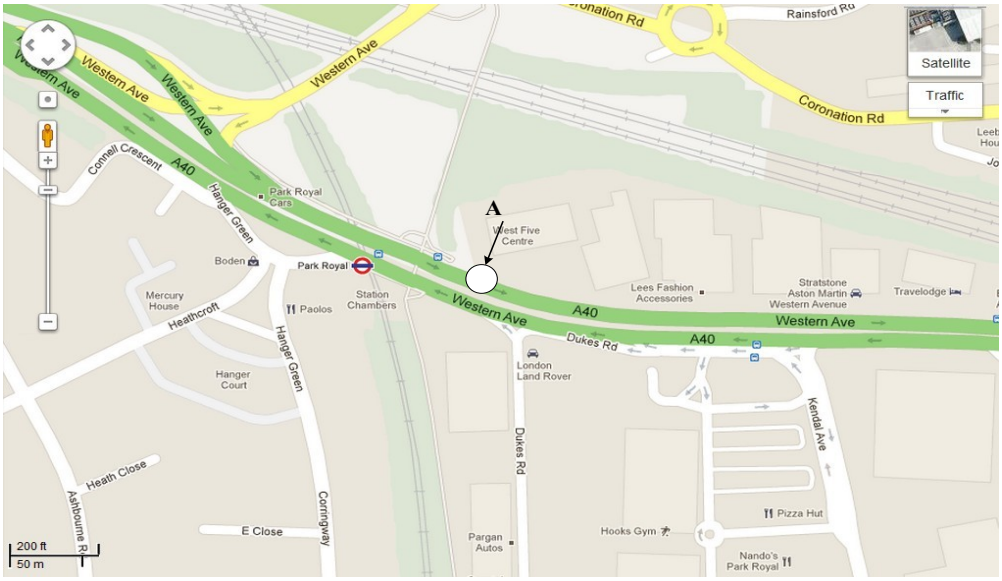


Figure 5.20: Location of the abnormal event on 21st December 2010 on link 1309 of the A40 road in central London (Source: Google Maps)

The results comparing the prediction accuracy of different frameworks with different machine learning methods during the abnormal period are shown in Table 5.4. Figure 5.21 gives the values of the MAPE metric and shows a decrease with the use of data smoothing and feedback structures. The average improvement is 21.5% in MAPE value for five different machine learning methods on the A40 road during the abnormal period.

The abnormal event happened around 17:57 and cleared around 18:40. The SSA data smoothing structure and feedback mechanism was found to improve the prediction accuracy during the abnormal event period. Moreover, the kNN based method could detect the drop in the traffic profile better than other methods and provided the best predictions. Figure 5.22 (a) shows the prediction results for the three frameworks using kNN; Figure 5.22 (b) shows the results using data during the incident period. Figures 5.23, 5.24 and 5.25 show the scatter-plot of predicted and observed travel time data, the error auto-correlation plot of predictions, the histogram

of error distribution and the time-series plot between predicted and observed travel time of the kNN method for each of the 1-stage, 2-stage and 3-stage prediction frameworks. It can be seen that the estimated bias for the 3-stage prediction framework is lower than that of 1-stage and 2-stage frameworks using the same kNN method. Hence, both the data smoothing structure and feedback mechanism can improve the prediction accuracy and reduce the prediction error in short-term traffic prediction under abnormal traffic conditions using the kNN method.

Table 5.4: Comparison of prediction accuracy of travel time from link 1309 on the A40 road using three different frameworks with five machine learning methods during the abnormal period

	MPE(%)			**MAPE(%)**			**RMSE(sec)**		
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	0.95	1.24	0.87	8.70	7.52	6.27	169.37	151.03	128.05
GM	0.14	-0.39	-0.25	9.31	7.83	7.55	180.40	170.73	163.63
NN	0.49	0.32	0.46	10.47	9.52	9.10	196.83	181.72	171.10
RF	-0.65	-0.78	-0.49	11.10	9.63	9.03	207.69	179.77	170.22
SVR	-0.23	-2.10	-1.81	11.06	8.06	7.79	211.50	154.68	152.98
Mean	0.14	-0.34	-0.24	10.13	8.51	7.95	193.16	167.59	157.20

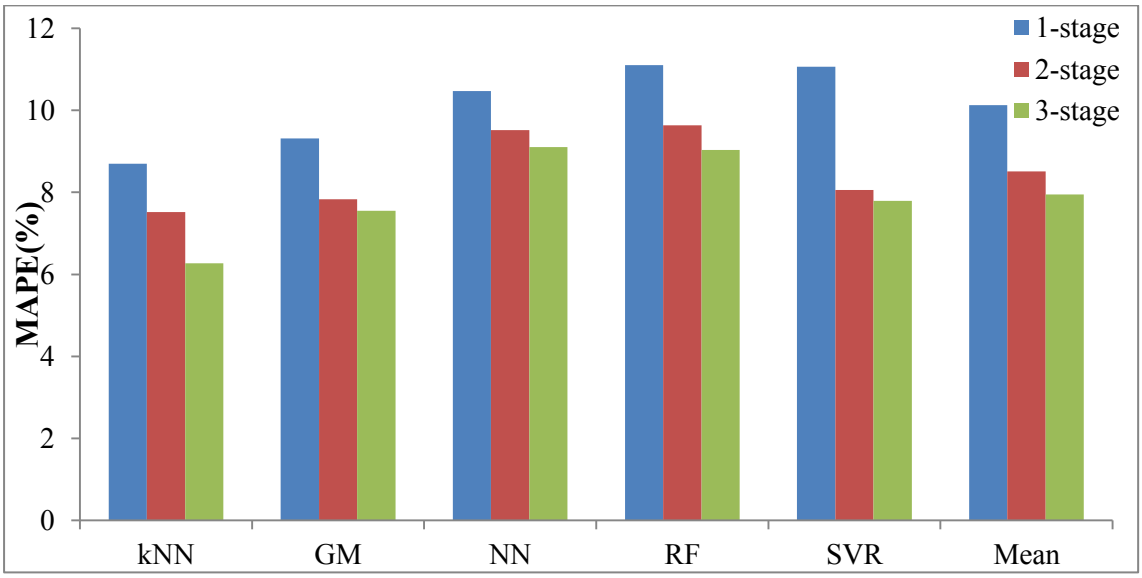
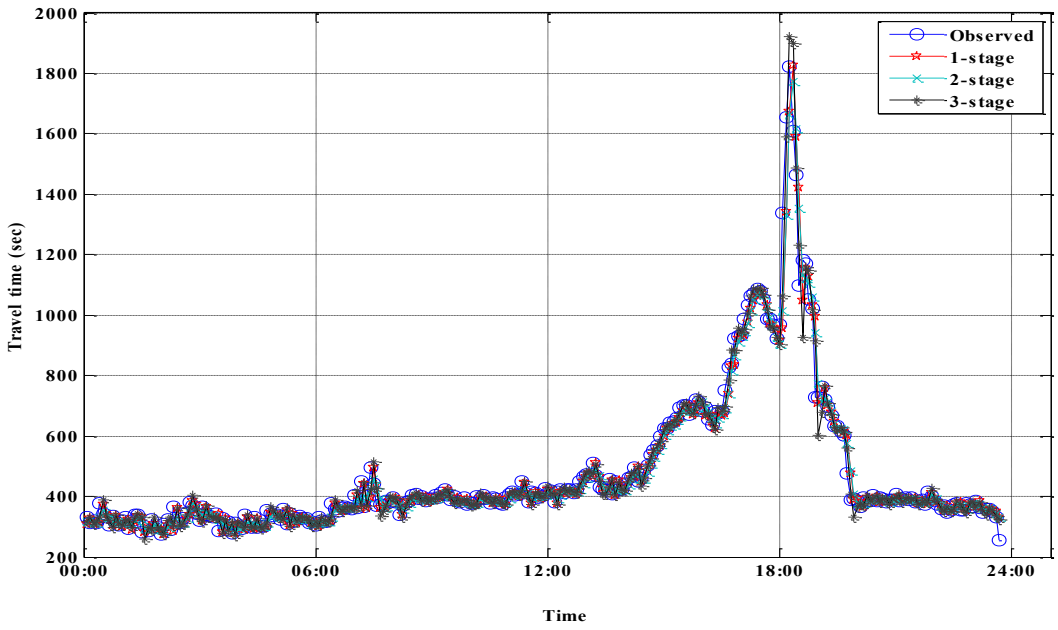
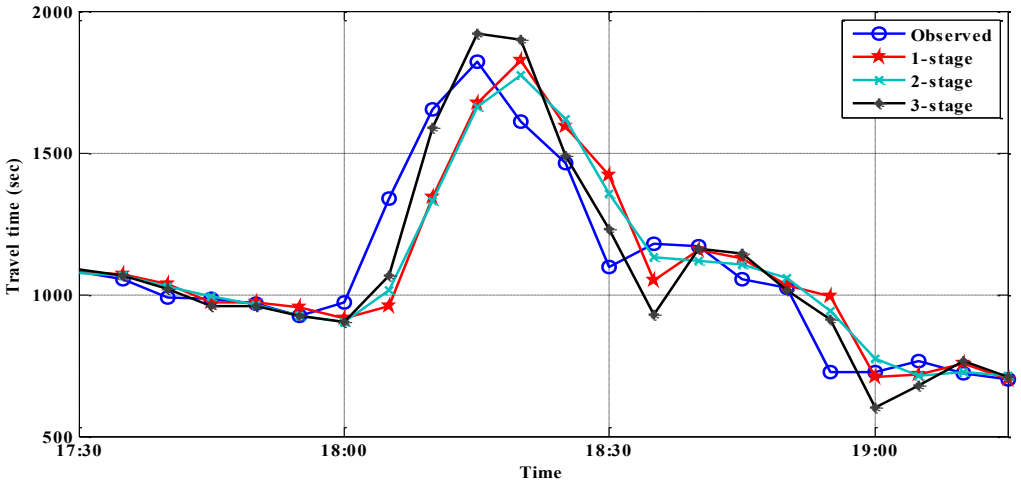


Figure 5.21: MAPE for five machine learning methods and three prediction frameworks during the abnormal period using data from link 1309 on the A40 road



(a)



(b)

Figure 5.22: Comparison of observed and predicted travel time using three prediction frameworks with the kNN based method (a) Prediction comparison during the day when the abnormal event occurred and (b) Prediction comparison during the abnormal period on the testing day

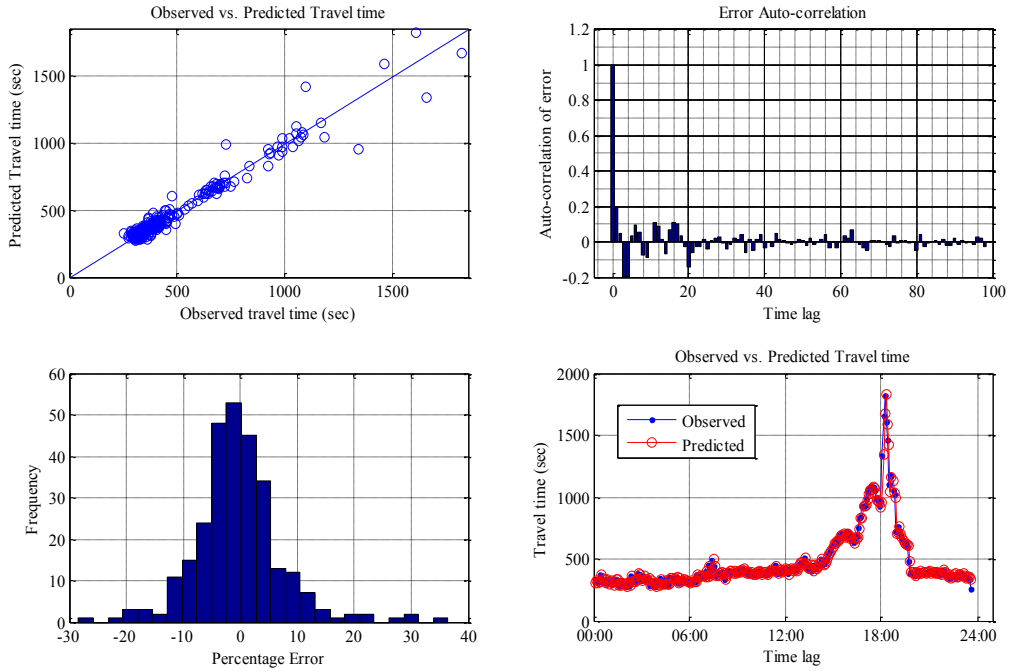


Figure 5.23: Travel time prediction performance using kNN with the 1-stage framework on the A40 road in London under abnormal traffic conditions

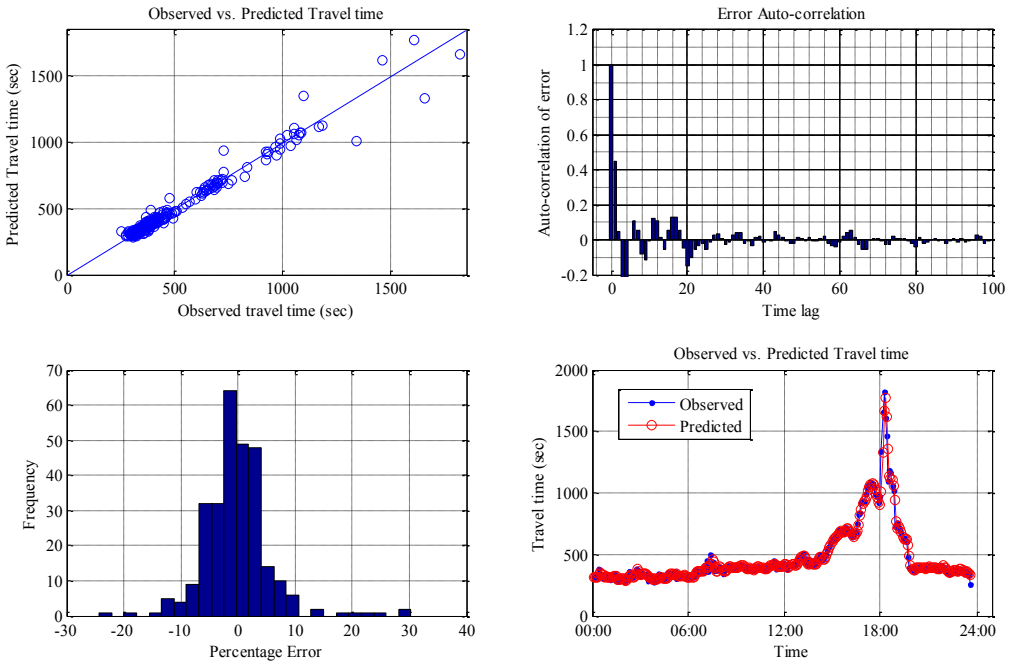


Figure 5.24: Travel time prediction performance using kNN with the 2-stage framework on the A40 road in London under abnormal traffic conditions

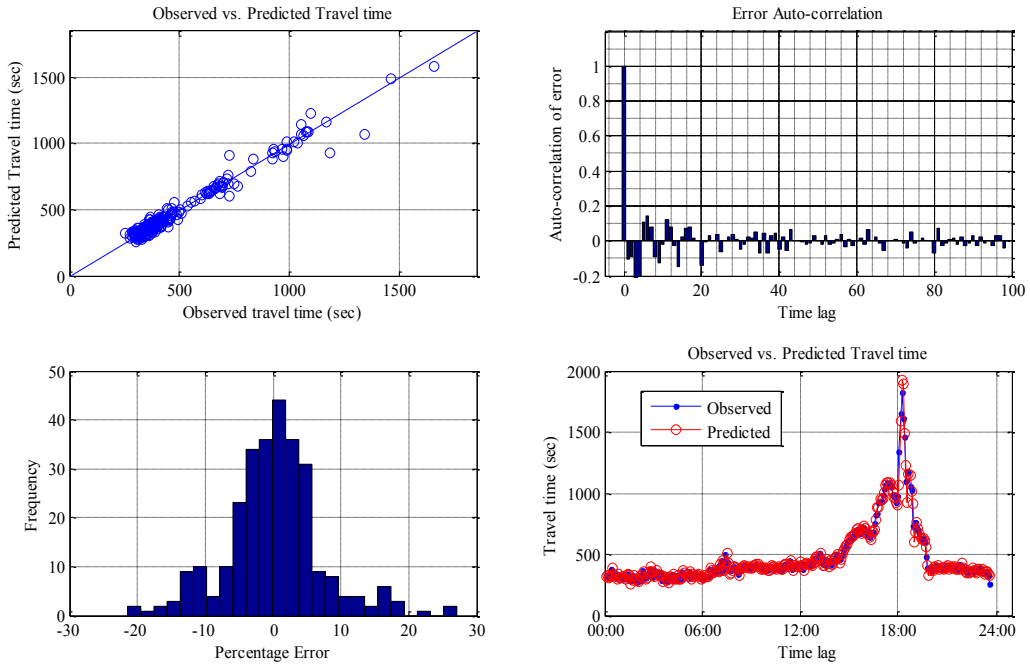


Figure 5.25: Travel time prediction performance using kNN with the 3-stage framework on the A40 road in London under abnormal traffic conditions

5.4.2 Short-term travel time prediction using data from Maidstone under abnormal traffic conditions

An accident happened on Link 99AL0005D in the Maidstone area of Kent on 26th August 2011, which caused the northbound A229 Royal Engineers Road to become partially blocked. This abnormal information was directly provided by KCC, including event date, start time, end time, category, location and severity. Figure 5.26 shows the location of the abnormal event. Detailed information of this abnormal event is as below:

- Abnormal event date: 26th Aug 2011
- Abnormal event period: about 30 minutes from 16:23 to 16:51
- Abnormal event category: accident
- Abnormal event location: A229 Royal Engineers Road, Maidstone area (point A in Figure 5.26)

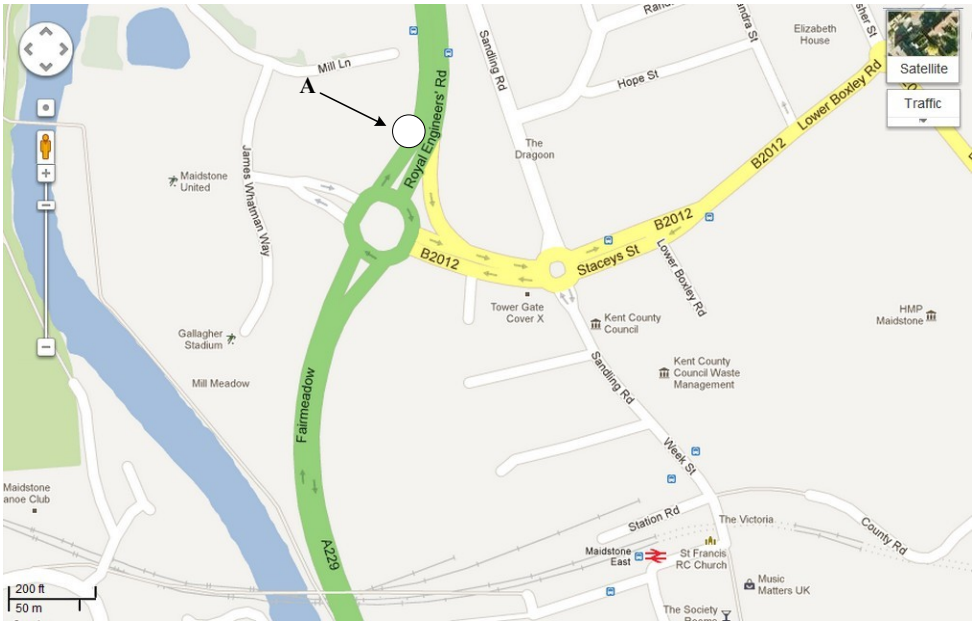


Figure 5.26: Location of abnormal event on 26th August 2011 on Link 99AL0005D in the Maidstone area of Kent (Source: Google Maps)

The training data was collected from 1st June 2011 to 25th August 2011. Only weekday travel time data was included in this dataset. Due to the accessibility of travel time data from Link 99AL0005D, traffic time data from the morning peak period (07:00-10:00) and evening peak period (16:00-19:00) was extracted in the training dataset.

Table 5.5 shows the prediction accuracy results for the three frameworks with five different machine learning methods during the abnormal period on 26th August 2011. The MAPE values of different combinations were given in Figure 5.27. The average improvement in MAPE value is 42.5% for five different machine learning methods using the 3-stage framework; while the improvement in the RMSE metric is 41.3%.

Table 5.5: Comparison of prediction accuracy of travel time from Link 99AL0005D in Maidstone using three different frameworks with five machine learning methods during the abnormal period

	MPE(%)			MAPE(%)			RMSE(sec)		
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-1.04	-0.57	-0.53	6.54	2.37	2.31	108.02	46.05	44.71
GM	-4.49	-4.26	-2.88	8.50	7.36	6.22	115.20	102.06	81.55
NN	2.05	-0.57	-0.21	6.13	3.72	3.21	92.03	58.83	50.22
RF	-0.81	0.57	-0.41	6.71	6.19	5.45	90.38	89.97	71.19
SVR	0.23	0.33	0.31	6.55	2.72	2.60	93.73	50.75	45.61
Mean	-0.81	-0.90	-0.74	6.89	4.47	3.96	99.87	69.53	58.66

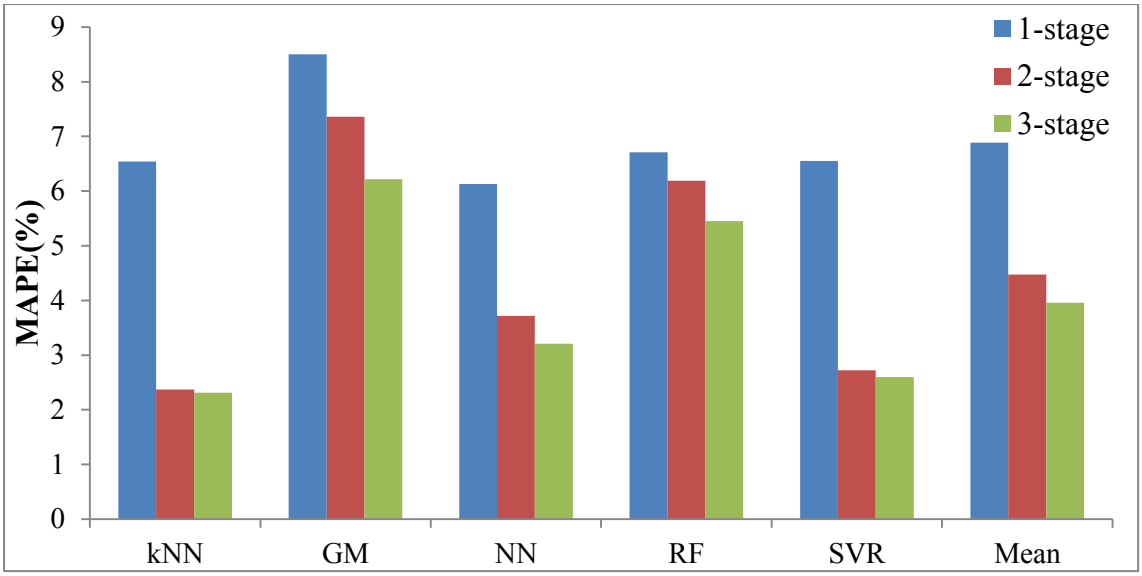


Figure 5.27: MAPE for five machine learning methods and three prediction frameworks during the abnormal period using data from Link 99AL0005D in Maidstone

Using the MAPE and RMSE metrics, the kNN based method is still the best for traffic prediction under abnormal traffic conditions. Figure 5.28 is the time-series plot between predicted and observed travel time of the kNN method for the 1-stage, 2-stage and 3-stage prediction frameworks during the abnormal period.

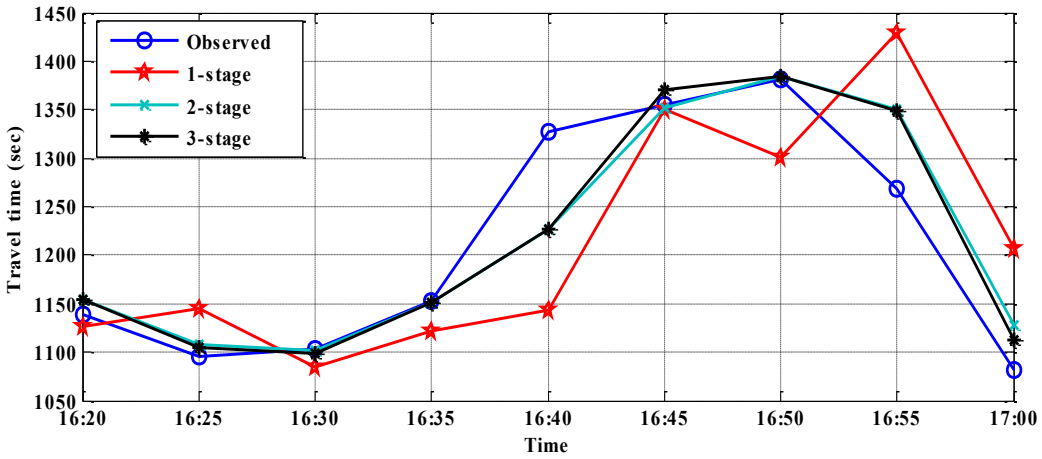


Figure 5.28: Comparison of observed and predicted travel time using three prediction frameworks with the kNN based method during the abnormal period

5.4.3 Short-term traffic flow prediction using data from London Marylebone corridor under abnormal traffic conditions

In this experiment, the training dataset was from April and May 2008. A severe traffic incident happened on the testing day, 20th June 2008, a Friday. The incident period was around 18:59 to 21:01 as per the records obtained from the BBC. The incident location was near the intersection of Macfarren Place and Marylebone Road (point A in Figure 5.4).

Figure 5.29 presents the comparison of traffic flow profiles between the testing day under abnormal traffic conditions (the solid blue line) and a normal weekday under normal traffic conditions (the solid black line). It is clear that traffic flow suddenly dropped on the testing day because of the occurrence of a severe traffic incident. Table 5.6 presents a comparison of prediction accuracy for the three frameworks and each machine learning method relating to the Marylebone Road corridor during the abnormal period, while Figure 5.30 depicts the MAPE scores. Prediction accuracy in terms of MPE, MAPE and RMSE increases when the data smoothing structure and feedback mechanism are applied regardless of the machine learning method used.

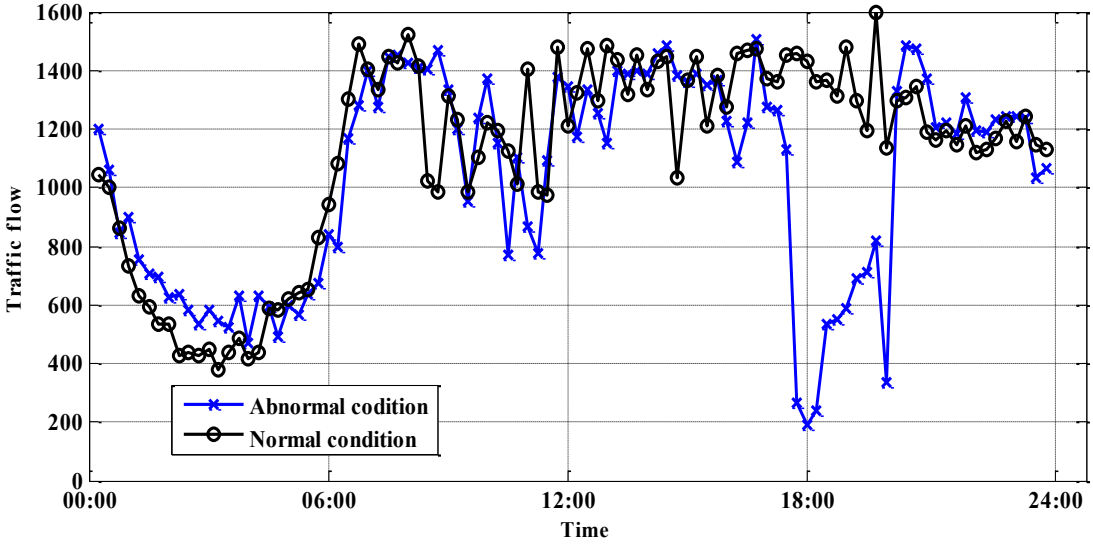


Figure 5.29: Time-series plot between the profiles under normal traffic conditions and abnormal traffic conditions

Table 5.6: Comparison of prediction accuracy of traffic flow on the Marylebone corridor using three different frameworks with five machine learning methods during the abnormal period

	MPE (%)			MAPE (%)			RMSE (vehicles/hour)		
	1-stage	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**	**1-stage**	**2-stage**	**3-stage**
kNN	-20.46	-26.31	-18.06	39.48	35.50	32.71	341.35	234.90	234.50
GM	-42.24	-39.65	-29.87	61.01	59.96	41.91	379.86	348.32	281.05
NN	-30.08	-17.67	-11.99	44.40	39.71	32.78	336.45	260.60	252.16
RF	-44.17	-32.81	-21.45	58.54	45.68	35.13	345.77	272.49	251.15
SVR	-72.85	-44.55	-31.78	87.29	52.05	39.29	483.49	278.05	251.27
Mean	-41.96	-32.20	-22.63	58.14	46.58	36.36	377.38	278.87	254.03

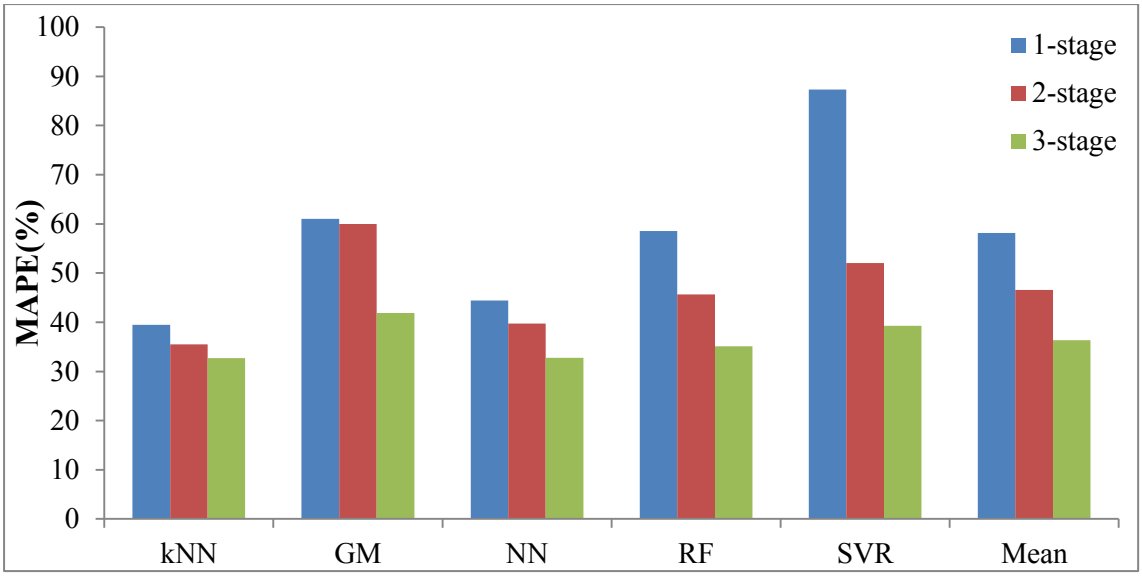


Figure 5.30: MAPE for five machine learning methods and three prediction frameworks during the abnormal period using data from the Marylebone corridor

Figure 5.31 is the time-series plot between predicted and observed traffic flow of the kNN method for the three different prediction frameworks during the abnormal period. Figure 5.32 presents the scatter-plot of predicted and observed traffic flow, the error auto-correlation plot of predictions, the histogram of error distribution and the time-series plot between predicted and observed traffic flow for the kNN method with the 3-stage prediction framework. The time-series plot shows the general overestimation of traffic flows during the beginning period of abnormal conditions and underestimation of traffic flows during the clearance period of abnormal conditions. The tendency during the abnormal periods is for a light lag behind the actual travel times. When the traffic profile suddenly drops, the models cannot immediately detect this change. More detailed information about abnormal events such as level of lane closure and severity of incidents and accidents may help models estimate the current traffic states and predict traffic variables more accurately during abnormal traffic conditions.

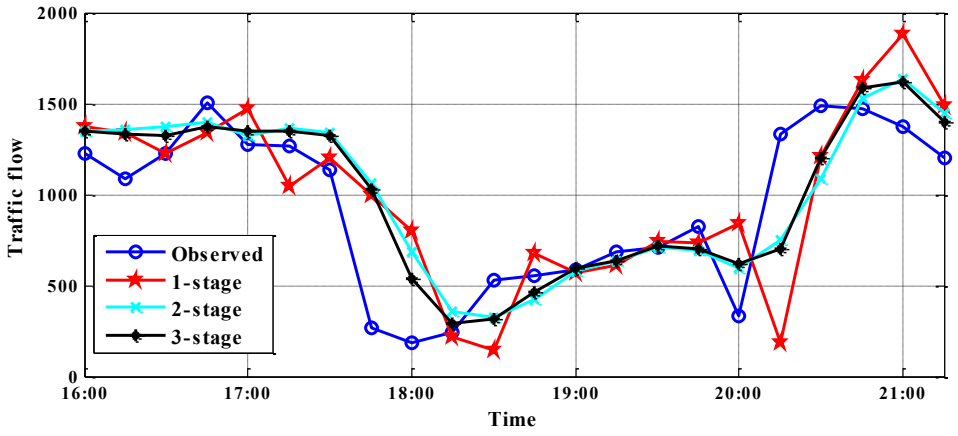


Figure 5.31: Comparison of observed and predicted traffic flow using three prediction frameworks with the kNN based method during the abnormal period

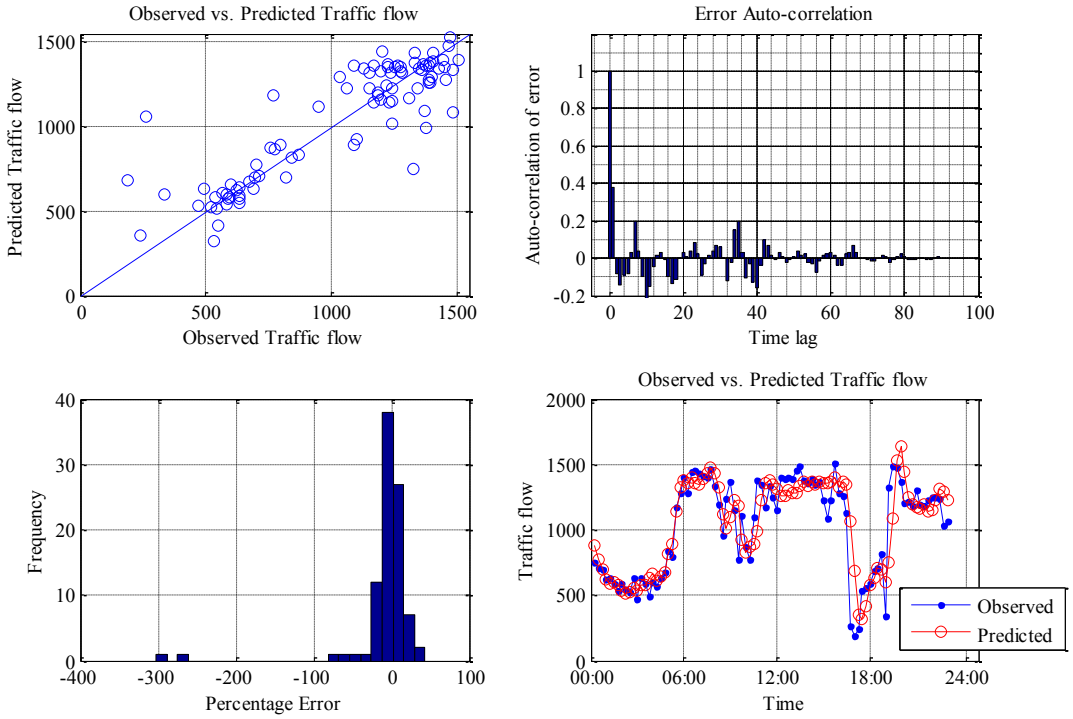


Figure 5.32: Traffic flow prediction performance using kNN with the 3-stage framework on the Marylebone corridor under abnormal traffic conditions

5.5 Conclusions

This chapter has presented the results of a series of experiments using real-world traffic data from London and Maidstone, which were undertaken to evaluate the performance of three prediction frameworks with five advanced machine learning methods.

The prediction results show that the data smoothing structure and feedback mechanism do improve the accuracy of prediction regardless of the machine learning method used, and do so under both normal and abnormal traffic conditions. Under normal traffic conditions, the structure of data smoothing can help machine learning methods significantly improve one-step ahead prediction accuracy. The feedback mechanism does not help much on one-step ahead prediction; however, this structure can significantly improve prediction accuracy for multi-step ahead prediction. Under abnormal traffic conditions, both data smoothing and feedback in general improve traffic prediction accuracy. The smoothing stage can help machine learning methods quickly detect a change in traffic patterns, while feedback information provides prediction errors from previous time intervals, which can significantly improve prediction accuracy under abnormal traffic conditions.

The chapter has evaluated the proposed prediction frameworks with five machine learning methods using both link travel time and traffic flow data. The results show that regardless of the traffic variables used as the input for a prediction model, the 2-stage and 3-stage frameworks perform better than the 1-stage framework without data smoothing and feedback structures under both normal and abnormal traffic conditions.

Regarding the issue of the machine learning method, the five methods used in this chapter have a similar level of prediction accuracy during normal conditions. The

kNN based method has the best ability to respond to a sudden change of traffic patterns caused by abnormal traffic events. It was observed that the kNN based method used in the context of the proposed 3-stage prediction framework resulted in the most accurate prediction among machine learning methods used in this research under abnormal traffic conditions. This is because lazy learning approaches such as kNN can quickly detect pattern changes and have the flexibility to match the best patterns from historical datasets.

Chapter 6 Conclusions and Future Research

This chapter summarises the findings of this research and discusses future research avenues based on these findings.

6.1 Revisiting the objectives

The aim of this research was to develop improved methods for the short-term prediction of traffic state variables on urban arterial roads under both normal and abnormal traffic conditions. The main objectives of this research as set out in Chapter 1, were as follows:

- Develop traffic prediction models to improve machine learning methods based on a more comprehensive prediction framework;
- Develop robust models to accurately predict traffic during both normal and abnormal traffic conditions on urban arterial roads;
- Develop traffic prediction models that can be easily implemented without laborious calibration and maintenance, and that have the quality of location transferability; and
- Develop methods to provide both one-step and multi-step ahead traffic prediction.

For the first objective, a systematic 2-stage traffic prediction framework, combining data smoothing and a machine learning prediction method was proposed in Section 3.2 and Section 3.3 to effectively learn trends in data and to enable more accurate prediction. The proposed framework was evaluated with five machine

learning methods using both simulated and real-world traffic data to investigate systematically the impact of the data smoothing technique. The results showed that the 2-stage prediction framework with data smoothing can improve the prediction accuracy regardless of the machine learning method used for all evaluated scenarios.

In Section 3.4, an error feedback mechanism was added to the 2-stage traffic prediction framework to create a 3-stage prediction framework. In the experiments undertaken, the 3-stage prediction framework significantly outperformed both the 1-stage framework without data smoothing and feedback structures, as well as the 2-stage prediction framework with data smoothing techniques for each of the five different machine learning methods using both simulated and real-world traffic data.

For the second objective, three simulation scenarios were designed to evaluate the performance of the proposed prediction framework in Chapter 4. The details of the simulation experiments and the process of data collection were presented in Sections 4.3-4.5. In the simulation experiments, traffic data was collected under a range of different traffic conditions from urban road networks. The results showed that traffic prediction accuracy improved when the proposed framework was applied, regardless of the machine learning method used under both normal and abnormal traffic conditions. Under normal traffic conditions, the average MAPE score of the five machine learning methods showed an improvement from 6.56% using the 1-stage framework to 4.53% using the 2-stage framework and 4.08% using the 3-stage framework; a total increase in accuracy of 38.7%. The average improvement in MAPE was 31.75% with the 3-stage framework under abnormal traffic conditions when one lane was blocked in simulation experiments.

In Chapter 5, the proposed prediction frameworks with machine learning methods were also evaluated using real-world traffic data. Under normal traffic conditions, the data smoothing and feedback structures were found to significantly improve prediction accuracy. Under abnormal traffic conditions, the average improvement of the five machine learning methods was 33.8% in MAPE value. In addition, the five machine learning methods have a similar level of prediction accuracy using the 3-stage framework; however, the kNN based method had the best ability to predict traffic variables when traffic patterns suddenly changed during abnormal traffic conditions. Therefore, overall, based on the evaluations carried out in this research, the kNN based 3-stage prediction framework has the best prediction accuracy under abnormal traffic conditions.

For the third objective, the proposed 3-stage prediction framework for short-term traffic prediction does not require an elaborate calibration process. Experimental results indicated that the proposed frameworks are transferable to different sites. The proposed frameworks were therefore tested using traffic flow data from the Russell Square corridor and Marylebone corridor in central London, as well as travel time data from London and Maidstone with minimal calibration effort.

For the fourth objective, the proposed models were tested for one-step ahead prediction under both normal and abnormal traffic conditions. The results showed that the data smoothing structure was an advantage for one-step ahead traffic prediction under both normal and abnormal traffic conditions. The feedback mechanism could also slightly improve one-step ahead traffic prediction accuracy under normal traffic conditions. Multi-step ahead traffic variables were predicted using the proposed frameworks under normal traffic conditions. The results showed that both data

smoothing and error feedback structures could indeed improve multi-step ahead prediction accuracy under normal and abnormal traffic conditions.

6.2 Contributions

This PhD research has made contributions to knowledge by developing a general framework to predict traffic variables under both normal and abnormal traffic conditions. The empirical results show that this framework outperforms a number of existing approaches. The main contributions of the research are summarised below.

Existing literature shows that a wide range of methods have been used for short-term traffic prediction. Most studies on short-term traffic prediction focus on one specific prediction method e.g. time series or machine learning methods. Recent work has demonstrated that model structure and error feedback mechanisms can play an important role in traffic prediction (e.g. Krishnan & Polak (2008); Guo *et al.* (2010)). More recently still, researchers have begun to investigate the effect of formal data smoothing and de-noising techniques to improve prediction accuracy (e.g. Simoes *et al.* (2011); Guo *et al.* (2013)). None of these studies, however, has attempted to combine the above aspects into a general framework. This research systematically investigated and generalised the impact of data smoothing techniques, model structure and error feedback structure on traffic prediction accuracy across a number of different machine learning methods. The results demonstrated that the proposed prediction framework can improve prediction accuracy regardless of the machine learning method used.

The majority of existing work has focused on short-term traffic prediction in normal traffic conditions. Few studies have developed prediction models to apply

under abnormal traffic conditions. This research, therefore, systematically tested and analysed the proposed prediction framework under a range of different traffic situations. The findings show that the best performing methods during abnormal traffic conditions are lazy learning based approaches, such as the kNN. This is because the lazy learning method does not require any explicit model construction for prediction and has the flexibility to match the best patterns from historical datasets. They are also not constrained by particular theoretical flow propagation or driver behaviour models, which might not apply well in abnormal conditions.

6.3 A note on practical implementation

This thesis has presented models for short-term traffic prediction on urban roads. Figure 6.1 shows the general flow-chart of prediction implementation. In the practical implementation of these models, both near-real-time and historical traffic variables such as link travel time and traffic flow from urban networks are required. These traffic variables need to be stored in a database. Near-real-time traffic data is the input of the prediction model. Historical data for a few weeks, preferably 3 months, is needed for practical implementation.

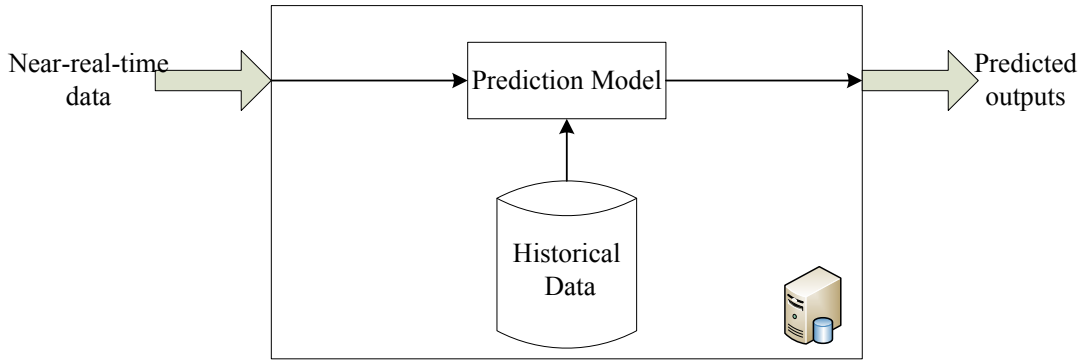


Figure 6.1: Summary of prediction implementation

When traffic patterns permanently change due to factors such as the introduction of a new bus route, introduction of road pricing or physical changes to the road network, the historic database will become out-of-date. Historical data under the new environment should be collected for a few weeks to create a new historic database before the proposed methods can be applied.

Most of the data described above are in principle available as a by-product of the operation of urban traffic control systems such as SCOOT or SCATS, which are deployed in many towns and cities globally. Therefore, the practical implementation of the methods described in this thesis is within the scope of many authorities.

6.4 Future research

A number of research avenues have been opened up based on this PhD thesis, as summarised below:

1) This research uses SSA as the data smoothing and de-noising method. Further research could evaluate the proposed traffic prediction framework in combination with machine learning methods using other data smoothing methods such as the wavelet transform (Xie *et al.*, 2007), Savitzky-Golay smoothing (Barclay *et al.*, 1997) and Fourier filtering (Kosarev & Pantos, 1983) to systematically investigate more fully the impact of data smoothing on short-term traffic prediction accuracy.
2) The literature review suggested that both temporal and spatial information on traffic networks have positive impacts on some short-term traffic prediction models. Travel time data from relevant links connected to the predicted link can be used as additional explanatory variables to increase the accuracy of

prediction models. However, the proposed frameworks do not make use of spatially-lagged information. It will be interesting to further develop the frameworks to include the spatial dimension.

3) This research focuses on traffic prediction using machine learning methods. Comparison should be made in further research between machine learning methods and well-calibrated simulation methods systematically to investigate their performance in short-term traffic prediction.
4) Five different machine learning methods (kNN, GM, NN, RF and SVR) were evaluated in this research. Under normal traffic conditions, these methods have quite similar prediction performance. Under abnormal traffic conditions, the prediction accuracy is dependent on both the prediction framework and machine learning method used. Different predictors have different characteristics and performance. Some may perform better in certain conditions than others. Hence, a combined predictor might be desirable. The MAPE result of averaging prediction using five different machine learning methods with a 3-stage framework using data from the A40 road in London under abnormal traffic conditions is 7.27%, which outperforms some of the individual machine learning methods. Further research can change the weights of the different methods to create an adaptive hybrid prediction method. In addition, other machine learning methods, such as Chan *et al.* (2012) and Li *et al.* (2013), can also be further tested with the proposed prediction frameworks under normal and abnormal traffic conditions.
5) The framework could be extended in the future to include exogenous variables that cause traffic abnormality, when weather, abnormal events information and signal plan information is available online.

a. *Weather information*: The prediction frameworks were tested under a range of different abnormal traffic conditions that can affect traffic profiles. Weather is another factor that may affect traffic profiles in the real world. When information on weather conditions is available online and accessible, it can be added to the prediction framework as an explanatory element.

b. *Incident feed*: This research tested the proposed prediction model during abnormal conditions. Information on live traffic disruptions including planned events is increasingly available online in a machine-read format. For example, a new feed called Live Traffic Disruptions (TIMS) run by TfL is available online (see http://www.tfl.gov.uk/) from 1 April 2013. TIMS can capture a richer range of information about road disruptions, including improved spatial information, details of closures and more in-depth categorisation of the cause of a disruption. A more detailed introduction to TIMS can be found in TfL (2013). This information can be used as an explanatory variable in further model development.

c. *Signal plan*: Signal control information may be included in traffic prediction models to improve the accuracy of prediction. The experiments in this research were undertaken within the context of an urban arterial road that is controlled by adaptive signal plans. Further research may consider the impact of signal control plans to improve the accuracy of prediction, when online information about signal control can be obtained.

Appendiex A Conceptual Impacts of Traffic Variables Caused by Abnormal Traffic Conditions

A.1 Basic queuing theory

This section introduces the queuing theory and its applications in traffic analysis during abnormal traffic conditions. Traffic stream characteristics and diagrams are used to explain the nature of traffic flows and reduced capacity impacts. The conceptual change of traffic states due to abnormal traffic conditions is described in this section.

Most abnormal traffic conditions are caused by planned events and unplanned events that result in link blockage or closure of lanes. During these abnormal traffic conditions, the available link capacity is lower than traffic demand (the amount of vehicles that intend to pass the link per unit time), congestion and queues will set in at the event location such as incident location (Knoop, 2009). The analysis of queues caused by abnormal traffic conditions is concerned with the fundamental of queuing theory. Hillier & Lieberman (2005) gave a concise definition of queuing theory that "*involves the mathematical study of queue that is a common phenomenon that occurs whenever the current demand for a service exceeds the current capacity to provide that service*". Queuing theory is used to analyse the theories of queuing behaviour in many fields such as telecommunication, computing and finance. Some academic studies used queuing theory to model traffic waiting lines, determine the network performances and analyse signalised intersection queuing problems (May, 1965; Daganzo, 1997; Baykal-Gürsoy *et al.*, 2009). This section uses queuing theory to

examine the characteristics of traffic variables during abnormal traffic conditions. Figure A.1 is a general queuing system.

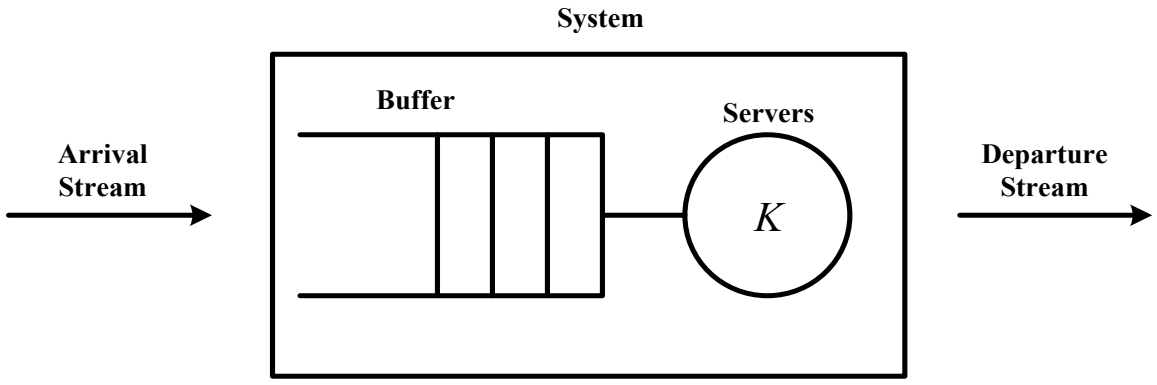


Figure A.1**:** A general queuing system

A universal notation of a queuing system introduced by Kendall (1953) is:

A/S/K/N/QD

where

A = Type of arrival-time distribution

M used for Poisson distribution

D used for Deterministic distribution

G used for General distribution

S = Type of service-time distribution

M used for Exponential distribution

D used for Deterministic distribution

G used for General distribution

K = Number of servers

N = System capacity (i.e. the amount of items in system when it is saturated) that can be infinite or finite

QD = Queue discipline

FIFO used for first-in-first-out (i.e. service in order of arrival)

SIRO used for service in random order

LIFO used for last-in-first-out

A more concise notation version is:

$$A/S/K$$

where it is assumed that the system capacity is infinite (i.e. $N = \infty$) and the queue discipline is FIFO (i.e. $QD = FIFO$).

A.2 Queuing theory in traffic modelling interrupted by abnormal conditions

The existing academic literature on modelling traffic flow interrupted by unplanned events such as incidents presented different queuing models (Baykal-Gürsoy *et al.*, 2009). Among this queuing models, deterministic arrivals and departures with single server (D/D/1) is the simplest queuing model (Martin *et al.*, 2011). As stated by Jain & Smith (1997), a road segment occupied by a stopped vehicle can be considered as a *server*. The service starts when an individual vehicle joins this link and ends when this vehicle passes the end of the link. In a general D/D/1 queuing model, there are three key input variables:

λ: normal vehicle rate that represents the traffic demand;

μ_d: vehicle departure rate during abnormal events that represents capacity during abnormal event;

μ: maximum vehicle departure rate that represents road capacity.

This relationship of the above variables is simply illustrated by a fundamental diagram in Figure A.2, and more details can refer to May (1990) and Martin *et al.* (2011). Table A.1 is the description of the variables used in Figure A.2. Traffic characteristics such as maximum queue length, average queue length, and total delay can be estimated using queuing theory (Qin & Smith, 2001). The estimated traffic characteristics under abnormal traffic conditions are given in Table A.2. In the real-world delay analysis to implement Traffic Incident Management, some actions will be taken after the incidents such as using Variable Message Signs (VMS), which will change the vehicle queuing diagram. A more complicated delay analysis of queuing theory during this circumstance can refer to Martin *et al.* (2011).

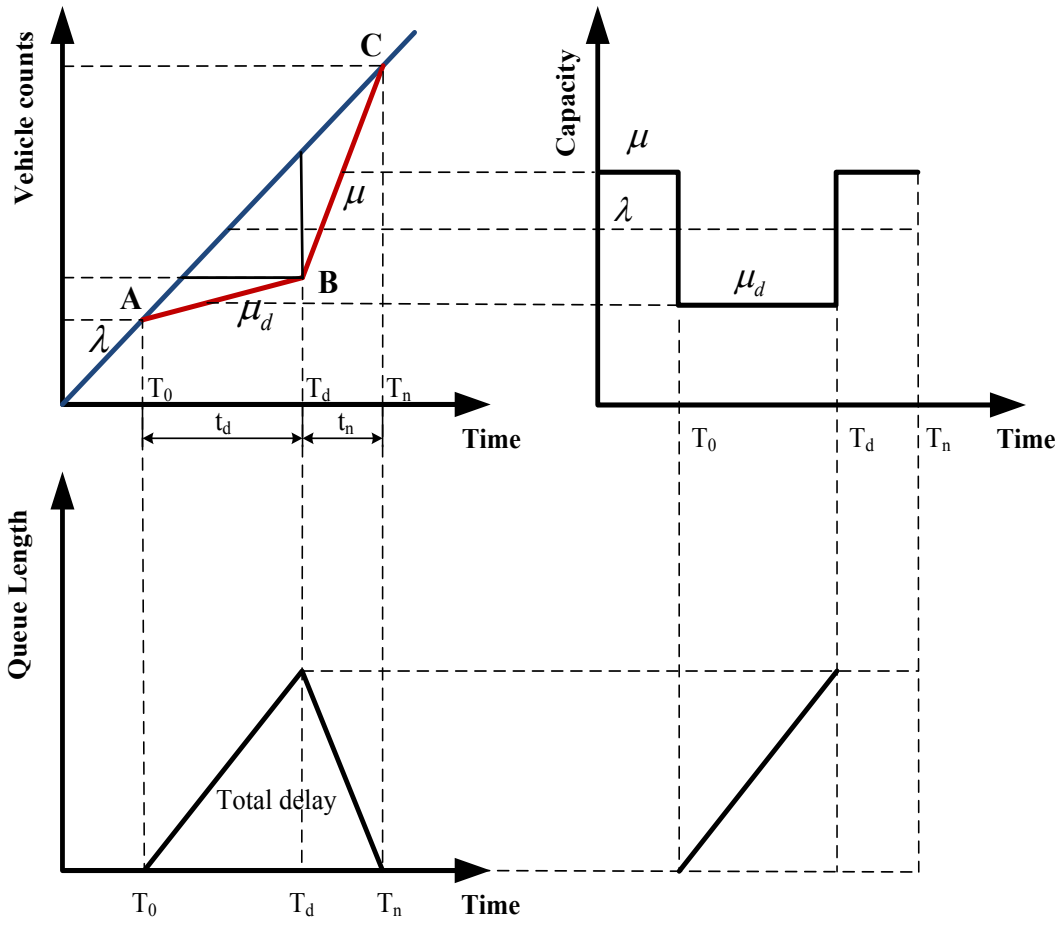


Figure A.2. Vehicle queuing-capacity-time diagram

Table A.1: Description of parameters in Figure A.2

Group	Parameters	Description
Points	A	Start point of an abnormal event;
	B	End point of the event
	C	Moment when traffic state is back to restoration
Times	t_d	Duration of abnormal event
	t_n	Duration from abnormal traffic condition to normal condition
	T_0	Abnormal event stat time
	T_d	Abnormal event end time
	T_n	Time of flow restoration
Slope	AC	Depends on traffic volumes
	AB	Depend on the level of lane closure
	BC	Depend on the number of lanes (mainline capacity)
Area	ABC	Total delay

Table A.2: Estimated traffic characteristics using queuing theory (Source: Qin & Smith (2001))

Estimated traffic characteristics	Equation
t_q: Time duration in queue (hour)	$t_d + t_n$
N_q: Number of vehicles queued(veh)	λt_q
Q_{max}: Maximum queue length(veh)	$t_d(\lambda - \mu_d)$
Q_m: Average queue length(veh)	$Q_{max}/2$
D: Total delay (veh)	$t_q Q_m$

Appendiex B Traffic Data Cleaning Methods

B.1 LCAP data cleaning methods

The most common invalid individual records might be caused by number plate recognition errors or by vehicles stopping en-route. The following removal strategy has been developed to remove the outliers:

1) Vehicles with journey times below a minimum threshold, i.e. travelling at excessive/unrealistic speed (presently 100 km/hour);
2) Vehicles that have overtaken more than *x* (presently 6) of the previous 10 vehicles, i.e. travelling excessively fast relative to the rest of the traffic;
3) Vehicles that have been overtaken by any of the remaining 10 following vehicles by more than *x* (presently 40 seconds/km) seconds, i.e. those travelling excessively slowly (or taking a detour). Time rather than number of vehicles is used in order to account for situations where capture rate is low, which isn't such an issue for the preceding rule;
4) Vehicles when counts are low (at present the next but one vehicle didn't arrive for 4 minutes), which are slower than all but *x* (presently 2) of the 5 vehicles both sides of them, and travelling below a threshold speed (presently 50 km/hour). This is designed to remove excessively slow journey times which are not captured by the overtaking rule due to low counts;
5) Data points (binned in 5 min intervals) which are greater/smaller than the both of the two data points on either side by more than *x* (presently 240) seconds. NB This is designed to remove excessively slow/fast journey times which are not captured by the previous rules. It primarily affects the early hours when

flows are low and journey times should be quick, but can also remove excessively fast journey times when there is congestion.

Another serious problem of raw link travel time data in LCAP system is missing data caused by no vehicles passing through the camera sites during some time periods such as midnight or camera failure. The strategy used to patch missing travel time data in LCAP system is based on the number of missing time intervals in succession given in Table B.1.

Table B.1: Methods to patch missing data in LCAP

NO. of missing time interval	**Patching method**
1	Average of observations in previous and next time intervals
2-6	Interpolated from observations in adjacent time intervals
>6	Replaced with historical average data of every time interval

B.2 ANPR data cleaning methods used in Maidstone

Data cleaning methods used by Kent County Council (KCC) remove invalid traffic data caused by device failure and vehicles stopping en-route. The main data cleaning strategies are described as follows:

1) Extracting: the travel times are based on arrivals at the downstream end of the link, and the raw data (observations of number plates) is taken from the camera feeds. The travel times for individual vehicles are determined by the time difference of the matched number plates recorded at upstream and downstream sites of the ANPR link.
2) Coarse cleaning: This process eliminates the matched travel time which would imply exceedingly long journeys (over 30 minutes) and those where the downstream readings preceded the matched upstream one.
3) Fine cleaning: After the coarse cleaning, each measured travel time then undergoes a validation process based on the average of the travel times for the five vehicles before and after the target vehicle arriving at the end point of the link. The target vehicle being validated is determined as invalid and discarded if its travel time exceeds twice that the average.

Appendiex C Main Traffic Modelling in AIMSUN

C.1 Car-following model

The car-following model in AIMSUN is based on the Gipps model (Gipps, 1981). It basically consists of two components, acceleration and deceleration. The first represents the intention of a vehicle to achieve a certain desired speed, while the second reproduces the limitations imposed by the preceding vehicle when trying to drive at the desired speed. The maximum speed of a vehicle *n* can accelerate during a time period $(t, t+T)$ is

$$V_a(n, t+T) = V(n,t) + 2.5a(n)T * (1 - \frac{V(n,t)}{V^*(n)})\sqrt{0.025 + \frac{V(n,t)}{V^*(n)}} \quad \text{(A.1)}$$

where

$V(n,t)$: the speed of vehicle *n* at time *t*

$a(n)$: the maximum acceleration of vehicle *n*

T: the reaction time (equal to simulation step)

$V^*(n)$: the desired speed of vehicle *n* for the current section

The limitation is

$$V_b(n, t+T) = d(n)T + \sqrt{\begin{matrix} d(n)^2T^2 - d(n)(2\{x(n-1,t) - s(n-1) - x(n,t)\} \\ -V(n,t)T - \frac{V(n-1,t)^2}{d'(n-1)}) \end{matrix}} \quad \text{(A.2)}$$

where

$d(n)$: the maximum deceleration desired by vehicle *n*

$x(n,t)$: the position of vehicle *n* at time *t*

$x(n-1,t)$: the position of preceding vehicle *n-1* at time *t*

$s(n)$: the effective length of vehicle *n*

$d'(n)$: an estimation of the nth vehicle's desired deceleration

The definitive speed of vehicle n during time interval $(t, t+T)$ is the minimum of those previously defined speeds: $V(n,t+T) = min\{V_a(n,t+T), V_b(n,t+T)\}$. Further details of car-following model in AIMSUN can be found in TSS (2004).

C.2 Lane changing model

The lane-changing model can also be considered as a development of the Gipps lane-changing model (Gipps, 1986). Lane change is modelled as a decision process, analysing the necessity of the lane change (such as for turning manoeuvres determined by the route), the desirability of the lane change (to reach the desired speed when the leader vehicle is slower, for example), and the feasibility conditions for the lane change that are also local, depending on the location of the vehicle in the road network.

In order to achieve a more accurate representation of the driver's behaviour in the lane-changing decision process, three different zones inside a section are considered, each one corresponding to a different lane changing motivation. These zones are characterised by the distance up to the end of the section, i.e., the next point of turning (see Figure C.1).

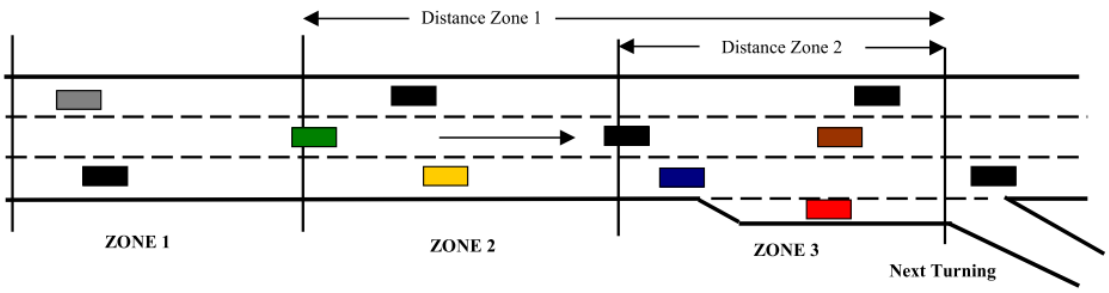


Figure C.1: Lane changing zones

1) Zone 1: This is the farthest distance from the next turning point. The lane-changing decisions are mainly governed by the traffic conditions of the lanes involved. The feasibility of the next desired turning movement is not yet taken into account. To measure the improvement that the driver will get from changing lanes, we consider several parameters: desired speed of driver, speed and distance of current preceding vehicle, speed and distance of future preceding vehicle.
2) Zone 2: This is the intermediate zone. It is mainly the desired turning lane that affects the lane-changing decision. Vehicles not driving in valid lanes (i.e. lanes where the desired turning movement can be made) tend to get closer to the correct side of the road from which the turn is allowed. Vehicles look for a gap may try to adapt to it, but do not affect the behaviour of vehicles in the adjacent lanes.
3) Zone 3: This is the shortest distance to the next turning point. Vehicles are forced to reach their desired turning lanes, reducing speed if necessary, and even coming to a complete stop in order to make the change possible. Also, vehicles in the adjacent lane can modify their behaviour in order to provide a gap big enough for the vehicle to succeed in changing lanes.

Further details of lane changing model in AIMSUN can be found in TSS (2004).

C.3 Gap Acceptance Model

In order to answer the question *"Is it possible to change lanes?"* the algorithm shown in Table C.1 is applied in AIMSUN to check whether a gap is acceptable or not.

Table C.1: Algorithm used in gap acceptance model (Source: TSS (2004))

Get downstream and upstream vehicles in target lane
Calculate gap between downstream and upstream vehicles: *TargetGap*
if ((TargetGap > VehicleLengh) & (it is aligned)) **then**
    Calculate the distance between vehicle and downstream vehicle in target lane: *DistanceDown*
    Calculate the speed imposed by downstream vehicle to vehicle, according to Gipps Car-following Model: *ImposedDownSpeed*
if (ImposedUpSpeed is acceptable for upstream vehicle, according to the deceleration rate) **then**
    Calculate the distance between upstream vehicle in target lane and vehicle: *DistanceUp*
    Calculate the speed imposed by vehicle to upstream vehicle, according to Gipps
    Car-following Model: *ImposedUpSpeed*
if (ImposedDownSpeed is acceptable for vehicle, according to the deceleration rate) **then**
    Lane Change is Feasible
    CarryOutLaneChange
else
    The gap is not acceptable because of the upstream vehicle
endif
else
The gap is not acceptable because of the downstream vehicle
endif
else
There is no gap aligned with the vehicle
endif

References

Abbas, M., Chaudhary, N. A., Pesti, G. & Sharma, A. (2005) *Guidelines for determination of optimal traffic responsive plan selection control parameters*. Texas Transportation Institute, The Texas A&M University System College Station, Texas, Report number: FHWA/TX-05/0-4421-2.

Abdulhai, B., Porwal, H. & Recker, W. (1999) Short-term freeway traffic flow prediction using genetically optimized time-delay-based neural networks. In: *Proceedings of the 87th Annual Meeting of the Transportation Research Board* Washington D.C., USA.

Abdulhai, B., Porwal, H. & Recker, W. (2002) Short-term traffic flow prediction using neuro-genetic algorithms. *ITS Journal,* 7 (1), 3-41.

Abu-Mostafa, Y. S. & Atiya, A. F. (1996) Introduction to financial forecasting. *Applied Intelligence,* 6 (3), 205-213.

Adams, J. C., Brainerd, W. S., Hendrickson, R. A., Maine, R. E., Martin, J. T. & Smith, B. T. (2008) *The Fortran 2003 Handbook: the Complete Syntax, Features and Procedures.* Springer, ISBN: 1846283787.

Ahmed, M. & Cook, A. (1979) Analysis of freeway traffic time series data by using Box-Jenkins techniques. *Transportation Research Board,* 722, 1-9.

Al-Anazi, A. & Gates, I. (2010) Support vector regression for porosity prediction in a heterogeneous reservoir: A comparative study. *Computers & geosciences,* 36 (12), 1494-1503.

Algers, S., Bernauer, E., Boero, M., Breheret, L., Di Taranto, C., Dougherty, M., Fox, K. & Gabard, J. F. (1997) *Review of micro-simulation models*. Institute for Transport Studies, University of Leeds, Report number: RO-97-SC.

Amemiya, T. (1985) *Advanced Econometrics.* Harvard University Press, ISBN 0-674-00560-0.

Antoniou, C., Koutsopoulos, H. N. & Yannis, G. (2007) An efficient non-linear Kalman filtering algorithm using simultaneous perturbation and applications in traffic estimation and prediction. In: *Proceedings of the 13th International IEEE Annual Conference on Intelligent Transportation Systems*, Seattle, USA. 217-222.

Barcelo, J. & Casas, J. (2004) Methodological notes on the calibration and validation of microscopic traffic simulation models. In: *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington D.C., USA.

Barcelo, J., Codina, E., Casas, J., Ferrer, J. & Garcia, D. (2005) Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems. *Journal of Intelligent & Robotic Systems,* 41 (2), 173-203.

Barceló, J., Ferrer, J., Casas, J., Montero, L. & Perarnau, J. (2002) Microscopic simulation with AIMSUN for the assessment of incident management strategies. In: *e-safety Congress and Exhibition, 2002, Lyon, France*.

Barclay, V., Bonner, R. & Hamilton, I. (1997) Application of wavelet transforms to experimental spectra: smoothing, denoising, and data set compression. *Analytical Chemistry,* 69 (1), 78-90.

Baykal-Gürsoy, M., Xiao, W. & Ozbay, K. (2009) Modeling traffic flow interrupted by incidents. *European Journal of Operational Research,* 195 (1), 127-138.

Beale, M., Hagan, M. & Demuth, H. (2012) *Neural Network Toolbox For Use with MATLAB User's Guide Version 7.* Natick, The MathWorks.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. & Mishalani, R. (1998) DynaMIT: a simulation-based system for traffic prediction and guidance generation. *TRISTAN III, San Juan, Porto Rico*.

Bollerslev, T. & Domowitz, I. (1993) Trading patterns and prices in the interbank foreign exchange market. *Journal of Finance,* 48 (4), 1421-1443.

Box, G. E. P. & Jenkins, G. M. (1970) *Time Series Analysis: Forecasting and Control.* San Francisco, Holden-Day.

Breiman, L. (2001) Random forests. *Machine learning,* 45 (1), 5-32.

Breiman, L. & Cutler, A. (2005) *Random Forests*. [Online]. Berkeley. Available from: http://www.stat.berkeley.edu/users/breiman/RandomForests/cc_manual.htm [Accessed 09/20/2012].

Broomhead, D. & King, G. P. (1986) Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena,* 20 (2-3), 217-236.

Cao, L. J. & Tay, F. E. H. (2003) Support vector machine with adaptive parameters in financial time series forecasting. *Neural Networks, IEEE Transactions on,* 14 (6), 1506-1518.

Castro-Neto, M., Jeong, Y. S., Jeong, M. K. & Han, L. D. (2009) Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications,* 36 (3), 6164-6173.

Chan, K. Y., Dillon, T. S., Singh, J. & Chang, E. (2012) Neural-network-based models for short-term traffic flow forecasting using a hybrid Exponential smoothing and Levenberg–Marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems,* 13 (2), 644-654.

Chang, B. R. & Tsai, H. F. (2008) Forecast approach using neural network adaptation to support vector regression grey model and generalized auto-regressive conditional heteroscedasticity. *Expert Systems with Applications,* 34 (2), 925-934.

Chapelle, O. & Vapnik, V. (1999) Model selection for support vector machines. *Advances in Neural Information Processing Systems,* 12, 230-236.

Charytoniuk, W., Chen, M. & Van Olinda, P. (1998) Nonparametric regression based short-term load forecasting. *IEEE Transactions on Power Systems,* 13 (3), 725-730.

Cheese, J. J., Cartwright, M., Routledge, I. W. & Radia, B. (1998) UTMC - the UK initiative for ITS. In: *9th International Conference on Road Transport Information and Control, 21-23 April 1998*, London, UK.

Chen, C., Kwon, J., Rice, J., Skabardonis, A. & Varaiya, P. (2003) Detecting errors and imputing missing data for single-loop surveillance systems. In: *Proceedings of the 82nd Annual Meeting of the Transportation Research Board*, Washington D.C,USA.

Cheng, T., Haworth, J. & Wang, J. (2011) Spatio-temporal autocorrelation of road network data. *Journal of Geographical Systems,* 14 (4), 1-25.

Cherkassky, V. & Ma, Y. (2004) Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks,* 17 (1), 113-126.

Cherkassky, V. & Mulier, F. M. (2007) *Learning from Data: Concepts, Theory, and Methods.* Wiley-Blackwel l Press, ISBN: 0471681822.

Chien, S. I. J. & Kuchipudi, C. M. (2003) Dynamic travel time prediction with real-time and historic data. *Journal of Transportation Engineering-ASCE,* 129 (6), 608-616.

Clark, S. (2003) Traffic prediction using multivariate nonparametric regression. *Journal of Transportation Engineering-ASCE,* 129 (2), 161-168.

Daganzo, C. F. (1997) *Fundamentals of transportation and traffic operations.* Pergamon, ISBN: 0080427855.

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

De Lurgio, S. A. (1998) *Forecasting Principles and Applications.* New York, Irwin/McGraw Hill,Inc, ISBN: 0256134332.

Deng, J. L. (1982) Control problems of grey systems. *Systems & Control Letters,* 1 (5), 288-294.

Deng, J. L. (1989) Introduction to grey system theory. *The Journal of Grey System,* 1 (1), 1-24.

Devijver, P. A. & Kittler, J. (1982) *Pattern Recognition: A statistical approach.* Prentice Hall International, ISBN 0136542360.

DfT (1999) *The "SCOOT" urban traffic control system*. [Online]. Available from: http://www.ukroads.org/webfiles/tal07-99.pdf [Accessed 27/02/2013].

Dia, H. & Cottman, N. (2006) Evaluation of arterial incident management impacts using traffic simulation. *IEE Proceedings Intelligent Transport Systems,* 153 (3), 242-252.

Dougherty, M. (1995) A review of neural networks applied to transport. *Transportation Research Part C: Emerging Technologies,* 3 (4), 247-260.

Dougherty, M. S. & Cobbett, M. R. (1997) Short-term inter-urban traffic forecasts using neural networks. *International Journal of Forecasting,* 13 (1), 21-31.

Drew, D. R. (1968) *Traffic Flow Theory And Control.* New York, McGraw-Hill.

Duan, Q., Sorooshian, S. & Gupta, V. K. (1994) Optimal use of the SCE-UA global optimization method for calibrating watershed models. *Journal of Hydrology,* 158 (3), 265-284.

El Faouzi, N. E. (1996) Nonparametric traffic flow prediction using kernel estimator. In: *Proceedings of Internaional symposium on transportation and traffic theory*. 41-54.

Espinoza, M., Suykens, J. A. K., Belmans, R. & De Moor, B. (2007) Electric load forecasting. *IEEE Control Systems Magazine,* 27 (5), 43-57.

Faragher, R. (2012) Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation. *IEEE Signal Processing Magazine,* 29 (5), 128-132.

FHWA (1973) *Urban traffic control system and bus priority system traffic adaptive network signal timing program: software description*. Federal Highway Administration, US. Dept. of Transportation, Washington D.C.

Florio, L. & Mussone, L. (1996) Neural-network models for classification and forecasting of freeway traffic flow stability. *Control Engineering Practice,* 4 (2), 153-164.

French, M. N., Krajewski, W. F. & Cuykendall, R. R. (1992) Rainfall forecasting in space and time using a neural network. *Journal of Hydrology,* 137 (1-4), 1-31.

Genuer, R., Poggi, J.-M. & Tuleau-Malot, C. (2010) Variable selection using random forests. *Pattern Recognition Letters,* 31 (14), 2225-2236.

Ghil, M. & Vautard, R. (1991) Interdecadal oscillations and the warming trend in global temperature time series. *Nature,* 350 (6316), 324-327.

Ghosh, B., Basu, B. & O'Mahony, M. (2007) Bayesian time-series model for short-term traffic flow forecasting. *Journal of Transportation Engineering-ASCE,* 133 (3), 180-189.

Gipps, P. (1986) A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological,* 20 (5), 403-414.

Gipps, P. G. (1981) A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological,* 15 (2), 105-111.

Golyandina, N., Nekrutkin, V. & Zhigljavsky, A. (2001) *Analysis Of Time Series Structure: SSA And Related Techniques.* Chapman & Hall/CRC Press, ISBN: 1584881941.

Golyandina, N. & Zhigljavsky, A. (2013) *Singular Spectrum Analysis for Time Series.* Springer, New York, ISBN: 3642349129.

Gunn, S. R. (1998) *Support vector machines for classification and regression*. Technical report. University of Southampton.

Guo, F., Krishnan, R. & Polak, J. (2012a) A computationally efficient 2-stage method for short-term traffic prediction on urban roads. In: *44th Annual Universities Transport Studies Group (UTSG) Conference*, Aberdeen, UK.

Guo, F., Krishnan, R. & Polak, J. (2012b) Short-term traffic prediction under normal and abnormal traffic conditions on urban roads. In: *Proceedings of the 91st Annual Meeting of the Transportation Research Board*, Washington D.C, USA.

Guo, F., Krishnan, R. & Polak, J. (2012c) Short-term traffic prediction under normal and incident conditions using singular spectrum analysis and the k-nearest neighbour method. In: *Proceedings of the 17th International Conference on Road Transport Information and Control (RTIC)*, London, UK.

Guo, F., Krishnan, R. & Polak, J. (2013) A computationally efficient two-stage method for short-term traffic prediction on urban roads. *Transportation Planning and Technology,* 36 (1), 62-75.

Guo, F., Polak, J. & Krishnan, R. (2010) Comparison of modelling approaches for short term traffic prediction under normal and abnormal conditions. In: *Proceedings of the 13th International IEEE Annual Conference on Intelligent Transportation Systems*, Madeira Island, Portugal.

Haas, C. P. (2001) Assessing developments using AIMSUN. In: *Institution of Professional Engineers New Zealand Annual Conference*, Auckland, New Zealand.

Hadi, M., Sinha, P. & Wang, A. (2007) Modeling reductions in freeway capacity due to incidents in microscopic simulation models. *Transportation Research Record,* 1999, 62-68.

Hamed, M. M., Almasaeid, H. R. & Said, Z. M. B. (1995) Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering-ASCE,* 121 (3), 249-254.

Han, J. (2012) *Multi-sensor data fusion for travel time estimation*. PhD Thesis. Centre for Transport Studies, Imperial College London.

Hassan, M. R., Nath, B. & Kirley, M. (2007) A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Systems with Applications,* 33 (1), 171-180.

Hassani, H. (2007) Singular spectrum analysis: methodology and comparison. *Journal of Data Science,* 5 (2), 239-257.

Hastie, T., Tibshirani, R. & Friedman, J. H. (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York, Springer Verlag, ISBN: 0387952845.

Hastie, T., Tibshirani, R. & Friedman, J. H. (2008) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 2nd edition. New York, Springer Verlag, ISBN: 0387848576.

Hidas, P. (2002) Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies,* 10 (5–6), 351-371.

Hillier, F. S. & Lieberman, G. J. (2005) *Introduction to Operations Research.* 8th Edition. McGraw-Hill Higher Education, ISBN: 007123828X.

Hong, W.-C. (2008) Rainfall forecasting by technological machine learning models. *Applied Mathematics and Computation,* 200 (1), 41-57.

Hounsell, N. & McLeod, F. (1990) *ASTRID: Automatic SCOOT Traffic Information Database*. Technical Report Contractor Report, Transport and Road Research Laboratory, Department of Transport, Report number: 0266-7045.

Hourdakis, J., Michalopoulos, P. G. & Kottommannil, J. (2003) Practical procedure for calibrating microscopic traffic simulation models. *Transportation Research Record,* 1852 (1), 130-139.

Hu, J. (2011) *Short-term congestion prediction for vehicle navigation*. PhD Thesis. Centre for Transport Studies, Imperial College London.

Hu, J., Krishnan, R. & Bell, M. G. H. (2008) TPEG feed from the BBC: A potential source of ITS data? In: *Proceedings of the 13th International Conference on Road Transport Information and Control*, Manchester, UK. IET.

Huang, B. & Pan, X. (2007) GIS coupled with traffic simulation and optimization for incident response. *Computers, Environment and Urban Systems,* 31 (2), 116-132.

Huang, S. & Ran, B. (2002) An application of neural network on traffic speed prediction under adverse weather condition. In: *Proceedings of the Transportation Research Board 82nd Annual Meeting*, Washington D.C, USA.

Huang, S. & Sadek, A. W. (2009) A novel forecasting approach inspired by human memory: The example of short-term traffic volume forecasting. *Transportation Research Part C: Emerging Technologies,* 17 (5), 510-525.

Hunt, P. B., Robertson, D. I., Bretherton, R. D. & Winton, R. I. (1981) *SCOOT - a traffic responsive method of coordinating signals*. Transport and Road Research Laboratory, Crowthorne, Berkshire, UK, TRRL Laboratory Report 1014.

Innamaa, S. (2000) Short-term prediction of traffic situation using MLP-neural networks. In: *7th World Congress on Intelligent Transportation Systems*, Turin, Italy. 6-9.

Ishak, S. & Alecsandru, C. (2004) Optimizing traffic prediction performance of neural networks under various topological, input, and traffic condition settings. *Journal of transportation engineering,* 130 (4), 452-465.

Jain, R. & Smith, J. M. (1997) Modeling vehicular traffic flow using M/G/C/C state dependent queueing models. *Transportation Science,* 31 (4), 324-336.

Jeffery, D. J., Russam, K. & Robertson, D. I. (1987) Electronic route guidance by AUTOGUIDE: the research background. *Traffic engineering & control,* 28 (10), 525-529.

Jha, M., Gopalan, G., Garms, A., Mahanti, B. P., Toledo, T. & Ben-Akiva, M. E. (2004) Development and calibration of a large-scale microscopic traffic simulation model. *Transportation Research Record,* 1876 (1), 121-131.

Joachims, T. (1999) Making large scale SVM learning practical. In: Scholkopf, B., Burges, C. J. C. and Smola, A. J. (eds.) *Advances in Kernel Methods - Support Vector Learning*, MIT Press, pp. 169-184.

Ju, Y., Kim, C. & Shim, J. (1997) Genetic-based fuzzy models: interest rate forecasting problem. *Computers & industrial engineering,* 33 (3-4), 561-564.

Kamarianakis, Y. & Prastacos, P. (2005) Space-time modeling of traffic flow. *Computers & Geosciences,* 31 (2), 119-133.

Kayacan, E., Ulutas, B. & Kaynak, O. (2010) Grey system theory-based models in time series prediction. *Expert Systems with Applications,* 37 (2), 1784-1789.

Kendall, D. G. (1953) Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. *The Annals of Mathematical Statistics,* 24 (3), 338-354.

Kernighan, B. W. & Ritchie, D. M. (1988) *The C Programming Language.* Prentice Hall, ISBN: 0131103628.

Kim, K. J. (2003) Financial time series forecasting using support vector machines. *Neurocomputing,* 55 (1-2), 307-319.

Knoop, V. L. (2009) *Road incidents and networking dynamics effects on driving behaviour and traffic congestion*. PhD Thesis. Delft University of Technology, Delft, The Netherlands.

Kosarev, E. & Pantos, E. (1983) Optimal smoothing of'noisy'data by fast Fourier transform. *Journal of Physics E: Scientific Instruments,* 16 (6), 537-543.

Kriesel, D. (2007) *A brief introduction to neural networks*. [Online]. Available from: http://www.dkriesel.com [Accessed 04/12/2012].

Krishnan, R. (2008) *Travel time estimation and forecasting on urban roads*. PhD Thesis. Centre for Transport Studies, Imperial College London.

Krishnan, R. & Polak, J. W. (2008) Short-term travel time prediction: An overview of methods and recurring themes. In: *Proceedings of the Transportation Planning and Implementation Methodologies for Developing Countries Conference (TPMDC 2008)*, Mumbai, India.

Kruskal, J. B. (1964) Nonmetric multidimensional scaling: a numerical method. *Psychometrika,* 29 (2), 115-129.

Kuo, R., Chen, C. & Hwang, Y. (2001) An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy Sets and Systems,* 118 (1), 21-45.

Kusiak, A., Zheng, H. & Song, Z. (2009) Short-term prediction of wind farm power: a data mining approach. *IEEE Transactions on Energy Conversion,* 24 (1), 125-136.

Lawrence, R. (1997) Using neural networks to forecast stock market prices. *University of Manitoba*.

Lee, T.-C. (2007) *An agent-based model to simulate motorcycle behaviour in mixed traffic flow*. PhD Thesis. Centre for Transport Studies, Imperial College London.

Leshem, G. & Ritov, Y. (2007) Traffic flow prediction using adaboost algorithm with random forests as a weak learner. In: *World Academy of Science, Engineering and Technology*, Bangkok, Thailand. Citeseer, 193-198.

Levin, M. & Tsao, Y. D. (1980) On forecasting freeway occupancies and volumes. *Transportation Research Record,* 773, 47-49.

Li, M. W., Hong, W. C. & Kang, H. G. (2013) Urban traffic flow forecasting using Gauss-SVR with Cat mapping, Cloud model and PSO hybrid algorithm. *Neurocomputing,* 99 (1), 230-240.

Liaw, A. & Wiener, M. (2002) Classification and regression by randomForest. *R news,* 2 (3), 18-22.

Lin, W.-H., Lu, Q. & Dahlgren, J. (2002) Dynamic procedure for short-term prediction of traffic conditions. *Transportation Research Record,* 1783, 149-157.

Lin, W. H. (2002) A Gaussian maximum likelihood formulation for short-term forecasting of traffic flow. In: *Proceedings of the 4th International IEEE Annual Conference on Intelligent Transportation Systems*, Oakland, USA.

Lu, C. J., Lee, T. S. & Chiu, C. C. (2009) Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems,* 47 (2), 115-125.

Majhi, R., Panda, G. & Sahoo, G. (2009) Efficient prediction of exchange rates with low complexity artificial neural network models. *Expert Systems with Applications,* 36 (1), 181-189.

Martin, P. T., Chaudhuri, P., Tasic, I. & Zlatkovic, M. (2011) *Freeway incidents: simulation and analysis*. Civil and Environmental Engineering, University of Utah.

May, A. D. (1965) Traffic flow theory-the traffic engineers challenge. *Proc. Inst. Traf. Eng*, 290-303.

May, A. D. (1990) *Traffic Flow Fundamentals.* Prentice Hall, ISBN: 0139260722.

Maybeck, P. S. (1979) *Stochastic Models, Estimation and Control, Volume 1*. Academic Press, Inc., ISBN: 0-12-480701-1.

Meinshausen, N. (2006) Quantile regression forests. *The Journal of Machine Learning Research,* 7, 983-999.

Miles, J. C. & Chen, K. (2004) *ITS Handbook:Recommendations From the World Road Association (PIARC).* 2nd Edition. Artech House, ISBN: 2-84060-174-5.

Mineva, A. & Popivanov, D. (1996) Method for single-trial readiness potential identification, based on singular spectrum analysis. *Journal of Neuroscience Methods,* 68 (1), 91-99.

Mitchell, T. M. (1997) *Machine Learning.* New York, McGraw-Hill, ISBN: 0071154671.

Mulhern, F. J. & Caprara, R. J. (1994) A nearest neighbor model for forecasting market response. *International Journal of Forecasting,* 10 (2), 191-207.

Müller, K. R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J. & Vapnik, V. (1997) Predicting time series with support vector machines. In: Gerstner, W., Germond, A., Hasler, M. and Nicoud, J.-D. (eds.) *Artificial Neural Networks — ICANN'97*, Springer, Berlin Heidelberg, pp. 999-1004.

OECD/ECMT (2007) *Managing Urban Traffic Congestion.* France, OECD Publishing, ISBN: 9282101282.

Okutani, I. & Stephanedes, Y. (1984) Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B: Methodological,* 18 (1), 1-11.

Pai, P. F. & Lin, C. S. (2005) A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega,* 33 (6), 497-505.

Panwai, S. & Dia, H. (2005) Comparative evaluation of microscopic car-following behavior. *IEEE Transactions on Intelligent Transportation Systems,* 6 (3), 314-325.

Park, B., Messer, C. J. & Urbanik II, T. (1998) Short-term freeway traffic volume forecasting using radial basis function neural network. *Transportation Research Record,* 1651, 39-47.

Park, D. & Rilett, L. R. (1999) Forecasting freeway link travel times with a multilayer feedforward neural network. *Computer-Aided Civil and Infrastructure Engineering,* 14 (5), 357-367.

Park, D. C., El-Sharkawi, M., Marks, R., Atlas, L. & Damborg, M. (1991) Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems,* 6 (2), 442-449.

Park, J. & Sandberg, I. W. (1991) Universal approximation using radial-basis-function networks. *Neural Computation,* 3 (2), 246-257.

Perales Roehrs, J. (2001) *Incident modelling using a micro-simulation approach*. MSc Thesis. Centre for Transport Studies, Imperial College London and University College London.

Prasad, A., Iverson, L. & Liaw, A. (2006) Newer classification and regression tree techniques: bagging and Random Forests for ecological prediction. *Ecosystems,* 9 (2), 181-199.

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1992) *Numerical Recipes in FORTRAN 77: Volume 1, Volume 1 of Fortran Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, ISBN: 052143064X.

PTV (2009) *VISSIM 5.20 User Manual*. PTV Planung Transport Verkehr AG, Karlsruhe, Germany.

Qiao, F., Yang, H. & Lam, W. H. K. (2001) Intelligent simulation and prediction of traffic flow dispersion. *Transportation Research Part B: Methodological,* 35 (9), 843-863.

Qin, L. & Smith, B. L. (2001) *Characterization of accident capacity reduction*. University of Virginia, Report number: UVACTS-15-0-48.

Quadstone (2003) *Quadstone Paramics V4.2: Analyser Reference Manual*. Quadstone Limited, Edinburgh, UK.

Robinson, S. (2005) *The development and application of an urban link travel time model using data derived from inductive loop detectors*. PhD Thesis. Centre for Transport Studies, Imperial College London.

Robinson, S. & Polak, J. (2006) Overtaking rule method for the cleaning of matched license-plate data. *Journal of Transportation Engineering,* 132 (8), 609-617.

Rokach, L. (2010) *Pattern Classification Using Ensemble Methods.* World Scientific Publishing Company Incorporated, ISBN: 9814271063.

Ruping, S. (2000) *mySVM - Manual*. [Online]. University of Dortmund. Available from: http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/mysvm-manual.pdf [Accessed 12/06/2009].

Saad, E. W., Prokhorov, D. V. & Wunsch, D. C. (1998) Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks,* 9 (6), 1456-1470.

Saffari, A., Leistner, C., Santner, J., Godec, M. & Bischof, H. (2009) On-line random forests. In: *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, Kyoto, Japan. IEEE, 1393-1400.

Samoili, S. & Dumont, A. (2012) Framework for real-time traffic forecasting methodology under exogenous parameters. In: *Proceedings of the 12th Swiss Transport Research Conference (STRC)*, Ascona, Switzerland.

Samuel, A. (1959) Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development,* 3 (3), 210-229.

Sapankevych, N. L. & Sankar, R. (2009) Time series prediction using support vector machines: A survey. *IEEE Computational Intelligence Magazine,* 4 (2), 24-38.

Schölkopf, B., Burges, C. J. C. & Smola, A. J. (1998) *Advances in kernel methods: support vector learning.* MIT press, ISBN: 0262194163.

Sharma, S. K. & Sharma, V. (2012) Time series prediction using kNN algorithms via euclidian distance function: a case of foreign exchange rate prediction. *Asian Journal of Computer Science and Information Technology,* 2 (7), 219-221.

Short, R., D & Fukunaga, K. (1981) The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory,* 27 (5), 622-627.

SIAS (2005) *S-Paramics 2005 - SNMP Reference Manual*. SIAS Limited, Edinburgh, UK.

Simoes, N. (2012) *Urban pluvial flood forecasting*. PhD Thesis. Imperial College London.

Simoes, N., Wang, L., Ochoa, S., Leitao, J. P., Pina, R., Onof, C., Sa Marques, A. & Maksimovic, C. (2011) A coupled SSA-SVM technique for stochastic short-term rainfall forecasting. In: *Proceedings of the 12th International Conference on Urban Drainage*, Porto Alegre, Brazil.

Sivapragasam, C., Liong, S. Y. & Pasha, M. (2001) Rainfall and runoff forecasting with SSA-SVM approach. *Journal of Hydroinformatics,* 3 (3), 141-152.

Smith, B. L. & Demetsky, M. J. (1997) Traffic flow forecasting: Comparison of modeling approaches. *Journal of Transportation Engineering-ASCE,* 123 (4), 261-266.

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

Stathopoulos, A. & 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.

Stephanedes, Y. J., Michalopoulos, P. G. & Plum, R. A. (1981) Improved estimation of traffic flow for real-time control. *Transportation Research Record,* 795, 28-39.

Stone, C. J. (1977) Consistent nonparametric regression. *The Annals of Statistics,* 5, 595-620.

Sun, H., Liu, H. X., Xiao, H., He, R. R. & Ran, B. (2003) Short term traffic forecasting using the local linear regression model. In: *Proceedings of the 82nd Annual Meeting of the Transportation Research Board*, Washington D.C., USA.

Tam, M. & Lam, W. (2009) Short-term travel time prediction for congested urban road networks. In: *Proceedings of the Transportation Research Board 88th Annual Meeting*, Washington D.C,USA.

Tao, Y., Yang, F., Qiu, Z. J. & Ran, B. (2005) Travel time prediction in the presence of traffic incidents using different types of neural networks. In: *Proceedings of the Transportation Research Board 85th Annual Meeting*, Washington D.C,USA.

Tay, F. E. H. & Cao, L. (2001) Application of support vector machines in financial time series forecasting. *Omega,* 29 (4), 309-317.

Tenti, P. (1996) Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence,* 10 (6), 567-582.

TfL (2010) *Travel in London Report 2*. [Online]. Available from: http://www.tfl.gov.uk/assets/downloads/Travel_in_London_Report_2.pdf [Accessed 11/05/2012].

TfL (2013) *Data Feed Specification for Developers*. [Online]. Available from: http://www.tfl.gov.uk/assets/downloads/businessandpartners/TIMS_Feed_Technical_Specification_-_010313.PDF [Accessed 23/04/2013].

Thacker, N. A. & Lacey, A. J. (1996) *Tutorial: The Kalman Filter*. [Online]. Available from: http://www.cc.gatech.edu/classes/cs7322_98_spring/PS/kf1.pdf [Accessed 01/13/2011].

Toledo, T. & Koutsopoulos, H. N. (2004) Statistical validation of traffic simulation models. *Transportation Research Record,* 1876 (1), 142-150.

Trafalis, T. B., Santosa, B. & Richman, M. B. (2003) Prediction of rainfall from WSR-88D radar using kernel-based methods. *International Journal of Smart Engineering System Design,* 5 (4), 429-438.

Transport for London (2007) *London Travel Report 2007*. [Online]. Available from: http://www.tfl.gov.uk/assets/downloads/corporate/London-Travel-Report-2007-final.pdf [Accessed 18/11/2010].

Trivedi, H. V. & Singh, J. K. (2005) Application of grey system theory in the development of a runoff prediction model. *Biosystems Engineering,* 92 (4), 521-526.

TSS (2004) *Aimsun Version 4.2 Users Manual*. TSS-Transport Simulation Systems, Barcelona, Spain.

TSS (2010) *Aimsun 6.1 Users Manual*. TSS-Transport Simulation Systems, Barcelona, Spain.

Turochy, R. E. (2006) Enhancing short-term traffic forecasting with traffic condition information. *Journal of Transportation Engineering,* 132 (6), 469-474.

University of Vermont (2008) *Traffic volume forecasting tool simulates human memory*. [Online]. Available from: http://www.uvminnovations.com/graphics/PDF/SPN.pdf [Accessed 11/12/2012].

Van Lint, J. W. C. (2004) *Reliable travel time prediction for freeways*. PhD Thesis. Delft University of Technology, Delft, The Netherlands.

Van Lint, J. W. C., Van Zuylen, H. J. & Tu, H. (2008) Travel time unreliability on freeways: Why measures based on variance tell only half the story. *Transportation Research Part A: Policy and Practice,* 42 (1), 258-277.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*. New York, Springer Verlag.

Vapnik, V. (1998) *Statistical Learning Theory*. Wiley-Interscience, ISBN: 978-0471030034.

Venables, W. N., Smith, D. M. & Team, R. D. C. (2011) *An introduction to R*. Version 2.13.1. R Development Core Team, ISBN: 3-900051-12-7.

Venkatanarayana, R. & Smith, B. L. (2008) *Automated identification of traffic patterns*. University of Virginia, Report number: UVACTS-15-0-104.

Verikas, A., Gelzinis, A. & Bacauskiene, M. (2011) Mining data with random forests: A survey and results of new tests. *Pattern Recognition,* 44 (2), 330-349.

Vilarinho, C. & Tavares, J. P. (2012) Traffic model calibration: a sensitivity analysis. In: *15th edition of the EURO working group of transportation*, Paris, France.

Vlahogianni, E. I. (2009) Enhancing predictions in signalized arterials with information on short-term traffic flow dynamics. *Journal of Intelligent Transportation Systems,* 13 (2), 73-84.

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

Wand, M. P. & Jones, M. C. (1995) *Kernel Smoothing (Monographs on Statistics and Applied Probability).* New York, Chapman & Hill, ISBN: 0412552701.

Wang, Y. F. (2002) Predicting stock price using fuzzy grey prediction system. *Expert Systems with Applications,* 22 (1), 33-38.

Wiedemann, R. (1974) *Simulation des Straßenverkehrsflusses*. Univ., Inst. für Verkehrswesen.

Williams, B. M., Durvasula, P. K. & Brown, D. E. (1998) Urban freeway traffic flow prediction - Application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Board,* 1644, 132-141.

Williams, B. M. & Hoel, L. A. (2003) Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering-ASCE,* 129 (6), 664-672.

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

Wylie, M. (2012) *Martin Wylie on devising and evaluating urban active management strategies though micro-simulation*. [Online]. Available from: http://www.aimsun.com/press/THV6N4_Microsimulation%20Martin%20Wylie.pdf [Accessed 07/05/2012].

Xiao, H., Ambadipudi, R., Hourdakis, J. & Michalopoulos, P. (2005) *Methodology for selecting microscopic simulators: Comparative evaluation of AIMSUN and VISSIM*. University of Minnesota, Minneapolis, US, Report number: CTS 05-05.

Xie, Y., Zhang, Y. & Ye, Z. (2007) Short term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Computer Aided Civil and Infrastructure Engineering,* 22 (5), 326-334.

Yang, J. (2005) Travel time prediction using the GPS test vehicle and Kalman filtering techniques. In: *Proceedings of the American Control Conference*. 2128-2133 vol. 3.

Yapo, P. O., Gupta, H. V. & Sorooshian, S. (1996) Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data. *Journal of Hydrology,* 181 (1), 23-48.

Zhang, J., Hounsell, N. & Shrestha, B. (2012) Calibration of bus parameters in microsimulation traffic modelling. *Transportation Planning and Technology,* 35 (1), 107-120.

Zhang, J. & Hounsell, N. B. (2010) A comparison study on environmental impacts caused by bus signal priority strategies. In: *42nd Annual Universities Transport Studies Group (UTSG) Conference*, Plymouth, UK.

Zhang, J. & Zulkernine, M. (2006) A hybrid network intrusion detection technique using random forests. In: *Proceedings of the First International Conference on Availability, Reliability and Security (ARES' 06)*, Vienna, Austria. 262-269.

Zhang, X. Y. & Rice, J. A. (2003) Short-term travel time prediction. *Transportation Research Part C-Emerging Technologies,* 11 (3-4), 187-210.

Zheng, W., Lee, D. H. & Shi, Q. (2006) Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of Transportation Engineering,* 132 (2), 114-121.

Zhigljavsky, A. (2010) Singular spectrum analysis for time series: introduction. *Statistics and Its Interface,* 3 (3), 255-258.

Zhu, T., Kong, X. & Lv, W. (2009) Large-Scale Travel Time Prediction for Urban Arterial Roads Based on Kalman Filter. In: *International Conference on Computational Intelligence and Software Engineering, 2009*. 1-5.

Zurada, J. M. (1992) *Introduction to Artificial Neural Systems.* New York, West Publishing Company, ISBN: 0314933913.