Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.
Short-Term Traffic Condition Variables Forecasting using Artificial Neural Networks

by

STEPHEN DUNNE

A dissertation submitted to the University of Dublin in the partial fulfilment of
the requirements for the Degree of Doctor of Philosophy

Department of Civil, Structural and Environmental Engineering

October 2013
DECLARATION

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Stephen Dunne

October 2013
DEDICATION

This work is dedicated to my girlfriend, family and friends for constantly supporting me throughout the ups and downs of postgraduate research.
SUMMARY

Short-term traffic forecasting (STTF) is a critical element of Intelligent Transport Systems (ITS). The use of ITS is vital in order to ensure the sustainability and increase the efficiency of the transportation network. ITS bases decisions on traffic conditions and so it can only be effective if the forecasted future traffic conditions provided are accurate. There exists multiple STTF algorithms, of which Artificial Neural Networks (ANNs) are the predominant non-parametric approach. Prediction of traffic variables using ANN is a well-researched area (Karlaftis & Vlahogianni, 2011) and the different structures and applications of ANN in traffic forecasting have established the strength of these models compared to other existing methodologies. Due to their ability of precise predictions, adaptability, flexibility and availability of numerous software, ANN models are the focus of this thesis from the outset, while other models are used for accuracy comparisons. The research in this dissertation focused on adapting and improving further the ANN based STTF algorithms for predicting traffic conditions in more complex paradigms.

The input vectors to ANN algorithms and the learning rules employed within the network structure have been optimised for improving efficiency of traditional ANN algorithms in STTF literature. In the literature there are arguments for the use of reasonably short intervals (15 minutes) as traffic flow fluctuates over short time intervals and this suggests a loss of information when using coarser time aggregations. By forecasting data at different time aggregations in this work, conclusions could be drawn on which time aggregation is the most sensible based on the forecasting results.

The relationship between traffic flow and speed was also investigated in this work. A novel regime isolation methodology is described which takes both flow and speed into account when forecasting. This multivariate forecasting model produced both speed and flow forecasts and used a separate forecasting models depending on the current traffic conditions i.e. congested or uncongested. The ANN algorithms developed in the study were applied to predict
traffic condition variables separately under congested and uncongested regimes. The relationship between traffic speed and flow was utilised in identifying the congestion levels, with flow and speed having a linear relationship in the uncongested regime and a non-linear relationship in the congested regime. The multivariate algorithm predicted both speed and flow in near future.

One of the major contributions of this thesis involved the use of rainfall as an exogenous variable input to ANN prediction models for predicting traffic flow. With climate change an ever looming issue, it was important to factor in the effect of weather on traffic conditions. This was achieved through a multivariate prediction model which made use of rainfall as an additional model input, only when rainfall was expected in the coming prediction interval. Rainfall does not influence traffic conditions instantaneously but it changes long term traffic dynamics within the day. The effect of rainfall on traffic dynamics is best viewed by looking at different frequency levels of traffic condition variable time-series. As such, another novel feature of the work in this thesis involved the introduction of Stationary Wavelet Transforms (SWTs) to decompose traffic flow time-series. The use of SWT decomposition allowed the separate resolution components to be predicted separately in an effort to improve forecasting accuracy. This was among the first uses of SWTs in the field of traffic flow forecasting.

The ANN and SWT based weather adaptive traffic flow prediction model in this thesis was also applied to predicting travel time. In this work, both SWT and the multivariate rainfall approach were used to predict travel time sourced from traffic cameras on Pearse St., Dublin. Travel times change based on driver behaviour and hence predicting a range is often more meaningful to commuters than point forecasts. Hence, Forecast Regions (FRs) were constructed for travel time prediction using four different approaches.

The forecasts of traffic condition variables on both motorways and urban arterials were investigated and compared in this work. Urban arterial data was sourced from Dublin city
centre and forecasts were also conducted using highway data from the Motorway Incident Detection and Automated Signalling (MIDAS) dataset in the U.K. There have been many instances of urban arterial or highway traffic flow data forecasting but the two were rarely compared. The differences in the traffic behaviour at each site, and the relative forecasts, were examined. Both urban and highway datasets were used based on the scope of the different prediction algorithms developed in this work.

In summary, the objective of this thesis was to design innovative and accurate forecasting models applicable to real elements of ITS. Different traffic condition variables from different locations were predicted using various forecasting algorithms. Additional techniques were designed and applied to these forecasting algorithms, where doing so was advantageous to the models. The work in this thesis covered many areas of STTF and successfully contributed various prediction algorithms and modelling techniques to the STTF literature.
ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my thesis supervisor Dr. Bidisha Ghosh. Dr. Ghosh was always available to me if I had any issues whatsoever and the conversations we had over the course of my research were invaluable when shaping my dissertation. Her wealth of knowledge regarding my field was also a huge help to me when I had queries about specific models.

I would like to thank Dr. Kevin Ryan of the Civil, Structural and Environmental Engineering Department in Trinity College for all his help in the lab and on site, particularly with the ANPR cameras and storage media. Also, the people at DCC, particularly Brian Carrig and Gary Keegan, were a great help with obtaining data and getting the ANPR cameras set up. I would also like to thank the staff of the Austrian Institute of Technology in Vienna, especially Bernhard Heilmann, who shared their data with me and made me feel very at home during my working visit.

I would also like to thank my girlfriend Nicole who helped me through my four years of research. She was always there to spend time with me when a break from the research was needed and she supported and encouraged me from day one. She understood how much of a time commitment was required to complete postgraduate research and always encouraged me to spend as much time on my work as was needed.

Thirdly, I would like to thank my family for being a constant source of encouragement throughout my four years of postgraduate research. With the help of their constant support, the research was never allowed to get on top of me. It was always enjoyable to relax after a long day of research by spending time with my family.

Finally, I wish to thank my friends, particularly those who undertook postgraduate research at the same time as me. Simply having people to talk to who were going through the
same thing as me was a huge source of support. As well as this, they made my postgraduate research experience a time of fun which I will always look back on fondly.
TABLE OF CONTENTS

DECLARATION ii
DEDICATION iii
SUMMARY iv
ACKNOWLEDGEMENTS vii
TABLE OF CONTENTS ix
LIST OF FIGURES xii
LIST OF TABLES xv
LIST OF ACRONYMS xvi

1 Chapter 1 1
   1.1 Background 1
   1.2 Literature Review 2

2 Chapter 2 25
   2.1 Introduction 25
   2.2 Methodology 26
      2.2.1 Feed Forward Back Propagation Neural Network 29
      2.2.2 Radial Basis Function Neural Network 35
      2.2.3 Generalized Regression Neural Network 37
      2.2.4 Support Vector Machine 38
      2.2.5 Autocorrelation Neural Network (ACNN) 42
   2.3 Traffic Flow Data 43
      2.3.1 Urban Arterial Traffic Flow Data 43
      2.3.2 Motorway Traffic Flow Data 48
   2.4 Results 52
      2.4.1 Artificial Neural Network and Support Vector Machine Modelling Setup 53
      2.4.2 Comparison of Predictions 55
   2.5 Conclusion 73

3 Chapter 3 76
   3.1 Introduction 76
   3.2 Methodology 77
      3.2.1 Regime Isolation Methodology 78
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A Biological Neuron</td>
<td>27</td>
</tr>
<tr>
<td>2.2</td>
<td>A Basic Artificial Neuron</td>
<td>28</td>
</tr>
<tr>
<td>2.3</td>
<td>Basic Feed Forward Phase of a Neural Network Structure</td>
<td>30</td>
</tr>
<tr>
<td>2.4</td>
<td>Feed Forward Back Propagation Neural Network Structure</td>
<td>33</td>
</tr>
<tr>
<td>2.5</td>
<td>Radial Basis Function Neural Network</td>
<td>36</td>
</tr>
<tr>
<td>2.6</td>
<td>Generalized Regression Neural Network</td>
<td>38</td>
</tr>
<tr>
<td>2.7</td>
<td>Order of Processes</td>
<td>42</td>
</tr>
<tr>
<td>2.8</td>
<td>Map of Urban Arterial Junctions</td>
<td>44</td>
</tr>
<tr>
<td>2.9</td>
<td>Typical Hourly Traffic Flow at Junctions TCS 141, TCS 166 and TCS 183</td>
<td>45</td>
</tr>
<tr>
<td>2.10</td>
<td>Correlograms of (a) Non-Stationary Data &amp; (b) Stationary Data - Junction TCS 141</td>
<td>47</td>
</tr>
<tr>
<td>2.11</td>
<td>Map of Motorway Locations</td>
<td>49</td>
</tr>
<tr>
<td>2.12</td>
<td>Typical Hourly Traffic Flow at the A1 and M6 Motorways</td>
<td>50</td>
</tr>
<tr>
<td>2.13</td>
<td>Correlogram of (a) Non-Stationary and (b) Stationary A1 Motorway 15 Minute Data</td>
<td>51</td>
</tr>
<tr>
<td>2.14</td>
<td>FFBPNN Predictions at Junction TCS 141</td>
<td>55</td>
</tr>
<tr>
<td>2.15</td>
<td>FFBPNN Predictions at Junction TCS 166</td>
<td>56</td>
</tr>
<tr>
<td>2.16</td>
<td>FFBPNN Predictions at Junction TCS 183</td>
<td>56</td>
</tr>
<tr>
<td>2.17</td>
<td>FFBPNN Predictions at the A1 Motorway</td>
<td>57</td>
</tr>
<tr>
<td>2.18</td>
<td>FFBPNN Predictions at the M6 Motorway</td>
<td>57</td>
</tr>
<tr>
<td>2.19</td>
<td>RBFNN Predictions at Junction TCS 141</td>
<td>58</td>
</tr>
<tr>
<td>2.20</td>
<td>RBFNN Predictions at Junction TCS 166</td>
<td>58</td>
</tr>
<tr>
<td>2.21</td>
<td>RBFNN Predictions at Junction TCS 183</td>
<td>59</td>
</tr>
<tr>
<td>2.22</td>
<td>RBFNN Predictions at the A1 Motorway</td>
<td>59</td>
</tr>
<tr>
<td>2.23</td>
<td>RBFNN Predictions at the M6 Motorway</td>
<td>60</td>
</tr>
<tr>
<td>2.24</td>
<td>GRNN Predictions at Junction TCS 141</td>
<td>60</td>
</tr>
<tr>
<td>2.25</td>
<td>GRNN Predictions at Junction TCS 166</td>
<td>61</td>
</tr>
<tr>
<td>2.26</td>
<td>GRNN Predictions at Junction TCS 183</td>
<td>61</td>
</tr>
<tr>
<td>2.27</td>
<td>GRNN Predictions at the A1 Motorway</td>
<td>62</td>
</tr>
<tr>
<td>2.28</td>
<td>GRNN Predictions at the M6 Motorway</td>
<td>62</td>
</tr>
<tr>
<td>2.29</td>
<td>SVM Predictions at Junction TCS 141</td>
<td>63</td>
</tr>
<tr>
<td>2.30</td>
<td>SVM Predictions at Junction TCS 166</td>
<td>63</td>
</tr>
<tr>
<td>2.31</td>
<td>SVM Predictions at Junction TCS 183</td>
<td>64</td>
</tr>
<tr>
<td>2.32</td>
<td>SVM Predictions at the A1 Motorway</td>
<td>64</td>
</tr>
</tbody>
</table>
Figure 2.33: SVM Predictions at the M6 Motorway 65
Figure 2.34: MAPE vs. Time Aggregation Subplot 67
Figure 2.35: FFBPNN TCS 141 15 Minute Data - ACF vs. Non-ACF Predictions 71
Figure 2.36: TCS 141 Predictions - ACNN vs. Naive Method vs. Moving Average 72
Figure 3.1: Flow Diagram for the Multivariate Prediction Methodology 78
Figure 3.2: (a) Flow-Speed Scatterplot & (b) Difference of Slope Graph 79
Figure 3.3: Weekday 15 Minute Aggregate Traffic Flow 84
Figure 3.4: Weekday 15 Minute Aggregate Average Speed 85
Figure 3.5: Flow-Speed Scatterplots for the 4 Lanes 87
Figure 3.6: Prediction Results for 15 Minute Time Aggregations using ALRLM at Lane 1 of A1/9340A 94
Figure 3.7: Prediction Accuracy vs. Time Aggregations using ALRLM for 4 Motorway Lanes 95
Figure 4.1: Stationary Wavelet Transform Decomposition Tree 102
Figure 4.2: Stationary Neuro-Wavelet Prediction Framework 104
Figure 4.3: Location of SCATS Data Collection Sites 107
Figure 4.4: Typical Weekday Hourly Traffic Flow Data 108
Figure 4.5: Barchart of Hourly Rainfall Data for the first 10 Weekdays of 2009 109
Figure 4.6: Hourly Traffic Flow db3 Wavelet Decomposition 111
Figure 4.7: TCS 106 Prediction Results 113
Figure 4.8: TCS 125 Prediction Results 114
Figure 5.1: Travel Time Dataset Creation Flow Chart 120
Figure 5.2: ANPR capturing a Registration Plate 122
Figure 5.3: ANPR Camera and Waterproof Box 123
Figure 5.4: Schematics of Waterproof Box 124
Figure 5.5: Overview of ANPR Camera Sites 126
Figure 5.6: ANPR Site 1 Camera Locations 127
Figure 5.7: ANPR Site 2 Camera Locations 127
Figure 5.8: FTP Settings on ANPR Camera 129
Figure 5.9: Pearse St. 15 Minute Aggregated Average Travel Time Example 132
Figure 5.10: Location of ANPR Cameras in Tallaght 134
Figure 5.11: Tallaght Daily Travel Time Example (15 Minute Aggregation) 135
Figure 5.12: Schoenbrunnerstrasse as part of the Vienna free speed map 136
Figure 5.13: Schoenbrunnerstrasse Daily Travel Time Example (15 Minute Aggregation) 137
Figure 6.1: Average Travel Time in 15 Minute Intervals from 08:00 - 20:00 hours 152
Figure 6.2: Rainfall Barchart for 08:00 hours to 20:00 hours from July 30th to August 3rd 2012 153
<table>
<thead>
<tr>
<th>Table Number</th>
<th>Table Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Urban Arterial Traffic Flow Characteristics</td>
<td>46</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Urban Arterial Influential Points based on Correlograms</td>
<td>48</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Motorway Traffic Flow Characteristics</td>
<td>50</td>
</tr>
<tr>
<td>Table 2.4</td>
<td>Motorway Data Influential Points based on Correlograms</td>
<td>52</td>
</tr>
<tr>
<td>Table 2.5</td>
<td>Prediction Results - MAPE (%) and RMSE</td>
<td>66</td>
</tr>
<tr>
<td>Table 2.6</td>
<td>Computational Time for the Prediction Models</td>
<td>69</td>
</tr>
<tr>
<td>Table 2.7</td>
<td>ACF vs. Non-ACF Prediction Error Values</td>
<td>71</td>
</tr>
<tr>
<td>Table 2.8</td>
<td>TCS 141 15 Min Predictions- ACNN vs. Naive Model vs. MA technique</td>
<td>73</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Statistical Description of Traffic Flow &amp; Average Speed Observations from</td>
<td>86</td>
</tr>
<tr>
<td>Motorway Sites</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table 3.2</td>
<td>Traffic Flow Prediction Results</td>
<td>91</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>Average Traffic Speed Prediction Results</td>
<td>92</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>TCS 106 Rainfall Data Correlation Coefficients</td>
<td>112</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Prediction Accuracies at TCS 106 and TCS 125 under different Weather Conditions</td>
<td>115</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Pearse St. Travel Time Dataset Descriptive Statistics</td>
<td>133</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Tallaght Travel Time Dataset Descriptive Statistics</td>
<td>135</td>
</tr>
<tr>
<td>Table 5.3</td>
<td>Schoenbrunnerstrasse Travel Time Dataset Descriptive Statistics</td>
<td>138</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Travel Time Dataset Descriptive Statistics</td>
<td>153</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Travel Time Prediction Results</td>
<td>154</td>
</tr>
<tr>
<td>Table 6.3</td>
<td>FR Assessment Indices</td>
<td>161</td>
</tr>
</tbody>
</table>
# LIST OF ACRONYMS

<p>| ACF  | Autocorrelation Function                  |
| ACNN | Autocorrelation Neural Network            |
| ACO  | Ant Colony Optimisation                  |
| ALRLM| Levenberg-Marquardt Model                |
| ALRM | Adaptive Learning Rate Model             |
| ALRMM| Adaptive Learning Rate with Momentum Model|
| ANN  | Artificial Neural Network                |
| ANPR | Automatic Number Plate Recognition       |
| ARIMA| Autoregressive Integrated Moving Average  |
| ARIMAX| Autoregressive Integrated Moving Average Errors |
| ATIS | Advanced Traveller Information System    |
| ATMS | Advanced Traffic Management System       |
| AVD  | Alpha Vision Design                      |
| BHIE | Bayesian Hierarchical Interval Estimation|
| BP   | Back Propagation                         |
| BPNN | Back Propagation Neural Network           |
| CCTV | Closed Circuit Television                |
| CLC  | Coverage Length-based Criterion          |
| COV  | Coefficient of Variation                 |
| CP   | Conformal Prediction                      |
| DCC  | Dublin City Council                      |
| DWT  | Discrete Wavelet Transform               |
| EMD  | Empirical Mode Decomposition             |
| ERM  | Empirical Risk Minimisation              |
| FF   | Feed Forward                              |
| FFBPNN| Feed Forward Back Propagation Neural Network |
| FNN  | Fuzzy Neural Network                      |
| FR   | Forecast Region                           |
| FRBS | Fuzzy Rule-Based System                   |
| FRCP | Forecast Region Coverage Probability      |
| FTP  | File Transfer Protocol                    |
| GA   | Genetic Algorithm                         |
| GD   | Gradient Descent                          |
| GN   | Gauss-Newton                              |</p>
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>Global Positioning Systems</td>
</tr>
<tr>
<td>GRNN</td>
<td>Generalized Regression Neural Network</td>
</tr>
<tr>
<td>HA</td>
<td>Historical Average</td>
</tr>
<tr>
<td>k-NN</td>
<td>k-Nearest Neighbours</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman Filter</td>
</tr>
<tr>
<td>LBS</td>
<td>Location Based Sensor</td>
</tr>
<tr>
<td>LM</td>
<td>Levenberg-Marquardt</td>
</tr>
<tr>
<td>LS</td>
<td>Least Squares</td>
</tr>
<tr>
<td>ICP</td>
<td>Inductive Conformal Prediction</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISWT</td>
<td>Inverse Stationary Wavelet Transform</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Transportation Systems</td>
</tr>
<tr>
<td>MAPE</td>
<td>Mean Absolute Percentage Error</td>
</tr>
<tr>
<td>MCMC</td>
<td>Markov Chain Monte Carlo</td>
</tr>
<tr>
<td>MFRL</td>
<td>Mean Forecast Region Length</td>
</tr>
<tr>
<td>MIDAS</td>
<td>Motorway Incident Detection and Automated Signalling</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>MNN</td>
<td>Modular Neural Network</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MTL</td>
<td>Multi Task Learning</td>
</tr>
<tr>
<td>N-ICP</td>
<td>ICP using a Normalised Nonconformity Measure</td>
</tr>
<tr>
<td>NLR</td>
<td>Non-Linear Regression</td>
</tr>
<tr>
<td>NMFRKL</td>
<td>Normalised Mean Forecast Region Length</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimisation</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RBFNN</td>
<td>Radial Basis Function Neural Network</td>
</tr>
<tr>
<td>RCBO</td>
<td>Residual Current Circuit Breaker with Overcurrent Protection</td>
</tr>
<tr>
<td>RGS</td>
<td>Route Guidance Systems</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>SCATS</td>
<td>Sydney Coordinated Adaptive Traffic System</td>
</tr>
<tr>
<td>SDCC</td>
<td>South Dublin County Council</td>
</tr>
<tr>
<td>SHW</td>
<td>Seasonal Holt-Winters</td>
</tr>
<tr>
<td>S-ICP</td>
<td>ICP using a Standard Nonconformity Measure</td>
</tr>
<tr>
<td>SRM</td>
<td>Structural Risk Minimisation</td>
</tr>
<tr>
<td>SSE</td>
<td>Sum of Squared Error</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>SSNN</td>
<td>State Space Neural Network</td>
</tr>
<tr>
<td>STM</td>
<td>Structural Time-series Model</td>
</tr>
<tr>
<td>STTF</td>
<td>Short-Term Traffic Forecasting</td>
</tr>
<tr>
<td>SV</td>
<td>Support Vector</td>
</tr>
<tr>
<td>SVC</td>
<td>Support Vector Classification</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>SVR</td>
<td>Support Vector Regression</td>
</tr>
<tr>
<td>SWT</td>
<td>Stationary Wavelet Transform</td>
</tr>
<tr>
<td>U.K.</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>VAR</td>
<td>Vector Autoregressive</td>
</tr>
<tr>
<td>VPH</td>
<td>Vehicles Per Hour</td>
</tr>
<tr>
<td>WT</td>
<td>Wavelet Transform</td>
</tr>
</tbody>
</table>
1 Chapter 1

INTRODUCTION AND LITERATURE REVIEW

1.1 Background

The process of transporting people, goods and services from one location to another has been, and will always be, a fundamental aspect of human life. As such, different methods of transportation have been invented, tested and accepted throughout history. Different modes of transport have been popular at different times in the past. Air, rail and road transport have all maintained certain popularity since their inception as transportation modes. Of these modes, road transport is recognised as the most popular mode in modern times. However, the popularity of road transport has put a strain on the existing road infrastructure, particularly in highly populated urban areas. The more popular road transport becomes, the more vehicles take to the roads around the world. Accordingly, congestion on roads is a major problem.

Congestion results in delays, a decrease in the reliability of the transport network and damage to the environment through vehicle emissions. It is becoming a more prevalent issue in modern society and must be tackled. There are thought to be three methods of dealing with congestion; improving the transport network and infrastructure, increasing the number of passengers choosing public transport over private vehicle use, or improving the efficiency of the existing transport network. Problems encountered when trying to implement the first two methods are widely known. Firstly, roads can only become so wide; there is always a physical limit to how much a section of road can be improved e.g. roads in built up areas may have no room for expansion. The cost of such road expansion projects has also become a greater issue in light of the economic situation facing most countries in the world. Capital projects are under more scrutiny than ever as the purse strings are monitored carefully. Secondly, the matter of convenience means that there are plenty of road users who would not consider undertaking a journey using public transport. This will remain an issue unless the cost of public transport
drops enough to prompt users to reconsider. Therefore, the method of tackling congestion with the most promise is that of improving the performance of the transport network in situ.

The task of improving the efficiency and sustainability of the existing road network is not a simple one. Intelligent Transportation Systems (ITS) have been devised as a means of performing this task. ITS are a collection of applications designed to manage the traffic on the roads at present and to ensure the highest levels of capacity on our roads are reached, all while avoiding delays wherever possible. ITS are composed of many aspects geared towards the task of improving network efficiency and sustainability. These aspects include intelligent traffic signal control systems, motorway management systems, transit management systems, incident management systems, electronic toll collection systems, electronic fare payment systems, emergency response systems, travel information systems and route guidance systems (Khisty & Lall, 2002). These components can all be considered to fall under two major umbrella concepts of ITS; Advanced Traffic Management Systems (ATMS) and Advanced Traveller Information Systems (ATIS). ATMS incorporates traffic signal monitoring and adaptive control and traffic condition variable data recording through the use of data collectors such as inductance loop detectors embedded in the roads. Based on the traffic state as seen by ATMS' detectors, ATIS gives information regarding congestion to the motorists on the roads through such media as Variable Message Signs (VMS) above or beside roads or through the radio directly into vehicles. It follows that ATIS needs ATMS to function efficiently in order to be effective. Similarly, in order for ATMS to function efficiently, information about the actual and the near-term future traffic state is critical. It is this task of accurately forecasting traffic conditions upon which the work in this thesis is based. With this in mind, the literature review in Section 1.2 describes previous work in the field of traffic condition forecasting.

1.2 Literature Review

The requirement for accurate near-term future traffic state data necessitates the ability to make and continuously update predictions of traffic flows into the future (Cheslow et al.,
This was first noted by Cheslow et al. (1992) who surmised that "The ability to make and continuously update predictions of traffic flows and link times for several minutes into the future using real-time data is a major requirement for providing dynamic traffic control". Traffic flow prediction has thus become an active field of research over the last fifteen years. With the increasing amount of ITS implementation globally, a vast volume of real time traffic flow data has become available to those in traffic management positions. There are a lot of different data collection techniques including using inductance loop detectors embedded in the road surface which count the number of cars passing above, while more recently video surveillance of roads and satellite imaging technology have also come into use. As the data is now more easily available, the problem is how best to use this data to predict traffic flow at select intervals, as ITS cannot function optimally without the ability to forecast traffic flow volumes in the short-term.

The difference between traffic flow behaviour on urban arterials and highways should also be considered. The majority of studies focus on motorway data as the traffic conditions on motorways are generally uncongested, giving rise to theoretically easier predictions. One of the key aims of traffic modelling on motorways is incident detection, as it is incidents that have the capacity to cause congested conditions on motorways. Urban arterials on the other hand generally exhibit periodic traffic flow behaviour, with congestion expected during key commuting hours. The traffic flow on urban arterials displays distinct stop and go patterns, which is often due to the presences of traffic signals on these arterials. Taking into account the distinct differences between traffic flow behaviour on urban arterials and highways, this section aims to give a review of the various different models used to predict short-term traffic flow on both road types, particularly given the importance of short-term traffic flow predictions to well-functioning ITS. The literature contains numerous different approaches to traffic flow forecasting. However, short-term traffic flow forecasting models can generally be classified as either parametric models or non-parametric models and hence these two families of models are presented in separate subsections.
Parametric Models

Parametric models are based on the assumption that the structure of the underlying process of a time-series to be forecasted can be described by a few parameters. Parametric models used to forecast short-term traffic flow in the literature include smoothing techniques, Historical Average (HA) algorithms, auto-regressive linear processes and Kalman filtering. Examples of each of these forecasting methodologies are discussed in the following subsections:

(a) Historical Average Algorithms

HA models simply use an average of past traffic volumes over multiple time instants to forecast future traffic volume and consequently they rely upon the periodic nature of traffic flow. However, there is one major flaw with this type of model i.e. the model has no way to react to dynamic changes in the transportation system, such as collisions or oil spills etc. Due to this, the HA forecasting model produced the largest error in a study by Smith and Demetsky (1997), which compared the HA model with an AutoRegressive Integrated Moving Average (ARIMA) model, a Back Propagation Neural Network (BPNN) and a nonparametric regression model.

(b) Smoothing Techniques

Exponential smoothing works by weighting the observed time-series data unequally, with the more recent observations being given a larger weight compared to those observations which were observed in the more distant past. The unequal weighting is accomplished using one or more smoothing parameters, which determine how much weight is given to each observation. The advantages of this model are that it is generally intuitive and gives a good forecast for an inexpensive technique. Exponential smoothing has been used earlier in the field by Williams et al. (1998) and Smith et al. (2002) before Li et al. (2008) improved the model with their work. Smoothing techniques have also been used more recently in the literature, with one of the most popular and long lasting models of the exponential smoothing family, the Seasonal Holt-Winters (SHW) model, used by Hong (2011).
The ARIMA set of models are the best known and most successful short-term traffic flow predictors of autoregressive linear processes. ARIMA models were developed to incorporate the advantages of Autoregressive (AR) and Moving Average (MA) models in a single structure. AR models can be described as linear regressions of the current value of a time-series against one or more previous values of the same time-series. MA models on the other hand are linear regressions of the current value of the time-series against the white noise components of one or more values of the time-series. Box and Jenkins (1976) developed a methodology to combine the two, producing a model known as the autoregressive moving average (ARMA) model, for use on stationary time-series. For non-stationary time-series, Box and Jenkins created the ARIMA model which removed the non-stationarity nature of the dataset. The methodology was organised in the iterative steps of model identification, model estimation and model diagnosis i.e. identifying an appropriate ARIMA process, fitting it to the data and then using the fitted model for forecasting.

Ahmed and Cook (1979) introduced the ARIMA model to the traffic flow forecasting literature when they compared their ARIMA (0, 1, 3) model with a double exponential smoothing model, an exponential smoothing model with adaptive response and a simple MA model. The authors determined that the ARIMA model produced the best forecast of the 4 models selected. In the nineties, Hamed et al. (1995) applied an ARIMA model to forecast urban traffic volume. Traffic forecasting using the ARIMA model has also been documented in a paper by Kirby et al. (1997). They stated that the ARIMA model performed well for traffic flow prediction. Hybrid approaches combining the strengths of ARIMA and other models have also produced favourable results (Huang and Tang, 2007). The ARIMA model has maintained popularity in the traffic variable forecasting literature for many years and further more recent examples of the model are covered in the traffic speed and travel time sections of the literature review.
(d) Kalman Filtering

Developed by Rudolf Kalman in 1960, the Kalman Filter (KF) is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. Continuous learning based on gradient descent (GD) can be quite slow due to the reliance on instantaneous estimates of gradients. This can be overcome by viewing the supervised training of a recurrent network as an optimum filtering problem, the solution of which recursively utilises information contained in the training data in a manner going back to the first iteration of the learning process. One feature of KFs is that the theory is formulated in terms of state-space concepts, providing efficient utilisation of the information contained in the input data. Also, estimation of the state is computed recursively i.e. each updated estimate and the data currently available, so only the previous estimate requires storage.

The potential of the KF in terms of traffic flow forecasting was first demonstrated by Okutani and Stephanedes (1984) who used Kalman filtering in urban traffic volume prediction and then developed an extended KF to predict traffic diversion in freeway entrance ramp areas. Subsequently, Whittaker et al. (1997) demonstrated the potential of this method in a multivariate setting. Following research with M25 motorway flow data, Chen and Grant-Muller (1999) suggested that a percentage absolute error (PAE) of approximately 9.5% could be achieved for a KF type network with five hidden units. Stathopoulos and Karlaftis (2003) demonstrated the KF’s superiority over a simple ARIMA formulation when modelling traffic data from different periods of the day. The authors also clarified that state space models referred to in the literature generally have the same basic underlying theory as KF models. In general, the term state space refers to the model and the KF term refers to the estimation of the state.

This concludes the section covering the use of various parametric models for traffic flow prediction in the literature. The following section contains descriptions of different non-parametric models, along with instances of their application as traffic flow forecasting models in the literature.
Non-Parametric Models

Nonparametric models used in the literature include nonparametric regression and Support Vector Regression (SVR) based models. However, the most commonly used non-parametric models are those involving Artificial Neural Networks (ANNs) and as such, these models are discussed in greater depth in the forthcoming section.

Nonparametric Regression

Nonparametric regression relies on data describing the relationship between dependent and independent variables. The basic approach of nonparametric regression is influenced by its background in pattern recognition (Karlsson and Yakowitz, 1987). The approach locates the state of the system, defined by the independent variables, in a neighbourhood of past, similar states. Once this neighbourhood has been established, the past cases in the neighbourhood are used to estimate the value of the dependent variable. The methodology is based on classical set theory and employs the concepts and techniques of probability theory. These models do not require any strict assumptions regarding a functional relationship between dependent and independent variables. Nonparametric regression has roots in pattern recognition, as described by Han and Song (2003) in their review paper: “the approach locates the state of the system (defined by the independent variables) in a neighbourhood of past, similar states. Once this neighbourhood has been established, the past cases in the neighbourhood are used to estimate the value of the dependent variable.”

Davis and Nihan (1991) were among the first to use a nonparametric regression method to forecast traffic flow and they found that the method performed comparably to a linear time-series approach. As documented previously, Smith and Demetsky (1997) compared their nearest neighbour formulation of nonparametric regression with a HA model, an ARIMA model and an ANN model at two sites and concluded that “the nearest neighbour formulation of nonparametric regression holds considerable promise for application to traffic flow forecasting.”
and noted that this model "experienced significantly less error than the other three models at both sites". They believed that the introduction of heuristic forecast generation models improved nonparametric regression and showed that there were other opportunities to increase further the performance of nonparametric regression models. One obvious improvement they suggested was increasing the size of the databases to provide a better set of neighbours to use in forecasting. Kernel neighbourhoods have also been used and Smith et al. (2002) suggested that the method produced predictions with an accuracy comparable with that of the seasonal version of an ARIMA model. Finally, Clark (2003) found that non-parametric regression was more accurate when predicting flow and occupancy in motorways than when predicting speed.

**Support Vector Machines**

The theory of Support Vector Machines (SVMs) (Vapnik, 1995) is covered in detail in Section 2.2.4 of this thesis. SVM models are relatively new models in traffic flow forecasting literature, with papers only becoming common very recently. These models can be used to perform real time forecasting for an online situation (Castro-Neto et al., 2009). The authors found their online SVM model performed particularly well under non-recurring atypical traffic conditions.

Data mining technology has been utilised to decrease the size of training data required for SVMs (Wang et al., 2009) and the authors confirmed improvements in computation time and accuracy of prediction. The choice of kernel can have a big impact on forecasting accuracy when using SVM models. Wei et al. (2010) used a wavelet Least Squares Support Vector Regression (LS-SVR) model for forecasting traffic flow and the authors chose a Mexican hat function as the kernel function (See Section 2.2.4). They found that for 5, 10 and 15 minute predictions, their model outperformed ARIMA and ANN forecasting algorithms.

Hong (2011) forecasted real traffic flow values from northern Taiwan with a seasonal SVR with a chaotic simulated annealing algorithm and presented predictions of greater accuracy than the compared SARIMA, BPNN and SHW models. Hong et al. (2011) also performed a forecast using an SVR model with Ant Colony Optimisation (ACO) and again
found the SVR model to outperform a SARIMA model using the same traffic flow data. Finally, Zhang et al. (2011) developed a hybrid model to use the strengths of both SVMs and SARIMA models. The authors found the hybrid model to be superior to either of the individual models in terms of Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). As can be seen in the dates of the references presented in this section, the use of SVMs in traffic condition variable forecasting is a fledgling topic at the moment. However, the promise of the model is clearly evident. Additional SVM models are documented in later sections of the literature review.

Artificial Neural Networks

As evident from the literature review to this point, there exist multiple STTF algorithms. Of these algorithms, Artificial Neural Networks (ANNs) are the most efficient to date. ANNs can predict accurately in greatly differing scenarios, once they have trained with data from a given forecasting location. They need only a small amount of training data before they can compute accurate and fast predictions. There are many different types of ANNs in the literature and it can be seen that different ANN algorithms can be more suitable for different forecasting tasks. This versatility of ANNs is increased further by their adaptability in relation to being used alongside other forecasting techniques as part of a hybrid forecasting model. Due to the benefits described, ANN models are the focus of this thesis from the outset, while other models are used for accuracy comparisons. The research in this dissertation is focused on adapting and improving further the ANN based STTF algorithms for predicting traffic conditions in more complex paradigms.

As ANN models are used as the primary traffic forecasting tool in this thesis, they are discussed in greater detail than other techniques. The methodology of ANNs and how they work is covered in detail with diagrams and equations in Section 2.2. This section is devoted to the history of the application of ANNs in short-term traffic flow forecasting. Clark et al. (1993), Kwon and Stephanedes (1994) and Smith and Demetsky (1994) all predicted traffic flow using
ANNs involving simple Multi-Layer Perceptrons (MLPs). Vythoulkas (1993) took this approach but added complexity, using a MLP network with a learning rule based on a KF. Early work by Smith and Demetsky (1994) compared an ANN model with a nearest neighbour forecasting model. The models were used to forecast traffic flow at two different locations in Virginia, America. Their ANN model consisted of a 10 neuron hidden layer and was trained with data from one location but was used to predict traffic flow at the two different locations. The errors recorded were reasonable at the first location, but increased when the network was used to predict traffic volume at the second location; this was due to the fact the network was not retrained, hence it could not pick up the vagaries of the traffic conditions at the second location as accurately as the first location's data with which it was trained. This highlighted a potential weakness with ANNs i.e. a general model is very difficult to produce; individual models at individual locations almost always produce more accurate forecasts. However, it is important to note that developing a model that can generalize is a very difficult task. In effect, all forecasting models suffer from this same weakness.

Further early examples of ANNS in traffic forecasting flow included Ulbricht (1994) using multi Recurrent Neural Networks (RNNs) in predictions and Gilmore and Abe (1995) using a Hopfield network. Dougherty and Cobbett (1997) used a BPNN to forecast short-term traffic flow, speed and occupancy on inter-urban motorways in The Netherlands. It was found that while the network produced reasonable estimates of traffic flow and occupancy, the results did not outperform naïve methods of forecasting in use at the time. Ledoux (1997) designed a two-step cooperation based ANN traffic flow forecasting model. The first step involved using local ANNs to model traffic flow on single signalised links. The second step then modelled the traffic flow over a large network of junctions, based on communications between the local ANNs. This research showed the potential of integrating ANN forecasting models into adaptive traffic control systems such as ITS. This potential would be realised in the following years. Smith and Demetsky (1997) applied a BPNN model to short-term traffic flow forecasting, considering the success of the use of Back Propagation (BP) as the mapping model in other
fields. The results achieved by this model were reasonable but there was quite a large percentage of 'bad misses' in the forecast i.e. when some predictions were off, they were off by some way. Conventional ANN structures, such as the Feed Forward Back Propagation Neural Network (FFBPNN) algorithm have been utilised widely by researchers to predict traffic flow in short-term future (Kirby et al., 1997; Dougherty and Cobbett, 1997; Zhang, 2000). In terms of applying adaptive learning rules to FFBPNN structures, as in Chapter 3 of this thesis, papers from various fields have reported on the improvements found when adaptive learning rules were applied. Adaptive learning rules in conjunction with the conventional FFBPNN structures have been widely used in such fields as language modelling (Chen and Lee, 1995), power system voltage profile prediction (Ahmed et al., 1997), pattern recognition (Yu et al., 2001), transformer fault diagnosis (Sun et al., 2007), and system identification (Zhang and Shi, 2009). The consensus among these papers is that on applying the adaptive learning algorithms, the training performances were significantly improved in terms of both faster convergence and smaller error.

ANNs with a radial basis function as the activation function have also been widely used in the literature. Wang & Xiao (2003) used a distributed Radial Basis Function Neural Network (RBFNN) model based on an adaptive fuzzy C-means clustering algorithm to forecast short-term traffic flow. Their method involved using the fuzzy C-means clustering algorithm "to classify training objects into a couple of clusters, each cluster is trained by a sub RBFNN, and membership values are used for combining several RBFNN outputs to obtain the final result. The performance of the model was promising, with the results compared with real traffic data for validation. Zhao et al. (2006) also used RBFNNs to forecast urban traffic flows. They used a double RBFNN based on Particle Swarm Optimisation (PSO). The PSO algorithm was used to determine the parameters of the two radial basis function networks. The authors described this method favourably in terms of accuracy, while also stating that "the method not only simplifies the structure of RBFNNs, but also enhances training speed." Upon comparing their model with a traditional single radial basis function model, the performance of the double RBFNN was
found to outperform the single RBFNN and also had more practical value. Jin and Sun (2008) focused on incorporating Multi Task Learning (MTL) in the design of ANN forecasting models and stated that MTL has the potential to improve generalization by transferring information of extra tasks in training signals. In this case, a MTL BPNN was developed by “incorporating traffic flow data at several contiguous time instants into an output layer.” Results from the model proved it to be an effective method of traffic flow forecasting by showing that the MTL BPNN results outperformed a single task learning model using the same traffic flow input data.

Having discussed some papers in detail and displayed the variety of models within the ANN forecasting family and the methods in which researchers have put ANNs to use with the aim of forecasting traffic flow, the following is a brief review of various other ANN models: Yasdi (1999) developed Jordan’s sequential network and dynamic ANNs were introduced to traffic flow forecasting by Ishak and Alescandru (2004). Apart from conventional ANNs, usage of non-conventional ANN structures (Hamad et al., 2009), such as time-delayed ANNs (Yun et al., 1998; Abdulhai et al., 1999; Lingras and Mountford, 2001), RNNs (Lint et al., 2002) and genetically optimised ANNs (Abdulhai et al., 2002; Vlahogianni et al., 2007) are well-known in this field. Hybrid ANN structures (ANN structures used in conjunction with other signal processing algorithms) are also well-known for their applicability in traffic flow predictions. Spectral basis ANNs (Park et al., 1999), ANNs combined with a fuzzy modelling approach involving Kalman filtering (Stathopolous et al., 2008), ANNs combined with principal component analysis (Zhang and He, 2007) and pattern-based ANNs (Vlahogianni et al., 2006) are a few examples of such studies. An excellent comprehensive recent review of the different approaches and applications of ANNs in traffic prediction is provided in a review paper by Karlaftis and Vlahogianni (2011). The different structures and applications of ANNs in traffic forecasting have established the versatility of these models compared to other existing methodologies. As mentioned earlier, due to their ability to predict precisely, to adapt and to be flexible for use in both univariate and multivariate paradigms, ANN algorithms have been chosen as the primary forecasting algorithms for modelling traffic conditions in this thesis.
The sheer volume of the literature is itself a statement that ANNs are seen as an excellent tool with which to produce accurate short-term traffic flow forecasts. Vlahogianni et al. (2003) discussed various paradigms applied to short-term traffic flow forecasting in their review paper and that review paper should also be referred to for further examples of ANN in the field of STTF. The authors also stated the real power of ANNs is not only their proven ability to provide good predictions, but also their overall performance and robustness in modelling traffic data sets. Other advantages of using ANN algorithms to forecast short-term traffic include the fact that ANNs can produce accurate multiple step-ahead forecasts with less computational effort, that they have been tested with significant success in modelling the complex temporal and spatial relationships lying in many transportation data sets and that, as proven by Zhang et al., (1998) they are capable of modelling highly non-linear relationships in a multivariate setting. All these advantages contributed to the decision to use ANNs as the central forecasting model in the work in this thesis as ANNs are models capable of accurate prediction whether univariate or multivariate and regardless of variable chosen.

The previous paragraphs have documented many approaches to forecasting traffic flow. However, traffic flow is only one parameter of several that can indicate the performance of a road network. It is also important to consider additional variables such as traffic speed and route travel times. Traffic speed is an important indicator for several parts of ITS and prediction of traffic speed is particularly crucial for ATIS and ATMS. The traffic speed forecasting literature is underdeveloped in comparison to that of traffic flow forecasting and historically researchers have found speed more difficult to predict than flow. One of the earliest papers on traffic speed predictions is the work by Dougherty and Cobbett (1997). In this work, the authors used ANNs to forecast speed, flow and occupancy in the Rotterdam region of Holland. Although the methodology was not specifically geared towards traffic speed prediction, the authors were still disappointed with the poor prediction of traffic speed, with flow and occupancy forecasts proving to be much more accurate. The authors cited the distorting effect of slow moving vehicles, particularly in low flow conditions as a major reason for the poor performance of the
traffic speed prediction. This research confirmed that predicting traffic flow and traffic speed are different tasks entirely due to the fundamental differences between the two datasets.

Lee et al. (1998) compared multiple regression, ARIMA, ANN and Kalman filtering models for predicting short-term traffic speeds. The authors stated that all results were ‘reliable’ but that the ANN and KF models were more accurate and realistic than the others. ANNs were also chosen to predict traffic speeds on two lane rural highways and again the performance of the forecasting model was comparable to regression based models (McFadden et al., 2001). Huang and Ran (2003) looked at the impact of weather on traffic speed by using weather variables among the input variables to a FFBPNN traffic speed forecasting model and reported accurate forecasts. SVM forecasting models were compared with ANN models for predicting traffic speed using San Antonio freeway data (Vanajakshi and Rilett, 2004). The authors found the models were fairly evenly matched but noted that SVMs outperformed ANN models when the training data was of a lower quality and quantity. Further research on traffic speed prediction using FFBPNN models has been undertaken more recently by Lee et al. (2007).

Yildirim and Cataltepe (2008) approached the problem of traffic speed prediction by using many different data sensor locations on roads in Istanbul, Turkey. They compared the performance of SVM and k-Nearest Neighbour (k-NN) models and noted that the most accurate predictions occurred when only strongly correlated sensor’s data was used as input to the models. Hamad et al. (2009) proposed a hybrid model, melding a multilayer FFBPNN with the use of the Empirical Mode Decomposition (EMD), a key part of the Hilbert-Huang transform. The authors tested this approach on link speed data obtained from detectors on the I-66 in Fairfax, Virginia and produced positive MAPE values. Another different approach in the literature was that of Yue et al. (2009) who applied a phase-space reconstruction technique to the prediction of traffic speed time-series. Their results suggested that chaotic characteristics exist in traffic systems and they also calculated an expected maximum window for accurate speed predictions of less than sixteen minutes in the future. Their predictions backed up this finding as they were in good agreement with the observed values up to sixteen minutes in the
future, but the accuracy dropped off dramatically beyond twenty minutes. Guo and Williams (2010) implemented an online algorithm based on layered KFs to generate short-term traffic speed forecasts and associated prediction intervals. Finally, Qu et al. (2011) estimated speed by fusing low-resolution positioning data with other sources. Importantly, regardless of approach or methodology, the existence of papers listed previously emphasises the importance of traffic speed predictions to the correct and effective functioning of ATIS and ATMS.

The final traffic condition variable forecasted in this thesis is travel time in Chapter 6. Travel time is very important for ATIS as the travel time information helps drivers make decisions on which route to take. Similarly, travel time is a key indicator for Route Guidance Systems (RGS), which are becoming more popular inside vehicles. The following is a brief review of the travel time forecasting literature. Early work was based on simulated travel time data, rather than real world data (Oda, T., 1990; Anderson et al., 1994; Al-Deek et al., 1998). Chen and Chien (2001) and Chien and Kuchipudi (2002) used Kalman filtering for travel time prediction on simulated and real travel time data respectively. Travel time prediction can be done using indirect or direct means. Kisgyorgy and Rilett (2002) compared both methods, based on information collected by loop sensors and GPS. The authors forecasted traffic parameters (speed, occupancy and volume) multiple periods ahead using Modular Neural Networks (MNNs) and then determined the travel time based on these predicted values. This method is known as indirect travel time prediction. Their second approach was to directly predict travel time using MNNs with loop detector data and they reported better accuracy using the direct prediction method. SVR has also been used to forecast travel time, with accurate predictions reported (Wu et al., 2004).

Different ANN models have appeared in the travel time prediction literature, with travel time on interurban highways in Finland having been predicted using FFBPNNs (Innamaa, 2007) and State Space Neural Networks (SSNNs) having been employed in a real-time direct travel time prediction framework in Holland (Van Lint, 2006). The ARIMA model also features in the literature. Yang (2006) used ARIMA modelling on data collected using GPS on a
highway in Minnesota and reported that the travel time predictions were accurate and the model could be easily modified and transferred to a different site. Lee et al. (2009) focused on predicting travel time in an urban network, as opposed to on a freeway. The authors used real-time and historical travel time predictors to discover patterns in the data and thus create travel time prediction rules based on these patterns. They stated that their work demonstrated that travel time prediction for an urban network could be achieved cost effectively by utilising the raw data of Location Based Sensor (LBS) applications. The impact of weather on travel time prediction has also been investigated (El Faouzi et al., 2010). They successfully used toll collection data to derive speed and travel time estimations and predictions, taking weather effects into account. Chang et al. (2010) developed a naïve Bayesian classification algorithm which provided a velocity class to be used for measuring travel time accurately. The authors noted high degrees of accuracy when a large historical database was available. Finally, Myung et al. (2011) used the k-NN method, with combined data from a vehicle detector system and an automatic toll collection system, to predict real travel time data and found the model to be capable of producing accurate forecasts even in instances where missing data was present.

In addition to point travel time forecasts, the creation of a Forecast Region (FR) based on travel time forecasts is also popular in the literature. FRs are used to measure the uncertainty of predictions. Yar and Chatfield (1990) and Chatfield and Yar (1991) were among the first to introduce FRs when they defined prediction intervals for SHW forecasting models. They found that, at the time, the FRs were generally too narrow to be useful in practice. Later in the literature, Heskes (1997) defined FRs as confidence intervals created for the unknown future values and stated that consequently, FRs were larger than confidence intervals. In essence, a FR consists of a range of values with a predetermined probability, or confidence level, of containing the mean value of future observations (Khosravi et al., 2011). There are many different ways to construct FRs in the literature. Sun and Zhang (2004) used the bootstrap technique for traffic time-series FR creation using a local linear predictor. Bayesian Hierarchical Interval Estimation (BHIE) can also been used to create FRs (Ghosh et al., 2010).
The Delta method is another of the most popular FR construction techniques. This method involves representing ANNs as Non-Linear Regression (NLR) models, thus allowing the application of standard asymptotic theory to the models in order to construct FRs (Khosravi et al., 2011). Papadopolous and Haralambous (2011) followed a machine learning framework called Conformal Prediction (CP) to create FRs. CP assigned confidence measures to forecasts on the assumption that the data were independent and identically distributed. It is clear from the papers described here that there are numerous ways to construct FRs and these ways are compared in Chapter 6 of this thesis.

The majority of research discussed thus far in the literature review falls into the category of univariate forecasting i.e. a single parameter is forecasted using data from a single station. However, the field of multivariate forecasting is growing. Multivariate forecasting includes forecasting a single variable using data from multiple stations, forecasting multiple variables from multiple stations, and forecasting a traffic variable using an exogenous variable such as rainfall. It is necessary to conduct a brief review of the multivariate traffic condition variables forecasting literature for completeness.

An early effort at multivariate traffic flow prediction involved using upstream sensor data, in addition to previous observations at the target location, to predict traffic flow on French motorways (Williams, 2001). The author used transfer functions with Autoregressive Integrated Moving Average Errors (ARIMAX) and reported that the model outperformed previous univariate forecasting models tested on the same motorway data. Motorway data from London's orbital M25 was used to test a multivariate nonparametric regression traffic prediction approach (Clark, 2003). The author used MAPE to compare univariate forecasts of speed, flow and occupancy with multivariate forecasts of combinations of the same traffic variables and reported that the multivariate predictions performed better for the most part. Stathopoulos and Karlaftis (2003) used a multivariate state-space approach for urban traffic flow prediction using three minute data from urban arterials in Athens, Greece. They found that their multivariate model held much promise for predictions in an urban environment.
In work to develop a Genetic Algorithm (GA) approach to optimise ANN structures, Vlahogianni et al. (2005) tested their ANN structures on both univariate and multivariate urban signalised arterial traffic flow data with good forecasting accuracy reported. Dimitriou et al. (2008) also used GA for tuning; however they were tuning a Fuzzy Rule-Based System (FRBS), as opposed to an ANN structure, and again reported good predictions on urban signalised arterials. Ghosh et al. (2009) developed a Structural Time-series Model (STM) that outperformed a SARIMA model for multivariate traffic flow predictions in congested urban arterials in Dublin, Ireland. Further work using the urban arterial data in Dublin involved integrating a SARIMA model with the theoretical based cell transmission model (Szeto et al., 2009). The results indicated this technique could be very useful at locations where continuous data is not available. Chandra and Al-Deek (2009) used a Vector Autoregressive (VAR) approach to predict speed and flow on freeways. Their work included cross-correlation checks on speeds at different stations and good MAPE values were documented.

The regime isolation multivariate methodology consideration described in Chapter 3 focuses on the relationship between flow and speed under different conditions. The regime-based consideration has been studied previously in smooth-transition regression models to characterise regimes in daily cycles of traffic flow (Kamarianakis et al., 2010) and in a hybrid methodology on the basis of genetically optimised MLPs and non-linear dynamics (Vlahogianni, 2009) for traffic volume and occupancy predictions in predominantly signalised urban transport networks. Predicting two traffic condition variables simultaneously, as in the work in Chapter 3, is seen as a standard multivariate forecasting algorithm. However, multivariate forecasting can also refer to the use of exogenous input variables when predicting a single traffic condition variable. Forecasting methodologies are developed in this thesis using exogenous model input variables when predicting traffic flow and travel time variables respectively. Rainfall is the main exogenous variable being discussed and therefore a brief review of the literature dealing with the effect of weather variables on traffic conditions is conducted here. The effect of rain on travel demand and traffic accidents has been confirmed in
the literature (Chung et al., 2005). The study illustrated that rain had a very real effect on travel demand, with the demand decreasing with increasing rain and it was also determined that there are more accidents during rainy conditions. The effect of rainfall and other weather parameters on traffic volume was also studied in Melbourne, Australia using data covering the period 1989 to 1996 (Keay and Simmonds, 2005). The authors found rainfall to be the most strongly correlated weather variable with traffic volume and proved statistically that volume reduced on wet days. Findings in a technical report (Maze et al., 2005) identified that adverse weather reduced the capacities and operating speeds on roadways, resulting in congestion and productivity loss.

Rakha et al. (2008) examined the impact of inclement weather on freeway traffic stream behaviour specifically using weather data and loop detector data from several different areas in America. Their main finding was that the reductions in free-flow speed and speed at capacity increased as the rain and snow intensities increased. The effects of weather on daily crash counts were also investigated (Brijs et al., 2008) and it was shown that several presumptions related to the effect of weather conditions on crash counts were found to be significant in the data. Camacho et al. (2010) applied a multiple non-linear regression model to a set of correlated traffic and weather parameters and found again that rain and snow both caused a reduction in speed. They also determined that wind speed over 8 metres per second affected the traffic conditions at their site but they noted that their results may have been somewhat site-specific. Further studies on the impact of weather on traffic conditions have been conducted recently (Datla and Sharma, 2010; El Faouzi et al., 2010). The effect of weather on traffic congestion has also been investigated and it was found that rain had a clear extend effect on morning peak congestion (Wang et al., 2010). These conclusions suggest that rain has a definite impact on traffic flow. Finally, Billot et al. (2010) showed the benefits of integrating the impact of rain into traffic state estimation. The combination of all these findings confirm the potential benefits of including weather variables in traffic condition variable forecasting, as in the weather adaptive forecasting models developed in this thesis.
The work regarding the weather adaptive forecasting models in this thesis discerned that using rainfall as an exogenous variable is most useful when the rainfall data is split into its component frequencies through the use of wavelet decomposition to predict traffic time-series at different frequency levels. Hence, a review of the use of wavelets in traffic forecasting is presented here. The Wavelet Transform (WT) is a popular signal processing technique. The theory behind the wavelet transform was developed around the start of the nineties (Mallat, 1989; Daubechies, 1992). WTs have been used in conjunction with ANNs for various purposes in transportation research. One of the first instances of the use of WTs combined with ANNs in the transportation field was work on the utilization of the Discrete Wavelet Transform (DWT) for data filtering to improve the performance of a neuro-fuzzy ANN incident detection algorithm (Samant and Adeli, 2001). Other examples include the development of a wavelet energy algorithm featuring a RBFNN for fast incident detection on rural and urban roads (Karim and Adeli, 2003) and the use of WTs with Recurrent Neural Networks (RNNs) for online modelling and control of traffic flow (Liang and Wei, 2007).

The use of wavelets in traffic condition variable forecasting specifically is reasonably uncommon in the literature as the field is still quite new. Jiang and Adeli (2005) developed a novel time-delay RNN with wavelets, dubbed a dynamic wavelet ANN model for traffic flow forecasting. The authors stated their model was effective using data aggregated in 2, 5 or 10 minute intervals and noted the model’s prediction accuracy was within 10% of the observed flow, despite having limited training data. They also noted the lack of traffic flow forecasting research using wavelets in their literature review. Xie et al. (2007) created a wavelet KF and tested db4 and Haar wavelets as mother wavelets. Their wavelet KF outperformed a standard KF, again showing the potential of including wavelet decomposition in time-series forecasting. Another example of wavelets in traffic flow prediction involved using WTs to process the non-linear and stochastic characteristics of the original traffic flow data (Gu et al., 2011). The authors then used a GA optimised Fuzzy Neural Network (FNN) to predict short-term traffic flow and accurate predictions were reported.
In the few instances of the WT in transportation literature, it has mainly been employed as a denoising procedure as the coefficients generated using Discrete Wavelet Transforms (DWTs) are non-stationary in nature and hence the regular time-series prediction algorithms cannot be used successfully with DWT coefficient data series. To eliminate the problem of non-stationarity in time-series datasets decomposed using DWTs, a novel redundant WT (also referred to in the literature as nondecimated WT, Stationary Wavelet Transform (SWT) or a-trous algorithm) has been introduced by researchers in different fields (Zhang et al., 2001; Liu, 2009). In summary, the trend in recent years has been to use the SWT, a stationary version of the DWT, to develop robust and efficient time-series prediction algorithms. In the field of traffic flow forecasting, SWTs have been used for denoising traffic volume time-series data from highways, prior to prediction with self-organising ANNs (Boto-Giralda et al., 2010). However, the multiresolution structure of SWTs, involving independent modelling of the higher and lower resolution components, has yet to be exploited in an urban arterial traffic prediction framework.

In summary, the work in this thesis is an effort to improve the prediction of traffic flow, traffic speed and travel times, in both urban road networks and on major highways, through the use of novel ANN based time-series forecasting methodologies. The research presented hereafter is predominantly based on traffic condition variable forecasting. This thesis is presented in six chapters following this introductory chapter.

The majority of the work in this thesis centres on forecasting traffic condition variables of differing time-aggregations using various different prediction algorithms and models. **Chapter 2** is based on employing the technique of autocorrelation as a pre-processing technique when forecasting short-term traffic flow using ANN prediction models. The benefits of such a pre-processing technique include faster computation times of said prediction algorithms along with improved prediction accuracy. This chapter also compared three ANN model structures (Feed Forward Back Propagation Neural Network (FFBPNN), Radial Basis Function Neural Network (RBFNN) and Generalized Regression Neural Network (GRNN))
with a SVM model and a Naïve model to ascertain which model produces the most accurate forecasts. Among the models studied, and based on the results at different time aggregations and locations, the FFBPNN would be the most recommended prediction algorithm in this case.

Studies on modelling traffic speed are much less common than those involving traffic flows. The scarcity of literature on traffic speed forecasting is even more apparent for multivariate conditions. To address this, a regime-adjusted multivariate dual-traffic flow and speed prediction algorithm was proposed in Chapter 3. The proposed regime adjustment methodology in this chapter utilised a time-series classification approach to isolate the observations in congested or non-linear regimes and then subsequently preprocessed such information for further prediction. The multivariate short-term traffic condition prediction model proposed utilised the FFBPNN structure in conjunction with adaptive learning rules. In this regard, four different learning algorithms were used and compared in this chapter. These comprised a simple GD algorithm, a GD algorithm with ALR, and a GD algorithm with momentum and a Levenberg-Marquardt (LM) training algorithm. These FFBPNN models with four different learning rules were compared to quantify the extent of traffic condition variable prediction accuracy through the introduction of adaptive learning rules to FFBPNN algorithms. The FFBPNN structure using the adaptive LM training algorithm has been observed to be the most accurate traffic flow predictor in this work.

Chapter 4 included the introduction of rain as an additional input to the FFBPNN prediction algorithms. The effect of rain on travel demand and traffic accidents has been confirmed in the literature. In this regard, it was thought of as imperative to develop a traffic flow forecasting model that takes into account the weather conditions of the forecasted time interval. In addition to improvements found in previous chapters, there has been some research to suggest that breaking a time-series down (to its smooth and noisy components in essence) can also result in a prediction of greater accuracy. The task of breaking down a time-series is best accomplished through the use of wavelet decomposition by Stationary Wavelet Transforms (SWT). However, the multiresolution structure of SWTs, involving independent modelling of
the higher and lower resolution components, has yet to be exploited in an urban arterial traffic prediction framework. Therefore, the time-series are decomposed using SWTs, so that the individual components can be predicted separately by prediction algorithms specifically tuned to deal with the behaviour present in each component of the time-series. In summary, the work in this chapter incorporated using the multiresolution structure of SWTs to its fullest potential in developing a weather adaptive neuro-wavelet traffic forecasting algorithm which took into account the effect of weather at different resolution levels.

Having investigated traffic flow and speed modelling in different conditions in previous chapters, the next aim of the work in this thesis was to examine travel time modelling. However, the traffic condition data used for modelling in previous chapters had come from the Sydney Coordinated Adaptive Traffic System (SCATS) dataset in Dublin and the Motorway Incident Detection and Automated Signalling (MIDAS) dataset in the United Kingdom (U.K.). The SCATS system records only traffic flows at urban arterial junctions. MIDAS data contains flows, headways, speeds and occupancy from lanes on British highways. SCATS data was recorded at intersections in urban areas, while MIDAS data was recorded at points on motorways in the U.K. but neither of these datasets contained information on travel time. Therefore, there was a requirement for an alternative data resource, which makes use of data recorded over a route, rather than point based data, in order to work on travel time. Hence, in Chapter 5, an Automatic Number Plate Recognition (ANPR) camera setup capable of recording travel time in Dublin, Ireland and a probe vehicle data collection system in Vienna, Austria are described.

Chapter 6 applied the ANN and SWT based weather adaptive traffic flow prediction models to predicting travel time. In this work, both SWTs and the multivariate rainfall approach were used to predict travel time sourced from ANPR cameras on Pearse St., Dublin. Travel time is a variable much desired by motorists. However, travel time can vary vastly based on driver behaviour. Therefore, this chapter also included an investigation into FRs, with five different FR construction techniques tested in combination with the travel time point forecasts. The five
FR construction techniques compared were the Bootstrap method, the Delta method, the Bayesian Hierarchical Interval Estimation (BHIE) method and two models based on Inductive Conformal Prediction (ICP) with a standard nonconformity measure (S-ICP) and a normalized nonconformity measure (N-ICP).

The thesis is concluded in Chapter 7. This chapter included a summary and critical analysis of the previously documented work along with suggestions for future possible paths of research determined through the work in this thesis.
2 Chapter 2

SHORT-TERM TRAFFIC FLOW FORECASTING USING ARTIFICIAL NEURAL NETWORKS AND AUTOCORRELATION

2.1 Introduction

As discussed in the literature review, there have been many different types of ANN models with various paradigms applied to traffic flow forecasting. Based on the existing literature, it is evident that generally ANN algorithms use large input and training datasets. It is important to develop techniques or strategies to reduce the size of the dataset. It is expected that a reduction in training dataset will result in a reduction in computational time. However, a reduction in dataset size alone is not acceptable if the accuracy of the ANN prediction algorithms is compromised as a result. Hence it is crucial that the methods also preserve, and in some cases improve, the accuracy of the networks despite the reduction in training dataset size.

In this chapter, three different ANN models are used for forecasting traffic flow time-series at four different time aggregations. These three ANN algorithms are the Feed Forward Back Propagation Neural Network (FFBPNN) model, the Radial Basis Function Neural Network (RBFNN) model and the Generalized Regression Neural Network (GRNN) model. These three models are discussed as part of an in-depth introduction to ANN models. The ANN models used throughout the thesis are all based on the foundations explained in this chapter. A fourth prediction model, Support Vector Machine (SVM), is also examined in this work. The four prediction models are also compared to a simple naïve model and a moving average model in order to quantify their accuracy compared to standard prediction models. The traffic flow data used herein is divided into two categories; urban arterial traffic flow data from Dublin city centre and motorway traffic flow data from the U.K. Prior to prediction, the autocorrelation technique is used to determine which sections of the past traffic flow information are the most influential in predicting the future traffic flow. These influential points are used as the inputs to
the ANN and SVM prediction models. So, traffic flow data from urban signalised junctions and highways is modelled to evaluate the predictive capacity of four different time-series prediction algorithms. The accuracy of the ANN and SVM models is checked by calculating and comparing the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) of each prediction algorithm.

The work in this chapter is intended to improve accuracy and minimise the size of traffic flow datasets required while predicting traffic flow by employing an autocorrelation based filtering technique on the traffic flow data. The use of autocorrelation with ANN models leads to the term Autocorrelation Neural Network (ACNN) to describe the models used in this and subsequent chapters. Three different ACNN algorithms and a SVM model are used to model traffic flow of four different time aggregations at urban arterials and motorways. Stricter rules regarding the optimal data aggregation for forecasting have been indicated, by comparing the errors at different aggregate times. The performances of the FFBPNN, RBFNN, GRNN and SVM forecasting models are analyzed to determine their suitability at the different time aggregations.

2.2 Methodology

In this chapter and subsequent chapters, various ANN models are used to predict traffic condition variables. Therefore, the basics of the ANN models used in this thesis are discussed here to give some understanding of the theory behind them.

ANNs are an attempt to recreate the computing power of the brain artificially. A good insight into the comparison between biological networks and artificial networks can be found in books by Rojas (1996) and Pfeifer et al. (2010). Rojas states that nervous systems possess global architectures of variable complexity, but notes that all architectures are composed of similar building blocks; the neural cells or neurons. In the nervous system, neurons receive a signal and then produce a response to said signal. The basic architecture of a biological neuron is shown in Fig 2.1.
The dendrites in the Fig 2.1 are the channels for incoming information. Dendrites receive information at synapses which are contact regions with other cells. Organelles in the body of the cell produce all necessary chemicals for the continuous working of the neuron. The task of the cell body is to sum the incoming activation received from the dendrites. The output signals are transmitted by the axon, of which each cell has at most one. Pfeifer states that “The axon makes connections to other neurons. The dendrites can be excitatory, which means that they influence the activation level of a neuron positively, or they can be inhibitory in which case they potentially decrease the activity of a neuron. The impulses reaching the cell body (soma) from the dendrites arrive asynchronously at any point in time. If enough excitatory impulses arrive within a certain small time interval, the axon will send out signals in the form of spikes. These spikes can have varying frequencies.” So, in a simplification, there are four elements, dendrites,
synapses, cell body, and axon, which compose the minimal structure adopted from the biological model when designing ANNs i.e. artificial neurons for computing will also have input channels, a cell body and an output channel. In the biological neuron, synapses are simulated by contact points between the cell body and input or output connections. In the artificial case, a weight will be associated with these points. Figure 2.2 shows a very basic artificial neuron layout, in order to explain similarities with its biological counterpart.

\[ f(x) = f(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n) \]

**Figure 2.2: A Basic Artificial Neuron**

The artificial neuron in Fig 2.2 has \( n \) inputs (represented by white arrows in the biological case in Fig. 2.1). Each of these inputs \((x_1, \ldots, x_n)\) can be a real value transmitted on input channel \( i \) (the input channels in the biological case are the dendrites in Fig 2.1). The function \( f \), or activation function (represented by the cell body in Fig. 2.1) as it is often referred to, can be chosen arbitrarily depending upon the desired function of the network. It is commonplace for each input channel to have an associated weight \( w_i \) which means that the incoming information \( x_i \) is multiplied by the corresponding weight \( w_i \), as displayed by the function output (represented by the axon in the biological neuron in Fig. 2.1) in Fig 2.2. In the simplest terms, an ANN is a network composed of many of these basic artificial neurons.

The nervous system in the body can perform many different functions, which in turn leads to a very variable morphology. This ability to perform different functions is replicated by ANNs to some degree, as ANN models exist capable of time-series prediction, pattern
recognition, optimisation and classification etc. Each different ANN model is based on a network of simple neurons as in Fig 2.2. As summarised by Rojas, "Different models of artificial neural networks differ mainly in the assumptions about the primitive functions used, the interconnection pattern, and the timing of the transmission of information." All different types of ANNs can be looked at in terms of "four basics" (Pfeifer et al., 2010). These four basics are listed in the book as:

(1) The characteristics of the node. We use the terms nodes, units, processing elements, neurons and artificial neurons synonymously. We have to define the way in which the node sums the inputs, how they are transformed into level of activation, how this level of activation is updated, and how it is transformed into an output which is transmitted along the axon.

(2) The connectivity. It must be specified which nodes are connected to which and in what direction.

(3) The propagation rule. It must be specified how a given activation that is traveling along an axon is transmitted to the neurons to which it is connected.

(4) The learning rule. It must be specified how the strengths of the connections between the neurons change over time.

The different specifications of these four basics are what separate the various types of ANN mentioned in the literature and used in this thesis. With the basics of ANNs covered, the following three subsections will discuss the particular types of ANN used in this chapter; the FFBPNN, RBFNN and GRNN respectively.

2.2.1 Feed Forward Back Propagation Neural Network

The first traffic flow forecasting model used in this work is the FFBPNN. The Feed Forward (FF) phase of this network structure behaves similarly to the 'network of basic artificial neurons' described in Section 2.2. A collection of neurons and arrows displaying the basic FF phase of an ANN is shown in Fig 2.3.
The ANN pictured in Fig. 2.3 consists of a set of inputs, which are multiplied by a set of weights to give a net input. A bias is added to the net input and the result is put through an activation function to produce the network's output. Two different kinds of parameters can be adjusted during the training of an ANN, namely the weights and the value in the activation functions. It is difficult to optimise the ANN with two parameters to be adjusted at this stage and it would be easier if only one of the parameters is adjusted. Hence, the bias neuron is introduced to the structure. The bias neuron lies in one layer, is connected to all the neurons in the next layer, but none in the previous layer and it always emits 1. Since the bias neuron emits 1 the weights, connected to the bias neuron, are added directly to the combined sum of the other weights. Therefore, the bias is included in the structure to allow the effect of changes in weight to be seen more clearly, as weight changes become the predominant alteration in each iteration.
of the network training cycle. Mathematically, the neuron \( j \) can be described by Equation (2.1):

\[
\begin{align*}
    u_j &= \sum_{i=1}^{n} w_{ij} x_i \\
    y_j &= \varphi(u_j + b_j)
\end{align*}
\]  

(2.1)

where, \( n \) is the number of inputs, \( x_1, x_2, \ldots, x_n \) are the input signals, \( w_{ij}, w_{i2}, \ldots, w_{iy} \) are the synaptic weights of the neuron \( j \), \( u_j \) is the linear combiner output (or net input) due to the input signals, \( b_j \) is the bias, \( \varphi(.) \) is the activation function and \( y_j \) is the output signal of the neuron. When there are several connected layers in a network all going the same direction i.e. no loops, the network is said to be a multi-layer feed forward network or a Multi-Layer Perceptron (MLP).

FFBPNNS consist of an input layer, hidden layer(s) and an output layer. Hidden layers are the elements which distinguish such networks from other ANNs. The function of hidden neurons in the hidden layers is to intervene between the external input provided to the network and the output produced by the network in some useful manner. Multiple layers of neurons allow the network to learn non-linear and linear relationships between input and output vectors. The model used in this chapter contains a single hidden layer, along with an input layer and an output layer, both of which are connected to the outside world. It is common for feed forward networks such as these to have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Therefore, a log-sigmoid activation function is used in the hidden layer in this work. The sigmoid function gives an s-shaped graph. It is defined by Haykin (1994) as “a strictly increasing function that exhibits a graceful balance between linear and non-linear behaviour”. The log-sigmoid function is defined as:

\[
\varphi(v) = \frac{1}{1 + e^{-Av}}
\]  

(2.2)
where A is the slope parameter of the sigmoid function. Altering the value of the slope parameter gives sigmoid functions of different slopes. As the slope parameter approaches infinity, the sigmoid function becomes a Heaviside function. However, the Heaviside function has values of either 0 or 1, whereas the sigmoid function has a continuous range of values from 0 to 1. Also, the sigmoid function is differentiable, whereas the Heaviside function is not. The output layer in the FFBPNN in this chapter uses a linear function. These particular functions were chosen because, if the last layer of a multilayer network has sigmoid neurons, the outputs of the network are limited to a small range; however if linear output neurons are used in the last layer, the network outputs can take on any value.

In order for an ANN to learn the behaviour of the inputs presented to it, there must be a system whereby the weights are adjusted over time to take into account the different values of inputs and respective outputs generated over time. This system is the learning rule or learning algorithm, point 4 of the “4 basics” listed in Section 2.2, and in all NNs there must be a learning rule defined which determines how and when connection weights are updated. Dougherty (1995) suggested that ANNs be classified according to the type of learning rule employed. There are three main learning techniques that ANNs use. These are supervised learning, reinforcement learning and self-organising learning. FFBPNNs use supervised learning. In supervised learning an input is presented to one side of a FF network and the resultant output is computed. This is compared with the desired or target output for the given input. This is then used to update the weights in order to move the output closer to the desired output. If this is done over a few iterations i.e. many inputs presented to the ANN, it is hoped that the error will decline gradually as the network converges to a steady state. This target output approach is unique to supervised learning. By giving the network a target, the network is taught or supervised on how to behave when given other inputs. Hence the learning is described as supervised, due to the fact that the network is given an exact description of the behaviour required after each iteration is processed.
The full FFBPNN structure (building on the FF phase represented in Fig. 2.3), shown in Fig. 2.4 for clarity, is based on one of the most common learning algorithms in ANN literature, the GD learning algorithm.

The full FFBPNN structure (building on the FF phase represented in Fig. 2.3), shown in Fig. 2.4 for clarity, is based on one of the most common learning algorithms in ANN literature, the GD learning algorithm.

Figure 2.4: Feed Forward Back Propagation Neural Network Structure

The GD algorithm minimises the network error by modifying weights using supervised learning. An important generalisation of GD to a non-linear, multi-layer feed forward network is Back Propagation (BP), or backward error propagation. The BP learning algorithm is a well known supervised training method for FFNNs (Anderson and Donaldson, 1995). It learns by first computing an error signal and then propagating the error backward through the network as part of the weight updation process. It propagates the error backwards by assuming the network...
weights are the same in both the forward and backward directions. Mathematically, the BP phase of the network compares the calculated network outputs, $\hat{y}_j$, with the desired or target values, $y_j$, and is an iterative optimization of the error function that represents the performance of the network. This function of error, $E$, is defined as:

$$E = \frac{1}{2} \sum_{j=1}^{m} \sum_{i=1}^{n} (\hat{y}_j - y_j)^2$$  \hspace{1cm} (2.3)$$

where $m$ is the number of neurons. $E$ is minimised using the GD optimisation technique. The partial derivative of the error function in relation to each weight provides a direction of steepest descent. The corrections to the connection weights within the network are determined for each iteration using this partial derivative. The weight updation equation is:

$$w_j(k+1) - w_j(k) = \Delta w_j(k) = -\eta \frac{\delta E(k)}{\delta w_j(k)}$$  \hspace{1cm} (2.4)$$

where $\eta$ is the learning rate, a small positive constant, and $k$ is the iteration of the current weight. If the difference between network output and desired output becomes negligible or acceptable then the learning process terminates. BP is a very popular and widely used network-learning algorithm.

The Levenberg-Marquardt (LM) algorithm (Levenberg, 1944 and Marquardt, 1963) is an improvement over the basic GD based optimisation (El-Bakyr, 2003). Therefore, LM is used with the FFBPNN model in most applications in this thesis, unless specified otherwise. This algorithm is an iterative optimisation technique that locates the minimum of a multivariate non-linear function. The error function in Equation (2.3) can be rewritten as a function of the weights of the network as:

$$f(W) = \frac{1}{2} E^T E = \frac{1}{2} E^T (W)E^T (W) = E = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} (\hat{y}_j - y_j)^2, \text{where } W = [w_1, w_2, w_3, ..., w_n]^T$$  \hspace{1cm} (2.5)$$
and $E = [e_1, ..., e_m]$. $W$ consists of all the weights of the network. The weight updation is then achieved as:

$$W(k + 1) = W(k) - (JJ^T)J^T E_k$$

(2.6)

where the Jacobian matrix $J$, contains the first derivatives of the network errors with respect to weights and biases, and $E$ is a vector of network errors. $(JJ^T)$ is positive definite, but if it is not, then some perturbations are made into it to control the probability of it being non positive. Thus:

$$W(k + 1) = W(k) - (JJ^T + \lambda I)^{-1}J^T E_k$$

(2.7)

The quantity $\lambda$ is a learning parameter which decreases as the iterative process approaches to a minimum. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one (i.e. large $\lambda$), the algorithm behaves like a steepest descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton (GN) method ($\lambda = 0$). It is this value $\lambda$ that displays adaptive abilities.

### 2.2.2 Radial Basis Function Neural Network

RBFNNs are made with the same components as the FFBPNN model i.e. input layer, hidden layer and output layer. The difference lies with the hidden layer calculations, described by Equation (2.8):

$$y_j = \sum_{i=1}^{n} w_{ij} \varphi_i(x_i)$$

$$\varphi_i(x_i) = \exp\left( -\frac{\|x_i - \chi_i\|^2}{2\kappa^2} \right)$$

(2.8)

where, $x_1, x_2, ..., x_n$ are the input signals, $\chi_1, \chi_2, ..., \chi_n$ are the radial basis centres, $w_{ij}, w_{2j}, ..., w_{nj}$ are the synaptic weights of the neuron $j$, $\varphi_i()$ is the activation function, $\kappa$ is the Gaussian centre, $\|x_i - \chi_i\|$ is the Euclidean distance between the inputs and centres and $y_j$
is the output signal of the neuron. In RBFNNs, the activation function, $\phi(.)$ in Equation (2.8), is a bell-shaped curve, as can be seen in Figure 2.5, also known as Gaussian or Radial Basis Functions (RBFs). The figure shows how the inputs are presented to the Gaussian activation functions, before being summed and outputted. The radial basis activation functions are what make these networks unique. In the illustrative example there are $n$ inputs to the network and $n$ neurons in the hidden layer of the network.

Figure 2.5: Radial Basis Function Neural Network

These are continuous functions i.e. the output values asymptotically approach some constant for large magnitudes of input values and there is a single maximum output value. RBFs operate not on the input vector, but on the distance of input data vectors from pre-selected centres i.e. the hidden layer in a RBFNN computes the distance from the input layer data to each of the centres,
and then this set of distances is transformed before being passed to the output layer. RBF centres can be chosen using K-means sampling or sub-sampling or by assigning each input data point to be a centre. Once the centres are set, the Gaussian widths i.e. the radius of the bell shaped function, must be determined. This can be a delicate process, as if the Gaussian widths are too small, the network loses the ability to generalize but if the Gaussian widths are too broad, the network can lose fine detail. This results in the input space being covered by a number of localised basis functions. A given input typically only activates a limited number of hidden units significantly i.e. those within a close distance of the input.

2.2.3 Generalized Regression Neural Network

The final network examined in this chapter is the GRNN. This network is quite similar to the RBF type networks, but has a different second layer. GRNNs are memory-based networks that provide estimates of continuous variables and converge to the underlying (linear or non-linear) regression surface (Specht, 1991). These networks are commonly used for estimation of continuous variables, as in standard regression techniques. GRNNs do not require an iterative training procedure as in the BP method. They approximate any arbitrary function between input and output vectors, drawing the function estimate directly from the training data. These networks are related to RBFNNs which also have an approximation property based on a standard statistical technique called kernel regression. The adjustable parameters of such networks are the centre (the location of basis functions), the width of the Gaussians (the spread), the shape of the receptive field and the linear output weights. A GRNN can be treated as a normalised RBFNN in which there is a hidden unit centred at every training case. GRNNs can be thought of as a method for estimating the joint probability density function of an input vector and an output vector, given only a training set. Because the probability density function is derived from the data with no preconceptions about its form, the system is perfectly general (Kisi and Cigizoglu, 2007). The main difference between GRNNs and RBFNNs is that GRNNs have one neuron for each input point in the training, whereas RBFNNs have a variable number of neurons. An example of the GRNN structure is displayed in Fig. 2.6.
Figure 2.6: Generalized Regression Neural Network

The summation layer in a GRNN consists of a number of summation units (depicted in Fig. 2.6 as S) and a single division unit (depicted in Fig. 2.6 as D). The number of summation units is always equal to the number of outputs in the output layer, with two summation units and two outputs shown in the example network structure in Fig. 2.6. Another interesting part of the GRNN structure is that the final layers are not fully connected i.e. the outputs are connected to a single summation unit each along with the division unit, rather than being connected to every element of the summation layer. The outputs are calculated through a division of the summation unit's signal by the divisions unit's signals (Ripley, 1996).

2.2.4 Support Vector Machine

The fourth prediction algorithm used in this chapter is not an ANN; rather it is a SVM. A comprehensive technical report by Gunn (1998) covers the theory behind SVMs in great detail. The report explains that the foundations of SVMs were developed by Vapnik (1995). The SVM
algorithm is based on the Structural Risk Minimisation (SRM) principle. SRM has been shown to outperform the standard Empirical Risk Minimisation (ERM) principle (Gunn et al., 1997), which is used in ANN algorithms. SRM minimises an upper bound on the expected risk, whereas ERM in ANN algorithms attempts to minimise the error on the training data. This difference between SRM and ERM is said to leave SVMs with a greater ability to generalise, an important and practical capability in prediction models. SVMs were originally created to tackle the problem of classification, but they have since been put to use in the field of regression (Vapnik, 1995).

The basic idea of SVMs is that they define an optimal hyperplane, or generalisation of a plane into a different number of dimensions, for linearly separable patterns. So Support Vector Classification (SVC) is a more suitable title for the initial theory of SVMs. It is useful to consider the following classification problem:

\[ D = \{(x_1, y_1), \ldots, (x_n, y_n)\}, \quad x \in \mathbb{R}^n, y \in \{-1, 1\} \]  \tag{2.9}

where, \( D \) is a set of training vectors (\( x_i \) being the input and \( y_i \) the target as in the ANN theory described previously) belonging to two separate classes. In order to separate the two classes in the training set, a hyperplane is required:

\[ (\omega \cdot x) + B = 0 \]  \tag{2.10}

where, \( \omega \cdot x \) represents the dot product and \( B \) is the bias of the hyperplane. The hyperplane defined in Equation (2.10) is considered to be an optimal hyperplane i.e. the hyperplane which separates the two classes optimally, if the classes are separated without error, and if the distance between the closest vector to the hyperplane is maximised (Gunn, 1998). By constraining \( \omega \) and \( B \) in a canonical hyperplane, it is found that the hyperplane that optimally separates the two classes of data is the one that minimises the following equation (Vapnik, 1995):

\[ \Phi(\omega) = \frac{1}{2} \| \omega \|_2^2 \]  \tag{2.11}
It should be noted that a separating hyperplane in canonical form must satisfy the following constraints:

\[ y_i \left[ (\omega \cdot x_i) + B \right] \geq 1, \quad i = 1, \ldots, n \] (2.12)

The next step in understanding SVC theory is to optimise Equation (2.11) with respect to the constraints defined in Equation (2.12). The solution to this optimisation problem is the saddle point of the Lagrangian or Lagrange function (Minoux, 1986):

\[
L(\omega, B, \nu) = \frac{1}{2} \|\omega\|^2 - \sum_{i=1}^{n} \nu_i \left( y_i \left[ (\omega \cdot x_i) + B \right] - 1 \right)
\] (2.13)

where \( \nu \) are the Lagrange multipliers. The next task is to minimise the Lagrangian with respect to \( \omega \) and \( B \), and maximise it with respect to \( \nu \geq 0 \). Making use of Lagrangian duality and Kuhn-Tucker conditions to calculate the solution to the abovementioned equations, Support Vector (SVs) are defined as those points \( x_i \) which satisfy the following equation:

\[ y_i \left[ (\omega \cdot x_i) + B \right] = 1 \] (2.14)

The points \( x_i \) or SVs that satisfy Equation (2.14) are those points which have non-zero Lagrangian multipliers. These SVs will lie on the margin if the data is linearly separable. In some linearly separable cases, the hyperplane can in fact be determined by very few SV, which is a useful property.

However, not all datasets will be linearly separable. To deal with non-linearly separable cases, the concept of kernels is introduced. A polynomial kernel is shown in Equation (2.15) as an example.

\[ K(x, y) = (x \cdot y + 1)^2 \] (2.15)

This works by mapping the original two dimensional input vector into a six dimensional feature space. There are many different kernel types which work on the same principle of mapping the
input vector to a feature space of higher dimensionality. Kernel functions are discussed in the Gunn report paper (1998). After the mapping, the margin of the hyperplane is no longer of constant width. Thus, this allows for the separation of non-linear classes.

The introduction of a modified loss function, including a distance measure, paves the way for SVMs to tackle regression, which is crucial when applying SVMs to time-series prediction. Following the template in the Gunn report, it is now necessary to look at a regression problem to aid the explanation of Support Vector Regression (SVR). The task is to approximate the dataset in Equation (2.9) with the linear function in Equation (2.16).

\[ f(x) = (\omega \cdot x) + b \quad (2.16) \]

Based on Equations (2.9) and (2.16), the optimal regression function is defined as:

\[ \Phi(\omega, \xi^-, \xi^+) = \frac{1}{2} \| \omega \|^2 + q \sum_{i=1}^{n} \left( \xi_i^- + \xi_i^+ \right) \quad (2.17) \]

where, \( q \) is a pre-specified value, and \( \xi_i^- \) and \( \xi_i^+ \) are slack variables which represent lower and upper bounds respectively on the outputs of the system. In this work, the \( \varepsilon \)-insensitive loss function (Equation (2.18)) is used with the SVM prediction models.

\[ L_\varepsilon(y) = \begin{cases} 0 & \text{for } |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases} \quad (2.18) \]

Working down through the equations in a similar manner to the SVC case, again introducing Lagrangians and moving from linear to non-linear cases by transforming data using Kernel functions, leads to the following final function for regression:

\[ f(x) = \sum_{i=1}^{n} (\tilde{u}_i - \tilde{u}_i^*) K(x_i, x) \quad (2.19) \]

It is this outcome in Equation (2.19) that is vital when using SVMs for time-series prediction.
Two advantages of SVMs are that they are not affected by local minima and do not suffer from the curse of dimensionality. For the purposes of the work in this chapter, SVMs operate in a similar way to ANNs in that they take a training input and output and learn the behaviour of the training sets. Then, when presented with a test input, the SVM computes a test result which is then compared to the test target in order to calculate the error in prediction. Also, like the different ANN models explained previously, SVMs have certain options, such as the Kernel type, which can be tweaked to improve the performance of the prediction algorithm. These tweaks, for both ANN and SVM models, are discussed in Sections 2.4.1 to 2.4.4 inclusively.

2.2.5 Autocorrelation Neural Network (ACNN)

Prior to prediction, the data used in the forecasting algorithms in this and subsequent chapters is filtered using the AutoCorrelation Function (ACF). Autocorrelation is used to determine the most influential points for predicting the data. Figure 2.7 shows the order of processes in the methodology, specific to the ANN models.

![Figure 2.7: Order of Processes]

Autocorrelation can only be used to its full potential on stationary time-series. Full stationarity in a time-series occurs when all the moments of a time-series are equal to zero at any time. However, it is impossible to check all the moments and hence the term weakly stationary is
commonly used when defining the behaviour of a time-series. A time-series is said to be weakly stationary if there is no systematic change in the first and second moments i.e. mean and variance are equal to zero (Chatfield, 2004). Thus, in a stationary time-series, the properties of one section of the data are much like those of any other section e.g. the mean and variance do not change significantly over time or position. Traffic data is generally non-stationary (Ghosh et al. 2005). Therefore, in order to attain stationarity, the original traffic flow data has to be filtered. The ACF values of the filtered data are then used to determine the data points with the most influence on future values. These influential points are then used as inputs to the four different time-series prediction models. The process of seasonal filtering and autocorrelation, including the determination of influential points is explained in Section 2.3. Examples of the process using urban arterial data and motorway data are found in Sections 2.3.1 and 2.3.2 respectively.

2.3 Traffic Flow Data

In this chapter, traffic flow data is obtained and subsequently split into four different time aggregations; 5 minutes, 15 minutes, 30 minutes and 1 hour. The traffic flow data is collected from two different sources; urban arterials in the city centre of Dublin, Ireland, and motorways in the U.K. The urban arterial traffic flow data, obtained from three different urban arterial junctions in Dublin, is discussed in Section 2.3.1, while the motorway data, obtained from the Highway Agency's MIDAS dataset, is discussed in Section 2.3.2.

2.3.1 Urban Arterial Traffic Flow Data

Traffic flow data for 25 weekdays, from Wednesday 11th June 2008 to Tuesday 15th July 2008, at three different junctions was obtained from the Sydney Coordinated Adaptive Traffic System (SCATS) database in Dublin City Council for use in the time-series prediction models. In the SCATS system, the total number of vehicles passing over inductance loop detectors over a given time interval is measured at different junctions around the city. For this chapter, the data is taken from three junctions across the city. These junctions are TCS141, TCS166 and TCS183.
on the SCATS database in Dublin City Council. The locations of these three junctions in Dublin city centre are shown on the map in Figure 2.8.

![Map of Urban Arterial Junctions](image)

**Figure 2.8: Map of Urban Arterial Junctions**

Junction TCS141 in the north of the city centre is the junction of Parnell St. and Gardiner St. Junction TCS166 is the junction of Gardiner St. and Sean McDermott St. Junction TCS183 is the junction of Tara Street & Butt Bridge with Burgh Quay. In all three junctions, the traffic exiting northbound is that which is modelled in this chapter.
The traffic flows recorded on a weekday are substantially different to the same flows recorded on the weekend. Therefore it is important to model the weekday and weekend traffic behaviours separately. Only weekday traffic observations are taken into account in the models in this work. This will ensure an increased accuracy because the networks will not need to take into account the change in the peak hour traffic flow patterns at weekends. Following a similar procedure as described in this chapter, networks could be set up to deal with only weekend traffic flow data and could predict weekend traffic flows accurately. Figure 2.9 shows the typical hourly aggregate traffic flow levels for 5 consecutive weekdays starting from Wednesday 11th June 2008 on each of the three junctions.

Figure 2.9: Typical Hourly Traffic Flow at Junctions TCS 141, TCS 166 and TCS 183

The trend of increased traffic volume during peak hours while commuters are travelling to and from work is clearly visible on the graph. Typically the values are highest around 08:00 hours and 17:00 hours each day as commuters travel to and from work respectively, while the
junctions are quietest from midnight each day until approximately 06:00 hours the next morning. It is evident from the figure that the peak periods occur at different times of the day in different junctions. Traffic flow data at 5 minute, 15 minute, 30 minute and 1 hour aggregate time intervals have been modelled using the ANN algorithms in an effort to discern the most suitable prediction horizon to be presented to motorists through ITS. Table 2.1 contains statistical information about the traffic flow data at the three modelled junctions. The mean and standard deviation of the modelled traffic flow time-series datasets are presented along with the coefficient of variation in this table.

Table 2.1: Urban Arterial Traffic Flow Characteristics

<table>
<thead>
<tr>
<th></th>
<th>J 141</th>
<th>J 166</th>
<th>J 183</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM Average Peak Hour</td>
<td>09:00 - 10:00</td>
<td>10:40 - 11:40</td>
<td>11:00 - 12:00</td>
</tr>
<tr>
<td>PM Average Peak Hour</td>
<td>17:30 - 18:30</td>
<td>17:00 - 18:00</td>
<td>15:10 - 16:10</td>
</tr>
<tr>
<td>5 Minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 5 Minutes) (μ)</td>
<td>81.49</td>
<td>53.82</td>
<td>109.04</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>45.02</td>
<td>26.43</td>
<td>59.25</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.55</td>
<td>0.49</td>
<td>0.54</td>
</tr>
<tr>
<td>15 Minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 15 Minutes)(μ)</td>
<td>245.85</td>
<td>162.55</td>
<td>328.64</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>121.78</td>
<td>75.48</td>
<td>173.24</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.50</td>
<td>0.46</td>
<td>0.53</td>
</tr>
<tr>
<td>30 Minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 30 Minutes)(μ)</td>
<td>491.70</td>
<td>325.09</td>
<td>657.28</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>228.76</td>
<td>148.36</td>
<td>342.52</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.47</td>
<td>0.46</td>
<td>0.52</td>
</tr>
<tr>
<td>1 Hour</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 3 Hours)(μ)</td>
<td>1010.00</td>
<td>669.90</td>
<td>1350.50</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>439.19</td>
<td>293.25</td>
<td>664.89</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.43</td>
<td>0.44</td>
<td>0.49</td>
</tr>
</tbody>
</table>

The traffic data at these junctions is non-stationary, due to its seasonal nature, as can be seen in Figure 2.9. Figure 2.10(a) is the correlogram, or autocorrelation graph, of unfiltered, non-stationary 15 minute traffic flow data from junction TCS141. Five days of data (96 points per day) are presented in the figure and the repeating pattern as each day passes can be seen clearly from the graph, displaying the strong correlation present in the data.
The data needs to be filtered to achieve stationarity. Seasonal filtering is used to remove the daily seasonality typically found in traffic behaviour over weekdays. The autocorrelation plot of this filtered, and now stationary, data is then used to determine the most influential data points.
that are to be used to predict traffic flow in the future. Autocorrelations for all junctions and
time aggregations are calculated based on this time-lagged data. As an illustrative example, the
correlogram for one day of 15 minute aggregate seasonally filtered data from junction TCS141
is presented in Figure 2.10 (b).

Based on the graphs in Figure 2.10, the influential points were chosen as those points
with an ACF value greater than 0.2. It is expected that these are the points which have the most
influence on the near-term future prediction of traffic flow data. These points can be seen in
Figure 2.10 (b) as those outside the parallel lines at 0.2 and -0.2 on the y-axis (points 1, 2 and
96 in the case of TCS 141 15 minute aggregate data). Choice of influential points helps to
reduce the size of input dataset considerably. It is expected that this procedure will reduce the
computational time. It is assumed that any point with ACF value less than +/- 0.2 will not have
any significant influence in prediction of the future traffic volumes. The influential points
discerned from studying the autocorrelation graphs of junctions TCS 141, TCS 166 and TCS 183
are listed in Table 2.2.

Table 2.2: Urban Arterial Influential Points based on Correlograms

<table>
<thead>
<tr>
<th>Time Aggregation</th>
<th>5 min</th>
<th>15 min</th>
<th>30 min</th>
<th>1 hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>J141</td>
<td>2, 4, 6, 8, 10, 12, 14, 16, 288</td>
<td>1, 2, 96</td>
<td>1, 2, 48</td>
<td>1, 2, 3, 4, 24</td>
</tr>
<tr>
<td>J166</td>
<td>2, 3, 5, 7, 9, 11, 288</td>
<td>1, 2, 96</td>
<td>1, 28, 48</td>
<td>1, 2, 3, 4, 24</td>
</tr>
<tr>
<td>J183</td>
<td>1, 2, 4, 6, 288</td>
<td>1, 2, 4, 6, 96</td>
<td>1, 2, 48</td>
<td>1, 2, 3, 4, 5, 24</td>
</tr>
</tbody>
</table>

2.3.2 Motorway Traffic Flow Data

The motorway traffic data modelled is obtained from the Motorway Incident Detection and
Automated Signalling (MIDAS) database of the U.K. Highways Agency. The MIDAS database
consists of traffic condition related information from all the major motorways in the United
Kingdom. MIDAS datasets contain traffic volume, average speed, headway and occupancy
observations aggregated over every 60 seconds in every lane of a given motorway section. For
this chapter, 10 days of traffic data from January 2010 are collected from two motorway sites, titled “A1/9340A” and “M6/6615A”. There are two lanes in each direction in the motorway sections chosen. In both cases, traffic observations from the northbound lanes of the chosen motorway sections are modelled. Maps of these two motorway locations are provided in Fig. 2.9.

Figure 2.11: Map of Motorway Locations

Site “A1/9340A” shown in Fig. 2.11(a) is a northbound section of the A1 motorway heading away from Doncaster, England. This motorway section is located just south of Junction 40, Darrington, which is just south of the junction with another motorway, the M62. Section “M6/6615A” shown in Fig. 2.11(b) is a northbound section of the M6 motorway, located west of Stoke-on-Trent, England. This section of the M6 is just north of Junction 16 where the M6 meets the A500, a major primary road. Fig 2.12 shows typical hourly traffic volumes for 5 consecutive weekdays at both locations.
Figure 2.12: Typical Hourly Traffic Flow at the A1 and M6 Motorways

The A1 motorway contains two strongly defined peaks each day, whereas the flow is spread more evenly throughout the day on the M6 motorway. Site specific information for both motorways is listed in Table 2.3.

Table 2.3: Motorway Traffic Flow Characteristics

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM Average Peak Hour</td>
<td>07:30 - 08:30</td>
<td>07:30 - 08:30</td>
</tr>
<tr>
<td>PM Average Peak Hour</td>
<td>17:00 - 18:00</td>
<td>15:30 - 16:30</td>
</tr>
<tr>
<td>5 Minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 5 Minutes) (μ)</td>
<td>128.4545</td>
<td>91.5677</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>81.3671</td>
<td>46.8197</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.6334</td>
<td>0.5113</td>
</tr>
<tr>
<td>15 Minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 15 Minutes) (μ)</td>
<td>385.3635</td>
<td>274.7031</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>240.4777</td>
<td>138.4716</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.6240</td>
<td>0.5041</td>
</tr>
<tr>
<td>30 Minutes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / 30 Minutes) (μ)</td>
<td>770.7271</td>
<td>549.4063</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>478.5517</td>
<td>275.4134</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.6210</td>
<td>0.5013</td>
</tr>
<tr>
<td>1 Hour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (Vehicles / Hour) (μ)</td>
<td>1541.5000</td>
<td>1098.8000</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>948.5544</td>
<td>547.5634</td>
</tr>
<tr>
<td>Coefficient of Variance (σ/μ)</td>
<td>0.6153</td>
<td>0.4983</td>
</tr>
</tbody>
</table>
As with the urban arterial traffic flow data, the motorway data is initially non-stationary. This can be seen in the correlogram in Fig 2.13(a), using 5 days of 15 minute aggregate data taken from the A1 motorway.

Figure 2.13: Correlogram of (a) Non-Stationary and (b) Stationary A1 Motorway 15 Minute Data
Fig 2.13 (a) displays the same traits as those seen in Fig 2.10 (a) i.e. peaks and troughs slowly reducing over time. Hence, the motorway data must be also made stationary, as was the case with the urban arterial traffic flow data. The influential points are chosen based on Fig 2.13 (b), using the same method as in Section 2.3.1. The influential points for the A1 and M6 motorways data are shown in Table 2.4.

Table 2.4: Motorway Data Influential Points based on Correlograms

<table>
<thead>
<tr>
<th>Time Aggregation</th>
<th>A1</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 min</td>
<td>1, 2, 3, 288</td>
<td>1 - 154, 165, 288</td>
</tr>
<tr>
<td>15 min</td>
<td>1 - 19, 93, 94, 95, 96</td>
<td>1 - 46, 50, 68 - 83, 96</td>
</tr>
<tr>
<td>30 min</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 48</td>
<td>1 - 25, 34 - 42, 48</td>
</tr>
<tr>
<td>1 hour</td>
<td>1, 2, 3, 4, 5, 6, 24</td>
<td>1 - 21, 24</td>
</tr>
</tbody>
</table>

The use of two separate and fundamentally different traffic flow datasets in this chapter enables broader conclusions to be made about the effectiveness of the forecasting models. Urban arterial traffic flow is likely affected by traffic signals and congestion, which results in noticeable stop and go patterns of traffic behaviour. Motorway traffic flow however is primarily uninhibited and motorists will often cruise along in free flow conditions near the speed limit. If a forecasting model can produce accurate predictions for both of these inherently different datasets, it suggests the model can generalize well and generalisation is always a desired trait in any forecasting model.

As a final point, the existence of outliers should be considered. In the two previously described datasets, some minor outliers are present but remedial action is not taken on the datasets. Instead, the outliers are included with the datasets and the prediction models are tested on the true traffic flow datasets in order to give a real indication of the model’s accuracy were it to be put to use in an ITS situation using live traffic flow data.

2.4 Results

Three different ANN models, the RBFNN, GRNN and FFBPNN, along with a SVM model were trained on traffic flow data of four different time aggregations (5 minutes, 15 minutes, 30
minutes and 1 hour) in order to produce one day's worth of future traffic flow forecasts at these different time aggregations. Ten days of traffic flow data were used to train each network and hence, the eleventh day was predicted by the networks. Performance of the prediction models was evaluated by comparing the prediction results with the actual observations of the day. It is important to note that the training datasets are of different sizes for different time-aggregation levels i.e. 5 minute aggregate dataset has 2880 points (288 points for each day over 10 days), 15 minute aggregate dataset has 960 points, 1 hour aggregate dataset has 240 points and 3 hour aggregate dataset has 80 points. However, only the number of predefined influential points were used as input to the ANN algorithms (Table 2.2). As an example, for 5 minute aggregate traffic volume from junction TCS166, 7 points were used for one step ahead prediction. This compares favourably with the conventional methods where all 288 points are used for the same. The particular setup of the three network algorithms is discussed in the next subsection. 12 different networks (4 time aggregations at 3 junctions) were developed using each different ANN algorithm.

2.4.1 Artificial Neural Network and Support Vector Machine Modelling Setup

This section contains descriptions of the model setups specific to the ANN and SVM forecasting algorithms used in this chapter. All the forecasts in this thesis were computed in Matlab, with the ANN algorithms using the built in NN toolbox and the SVM model using the Steve Gunn toolbox to compute forecasts.

_Feed Forward Back Propagation Neural Network_

Based on the theory discussed in Section 2.2.1, all 12 FFBPNN algorithms developed in this work used a log-sigmoid transfer function in the hidden layer, with a linear function in the output layer. The number of neurons chosen for the hidden layer depended upon the time aggregation of the data; less neurons were used as the time aggregation increased from 5 minutes to 3 hours, but all networks had a single neuron in the output layer. The networks were trained using the LM learning algorithm mentioned in Section 2.2.1.
Radial Basis Function Neural Network

As described in Section 2.2.2, the main unique component of a RBFNN, compared with other ANNs, is the presence of radial basis activation functions in the hidden layer. The setup of this network is based on choosing correct parameters for the Gaussian centres. Therefore, the main difference between the various network configurations over different time aggregations in the case of the RBFNN is the choice of Gaussian width. Gaussian width was set at smaller values for the smaller time steps i.e. the smallest value of Gaussian width used was for the 5 minute data, whereas the 3 hour data required the largest value of Gaussian width. The centres of the networks were created from the input data, with each input becoming a centre (or first layer weight). The first layer biases were set to 0.8326 divided by Gaussian width, resulting in Gaussian functions that cross 0.5 at weighted inputs of +/- Gaussian width. The pseudo inverse technique was then performed to determine the weights and bias of the second layer. This technique is a general way to find the solution to the system of linear equations including the first layer weights.

Generalized Regression Neural Network

Section 2.2.3 described how the main difference between the GRNN and the RBFNN is the way they establish the values of the centres and weights. Aside from this difference, the network setup follows a similar procedure to that of the RBFNN. Hence, again the major variable parameter in GRNN design is Gaussian width. The same pattern used in the RBFNN design was used here i.e. the larger values of Gaussian width were reserved for the larger time steps in order to generate a more accurate prediction.

Support Vector Machine

A linear kernel was chosen along with the $\varepsilon$-insensitive loss function. The insensitivity was set to 0 prior to prediction and the upper bound was treated as infinity. The number of support vectors, the bias term and the difference of Lagrange multipliers were all calculated as part of
the SVM process i.e. they were defined after calculation was done using a linear kernel and the
time-aggregation dependant input traffic flow time-series dataset. These choices are primarily
referred to as the default options in the toolbox and so they allow for useful comparisons
between the SVM predictions and the predictions of the three ANN forecasting models.

2.4.2 Comparison of Predictions

The following figures display how closely the predicted traffic flow values matched the targeted
values at the different junctions and time aggregations using each prediction algorithm. The
targeted traffic flow data is blue in all the graphs, while the predicted traffic flow values are red.

![Figure 2.14: FFBPNN Predictions at Junction TCS 141](image)

Figure 2.14: FFBPNN Predictions at Junction TCS 141
Figure 2.15: FFBPNN Predictions at Junction TCS 166

Figure 2.16: FFBPNN Predictions at Junction TCS 183
Figure 2.17: FFBPNN Predictions at the A1 Motorway

Figure 2.18: FFBPNN Predictions at the M6 Motorway
Figure 2.19: RBFNN Predictions at Junction TCS 141

Figure 2.20: RBFNN Predictions at Junction TCS 166
Figure 2.21: RBFNN Predictions at Junction TCS 183

Figure 2.22: RBFNN Predictions at the A1 Motorway
Figure 2.23: RBFNN Predictions at the M6 Motorway

Figure 2.24: GRNN Predictions at Junction TCS 141
Figure 2.25: GRNN Predictions at Junction TCS 166

Figure 2.26: GRNN Predictions at Junction TCS 183
Figure 2.27: GRNN Predictions at the A1 Motorway

Figure 2.28: GRNN Predictions at the M6 Motorway
Figure 2.29: SVM Predictions at Junction TCS 141

Figure 2.30: SVM Predictions at Junction TCS 166
Figure 2.31: SVM Predictions at Junction TCS 183

Figure 2.32: SVM Predictions at the A1 Motorway
Tables 2.5 presents the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) values in predictions using autocorrelation with the different networks and time aggregations. Table 2.5 contains the results of predictions made using FFBPNN, RBFNN, GRNN and SVM models.

Figure 2.33: SVM Predictions at the M6 Motorway
Analysis of Table 2.5 leads to the finding that for the most part the FFBPNN model produced the most accurate forecasts. In the case of 5 min predictions, it can be seen that the GRNN model was the least accurate. However, the GRNN model was among the most accurate at most locations for 1 hour predictions. These results, with differing models excelling at different time aggregations, confirm that no single model is clearly more adept at predicting future traffic flow at all time aggregations as compared to others. Another finding, in the case of the FFBPNN model...
models particularly, is that the MAPE generally reduced as the time aggregation of the data lengthened, as can be seen in the top left plot of Figure 2.34.

**Figure 2.34: MAPE vs. Time Aggregation Subplot**

In the case of the FFBPNN models, the MAPE always reduced as time aggregation level increases, apart from a single anomaly in the M6 dataset. The figure shows clearly that the largest decrease in MAPE occurred between the 5 minute and 15 minute time aggregations. The MAPE of the FFBPNN models used at each location decreased at approximately the same speed, with the models following roughly the same trajectory of diminishing decreases with increased time aggregation. The MAPE values specific to the RBFNN prediction models are more inconclusive when it comes to identifying trends in the data. This can be seen in the 'up
and down' nature of MAPE vs. time aggregation in the top right graph of Fig 2.34. It is also immediately noticeable from the graph that the MAPE associated with data from the M6 is far greater than the other four locations studied. Another interesting point is that three of the five locations (TCS 183, A1 and M6) had their minimum error at the 15 minute time aggregation, while the graph mirrored the behaviour of the FFBPNN prediction algorithms in that the largest difference in MAPE values is between the MAPE at 5 minute and 15 minute time aggregations. The MAPE values in Table 2.7 show that the trend of decreasing MAPE with increasing time aggregation was also present in the GRNN forecasting models. The exact behaviour of MAPE with relation to time aggregation can be seen in the bottom left sector of Fig 2.34. This graph again suggests that the difference between 5 minute and 15 minute predictions was the largest difference between any consecutive time aggregations. It can also be seen that the behaviour was different than that of either the FFBPNN or the GRNN prediction models, although the overriding trend is that MAPE decreased as time aggregation increased. The final graph of MAPE vs. time aggregation is that of the SVM model in the bottom right of Fig. 2.34. The SVM graph displays a different type of behaviour to the ANN models. The big decrease in MAPE from 5 minute aggregate data to 15 minute aggregate data was again present, but two of the five models showed an increase in error from 15 minute aggregate data to 30 minute aggregate data, and four of the five models predicted hourly flow less accurately than traffic flow aggregated at 30 minute intervals.

The overall trend from the four prediction models was that coarser time aggregation generally meant greater prediction accuracy than finer time aggregation. This is consistent with the fact that there is likely to be much greater variability between two 5 minute periods on different days than there is likely to be a big difference between two 3 hour periods on different days. This can be explained by the fact that the Coefficient of Variation (COV), the ratio of standard deviation to mean, decreased as the time aggregation increased (Tables 2.1 and 2.3). The greater variability present in 5 minute periods represented a tougher prediction problem,
hence as the time aggregation increased, the error generally decreased. However, the decrease was not always linear or monotonous in nature.

An additional consideration when forecasting traffic flow is that of the time it takes for each model to compute a prediction. Hence, Table 2.6 contains the computational time (in seconds) of the different models at each time aggregation for comparison.

<table>
<thead>
<tr>
<th>Model</th>
<th>5 Min</th>
<th>15 Min</th>
<th>30 Min</th>
<th>1 Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FFBPNN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>7.40</td>
<td>1.60</td>
<td>1.04</td>
<td>2.28</td>
</tr>
<tr>
<td>166</td>
<td>2.71</td>
<td>5.69</td>
<td>1.00</td>
<td>0.86</td>
</tr>
<tr>
<td>183</td>
<td>2.33</td>
<td>5.39</td>
<td>0.83</td>
<td>2.19</td>
</tr>
<tr>
<td>A1</td>
<td>10.83</td>
<td>27.13</td>
<td>0.94</td>
<td>7.58</td>
</tr>
<tr>
<td>M6</td>
<td>2.53</td>
<td>37.22</td>
<td>64.15</td>
<td>5.12</td>
</tr>
<tr>
<td><strong>RBFNN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>13.44</td>
<td>0.75</td>
<td>0.19</td>
<td>0.12</td>
</tr>
<tr>
<td>166</td>
<td>13.72</td>
<td>0.57</td>
<td>0.19</td>
<td>0.12</td>
</tr>
<tr>
<td>183</td>
<td>15.01</td>
<td>0.70</td>
<td>0.18</td>
<td>0.12</td>
</tr>
<tr>
<td>A1</td>
<td>15.22</td>
<td>0.79</td>
<td>0.25</td>
<td>0.14</td>
</tr>
<tr>
<td>M6</td>
<td>39.35</td>
<td>1.06</td>
<td>0.28</td>
<td>0.13</td>
</tr>
<tr>
<td><strong>GRNN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>1.91</td>
<td>0.25</td>
<td>0.19</td>
<td>0.09</td>
</tr>
<tr>
<td>166</td>
<td>1.72</td>
<td>0.28</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>183</td>
<td>2.15</td>
<td>0.25</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>A1</td>
<td>2.21</td>
<td>0.27</td>
<td>0.14</td>
<td>0.16</td>
</tr>
<tr>
<td>M6</td>
<td>17.61</td>
<td>0.49</td>
<td>0.15</td>
<td>0.13</td>
</tr>
<tr>
<td><strong>SVM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>329.87</td>
<td>27.59</td>
<td>6.37</td>
<td>1.65</td>
</tr>
<tr>
<td>166</td>
<td>320.51</td>
<td>28.88</td>
<td>6.39</td>
<td>1.66</td>
</tr>
<tr>
<td>183</td>
<td>332.46</td>
<td>29.19</td>
<td>6.34</td>
<td>1.60</td>
</tr>
<tr>
<td>A1</td>
<td>320.83</td>
<td>30.46</td>
<td>6.91</td>
<td>1.60</td>
</tr>
<tr>
<td>M6</td>
<td>346.37</td>
<td>28.10</td>
<td>6.61</td>
<td>1.64</td>
</tr>
</tbody>
</table>

It is expected that the finer resolution data should take longer to model than the coarser resolution data i.e. the 5 minute time aggregation data predictions should be the forecasts which
take the longest time to compute, due to the fact that there are many more points per day. This trend of fine data resolution resulting in relatively long computational time can be seen clearly in Table 2.6. For comparison sake, the number of points per day in 5 minute data is 288 points, 15 minute data contains 96 points per day, 30 minute data has 48 points per day, and hourly data consists of 24 points per day. The table is presented in order of increasing time aggregation.

The computational times displayed in Table 2.6 clarify the point that finer resolution data took longer to model than coarser resolution data. Also, it can be seen that in almost all cases, the SVM model took the longest to compute a result. However, there was very little trial and error associated with the SVM model, whereas the FFBPNN models had to be tested for the optimal number of neurons in the hidden layer, and both the RBFNN and GRNN models had to be tested for the best value of Gaussian width. In this sense it is hard to truly compare the models. The SVM computational time decreased dramatically with increasing time aggregation i.e. there was a correlation between the number of input points and the time taken for the SVM model to compute. The other three models computed predictions reasonably quickly at each time aggregation, once the background optimising had been performed. In summary, all four prediction models computed predictions in an adequate time interval.

A final consideration in this chapter is the issue of quantifying the improvement in prediction accuracy brought by the introduction of autocorrelation in the ACNN models. The following figure shows how the models using ACF (ACNN) outperform the basic prediction models. The figure focuses on 15 minute aggregate data from TCS 141 using an ACNN and a standard FFBPNN.
The graphs in Fig 2.35 show that the ACNN outperformed the standard FFBPNN structure. This figure was used as an illustrative example but the property is common to all ANN structures. Table 2.7 contains the MAPE and RMSE values for the ACNN and the standard FFBPNN.

Table 2.7: ACF vs. Non-ACF Prediction Error Values

<table>
<thead>
<tr>
<th></th>
<th>MAPE (%)</th>
<th>RMSE</th>
<th>Computational Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACNN</td>
<td>10.31</td>
<td>23.11</td>
<td>0.64</td>
</tr>
<tr>
<td>Standard FFBPNN</td>
<td>13.14</td>
<td>26.89</td>
<td>1.34</td>
</tr>
</tbody>
</table>

From Table 2.7, it can be seen that the accuracy advantage held by the ACNN model over the standard FFBPNN model was significant. In addition to this, the computational time was approximately twice as quick using the ACNN model compared to the standard FFBPNN model. It should be noted that the computational times in Table 2.7 are not directly comparable with those in Table 2.6 as a computer with a faster processor was used for the computational...
As a final comparison, an ACNN forecast using FFBPNN is displayed alongside a traditional Naïve forecast and a prediction computed using the Moving Average (MA) technique in Fig. 2.36. For clarity, the Naïve forecasting method operates on the assumption that what happens on Day (x) will repeat on Day (x + 1). The Moving Average technique predicts that the average behaviour of traffic flow on Days (x to x+4) will be the traffic flow situation on Day (x + 5). Again, the data used for this comparison was 15 minute aggregate traffic flow data from TCS 141.

Figure 2.36: TCS 141 Predictions - ACNN vs. Naïve Method vs. Moving Average

Figure 2.36 shows how the FFBPNN model using autocorrelation easily outperforms standard predictive techniques such as the Naïve method and the MA technique. This finding is important because further advances to the ACNN model are discussed throughout the thesis and...
it is clear that already the ACNN model discussed in this chapter clearly outperforms basic prediction algorithms. The specific forecasting error measures are displayed in Table 2.8.

Table 2.8: TCS 141 15 Min Predictions- ACNN vs. Naive Model vs. MA technique

<table>
<thead>
<tr>
<th></th>
<th>MAPE (%)</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACNN</td>
<td>10.7940</td>
<td>25.1874</td>
</tr>
<tr>
<td>Naive Model</td>
<td>12.6323</td>
<td>27.4747</td>
</tr>
<tr>
<td>Moving Average</td>
<td>13.6584</td>
<td>35.3820</td>
</tr>
</tbody>
</table>

2.5 Conclusion

Three different ANN algorithms and a SVM prediction model were used to predict traffic flow of four different time aggregations. The traffic flow data came from three junctions in the city centre of Dublin, Ireland and two motorways in England. Seasonal filtering and autocorrelation were applied to the traffic data in order to reduce the dataset sizes. The main conclusion of the work in this chapter is that the application of autocorrelation to filtered, stationary traffic data can result in a more accurate forecast than a prediction without autocorrelation. In some cases, the improvement in prediction accuracy is vast, while in other cases the improvement is comparatively small. Importantly, all four prediction algorithms produce forecasts of greater accuracy when using the influential points defined by the ACF procedure, as opposed to predicting using the entire dataset.

Additionally, the computation time is hugely improved with the use of ACF based input point selection, due to the reduced number of input points. The maximum improvement is in the case of large datasets such as the 5 minute aggregate traffic flow time-series observations. In general, the higher the resolution of the data, the bigger the training dataset and the higher the computational time. However, using an ACF based procedure, as described in this chapter, it is found that the size of the training dataset is dependent on the autocorrelation and not the resolution of traffic volume observations. Therefore, the computational time is generally lower in the case of the ACF based procedure.
Among the models studied, the FFBPNN algorithm produced the most accurate forecasts in the majority of cases. The RBFNN and GRNN models also produced forecasts with very acceptable MAPE values, but they performed worst of all the models when predicting 5 minute time aggregation traffic flow data. The SVM prediction algorithm was very competitive at all time aggregations except for the data in hourly intervals. Here it showed the unusual behaviour of MAPE and RMSE values rising from 30 minute to hourly time aggregation, a property not shared by any of the other models. Therefore, based on the results at different time aggregations and locations, the FFBPNN would be the most recommended prediction algorithm in this case. Based on these findings, the FFBPNN model using autocorrelated input vectors is used as the primary forecasting algorithm throughout the rest of this thesis. Henceforth, ACNN refers to such a model structure. For completeness, the ACNN model was also shown to outperform standard prediction models such as the Naïve method and the Moving Average technique.

The location of the traffic flow data modelled did not bear any direct correlation with the MAPE and RMSE results recorded from each location. However, in general the A1 motorway data gave the most accurate predictions, while the M6 motorway was often the least accurate in terms of prediction. Amongst the urban arterial datasets, TCS 166 was generally the location of least accurate predictions. TCS 141 and TCS 183 were closely matched with TCS 141 outperforming TCS 183 with certain prediction models and time aggregation and vice versa. Thus, it can be concluded that the type of data does have an impact on prediction accuracy, with different junctions offering different prediction challenges, but based on the 5 locations in this work, it is inconclusive as to whether urban arterial or motorway traffic flow data is likely to generate a more accurate prediction. The results suggested that there is no simplistic view whereby urban arterial junctions are more or less difficult to predict than motorways, the accuracy is simply linked to the behaviour of the traffic at a given location, and accurate forecasts can be computed at both urban arterial junctions and motorway locations.
The chapter also showed that time aggregation impacts directly on the MAPE values of the prediction algorithms. It was found that the shorter the time aggregation, the poorer the forecast accuracy, whether using FFBPNN, RBFNN, GRNN or SVM forecasting algorithms. However the error accuracy is not linearly proportional with time aggregation. As mentioned previously, one exception to this finding is that the hourly forecasts with SVM models were less accurate than the respective 30 minute aggregation predictions.

The prediction results show that the error decreases with increasing time aggregation. However, it is not plausible to establish an optimal data aggregation interval without knowledge of the purpose. Hence, the choice of optimal time aggregation lies with the traffic managers who must find a balance between data resolution, error accuracy, prediction horizon requirement and computational time in order to decide which time aggregation is the most suitable for their needs. There is ample scope to expand this research further using higher resolution data to establish a greater understanding of the relationships between the aforementioned factors.
Chapter 3

A REGIME BASED SHORT-TERM MULTIVARIATE TRAFFIC CONDITION VARIABLE FORECASTING ALGORITHM USING NEURAL NETWORKS WITH ADAPTIVE LEARNING

3.1 Introduction

Traffic dynamics can be broadly classified into two regimes: congested and uncongested. From studying the fundamental diagram and plotting real time traffic flow vs. speed data in this work, it can be seen that uncongested regimes have approximately linear speed-flow relationships and congested regimes have approximately non-linear speed-flow relationships. ANN algorithms can model both linear and non-linear relationships. However, the same network structure and weights may not be able to predict traffic condition variables in both regimes with similar accuracy. The work in this chapter focuses on modelling regime identified traffic flow and speed data. Studies on modelling traffic speed are much less common than those involving traffic flows and the scarcity of literature on traffic speed forecasting is even more apparent for multivariate conditions. Hence, a regime-adjusted multivariate dual-traffic flow and speed prediction algorithm is proposed in this chapter.

The proposed regime adjustment methodology in this chapter utilises a time-series classification approach to isolate the observations in congested or non-linear regimes and then subsequently preprocesses such information for further prediction. This approach is on the basis of identification of non-linearity, and not the order of non-linearity, and hence does not require segmenting the traffic data into multiple regimes. The simplicity of the model enhances its ease of implementation. The effectiveness of the proposed methodology in predicting freeway traffic condition variables is explored. The multivariate short-term traffic condition variable prediction model proposed in this chapter utilises the ACNN structure in conjunction with adaptive learning rules. In this regard, four different learning algorithms are used and compared in this chapter. These include a simple GD algorithm, a GD algorithm with ALR, a GD algorithm with
momentum and a LM training algorithm. These ACNN models with four different learning rules are compared to quantify the extent of traffic condition variable prediction accuracy through the introduction of adaptive learning rules to FFBP algorithms. The ACNN structure with the adaptive LM training algorithm has been observed to be the most accurate traffic flow predictor in this work. The ACNN structure with GD was significantly outperformed, both in terms of error and computational time. This justifies the use of adaptive learning algorithms to train ACNN structures in developing traffic condition variable prediction models.

The chapter is organised into five sections. Following the Introduction, a description of the multivariate traffic prediction methodology is presented in Section 3.2. The traffic flow data used in modelling and the model fitting process are described in Section 3.3. Section 3.4 presents the results of the prediction algorithms along with a discussion of the prediction accuracies of the studied models. Finally, the chapter is concluded in Section 3.5.

3.2 Methodology

A regime-adjusted multivariate short-term traffic prediction methodology involving an ACNN algorithm with adaptive learning strategies has been utilised in this chapter to simultaneously predict traffic flow and average traffic speed on highways. A schematic of the proposed prediction methodology is shown in Fig. 3.1.
To explain the method simply and broadly, initially traffic observations from uncongested (assumed as linear) and congested (assumed as non-linear) regimes are separated and preprocessed. Following this, regime-adjusted flow and average speed are used as input to the ACNN structure to obtain forecasts. The prediction methodology is discussed in detail in the following subsections.

3.2.1 Regime Isolation Methodology

An important feature of the prediction methodology in this work is traffic regime isolation. In this subsection the strategy used for separating the congested (or non-linear) regime from the uncongested (or linear) regime is described. To illustrate the regime-isolation methodology, typical traffic flow and speed observation from a motorway site has been plotted in Fig. 3.2(a).
Apart from three points indicating high flow and low speed values, the rest of the observations illustrate a seemingly linear behaviour which can be assumed to be indicative of free-flow conditions. The linearity has also been confirmed quantitatively by fitting a line through the speed flow scatter-plot ($R^2 = 0.7$). The three points which do not follow linearity or more precisely introduces non-linearity in the flow-speed relationship can be identified as those belonging to a non-linear or congested regime. These observations can be isolated using a time-series pattern matching approach. The approach is defined and described in the following steps;
Step 1 – Time-series: A time-series dataset $T = t_1, \ldots, t_a$, an ordered set of $a$ real-valued variables is considered. The data comprising the time-series used in this work is discussed and examined in Section 3.3.1.

Step 2 – Subsequence: Given a time-series $T$ of length $a$, a subsequence $S_p$ of $T$ is a sample of length $z < a$ of contiguous positions from $T$, that is, $S_p = t_p, \ldots, t_{p+z-1}$ for $1 \leq p \leq a - z + 1$. The process of extracting subsequences from a time-series is achieved through the implementation of a sliding window.

Step 3 – Sliding Window: Given a time-series $T$ of length $a$, and a user-defined subsequence length of $z$, all possible subsequences can be extracted by sliding a window of size $z$ across $T$ and extracting each subsequence $S_p$. Using this process, the subsequence following $S_p$ is $S_{p+1} = t_{p+1}, \ldots, t_{p+z}$.

Step 4 – Pattern Definition: The pattern of each subsequence $S_p$ is characterised by the slope of the best-fit line generated for each subsequence $S_p$. The change of slope between consecutive subsequences is then measured to identify the similarity of the subsequences. A sample difference in slope graph is displayed in Fig 3.2 (b).

Step 5 – Pattern Similarity: The inclusion of a point in a time-series subsequence which falls outside the linear regime introduces non-linearity which results in a large change of slope in the linear behaviour of the subsequence and also provides a low $R^2$ value for the best-fit line, indicating non-linear behaviour. Using this assumption, the points introducing non-linearity in the traffic time-series data are identified.

Step 6 – Time-series Classification: A subsequence $S_p$ which is characterised by a $R^2$ value of beyond a defined lower limit and the difference of slope, $\theta$, between $S_p$ and $S_{p-1}$ is beyond a threshold T classifies the sequence as non-linear. The threshold $T$ is set as the lower limit of
the 95% confidence interval of a change of slope distribution calculated using historical time-series datasets. The lower limit of the $R^2$ value is defined in this case as the average value calculated using historical time-series datasets.

The abovementioned steps are followed to identify the traffic observations located in the congested or non-linear regime. The observations in the congested regime are preprocessed to introduce an adjusted surrogate linear speed-flow calibration to ensure consistent traffic dynamics. It can be seen, that each set of information on traffic volume, average speed and time of observation, $t_i$, is a vector that can be represented as $X(q,v,t_i)$. The observations belonging to the congested regime are adjusted using the following conditions:

\[
X(q,v,t_i) = \Lambda(X(q,v,t_i)), \quad i = 1,...,n
\]

where, $X(q,v,t_i)$ defines the traffic time-series at flow $q$, speed $v$ and time $t$ of a day; $n$ is the number of days at which traffic data is modelled, $\Lambda$ denotes the expectation of the behaviour of the traffic and $\Omega$ is a limit factor taking a different value dependent on the site. The observations in the uncongested regime are not preprocessed.

### 3.2.2 ACNN with Various Learning Rules

For a basic overview of ANNs, Section 2.2 of this thesis should be referred. The following subsections explain the structure of the ANNs used in the work in Chapter 3, building on the basics explained in Section 2.2. The ACNN model based on the FFBPNN structure, as described in Section 2.2.5, serves as the basic structure to which the various adaptive learning rules are applied in this work.

**a) Adaptive Learning Rate Model (ALRM)**

The BP algorithm, used to adjust connection weights, is sensitive to the learning rate parameter, $\eta$, as shown in Equation (2.4). This parameter which conventionally stays constant throughout
the process of optimisation, determines the stability and speed of convergence of the weight vector towards the minimum error value. If \( \eta \) is too large, the BP algorithm can oscillate and become unstable. Conversely, too small a learning rate can affect the convergence time negatively. These problems can be predominantly avoided by allowing the learning rate to be altered during the process of optimising the network. Hence, ALR is used here. The introduction of an ALR, as shown in Equation (3.2), allows the learning rate to be adjusted during the running of the BP algorithm.

\[
\eta(k+1) = \eta(k) \times \eta_{factor}
\]

\[
\eta_{factor} = \begin{cases} 
1.05, & \Delta E > 0 \\
0.7, & \text{otherwise} 
\end{cases}
\]  

(3.2)

where \( \eta_{factor} \) is an adaptive multiplicative factor which alters the value of \( \eta \) depending on how the error value has changed in the previous iteration and \( k \) simply represents the iteration. In this study, two values of \( \eta_{factor} \) are contained within the ALR algorithm; one which increases the value of \( \eta \) if the error has decreased over the previous iteration and one which decreases the value of \( \eta \) otherwise (Equation (3.2)). These values of \( \eta_{factor} \) in Equation (3.2) are set at values common throughout the literature so as to enable comparisons of prediction accuracy with the other models in this work. In this chapter, the model using ALR will be referred to as ALRM.

(b) Levenberg-Marquardt Model (ALRLM)

The second learning algorithm investigated in this chapter is the Levenberg-Marquardt (LM) algorithm (Levenberg, 1944 and Marquardt, 1963). This algorithm has been discussed previously in Section 2.2.1 and is described in Equations (2.5) to (2.7) inclusively. This LM algorithm is referred to as ALRLM in this chapter.
Momentum is an alternative approach to dealing with the problems of the BP algorithm documented in Section 2.2.1. The introduction of momentum (Equation (3.3)) adjusts the weight alteration as follows:

\[
\Delta w_y(k) = -\eta \frac{\delta E(k)}{\delta w_y(k)} + \gamma \Delta w_y(k-1)
\]  

(3.3)

where \( \gamma \) is the momentum factor, set at 0.9 in this work in accordance with standards in the NN forecasting literature. Momentum alters the connection weight updating procedure. The momentum factor determines the influence that the previous weight change has on the current weight updation. Momentum produces an averaging effect whereby changes in connection weights take into account both the current error value as well as previous weight changes, leading to a more considered approach to weight updation (Parra et al., 2009). This overall view afforded to momentum limits the amount a weight might be changed at any given time step. Momentum is generally seen as a method to improve the training time of ANNs (Attoh-Okine, 1999). In this research, the ACNN algorithm using momentum is presented as ALRMM.

### 3.3 Application of the proposed Prediction Algorithm

The prediction methodology described in Section 3.2 is evaluated using real-time traffic data from motorway sites. In the following subsections the description of the modelled traffic data and the model fitting procedure are described.

#### 3.3.1 Traffic Data

The data modelled using the proposed prediction methodology is obtained from the MIDAS database of the U.K. Highways Agency. For this chapter, 10 days of traffic data from January 2010 are collected from two motorway sites, titled “A1/9340A” and “M6/6615A” (Fig. 2.11). There are two lanes in each direction in the motorway sections chosen. In both cases, traffic
observations from the northbound lanes of the chosen motorway sections are modelled. These locations, and the data recorded at the locations, are discussed in depth in Section 2.3.2. It is important to note that the traffic flow observations recorded on a weekday are substantially different to those traffic observations recorded on the weekend. Therefore for accuracy purposes, only weekdays of traffic condition variable observations are modelled in this chapter.

Fig. 3.3 shows typical traffic flow levels for a weekday on each of the four lanes modelled.

![Figure 3.3: Weekday 15 Minute Aggregate Traffic Flow](image)

Typically the values are highest around 08:00 hours and 17:00 hours as motorists commute to and from work, while the roads are quietest from midnight until approximately 06:00 hours the
next morning. Fig. 3.4 displays the typical average speed of vehicles in the four lanes modelled on a weekday.

![Weekday 15 Minute Aggregate Average Speed](image)

**Figure 3.4: Weekday 15 Minute Aggregate Average Speed**

The values of average speed stay within narrower confines than that of the traffic flow. However, large variations of average speed over small time periods do occur, notably in Lane 2 of the M6/6615A motorway section. In Chapter 2, the motorway traffic flow data is treated as a whole i.e. the flow from both lanes was combined to give a “route flow” rather than the lane specific approach taken in this chapter. The reason for focusing on a lane-based forecasting algorithm rather than a link-based forecast is linked to the methodology described in Section 3.2.1. As this methodology relies heavily on the flow speed relationship inherent in motorway traffic, it was deemed prudent to examine the lanes individually. The flow speed relationship does indeed vary from a link at one site to a link at another site. However, the flow speed
relationship can also vary greatly between lanes at the same link. Thus, the reason behind using a lane based forecast was to take into account the varying behaviour between lanes, rather than smooth out this behaviour by combining lanes into a single link traffic flow and averages speed dataset. As such, Table 3.1 contains lane specific statistical descriptions of the traffic volume and speed observations as collected at the aforementioned four motorway lanes. The traffic flow peaks at different times at different motorway locations. The mean and standard deviation across the 10 day datasets are also presented in this table.

Table 3.1: Statistical Description of Traffic Flow & Average Speed Observations from Motorway Sites

<table>
<thead>
<tr>
<th>15 Minute Interval Data</th>
<th>A1/9340A</th>
<th>M6/6615A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lane 1</td>
<td>Lane 2</td>
</tr>
<tr>
<td><strong>FLOW</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (vehicles /15 minutes)</td>
<td>162.86</td>
<td>222.51</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>71.72</td>
<td>172.44</td>
</tr>
<tr>
<td>AM Average Peak Hour</td>
<td>07:30-08:30</td>
<td>07:30-08:30</td>
</tr>
<tr>
<td>PM Average Peak Hour</td>
<td>17:30-18-30</td>
<td>17:30-18-30</td>
</tr>
<tr>
<td>Coefficient of Variation</td>
<td>0.44</td>
<td>0.77</td>
</tr>
<tr>
<td><strong>SPEED</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (kilometres / hour)</td>
<td>92.97</td>
<td>108.67</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>3.50</td>
<td>6.43</td>
</tr>
<tr>
<td>AM Average Peak Hour</td>
<td>02:00-03:00</td>
<td>01:00-02:00</td>
</tr>
<tr>
<td>PM Average Peak Hour</td>
<td>22:00-23:00</td>
<td>22:00-23:00</td>
</tr>
<tr>
<td>Coefficient of Variation</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td><strong>THRESHOLDS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change of slope</td>
<td>-0.03</td>
<td>-0.06</td>
</tr>
<tr>
<td>R-Squared</td>
<td>0.76</td>
<td>0.73</td>
</tr>
</tbody>
</table>

One important point to note is that the COV values of the average speed datasets are much lower than that of the flow datasets. Assuming that COV is indicative of inherent randomness of positive valued random variables, it can be expected that average speed predictions should be more accurate than the traffic flow forecasts. Data aggregated over 15 minute intervals is used to calculate the statistics in Table 3.1, so as to relate to the diagrams of typical weekday flow and average speed in Fig. 3.3 and Fig. 3.4. Models have been fitted for five different time aggregations; 1 minute, 5 minute, 15 minute, 30 minute and 1 hour.
The prediction methodology described in Section 3.2 is heavily dependent on the flow-speed relationship among the bi-variate traffic time-series data (Fig. 3.5). This relationship can vary between links at two different sites as it varies between lanes within the same link. To ensure uniqueness in flow-speed relationship and for effective classification of time-series subsequences, a lane based forecasting has been performed in this study.

3.3.2 Model Fitting

The model fitting was performed in two stages. At the first stage the bi-variate traffic time-series datasets from 4 sites were analysed to identify the following:
• Length of a subsequence: The length of the subsequence which demonstrates linearity at level $R^2 > 0.5$ was identified using trial-and-error method. This length was different for different time aggregation interval. Taking 15 minute aggregated data as an example, an interval of 12 represented a length of the subsequence of 3 hours.

• Change of slope threshold: The training datasets were analysed to identify the change of slope threshold. The change of slope between consecutive traffic time-series subsequences were calculated over the entire dataset and the lower limit of the 95% confidence interval of the mean was taken as the threshold. The values of this threshold for the chosen sites are provided in table 1. For illustrative purposes, the thresholds calculated for 15 minute interval are shown in the table.

• $R^2$ value thresholds: Similar to the previous threshold, the training datasets were analysed to identify this threshold. The $R^2$ value was calculated for all subsequences over the entire dataset and the mean was taken as the threshold. The values of this threshold for the chosen sites are provided in Table 3.1. For illustrative purposes, the thresholds calculated for 15 minute interval data are shown in the table.

On identification of the thresholds, the points identified within the non-linear regime were preprocessed using Equation (3.1). Further to preprocessing, the traffic data was used as input to the ANN structures described in Section 3.2.2 for prediction.

At the second stage, 20 different ACNN structures in total had been used to predict traffic volume and speed simultaneously in a multivariate paradigm. The 20 models comprise 3 different adaptive learning algorithms and the original GD learning rule used to predict traffic flow and average speed at 5 different time aggregation intervals. The number of elements, $n$, in the input vector representing a single day of a given dataset, corresponds to the time aggregation level of the data; $n = 1440$ for 1 minute data, $n = 288$ for 5 minute data, $n = 96$ for 15 minute data, $n = 48$ for 30 minute data and $n = 24$ for hourly traffic observations. All 20 different ACNN algorithms used in this chapter have a log-sigmoid transfer function in the
hidden layer, along with a linear function in the output layer. The number of neurons chosen for
the hidden layer in each of these models depended upon the time aggregation level of the data,
with less neurons used as the time-aggregation increased from 1 minute to 1 hour. For all
networks the output vector is same size as the input vector.

The training datasets are composed of regime-adjusted traffic flows and average speeds
from 10 consecutive weekdays for all 20 models. Prior to input, the regime-adjusted training
datasets are preprocessed using min-max normalization. The importance of preprocessing or
filtering datasets prior to modelling with ANN structures is well documented in the literature
(Yun et al., 1998). Traffic observations from the first 7 days are compiled to create the training
input. Observations from day 2 to day 8 are used as the training target. Once the network is
trained on this training dataset, a new day (Day 9), previously unseen by the networks, is
presented as the test input. The simulation of the network with this test input results in the
prediction of traffic flow and average speed information as the test output i.e. a predicted
version of Day 10. This is then compared with the test target i.e. the true values of Day 10, and
error values are generated based on the difference between output and target.

3.4 Forecasts

The proposed prediction methodology is used to predict traffic flow and speed data at the four
chosen locations as described in the previous section. In this chapter, one day ahead forecasts
were generated for all data aggregation levels. At the different time aggregations, a one day
forecast contains a different number of predicted values; one day predictions for 1 minute data
contain 1440 points, 5 minute data contains 288 points per day, 15 minute data has 96 points
per day, 30 minute data has 48 points per day and hourly data has 24 points per day.

The 20 different ANN models with 4 different learning strategies were compared to
identify which of the models produced the most accurate forecasts, and also whether an increase
in prediction accuracy compromised the computational speed of the model. The MAPE, RMSE
values and computation time for all 20 different ANN models for all four sites are provided in Table 3.2 (traffic volume) and Table 3.3 (average traffic speed).

From both flow and average speed prediction errors, it can be ascertained that at the majority of time-aggregations, the GDM algorithm produces the least accurate forecasts as well as the longest computation times. The ALRM, ALRMM and ALRLM all produce competitive forecasts and computation times. The computation times mentioned in Tables 3.2 and 3.3 are the same as the prediction methodology is bivariate and produces simultaneous flow and average speed predictions. Overall the computation times for all algorithms decrease with increasing time aggregation levels, as the size of the input vector reduces accordingly. The one minute traffic data has an input vector with 1440 elements and consequently has the longest computation time in all cases. Conversely hourly traffic data has the shortest computation time with an input vector of 24 elements. It can be concluded that an increase in prediction accuracy does not compromise the computation speed from an algorithm perspective, rather the MAPE, RMSE and computation times all follow the same trend.
Table 3.2: Traffic Flow Prediction Results

<table>
<thead>
<tr>
<th></th>
<th>A1 Lane 1</th>
<th>A1 Lane 2</th>
<th>M1 Lane 1</th>
<th>M1 Lane 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GDM</td>
<td>ALRM</td>
<td>ALRMM</td>
<td>ALRLM</td>
</tr>
<tr>
<td>MAPE</td>
<td>53.79</td>
<td>37.35</td>
<td>36.91</td>
<td>33.63</td>
</tr>
<tr>
<td>RMSE*</td>
<td>28.44</td>
<td>15.62</td>
<td>14.42</td>
<td>14.20</td>
</tr>
<tr>
<td>Time</td>
<td>168.30</td>
<td>125.90</td>
<td>126.40</td>
<td>119.80</td>
</tr>
<tr>
<td>MAPE</td>
<td>18.78</td>
<td>10.50</td>
<td>10.11</td>
<td>8.90</td>
</tr>
<tr>
<td>5 Minute</td>
<td>134.00</td>
<td>99.40</td>
<td>99.94</td>
<td>96.59</td>
</tr>
<tr>
<td>RMSE*</td>
<td>9.74</td>
<td>3.72</td>
<td>3.63</td>
<td>2.26</td>
</tr>
<tr>
<td>MAPE</td>
<td>128.40</td>
<td>93.40</td>
<td>94.35</td>
<td>95.43</td>
</tr>
<tr>
<td>30 Minute</td>
<td>15.83</td>
<td>3.86</td>
<td>2.24</td>
<td>2.17</td>
</tr>
<tr>
<td>Time</td>
<td>105.69</td>
<td>84.32</td>
<td>81.48</td>
<td>80.39</td>
</tr>
<tr>
<td>1 Hour</td>
<td>8.93</td>
<td>2.22</td>
<td>2.58</td>
<td>2.34</td>
</tr>
</tbody>
</table>

*The unit of RMSE values are vehicles per hour and the unit of time is sec.
The unit of RMS values are kilometers per hour and the unit of time is sec.

<table>
<thead>
<tr>
<th>Time</th>
<th>1 Hour RMS</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>8.93</td>
<td>2.22</td>
</tr>
<tr>
<td>0.05</td>
<td>2.22</td>
<td>1.19</td>
</tr>
<tr>
<td>0.10</td>
<td>2.90</td>
<td>1.41</td>
</tr>
<tr>
<td>0.15</td>
<td>2.70</td>
<td>1.64</td>
</tr>
<tr>
<td>0.20</td>
<td>2.94</td>
<td>1.87</td>
</tr>
<tr>
<td>0.25</td>
<td>2.32</td>
<td>2.10</td>
</tr>
<tr>
<td>0.30</td>
<td>3.49</td>
<td>2.33</td>
</tr>
<tr>
<td>0.35</td>
<td>2.69</td>
<td>2.56</td>
</tr>
<tr>
<td>0.40</td>
<td>3.57</td>
<td>2.80</td>
</tr>
<tr>
<td>0.45</td>
<td>2.97</td>
<td>3.03</td>
</tr>
<tr>
<td>0.50</td>
<td>4.79</td>
<td>3.26</td>
</tr>
<tr>
<td>0.55</td>
<td>3.57</td>
<td>3.49</td>
</tr>
<tr>
<td>0.60</td>
<td>4.14</td>
<td>3.72</td>
</tr>
<tr>
<td>0.65</td>
<td>3.02</td>
<td>3.95</td>
</tr>
<tr>
<td>0.70</td>
<td>4.84</td>
<td>4.17</td>
</tr>
<tr>
<td>0.75</td>
<td>3.72</td>
<td>4.40</td>
</tr>
<tr>
<td>0.80</td>
<td>5.41</td>
<td>4.63</td>
</tr>
<tr>
<td>0.85</td>
<td>4.50</td>
<td>4.86</td>
</tr>
<tr>
<td>0.90</td>
<td>6.19</td>
<td>5.09</td>
</tr>
<tr>
<td>0.95</td>
<td>5.30</td>
<td>5.32</td>
</tr>
</tbody>
</table>

Table 3: Average Traffic Speed Prediction Results
For all 4 motorway sites, the MAPE and RMSE values presented in Tables 3.2 and 3.3 correlate with the COV values in Table 3.1. The traffic flow forecasts for A1/9340A Lane 2, with a COV value of 0.77, are the least accurate of all sites. The traffic speed forecasts follow a similar pattern with A1/9340A Lane 2 and M6/6615 Lane 2 having the least accurate prediction accuracies and the highest COV values. In general, the average speed MAPE and RMSE values are less than the corresponding flow MAPE and RMSE values in every instance. In a motorway situation, the average speed varies over a comparatively smaller range than the traffic flow as can be seen in Fig. 3.6 and quantitatively it is evident from much lower COV values for speed than for flow throughout the datasets. However, a direct correlation between MAPE, RMSE and COV values should not be assumed. COV can only be considered as an approximate indicator of prediction accuracy.

For illustrative purposes, the prediction results using ALRLM at A1/9340A Lane 1 for 15 minute data interval have been shown in Fig. 3.6. In Fig 3.6(a) a flow-speed scatterplot of original observations and predicted values using the proposed methodology is plotted. The traffic flow and speed forecasts along with the original observations are shown in Fig. 3.6(b) and Fig. 3.6(c) respectively.
As documented in Tables 3.2 and 3.3, the proposed regime-adjusted ANN based prediction methodology provides accurate forecasts for the bivariate traffic time-series. The same traffic flow data when modelled using an ordinary univariate FFBPNN (without regime adjustment, any ALR element or autocorrelated input vectors), resulted in predictions with MAPE values equal to 20%. This proves that the proposed model provides better prediction accuracy through regime adjustment, learning rate improvement and the use of autocorrelated input vectors.
The MAPE (%) values for flow and speed predictions from regime-adjusted ALRLM algorithm are plotted against the time intervals in Fig. 3.7 for all 4 sites.

![Figure 3.7: Prediction Accuracy vs. Time Aggregations using ALRLM for 4 Motorway Lanes](image)

In all cases, the sharpest reduction in MAPE occurs between the finer (1 min, 5 min, 15 min) time aggregations. The change in MAPE values for time aggregations greater than 15 minutes is negligible compared to the finer aggregations. The curves for all 8 cases (flow and speed for all 4 sites) behave in the same fashion. At higher resolutions, there is an almost exponential decrease with increased aggregation, and at lower resolutions, the curve behaves asymptotically. This establishes that beyond a certain level, time aggregation does not influence the prediction levels greatly. The curves suggest that 15 minutes is the optimum forecasting time aggregation. However, more detailed analysis into time aggregation levels and signal-to-
noise ratios in traffic time-series is necessary to arrive at a definitive optimum time aggregation level.

3.5 Conclusion

Traffic conditions on motorways can generally be classified as being in one of two regimes: congested or uncongested. Traffic dynamics are a function of traffic condition variables such as flow and speed. A novel regime based multivariate traffic flow and speed forecasting algorithm is developed in this work to utilise the flow and speed relationship in an effort to predict traffic conditions accurately. This study features the application of adaptive learning strategies such as ALR or momentum to ANN based multiple traffic condition variable forecasting algorithms. The algorithm uses an ANN structure with three adaptive learning strategies as well as the conventional GD learning rule. The forecasts, the precision errors and the computation times of the ANN structure with four different learning rules were compared to establish the most efficient prediction strategy.

The ANN structure using the adaptive LM learning algorithm proves to be the best prediction methodology to predict traffic flow and average speed simultaneously. The proposed algorithm is capable of forecasting traffic data a day ahead. The traffic flow prediction accuracy is in the order of 7 per cent to 15 per cent and average speed predictions in the order of 3 per cent to 6 per cent for the cases studied. It can be concluded that the ALRLM algorithm produces the most accurate forecasts most often in the quickest time.

The main contribution of the proposed methodology lies in providing a regime adjustment methodology to predict traffic data. This algorithm isolates the traffic data in the congested or non-linear regime from the ones in the uncongested or linear regime and the isolated observations are preprocessed to achieve stable predictions through the above mentioned ANN structure. The proposed algorithm has been evaluated for modelling traffic observations on highways. As a future work, the prediction methodology can be applied to urban arterials with more significant congestion levels and also for the effect of traffic signals.
The study also identifies the effect of time-aggregation on MAPE values when predicting short-term traffic condition variables. Traffic flow prediction improves with the increase in aggregation time interval as the variability of the data reduces considerably with lower resolution. The same behaviour is exhibited in the average speed predictions, with the underlying trend being that as time-aggregation increases, the accuracy of prediction increases too. It is important to note that, the prediction errors do not have a linear relationship with the time aggregation levels. This relationship is generally site and flow dynamics specific.
4 Chapter 4

WEATHER ADAPTIVE TRAFFIC PREDICTION USING NEUROWAVELET MODELS

4.1 Introduction

Climate change is a prevalent issue facing the world today. One of the signatures of climate change is unexpected increase in rainfall, both in volume and intensity. Rainfall influences traffic demand and in turn traffic volume in urban arterials. The within-day traffic dynamics are altered or affected by rainfall events. With this in mind, a novel feature of the work in this chapter is the introduction of rain as an additional input to the ACNN prediction algorithms. The effect of rain on travel demand and traffic accidents has been discussed in the literature review. These findings regarding weather's effect on various traffic conditions suggest that rain has a definite impact on traffic flow. In this regard, it was thought of as imperative to develop a traffic flow forecasting model that takes into account the weather conditions of the forecasted time interval. It is urgent to develop this model because the presence of climate change decrees that many locations around the world will experience longer and more intense periods of inclement weather. Therefore, with the likelihood of more regular precipitation in many areas, it is necessary to be able to factor in the effect of various levels of precipitation on traffic flow conditions. However, this effect of rainfall on traffic conditions is not always immediate and the influence of rainfall on traffic volume is often unrecognizable in a direct correlation analysis between the two time-series datasets; it can only be observed at certain frequency levels. Accordingly, it is essential to employ a multiresolution prediction framework to develop a weather adaptive traffic forecasting algorithm.

In Chapters 2 and 3, traffic flow and traffic speed were predicted using ANN and SVM forecasting algorithms. The work in these chapters looked at the respective time-series as a whole and examined how preprocessing techniques such as autocorrelation can improve the
accuracy of traffic condition variable forecasting. In addition to improvements found in previous chapters, there has been some research to suggest that decomposing a time-series dataset (to its signal and noise components in essence) and predicting the different components separately can result in a prediction of greater accuracy. The task of breaking down a time-series is best accomplished through the use of wavelet decomposition by Stationary Wavelet Transforms (SWTs). However, the multiresolution structure of SWTs, involving independent modelling of the higher and lower resolution components, has yet to be exploited in an urban arterial traffic prediction framework. Therefore, in this chapter, the time-series are decomposed using SWTs, so that the individual components can be predicted separately by prediction algorithms specifically tuned to deal with the behaviour present in each component of the time-series. The use of SWTs in this manner ties in with the fact that, based on correlation analysis, the effect of rainfall on traffic conditions is best investigated at different frequency levels. As such, the effect of rainfall on traffic volume has been investigated at different resolution levels and incorporated in the prediction model accordingly. In summary, the work in this chapter involves using the multiresolution structure of SWTs to its fullest potential in developing a weather adaptive neuro-wavelet traffic forecasting algorithm which takes into account the effect of weather at different resolution levels.

The work in this chapter is organised into five sections. Following the introduction in 4.1, Section 4.2 describes the multiresolution weather adaptive traffic prediction methodology including the theory behind the SWT and how it is incorporated into an ANN structure. Section 4.3 discusses the traffic flow and precipitation data used in modelling. The results of the prediction algorithms are presented in Section 4.4, with a discussion on the prediction accuracies of the studied models. The chapter is concluded in Section 4.5.

4.2 Methodology

This section contains a brief description of the theory behind SWTs and a detailed description of the proposed neuro-wavelet prediction framework.
4.2.1 Stationary Wavelet Transform

The Discrete Wavelet Transform (DWT) (Mallat, 1989) is an implementation of the WT using dyadic scales and positions. The DWT of a time-series signal, \( x(i) \), can be described as the discretised form of the following:

\[
W_{\psi}(dil, tra) = \int_{-\infty}^{\infty} x(i) \psi_{dil, tra}(i) di
\]  
(4.1)

where,

\[
\psi_{dil, tra}(i) = \frac{1}{\sqrt{dil}} \psi\left(\frac{i - dil}{tra}\right)
\]  
(4.2)

Here, the wavelet \( \psi_{dil, tra}(i) \) is calculated from the mother wavelet, \( \psi(i) \), by dilation (governed by the dilation factor \( dil \)) and translation (governed by the translation parameter \( tra \)). The DWT process produces two different forms on decomposition; approximation and detail components. The discrete approximation coefficient at resolution \( 2^{-\rho} \) (\( \rho \) is the scale) is:

\[
a_{c, \rho, \Xi} = \int_{-\infty}^{\infty} 2^{-\rho/2} x(i) \phi^* \left( 2^{-\rho} i - \Xi \right) di
\]  
(4.3)

where, \( x(i) \) is the signal to be transformed, \( \phi_{\rho, \Xi}(i) \) is the scaling function at scale \( \rho \) and position \( \Xi \) and \( \phi^*(i) \) is the complex conjugation of the scaling function. The detail coefficients at the resolution \( 2^{-\rho} \) are obtained as:

\[
d_{c, \rho, \Xi} = \int_{-\infty}^{\infty} 2^{-\rho/2} x(i) \psi^* \left( 2^{-\rho} i - \Xi \right) di
\]  
(4.4)

The approximate component at level \( \rho \), \( a_{c, \rho} \), can be further decomposed into the approximation and detail components at the next level, \( \rho + 1 \). Equations (4.5) and (4.6) display the formulae required for extending the coefficient calculation at further decomposition levels for the approximation and detail cases respectively.

\[
a_{c, \rho + 1, \Xi} = \sum_{\Theta = -\infty}^{\infty} l(\Theta - 2\Xi) a_{c, \rho, \Theta}
\]  
(4.5)
\[ dc_{p+1, \Xi} = \sum_{\Theta=-\infty}^{\infty} h(\Theta - 2\Xi)ac_{p, \Theta} \]  

(4.6)

where, the functions \( l(\Theta) \) and \( h(\Theta) \) are the impulse responses of low-pass and high-pass paraunitary quadrature mirror filters respectively. At each level of decomposition, DWT downsamples the signal of the previous level. The presence of downsampling in DWT means that the approximate and detail coefficients of a signal \( x(i) \) is not the same length as the original signal. The shifting and dilation operations cause the loss of time-invariance. For all practical time-series modelling related purposes, it is important to achieve or retain the stationarity property of the signal. Therefore, methods have been devised to incorporate the property of shift-invariance in WT. In a DWT, the signal \( x(i) \) is convolved and decimated. This decimation removes the time-invariant structure. There exists, at every step of the decimation process, the choice of carrying out decimation by choosing either the odd indexed elements or the even indexed elements.
where: \* = convolution & \uparrow 2 = upsampling

Figure 4.1: Stationary Wavelet Transform Decomposition Tree

The SWT structure displayed in Fig. 4.1 however, can calculate all possible DWT decimations (i.e. even or odd indexed elements at each level) at the same time. It is through this property that the SWT retains the time invariance structure of the dataset i.e. there is no downsampling of the signal. At level 1, the SWT is obtained by convolving the signal \( x(i) \) with the appropriate filters as in the DWT but without downsampling. For comparison with the DWT case, Equations (4.7) and (4.8), represent the SWT approximation and detail coefficients respectively.

\[
\tilde{a}_p, \Xi = \left( x(i), \frac{1}{2^{\rho/2}} \phi \left( \frac{i-\Xi}{2^\rho} \right) \right) \tilde{a}_{p+1, \Xi} = \sum_{\theta=-\infty}^{\infty} I(\Theta) \tilde{a}_{p+2, \Xi + 2^\rho \theta}
\]  

\[4.7\]
The crucial result of this is that the coefficients of the approximation and detail at level 1 are the same length as the original signal. Level-adaptive size-varying highpass and lowpass filters are then employed to ensure equal length wavelet coefficients at each level, thus maintaining the original signal length. The general step $\rho$ convolves the approximation coefficients at level $\rho - 1$, with appropriate filters but without downsampling, to produce the approximation and detail coefficients at level $\rho$. For further details on SWTs, Zhong & Olutunde Oyadiji (2007) can be consulted for a thorough description of the mathematical theory.

### 4.2.2 Neuro-Wavelet Prediction Framework

A schematic of the prediction framework is shown in Fig. 4.2.

\[
\tilde{d}_p,\Xi = \left( x(i) \frac{1}{2^\rho} \psi \left( \frac{i-\Xi}{2^\rho} \right) \right) \tilde{d}_p,1,\Xi = \sum_{\Theta=0}^{\infty} h(\Theta) \tilde{d}_p,\Xi + 2^\rho \Theta
\] (4.8)
Figure 4.2: Stationary Neuro-Wavelet Prediction Framework

The proposed framework uses the SWT to decompose traffic flow and precipitation time-series data into approximation and detail components. These components are then predicted individually using separate ACNN models, with the forecasted components then recombined using an Inverse Stationary Wavelet Transform (ISWT). This recombined and reconstructed time-series is the predicted traffic flow. A novel element of the work is the method whereby precipitation data is used to compliment historical traffic flow data as an additional input to the ACNN forecasting models to develop a weather adaptive traffic prediction algorithm. Accordingly the proposed framework comprises of two parts: One part is a Dry model where

<table>
<thead>
<tr>
<th>Is rainfall expected in the next hour?</th>
<th>YES</th>
<th>Wet Model Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO</td>
<td>Dry Model Framework</td>
</tr>
</tbody>
</table>

**Dry Model Framework**

- Past Traffic Flow Data
- Stationary Wavelet Transform (decomposition)
- Approximation 1 → ACNN
- Approximation 2 → ACNN
- Approximation 3 → ACNN
- Approximation 4 → ACNN
- Detail 1 → ACNN
- Detail 2 → ACNN
- Detail 3 → ACNN
- Detail n → ACNN
- Inverse Stationary Wavelet Transform → Predicted Future Traffic Flow Data

**Wet Model Framework**

- Past Traffic Flow Data
- Stationary Wavelet Transform (decomposition)
- Approximation 1 (FLOW) → ACNN
- Approximation 1 (RAIN) → ACNN
- Approximation 2 (FLOW) → ACNN
- Approximation 2 (RAIN) → ACNN
- Approximation 3 (FLOW) → ACNN
- Approximation 3 (RAIN) → ACNN
- Approximation 4 (FLOW) → ACNN
- Approximation 4 (RAIN) → ACNN
- Detail 1 (FLOW) → ACNN
- Detail 1 (RAIN) → ACNN
- Detail 2 (FLOW) → ACNN
- Detail 2 (RAIN) → ACNN
- Detail 3 (FLOW) → ACNN
- Detail 3 (RAIN) → ACNN
- Detail n (FLOW) → ACNN
- Detail n (RAIN) → ACNN
- Inverse Stationary Wavelet Transform (reconstruction) → Predicted Future Traffic Flow Data
traffic flow is predicted using a stationary neuro-wavelet algorithm. The other part is a Wet model which predicts traffic flow utilising both traffic flow and rainfall information. The Wet model is activated if rainfall is expected in the forthcoming hour, based on local weather forecasts. The following two subsections describe the Dry and Wet forecasting models.

**Dry Model**

The Dry forecasting model predicts traffic flow in near future using current and recent past observations of the same. The first step of the procedure is to decompose traffic flow data using the SWT, to an optimised level, \( p \). The approximation and detail components at all levels of decomposition are then forecasted with individual ACNN models, as seen in Fig. 4.2. These separately forecasted components are then reconstructed using the ISWT. The recombination of the separate approximation and detail elements in this way produces a final forecasted time-series, of the same order as the original hourly traffic flow time-series input to the model.

**Wet Model**

The Wet forecasting model follows the same basic idea as in the Dry model. However, it is unique in incorporating the rainfall data to augment the historical traffic flow data as an extraneous variable for predicting future traffic flow in wet conditions. The current and time-lagged correlation between traffic flow and precipitation time-series can be assumed as an indicator of how much traffic flow is affected by rainfall. This relationship can exist at different frequency levels and by exploiting the multiresolution framework of the SWT, the correlation between the two aforementioned time-series has been calculated for the original time-series datasets as well as at different levels of approximation and detail components. The component series showing maximum correlation between traffic and rainfall data have been used as input to the ACNN structure. In most cases, it has been observed that the approximation coefficients obtained by decomposing the traffic flow and rainfall time-series show the maximum correlation. Therefore in the Wet model as described Fig. 4.2, the approximation coefficients for both rainfall and traffic flow data are presented as input to the ACNN structure at each level.
of decomposition, for illustrative purposes. The ACNN algorithms in the Wet model are multivariate models, including both rainfall and traffic flow as input, and producing future traffic flow levels as output.

4.3 Evaluation of Neuro-Wavelet Model

The prediction methodology described in Section 4.2 is evaluated using real-time traffic flow data from urban arterials along with hourly precipitation data in Dublin, Ireland. In the following subsections the description of the modelled traffic flow data, rainfall data and the model fitting procedure are described.

4.3.1 Data

The data modelled using the proposed prediction methodology is obtained from the SCATS database in the Traffic Control Centre of Dublin City Council. For evaluating the prediction algorithms in this chapter, hourly traffic data from January 2009 were collected from two SCATS sites, TCS 106 and TCS 125, as seen in Fig. 4.3. Both these sites lie to the west of Dublin City centre. TCS 106 is a 5 phase junction consisting of a T-junction where the Kylemore Road meets the Lucan Road. TCS 125 is a 4 phase crossroads where the Nephin Road intersects the Navan Road. It is expected that both sites will be influenced by commuter traffic, with TCS 106 and TCS 125 being situated close to the two major commuter routes, N4 and N3 respectively, in and out of the city.
Only weekday traffic condition variable observations are modelled in this chapter. Fig. 4.4 shows typical traffic flow levels for a weekday at each of the two junctions modelled. Typically the values are highest around 08:00 hours as commuters travel to work during the morning peak period and again at 17:00 hours during the evening peak period, while the roads are quietest from midnight until approximately 06:00 hours the next morning. It is also evident from Fig. 4.4 that TCS 106 encounters heavier traffic than TCS 125. Also, the bimodal nature of the traffic flow at TCS 106 indicates that the junction is more affected by commuter traffic. The traffic flow time-series for TCS 125 does not display such prominent peaks and troughs during expected commuter travel hours. The graph shows that the variation of traffic volume at TCS 125 is more stable, with a fairly consistent flow of traffic from 07:00 hours to 19:00 hours.
The rainfall data used in this model were collected from the Phoenix Park weather station, as provided by Met Eireann, the Irish National Meteorological Service (http://www.met.ie). The weather station is located in close proximity of the two chosen traffic junctions, which ensures that local rainfall events which may affect the traffic flow at the two chosen sites will be reflected in the rainfall records of the chosen weather station (see Fig. 4.3). Hourly rainfall records were available from the Phoenix Park weather station and hence, hourly traffic flow data were used for modelling in this chapter. The rainfall levels (measured in mm/hour) during the weekdays between 1st and 14th January 2009 are shown in Fig. 4.5. Generally, January and February are the wettest months of the year in Dublin. However, in this instance there were not many rain events during the majority of the time period as can be seen in the figure.
4.3.2 Traffic modelling using SWT-ACNN Structure

Traffic flow time-series data are modelled using a SWT-ACNN structure for future predictions. The framework consists of two parts, as discussed in the methodology, the Dry model and the Wet model.

Dry Model

The proposed neuro-wavelet traffic prediction framework employed the Dry model to predict traffic flow provided there were no rainfall events existing or expected in the next hour. In this model at the first step, the SWT involving the db3 wavelet basis function (Daubechies, I., 1992) was used to decompose the traffic flow time-series into separate approximation and detail coefficients (see Fig. 4.6). This decomposition was performed multiple times to achieve an optimum level of approximation for representing the within-day traffic dynamics. At each level the approximation coefficients were decomposed using the SWT to generate approximation and
detail coefficients for the next level; the transformation was undecimated and the generated datasets were of the same length as the original traffic flow data. For illustrative purposes, the wavelet decomposition of the traffic flow data from TCS 106 only is shown in Fig. 4.6. Each separate component series, generated through the use of the SWT as seen in Fig. 4.6, was modelled by an individual ACNN and the output from all ACNN structures were reconstructed using the ISWT. In all, 240 observations of hourly traffic flow were used in this model. Eight days were used to train the ACNN (points 1 - 192) while the points from the ninth day (points 193 - 216) were used to predict hourly traffic flow levels on the tenth day (points 217 - 240). The size of the input vectors for the ACNN prediction algorithms were dependent upon the results of the autocorrelation procedure on both hourly traffic flow datasets. Study of the autocorrelation coefficients for TCS 106 showed that 5 points from a 24 hour period were deemed to have an autocorrelation coefficient of sufficient size to be influential in future predictions. Therefore, in the prediction of traffic flow at TCS 106, the training and test datasets made use of these 5 influential points when creating the input vectors for the ACNN. Thus, 5 influential historic hourly traffic flow data points were used to predict a single future hourly traffic flow value, rather than using the entire 24 points of the day at each iteration of the prediction algorithm. This procedure was repeated in steps to predict a full 24 hour period. The same methodology was followed for the TCS 125 traffic flow dataset, although only 3 points were deemed influential in this case. In case of rainfall events, the framework switched to a Wet model. The standard ANN model used for comparison in this work used the entire 24 points from a day for prediction. In this way, the SWT-ACNN model is also valuable as a model that reduces computational time e.g. in the case of TCS 125, the standard ANN prediction dataset was 8 times larger than that used in the SWT-ACNN model, thus the SWT-ACNN model computed a prediction in a shorter time.
As discussed in the methodology in Section 4.2, traffic flow is affected by weather (Chung et al., 2005) and hence rainfall data was included in the Wet model framework. Rainfall data time-series were decomposed using SWTs in the same way as the traffic flow data. Current and time-lagged correlation coefficients were calculated to determine whether rainfall events and traffic flow were related substantially at different frequencies to improve the forecast accuracy. The correlation coefficients between rainfall data and traffic flow data at junction TCS 106 are shown in Table 4.1, as an illustrative example. Correlation coefficients for the original time-series and their SWT components (approximation and detail coefficients at 3 levels in this case)
were all looked at individually to ascertain which components had the highest correlation. The labels A, B and C in Table 4.1 refer to conditions under which the correlation coefficients were calculated. In condition A, the entire rainfall and traffic flow datasets were used for calculating the correlation coefficients. This condition resulted in the lowest levels of correlation as expected because, for a large proportion of the time, the rainfall data is 0 mm/hour and this skews the relationship between traffic flow and rainfall. Condition B focused on the correlation present only in instances where there is some level of rainfall in a given hour. As seen in Table 4.1, this resulted in an improved level of correlation. The results for Condition B in Table 4.1 along with visual inspection of Fig. 4.5 vindicate the decision to calculate correlation coefficients at times when rain is present only, to negate the effect of the large proportion of zero values in the rainfall dataset. Condition C looked at hours where rain was present along with a single following hour of such time periods. This was to allow for the delayed affect that inclement weather can cause to traffic flow levels and behaviour i.e. traffic jams formed under rainy conditions may still be present for a percentage of the following dry hour.

Table 4.1: TCS 106 Rainfall Data Correlation Coefficients

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Time-Series</td>
<td>0.1114</td>
<td>0.2998</td>
<td>0.2808</td>
</tr>
<tr>
<td>Approximation Level 1</td>
<td>0.1199</td>
<td>0.3157</td>
<td>0.3459</td>
</tr>
<tr>
<td>Approximation Level 2</td>
<td>0.1316</td>
<td>0.3860</td>
<td>0.3879</td>
</tr>
<tr>
<td>Approximation Level 3</td>
<td>0.1404</td>
<td>0.3224</td>
<td>0.2943</td>
</tr>
<tr>
<td>Detail Level 1</td>
<td>0.0140</td>
<td>0.0793</td>
<td>0.0492</td>
</tr>
<tr>
<td>Detail Level 2</td>
<td>0.0087</td>
<td>0.0341</td>
<td>0.0153</td>
</tr>
<tr>
<td>Detail Level 3</td>
<td>0.0889</td>
<td>0.1796</td>
<td>0.2116</td>
</tr>
</tbody>
</table>

Conditions B and C both produced similar levels of correlation between rainfall and traffic flow at junction TCS 106. An interesting result was that there was a much greater correlation present in the approximation coefficients than the detail coefficients. Based on this behaviour, the decision was made to only use the approximation components of the decomposed hourly rainfall data as part of the inputs in the Wet model framework. It was also decided that when rainfall was expected in the forthcoming hour, the Wet model would be activated for both the forthcoming hour and the hour after that to allow for the time-lagged effect of rainfall on
within-day traffic flow dynamics. Also, in the Wet model, the input vector sizes for the individual ACNN were twice the size as the respective Dry model input vectors i.e. where 3 hourly traffic flow data points were used to predict 1 future data point at TCS 125 in the Dry model, 6 points (3 hourly traffic flow and 3 hourly rainfall) were used for the same in the Wet model case.

### 4.4 Prediction and Comparisons

The SWT-ACNN model has been used to predict the traffic volume on Wednesday, January 14th 2009 for junctions TCS106 and TCS 125. The forecasts and the observed traffic volumes are plotted in Figures 4.7 and 4.8.

![Figure 4.7: TCS 106 Prediction Results](image-url)
To illustrate the efficiency of the SWT-ACNN structure, a standard ANN algorithm, comparable to the fundamental FFBPNN structure used in the ACNN model, has been used to predict traffic volume on the same day over a 24 hour period. The shaded boxes in Fig. 4.7 and 4.8 show periods of rainfall (from 14:00 hours to 19:00 hours) where the Wet model framework took over from the Dry model framework, as discussed in the methodology. It is important to note here that the rainfall in this period was actually intermittent, as seen by close inspection of Fig. 4.5. Rainfall events occurred at 14:00 hours, 16:00 hours and 18:00 hours but the hours in between were also subject to the Wet model prediction algorithm in order to address the time-lagged effect of precipitation on traffic flow i.e. the Wet model activates for hours $p$ and $p +1$ when rain is expected at hour $p$. It is immediately visible from the graph that the wavelet model outperforms the non-wavelet model significantly. Table 4.2 presents the MAPE and RMSE from the predictions carried out at TCS 106 and TCS 125, including error values specific to both wet and dry times for clarity.
Table 4.2: Prediction Accuracies at TCS 106 and TCS 125 under different Weather Conditions

<table>
<thead>
<tr>
<th></th>
<th>TCS 106</th>
<th>TCS 125</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (vehicles/hour)</td>
<td>1664.50</td>
<td>1446.50</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>1087.80</td>
<td>745.09</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dry</td>
<td>9.09</td>
<td>8.01</td>
</tr>
<tr>
<td>Wet</td>
<td>4.44</td>
<td>9.91</td>
</tr>
<tr>
<td>SWT-ACNN MAPE (%)</td>
<td>10.65</td>
<td>144.40</td>
</tr>
<tr>
<td>RMSE</td>
<td>125.42</td>
<td>67.23</td>
</tr>
<tr>
<td>ANN MAPE (%)</td>
<td>16.57</td>
<td>14.34</td>
</tr>
<tr>
<td>RMSE</td>
<td>165.89</td>
<td>150.56</td>
</tr>
</tbody>
</table>

The values in Tables 4.2 emphasise the importance of using a multi-resolution framework in incorporating the effect of rainfall in traffic prediction, with the SWT-ACNN model more accurate at both TCS 106 and TCS 125, based on the MAPE and RMSE values. As an illustrative example of the effectiveness of the Wet model in predicting traffic flow, the MAPE for the neuro-wavelet prediction model during dry and wet times at junctions TCS 106 and TCS 125 are also shown in the table. The table is based on the predictions shown in Fig. 4.7 and 4.8, where a rainfall event occurs from 14:00 hours to 19:00 hours. Hence, for this time period, the Wet model is enabled as the chosen prediction methodology. The findings in Table 4.2 emphasise how effective it can be to include rainfall as an input to the neuro-wavelet model.

4.5 Conclusion

Rainfall affects travel dynamics in urban transport networks in a complex manner and causes congestion and often an increased frequency of accidents on the roads. A neuro-wavelet prediction algorithm has been proposed in the chapter to forecast hourly traffic flow taking into account the effect of precipitation. The neuro-wavelet structure utilises a multiresolution wavelet analysis to address the fact that rain can affect traffic flow dynamics at different resolution levels. To retain the time-invariance of the original time-series datasets an undecimated form of WT, namely the SWT, has been used. This model taking rain into account is a novel application of the SWT in traffic prediction literature.

The algorithm toggles between a Dry model and a Wet model depending on whether rainfall is expected in the forthcoming hour. The algorithm was evaluated using real-time
hourly traffic volume and precipitation observations. The prediction accuracy of the Wet model is superior to the Dry model and the standard ANN model during rainfall periods. This emphasises that rainfall affects traffic volume and the inclusion of rainfall as a model input improves the prediction accuracy during rainfall events. The algorithm also outperformed the conventional ANN algorithms in dry situations (where traffic flow data alone were used as model input). This confirms that the proposed algorithm is flexible and can be applied to situations where and when rainfall data is not available. For future research in this area, it will be important to investigate the performance of this algorithm using higher resolution traffic and rainfall time-series datasets (for e.g. data time aggregation of 15 minutes or less). The excellent results using hourly data suggest that forecasts using higher resolution data will also benefit greatly from the novel approach described in this work, once traffic flow and rainfall data of suitable lower time aggregations are obtained. Having proven the merits of using the SWT-ACNN model with rainfall for predicting traffic volume, it will also be important to investigate if the addition of rainfall as an exogenous variable is effective when predicting alternative traffic condition variables such as speed, occupancy or travel time.
5 Chapter 5

TRAVEL TIME DATA PROCUREMENT AND INVESTIGATION

5.1 Introduction

So far in this thesis, the traffic information modelled (flow and speed specifically) cannot be directly reported to travellers through ATIS. The traffic condition variable of most use to commuters is travel time and given its relevance to commuters, it is important to estimate and predict travel time. In the work in Chapters 2, 3 and 4, the traffic condition variable data used for modelling has come from the SCATS dataset in Dublin and the MIDAS dataset in the U.K. The SCATS dataset consists only of traffic flows at urban arterial junctions. MIDAS data contains flows, headways, speeds and occupancy from lanes on British highways. SCATS data was recorded at intersections in urban areas, while MIDAS data was recorded at points on motorways in the U.K. but neither of these datasets contained information on travel time.

Having investigated traffic flow and speed modelling in different conditions, it is intended to examine travel time forecasting next in the thesis. Consequently, there is a requirement for an alternative data resource, which records travel time observations over a road-section or route, rather than at a point within a network. In this chapter, the various methods of collecting travel time data are discussed. Also, two separate travel time datasets were investigated; namely an Automatic Number Plate Recognition (ANPR) camera data collection study in two separate locations in Dublin, Ireland and a GPS based probe vehicle data collection method used in Vienna, Austria. Travel time data collected through the most suitable technique was chosen for further modelling in this thesis.

5.2 Summary of Travel Time Data Collection Methods

This section describes the various techniques used to collect travel time data for use with ITS. These can broadly be described as stationary or moving methods. Stationary methods work by recording vehicles that pass by fixed points and calculating the travel time based on the time
taken for a vehicle to travel from one stationary detector point to another e.g. ANPR cameras. The moving methods involve an object moving within the traffic flow and the travel time is recorded based on the position of the object at different times e.g. GPS systems fitted to probe vehicles.

Stationary travel time collection techniques have been popular for a long time due to the ease with which small datasets can be collected. The simplest method involves having two people record times at which certain vehicles pass fixed points on the road network. The times are subtracted to give a travel time for that section of road and in this manner a small travel time dataset can be created in a reasonably short space of time. This small dataset would give an indication of the travel time over a certain route but the major drawback is the time consuming nature of the data recording process, combined with the relatively small dataset yielded by such an approach. However, as technology has improved, it became possible to have certain devices record the timestamps of vehicles at fixed locations, without the need for people to be on site. One such device is the standout stationary travel time collection method; the ANPR camera. ANPR cameras record images of licence plates at different points along a route. The process of using ANPR cameras to create travel time datasets, along with a study of a created dataset, is discussed in greater detail in Sections 5.3 and 5.4 in this chapter. In simple terms however, the travel time dataset is generated by calculating the time spent by vehicles between stationary recording locations.

These modern stationary methods have the advantages of simplicity and of collecting travel times from a large amount of motorists, each with their own unique driver behaviour. They also benefit greatly from being active throughout the day, creating a dataset that can fully describe how the travel time varies within the day and between peak and off peak times. However, these methods do have minor drawbacks. The locations for installing ANPR cameras are not always readily available. Each location must offer a good view of the traffic without impeding either motorists or pedestrians, while also allowing the ANPR cameras to safely receive power at these locations. Also, the very nature of stationary locations means the travel
time dataset is restricted to the selected part of the road network in which the cameras operate. ANPR cameras allow for an excellent travel time dataset to be obtained for the specific section of road between two fixed points. However, they do not offer the same versatility as the ‘moving’ travel time dataset collection techniques described hereafter.

There are a growing number of “moving” techniques used by those who wish to record travel time on road networks. These techniques work by recording time and position data on board a vehicle as it traverses the road network. The simplest means of producing travel time data in this manner is to have a passenger use a stopwatch in a vehicle to record the time taken by the vehicle to cover the distance between predefined points. As with the stationary methods, advances in technology have made these methods of travel time data collection less laborious. These advances include the use of built in Global Positioning Systems (GPS) in vehicles which allow the user to download the time and location data at a later date. GPS has been used in conjunction with certain bus companies and taxi firms as a means of mapping the travel time around an urban area. Mobile phones can also be tracked during a trip and thus a travel time dataset can be built up. The advantages of these methods include the ability to learn about travel time all along a route, as opposed to certain fixed locations with the stationary methods. A broader knowledge of travel time throughout the network can be obtained as different probe vehicles complete different trips. However, if only one vehicle is actively collecting data (it can be expensive to equip many vehicles with GPS), the travel time dataset will be sparse compared to a fixed stationary system with hundreds of vehicles being recorded every hour. As such, the data collector with limited finances must choose between a stationary method which gives a detailed view of travel time along a certain route at all times of the day, or a moving method which gives travel time data for a larger area, but with less vehicles to quantify the recorded times. To further investigate the advantages and disadvantages of the different methods, One recording technique from each method type is analysed in the case studies in this chapter; the ANPR method in Section 5.3 and the GPS probe vehicle method in Section 5.5.
5.3 Case Study 1: Dublin Automatic Number Plate Recognition (ANPR) Camera Dataset

In this section, an ANPR camera setup capable of recording travel time in Dublin, Ireland is described. Figure 5.1 shows a flow chart describing the different steps involved in creating a travel time dataset using ANPR cameras.

![Diagram showing ANPR Camera, eBox, Laptop for Data Processing, and Travel Time Data](image)

Figure 5.1: Travel Time Dataset Creation Flow Chart

ANPR cameras were used to record images of licence plates at different locations. These images of licence plates, and the time those images were captured, were then stored on an eBox data storage unit (one eBox for each camera). A laptop could be connected to the eBox to retrieve the stored licence plate data. The data from two consecutive ANPR cameras on the same route and lane was then processed to produce the final result: travel time data. The different parts of this flow chart are described starting here with a description of the camera and waterproof boxes as they existed in situ including the ANPR camera setup and the eBox in Section 5.3.1. The method by which data was extracted from the setup is explained in Section
5.3.2. Examples of travel time time-series data from the ANPR camera setup travel time dataset are shown in Section 5.4.

5.3.1 Camera and eBox Description

This section describes the various physical elements of the entire ANPR travel time data collection setup.

**ANPR Cameras**

The ANPR cameras, the first component of the flow chart in Fig 5.1., record images of licence plates along with a timestamp. For traffic flow, the number of unique images is summed to give traffic flow data. Travel times are calculated by subtracting the timestamp of a certain image upstream from its respective downstream timestamp, giving a time taken for the vehicle in question to cover the distance between pairs of cameras. Each of these ANPR cameras is specifically designed to recognise registration plates through the use of a specially designed lens in a high resolution digital camera. Any licence plate seen by the camera is isolated from the surrounding environment by the on board ANPR software.

The cameras used in this work were designed by Alpha Vision Design (AVD). They state that closed circuit television cameras (CCTV) technology is 50 years old and conventional CCTV based ANPR systems are not robust and require unacceptable levels of maintenance. AVD developed their own self-contained ANPR system capable of a read rate exceeding 99%, far greater than traditional CCTV systems where the read rate was generally below 60%. AVD's ANPR system is contained in security housing and includes an integrated illuminator, a high resolution digital camera, a digital analyser and on-board relays. Mounting and power cables were all that was required for setup for the use of these cameras in this thesis. Dublin City Council (DCC) generously agreed to mount and power the four ANPR cameras used in this work. An example of a captured image of a vehicle's registration plate as recorded at the site in Dublin is shown in Fig. 5.2.
The top left of the image in Fig 5.2 displays the timestamp information recorded with every captured registration plate. The details recorded in every instance of a successful number plate recognition include, the registration plate itself, the date and time of recognition, and three parameters (gain, shutter and strobo) which alter the quality of the picture seen by the camera. Gain is a means of artificially brightening an image recorded in low light, while the shutter and strobo parameters are related to the amount of light entering a lens and the flash frequency respectively. These three parameters were tested both in the lab and on site to find the best settings for each camera location.

**eBox and Waterproof Enclosure Setup**

The ANPR cameras are excellent tools for recognising licence plates in the field. However, as the cameras themselves are incapable of data storage, a separate storage medium (VESA PC eBox-2300) needed to be installed with each camera (the second component of the flow chart in
Therefore, the final setup at each of the four locations involved an ANPR camera linked to a waterproof enclosure which contained the eBox storage medium, a hub allowing for outside laptop connectivity, and a circuit breaker for safety. This waterproof setup was designed and built in the lab. The camera and waterproof box, as set up in the lab for testing prior to data collection in the field, is shown in Fig. 5.3.

![ANPR Camera and Waterproof Box](image)

**Figure 5.3: ANPR Camera and Waterproof Box**

The configuration of the ANPR camera as shown in Fig 5.3 in the laboratory set up was very similar to that on site, with the exact same brackets used for connecting the system to the traffic light poles and the same waterproof housing used for the connections between the camera and the power supply and the network hub and eBox in the waterproof enclosure. The contents of the waterproof enclosure on the left hand side of Fig. 5.3 were arranged as shown in the schematic in Fig. 5.4.
The power for the components of the waterproof enclosure was drawn from the power source already present to power the traffic lights on the poles to which the ANPR setup was affixed. The Residual Current Circuit Breaker with Overcurrent Protection (RCBO) was installed as a safety measure. RCBOs act by disconnecting the circuit if an imbalance in the electric current between the energised and return neutral conductors is detected. It is important to note the presence of a crossover switch or adapter on the line allowing data to flow from the camera to the network hub. Crossover adapters allow two computing devices of the same type to connect.
Without this crossover switch, the hub would be unable to direct the images coming in from the camera to go to the eBox for storage. The hub also allows connectivity between an external laptop, via the RJ45 connection, and the eBox. These connections are discussed under the following heading.

**Connection and Access**

The hub in Fig. 5.4 is crucial for connectivity. It allowed the ANPR camera to communicate with the eBox i.e. it is the link between the first two components of the flow chart in Fig. 5.1, thus allowing all the recorded licence plate images captured by the camera to be stored on the eBox. The hub also allowed the connection of a laptop to the system from the outside, thus facilitating the downloading of licence plate data in situ. Using a laptop and a crossover cable, the data could be downloaded every week or so and stored on the laptop, thus always removing the risk of the eBox reaching capacity. Connection to the camera or eBox using a laptop (via crossover cable at the RJ45 socket in Fig. 5.4) involved connecting to a certain Internet Protocol (IP) address e.g. the camera was located at http://192.168.1.13. The eBoxes had different IP addresses and were accessed using File Transfer Protocol (FTP). The process of using the software on the camera and eBox is discussed in Section 5.3.2.

**ANPR Camera Locations**

The locations of the four ANPR cameras used to record travel time information are described in this section. The two pairs of ANPR cameras in this work were used to capture the registrations of passing vehicles at two locations either end of Pearse St. in Dublin, Ireland. The cameras were mounted on existing traffic poles at the junctions, which were already in effective positions from which the cameras could record licence plates from vehicles in different lanes. The exact location of each camera pair is shown in Fig. 5.5.
One ANPR pair is located at the top of Pearse St. near the junction with Tara St. (ANPR Site 1 in Fig. 5.5) while the other was set up at the junction of Pearse St. and Westland Row (ANPR Site 2 in Fig. 5.5). Pearse St. is an important commuter route, particularly in the evenings as it is one of the main arterials funnelling traffic onto the quays to travel from the southern side to the northern side of the city. A large amount of commuter traffic uses the quays as part of their route home, hence Pearse St. is extremely busy during peak commuting hours. Fig 5.6 shows the specific traffic light poles on which the ANPR cameras were installed at ANPR Site 1.
Figure 5.6: ANPR Site 1 Camera Locations

The cameras at the two locations in Fig 5.6 were orientated in such a way as to try capture as many of the four lanes (three standard lanes and one bus lane) on Pearse St. as possible. Figure 5.7 again highlights the specific camera locations in red ovals. These two cameras are from ANPR Site 2.

Figure 5.7: ANPR Site 2 Camera Locations
As well as offering good height from which the ANPR cameras could view a large area of the road, the traffic light poles also offered a secondary benefit; as mentioned previously, DCC kindly agreed to install the cameras on traffic light poles and power them in situ using the existing power supply that powers the traffic lights.

5.3.2 Data Software Interface and Travel Time Dataset Creation

This section describes some of the software and related codes and algorithms involved in extracting travel time data from the camera and eBox system.

SmartReg 25 ANPR Software

The SmartReg 25 software built into the ANPR camera was accessed through a browser using http://192.168.1.13 as the IP address. Following successful username and password validation, various settings on the camera software could be accessed. The most important settings for this work involved instructing the camera to send all saved images to the eBox through FTP. This was achieved by specifying the IP address, username and password of the eBox. These settings instructed the camera of the FTP IP address to which data would be sent and also specified which details would be included in the filename of each image e.g. vehicle registration number, date and time of passage. The specific settings used are displayed in Fig. 5.8.
In this work, the filename was critical to data processing, as the filenames contained the license plate numbers and time stamps (the details are provided in the next subsection). Instead of saving and processing individual images, the filenames were stored and processed. This alleviated the issue of storage space due to the storage of approximately 200,000 images per location during data collection periods, an image captured for each individual registration plate record. As seen in Fig 5.8, the vehicle registration number, date and time of passage were all included in the filename of each licence plate image. The image filenames were stored in a text file as a list of character strings. In order to perform this conversion, algorithms making use of FTP command line code were developed. Using ftp://192.168.1.20 as the eBox IP address, the
following algorithm copied the file names of image files in folders as character strings and stored them as a list in a new text file:

1. Access FTP through command prompt at the following address: 192.168.1.20
2. Input username and password
3. List the folders to be examined using the ‘mdir’ FTP command (mdir lists the file names of the contents of a folder)
4. Specify the name of the text file where the file names will be stored
5. Exit FTP

The code used the command line prompt in Windows to access the address of the eBox. Once the username and password are accepted, the command ‘mdir’ listed the names of each file in the chosen folders and stored the file names in a new text file. Once the entries were stored, the text file was then manipulated to obtain those parts of the filename which were most useful.

Data Processing Algorithms

The text files created using the abovementioned algorithm contained lists of character strings of the following form: “07KE7438-IRL_PearseSt_2011-02-14_00-00-18-013”. In this string, the term “07KE7438-IRL” refers to an Irish licence plate registration: year of registration (07 means the car was registered in 2007), county of registration (KE means it was registered in Co. Kildare in Ireland), and registration number (7438 means it was the 7438th car registered in Co. Kildare in 2007). The term “PearseSt” refers to the location of the ANPR camera. The date of the image capture is found in the term “2011-02-14” (14th February 2011). Finally, the last term in the filename, “00-00-18-013”, refers to the timing of the image recording (in this case the image was captured at 18 seconds and 13 milliseconds past midnight). This information was isolated from each filename in order to retrieve the vehicle registration number, date and passage of time.
With separate files for each parameter (vehicle registration number, hour, minute etc.) at each location, the next task was to calculate travel time based on the available data. The vehicle registration numbers and the passage timestamps from a pair of upstream and downstream ANPR cameras were analysed to calculate the travel time. Each vehicle registration number appearing in the upstream camera was searched in the records captured by the downstream camera. For every match found, the number of vehicles passing was increased by one and the position of the matched vehicle registration in the text files containing filenames from both ANPR locations was recorded. Thus the number of vehicles which appeared and were captured at ANPR Site 1 and ANPR Site 2 were counted.

It is important to note that not every vehicle registration that passed by both ANPR sites was photographed correctly by the ANPR cameras. There were a few obstacles that had the potential to cause licence plates to be missed by cameras at either site. Although the ANPR cameras are highly accurate and responsive, if a licence plate was very dirty, broken or warped it may not have been recognised. Turning movements may also have obscured traffic in certain lanes, particularly at ANPR Site 1, where large buses often occupied the lane closest to the cameras during turning movements. However, the cameras did pick up the large majority of passing registration plates. With the number of registration matches counted, the next step was to calculate the travel time taken for each of these registrations to get from ANPR Site 2 to ANPR Site 1. The most intuitive way of doing this was to create a code to convert all timings to seconds and simply subtract the time at one ANPR site from the other. It was also decided to filter out any times above a chosen upper limit i.e. if the travel time was unreasonably long, there was a possibility the matching algorithm had matched the same car more than once if that car had undertaken more than one journey on Pearse St. on the same day. This code successfully calculated the travel time (in seconds) of any registration captured at both ANPR sites. On identification of the dataset of travel times (in seconds), the average travel times over every 15 minutes in the day were calculated and stored for future modelling purposes. In summary, by applying the various algorithms to the initial text file containing the file names of...
images recorded by the ANPR cameras, a dataset containing average travel time in 15 minute aggregations was constructed. This finalised travel time dataset is discussed in Section 5.4.

5.4 ANPR Travel Time Data Description

This section contains the descriptions of two different ANPR travel time datasets looked at in this study.

5.4.1 Pearse St. Travel Time Dataset

Fifteen minute aggregated travel time was calculated using licence plates and time stamps recorded by ANPR cameras in Dublin. Figure 5.9 displays a single day of fifteen minute aggregated average travel time between ANPR Sites 1 and 2.

![Figure 5.9: Pearse St. 15 Minute Aggregated Average Travel Time Example](image)

The main problem noticeable was that there were gaps (points with 0 travel time on the y-axis) in the data during very early morning hours, specifically between 01:00 hours and 07:00 hours in this particular example. This is because the roads themselves, even in a busy area of Dublin
city centre, can have fifteen minute spells where no vehicles pass by both ANPR sites. Therefore, no licence plate matches are recorded and as such, no travel time calculations can be computed for these early morning periods. Aside from the gaps in the early period of the day, the travel time data is fairly consistent for the most part, with an average travel time of approximately 30-60 seconds in the example in Fig. 5.9. This travel time length seems sensible as there are three sets of traffic lights in total on the road so it is quite likely that many cars that drive on Pearse St. in a given fifteen minute time interval will be stopped by at least one set of traffic lights. As with the other models in this thesis, weekday data is investigated separately to weekend data. Table 5.1 shows some descriptive statistics about 15 minute average travel time data from 20 weekdays at the Pearse St. site.

Table 5.1: Pearse St. Travel Time Dataset Descriptive Statistics

| Mean (μ) | 44.8872 (s) |
| Standard Deviation (σ) | 38.1315 |

5.4.2 Tallaght Travel Time Dataset

This travel time dataset was also created with the use of ANPR cameras and is looked at for comparison purposes and to show that slightly different data collecting methodologies exist even under the broad heading of ANPR cameras. Eight cameras were used in total as part of a Master’s Thesis by Martin Maycock (2009), a student in the Civil, Structural and Environmental Department of Trinity College Dublin. The thesis was a study on improving traffic signal control systems in the South Dublin County Council (SDCC) region and as part of this, an ANPR journey time survey was completed. The test area was located on Greenhills Road, an arterial route going from Tallaght towards Dublin. The location of the eight ANPR cameras on and around Greenhills Road is shown in Figure 5.10.
In the work, wireless radio communications were used to communicate between a 'master' laptop and eight 'slave' netbooks, one located with each camera. Thus, the licence plate data was transferred using a wireless file transfer system, as opposed to the 'wired' connection used in the Pearse Street travel time data collection process. Once the data was stored, the travel time was computed in the same manner as described in Section 5.3.2, although Microsoft Access was used as the software package in this case. The use of eight separate cameras and netbooks made this ANPR data collection technique a lot more expensive than the data collection carried out on Pearse St. An example of travel time throughout the day between two of the cameras in the Tallaght dataset is shown in Figure 5.11.
The behaviour of the fifteen minute average travel time in Fig. 5.11 is unusual in that there are a lot of spikes throughout the day. In this way, the travel time data from this location seems highly unpredictable. However, a smoothed view of the data would show that in general the shortest travel times are at night, while during the day the travel time increases, before reducing again after 20:00 hours or so. Descriptive statistics for this dataset are displayed in Table 5.2.

Table 5.2: Tallaght Travel Time Dataset Descriptive Statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ($\mu$)</td>
<td>304.4268 (s)</td>
</tr>
<tr>
<td>Standard Deviation ($\sigma$)</td>
<td>132.4808</td>
</tr>
</tbody>
</table>

The 'spiky' nature of this dataset makes accurate predictions very challenging. As such, the dataset obtained from the ANPR cameras on Pearse St. was chosen as the favourable 'stationary method travel time dataset upon which to apply travel time prediction algorithms in work described later in this thesis.
5.5 Case Study 2: Vienna Probe Vehicle Travel Time Dataset

The second travel time collection methodology considered in this work is a travel time dataset sourced from Vienna in Austria. In this case, travel time was collected by a fleet of taxis equipped with GPS. The use of such GPS equipped probe vehicles is also known as floating car detection. It is important to note that in this case, travel time was estimated using harmonic average speed data and distance measurements, as opposed to recording travel time directly. The taxi fleet helped to build up a map of free speeds around Vienna city centre and from this, travel time was estimated. An example of this free speed map of Vienna is shown in Fig 5.12.

The data specific to the work in this thesis was centred on Schoenbrunnerstrasse, an urban arterial in Vienna, and the location of this arterial is depicted by the thick black oval in Fig 5.2.

![Road Network Free Speed Classes](image)

**Figure 5.12: Schoenbrunnerstrasse as part of the Vienna free speed map**

The free speed map illustrates the different speeds expected on different road types throughout Vienna. This emphasises the point made in Section 5.2 that stated that moving methodologies were very adept at building up a network wide view of the travel time, as opposed to the very...
detailed views of travel time data on specific routes using ANPR cameras. However, it is important to note that a fleet of taxis will not always be available to aid with data collection and the fact that this dataset was built up using a fleet of cars as opposed to single probe vehicle should not be downplayed. In relation to the travel time data itself, it can be seen how the motorways to the south of the map, with speeds of 90 km/hour or more recorded, contrast greatly with the lower speeds on the urban arterials of Vienna. Schoenbrunnerstrasse (within the thick black oval in Fig 5.12) largely contains quite low vehicle speeds. The travel time is estimated based on the speeds recorded throughout the day and an illustrative example of how travel time fluctuated during a day on Schoenbrunnerstrasse is shown in Fig. 5.13.

![Figure 5.13: Schoenbrunnerstrasse Daily Travel Time Example (15 Minute Aggregation)](image)

The main issue with floating car detection is that the quality of the dataset is dependent solely on the movements of a taxi fleet. This can cause large gaps in the dataset at times when the taxi fleet are not busy and at times when for different reasons, some areas of the road network are not travelled upon for an extended period of time. With such large gaps present in the dataset, it
is necessary to average out the travel times recorded over a period of a number of days, just to
give an idea of what a single day’s travel time would be like. The travel time shown in Fig. 5.13
was calculated by averaging the travel times recorded on a number of weekdays. This is enough
to give an idea of travel time on a given day, but it makes predictions very difficult due to the
large amount of gaps overall in the dataset. Table 5.3 contains descriptive statistics for this
travel time dataset.

Table 5.3: Schoenbrunnerstrasse Travel Time Dataset Descriptive Statistics

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (μ)</td>
<td>15.7525 (s)</td>
</tr>
<tr>
<td>Standard Deviation (σ)</td>
<td>5.1821</td>
</tr>
</tbody>
</table>

This dataset displays the potential for collecting network wide statistics using probe vehicle
data collection techniques. However, the particular results of this case study are underwhelming
when compared to the dataset created using ANPR cameras on Pearse St. described in Section
5.4.1. As mentioned previously, Figure 5.13 was generated using averages of different travel
time recordings throughout the day. The graph displays unusual behaviour, in that the travel
times are quite long in the early morning hours when the road network would be expected to be
quiet. Additionally, the shortest travel times recorded for use in the graph occur around 19:00
hours. Again, this defies convention as it would be expected that the road network would be
congested or just recovering from congestion at this point, thus resulting in slow travel times.
Therefore, having weighed up the previously discussed pros and cons, the Pearse St. ANPR
dataset is chosen as the ‘best’ travel time dataset investigated in this work.

5.6 Conclusion

In this chapter, various travel time data collection methodologies are discussed. These are split
primarily between ‘stationary’ techniques and ‘moving’ techniques. Following the discussion,
three separate travel time datasets are analysed. The locations from which said datasets were
obtained and the methods of data collection were described. Of the three travel time dataset
options, the ANPR dataset from Pearse St., Dublin was chosen for use in Chapter 6 of this
research. This decision was based on investigations carried out on each of the datasets. Of the two ANPR based travel time datasets discussed, the Pearse St. dataset is on a more relevant section of the road network than the Tallaght dataset in terms of investigating travel time on urban arterials, as Pearse St. is one of the busiest commuter routes in Dublin. The Vienna dataset is also taken from an important urban arterial. However, the use of floating car detection means the dataset is more an estimation of travel time based on the speed with which a fleet of taxis travel rather than a direct calculation as in the ANPR cases. The main issue of the Vienna dataset however is the large amount of missing data, with averages of travel times on several days needed to give an idea of the behaviour of travel time on a single complete day. Therefore, the Pearse St. ANPR dataset is used in Chapter 6 to test time travel prediction algorithms.
6 Chapter 6

TRAVEL TIME FORECASTING USING FORECAST REGIONS

6.1 Introduction

Of all the traffic parameters recorded and modelled by ITS, travel time has become the measurement most desired by motorists both pre-trip and during trip. Accurate travel time information can help motorists to plan their trips efficiently through ATIS. Also, from a traffic management point of view, estimation and prediction of travel time in near future have become an essential part of ATMS implementation in urban transport networks.

As such, travel time prediction is a key research question in short-term traffic forecasting literature. Recently, the focus of research in this area has moved from purely point based forecasts to predicting ranges of possible travel time within a higher and lower bound. Such prediction is more useful as it allows the motorist to estimate their personal travel time based on their driver behaviour i.e. risk-averse motorists might consider values near the higher bound of the predicted travel time range as the more likely estimates of the travel times of their trip. It has also been suggested that the provision of a travel time range rather than a point forecast may improve driver satisfaction.

In this chapter, prediction algorithms have been developed to forecast travel time ranges. There have been various Forecast Region (FR) construction techniques recorded in the literature on travel time forecasting. One of the most popular prediction interval techniques throughout the literature is the bootstrap method. As such, the bootstrap technique is used in this work to create a plausibility interval, within which it is very likely that future values of a given time-series will fall. Four more novel techniques for FR creation are compared against the base results of the bootstrap method. In order to produce the four additional FRs, a prediction framework is necessary. In this regard, the main prediction algorithm is based on ANN algorithms as described in Chapters 2, 3 and 4. The weather adaptive SWT-ACNN algorithm
described in Chapter 4 has been utilised in this chapter. The travel time data as collected by ANPR cameras in the Pearse Street area in Dublin city centre (as described in Chapter 5) has been modelled using the aforementioned prediction algorithm. In conjunction with the SWT-ACNN prediction framework four different FR construction algorithms are used: Delta method, Bayesian Hierarchical Interval Estimation (BHIE) method and Inductive Conformal Prediction (ICP) methods using a Standard nonconformity measure (S-ICP) and a Normalised nonconformity measure (N-ICP) (Papadopoulous and Haralambous, 2011).

The work in this chapter is organised into five sections. Following the introduction in Section 6.1, the methodology of the prediction models is discussed in Section 6.2. Section 6.3 briefly describes the data used in this chapter and the setup of the prediction models used. The results of the predictions are presented in Section 6.4 while the conclusions are discussed in Section 6.5.

6.2 Methodology

In this chapter four different prediction model setups are used: standard ACNN using only travel time data as input, standard ACNN using travel time along with rainfall data as an additional input, SWT-ACNN using only travel time data as input and SWT-ACNN using travel time along with rainfall data as an additional input. These four models are used to predict future 15 minute average travel time observations. Then, four different types of FR construction techniques are used to generate minimum and maximum expected deviations from the point forecasts. A fifth FR is created using the bootstrap method, independent of the forecasting model type. The bootstrap method used in this work is based on the travel time dataset itself and is used as a base FR against which the other FR creation techniques are compared. In order to ascertain the quality of the predictions and their respective FRs, the four prediction models are evaluated using MAPE and the FRs are evaluated using FR assessment indices.

The standard ACNN is the same as described in Section 2.2 in Chapter 2. The ANN models used in this study are based on the SWT-ACNN theory described in Section 4.2. As
described, autocorrelated input vectors are used in tandem with wavelet decomposition in an
effort to achieve accurate travel time prediction. The SWT-ACNN models produce a point
forecast. However, the construction of FR, as described in the next subsection, generates an
expected range based on the point forecasts produced.

6.2.1 Forecast Region Construction

The construction of the FRs for use in this chapter is documented in the forthcoming
subsections, starting with the Bootstrap technique (Sun and Zhang (2004)).

The Bootstrap Technique

The Bootstrap technique as used in this work is a technique is based on the travel time data set,
as opposed to being based on the prediction technique. It assumes that the travel time on a given
day will be reasonably similar to the travel time in the next day or next week i.e. it assumes the
travel time dataset to be somewhat consistent over given timeframes (day to day, week to week
etc.). The idea was to use a simplistic FR creation technique and then compare the other more
novel methods against this base FR. The Bootstrap technique works as follows. Assume the
input and targets for the prediction models are all members of the set, $S$ as in Equation (6.1):

$$S = \{(x_1, y_1), ..., (x_n, y_n)\}$$

(6.1)

Set $S$ is in fact the travel time dataset used by the prediction models in this work. However, the
Bootstrap model used does not need to know which predictive models are used. It is a simple
and quick method of generating a plausibility interval inside which future points of this travel
time dataset are likely to reside. Several random samples, $P_{(r)}$, are taken from within the entire
set $S$ as shown in Equation (6.2):

$$P_{(r)} = \{(x_3, y_3), ..., (x_{22}, y_{22})\}$$

(6.2)

Note that Equation (6.2) just contains an example random subset taken from the set $S$. Many
different random subsets are generated and the mean of all these different subsets $P_{(r)}$ are
recorded as in Equation (6.3):
\[ \mu(i) = \text{mean}(P_{(i)}) \] (6.3)

A histogram is then generated based on the values of all the different subset means recorded. The 5th and 95th percentiles \((a \& b \text{ respectively})\) of this distribution are then taken and the upper and lower bounds of the FR are generated as in Equation (6.4):

\[ \hat{y}_i \pm \left( \frac{b - a}{2} \right) \] (6.4)

Based on the percentiles, it is assumed that any future travel time data point has a high chance of falling within the bounds of this simply constructed dataset-based FR.

**The Delta Technique**

In order to construct FRs using the Delta technique (Khosravi et al., 2011), ANNs must be interpreted as Non-Linear Regression (NLR) models. Based on this representation, ANNs can be formulated as in Equation (6.5):

\[ y_i = f(x_i, i^*) + v_i, \quad i = 1, 2, \ldots, n \] (6.5)

where \(x_i\) and \(y_i\) are the \(i^{th}\) set of inputs and the corresponding targets (\(n\) observations) respectively, \(v_i\) is the noise of the model with zero mean, \(f(\cdot)\) with \(i^*\) is the non-linear function representing the true regression function. An estimate of \(i^*\), called \(\hat{i}\) can be found by minimising the Sum of Squared Error (SSE) cost function:

\[ SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - f(x_i, \hat{i}))^2 \] (6.6)

A first-order Taylor's expansion of \(f(x_i, i^*)\) around the true values of the model parameters \(i^*\) can be expressed as:

\[ \hat{y}_i = f(x_i, i^*) + g^T \hat{y}_i (\hat{i} - i^*) \] (6.7)
where \( g^T \) is the gradient of \( f(\cdot) \) (which represents the ANN models) with respect to the parameters \( \hat{i} \) calculated for \( i^* \). Assuming the noise, \( \nu_i \) in Equation (6.5), is independently and normally distributed, \( N(0, \sigma^2) \), then the \((1 - \alpha)\%\) FR for \( \hat{y}_i \) is:

\[
\hat{y}_i \pm t_{df, \frac{\alpha}{2}} \frac{\sigma}{\sqrt{1 + g^T \hat{y}_i (J^T J)^{-1} g \hat{y}_i}}
\]

(6.8)

where \( t_{df, \frac{\alpha}{2}} \) is the \( \frac{\alpha}{2} \) quantile of cumulative t-distribution function with \( df \) degrees of freedom. In this case \( df \) is the difference in the number of training samples, \( n \), and the number of ANN parameters, \( p \), \( s \) is the estimation of standard deviation and \( J \) is the Jacobian matrix of the ANN model with respect to its parameters. This concludes the theory behind the Delta method FR construction in this chapter.

**Bayesian Hierarchical Interval Estimation**

The second FR construction technique described in this work is the Bayesian Hierarchical Interval Estimation (BHIE) technique. Bayesian methods have often been used to estimate the parameters of traffic flow models. In this work, Bayesian methods are combined with wavelet decomposition to create a novel FR. The technique was described as a Bayesian hierarchical model of the residues of a time-series decomposed by DWT (Ghosh et al., 2010). The variance of the residual was observed to be varying with time and so the residual (the difference between each point of the time-series and the trend or DWT approximation of that point) of the travel time time-series in this work, generated after fitting a trend model, is modelled using a Bayesian hierarchical model to take into account the time-variant changes in variance values. A normal hierarchical model can be represented as:

\[
R_i \sim N(\mu, \sigma_i^2) \quad i = 1, 2, \ldots, n
\]

(6.9)

where, \( \mu \) is the sample mean of the residual and \( \sigma_i \) is the standard deviation of the residual at time \( i \). Analysing 48 point daily travel time datasets means that \( n = 48 \) in this instance. As
mentioned earlier, the variance \( \sigma^2 \) of the residual dataset \( R \) varies throughout the day. To take into account this time-varying variance, the following model was developed:

\[
\log(\sigma) \sim N\left(\log(y), \tau^2\right) \quad \text{or} \quad \sigma \sim LN\left(\log(y), \tau^2\right) \quad (6.10)
\]

The lognormal distribution in Equation (6.10) ensures all \( \sigma \) fall within \((0, \infty)\), as \( \sigma \) will always be positive. The variance \( \tau^2 \) in Equation (6.10) is assumed to follow a uniform distribution, within the range \((0, v)\), where \( v \) is an arbitrary constant defining the maximum limit of \( \tau \). The unknown parameters to be estimated in the Bayesian hierarchical model can be represented as the vector \( \vartheta = (\tau, \sigma_1, \sigma_2, \ldots, \sigma_{48})^T \). This vector is estimated using a Bayesian estimation technique described briefly here. For the Bayesian inference, the posterior density of the normal hierarchical model is:

\[
p(\vartheta | R, i) = p(\tau) L(\sigma | R, i) L(\tau | \sigma, i) \quad (6.11)
\]

where, \( p(\vartheta | R, i) \) is the posterior density of \( \theta \), \( L(\sigma | R, i) \) is the likelihood function of \( \sigma \), \( L(\tau | \sigma, i) \) is the likelihood function of \( \tau \), and \( p(\tau) \) is the prior density of parameter \( \tau \). Having stated in Equation (6.9) that \( R \) follows a normal distribution, \( L(\sigma | R, i) \) can be rewritten as:

\[
L(\sigma | R, i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi \sigma_i^2}} \exp\left(-\frac{R_i^2}{2\sigma_i^2}\right) \quad (6.12)
\]

In a similar vein, building on the assumption in Equation (6.10), \( L(\tau | \sigma, i) \) can be rewritten as:

\[
L(\tau | \sigma, i) = \prod_{i=1}^{n} \frac{1}{\sigma_i \tau \sqrt{2\pi}} \exp\left(-\frac{(\log \sigma_i - \log y_i)^2}{2\tau^2}\right) \quad (6.13)
\]

The prior density of \( \tau \) is assumed to be flat on the range \((0, \infty)\) and thus \( p(\tau) \) can be treated as a constant. Taking all these expansions into account, the posterior density from Equation (6.11) can be rewritten as:
Equation (6.14) can be simplified further to:

\[ p(\theta | R, t) \propto \prod_{i=1}^{n} \frac{1}{\sigma_i} \exp \left(-\frac{R_i^2}{2\sigma_i^2} \right) \prod_{i=1}^{n} \frac{1}{\sigma_i} \exp \left[-\frac{(\log \sigma_i - \log y_i)^2}{2\tau^2} \right] \]  

(6.14)

This in turn can be rewritten as follows:

\[ p(\theta | R, t) \propto \frac{1}{\tau^n} \prod_{i=1}^{n} \frac{1}{\sigma_i} \exp \left(-\frac{R_i^2}{2\sigma_i^2} \right) \left[ (\log \sigma_i - \log y_i)^2 \right] \]  

(6.15)

The Markov Chain Monte Carlo (MCMC) simulation method (Ghosh et al., 2010) is then utilised to find the marginal distributions of each of the unknown parameters from equation (6.16). In order to simulate the marginal probability distributions for the unknown parameters in \( \theta \), the initial conditions are defined as \( \theta = \{ \tau^{(mn)}_{i}, \sigma_1^{(mn)}, \sigma_2^{(mn)}, ..., \sigma_n^{(mn)} \} \) and the following 49 steps are iterated, with \( mn \) denoting the iteration:

1. Use the Gibbs sampler technique to sample \( \tau^{(mn+1)} \) from

\[ p(\tau^{(mn+1)} | \sigma_1^{(mn)}, \sigma_2^{(mn)}, ..., \sigma_n^{(mn)}, y_i) \]

2. Use the Metropolis Hastings technique to sample \( \sigma_1^{(mn+1)} \)

48. Use the Metropolis Hastings technique to sample \( \sigma_n^{(mn+1)} \)

The initial conditions are defined as:
In step 1, the Gibbs sampler technique is utilised for simulation of the distribution of $\tau$. Building on the posterior density described in Equation (6.16), the full conditional distribution for $\tau$ is:

$$p(\tau | \sigma, R, i) \propto \left( \frac{1}{\tau^n} \right) \exp \left( -\frac{\sum_{i=1}^{n} (\log \sigma_i - \log y_i)^2}{2\tau^2} \right)$$ (6.18)

The full conditional distribution of $\tau$ can be observed as an inverse gamma distribution with parameters defined as follows:

$$Y = \frac{n}{2} - 0.5$$ (6.19)

$$Z = 0.5 \sum_{i=1}^{n} (\log \sigma_i - \log y_i)^2$$ (6.20)

where the density function of the inverse gamma distribution is described as follows:

$$p(\tau) = \frac{Z^\tau}{\Gamma(Y)} \tau^{-(Y+1)} \exp \left( -\frac{Z}{\tau} \right)$$ (6.21)

Steps 2 to 49 of the algorithm simulate the values of $\sigma_1, \sigma_2, \ldots, \sigma_n$ in each iteration, using the Metropolis Hastings technique. The candidate values of each element of the $\sigma$ vector are simulated from the following proposal distribution:

$$\sigma^{mn} \sim LN[\log(y), (\tau^{mn})^2]$$ (6.22)
According to the Metropolis algorithm, each simulated value of the elements of the vector $\sigma^{mn}$ in each iteration is accepted with probability of either:

$$p(\sigma_i) = \frac{p(\sigma_i | \tau^{mn}, \sigma_1^{mn-1}, \sigma_2^{mn-1}, ..., \sigma_{i-1}^{mn-1}, \sigma_i, \sigma_{i+1}^{mn-1}, ..., \sigma_n^{mn-1}, R, i)}{p(\sigma_i | \tau^{mn}, \sigma_1^{mn-1}, \sigma_2^{mn-1}, ..., \sigma_{i-1}^{mn-1}, \sigma_i, \sigma_{i+1}^{mn-1}, ..., \sigma_n^{mn-1}, R, i)}$$

(6.23)

or 1, based on whichever is the minimum value. Steps 1 to 49 are repeated 10000 times in order to simulate 10000 values for each of the unknown parameters. Plotting the mean of these 10000 simulated values, at each iteration, displays the effectiveness of the Bayesian hierarchical model at capturing the time varying variance present in travel time time-series. Finally, the FR itself is created by constructing a $(1 - \alpha)\%$ confidence interval on the regular average trend of the travel time time-series. By utilising the advantages of such a Bayesian model, the FR adapts according to the variability of the residual data. The novelty of this FR is considerable and it has not been used in combination with forecasting travel time data in the literature.

**Inductive Conformal Prediction based Methods**

The two FR techniques described in this section are documented in a paper by Papadopolous and Haralambous (2011). They follow a machine learning framework called Conformal Prediction (CP) that assigns confidence measures to forecasts on the simple assumption that the data are independent and identically distributed. For more background on this subject, Papadopolous and Haralambous (2011) should be referred to. The first model uses an ANN regression ICP technique with a standard nonconformity measure (S-ICP), while the second builds upon the first technique through the introduction of a normalised nonconformity measure (N-ICP). The first step of this FR construction method is to define a nonconformity measure, a function that measures the difference between the true value of $y_i$ and the predicted $\hat{y}_i$ created by forecasting with the input $x_i$. In the case of regression, the nonconformity measure is defined as:
With a nonconformity measure defined, the first steps of the FR construction scheme are detailed as follows:

- The training set \( \{(x_i, y_i), \ldots, (x_n, y_n)\} \) is divided into two subsets.

- The first subset is deemed the proper training set: \( \{(x_i, y_i), \ldots, (x_p, y_p)\} \)

- The second subset is known as the calibration set: \( \{(x_{p+1}, y_{p+1}), \ldots, (x_q, y_q)\} \)

- The ANN model is trained on the proper training set: \( \{(x_i, y_i), \ldots, (x_p, y_p)\} \)

- Then, for each pair of the calibration set, the input pattern \( x_{j+1} \) is presented to the trained ANN to compute a prediction \( \hat{y}_{j+1} \)

- The nonconformity score \( I_{j+1} \) is then calculated for these predictions using Equation (6.24)

- The nonconformity scores of the calibration set are then sorted in descending order which yields the sequence: \( I_{(j+1)}, \ldots, I_{(j+q)} \)

- For any new input pattern \( x_{i+p} \) presented to the ANN, the prediction \( \hat{y}_{i+p} \) is used to compute the FR using the formula: \( (\hat{y}_{i+p} - I_{(j+1)}, \hat{y}_{i+p} + I_{(j+1)}) \), where \( s = [\hat{\delta}(q + 1)] \) and \( \delta \) is chosen as a variable such that p-value of the maximum and minimum \( \hat{y} \) exceeds \( \delta \)

For clarity, the third FR construction technique (S-ICP) presented in this chapter is:

\[
\hat{y}_i \pm I_{(j+2)} \quad (6.25)
\]

The final FR used in this chapter builds on the FR in Equation (6.25) by introducing a normalised nonconformity measure. The nonconformity measure is normalised in an attempt to create FRs that are larger for more difficult predictions and smaller for forecasts expected to be easier. The new normalised nonconformity measure is defined as:
\[
I_i = \frac{|y_i - \hat{y}_i|}{\exp(D_i) + M}
\]  

(6.26)

where \( \hat{y}_i \) is the forecasted value of \( \ln(|y_i - \hat{y}_i|) \) generated using a separate linear ANN trained on the proper training set and \( M \) is a sensitivity parameter designed to control the effect of changes in \( D \), on the normalised nonconformity measure. Accordingly, the final FR in this chapter (N-ICP) is as follows:

\[
\hat{y}_i \pm I_i \left( \exp(D_i) + M \right)
\]  

(6.27)

where the value of \( s \) is as defined earlier in this section.

### 6.2.2 Forecast Region Assessment Indices

Having described the various FR construction techniques used in this chapter, the next step in this work is to ascertain how effective the different FRs are. Each FR consists of a lower and upper bound around a point forecast. These bounds create an area within which future predicted points are expected to lie with a set probability \((1 - \alpha)\%\). The first FR assessment index used in this work is a measure of how many forecasted points fall within the bounds of the FR. This measurement is called the Forecast Region Coverage Probability (FRCP) and is defined as:

\[
FRCP = \frac{1}{n} \sum_{i=1}^{n} c_i
\]  

(6.28)

where, \( c_i = 1 \) if \( y_i \in \left[ Lw(x_i), Up(x_i) \right] \) and \( c_i = 0 \) if \( y_i \notin \left[ Lw(x_i), Up(x_i) \right] \), \( Lw(x_i) \) and \( Up(x_i) \) representing the lower and upper bounds of the FR relating to the \( i^{th} \) sample. It follows that a high value of FRCP means that most predicted observations fall within the upper and lower bounds of the corresponding FR, while a low value of FRCP suggests the FR has not been effective at creating a range expected to contain future observations. However, FRCP alone is not enough to accurately gauge how good or effective a FR is. For instance, an extremely large FR can comfortably achieve a FRCP of 100%, but realistically such a FR is of little use. Therefore, other FR assessment indices exist to determine the usefulness of FRs. One such
measure is the Mean Forecast Region Length (MFRL) that quantifies the length of a FR. MFRL is defined as:

\[
MFRL = \frac{1}{n} \sum_{i=1}^{n} (Up(x_i) - Lw(x_i))
\]  

(6.29)

An additional FR assessment measure is the Normalised Mean Forecast Region Length (NMFRL) which is calculated by dividing MFRL by the target range \(\text{Range}\):

\[
NMFRL = \frac{MFRL}{\text{Range}}
\]  

(6.30)

The idea behind normalising against the range of the target is to allow comparisons of FRs of different targets. NMFRL is a dimensionless quantity that represents the average length of FRs as a percentage of the range of the target. In general, a high value of NMFRL means that the FRCP value will also be high. In fact if the upper and lower limits of a FR are set to the extreme target values, both NMFRL and FRCP will be 100%. Combining NMFRL and FRCP into a single index is the best way to judge the usefulness of a FR. The Coverage Length-based Criterion (CLC) is designed to penalise FRs whose FRCP value is below a nominal confidence interval, regardless of the related NMFRL value:

\[
CLC = NMFRL \left(1 + e^{-U(\text{FRCP} - N)}\right)
\]  

(6.31)

where, \(U\) and \(N\) are hyperparameters determining the level of penalty given to FRs with low coverage probability. As can be seen from Equation (6.31), \(U\) is used to amplify any difference between FRCP and \(N\). In this work, FRCP, MFRL, NMFRL and CLC will be used to give an indication as to which of the four FR construction techniques used in this chapter is the most suitable.
6.3 Data

The travel time data used in this chapter is taken from the travel time dataset constructed in Chapter 5. In keeping with the fact that weekday and weekend traffic behaviour are substantially different, the predictive models in this chapter focus on weekday travel time data only. The data is taken from four weeks in from 30th July, 2012 to 24th August, 2012. This comprises 20 weekdays of travel time data. As described in Section 5.4, there are gaps in the travel time data recorded by the ANPR cameras. Therefore, the work in this chapter focuses on 15 minute intervals within 12 daytime hours from 08:00 hours to 20:00 hours. A sample day composed of a single 48 point 12 hour segment is presented in Figure 6.1.

![Figure 6.1: Average Travel Time in 15 Minute Intervals from 08:00 - 20:00 hours](image)

The average 15 minute interval travel time data shown in Fig. 6.1 displays a clear increase in travel time around the evening peak at 18:00 hours. Although the data has a somewhat noisy look to it, in the main the average travel time just fluctuates between 30 and 60 seconds apart from the evening peak. Crucially, there are no missing values in the dataset which is an
advantage when it comes to predicting future observations. Table 6.1 contains the mean and standard deviation of the chosen 20 day, 08:00 to 20:00 hours, travel time dataset.

| Table 6.1: Travel Time Dataset Descriptive Statistics |
|---------------------------------|-----------------|
| Mean ($\mu$)                    | 61.9742         |
| Standard Deviation ($\sigma$)   | 27.7633         |

The rainfall data used in this chapter has again been sourced from Met Eireann, as in Section 4.3.1. In order to keep the rainfall in sync with the travel time data, the rainfall data has also been divided into 48 point segments, each covering the 12 hours from 08:00 hours to 20:00 hours. A barchart of the rainfall occurring in the five 48 point segments of the rainfall dataset representing July 30th to August 3rd 2012 is presented in Fig 6.2.

Figure 6.2: Rainfall Barchart for 08:00 hours to 20:00 hours from July 30th to August 3rd 2012
Rainfall was reasonably frequent in the 5 days presented in Fig 6.2. With the effect of rainfall on traffic conditions already documented in this thesis, the amount of rain in Fig 6.2 should allow thorough comparisons between travel time predictions using rainfall as an additional model input and those without. Having described the data used in this chapter, the next section discusses setup of the prediction models.

6.4 Predictions and Comparisons

In this chapter four different prediction model setups are used: SWT-ACNN with rainfall as an additional input, SWT-ACNN using only travel time data as input, standard ACNN with rainfall as an additional input and standard ACNN using only travel time data. These four models are used to predict future 15 minute average travel time observations. Then, four different types of FR construction techniques are used to generate minimum and maximum expected deviations from the point forecasts, in addition to the Bootstrap FR creation technique which is not dependent on prediction model. In order to ascertain the quality of the predictions and their respective FRs, the four prediction models were tested for accuracy using MAPE and the FRs were examined using the FR assessment indices described in Section 6.2.3. The accuracy of the point forecasts is examined first. Table 6.2 presents the prediction accuracy for the four alternative model setups.

<table>
<thead>
<tr>
<th></th>
<th>MAPE (%)</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWT-ACNN with rainfall</td>
<td>8.0367</td>
<td>7.8838</td>
</tr>
<tr>
<td>SWT-ACNN without rainfall</td>
<td>8.5288</td>
<td>8.2958</td>
</tr>
<tr>
<td>ACNN with rainfall</td>
<td>18.6455</td>
<td>22.6586</td>
</tr>
<tr>
<td>ACNN without rainfall</td>
<td>19.3992</td>
<td>25.8018</td>
</tr>
</tbody>
</table>

It is important to note that for each of the four prediction setups, by trial and error the optimum number of neurons for each model were determined. The values for MAPE and RMSE in Table 6.2 apply to the optimum prediction accuracy achieved by each model. The first conclusion to take from the results shown in Table 6.2 is that the SWT-ACNN models outperform the
standard ACNN models regardless of input selection. This proves again that wavelet decomposition is a very effective tool for time-series forecasting. SWT proved effective when predicting 15 minute aggregate average travel time data time-series having previously proven to be a great aid in computing traffic flow time-series forecasts in Chapter 4. The second big conclusion to be taken from Table 6.2 is that, of the two SWT based travel time forecasting models, the model that takes rainfall into account as an additional model input outperforms the model using travel time data alone. This suggests that weather conditions, and rainfall in particular, have a strong effect on travel time in urban arterials. This is an important finding as there was a reasonable amount of rainfall on the predicted day in question, and the rainfall versions of both the SWT-ACNN and the standard ACNN outperformed the models without rainfall as an additional input. Having discussed the point forecasts, the FRs are examined next.

In this work, five different FRs are created using various techniques: the Bootstrap technique, the Delta technique, the BHIE method and two methods based on ICP using a standard nonconformity measure and a normalised nonconformity measure. The first FR displayed is the FR created by the Bootstrap method. As discussed previously, this is a simplistic method of generating an FR based purely on the dataset at hand. The predictions from the 4 different forecasting algorithms are denoted by the legend in Fig 6.3 and the FR itself is shown as the continuous black lines on the graph.
The Bootstrap method creates a reasonable FR for the travel time data. It can be seen in Fig 6.3 that the FR is quite wide, resulting in most of the target points falling within the bounds. However, the large width also has drawbacks, such as the fact it does not necessarily give the user a small yet accurate region within which predicted travel time data should fall. Again, the Bootstrap method is the single method in this work where the FR is based on the dynamics of the travel time dataset, rather than the predictive model. The following FRs in this section are generated based on predictions. It is important to clarify that the FRs displayed in the following figures are based on the results of the SWT-ACNN with rainfall model as it produced the most accurate forecasts. The best forecasts from the three other prediction models are included in the figures for illustrative purposes, but the FR bounds are constructed using the SWT-ACNN with rainfall model predictions. The FR created using the Delta method is shown in Fig 6.4 along with the target travel time observations and the predictions generated by the four different prediction models.
It is clear from Fig 6.4 that the predictions of all the travel time point forecast models lie within the upper and lower bounds of the FR. This fact, coupled with the fact that the FR is not overly wide, apart from during the peak travel time around 18:00 hours, suggest that the Delta technique is a favourable option for FR construction. The relative inaccuracy of the two standard ACNN models is also apparent in the figure as predictions from these two models are generally further away from the original observations than the SWT-ACNN predictions, again particularly around 18:00 hours. The second FR, constructed using the Bayesian method, is shown in Figure 6.5.
The first notable conclusion to draw from Fig 6.5 is that the FR created using the BHIE method is tighter than the Delta FR in Fig 6.4. It is also immediately clear that the distance between the upper and lower bounds of the FR varies greatly throughout the figure, becoming noticeably small around 12:30 hours and much wider again at 19:00 hours. This is due to the methodology of the Bayesian FR construction technique as described in Section 6.2.2. It is promising that the prediction results of the two SWT-ACNN models almost all fall within the FR limits. The standard ACNN prediction model results lie outside the FR bounds when it tightens significantly around 12:30 hours and again when the travel time begins to rise substantially at 16:00 hours. The third FR, based on ICP and using a standard nonconformity measure is presented in Fig 6.6.

Figure 6.5: FR created using Bayesian Hierarchical Interval Estimation

The first notable conclusion to draw from Fig 6.5 is that the FR created using the BHIE method is tighter than the Delta FR in Fig 6.4. It is also immediately clear that the distance between the upper and lower bounds of the FR varies greatly throughout the figure, becoming noticeably small around 12:30 hours and much wider again at 19:00 hours. This is due to the methodology of the Bayesian FR construction technique as described in Section 6.2.2. It is promising that the prediction results of the two SWT-ACNN models almost all fall within the FR limits. The standard ACNN prediction model results lie outside the FR bounds when it tightens significantly around 12:30 hours and again when the travel time begins to rise substantially at 16:00 hours. The third FR, based on ICP and using a standard nonconformity measure is presented in Fig 6.6.
Figure 6.6: Standard Nonconformity Measure ICP Based FR Construction Method

It is immediately clear that this construction technique results in the tightest overall FR. This can be viewed in two alternative ways. From a positive point of view, the relative narrowness of the FR means that there is less uncertainty in the spread of possible future observations because the upper and lower limits are quite close to the point forecasts for the most part. However, the tightness of the FR means that very accurate point forecasts are needed; the forecasts generated using the standard ACNN models for instance are located outside the FR quite often from 16:00 hours to 18:00 hours and indeed at other portions of the graph. Again, the SWT-ACNN models perform strongly with almost all predicted points using these models lying within the bounds of the FR. The final FR, the ICP-based technique using a normalised nonconformity measure, is displayed in Fig 6.7.
The final FR appears to fit the travel time data quite well and the clear majority of points from all prediction types fall within the limits of the FR. This is partly due to the relatively large width of the FR. However, the large width of the FR either side of the target observations means that this is the only FR to have a lower limit containing some negative values (particularly between 08:00 and 11:00 in Figure 6.7). In summary, this FR contains the vast majority of points due to its large width, but the value of such a wide FR is questionable as it gives a very large range of possible forecasted average travel time values whereas motorists would obviously prefer a little less uncertainty in predictions. In order to empirically judge the worth of each of the four FRs used in this chapter, the FR assessment measures described in Section 6.2.3 are presented in Table 6.3
Table 6.3: FR Assessment Indices

<table>
<thead>
<tr>
<th></th>
<th>FRCP</th>
<th>NMFRL</th>
<th>CLC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SWT-ACNN with Rainfall</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bootstrap Method</td>
<td>100.0000</td>
<td>32.7273</td>
<td>32.7273</td>
</tr>
<tr>
<td>Delta Method</td>
<td>100.0000</td>
<td>46.3662</td>
<td>46.3662</td>
</tr>
<tr>
<td>BHIE</td>
<td>100.0000</td>
<td>24.9676</td>
<td>24.9676</td>
</tr>
<tr>
<td>S-ICP</td>
<td>91.6667</td>
<td>17.4275</td>
<td>17.4316</td>
</tr>
<tr>
<td>N-ICP</td>
<td>100.0000</td>
<td>47.9852</td>
<td>47.9852</td>
</tr>
<tr>
<td><strong>SWT-ACNN with Travel Time Only</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bootstrap Method</td>
<td>100.0000</td>
<td>32.7273</td>
<td>32.7273</td>
</tr>
<tr>
<td>Delta Method</td>
<td>100.0000</td>
<td>46.3662</td>
<td>46.3662</td>
</tr>
<tr>
<td>BHIE</td>
<td>97.9167</td>
<td>24.9676</td>
<td>24.9676</td>
</tr>
<tr>
<td>S-ICP</td>
<td>91.6667</td>
<td>17.4275</td>
<td>17.4316</td>
</tr>
<tr>
<td>N-ICP</td>
<td>100.0000</td>
<td>47.9852</td>
<td>47.9852</td>
</tr>
<tr>
<td><strong>ACNN with Rainfall</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bootstrap Method</td>
<td>89.5833</td>
<td>32.7273</td>
<td>33.2347</td>
</tr>
<tr>
<td>Delta Method</td>
<td>100.0000</td>
<td>46.3662</td>
<td>46.3662</td>
</tr>
<tr>
<td>BHIE</td>
<td>87.5000</td>
<td>24.9676</td>
<td>49.9353</td>
</tr>
<tr>
<td>S-ICP</td>
<td>72.9167</td>
<td>17.4275</td>
<td>8.0939e+13</td>
</tr>
<tr>
<td>N-ICP</td>
<td>91.6667</td>
<td>47.9852</td>
<td>47.9967</td>
</tr>
<tr>
<td><strong>ACNN with Travel Time Only</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bootstrap Method</td>
<td>81.2500</td>
<td>32.7273</td>
<td>8.7820e+06</td>
</tr>
<tr>
<td>Delta Method</td>
<td>100.0000</td>
<td>46.3662</td>
<td>46.3662</td>
</tr>
<tr>
<td>BHIE</td>
<td>77.0833</td>
<td>24.9676</td>
<td>2.7873e+10</td>
</tr>
<tr>
<td>S-ICP</td>
<td>91.6667</td>
<td>17.4275</td>
<td>5.2206e+15</td>
</tr>
<tr>
<td>N-ICP</td>
<td>93.7500</td>
<td>47.9852</td>
<td>47.9854</td>
</tr>
</tbody>
</table>

FRCP and NMFRL both contribute to the CLC calculation and the best FRs are those with a high FRCP and low NMFRL, thus also having a low CLC. Based on the information presented in Table 6.3, the FRs constructed using the S-ICP technique are the ‘best’ FRs when combined with the SWT-ACNN based forecasting models, with the lowest CLC values of 17.4316. This low score is due to how tight the S-ICP FR is to the point forecasts generated using the SWT-ACNN models. However, it must be noted that the S-ICP FRs perform very poorly with the standard ACNN travel time forecasting models. The Bootstrap method, Delta method and N-ICP FRs are the most well rounded of the FRs in that they generally have high FRCP values, with almost all predicted points from the four different forecasting models falling within the bounds of the FR, and reasonable NMFRL values; hence their CLC values are fairly uniform across the four prediction algorithms. The BHIE technique performs very well with the SWT-
ACNN models, as recognised by CLC values of 24.9676 in both cases, but suffer somewhat with the standard ACNN model forecasts.

In order to display the single best FR and best forecast, an alternative graph displaying the S-ICP FR and the SWT-ACNN prediction including rainfall as an input is shown in Fig 6.8. It is clear from this figure that the predicted travel time points (coloured in red) are almost always within the FR and very close to the target observations (coloured in blue). The FR itself is displayed as the vertical black lines above and below every target travel time observation.

Figure 6.8: SWT-ACNN with Rainfall Prediction (Red) vs. Target Travel Time Data (Blue)

This FR and forecasting model combination produces forecasts with ranges that are very acceptable and appealing to motorists on the basis that the forecasted points are almost always accurate enough to fall within the relatively tight FR bounds.
6.5 Conclusion

In this chapter, point forecasts of 15 minute aggregate average travel time data were computed using four different ACNN models. Two models availed of the SWT techniques described in Chapter 4 while the other two models were standard ACNN models as in Chapter 2. On top of this, one SWT-ACNN model and one standard ACNN model used rainfall data as an additional input, whereas the other two models forecasted using travel time data as the lone input. In terms of prediction accuracy, the SWT-ACNN models far outperformed the standard ACNN models. The findings based on using rainfall as an additional input for predicting travel time data were also conclusive. The SWT-ACNN and standard ACNN models with rainfall as an additional model input outperformed their respective models using travel time as the only input. As in Chapter 5, the rainfall input models learned through network training how travel time was affected when rainfall did occur, and with substantial rainfall occurring on the target day in question, the advantage of rainfall as an additional input was witnessed clearly in the results. The single best forecast produced by any model in this chapter was generated by the SWT-ACNN model using rainfall as an additional input.

The other notable research performed in this chapter related to the study of FR construction techniques for travel time forecasts. A range of possible values for future observations is something strongly desired by motorists as a point forecast can frustrate if it doesn't relate to the true traffic conditions. Simply put, motorists would rather have an accurate range of future travel time observations than single point forecasts which may not be 100% accurate. With that in mind, five different FR construction techniques were compared in this chapter: the Bootstrap FR construction technique, the Delta FR construction technique, the BHIE construction technique, the S-ICP based FR construction technique and the N-ICP based FR construction technique. The study concluded that the S-ICP method was the best FR construction technique when combined with the SWT-ACNN forecasts by virtue of having the lowest CLC value of the four tested FRs.
In summary, the work in this chapter made use of various techniques studied in previous chapters. The culmination of these different bodies of research was a travel time forecast generated by the SWT-ACNN model using rainfall and average travel time as inputs, coupled with the S-ICP method FR, giving motorists an excellent resource both pre-trip and during trip, allowing them to make educated route choice decisions.
Chapter 7

CONCLUSION

7.1 Research Summary

Models for forecasting traffic condition related variables are developed in this thesis with a view to improve the flexibility and implementation potential of ATMS and ATIS systems in the effort of enhancing the sustainability and efficiency of modern transport networks. The models have been generally created with ANN structures and have been validated using various traffic condition variable datasets from Dublin and the motorway network of the U.K.

The first set of models in this work was designed to forecast traffic flow on urban arterials and motorways using FFBPNN, RBFNN, GRNN and SVM algorithms. These models produced forecasts of traffic flow at four different time aggregations: 5 minute, 15 minute, 30 minute and 1 hour. The traffic flow data was obtained from the SCATS database in Dublin and the MIDAS database from the U.K. The concept of an ACNN was introduced and tested on both urban arterial and motorway data. The ACNN algorithm was designed to reduce input dataset size without compromising the accuracy of the predictions. The ACNN models were proved to outperform standard Naïve and Moving Average forecasts. ACNN-based forecasts of urban arterial traffic flows were compared with forecasts of motorway traffic flows in order to see how the different traffic behaviour at these sites would affect the prediction accuracy.

The next body of work focused on using the relationship between traffic speed and flow to improve forecast accuracy. Based on the linearity of the relationship between flow and speed, a novel algorithm was developed to determine whether the traffic conditions were congested or uncongested. Separate models for uncongested and congested conditions were then used to forecast the future observations of said traffic variables. This novel regime isolation methodology also used ANNs as the forecasting algorithm.
The introduction of stationary wavelet based techniques to traffic flow forecasting was another feature of the work in this thesis. Building on the use of DWT in the literature, SWT was used to decompose traffic flow time-series to its constituent components in an effort to model the behaviour at different frequency resolutions separately. This neuro-wavelet model was also novel in the manner in which it included rainfall data as an exogenous model input variable to take into account the effect of weather on traffic conditions. Rainfall does not impact traffic conditions immediately; rather it has a somewhat lagged effect. Hence, the use of SWT was particularly effective in combination with rainfall data as it allowed the model to view the effects of rainfall on the different decomposed frequency levels of traffic flow at different time instants.

Travel time is the most important variable that is required to be reported to drivers through ATIS systems to improve the efficiency of their journey planning. A travel time forecasting model has been developed using the neuro-wavelet technique accommodating the effect of rainfall. The modelled travel time dataset was collected from a major urban arterial route in Dublin through the use of ANPR cameras. To facilitate the applicability and practicality of the forecasting algorithm, a FR has been developed in addition to a point forecast. Five different FR construction techniques were compared in order to ascertain the most suitable prediction algorithm.

7.2 Research Findings and Critical Assessments

The thesis investigates and improves the different aspects of the field of STTF using the ANN algorithms. The thesis concentrates on improving the investigated paradigms through consideration of the physicality of traffic dynamics as well as the exogenous factors which have been identified to impact uncertainty in traffic patterns. In the following paragraphs the major inferences drawn through the research work are discussed.

The general ANN structure popular in STTF was augmented through the application of ACF to filter traffic variable inputs to the forecasting algorithms. This filtering improved the
prediction accuracies and computational times for both ANN and SVM based traffic flow forecasting models. The improvement in computational time was most noticeable at smaller data resolutions with large input vectors. Among the ANN structures, the RBFNN and GRNN models were competitive at all locations but in general showed poor performance accuracy at higher data resolutions. The SVM model on the other hand performed poorly at the largest time aggregation (1 hour) but was competitive at other intervals. The FFBPNN structure in conjunction with ACF based input filtering technique proved to be the most accurate forecasting model in the majority of locations and time aggregations. This finding was crucial for the development of further different forecasting algorithms in this thesis as it was chosen as the basic forecasting algorithm with a proven level of performance upon which novel techniques could be augmented.

An investigation into the effects of location on traffic flow forecasting confirmed that differing levels of prediction accuracies can be achieved at different locations despite the application of the same forecasting algorithm. This confirms that no traffic condition variable forecasting algorithms are truly generalizable. The differences in traffic dynamics at different locations are too great for a single forecasting model to be able to predict accurately at all locations. This finding holds true whether the different locations share the same classification e.g. two motorway locations, or if the locations are from different road types, e.g. a motorway and an urban arterial location. The lack of a generalizable model actually gives ANN models another positive attribute, as their fast computation times lessen the impact of having to retrain the forecasting model at each new location. The work in this thesis showed that ANN can be effective forecasting models for a variety of traffic condition variables (traffic flow, speed and travel time) at many different locations (various motorway sections and urban arterials), based on the forecast accuracies documented in Chapters 2, 3, 4 and 6.

The traffic flow forecasting study conducted in this thesis found that, in general, MAPE is related to time aggregation. It was found that in almost every case, the MAPE reduced with increasing time aggregation. This is mainly due to the fact that larger time aggregations consist
of coarser data as the noisy peaks found over small intervals in traffic flow data will be smoothed in the case of larger time aggregations. Based on the results in this work, it is suggested that a time aggregation of 15 minutes is a good compromise interval for traffic condition forecasting as it is neither too long so as to reduce dramatically the detail present in the data, nor is it too short so as to make future observations too unpredictable to be modelled accurately.

Traffic dynamics can be examined by looking at the traffic flow and speed relationships present at different locations. The relationship between flow and speed can give a great indication as to the type of traffic behaviour that occurs on a given urban arterial or motorway. In order to take advantage of the knowledge that the speed-flow relationship can give, a novel regime isolation methodology was developed in this work. This multivariate flow and speed forecasting algorithm used the speed-flow relationship to define whether the traffic conditions at a given time instance were congested or uncongested. Based on the identification of the regime, the model chose a suitable ANN model to forecast future observations of speed and flow. As described previously, no forecasting algorithm is truly generalizable and so it was of great benefit to have two separate ANN structures trained specifically to deal with observations from the congested and uncongested regions respectively.

Multivariate forecasting was approached from a different perspective in the creation of a novel weather adaptive neuro-wavelet forecasting model. Based on the knowledge that rainfall affects travel dynamics in a complex manner, the neuro-wavelet model was designed to be able to model these changes in traffic behaviour caused by rainfall at different frequency levels. SWT was used to decompose the original time-series into their approximation and detail components, in what is one of the first instances of SWT in the literature. On top of this, a switching algorithm was devised so that rainfall was only used as an exogenous model input when rainfall was expected in the near future. If rainfall was not expected, the model would perform as a simple neuro-wavelet model taking only traffic flow as its input. The model was evaluated using traffic flow and precipitation data from Dublin city centre and was found to be
more accurate than a standard ANN model in dry conditions. In addition, the Wet model outperformed the Dry model under inclement weather conditions, thus proving the potential of the weather adaptive neuro-wavelet prediction methodology.

The weather adaptive neuro-wavelet model was also employed in developing a travel time prediction methodology incorporating meaningful FRs for motorists. The travel time data was collected from a busy urban arterial in Dublin. Five different FR construction techniques were compared on the basis that travel time ranges are more suitable to ATIS applications than point forecasts. The Delta technique was found to be the best FR construction technique based on a series of FR assessment indices. The FR created using the Delta technique contained the vast majority of points from the point forecast, without being unnecessarily broad. This model would be ideal for ITS applications as it allows the motorist to estimate their travel time based on the knowledge of their individual driver behaviour.

7.3 Recommendations for Further Research

The work in this thesis can largely be summarised as traffic condition variable forecasting. Many different traffic condition variables are forecasted in this thesis, using several different forecasting models and several different time aggregations.

One avenue of future research that offers significant potential is that of the adaption of the regime isolation methodology in Chapter 3 to an urban transport network. The model produced excellent traffic flow and speed forecasts using motorway data sourced from the MIDAS database in the U.K. Adopting this framework for use in an urban environment would pose problems, notably in developing a method to take the effect of traffic signals into account. Additionally, the non-linear and linear speed-flow relationships present in motorway traffic, relating to congested and uncongested conditions respectively, may not be as distinct in an urban environment. However, if developed successfully, such a model would provide an excellent resource for future urban arterial traffic condition variable forecasting work.
The weather adaptive neuro-wavelet model developed in Chapter 4 of this thesis proved that taking rainfall into account when forecasting traffic flow can be very advantageous. However, the rainfall data in this work could only be obtained in hourly intervals. Therefore, an important extension of this work would be to test how the model performs at smaller time aggregations, should rainfall data of smaller time aggregations be obtainable. The excellent results using hourly data suggest that similar or better results could be achieved using 5, 15 or 30 minute interval traffic condition variable data.

The introduction of various FR construction techniques was an important part of the work in Chapter 6 of this thesis. These FRs built upon weather adaptive travel time predictions obtained using FFBPNN based prediction algorithms. FFBPNN models have been proven to be very successful predictors in other areas of this thesis and it is likely they are among the best at predicting travel time. However, travel time is a significantly different variable from traffic flow and average speed. Consequently, it would be worthwhile to explore the performance of other models, such as SVMs, at predicting travel time using rainfall as an exogenous model input. It would also be interesting to see if the immense potential of the weather adaptive SWT-ACNN prediction methodology, in conjunction with the Delta FR construction technique, can produce a similarly successful result using motorway data.
REFERENCES


176


Met Eireann website: http://www.met.ie


180


APPENDIX

MSE
Mean Square Error (MSE) is the expected value of the square of the error or the difference between the target value and the model output. MSE is calculated using the following equation:

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i)^2
\]

where, \(y_t\) and \(x_t\) are the predicted and real values of a statistical process at time instant \(t\)

RMSE
Root Mean Square Error (RMSE) is the square root of MSE. RMSE represents the magnitude of the error.

\[
\text{RMSE} = \sqrt{\text{MSE}}
\]

MAPE
Mean Absolute Percentage Error (MAPE) is a relative measure of the absolute prediction or simulation error.

\[
\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - x_i}{x_i} \right| \%
\]