DMU's Interdisciplinary Research Group in Intelligent Transport Systems, (DIGITS)
Faculty of Computing, Engineering and Media

Estimation of Travel Time using Temporal and Spatial Relationships in Sparse Data

Author:
Luong Huy Vu

Supervisors:
Dr. Benjamin N. Passow
Dr. Daniel Paluszczyszyn
Prof. Yingjie Yang
Dr. Lipika Deka

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

November 2018

Abstract

Travel time is a basic measure upon which e.g. traveller information systems, traffic management systems, public transportation planning and other intelligent transport systems are developed. Collecting travel time information in a large and dynamic road network is essential to managing the transportation systems strategically and efficiently. This is a challenging and expensive task that requires costly travel time measurements. Estimation techniques are employed to utilise data collected for the major roads and traffic network structure to approximate travel times for minor links.

Although many methodologies have been proposed, they have not yet adequately solved many challenges associated with travel time, in particular, travel time estimation for all links in a large and dynamic urban traffic network. Typically focus is placed on major roads such as motorways and main city arteries but there is an increasing need to know accurate travel times for minor urban roads. Such information is crucial for tackling air quality problems, accommodate a growing number of cars and provide accurate information for routing, e.g. self-driving vehicles.

This study aims to address the aforementioned challenges by introducing a methodology able to estimate travel times in near-real-time by using historical sparse travel time data. To this end, an investigation of temporal and spatial dependencies between travel time of traffic links in the datasets is carefully conducted. Two novel methodologies are proposed, Neighbouring Link Inference method (NLIM) and Similar Model Searching method (SMS). The NLIM learns the temporal and spatial relationship between the travel time of adjacent links and uses the relation to estimate travel time of the targeted link. For this purpose, several machine learning techniques including support vector machine regression, neural network and multi-linear regression are employed. Meanwhile, SMS looks for similar NLIM models from which to utilise data in order to improve the performance of a selected NLIM model. NLIM and SMS incorporates an additional novel application for travel time outlier detection and removal. By adapting a multivariate Gaussian mixture model, an improvement in travel time estimation is achieved.

Both introduced methods are evaluated on four distinct datasets and compared against benchmark techniques adopted from literature. They efficiently perform the task of travel time estimation in near-real-time of a target link using models learnt from adjacent traffic links. The training data from similar NLIM models provide more information for NLIM to learn the temporal and spatial relationship between the travel time of links to support the high variability of urban travel time and high data sparsity.

Acknowledgements

I would firstly like to thank Dr Benjamin N. Passow and Dr Daniel Paluszczyszyn for their non-stop support in every part of my PhD journey alongside the rest of my supervisory team, Prof. Yingjie Yang, Dr Lipika Deka and Prof. Eric Goodyer who assisted in supporting my efforts.

I would also like to thank members within the De Montfort University Interdisciplinary research Group in Intelligent Transport Systems (DIGITS) who offered assistance to my work, both technical and inspirational.

I would like to thank my family, and especially for my parents, who always support and encourage me. The greatest thanks, however, goes to my wife Phuong Nguyen, without her love and sharing every moment in this journey, I would not have been able to finish this research.

I gratefully acknowledge the Ministry of Education and Training of Vietnam funding me with the three-year scholarship for my study.

Contents

List of Figures

List of Tables

Abbreviations

NLIM	Neighbouring Link Inference Method
SMS	Similar Model Searching
GMM	Gaussian Mixture Model
ANN	Artificial Neural Network
FF-ANN	Feed-forward Artificial Neural Network
FF-ANN-EL	Feed-forward Evolution Learning Neural Network
FF-ANN-RPROP	Feed-forward Resilient Back-propagation Neural Network
SVM	Support Vector Machine
SVM-NLK	Support Vector Machine with Nonlinear Kernel
SVM-LK	Support Vector Machine with Linear Kernel
MLR	Multivariate Linear Regression
DR-M-GMM	Detection and Removal outliers using Multivariate GMM
MSE	Mean Square Error
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
RPROP	Resilient Back-propagation learning algorithm
EL	Evolution learning algorithm
BPR	US Bureau of Public Roads
FCD	Floating Car Data
MA	Moving Average
HA	Historical Average
NLIM-EL	NLIM with FF-EN-ANN
NLIM-RPROP	NLIM with FF-RPROP-ANN
NLIM-MLR	NLIM with MLR
NLIM-SVR-LK	NLIM with SVR-LK
NLIM-SVR-NLK	NLIM with SVR-NLK
NLIM-EL-OD	NLIM with FF-EN-ANN, DR-M-GMM
NLIM-RPROP-OD	NLIM with FF-RPROP-ANN, DR-M-GMM
NLIM-MLR-OD	NLIM with MLR, DR-M-GMM

Symbols

T^{in}	The input matrix
T^{out}	The output matrix
L_O	The target link
L_N	The neighbouring links of a target link
L_{NF}	The front links of a target link
L_{NR}	The rear links of a target link
$L_N^{targetlink}$	The neighbouring links of a specific "target link"
L_M	The set of neighbouring links in a specific traffic link model ($L_M \in L_N$)
S_f	The dataset for a traffic link model including blank data
S_f^{in}	The input dataset for training a traffic model including blank data
S_f^{out})	The output dataset for training a traffic model including blank data
R	The data sparsity
T_f	The dataset for a traffic link model
T_f^{in}	The input features for training a traffic model
T_f^{out}	The output features for training a traffic model
$\mathfrak{C}_{NLIM}$	The collection of NLIM models
$\mathfrak{C}_E$	The list of $\mathfrak{C}_{NLIM}$'s corresponding errors
$\mathfrak{C}_{PS}$	The collection of similar potential models
$\mathfrak{C}_{PE}$	The collection of $\mathfrak{C}_{PS}$'s corresponding errors
$\mathfrak{C}_{link}$	The collection of traffic links
$\mathfrak{C}_{model}$	The collection of traffic models
ϵ	The threshold parameter for outlier detection algorithm
Θ	The set of hyper-parameters
θ	The hyper-parameter
ξ	The number of traffic models in a link layout
$\gamma_{threshold}$	The minimum number of labelled data

I dedicate this thesis to my beloved Phuong, who is my spouse, lover, partner and best friend.

Chapter 1

Introduction

Travel time refers to a period of time spent for the movement of people or objects between locations. The travel time parameter is an important metric in analysing and understanding a traffic network. Define travel time estimation as the method of which calculates the travel time of vehicles on a given link during a given period. Global Navigation Satellite System (GNSS), loop detectors, camera surveillance systems and other existing technologies can provide the near real-time measurements of travel time.

The existing travel time estimation methods are regularly classified into two tradition classes: the direct methodologies and indirect methodologies, Lu et al. (2018). In the direct method, travel time data is measured based on sampling data that is obtained from moving observers, i.e. in-vehicle sensor, GNSS, automated vehicle identification (AVI) system, telecommunication activities (Figure 1.1). Travel time data from smart-phone, private navigation devices and intelligent transportation systems are expanding rapidly. The indirect methods use continuous data that is obtained from stationary observers, i.e. inductive loop detectors to utilise the correlation between travel time and traffic flow dynamic. The inductive loop detectors are stationed at junctions and segments of a major road. The indirect method can provide travel time data at a regular sampling rate.

Over the past ten years, interest in travel time estimation has been increasing due to the crucial roles of travel time in intelligent transport systems. The industry 4.0 revolution makes the purposes of travel time estimation even more critical, Lu et al. (2018). Different multivariate and univariate methodologies to model travel time are

(a) Loop detector. (b) GNSS receiver. (c) AVI system.

FIGURE 1.1: Loop detector, GNSS receiver and AVI system

therefore proposed. Most of the proposed methods use statistical and mathematical techniques. The remaining often utilise the artificial neural networks, support vector machines, linear regression, Bayesian methodologies, Monte Carlo Algorithms, queueing and non-linear least square.

1.1 Thesis summary

This thesis aims to address the aforementioned challenges by introducing a methodology able to estimate travel times in near real-time by using historical sparse travel time data. Two novel methods, Neighbouring Link Inference method (NLIM) and Similar Model Searching method (SMS), are presented. The NLIM learns the temporal and spatial relationship between the travel time of adjacent links and uses the relation to estimate travel time of the targeted link. For this purpose, several machine learning techniques including support vector machine regression, neural network and multi-linear regression are employed. Meanwhile, SMS looks for similar NLIM models from which to utilise data in order to improve the performance of a selected NLIM model. NLIM and SMS incorporates an additional novel application for travel time outlier detection and removal. By adapting a multivariate Gaussian mixture model, an improvement in travel time estimation is achieved. The NLIM have been previously presented in a number of papers, (Vu et al. (2016, 2017)).

The following section gives a further discussion of the motivation for the proposed methods.

1.2 Motivation

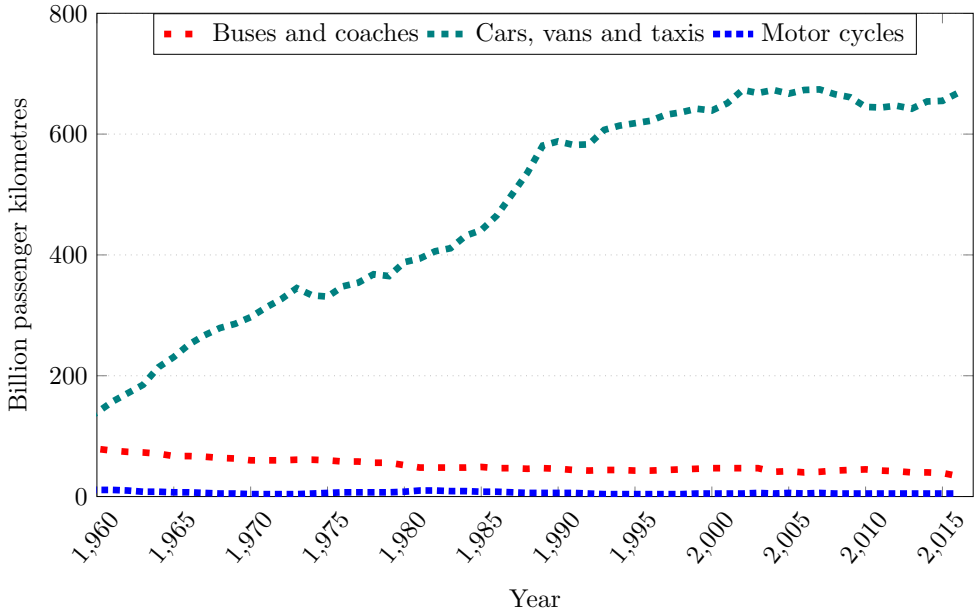
Traffic refers to all the vehicles that are moving along the roads in a particular area. According to Cookson and Pishue (2017), the worst country in Europe, regarding traffic congestion, is the United Kingdom, and the most congested city in Europe is also a city in the UK, London. More than £30 billion in 2016 is an estimated congestion cost for UK driver alone. One important reason for congestion is when the traffic demand exceeds the roadway capacity. While much work was undertaken to increase the UK's transport network capacity, in urban areas, transportation infrastructure development is constrained by land and financial resources, Petrovska and Stevanovic (2015).

According to the Transport Statistics Great Britain 2017, as can be seen in Figure 1.2, the number of cars, vans and taxis massively increases from 58 billion passenger kilometres to 668 billion passenger kilometres between the years 1960 and 2016. The number of buses and coaches and motorcycles remains similar. However, the road length for the major roads has not increased. Meanwhile, the road length for motorways slightly declined. The total length of minor roads seems not to grow after the 1990s.

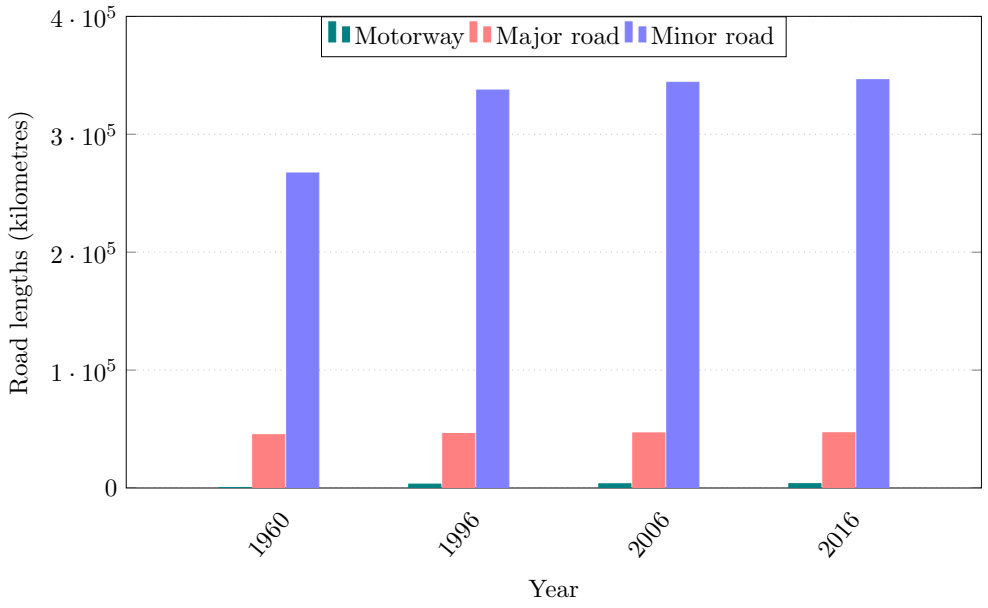
Another approach to deal with congestion is by improving the current traffic management strategies, Capes and Hewitt (2005). However, to effectively respond to daily traffic challenges operators need travel time data and accurate models of travel time.

Travel delays due to traffic congestion cause drivers' stress and increases such as unsafe traffic situations. They also increase adverse environmental and societal side effects, Hinsbergen et al. (2011). Congestion can be defined as the traffic demand exceeding the roadway capacity.

Travel time data on motorways regularly show relatively low variability (the variabilities are less than 3.5 seconds/km), especially in congested conditions. Because in congested conditions, speed limit reduces the speed difference between vehicles which results in higher and safer traffic flow, therefore lower travel time variability. They mainly depend on geometrical characteristics of motorways, such as the number of ramps weaving sections per unit road length (ramps refer to interchanges which permit traffic on a motorway to pass through the junction without interruption from any other traffic stream (Figure 1.3)), the number of lanes etc., Tu et al. (2006).



(a) Passenger kilometres by mode



(b) Road length by road type

FIGURE 1.2: Passenger kilometres by mode vs road length by road type, Great Britain: 1960 to 2016, Department of Transport (2016).

In contrast, urban travel times can be subject to very high variability because of traffic light signal cycles and queue delays. Pedestrians and cyclists and on-street parking also affect travel time, Hinsbergen et al. (2011), Ma and Koutsopoulos (2008). Hence, it is a challenge to design models or algorithms that can estimate accurately near real-time travel time in urban areas.

To deal with the growing problems that come with urbanisation and growing cities,

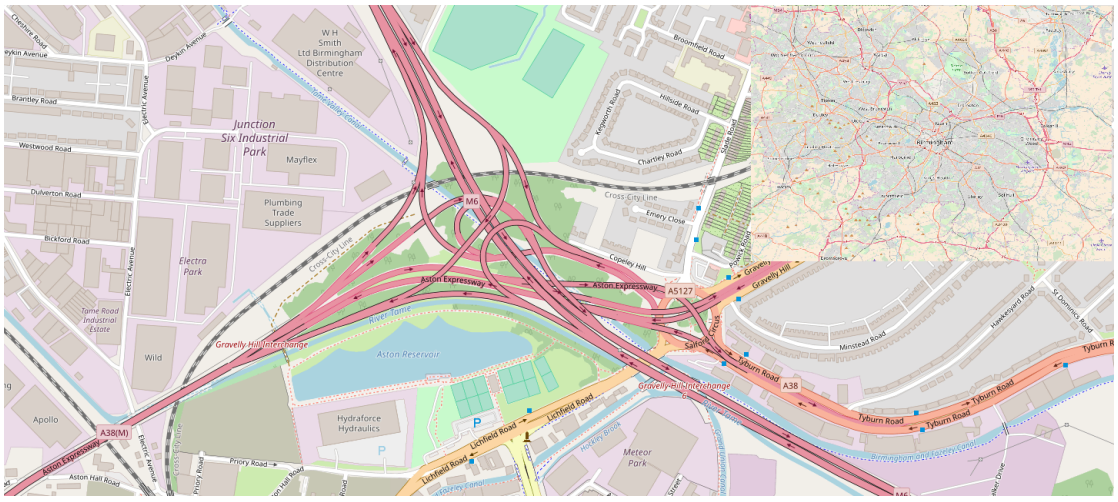


FIGURE 1.3: Spaghetti Junction in Birmingham, OpenStreetMap contributors (2017).

advance dynamic traffic management system is needed to manage existing transportation systems efficiently. Such systems require highly efficient and dynamic models. The models can provide crucial information for traffic optimisation such as signal control settings and to help commuters avoid traffic congestion. A valuable and objective type of traffic information is the travel time, Abu-Lebdeh and Singh (2011), Hinsbergen et al. (2011).

To address some of the aforementioned challenges a novel methodology is introduced in this thesis, namely the Neighbouring Link Inference Method (NLIM), to deal in particular with the highly sparse data which is collected from moving observers. Due to the high sparsity of travel time data observed in this study, the number of labelled data for the learning process of NLIM is limited. Another novel method, namely similar model searching (SMS) is proposed to enhance the amount of labelled travel time data for NLIM. A further improvement to the NLIM performance is achieved with the introduction of a novel application for travel time outlier detection/removal method which relies on a multivariate Gaussian mixture model.

In general, a temporal terminology refers to comparisons made within a defined time frame. If a process is temporally extended, it means that it happens over a period of time. If two events differ temporally, they occur at different points in time. Meanwhile, spatial terminology refers to comparisons or references within three dimension space. In this thesis, the term "temporal" relates to the time label associated with every datum. More specifically, travel time datasets used in this thesis contain a collection date and

a collection time interval. The spatial terminology is defined as traffic links where the travel time data are collected.

NLIM employs machine learning techniques to discover the temporal and spatial relationship between the travel time data of the target link and travel time data of its neighbouring links. Following the training process, travel times of a target link can be estimated from travel times observed on its adjacent links using the introduced NLIM models. Meanwhile, a similar model searching (SMS) method is proposed to enhance the amount of labelled travel time data for NLIM.

1.3 Hypotheses

In this research, three distinct hypothesis are set:

Hypothesis 1: *Relationships between temporal and spatial properties of travel times in neighbouring traffic links can be learnt to enhance the estimate of travel time of a target link.*

Four machine learning techniques are used to learn the relationships between temporal and spatial dependencies of travel times in traffic links from high data sparsity. They are the feed-forward resilient back-propagation artificial neural network (FF-RPROP-ANN), feed-forward evolution learning artificial neural network (FF-EL-ANN), support vector machine regression (SVR) and multivariate linear regression (MLR). Experiments are conducted on four distinct datasets. The details of the novel methodology are described in Chapter 4, and the obtained results are presented in Chapter 5. The outcomes from different case studies demonstrate that the proposed method can model the temporal and spatial relationships between traffic links. Such models can be subsequently used to estimate travel times for traffic links in transportation networks accurately. Datasets used in the experiments were acquired, gathered in different data sources including an artificial travel time dataset, a simulation travel time dataset and two real travel time datasets. Characteristics of the datasets are presented in Chapter 4.

Hypothesis 2: *Relationships between temporal and spatial properties of travel times in a traffic link model can be similar with those in other traffic link models in the same traffic network.*

A novel methodology is introduced that can look for similar traffic link models. A model is similar to another model if they satisfy two conditions: The number of neighbouring links in the two models is equal, and the relationship between neighbouring links and the targeted link in individual models is similar. The experiments were conducted in Chapter 4, and the results were presented in Chapter 5 to confirm the hypothesis.

Hypothesis 3: *Use of labelled data from similar traffic models for a selected traffic model can improve the performance of the traffic model regarding travel time estimation.*

Labelled data from similar models were utilised in a number of experiments to improve the performance of a selected traffic link model in Chapter 4. Results in Chapter 5 confirm that the use of travel data from similar traffic models can improve significantly the overall models' performance regarding travel time estimation, especially when the target link is a minor link.

1.4 Aims and objectives

This study is within the fields of Intelligence Transportation Systems, Computer Science, and Computational Intelligence and on the outer boundaries to Big Data. There are five main aims of this investigation:

- To provide an outline of the gaps of existing literature and research in urban travel time estimation for an extensive traffic network;

- To develop a traffic model to estimate travel time based on a historical sparse traffic data;

- To extend the knowledge of temporal and spatial properties in traffic links for gathered data based on the new model;

- To develop a methodology to consolidate the machine learning technique performance in learning of the temporal and spatial properties in traffic links using the data of similar traffic models;

- To analyse, compare and conclude on the performance of the models on unseen data.

1.5 Contributions

1.5.1 Major contributions

The major contributions of the thesis are summarised below:

1. A novel methodology to estimate travel times in complex and dynamic transportation networks is presented. The methodology, namely Neighbouring Link Inference Method (NLIM), employs machine learning techniques to learn temporal and spatial dependencies between traffic links resulting in a model of a transportation network. The developed model can be used to estimate travel times for traffic links. One of the advantages of this method is its capability to perform well on datasets with high sparsity and irregularity. The datasets or data feeds often have entries only for major links or entries collected at highly irregular intervals. Having embedded knowledge about the temporal and spatial dependencies between travel times of a target link and its adjacent links the model can overcome sparsity in input data and provide accurate estimations. Details are given in Chapter 4.

2. A novel methodology, namely similar model searching (SMS) has been introduced. The proposed method can enhance the learning performance of machine learning technique of temporal and spatial dependencies of travel times on traffic links' datasets with high sparsity and irregularity. SMS greatly improves the estimation capabilities of the final models. The main idea of SMS is to discover a list of traffic link models which are similar to the target traffic link model. After that, the labelled data of similarity models together with the target model training dataset is utilised as the new labelled dataset for training the target model. Details are given in Chapter 4.

3. A novel application of outliers detection and removal using multivariate Gaussian mixture models is presented. An outlier is an observation point that is distant from other observations. The outliers influence statistical characteristics, and they may lead to erroneous conclusions. To remove outliers in a matrix, the m-GMM is used to cluster the rows of a matrix into k row distributions where each element in a row is a variable of the multivariate. Structure and size of the rows distributions

(clusters of rows) are indicators to detect travel time outliers. Details are given in Chapter 4.

Part of this research was published in Vu et al. (2016, 2017). Details are presented in Appendix A.

1.5.2 Subsidiary contributions

The subsidiary contributions of the thesis are summarised as follows:

1. A comprehensive literature review which provides context and motivation for this research. There are six main topics that have been disused and analysed. The investigation is stressed on modelling travel time from sparse data with low sampling rates using machine learning techniques in extensive urban traffic networks. A comprehensive evaluation of the strengths and weaknesses of the existing travel time estimation methodologies is given. Related literature has been also reviewed to identify the gaps in previous research and to set a background of the study. Details are given in Chapter 2 and Chapter 3.

2. An insight into sparse and noisy traffic data. Many experiments and data analyses have been conducted to give an insight into sparse and noisy data. It provides critical information in order to select suitable techniques for travel time models and to select an appropriate type of intelligent transport system application to which the proposed methodologies intend to be integrated. Details are given in Chapter 4 and Chapter 5.

3. The application and evaluation of the developed methods on different datasets has been presented. It uses temporal and spatial dependencies of traffic links and their travel times to approximate travel time data which are currently not available. For this study, the methods were implemented and subsequently evaluated in four distinct case studies. Chapters 4 and 5 and Appendix B give a partial insight to some of the implementation issues and recommendation for future applications to other case studies.

1.6 Structure of the thesis

The structure of the thesis is as follows:

Chapter 2 contains a comprehensive literature review. It focuses on six major topics: travel time models and their roles, travel time data source, travel time characteristics, challenges for modelling travel time, travel time outlier detection and removal and appropriate model selection. Although the existing literature presents these topics in a variety of context, this section will primarily focus on the modelling travel time in extensive urban traffic networks where travel time typically exhibits non-stationary time series, volatility and non-linearity. Mainly, the review will focus on modelling travel time based on sparse and irregular dataset using machine learning techniques. Related literature has also been reviewed to outline the gaps in previous research and to set a background of the study.

Chapter 3 contains the theoretical framework and literature review that provides essential background information for a better understanding of the subsequent Chapters. A discussion will be given with reference to the fundamental elements that underpin the methods and introduce the application are used. It presents the background of multivariate linear regression, neural network and support vector machine techniques, and delivers details of components in each machine learning techniques. This chapter also offers an understanding of the hyper-parameters of each machine learning technique. It discusses the performance criteria used in this thesis and gives details of the process of selecting appropriate hyper-parameters for the support vector machines and artificial neural networks. A background on over-fitting and under-fitting while training machine learning based models, as well as clustering algorithms, are provided. Finally, some methodologies for proper selection of the number of clusters for clustering problems are reviewed.

Chapter 4 details the theoretical framework of the studied methodologies and the implementation for NLIM and SMS. It focuses on an investigation of the correlations between parameters on neighbouring traffic links. A novel Neighbouring Link Inference Method (NLIM), a methodology to model the temporal and spatial dependencies between travel times of a target link and its adjacent links is proposed. Besides, this chapter introduces another novel method, Similar Model Searching (SMS) as well as a

novel outliers detection/removal application based multivariate Gaussian mixture model. The SMS is a methodology that looks for NLIM similar models to deal with the high sparse and irregular data in traffic links in a traffic network. Datasets and their structures are also introduced and discussed in this chapter.

Chapter 5 evaluates the performance of NLIM and SMS methods. Where is feasible, the methods are compared against traditional statistics-based methods. For this purpose, unique case studies are used. Each case study thoroughly evaluated with the use of visual aids and performance criteria.

Chapter 6 contains conclusions, recommendations and future work. The major findings of the thesis are discussed with an overall summary of the contributions. The hypotheses are reconfirmed.

Chapter 2

Literature review

2.1 Introduction

Many methodologies have been proposed to estimate travel time data for the motorway, and arterial traffic network with some of them applicable to urban networks. Although the literature covers a wide variety of such methodologies, this review will focus on six major topics which emerge repeatedly throughout the literature reviewed. These topics include:

- Travel time models and their roles;
- Travel time data source;
- Travel time characteristics;
- Challenges for modelling travel time;
- Travel time outlier detection and removal;
- Appropriate model selection.

These topics are presented in existing literature in a wide range of contexts, this section, however, will primarily focus on the modelling travel time in extensive urban traffic networks where travel time typically exhibits non-stationary time series, volatility and non-linearity. The review will stress on modelling travel time based on from imperfect

datasets using machine learning techniques. Imperfect data refers to a dataset that has an irregular sampling rate and high data sparsity. Related literature is also reviewed to identify the gaps in previous research and to set a background of this study.

2.2 Transportation network

Traffic is defined as vehicles moving on roads. The transportation/traffic network refers to the primary way to accomplish the movement of people and goods. Junctions (interdependent points) and traffic link (lines of transportation) are the two main elements of the transport network, Meiying et al. (2015). The transportation network is responsible for the effective flow of people between different location, Cheng et al. (2013).

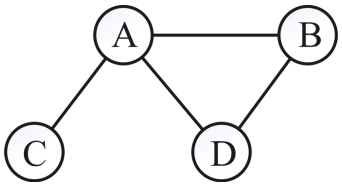


FIGURE 2.1: A graph represents a traffic network with nodes/junctions and traffic links/line of transportations.

Nodes are defined as intersections in the transportation network while links/traffic links are defined as connections between the adjacent nodes, Zhao and Spall (2016). Figure 2.1 illustrates an example of a transportation network using a graph. In particular, A, B, C and D are junctions (nodes), and AB, AD, BD and AC are traffic links (lines of transportation).

The transportation network scale refers to the number of nodes and its total length of links. A large scale transportation network relates to the traffic system consisting of thousands of traffic links. The traffic conditions in this network continuously change over time. The large scale transportation network is equivalent to a space where traffic congestion propagates temporally and spatially, Ma et al. (2015). In this thesis, traffic network is equivalent to transportation network.

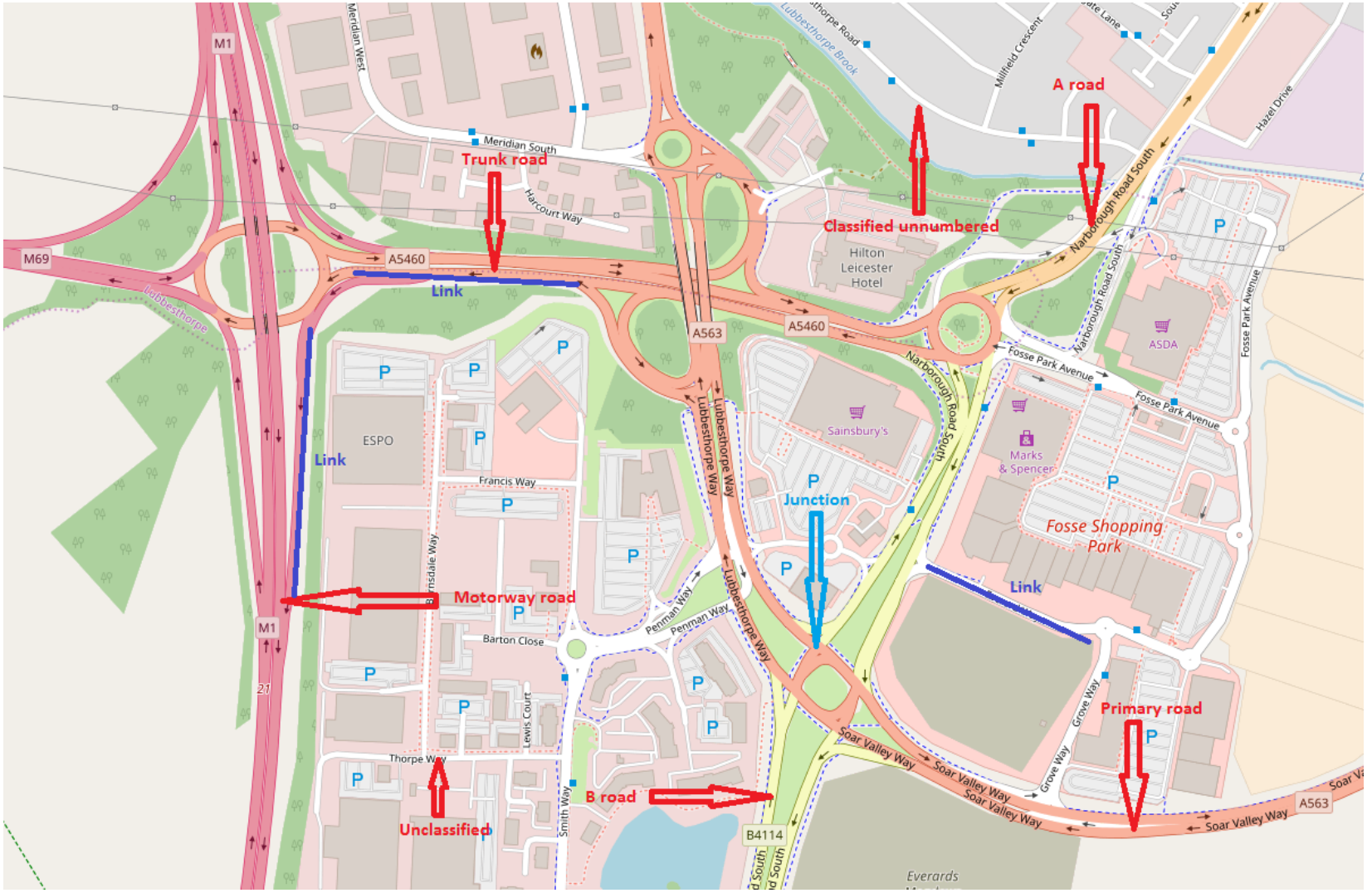


FIGURE 2.2: An example of a real traffic network and its elements, OpenStreetMap contributors (2017). The depicted is a section of Leicester city, UK.

Over the years national traffic networks grew in size and density in order to connect vital nodes, e.g. nowadays the complete network of England which consists of approximately 3.4 million separate links. The model of the Leicestershire traffic network is an illustration of a large traffic network. It consists of more than 236,000 traffic links. A total length of traffic links in the Leicestershire traffic network model is roughly 14,000 kilometres or 8,700 miles, Department of Transport (2012).

2.3 Travel time models and their roles

Models are by definition a compressed representation of the actual system, typically consisting of the most important aspects or components of the actual system, de Dios Ortuzar and G.Willumsen (2011). The quality of the models describes its capabilities to resemble the behaviour of a real system. A transportation network due to its size, complexity and dynamics is especially challenging to model. An accurate model of a transportation network would give insight to the network behaviour and lead to an improved decision making and planning transportation-related scenarios, strategies and policies.

In traffic control strategies and traffic management design, real-time travel time estimation can help massively to have appropriate responses to consistent changes in the transport network and its participants. Such systems can be used to reduce the level of congestion in peak hours. As a result, transportation practitioners are very interested in a travel time model which can estimate accurately and timely travel time, Lu et al. (2018).

Accurate travel time information is useful, e.g. commuters to make efficient travel decisions such as route choice, mode of transport and time of travel. It benefits a traffic policy sector in forecasting travel demand. It also helps evaluate the impact of policy instruments, e.g. congestion charges, Jenelius and Koutsopoulos (2013), Tang et al. (2018).

Accurate travel estimation plays a crucial role to improve the efficiency of the urban road network operation. However, travel time estimation models of an urban traffic network is a challenging subject in the intelligent transportation system as the delay from traffic signal controls, congestion effects, stochastic incidents, etc. has made urban traffic travel

time regularly uncertain, Meng et al. (2017). Travel time models can provide travel time between locations which is an essential factor of vehicle routing problem, Fleischmann et al. (2004), Kim (2017).

2.4 Traffic link classification

Different road categories produce different traffic travel time. According to the Department of Transport of United Kingdom, five groups of the UK roads are defined. They are Motorways, Trunk roads, Primary roads, A roads, B roads, classified unnumbered and unclassified, Department of Transport (2012). In other studies, roads are classified into three types: Motorway, Arterial (corridor level) and Urban arterial, Vlahogianni et al. (2014). In this thesis, the road category follows the road classification of Department of Transport of United Kingdom. The road categories in Department of Transport (2012) are defined in detail in Table 2.1. Additional classification by a link type is to determine whether road is a major or minor link.

Table 2.1: UK road categories, Department of Transport (2016)

Category	Definition	Type
Motorway	It is classified as special road where certain types of traffic is prohibited. This arrangement is determined by statute.	Major
Trunk road	It is nationally important road which is used for the distribution of goods and services and a network for the travelling public.	Major
Primary road	It provides most satisfactory transport for a regional or county level. It is mainly feeding into the Trunk roads for longer journeys.	Major
A road	It is a large-scale transport link which provides transport within or between areas.	Major
B road	It connects different areas. It is usually feeding traffic into A roads and smaller roads on the network.	Minor
Minor road (Classified unnumbered and Unclassified)	It is the smallest road that connects unclassified roads with A and B roads. It is regularly connecting a housing estate to the rest of the network. It is for local traffic.	Minor

The classified unnumbered and the unclassified categories cover 70% of links in the UK, Department of Transport (2012). In this research, major link category refers to a

combination of the motorway, trunk, primary and A link. Meanwhile, the minor link category applies to the rest of the road categories. An example of traffic link types in practice is illustrated in Figure 2.2 in the previous section.

2.5 Travel time data sources

Travel time is a traffic parameter. Real travel time can be measured and collected typically by using stationary or moving observers, Ma and Koutsopoulos (2008). Stationary observers include loop detectors and video surveillance, and moving observers that involve floating cars or probe cars (Figure 1.1). Travel time data source determines the characteristics of the resulting dataset.

According to Wright (1973), a Floating car was a concept used to obtain traffic flow and journey time. Since the 2000s, a Floating car is any car from which GNSS positions are continually recorded via in-car equipment, smart-phones, etc., de Fabritiis et al. (2008), Derrmann et al. (2016), Jones et al. (2013), Leodolter et al. (2015), Pan et al. (2011), Protschky, Feit and Linnhoff-Popien (2015), Protschky, Ruhhammer and Feit (2015), Rahmani et al. (2013, 2014), Wang et al. (2012). Floating Car Data (FCD) used in this research refers to travel time data which is gathered from GNSS tracking of floating cars by TrafficMaster Vu et al. (2016).

The stationary observers can collect real travel time data at regular and frequent intervals, Jones et al. (2013). However, it is possible that stationary traffic observers show more expensive, Ma and Koutsopoulos (2008), Wosyka and Pribyl (2012) than the moving traffic observers, and are therefore only available in some particular motorways or major roads. In contrast, the moving observers collect travel times at irregular and less frequent intervals. They use GNSS equipment to trace positions of actual cars across an entire traffic network. They can cover almost links existing in a traffic network despite the link categories, Shawn M. Turner and Holdener (1998).

Travel time data that is collected from moving observers are less frequent for links on minor roads, Jones et al. (2013). Thus, for many periods of time, in a particular traffic link, there may not be availability of any observers' travel times. This is a problem for any model that uses travel times as an input variable in this particular link.

Another limitation of moving observers' travel time data is sparsity. In Tang et al. (2018), a trajectory of taxi caps is used to approximate real travel times. Since the number of taxicabs which involved in traffic is limited, a link may not be covered by any trajectory. The travel time data from Floating car data (FCD) which is used by Department of Transport (2016) shares the same characteristics with the dataset used by Tang et al. (2018) but the dataset is more sparse in terms of data resolution/data sampling rate (i.e. 15 minutes intervals).

2.6 Travel time characteristics

Travel time data on motorways regularly shows relatively low variability (the travel times are less than 3.5 seconds/km), especially in congested conditions. Because in congested conditions, speed limit reduces the speed difference between vehicles which results in higher and safer traffic flow, therefore lower travel time variability. This is mainly because of geometrical characteristics of motorways, such as the number of ramps weaving sections per unit road length (ramps refer to interchanges which permit traffic on a motorway to pass through the junction without interruption from any other traffic stream), the number of lanes etc., Tu et al. (2006).

In contrast, urban travel times can be subject to very high variability because of traffic light signal cycles and queue delays. Also, pedestrians and cyclists and on-road parking often affect travel time, Hinsbergen et al. (2011), Ma and Koutsopoulos (2008). Hence, to design models or algorithms that can estimate accurately near real-time travel time in urban is a challenge.

2.7 Travel time estimation

Travel time, average speed (the total distance travelled by a vehicle divided by the elapsed time to cover that distance), congestion level (slower speeds, longer trip times, and increased vehicular queueing, etc.), traffic flow (flow of vehicles on a lane) and traffic delay (time difference between actual travel time and free-flow travel time) of a traffic segment/link are intercorrelated. A vital performance indicator of the traffic network is the travel time parameter. Travel time estimation is defined as the method which

approximates the travel time of vehicles on a given link during a given period. Data from GNSS equipment, loop detectors, camera surveillance systems and other existing technologies can be used to approximate travel times in near real-time.

The existing travel time estimation methods are regularly classified into two tradition strands: the direct methodologies and indirect methodologies Lu et al. (2018). In the direct method, travel time is estimated based on data samples that are obtained from moving observers i.e. in-car sensor equipment Ernst et al. (2014), Guo et al. (2015), Yeon and Ko (2007), GNSS-based floating car de Fabritiis et al. (2008), Department of Transport (2016), Hadachi et al. (2013), Jones et al. (2013), Lee et al. (2017), Maiti et al. (2014), Rahmani et al. (2014), Su et al. (2010), Wang et al. (2012), automated vehicle identification (AVI) system Ma and Koutsopoulos (2008), Rahmani et al. (2014), telecommunication activities Chitraranjan et al. (2016, 2015), Derrmann et al. (2016), Vidović et al. (2017). Furthermore, travel times can be estimated from locations of smart-phone users, from car satellite navigations systems or large car fleets operators. Nowadays many modern cars, e.g. BMW, Tesla, Nissan collect users travels information and feedback to their respective R&D centres.

The advantage of the direct method is that it requires limited expenses of infrastructure and it is capable of producing travel time data in small roads where loop detectors may not be deployed. The drawback of the direct method is that for example a car cannot collect data in different locations simultaneously. Also at different times the particular road may exhibit different dynamics which may not be captured by a car. Hence, uncovering a methodology for travel time estimation from incomplete datasets receives a great interest from researchers in the field of the intelligent transport systems.

A methodology to estimate the travel time from GNSS vehicle location reports was introduced by Department of Transport (2016). The GNSS signal from vehicles is mapped to real traffic links. Based on the time stamps of the GNSS vehicle location reports, travel time for full traffic link is approximately reconstructed. The interval of travel time is given in 15 minute intervals. The methodology is widely used in the UK for transport management and control, Department of Transport (2016), Vu et al. (2017).

On other hands, the indirect method uses data obtained by stationary observers, i.e. inductive loop detectors Dong and Mahmassani (2012), Huang and Barth (2008), Li

et al. (2013), Zhang and Mao (2015) to analyse the correlation between travel time and traffic flow. The inductive loop detectors are regularly deployed at junctions and segments of major roads. The indirect method can provide travel time data at a regular sampling rate.

For many years an interest in travel time estimation was growing due to its crucial role in intelligent transport systems. Nowadays, for the ongoing Industry 4.0 Revolution, which is expected to impact all disciplines, industries, and economies, the information about travel times of goods and people is even more critical, Lu et al. (2018). As a result different multivariate and univariate methodologies to model travel time are being proposed. Most of the proposed methods use statistical and mathematical techniques. The remaining often utilises artificial neural networks, support vector machines, linear regression, Bayesian methodologies, Monte Carlo Algorithms, queueing and non-linear least square. The present travel time estimation models and associated literature are presented in Table 2.2.

A number of earlier research employ statistical methodologies to estimate current travel time data. They include distributions of everyday historical travel time data in a traffic link/segment, Derrmann et al. (2016), Jenelius and Koutsopoulos (2013), Kim (2017), Rahmani et al. (2013), Wan and Vahidi (2014), distributions of historical travel time on a complete route Chitraranjan et al. (2016), Rahmani et al. (2014), travel time histogram Lee et al. (2017), Waury et al. (2017, 2018) and average travel time in link Ahn et al. (2014), Guo et al. (2015), Yi et al. (2015).

Mathematical methods for travel time model have recently received interests of researchers. They include a travel time allocation method Meng et al. (2017), tensor-based method Tang et al. (2018), maximum likelihood Zhao and Spall (2016), indexing trajectories Tomaras et al. (2015), local alignment Chitraranjan et al. (2015). Mathematical and statistical methodologies usually perform less accurate in urban traffic network where the traffic condition can be complex.

A number of research on travel time estimation focuses on machine learning techniques such as neural network Lu et al. (2018), support vector machine Leodolter et al. (2015), non-linear least square Zhan et al. (2013), linear regression Leodolter et al. (2015). And lately, Monte Carlo algorithm Hadachi et al. (2012, 2013) and queuing methodology Li et al. (2013) are not considered on recent research.

TABLE 2.2: Existing travel time estimation methodologies and relevant literature

Model	Relevant literature
Neural network	Lu et al. (2018)
Statistical	Ahn et al. (2014), Chitraranjan et al. (2016), Derrmann et al. (2016), Guo et al. (2015), Jenelius and Koutsopoulos (2013), Kim (2017), Lee et al. (2017), Pirc et al. (2016, 2015), Rahmani et al. (2013, 2014), Wan and Vahidi (2014), Waury et al. (2017, 2018), Yi et al. (2015)
Mathematical	Chitraranjan et al. (2015), Díaz et al. (2016), Meng et al. (2017), Tang et al. (2018), Tomaras et al. (2015), Zhao and Spall (2016)
Bayesian network	Deng et al. (2013), Derbel and Boujelbene (2015)
Linear regression	Leodolter et al. (2015)
Support vector machine	Narayanan et al. (2015)
Monte Carlo	Hadachi et al. (2012, 2013)
Queueing	Li et al. (2013)
Non-linear least square	Zhan et al. (2013)

Machine learning methodologies are regularly data-driven methods. They can learn relationships and create models using unstructured dataset. The approaches are often useful in many transportation applications because they are free of model assumptions and the uncertainty of traffic can be involved in the traffic model.

Recent developments in technology in the Industrial 4.0 Revolution and the non-stop introduction of new technology and powerful computers, big data analytic techniques and mathematical models provide researchers with a phenomenal opportunity to expand the knowledge in travel time estimation domain.

The application of machine learning techniques in traffic models and the development of new data acquisition instrumentation allow researchers to capture or model more precisely dynamics of a large traffic network. In this thesis, machine learning techniques are utilised to develop travel time models for a large size traffic network.

TABLE 2.3: Challenges in modelling for travel time estimation and relevant literature

Challenges	Relevant literature
Motorway link travel time estimation	Díaz et al. (2016), Dong and Mahmassani (2012), Fei et al. (2011), Huang and Barth (2008), Li and Chen (2013), Li et al. (2013), Lu et al. (2018), Rice and van Zwet (2004), Tu et al. (2006), Wang et al. (2014, 2012), Yeon and Ko (2007), Yildirimoglu and Geroliminis (2013), Zou et al. (2014)
Arterial link travel time estimation	de Fabritiis et al. (2008), Derrmann et al. (2016), Guo et al. (2015), Hadachi et al. (2011, 2012, 2013), Hage et al. (2012), Hinsbergen et al. (2011), Jenelius and Koutsopoulos (2013), Kim (2017), Krishnamoorthy (2008), Tang et al. (2018), van Hinsbergen et al. (2011), Vidović et al. (2017), Wei et al. (2010), Zhan et al. (2013), Zhao and Spall (2016)
Minor link travel time estimation	Vu et al. (2017)
Travel time estimation in large scale traffic network	Guo et al. (2015), Lee et al. (2017), Tang et al. (2018), Vidović et al. (2017), Zhan et al. (2013)
Travel time estimation on sparse and irregular datasets	Jenelius and Koutsopoulos (2013), Lu et al. (2018), Maiti et al. (2014), Meng et al. (2017), Passow et al. (2013), Pirc et al. (2015), Rahmani et al. (2013), Tang et al. (2018), Wan and Vahidi (2014)
Travel time estimation on temporal and spatial dependencies	Jones et al. (2013), Li et al. (2013), Tang et al. (2018), Waury et al. (2018)
Travel time outliers detection/removal	Jang (2016), Lin et al. (2014), Passow et al. (2013), Vu et al. (2017)

2.8 Challenges of travel time estimation

From the reviews of papers over the recent years, most research attention has gone into four challenging directions: (1)travel time estimation on the motorway, arterial, minor link and large-scale traffic network; (2) travel time estimation on sparse and irregular datasets; (3) travel time estimation on temporal and spatial dependencies; (4) travel time outliers detection/removal. These four challenges are summarised in Table 2.3.

2.8.1 Travel time estimation on motorway, arterial and minor link and large scale of a traffic network

It becomes clear that most research effort has gone into modelling travel time for motorway and major links (Table 2.3). There is a lack of research efforts on modelling the minor links. However, the minor link plays a crucial role in extensive traffic networks. They are a vast majority of links in the traffic network, Department of Transport (2012).

Minor links can essentially become links in an alternative route selection when traffic congestion appears on the major road in the traffic network. Therefore, not only travel time in major traffic links are essential, but also those of minor links. They are also important indicators for decision making. Not much research has been done to model travel times of all traffic links in large scale traffic networks likely due to challenges ahead, i.e. irregular sampling intervals, highly sparse and inconsistent data, complexity and scale of the problem.

2.8.2 Estimate travel time on sparse and irregular data

A number of studies explored approaches to calculate the travel time with sparse and irregular data. In Maiti et al. (2014), due to the inaccurate and missing data, a pre-processing data has been applied before the data are used in the model. A ANN-based filter was introduced in Passow et al. (2013). It identifies outliers by picking those readings that are higher than twice the maximum of the filter ANNs output. The ANN-based filter can be applied in our research to classify normal and abnormal average travel time in every link of the traffic network.

The authors proposed, using fuzzy, clustering techniques to interpret relations between particular travel time data to deal with complex data outlier generation mechanisms, Zheng and McDonald (2009). Their methodology can specify data thresholds to exclude outliers that help to use all available data. In Pirc et al. (2015) the vehicle travel time categories during traffic flow conditions are remaining unequal, a travel time estimation algorithm using robust statistics is introduced.

Statistic methods were used to an eliminated influence of slower heavy vehicles (HVs) to the overall results. In the study of Rahmani et al. (2014), a non-parametric route

travel time calculation is employed to estimate travel times based on a fusion of floating car data (FCD).

2.8.3 Temporal and spatial dependencies

Several studies have supported the existing of temporary and spatial dependencies in traffic, i.e. studies of Jones et al. (2013), Li et al. (2013), Tang et al. (2018). Integration temporal and spatial relationships of traffic information into traffic models are a valuable task in intelligent transport systems, Tang et al. (2018). This may be done by attempting to integrate relationships between travel time in links into travel time estimation models. Few of research attempt to utilise temporal and spatial relationships of traffic information into a traffic model.

An approach of applying temporal and spatial dependencies in travel time estimation has presented in the work of Li et al. (2013). The temporal-spatial queueing uses headway travel time series which are collected from upstream and downstream of a middle link, and recent vehicle speed to estimate the middle link's travel time data. The model utilises the relationship between upstream travel time and downstream travel time to enhance the accuracy of travel time estimations. The proposed method can model fast travel time variations. In another approach, traffic data of nearby links is used to forecast travel time of a selected road segment. The method was termed as geospatial inference in a study of Jones et al. (2013). Both studies used travel time data series which naturally have the temporal relationship. Still, travel time data series costly gather on extensive traffic networks.

Tang et al. (2018) have proposed a purely data-driven approach called Tensor-based citywide spatial-temporal travel time modelling. The proposed method utilises the spatial-temporal approach in modelling the travel time of all traffic links under different traffic condition and time slots. The methodology is complicated because of characteristics of tensor-based techniques as well as the correlation between travel times and the influential factors on the complexity of urban traffic networks.

The travel times on different traffic links in specific time slots are transformed into a 3-order tensor. There are two 3-order tensors. One is for recent travel time, and the other is for historical travel time data. The 3-order tensors are very sparse due to

the characteristics of travel time data approximated from trajectories of taxes. After the transformation of the data, a probabilistic traffic condition clustering is applied to discover the centroid of travel times into various categories. Travel time data of different drivers are separately processed. The centroid of the cluster is used to replace missing travel time, and the proportion of observations regarding the category are the probability of the corresponding traffic conditions.

The idea of the proposed method in Tang et al. (2018) is that similar traffic condition in the traffic link should produce a similar travel time for a specific driver. The centroid of the cluster represents the travel time of the corresponding traffic condition and corresponding driver. Based on the current traffic condition, a corresponding cluster of historical travel times is selected. The missing travel time is replaced by which is the centroid of the cluster.

The advantage of the method is that the travel time can be easily modelled as a 3-order tensor despite the complexity of urban traffic network, and the technique can work with high data sparsity, but it still produces promising travel time estimation results. However, clusters' centroids are used to represent all the members of the clusters which would lead to the less accuracy of travel time estimations. Furthermore, the centroid does not seem to describe correctly the traffic condition as well as the impact of other factors on the individual travel time at a specific time slot in the uncertain and dynamic of the urban traffic network.

The method in the work of Tang et al. (2018) does not express the relationships between links in travel trajectories and those on traffic links of two different travel time trajectories. The travel time in the clusters is selected based on the time slot, corresponding driver and corresponding traffic link; thus, travel times seem to have temporal relationship only.

The travel time dataset in Tang et al. (2018) is collected from GNSS equipment in 29083 taxicabs on 84100 links. The dataset contains information of drivers and vehicles regrading travel time trajectories. The dataset shows high sparsity in terms of existing travel time on links in corresponding junctions despite the fact that the dataset has high sampling rates (e.g. 96 seconds per point of over $6.7*10^8$ GNSS points).

Although the methodology provides techniques for travel time estimation for a dataset with sparsity, it cannot be used as a benchmark to compare or evaluate the proposed methods in this thesis. The methodology is not applicable to this study as it required specific data structures and data constraints meanwhile the datasets in this thesis do not contain such requirements' properties. i.e. The datasets in this thesis only provide information about sparsity travel times in each link in sparse time intervals. They do not have the trajectory of vehicles as well as information about cars and drivers that are required in the method in Tang et al. (2018).

2.8.4 Travel time outliers detection/removal

The travel times are usually collected in real time. Nevertheless, the collected dataset might contain the number of high-value data points because frequently stopping and starting vehicles would report report much slower travel time than that prevails on the road. In statistics, an outlier is an observation point that is distant from other observations. The outliers influence statistical characteristics, and they may lead to erroneous conclusions. Therefore detecting outliers is necessary before utilising data to obtain a reasonable solution to a problem, Lin et al. (2014). Several approaches have been used to identify and remove outliers; these range from statistics, to ANNs and fuzzy algorithms, Jang (2016), Lin et al. (2014), Passow et al. (2013), Tang et al. (2018), Vu et al. (2017).

In the study of Tang et al. (2018), outliers are merely defined as the estimation results which are less than zero or the probability of occurrences of the travel time is greater than 1 or less than 0. In other words, the travel time entries which cannot be estimated are considered as outliers. The value of outliers detected is set to zero.

The study of Yang et al. (2013) shows that GMM can produce a high rate of accuracy for vehicle stop/non-stop movement classification. Therefore, GMM can be utilised to detect outlier in sparse travel time data. Therefore, in Vu et al. (2017), GMM was applied to filter outliers of the travel time data in each link in a link layout. The structure and size of a travel time cluster in a link are indicators to determine/detect outliers in the proposed algorithm. Threshold parameters (ϵ) were predefined to distinguish normal data from outliers for individual vehicle class. The filtering outlier algorithm was separately applied for data of the individual vehicle class because different vehicle classes might have distinct

characteristics and behaviours. Therefore, they might produce different travel time distributions. In Vu et al. (2017) ϵ is set to 0.1 for all vehicle classes. The results in Vu et al. (2017) demonstrate that GMM can detect travel time outliers of individual vehicle class in particular traffic link.

This research introduces a novel application for travel time outliers detection/removal vectors of parameters. The novel application is an extension of the method in Vu et al. (2017). The GMM application is extended to fit a vector of parameters that retrieve from a specific traffic link model in a traffic link layout. Section 3.6 will discuss the details of the algorithm.

2.9 Model selection

Travel time estimation for links in extensive urban traffic networks is still a challenging subject. It is an excellent field for developing and testing complicated travel time estimation methodologies because of the availability of data in the traffic networks. Dataset collected from an extensive traffic network often shows irregular and sparse, Guo et al. (2015), Lee et al. (2017), Tang et al. (2018), Vidović et al. (2017), Zhan et al. (2013).

Travel time data obtained from urban traffic network always have non-stationary time series, volatility and nonlinearity. The availability and characteristics of a dataset decide selection results of the traffic model method, Krishnamoorthy (2008). The consequence of the selection method is the most important process to develop a traffic model from the dataset for the accuracy of travel time estimations.

The characteristics of the dataset in this research show sparse and irregular, therefore, a regression method is chosen for this thesis. Regression models based on machine learning techniques including support vector machines, artificial neural networks and multivariate linear regressions are employed in this research to model the relationship between temporal and spatial properties of travel time in links in an extensive urban traffic network.

Support vector machines, neural networks and multivariate linear regressions are chosen because these techniques are typically data-driven. They can learn relationships and

create models from unstructured datasets. They are free of model assumptions, and the uncertainty can be involved in the model parameter for travel time estimations. Meanwhile, statistical and mathematical methodologies are not chosen because they usually perform inadequately in urban traffic networks with complex and dynamic traffic conditions. The theory of regression models and utilised approaches are discussed in the theoretical framework chapter (Chapter 3).

Chapter 3

Theoretical framework

3.1 Introduction

This chapter provides essential background information for a better understanding of the subsequent chapters. This chapter is organised as follows: Section 3.2, 3.3 and 3.4 present the multivariate linear regression, neural network and support vector machine techniques, respectively. Each section delivers a background of the particular machine learning technique along with discussion their respective hyper-parameters. Section 3.5 discusses the performance criteria used in this thesis. Section 3.6 elaborates a process of selection of appropriate hyper-parameters for support vector machine and neural network techniques. Section 3.7 follows with a background of over-fitting and under-fitting that might occur while training machine learning models. Finally, Section 3.8 gives fundamentals about clustering algorithms, and reviews methodologies for correct selection of the number of clusters.

3.2 Multi-linear regression

Multiple-linear regression is a simple machine learning technique where the correlation coefficient between independent variables and a dependent variable is learnt. A linear equation is matched to observed data in multi-linear regression. Furthermore, the value of the independent variable x is correlated with a value of the dependent variable y Jobson (1991).

Y is a matrix of response values for n observations:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ \dots \\ y_n \end{bmatrix} \tag{3.1}$$

X is a design matrix which contains all predictors (x_{ij}). Size of the design matrix is $n * p + 1$ where p is the number of dependent variables:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \tag{3.2}$$

β is a slope vector which packs intercepts and slopes ($p + 1$ dimension vector):

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \dots \\ \beta_p \end{bmatrix} \tag{3.3}$$

and e is an error vector:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_n \end{bmatrix} \tag{3.4}$$

The regression line for n dependent variables is shown in Equation 3.5:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_p x_{i,p} + e_i \tag{3.5}$$

The e_i assumedly has a normal distribution ($\mu = 0$ and constant variance σ^2). The i refers to the i^{th} regression in the population. The sample of regression line is presented as below:

$$\hat{y}_i = \hat{\beta} X_i + \hat{e}_i \tag{3.6}$$

Where:

$$\hat{\beta} = (X'X)^{-1} X'Y \tag{3.7}$$

$X'X$ and $(X'X)^{-1}$ are $(p+1)*(p+1)$ symmetric matrices and $X'Y$ is a $(p+1)$ dimension vector.

The multivariate linear regression technique is used in this research as an initial methodology and as a machine learning technique for the proposed method.

3.3 Artificial neural network

Human brains are extremely complicated machines that are capable of solving complex problems. A human brain can be described as an extensive neural network. It is a part of the central nervous system of a human. The brain is a network that composes of connections between around 10^{11} neurons. And the brain is a parallel computer which is very complicated and non-linear, Tettamanzi and Tomassini (2011).

So far, it is impossible to make an artificial brain however a simplified artificial neuron and artificial neural networks (ANNs) can be created. An ANN can be defined as a parallel computing system because it consists of a large number of simple processors and interconnections between them and the processors and their connections can operate in parallel, Suzuki (2011). The ANN is a well-known machine learning technique. It was invented to simplify modelling method in which the human brain performs a precise task or a function of interest, Haykin (2008), Huemer et al. (2010), Suzuki (2011).

Warren McCulloch and Walter Pitts started the first mathematical model for a neural network in 1943, McCulloch and Pitts (1943). This model was mainly used for binary classifications. The fundamental of this model is presented using basic mathematics formulas in Equation 3.8.

$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{m} w_i x_i \geq T_h. \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

where T_h is a predefined threshold, $x_1, x_2, .., x_n$ are variant input. $w_1, w_2, ..., w_n$ are corresponding weights and m is number of input variables.

Warrant McCulloch and Frank Rosenblatt introduced a constant parameter to the neuron model, and it called the bias, Rosenblatt (1958). Frank Rosenblatt invented a neural model which is called perceptron. This model is also used for binary classification. The perceptron application is often used for solving linear separable classification problems. The mathematical representation is shown in Equation 3.9.

$$f(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{m} w_i x_i - b \geq T_h. \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

where b is the bias constant.

An activation function is included in the neuron model to enable ANNs to solve varying output condition. It is shown in Figure 3.1.

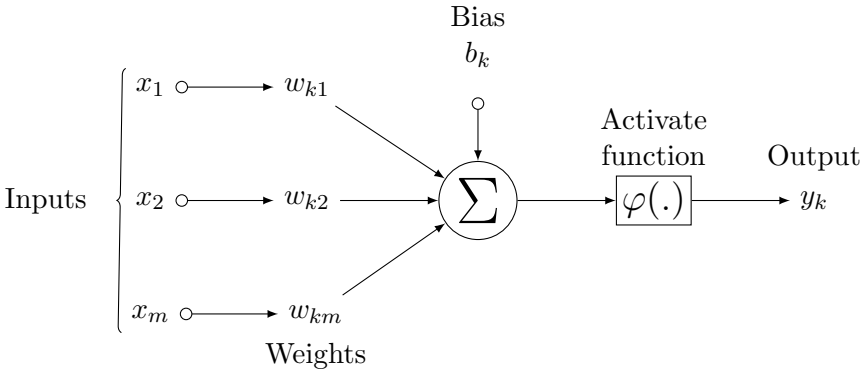


FIGURE 3.1: A neuron non-linear model of labelled k

The fundamental elements of a neural network:

1. A collection of interconnecting links that are identified by weight of their own.
2. A linear combiner aggregates the input signals. The corresponding neurons' synapse strengths are used to weight them.
3. An activation function is a function that limits the amplitude of neurons' outputs.

4. The bias component is a component that is used for increasing or reducing the activation functions' inputs.

The ANN could be defined by two equations (3.10 and 3.11):

$$u_i = \sum_{j=1}^{m} w_{ij} z_j \tag{3.10}$$

$$y_i = \varphi(u_i + b_i) \tag{3.11}$$

where u_i is the linear combiner output, $w_{i1}, w_{i2}, ..., w_{im}$ are the neuron i^{th}'s synaptic weights respectively, $z_1, z_2, ..., z_m$ are signals of the inputs of a neural network, b_i is bias i^{th}, activation function $\varphi(.)$ and output y_i.

Activation function

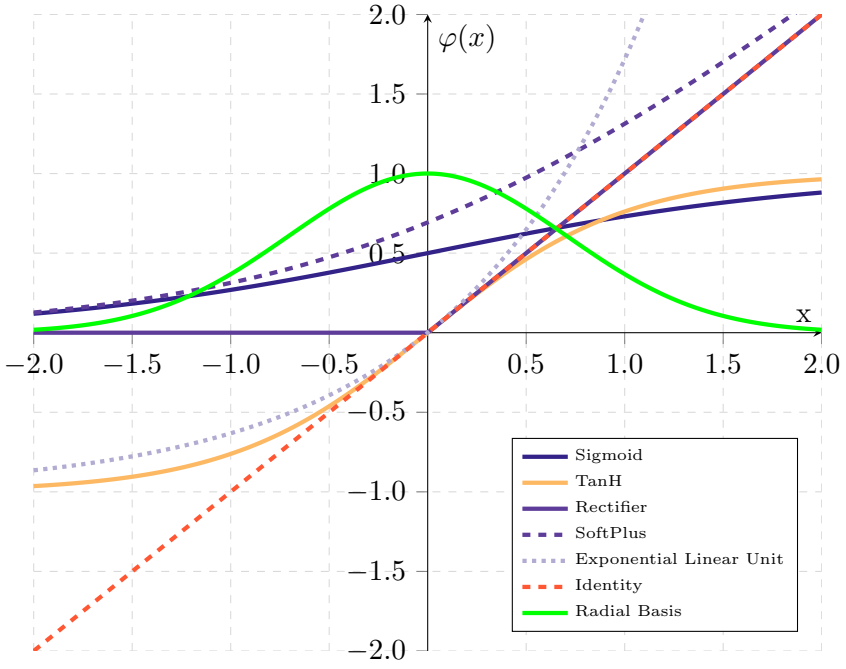


FIGURE 3.2: The essential activation function, donated by $\varphi(x)$, determines the output of a neuron where x is the input variable.

The activation function is a vital element in ANNs, however, choosing the best activation functions for neurons in a specific neural network is a problem that depends on labelled data patterns. They cannot be set to default or generalised for any data or neural network structure.

The primary purpose of most activation functions is to offer non-linearity so the network would be capable of learning more complex patterns. The classical activation functions are illustrated in Figure 3.2 and Equation 3.12. They are Sigmoid, TanH, Rectifier, SoftPlus, Exponential Linear Unit, Identity and Radial Basis function, respectively.

$$\varphi(x) = \begin{cases} \frac{1}{1+e^{-x}} & \text{Sigmoid} \\ \tanh(x) & \text{TanH} \\ \max(0, x) & \text{Rectifier} \\ \log(e^x + 1) & \text{SoftPlus} \\ \max(x, e^x - 1) & \text{Exponential Linear Unit} \\ x & \text{Identity} \\ e^{-x^2} & \text{Radial Basis} \end{cases} \tag{3.12}$$

These activation functions are often suitable for regression problems where inputs and outputs are continuous values. For the different data scale issues in this thesis, the input and output of neural networks are normalised using the MinMax algorithm where the input and output data are scaled between 0.0 and 1.0, Wu et al. (2017). Hence, activation functions use in output layer should limit the output value between 0.0 and 1.0 as well. Therefore, only three activation functions are considered for the output layer to eliminate out of bound of the output values. They are Sigmoid and Radial Basic.

Input layer and hidden layers can be suitable with any of the activation functions listed in Equation 3.12. However, to eliminate a loss stacking layers problems, the linear activation function will not be utilised for any experiments in this thesis.

Function of the hidden neurons

Hidden neurons perform a vital role in a performance of multilayer perceptrons. The hidden neurons work as feature detectors. During ANN learning progress, the hidden neurons perform non-linear conversions between input data and feature space. Salient features which characterise train data are gradually discovered by hidden neurons, Haykin (2008).

Network architectures

According to Haykin (2008), there are three types of neural network architectures, namely single layer feed-forward neural networks, multilayer feed-forward neural networks and recurrent networks. The single layer feed-forward neural network is a simple type of one layered neural network. The structures of the single layer feed forward neural network incorporates an input layer that projects directly onto neurons of the output layer. The number of neural network layers always exclude the input layer because the input layer is an interface and it does not perform any computation.

The second type of neural networks is identified by the appearance of more than one hidden layer. Multi hidden layers allow obtaining higher-order statistics of inputs. Figure 3.3 demonstrates a diagram of a neural network which consists of two hidden layers and one output layer.

The third type of neural network is recurrent neural networks. They differentiate themselves with two previous classes of feed-forward neural networks because they always have at least one feedback loop. Therefore, the training time of the feed-forward neural networks are often faster compared to that of the recurrent neural networks, but the recurrent neural networks have better memory capability for recalling past events.

ANN topologies used in this research have at least one hidden layer and one output. A neural network with two hidden layers and on output is illustrated in Figure 3.3. ANNs for traffic link models have following input features: day of a week (0-6), time of day (0-95), vehicle class (1-9) and travel time data of selected neighbouring links (in seconds). The output gives the travel time of a target link (in seconds). Travel times of the target link are modelled as non-linear functions of travel times in neighbouring links. Feed-forward neural networks are considered in this study.

Learning processes

The process of updating weights and biases of ANNs for minimising an empirical risk function is defined as a learning process. The learning process of ANNs may be categorised as follows:

- Supervised learning

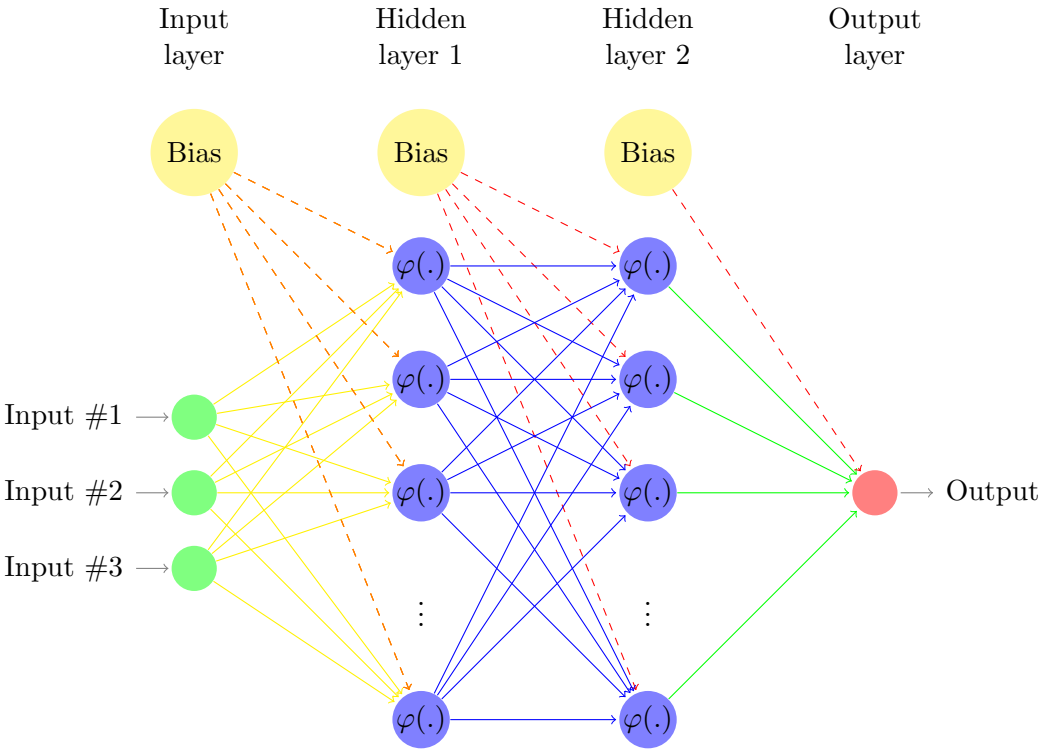


FIGURE 3.3: Two hidden layers one output ANN

The ANN is trained using a training dataset. The training dataset is made of input and output data. The input data is applied to the ANN then the output of the ANN is compared to the output data. An error is calculated by the difference between the ANN output and the actual output data. Weights and biases are adjusted based on the error's minimising process to move the ANN output closer to the target output data. The training process is repeated until the stop criteria/acceptable error is reached. Supervised learning is also referred to learning with a teacher. The supervised learning is shown in Figure 3.4.

Well-known supervised learnings include back-propagation, resilient-back-propagation and evolutionary, Haykin (2008). Thousands of regression problems in this research need to be modelled by training thousands of neural networks that have a different number of features and amount of labelled data varies.

The back-propagation (BP) algorithm is a learning procedure for training multilayer neural network models. The back-propagation algorithm is a gradient descent-based method. The characteristic of a gradient descent algorithm is that the solution searching process can get stuck in a local minimum but it does not happen too often. BP is widely used in neural network model, Gori and Tesi (1992), Ortega-Zamorano et al. (2016).

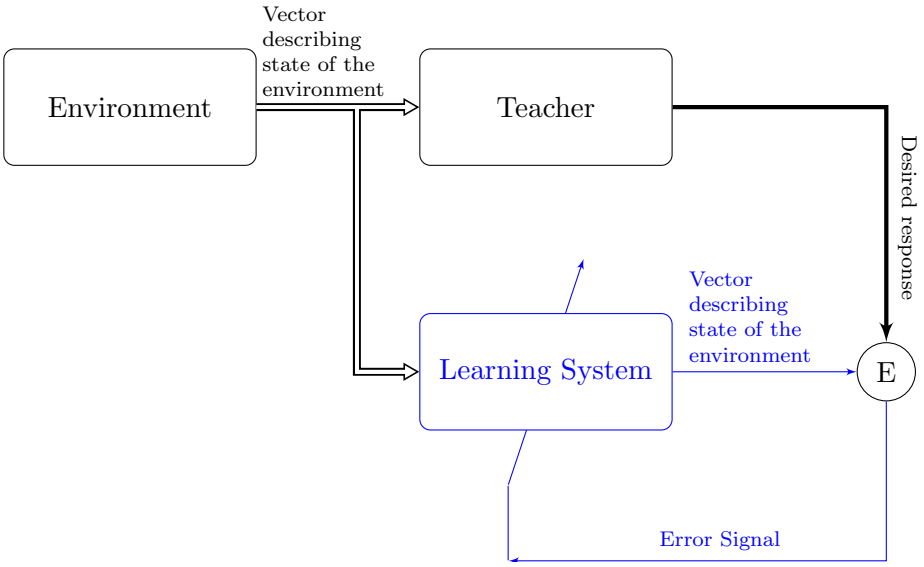


FIGURE 3.4: Diagram of learning with a teacher/supervised learning

The weights in back-propagation models are calculated using following Equation 3.13.

$$\Delta w_{ij}(t) = \alpha \times x_i(t) \times \delta_j(t) \tag{3.13}$$

where α is the learning rates, $x_i(t)$ are inputs propagating back to the i^{th} neuron at the time step t, and δ represents the corresponding error gradient.

Resilient back-propagation (RPROP) is a faster training algorithm compared to the backpropagation algorithm. The RPROP algorithm refers to the direction of the gradient (Equation 3.14). It works similarly to BP, except that the weight updates are done differently, Prasad et al. (2013).

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \times \Delta_{ij}^{(t-1)} & \text{, if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \times \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^- \times \Delta_{ij}^{(t-1)} & \text{, if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \times \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)} & \text{, otherwise} \end{cases} \tag{3.14}$$

where η is a factor and $0 < \eta^- < 1 < \eta^+$, $\Delta_{ij}^{(t)}$ is a update value at iteration t, $\frac{\partial E^{(t)}}{\partial w_{ij}}$ is an error gradient of iteration t. Weights of RPROP are updated based on two rules

given in Equation 3.15, 3.16 and 3.17:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & \text{, if } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)} & \text{, if } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0 & \text{,otherwise} \end{cases} \tag{3.15}$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \tag{3.16}$$

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)} \text{ , if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \tag{3.17}$$

If the current derivative and the previous derivative retain their signs then Equation 3.15 and Equation 3.16 are utilised. Otherwise, the Equation 3.17 is used.

Evolution learning (EL) are divided into three categories:

- weights values searching in a fixed structure ANN;
- network structure searching;
- activation functions searching;

EL in this thesis is a combination of the three above. Specificity, parameters, topology and rules of a feed-forward neural network are generated using an evolutionary algorithm. The evolution algorithm can be a genetic algorithm, Rodzin et al. (2016).

RPROP and EL are selected for the ANNs in this thesis in order to evaluate the performance of the novel methodologies as RPROP is one of the fastest learning algorithms and EL can likely eliminate the local minimum, Rodzin et al. (2016).

• Unsupervised learning

Training datasets in unsupervised learning have not known outputs (they are not labelled data), and there is no external teacher to supervise a training process (Figure 3.5). Reinforcement learning is a type of unsupervised learning. In reinforcement learning, a scalar index of performance is minimised by input-output mapping. Continued interaction with the environment conducts the process as shown in Figure 3.6. Unsupervised learning is often used for data clustering, features extraction and classification.

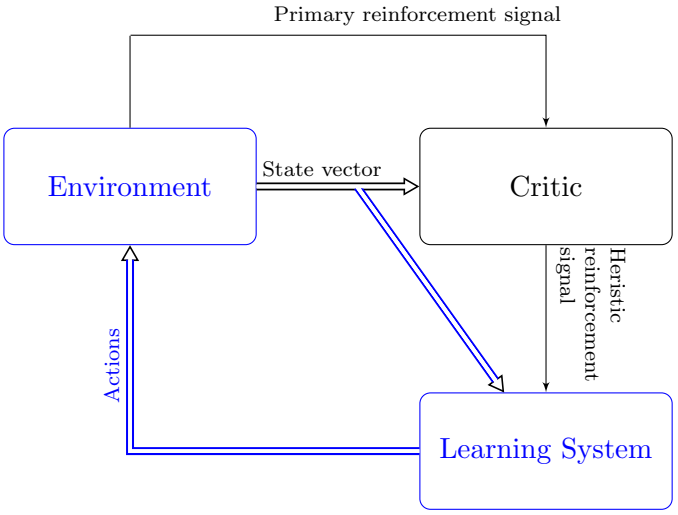


FIGURE 3.5: Diagram of learning without a teacher/unsupervised learning

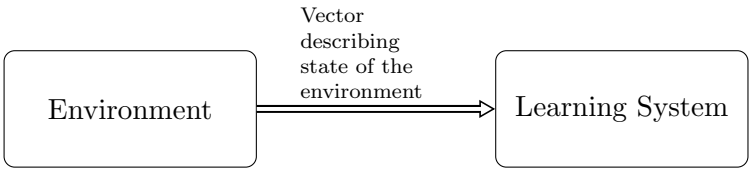


FIGURE 3.6: Block diagram of reinforcement learning

3.4 Support vector machine

Support vector machine (SVM) is a machine learning technique that uses supervised learning process. SVM techniques have been developed using statistical theory. The SVM is usually used for classification and pattern recognition problems by forming hyperplanes in a multidimensional vector space. The multi-dimensional vector space divides different labelled data into different data subsets. An extension of SVM which is called support vector machine regression (SVR) can solve non-linear regression problems, Haykin (2008). SVRs can manage multiple continuous and categorical variables.

Support vector machine constructs an optimal hyperplane by using an iterative training algorithm where an error function is minimised. Support vector machines can be technically classified into four distinct groups based on the form of the error function. In this section, two support vector regression types are focused:

- epsilon-SVM regression (regression SVM type 1)

Define the error function is:

$$\frac{1}{2}w^T w + C\sum_{i-1}^{N}\xi_i + C\sum_{i=1}^{N}\xi_i^* \tag{3.18}$$

where w is weight vector, ξ is slack variables, w^T is a transpose of w, C is a capacity constant and N is the number of training cases.

The error function is minimised subject to:

$$\begin{cases} w^t\phi(x_i) + b - y_i \leq \epsilon + \xi_i^* \\ y_i - w^T\phi(x_i) = b_i \leq \epsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0 \qquad\qquad \text{i=1,..,N} \end{cases} \tag{3.19}$$

where y_i is output ϕ is a kernel that is used to transform data from the input to the feature space, b is a constant and x_i represents independent variables.

• nu-SVM regression (regression SVM type 2).

Defined the error function is in Equation 3.20:

$$\frac{1}{2}w^T w - C(v\epsilon + \frac{1}{N}\sum_{i=1}^{N}(\xi_i + \xi_i^*)) \tag{3.20}$$

The error function is minimised subject to:

$$\begin{cases} (w^T\phi(x_i) + b) - y_i \leq \epsilon + \xi_i \\ y_i - (w^T\phi(x_i) + b_i) \leq \epsilon + \xi_i^* \\ \xi, \xi_i^* \geq 0 \qquad\qquad \text{i=1,..,N and } \epsilon \geq 0 \end{cases} \tag{3.21}$$

Kernel Functions

The matrix K is a non-negative definite matrix which is named kernel matrix:

$$a^T K a \geq 0 \tag{3.22}$$

where a is a real-valued vector that has the number of dimensions compatible with those of K. a^T is a transpose of a

Support vector machines can work with a number of kernels including Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid. Formulas of the kernels are presented in Equation 3.23.

$$K(X_i, X_j) = \begin{cases} X_i X_j & \text{Linear} \\ (\gamma X_i X_j + C)^d & \text{Polynomial} \\ exp(-\gamma \|X_i - X_j\|^2) & \text{Radial Basic Function} \\ tanh(\gamma X_i X_j + C) & \text{Sigmoid} \end{cases} \tag{3.23}$$

where X is a input vector. γ, d and C are adjustable parameters of kernel functions.

To find the solution scales, the computational complexity of a kernel-based support vector machine is $O(n^3)$ Williams and Seeger (2001) where n is the number of training instances. Storing a kernel matrix also requires a number of computer memory spaces that size quadratically with the number of training instances. Traditional SVM algorithms' training time often sizes super-linearly with the number of training instances. Consequently, support vector machines are not efficient machine learning techniques for large datasets, Williams and Seeger (2001). Therefore, SVMs are often used for small to medium-size of datasets, and they can adapt very high dimension feature space or even infinite dimensional, Krishna Menon (2018), Rahimi and Recht (2008).

The size of training datasets in this research varies from hundreds to one hundred thousands instances. Support vector machines (associated with any kernel) are challenged to utilise the datasets due to a limitation of computer resources. Hence, SVMs are employed in this work, but they do not expect to perform well not only regarding training/test time but also regarding output error.

3.5 Performance criteria

A model could demonstrate an excellent fit for the training and test dataset, but it is not necessary for the accuracy of the estimation on unseen datasets. The performance of models might be affected by a variety of matters such as over-fitting while training an artificial neural network. A model could estimate output accurately, but it could

fail in some or all error specification tests, Vlahogianni et al. (2014). Selecting the correct performance criteria for a model is an important task. The performance criteria will be used to evaluate how capable of a model. The choice of performance criteria influences how the performance of the machine learning technique is measured and compared Vlahogianni et al. (2014).

In a regression problem, estimated values are a result of approximation processes. The measure of the approximation can rely on different performance matrices. The performance matrices need to be able to evaluate how well the regression models can solve the problem. There is a load of measurements that can produce performance matrices for regression models. It is crucial to choose a performance matrix measurement appropriately. After reviewing different criteria it was decided that following performance indicators are the most suitable for the problems addressed in this research.

3.5.1 Mean squared error

Mean square error (MSE) is a highly sensitive performance matrix of an estimator as the result of a squared error. MSE is calculated based on the Equation below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \tag{3.24}$$

where n is the number of samples, y_i are actual values and $\bar{y}_i$ are estimated values.

MSE gives errors with the square of the actual unit. MSE is sensitive to large error, and it makes the error smaller if the error value is less than one. This performance matrix is an excellent choice for evaluation between traffic link models that belong to a traffic link layout. This performance metric is also used as a stop criterion in training machine learning.

3.5.2 Root mean squared error

Root mean squared error (RMSE) is a quadratic scoring rule. It is used to measures the average magnitude of the error between actual values and prediction values.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{3.25}$$

where n is the number of samples, y_i are actual values and $\bar{y}_i$ are estimated values.

RMSE gives errors with the same actual unit. RMSE ranges from 0 to ∞. The lower value of RMSE, the better the estimation is. It is indifferent to the direction of errors. RMSE is more useful when large errors are undesirable than the other performance matrices. And RMSE is not suitable for comparing the results of different size test samples. Therefore, the RMSE performance metric will not be used in this research when two different models are compared. This performance matrix is an excellent choice for the evaluation between traffic link models that belong to a traffic link layout.

3.5.3 Mean absolute error

Mean absolute error (MAE) is one of the popular performance matrices. MAE is the average of total values of a difference between an actual value and an estimated value where all individual differences have equal weight.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid y_i - \bar{y}_i \mid \tag{3.26}$$

where n is the number of samples, y_i are actual values and $\bar{y}_i$ are the estimated values.

MAE value ranges from 0 to ∞. The lower amount of MAE, the better the estimation is. It is indifferent to the direction of errors.

3.5.4 Mean absolute percentage error

Mean absolute percentage error (MAPE) is the average of the total value of the difference between an actual value and estimated value and divided by actual value

then multiplying by 100%.

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \mid \frac{y_i - \bar{y}_i}{y_i} \mid \tag{3.27}$$

where n is the number of samples, y_i are actual values and $\bar{y}_i$ are estimated values.

MAPE cannot be used if actual value might consist of zero values. An advantage of MAPE is that it can be applied to compare between estimation values and actual values of different datasets in different scales. Therefore, in this thesis, MAPE will be used to compare two traffic link models that belong to a traffic link layout or two different traffic link layouts.

3.6 Selection of meta-parameters of neural network and support vector machine

In machine learning techniques, a hyper-parameter is a parameter which is set before the learning precess starting. Different model training algorithms require different hyper-parameters (meta-parameters). The time require to train and test a model can depend upon the choice of the hyper-parameters. The hyper-parameter is usually is a real number or an integer number. Hyper-parameters are always dependent variables. Many approach can be applied to looking for a suitable set of hyper-parameters. Cross-validation technique is usually used along with hyper-parameter optimisation.

3.6.1 Cross-Validation

There is a need to validate the stability of machine learning models. The models should be generalised from most of the patterns of training datasets correctly, and they should not pick up too much noise. Validation is a process of deciding whether results are acceptable as descriptions of training datasets. Cross-validation is a statistical method of evaluating and comparing learning algorithms.

The cross-validation divides a training dataset into two segments: one segment is used to learn or train a model, and the other is used to validate the model, Refaeilzadeh et al. (2016). In a standard cross-validation process, training and validation sets must

crossover in successive rounds such that each labelled data has a chance of being validated against, Arlot and Celisse (2010), Fushiki (2011), Kan et al. (2018), Kim (2009), Molinaro et al. (2005). The cross-validation form is often k-fold cross-validation. There are some other forms of cross-validation which typically are individual cases of k-fold cross-validation.

FIGURE 3.7: K-fold cross validation (k=5)

In the k-fold cross-validation, a dataset is firstly divided into k equally partitions (folds). Afterwards, k iterations of training and validation are performed where each iteration uses a different fold of the data for the validation process, while the remaining k-1 folds are utilised for the learning process. Figure 3.7 illustrates 5-fold cross-validation.

3.6.2 Hyper-parameter optimisation

Hyper-parameter optimisation is a problem of choosing a set of optimal hyper-parameters for a learning algorithm in machine learning techniques. The same type of machine learning technique in different problems usually requires different parameters such as constraints, weights or learning rates to generalise different models. Set of those parameters is called hyper-parameters and they have to be specified regarding a specific problem so that the machine learning technique can optimally solve the problem.

Hyper-parameter optimisation is a process to find a set of hyper-parameters that yields an optimal model. Hence the model can minimise a predefined loss function on given independent data. The objective function takes a set of hyper-parameters and returns the associated loss (error), Claesen and Moor (2015). Cross-validation is often used to estimate generalisation performances, Bergstra and Bengio (2012).

A grid search algorithm conventionally performs hyper-parameters optimisation. The grid search is an exhaustive searching through a predefined subset of hyper-parameters' space of a learning algorithm. The grid search algorithm is driven by some performance metric such as cross-validation on the training set Hsu et al. (2016). The grid search algorithm suffers from the curse of dimensionality, but is often embarrassingly parallel because typically the hyper-parameters are independent, Bergstra and Bengio (2012), Zhang et al. (2009)

In this research, the grid search algorithm is utilised to optimise hyper-parameters for ANN and SVM models. Although several advanced methods can be used for optimisation processes, Burges et al. (1999), Shawe-Taylor and Cristianini (2004). They are iterative processes which cannot be easy to parallelise.

The hyper-parameters can be selected by experience, expertise and a priori knowledge of the problem, Cherkassky and Mulier (2007), Scholkopf and Smola (2001), Vapnik (2000). Manual selecting hyper-parameters may lead to many repeated trials and error attempts before getting optimums. The limitation of the methods is that they are only suitable for experts. Hyper-parameters of ANN and SVM are independent hence the grid search algorithm can perform hyper-parameter search simultaneously for both ANN and SVM.

Algorithm 3.1 Pseudo-code of grid searching for hyper-parameters where T^{in}, T^{out}, n, Θ are input matrix, output matrix, number of labelled data used for the search and a set of hyper-parameters, respectively

function GRIDSEARCH(T^{in}, T^{out}, n, Θ)
  if $SizeOf(T^{in}) > n$ **then**
    $S \leftarrow$ Randomly select n samples from $[T^{in}, T^{out}]$
  else
    $S \leftarrow [T^{in}, T^{out}]$
  end if
  $S = (t_1^{in}, t_1^{out}), (t_2^{in}, t_2^{out}), ..., (t_m^{in}, t_m^{out})$ ▷ $t_i^{in} \in T^{in}$, $t_i^{out} \in T^{out}$ respectively
  $k \leftarrow 5$
  partition S into $S_1, S_2, ..., S_k$
  $A \leftarrow$ Machine learning algorithm
  for each $\theta \in \Theta$ **do**
    for p=1 to k **do**
      $h_{i,\theta} = A(S \setminus S_i; \theta)$ ▷ limit under-fitting and over-fitting are applied
    end for
    $error(\theta) \leftarrow \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_i, \theta)$
  end for
  $\theta_{best} \leftarrow argmin_\theta[error(\theta)]$
  $h_{\theta_{best}} \leftarrow A(S; \theta_{best})$
end function

The number of data samples in traffic link models is varied from hundreds to thousands. The grid searching algorithm performing on the complete dataset is a time-consuming process when the number of labelled data is significantly large. In this research, a maximum of 1000 labelled data in the dataset are randomly chosen for the hyper-parameters optimisation process. The hyper-parameters which are a result of grid searching algorithm are applied to train ANN and SVM models on the complete dataset subsequently. The mean square error (MSE) is used to assess the performance of the ANN and the SVM models. The grid-search algorithm used in this research is to describe in Algorithm 3.1.

3.7 Over-fitting and under-fitting with machine learning techniques

The performance of a machine learning techniques depends on a number of factors, i.e. a sufficient number of training samples, the structure of the machine learning models, feature selection, etc. One of the primary elements of poor performance in machine learning models is either over-fitting or under-fitting labelled data while training the machine learning models. This section discusses the concept of generalisation in machine learning techniques, the over-fitting and under-fitting problems, and how to prevent over-fitting and under-fitting.

Generalisation in machine learning technique

A generalisation relates to how well machine learning techniques model a particular problem. The performance of the machine learning model is evaluated using a specific labelled data that had not seen by the model on the learning process. The performance of a machine learning technique is adequate if the model can adapt any data from the problem domain. The closer a model approximates values match the observed values, the better the model is.

Over-fitting vs Under-fitting in machine learning techniques

Figure 3.8 illustrates an example of three generalisation results after a machine learning model is trained. Under-fitting occurs when a model is too simple which makes it inflexible in learning from datasets. Few features or regularised too much would be used to indicate simple models. Simplistic learners conduce to have less variance in their estimations but more bias towards incorrect results. Meanwhile, complex learners offer to have higher variation in their estimates. Over-fitting usually originates from one of these scenarios: complicated model and an insufficient number of labelled data, and unlearnable labelled data, Bilbao and Bilbao (2017), Lawrence and Giles (2000), Liu et al. (2008).

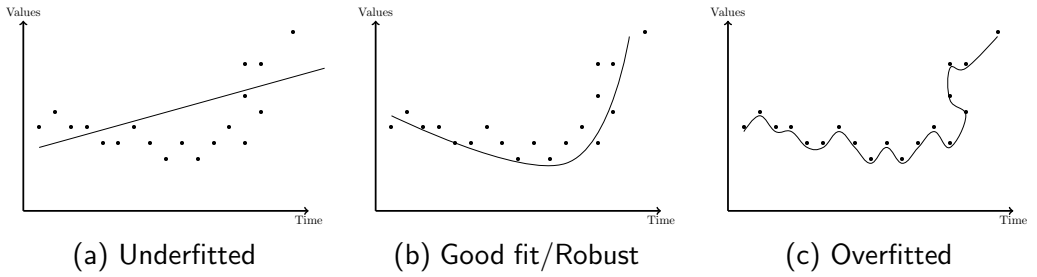
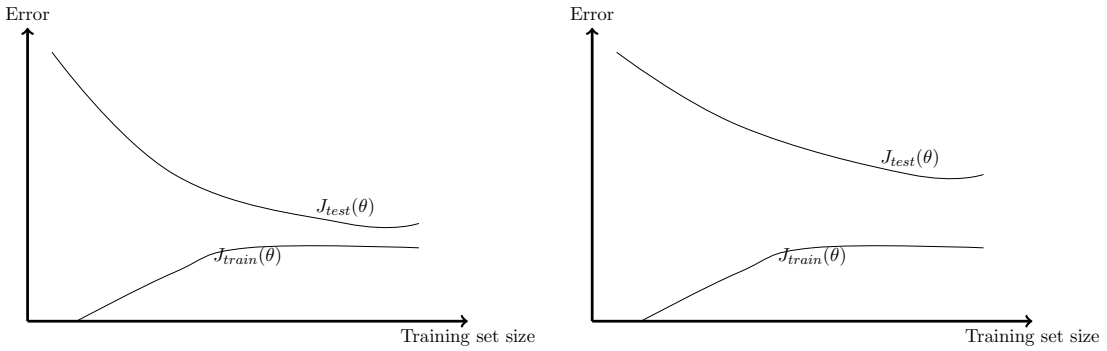


FIGURE 3.8: Under-fit, robust and over-fit when a machine learning model is trained.

Preventing of under-fitting

As mentioned in the previous section, a model is under-fitting when it is too simple regarding a labelled dataset which it is trying to model. Figure 3.9 illustrates training and test errors of a model which is high bias (Figure 3.9(a)) and high variance (Figure 3.9(b)).

To prevent high variance, increasing the number of training instances would probably help, otherwise the model complexity needs to be reduced. On other hands, to prevent high bias, the model complexity needs to be increased. Figure 3.10 shows the relationship between model complexity vs the error of the model on training data and error of the model on validation data, Liu et al. (2008).



(a) High bias (high error on both the training and test sets)

(b) High variance (a significant gap between training and test set errors).

FIGURE 3.9: High bias (a) and high variance (b) in training machine learning models.

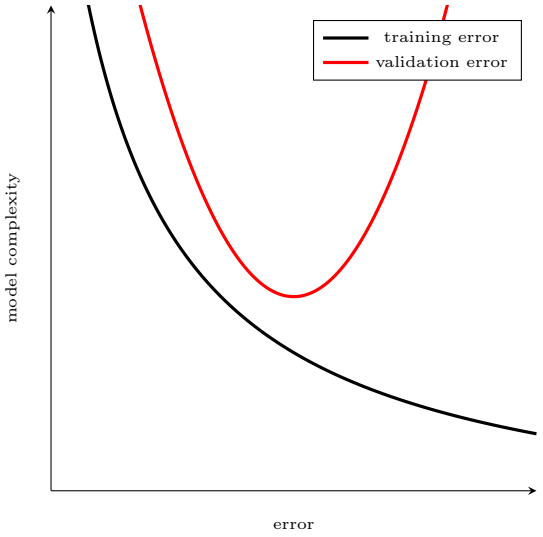


FIGURE 3.10: Model complexity vs error on training and evaluation dataset.

Preventing of over-fitting

There are several options to prevent over-fitting. They are cross-validation, increasing number of training instances, features selection, early stopped training, regularisation and ensembling. The cross-validation is the same technique that was already described in Section 3.6.1. A novel methodology for increasing the number of training instances is one of the contributions of the thesis where the similar model searching can provide more data in a training dataset of a selected model. Features selection and early stopped training are applied on the training algorithms which are separately designed for particular machine learning techniques. Regularisation and ensembling are not utilised in this thesis.

3.8 Clustering algorithms

Clustering is an unsupervised machine learning methodology. In theory, if data points belong to a group, they should share similar features and properties, while data points in different groups should have various distinct and properties.

3.8.1 K-mean clustering

K-mean probably is the most well-known unsupervised learning algorithm. The k-mean can cluster n unlabelled data into k clusters. The algorithm works iteratively to decide an unlabelled data to one of the k clusters. Unlabelled data is grouped based on the analysis similarity of the feature. The k-mean own the advantage of a linear complexity $O(n)$. A disadvantage of the k-mean algorithm is that the number of clusters needs to be predefined; however selecting an appropriate k is a difficult task. The k-mean initials the k cluster randomly, therefore, different runs may give different clustering results.

Given n observations $(x_1, x_2, .., x_n)$ where $x_i = \{x_{i1}, ...x_{ip}\}$, p is number of dimension of vector x_i. The k-mean aims to minimise an objective function:

$$O = \sum_{k=1}^{K} \sum_{i \in c_k} ||x_i - m_k||^2 \tag{3.28}$$

where O is an objective function, $||x_i - m_k||^2$ is a distance between x_i and m_k, $m_1, m_2, ...m_k$ are centroids of corresponding clusters: $c_1, c_2, ...c_k$.

Different values of k give a different distance between members of a cluster to the cluster centroid. Therefore, the mean distance between data points and their cluster centroid is used as an indicator to determine a suitable k for clustering problems. When k is increased, clusters' centroids have always reduced the distance to members of the cluster. The indicator will reach zero when k is an equal number of data points. K will roughly be determined where the indicator decrease sharply shifts (Figure 3.11).

3.8.2 Gaussian mixture model clustering

Gaussian mixture model (GMM) is a probabilistic model based on the Gaussian distribution. The mixture describes the probability distribution of an observation x in

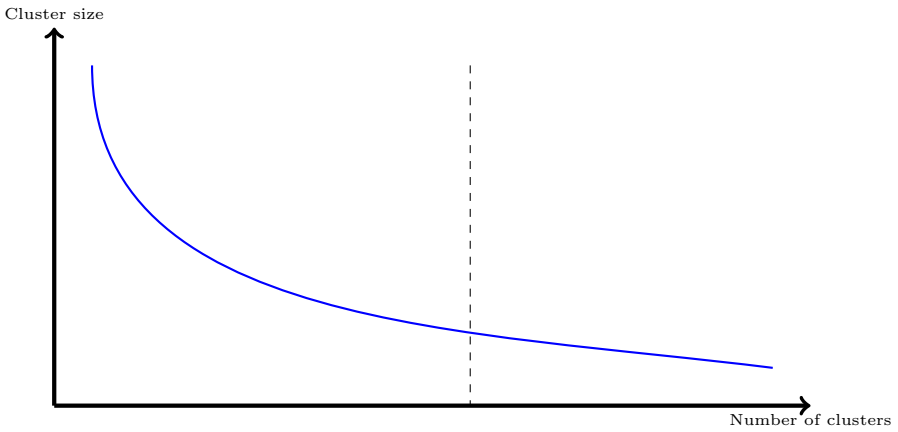


FIGURE 3.11: Size of cluster vs Number of clusters

the overall population. GMM is defined as:

$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\sigma|^{1/2}}\exp\{-\frac{1}{2}(x-\mu)^T\sigma^{-1}(x-\mu)\} \tag{3.29}$$

$$p(x) = \sigma_{k=1}^{K}\pi_k\mathcal{N}(x|\mu_k,\sigma_k) \tag{3.30}$$

where $\mathcal{N}(x|\mu_k,\sigma_k)$ is a probability density function of Gaussian distribution, $\mu_1...\mu_k$ are the means of components, μ is vector composed of all the individual $\mu_1...\mu_k$, $\sigma_1...\sigma_k$ are the variances of each components, σ is vector composed of all the individual $\sigma_1...\sigma_k$, K is the number of Gaussian components, D is the dimension of the observation vector x. $\pi_1...\pi_k$ are mixture weights, and $p(x)$ is posterior distribution of x.

GMM is an unsupervised clustering method which accommodates clusters that have different sizes and correlation structures within them. Every cluster is described by the mean vector μ and the covariance matrix σ. The cluster member is assigned based on the probability that has been generated using its μ and σ. This probability is computed using the formula:

$$p(x|\pi,\mu,\sigma) = \pi p(x) \tag{3.31}$$

3.8.3 Selection a number of clusters for clustering algorithm

Determining an optimal number of clusters in a clustering problem is a fundamental issue. Many clustering techniques require predefined the number of clusters i.e. k-means, k-medoids, hierarchical clustering, Gaussian mixture model etc. Unfortunately, there are no direct answers to the question. The number of clusters

depends on the specific problem and on which method for measuring similarities and parameters used for partitioning is employed. These methods include statistics methods, Bischof et al. (1999), Hamerly and Elkan (2003) and optimisation methods, Kassambara (2017), Pelleg and Moore (2000). Heuristics, rules of thumb might determine the number of clusters. There are more than thirty methods that can be used to calculate the number of clusters for clustering problems, Kassambara (2017).

There are no direct methods to select an appropriate k for a clustering algorithm. In this thesis, the multivariate Gaussian mixture model is extended as outliers detection methodology. According to Department of Transport (2012), a day usually has a morning peak and an evening peak. They divide time in a day into four intervals: between evening peak and morning peak, morning peak, between morning peak and evening peak, evening peak. Each time interval should produce a travel time distribution, Tang et al. (2018). Therefore, the number of clusters (k) for the outlier detection algorithm is heuristically set to 5 which represents four distributions corresponding to four intervals of time in a day and one is for outliers.

3.9 Genetic algorithm

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction to produce offspring of the next generation. Genetic algorithm are self-adapting strategies for searching, based on the random exploration of the solution space, Goldberg (1989), Holland (1992).

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance of surviving. This process keeps on iterating, and at the end, a generation with the fittest individuals will be found. Five phases are considered in a genetic algorithm: (1) Initial population, (2) Fitness function, (3) Selection, (4) Crossover, (5) Mutation, Steeb (2008).

Initial population

The process begins with a set of individuals which is called a Population. Each is a solution to the problem you want to solve. An individual is characterised by a set of parameters (variables) known as Genes. Genes are joined into a series to form a Chromosome (solution). Usually, binary values are used (series of 1s and 0s) in the individual (Figure 3.12), Steeb (2008).

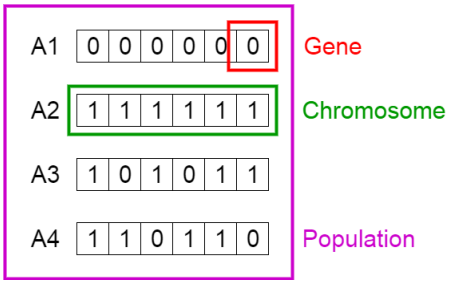


FIGURE 3.12: Gene, Chromosome and Population

Fitness function

The fitness function determines how good an individual is. It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score, Steeb (2008).

Selection

The idea of the selection phase is to select the fittest individuals and let them pass their genes to the next generation. Two pairs of individuals (parents) are chosen based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction, Steeb (2008).

Crossover

Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a crossover point is chosen at random from within the genes. For example, consider the crossover point to be three as shown in Figure 3.13.

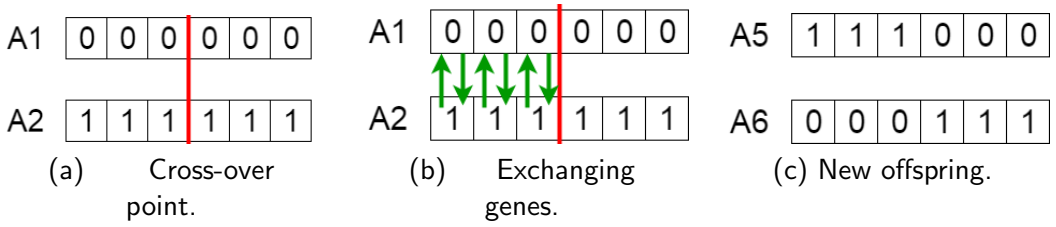


FIGURE 3.13: Cross-over process

Mutation

In particular new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped (Figure 3.14). The mutation occurs to maintain diversity within the population and prevent premature convergence, Steeb (2008).

FIGURE 3.14: Mutation

Combination of genetic algorithm and neural network

In this thesis, a fully connected feed-forward neural network (FF-ANN) is utilised. FF-ANNs need three parameters: the number of layers, the number of neurons per layer and an activation function. Every neuron and connection in the FF-ANN is specified directly and explicitly in the genotype of the GA. The population of genotypes is directly mapped to the population of neural network phenotypes. The fitness function is the performance of a neural network phenotype. The mutation probability and the crossover probability are fixed to 0.1 and 0.24 following a series of manual tuning and experiments. Population size is a pre-defined parameter of GA. The genetic algorithm is used to optimise FF-ANNs parameters. Therefore, hyper-parameter includes the number of layers, number of neurons per layer, activation function and population size of GA.

Chapter 4

Temporal and spatial dependencies in traffic links

4.1 Introduction

This chapter mainly focuses on an investigation of the correlations between parameters on neighbouring traffic links. A novel Neighbouring Link Inference Method (NLIM), a methodology to model the temporal and spatial dependencies between travel times of a target link and its adjacent links is proposed.

NLIM employs machine learning techniques to discover the temporal and spatial relationship between travel times of a target link and its neighbouring links. Following the training process, travel times of the target link can be estimated from travel times observed on its adjacent links using the introduce NLIM models.

Travel time data gathered from moving observers often show high data sparsity and data irregularity e.g. case studies described in Jenelius and Koutsopoulos (2013), Lu et al. (2018), Maiti et al. (2014), Meng et al. (2017), Passow et al. (2013), Pirc et al. (2015), Rahmani et al. (2013), Tang et al. (2018), Wan and Vahidi (2014). To cope with this difficulty, a novel similar model searching (SMS) methodology is proposed. The method is used to enhance the learning performance of machine techniques learning of temporal and spatial dependencies of travel times in traffic links. The main idea of SMS is to discover a list of traffic link models which are similar to the target traffic link model.

After that, the labelled data of similarity models together with the target model training dataset is utilised as the new labelled dataset for training the target model.

To improve the performance of NLIM, a travel time outlier detection/removal method which relies on multivariate Gaussian mixture models is also proposed in this chapter.

Four different machine learning techniques are employed in the proposed methods. They are multilinear regression (MLR), feed forward resilient backpropagation neural network (FF-RPROP-ANN), feed forward evolution learning neural network (FF-EL-ANN) and support vector machine regression (SVR). The NLIM technique and its implementation are discussed in detail in this chapter.

Four different datasets for experiments, including an artificial dataset, SUMO dataset, WebTRIS dataset and Floating Car Data (FCD) dataset, are described in this chapter. Different datasets have different data formats. Therefore the methods for pre-preprocessing the datasets to be usable as training and test data with NLIM are introduced and discussed.

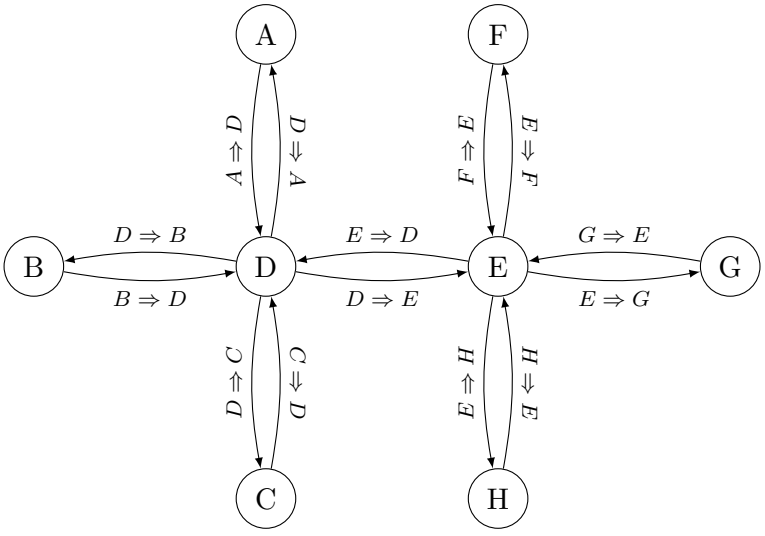
The chapter is set out as follows: Section 4.2 gives definitions for traffic link layout, traffic link model and presents a data transform method to convert from original data formats to usable formats for a traffic link model. Section 4.3 presents the preprocessing of data for traffic travel time modelling including a definition of data sparsity, a process for removing blank data and the novel application of Gaussian mixture model for detecting and eliminating travel time outliers and data normalisation. Section 4.4 and 4.5 describe the novel NLIM and SMS methodologies. Section 4.6 presents and discusses the experimental set-up of machine learning techniques mentioned above. Section 4.7 describes the four different datasets and their data structures.

4.2 Traffic link layout and traffic link model

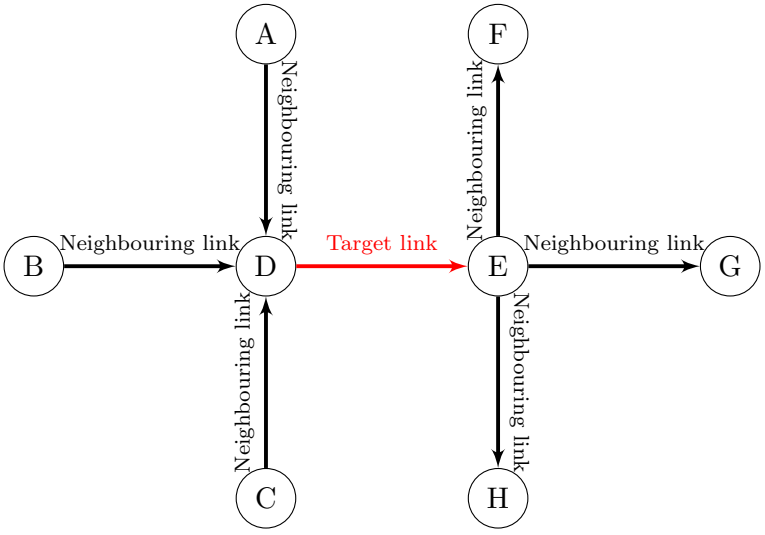
4.2.1 Definition of traffic link layout

Transportation systems which include structure and flows are commonly represented using networks as an analogy. It is a sub-category of the spatial network since transport

networks' design and evolution are physically constrained. A traffic links layout is a simplified representation of a small part within the traffic network. It depicts junctions and roads. However, roads are presented as unidirectional connections between the junctions to indicate a traffic flow direction. A node is a traffic link network term which indicates intersections to the transportation network, Zhao and Spall (2016) and the structure of links within a locations' system. A traffic link is a single direct route between two nodes in a network, Rodrigue et al. (2013), Zhao and Spall (2016).



(a) Normal traffic link layout



(b) Traffic link layout used in this thesis

FIGURE 4.1: A normal traffic link layout vs a traffic link layout used in this thesis.

For clarification, consider a normal traffic link layout shown in Figure 4.1(a) and a neighbouring traffic link layout defined in this thesis shown in Figure 4.1(b). Both of the normal traffic link layout and the neighbouring traffic link layout comprise of eight

junctions (note): A, B, C, D, E, F, G and H. The normal traffic link layout has 14 directional connections: AD, DA, BD, DB, CD, DC, DE, ED, DF, FD, DG, GD, DH and HD. In the neighbouring link layout, there are seven directional connections: BD, CD, DE, EF, EG and EH based on a assumption that traffic-related information in front connections (AD, BD, CD) and rear connections (DF, DG, DH) affect those in a middle connection (DE). From now on, in this thesis, traffic link layout means neighbouring traffic link layout.

In this thesis, these connections are defined as traffic links. Each traffic link is between start and end junction. Each junction has its geographic coordinates such as A(x_a, y_a), B(x_b, y_b), C(x_c, y_c), D(x_d, y_d), ..., H(x_h, y_h) where x is a longitude and y is a latitude. The coordinates are used to determine the neighbouring links of a link from a digitised traffic map.

Traffic links layout consists of a targeted link and adjacent links. The target link is a link where traffic-related information needs to be determined. The neighbouring links are links that might contain information that can be used for the traffic parameters estimation.

In the model shown in Figure 4.1(b), DE is the target link ($L_O = \{DE\}$) and AD, BD, CD, EF, EG, EH are neighbouring links ($L_N^{DE} = \{AD, BD, CD, EF, EG, EH\}$). Specifically, AD, BD, CD are rear neighbouring links ($L_{NR}^{DE} = \{AD, BDCD\}$) and EF, EG, EH are front neighbouring links($L_{NF}^{DE} = \{EF, EG, EH\}$) of link DE. They are determined based on the direction of the vehicles' movement. More precisely, e.g. all links that lead to the start node of the target link are adjacent rear links. Similarly, the links from the end node of the target link are adjacent front links and indicate vehicles movement outwards. In Figure 4.1(b), there are six neighbouring links in total. $[L_O, L_N^{DE}]$ denotes the link layout.

A traffic link is a neighbouring link of a target link if it shares a node with the target traffic link. The beginning node of a front link is always the end node of its target traffic link, and the end node of a rear link is always the beginning node of a target traffic link.

4.2.2 Definition of traffic link model

In this thesis, the traffic link model consists of a target link and at least one of the adjacent links of a traffic link layout. A full traffic link model includes the target link and all of the neighbouring links. Assume that a traffic link layout has N neighbouring links, then the total number of traffic link models can be calculated as follows:

$$\xi = \sum_{k=1}^{N} \frac{N!}{k!(N-k)!} \tag{4.1}$$

where ξ is the total number of traffic link models, N is the number of neighbouring links in the traffic link layout.

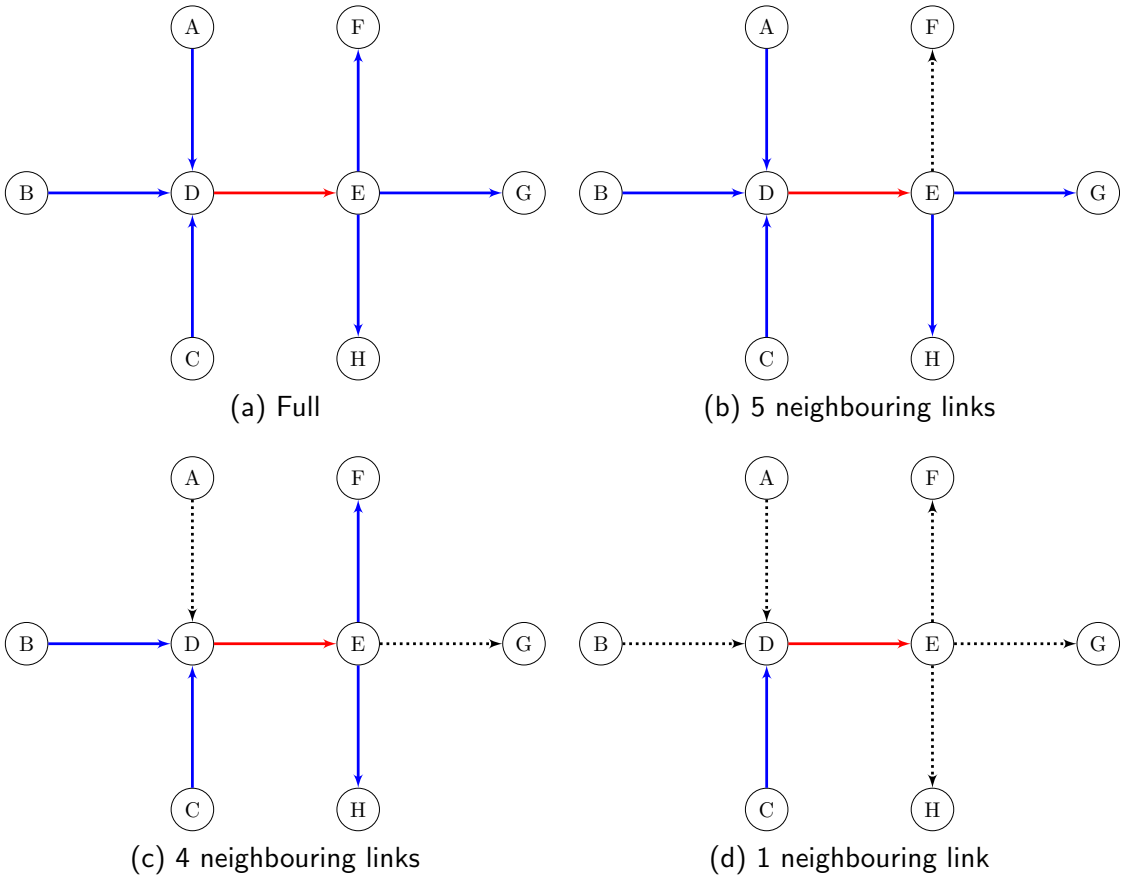


FIGURE 4.2: Traffic link model examples for the traffic link layout shown in Figure 4.1(b). The red arrows represent target links, blue arrows represent neighbouring links, and dashed arrows represent links not used in the model due to missing data. The angle and length of the links are not explicitly considered in the graph but implicitly in the travel time data itself.

L_N denotes a set of neighbouring links in a traffic link layout, L_M denotes a set of neighbouring links in a specific traffic link model ($L_M \in L_N$). According to the Equation 4.1, the number of adjacency links included in L_M ranges from 1 to N. Figure 4.2 demonstrates few examples of traffic link models with a different number of neighbouring links. These traffic links models were derived from the traffic link layout shown in Figure 4.1(b). Solid arrows represent the links that are included in the traffic link model. Traffic link $L_O = DE$ is the target link. Figure 4.2(a) presents a full traffic link model of the link layout that includes all neighbouring links($L_M^{DE} = L_N^{DE} = \{AD, BD, CD, EF, EG, EF\}$). Figures 4.2(b), 4.2(c) and 4.2(d) show examples of different traffic links models where only a subset of data is available for constructing the model. It is important to highlight that these models therefore have differing numbers of links; e.g. in Figure 4.2(b) $L_M^{DE} = \{AD, BD, CD, EG, EF\}$, in Figure 4.2(c) $L_M^{DE} = \{BD, CD, EF, EH\}$ and in Figure 4.2(d) $L_M^{DE} = \{CD\}$.

This proposed definition of a traffic link model has particular advantages that will be exploited later in this work. The work presented in this thesis introduces methods to estimate traffic parameters for the target link based on traffic parameters of adjacent links. However, it must be highlighted that data for traffic links might not always be available at the required time. Only a small portion of real-world traffic networks is monitored; mainly major roads. The proposed approach addresses this problem by creating a set of traffic link models based on the links that contain data. The accuracy of traffic parameters estimation depends on the availability of data and degree of the relationship between links that can be detected.

4.2.3 Data coding for a traffic link model

Travel time data of links in a traffic link model are coded in a matrix form where each entry represents whether travel time of a vehicle type is present at a specific day of a week and a particular time of interval in the day. The vehicle classes ranges from 1 to 9 ($vehCl$) (Details of vehicle classes will be described along with corresponding datasets). The day of a week ($dayofweek$) has a value range from 0 to 6 that represent Monday to Sunday and the interval of time ($timeItv$) is coded as variable *slot* in the Algorithm 2 or as subject to the data source behaviour (moving observers and stationary observers) which is already discussed on the literature review.

Define S is a data matrix of a traffic link layout which includes data of a target link and data of all its neighbouring links, S^{in} is the data matrix of the neighbouring links and S^{out} is the data matrix of the target link.

The structure of a data matrix for the full traffic model ($L_O = \{DE\}$ and $L_M^{DE} = \{AD, BD, CD, EF, EG, EH\}$) (defined as S_f, S_f^{in} and S_f^{out}) shown in Figure 4.3 is presented below:

$$\begin{cases} S_f = \left[\{dayofweek, timeItv, vehicleCl, t_{AD}, t_{BD}, t_{CD}, t_{DE}, t_{EF}, t_{EG}, t_{EH}\}\right] \\ S_f^{in} = \left[\{dayofweek, timeItv, vehicleCl, t_{AD}, t_{BD}, t_{CD}, t_{EF}, t_{EG}, t_{EH}\}\right] \\ S_f^{out} = \left[\{t_{DE}\}\right] \end{cases} \tag{4.2}$$

where $t_{AD}, t_{BD}, ..., t_{DE}$ are travel time data of $vehicleCl$ on corresponding traffic link AD, BD, ..., DE at a specific time of week ($dayofweek, timeItv$). If the travel time data does not exist at the specific time of a week, the value of corresponding T is set to blank ("-").

An example of matrix S_f, S_f^{in}, S_f^{out} is demonstrated in Equation 4.3, 4.4, 4.5. The values in these matrices were set arbitrarily to give an example how data could be populated.

$$S_f = \begin{bmatrix} [dayofweek] & [timeItv] & [vehicleCl] & [t_{AD}] & [t_{BD}] & [t_{CD}], & [t_{DE}] & [t_{EF}] & [t_{EG}] & [t_{EH}] \\ 1 & 0 & 0 & - & - & - & - & - & 300 & - \\ 1 & 1 & - & - & - & - & - & - & - & - \\ 1 & 2 & 0 & 105 & 500 & - & - & 270 & - & - \\ 1 & 3 & 0 & - & - & - & - & - & - & 122 \\ 1 & 4 & 0 & 105 & 520 & 200 & 250 & 279 & 310 & 123 \\ 1 & 5 & 0 & - & - & 200 & - & - & - & - \\ 1 & 6 & 0 & - & - & - & 253 & - & - & - \\ 1 & 7 & - & - & - & - & - & - & - & - \\ 1 & 8 & - & - & - & - & - & - & - & - \\ 1 & 9 & - & - & - & - & - & - & - & - \end{bmatrix} \tag{4.3}$$

$$S_f^{in} = \begin{bmatrix} [dayofweek] & [timeItv] & [vehicleCl] & [t_{AD}] & [t_{BD}] & [t_{CD}], & [t_{EF}] & [t_{EG}] & [t_{EH}] \\ 1 & 0 & 0 & - & - & - & - & 300 & - \\ 1 & 1 & - & - & - & - & - & - & - \\ 1 & 2 & 0 & 105 & 500 & - & 270 & - & - \\ 1 & 3 & 0 & - & - & - & - & - & 122 \\ 1 & 4 & 0 & 105 & 520 & 200 & 279 & 310 & 123 \\ 1 & 5 & 0 & - & - & 200 & - & - & - \\ 1 & 6 & 0 & - & - & - & - & - & - \\ 1 & 7 & - & - & - & - & - & - & - \\ 1 & 8 & - & - & - & - & - & - & - \\ 1 & 9 & - & - & - & - & - & - & - \end{bmatrix} \tag{4.4}$$

$$S_f^{out} = \begin{bmatrix} [t_{DE}] \\ - \\ - \\ - \\ - \\ 250 \\ - \\ 253 \\ - \\ - \\ - \end{bmatrix} \tag{4.5}$$

4.3 Preprocessing data

4.3.1 Data sparsity

In a database, sparsity and density describe the number of cells in a table that are empty (sparsity), and that contain information (density), Oracle (2018). In this thesis, the sparsity of a dataset is a measurement indicator. The lower the sparsity of the dataset, the higher amount of available travel time data from moving observers in the traffic link model. The sparsity (R) is the matrix sparsity of the data matrix corresponding to the traffic link model. R is calculated based on the equation below:

$$R = 100 \times \frac{numEmpties}{n \times m} \tag{4.6}$$

where $numEmpties$ is the number of empty entries in the matrix, n and m are the number of the rows and the number of the columns, respectively.

For example, the sparsity of the travel time dataset (R) that is shown in Equation 4.3 is calculated as below:

$$R = 100 \times \frac{60}{100} = 60\% \tag{4.7}$$

4.3.2 Empty data entries removal

Some machine learning techniques use labelled data to generalise the relationship between input and output data. If the data has empty entries, many machine learning techniques cannot utilise these instances for modelling. Therefore, in this work, all blank/empty entries need to be removed before the dataset can be used for training

and testing. Algorithm 4.2 is designed to pre-process data to remove any blank element within the source dataset and produce a dense target dataset($R = 0\%$). After applying Algorithm 4.2, matrix S is transformed into the matrix T. The details of the algorithm are described in Algorithm 4.2.

Algorithm 4.2 Pseudo-code of preprocessing to remove blank data

```
1: function RemoveBlank(S)
2:     for each row ∈ S do
3:         for each component ∈ row do
4:             ok ← true
5:             if component = blank then
6:                 ok ← false
7:             end if
8:         end for
9:         if ok = true then
10:            T ← row
11:        end if
12:    end for
13: end function
```

Appling the Algorithm 4.2 on S_f of the traffic link layout shown in Figure 4.1(b), the T_{full}, T^{in}_{full}, T^{out}_{full} shown as below:

$$T_f = \begin{bmatrix} 1 & 4 & 0 & 105 & 520 & 200 & 250 & 279 & 310 & 123 \end{bmatrix} \tag{4.8}$$

T^{in}_f and T^{out}_f are generated based on T_f:

$$\begin{cases} T^{in}_f = \begin{bmatrix} 1 & 4 & 0 & 105 & 520 & 200 & 279 & 310 & 123 \end{bmatrix} \\ T^{out}_f = \begin{bmatrix} 250 \end{bmatrix} \end{cases} \tag{4.9}$$

$size(T)$ is the size of the data set T which is the number of rows of matrix T, e.g in Matrix 4.8, the number of rows of matrix T_f is 1 , so $size(T_f) = 1$

4.3.3 Outlier detection based on multivariate Gaussian mixture model

In statistics, an outlier is an observation point that is distant from other observations. The outliers influence statistical characteristics, and they may lead to erroneous conclusions, Lin et al. (2014). To remove outliers in matrix T, an application for outlier detection methodology based on multivariate Gaussian mixture model

(m-GMM) is proposed. The m-GMM is used to cluster the rows of the matrix T into k row distributions where each element in a row is a variable of the multivariate. Structure and size of the rows distributions (clusters of rows) are indicators to detect travel time outliers.

Define ϵ as a predefined threshold that is used to distinguish normal travel times from outliers. The outliers (rows) are detected based on the mixture weights of the data components that have been clustered by the m-GMM. If a mixture weight π_i of data component i^{th} is less than or equal to ϵ then the rows in component i^{th} are detected as outliers. k is the number of clusters.

The steps to detect travel time outliers based on m-GMM are described in Algorithm 4.3.

Algorithm 4.3 Pseudo-code of outlier detection/removal algorithm based on multivariate Gaussian mixture model

```
function DR-M-GMM(T, ε, k)
    Apply m-GMM for T to get μ_i, σ_i and π_i (i=1,2,...,k) where Σ_{i=1}^{k} π_i = 1
    T_temp ← ∅
    for i = 1 to k do
        if π_i ≥ ε then
            T_temp ← Cluster^i
        end if
    end for
    T ← T_temp
end function
```

The algorithm is designed for outlier detection on a traffic link model. Matrix T contains labelled data of the traffic link model. Each row often belongs to a specific traffic pattern that represents a relationship between links in the traffic link model. If rows are classified into a cluster by the multivariate Gaussian mixture model, and if the weight of the cluster is less than or equal to the outlier indicator ϵ, the row cluster is corresponding to outliers.

4.3.4 Feature scaling

The travel time for a specific link depends on the length of that link and may vary based on day, time, and vehicle class. Many machine learning techniques require consistent input and output data and varying scales in the data may negatively affect the learning process, Youn and Jeong (2009). Therefore, feature scaling is applied in preprocessing before it is used by the machine learning technique.

The MinMax normalisation is used entirely in this research where data needs to be normalised. The MinMax normalisation is described in an equation below:

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.10}$$

and the invert normalisation is shown in Equation 4.11.

$$x = y(x_{max} - x_{min}) + x_{min} \tag{4.11}$$

where x_{max} and x_{min} are the maximum and minimum values in X, where X is the set of values x. It can be seen that if $x = x_{min}$ then y=0 and if $x = x_{max}$ then y=1.

4.4 Neighbouring inference method

In this section, a novel travel time estimation technique is introduced, namely the neighbouring link inference method (NLIM). The proposed method is introduced to estimate the travel time of a target link that currently has no observed travel time data available. NLIM has been designed to be scalable and hence applicable to large scale traffic models.

To estimate the travel time of the designated link, the proposed method models the relationships between travel time of a target link and traffic parameters of neighbouring links where data is available. In other words, the travel time of the target link is estimated using the measured travel time in its neighbouring links.

To deal with high data sparsity, relationships of all possible traffic link models of a traffic link layout are learned using machine learning techniques. The relationships of links in total ξ models is carefully trained and tested. Any traffic link model in ξ models which has $size(T)$ greater than a predefined threshold ($\gamma_{threshold}$) is utilised to train on a machine learning technique and the maximum number of models are trained in the traffic link layout depends on the data sparsity of links in a traffic link layout.

Sufficient data is required to provide reliable estimates of the relationship otherwise the estimates of the model are probably unstable, Bilbao and Bilbao (2017), Lawrence and Giles (2000), Liu et al. (2008). Performance of models trained with small sample sizes cannot be guaranteed. NLIM has been designed to be used in extensive traffic

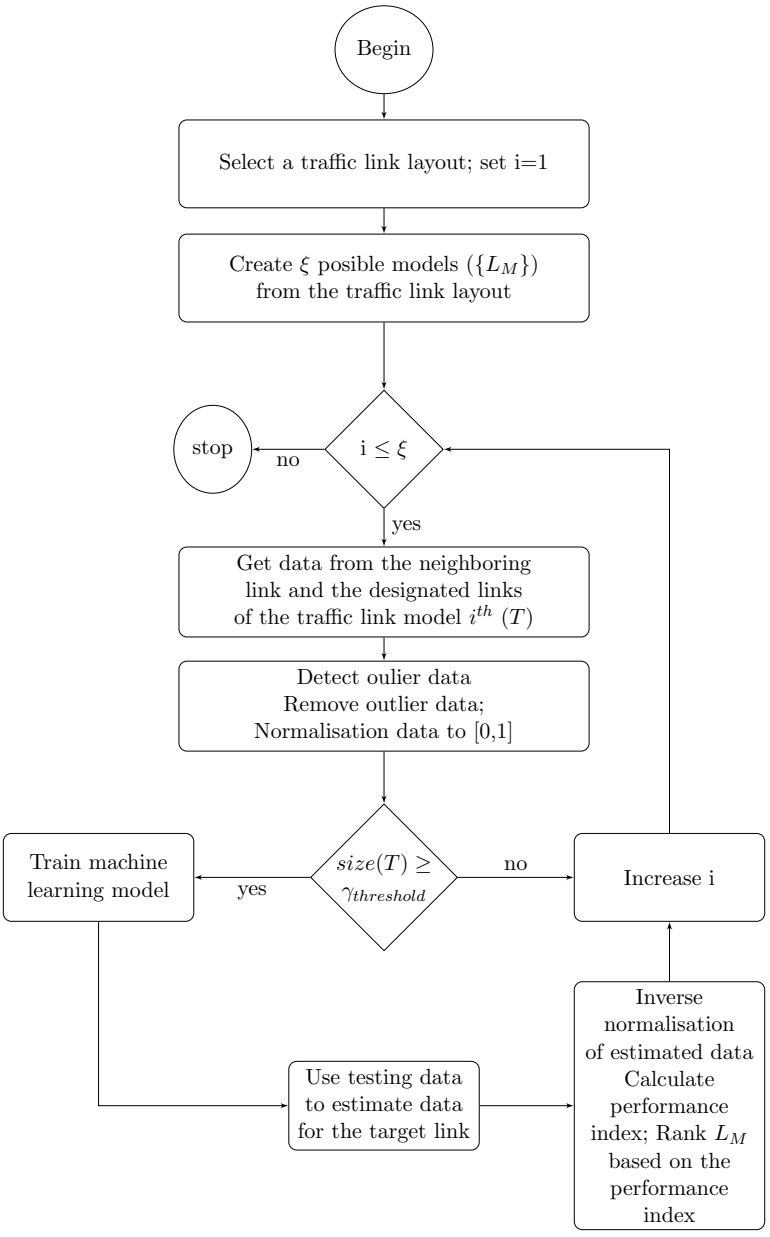


FIGURE 4.3: Diagram of Neighbouring Link Inference Method.

networks ultimately. The proposed needs to be able to model for most of the traffic links in the traffic networks. $\gamma_{threshold}$ is chosen based on the requirement of a standard machine learning technique: multi-linear regression. According to studies of Green (1991), Tabachnick and Fidel (2007), $\gamma_{threshold}$ is calculated using rules-of-thumb for which is used to determine an adequate number of labelled samples for multiple regression problems (Equation 4.12).

$$\gamma_{threshold} = 50 + 8 * f \tag{4.12}$$

where f is the number of features (predictors).

The overview of the proposed method is presented in a diagram in Figure 4.3. The training process of machine learning techniques will be described in the following section. Several machine learning techniques have been chosen to demonstrate the proposed method.

As discussed in Section 3.8.3, there is no method to select an appropriate k for a clustering algorithm. However, a working day usually has a traffic morning peak and an evening peak, Department of Transport (2012). They divide time in a day into four intervals: between evening peak and morning peak, morning peak, between morning peak and evening peak, evening peak. Each time interval should produce a travel time distribution, Tang et al. (2018). Therefore, the number of clusters (k) for the outlier detection algorithm has been set to 5 which represents four distributions corresponding to 4 intervals of time in a day and one for outliers.

The choice of ϵ affect the number of outliers. According to the study of Zheng and McDonald (2009), a majority of the travel times have fuzzy c-mean cluster membership values between 0.9 and 1. In the work of Lin et al. (2014), if the percentage of travel time in a cluster is less than or equal to 15%, then the travel times are outliers. In this thesis, due to the sparsity of the data, ϵ is set to 0.1 to utilise as much travel time data as possible.

Steps of the proposed NLIM method is described in Algorithm 4.4. In Algorithm 4.4, Hyper-parameter is a set of parameters of a machine learning technique which is optimised using GRIDSEARCH function (Algorithm 3.1). $LEARNING(T^{in}, T^{out}, \theta)$ is a training process which is customised to correspond with the machine learning technique used (Algorithm 4.6 for MLR and Algorithm 4.7 for ANN and SVR).

The proposed NLIM has been applied in different datasets to demonstrate its capabilities and performance. The datasets include an artificial dataset, SUMO dataset, WebTRIS dataset and FCD dataset.

Algorithm 4.4 Pseudo-code of Neighbouring Links Inference Method

function NLIM($Dlink$, ϵ, k, $\gamma_{threshold}$) ▷ Where $Dlink$ is a target link
  $L_O \leftarrow Dlink$
  $M \leftarrow \emptyset$
  for each $L_M^{L_O} \subset L_N^{L_O}$ **do**
    $M \leftarrow \{L_M^{L_O}, L_O\}$
  end for
  for each $m \in M$ **do**
    $T_m \leftarrow REMOVEBLANK(S_m)$
    if $size(T_m) > \gamma_{threshold}$ **then**
      $T_m \leftarrow normarlise(T_m)$ ▷ using Equation 4.10
      DR-M-GMM(T_m, ϵ, k) ▷ using Algorithm 4.3
      $n \leftarrow 1000$
      $\Theta \leftarrow$ Hyper-parameter
      $\theta_{bets} \leftarrow \emptyset$
      $(T_m^{in}, T_m^{out}) \leftarrow T_m$
      GRIDSEARCH($T_m^{in}, T_m^{out}, n, \Theta$)
      LEARNING($T_m^{in}, T_m^{out}, \theta_{best}$)
      $pIndex \leftarrow$Calculate performance matrix on unseen data
      Rank m based on $pIndex$
    end if
  end for
end function

4.5 Similar model searching

In this section, a novel similar model searching (SMS) methodology for NLIM is proposed to deal with high data sparsity and irregularity of links. The framework of NLIM and SMS is shown in Figure 4.4 where NLIM is shown as light blocks while the shaded blocks represent the SMS. In Figure 4.4, $\mathfrak{C}_{NLIM}$ is the collection NLIM and corresponding errors $\mathfrak{C}_E$, $\mathfrak{C}_{PS}$ is a collection of similar potential models, $\mathfrak{C}_{PE}$ is a collection of corresponding error.

The main idea of SMS is to discover a list of traffic link models which has the similarity with a target traffic link model. The target model $\{L_O, L_N\}$ is similar to a model $\{\bar{L}_O, \bar{L}_N\}$ if they satisfy two conditions:

1. The number of L_N is equal the number of $\bar{L}_N$.
2. The relationship between L_O and links in L_N is similar to the relationship between $\bar{L}_O$ and links in $\bar{L}_N$.

The condition 1 is simple to confirm while the condition 2 needs to use NLIM model of $\{L_O, L_N\}$ to examine the model of $\{\bar{L}_O, \bar{L}_N\}$. Furthermore, *Error* of $\{\bar{L}_O, \bar{L}_N\}$ tested

on test dataset of NLIM of $\{L_O, L_N\}$ is less than or equal to $Error$ of NLIM of $\{L_O, L_N\}$ tested on test dataset of $\{L_O, L_N\}$. The training dataset of similarity models can be used as training dataset in the training process of the target model.

It is to be noted that the link length of a link in the traffic link model, the shape of the traffic link layout and the shape of the traffic link model are not directly considered as conditions in similar model searching because they are already included in the link relationships. The steps of SMS are presented in Algorithm 4.5.

As mentioned in Section 4.4 and mentioned in Department of Transport (2016), the number of data samples in each model is not identical. Most motorway, trunk and primary links may have a large amount of travel time data that can be used for training and testing while A links, B links, and minor links may have a lower number of data samples. Consequently, the performance of the models might be affected. The proposed SMS methodology can be applied to address the insufficient number of data samples for minor roads.

SMS is not a stand-alone methodology. NLIM must be used before on a traffic network. SMS does not depend on a machine learning technique within the NLIM. However, the searching process should be applied to the same machine learning technique in both target traffic model and the similarity model in a large number of NLIM models. Different machine learning techniques could differently generalise relationships between links in traffic link models. Consequently, performance matrices of those traffic models could be slightly different. The condition 2 could depend on used machine learning techniques.

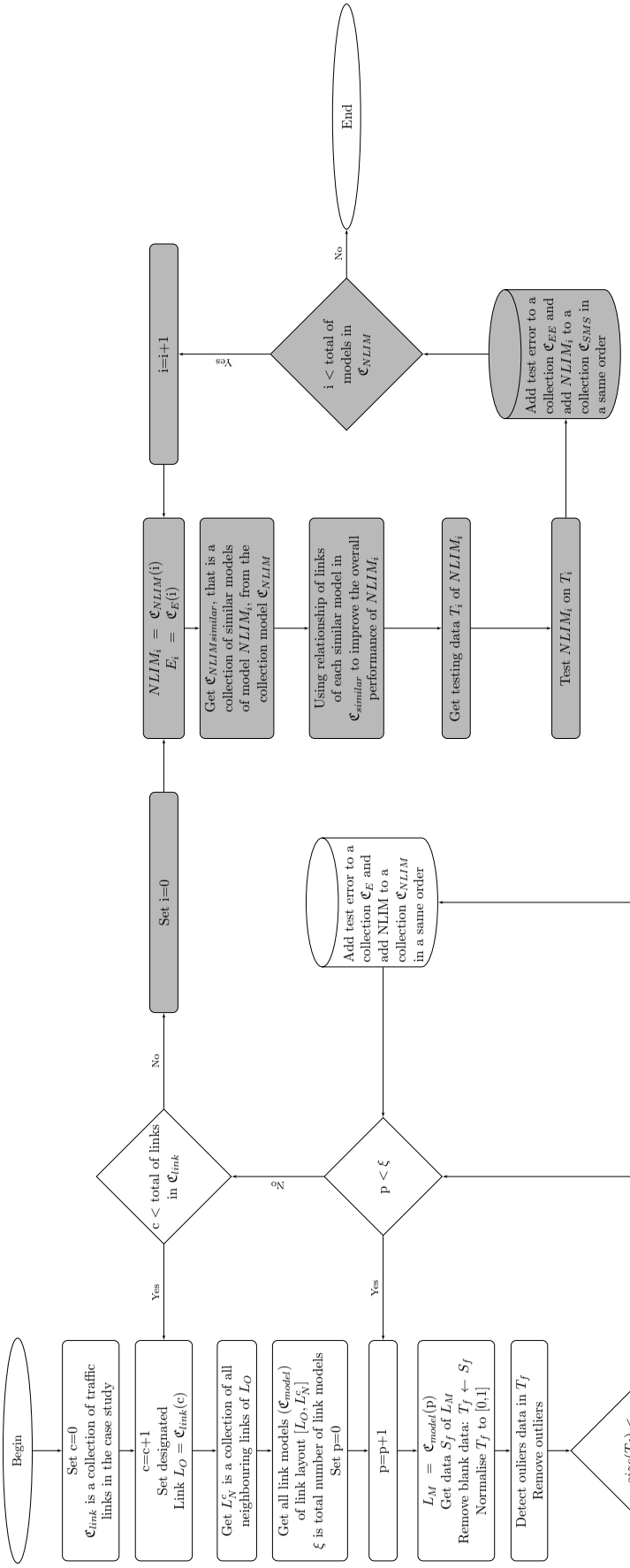


FIGURE 4.4: Diagram of Neighbouring Link Inference Method with Similar Models Searching (SMS)

Algorithm 4.5 Pseudo-code of Similar Model Searching (SMS)

function SMS($\mathfrak{C}_{NLIM}$, $\mathfrak{C}_E$)
  $\tau \leftarrow$ number of models in $\mathfrak{C}_{NLIM}$
  for i=1 **to** τ **do**
    do
      $NLIM_i \leftarrow \mathfrak{C}_{NLIM}(i)$
      $I_i \leftarrow$ number of $NLIM_i$'s inputs
      $T_i \leftarrow$ training data of $NLIM_i$
      $T_i \leftarrow normarlise(T_i)$
      $T_i \leftarrow$ DR-M-GMM(T_i, ϵ, k)
      $T_i' \leftarrow$ testing data of $NLIM_i$
      $Error_i \leftarrow \mathfrak{C}_E(i)$
      for j=0 **to** τ **do**
        $NLIM_j \leftarrow \mathfrak{C}_{NLIM}(j)$
        $Errror_j \leftarrow \mathfrak{C}_E(j)$
        $I_j \leftarrow$ number of $NLIM_j$'s inputs
        if i $\neq$ j **and** $I_i = I_j$ **then**
          Insert $NLIM_j$ into $\mathfrak{C}_{PS}$
          Insert $Error_j$ into $\mathfrak{C}_{PE}$
        end if
      end for
      $\bar{T}_i \leftarrow T_i$
      Sort $\mathfrak{C}_{PS}$ in descending order based on $\mathfrak{C}_{PE}$
      $Sk_{PS} = 0$
      for each $NLIM_{PS}$ **in** $\mathfrak{C}_{PS}$ **do**
        $Error_{PS} \leftarrow$ Error of $NLIM_{PS}$ on T_i'
        if $Error_{PSimilar} \leq Error_i$ **then**
          $T_{PS} \leftarrow$ training data of $NLIM_{PS}$
          $T_{PS} \leftarrow normarlise(T_{PS})$
          $T_{PS} \leftarrow$ DR-M-GMM(D_{PS}, ϵ, k)
          $\bar{T}_i \leftarrow T_i + T_{PS}$
          $(\bar{T}_i^{in}, \bar{T}_i^{out}) \leftarrow \bar{T}_i$
          $n \leftarrow 1000$ ▷ the number of labelled data for hyper-parameter searching
          $\Theta \leftarrow \emptyset$
          $GRIDSEARCH(\bar{T}_i^{in}, \bar{T}_i^{out}, n, \Theta)$
          $NLIM_i' \leftarrow$ LEARNING$(\bar{T}_i^{in}, \bar{T}_i^{out}, \theta_{best})$
          $Error_i'$ of $NLIM_i'$ on T_i'
          if $Error_i' \leq Error_i$ **then**
            Insert $NLIM_i'$ into $\mathfrak{C}_{SMS}$
            Insert $Error_i'$ into $\mathfrak{C}_{EE}$
            $T_i \leftarrow \bar{T}_i$
            $Sk_{PS} \leftarrow 0$
          else
            $\bar{T}_i \leftarrow T_i$
            $Sk_{PS} \leftarrow Sk_{PS} + 1$
          end if
        end if
      end for
    while $Sk_{PS} \geq 3$
  end for
end function

The existing link layouts in the FCD dataset consists of a large number of traffic link models. NLIM training produces a large number of models (approximately 340000 traffic link models). A target link in the collection has a combination of relationships with neighbouring links (NLIM models). The smallest size of the NLIM model consists of the target link and an adjacent link. The largest size of the NLIM model contains all the neighbouring links. The diversity of models with varying relationships between traffic links offers a possibility of having a number of similar potential models in the collection of NLIM models.

The SMS looks for similar models, which consist of all travel time models trained by NLIM. After similar models are found, the SMS does a further step to check if training data of the potential similar models can be adapted to enhance the performance of the selected NLIM model. The proposed method is described in Algorithm 4.5. By using SMS, the NLIM model does not only utilise data of its link model but also of other similar link models in the traffic network. This effectively strengthen the relationship between links of the considered link model.

SMS methodology requires a diversity of NLIM models. The number of traffic link layout in a case study should be large enough. To demonstrate the ability of the proposed method, SMS is applied in FCD dataset. The evaluation, discussion of the experiment results are presented in Chapter 5.

4.6 Machine learning techniques employed in NLIM

In NLIM, machine learning techniques play a critical role in modelling the relationship between links. The primary purpose of using different machine learning techniques is to study the effectiveness of the individual machine learning technique in terms of learning the relationship between a target link and its neighbouring links on the dataset provided. Three performance matrices are used in this thesis to evaluate those machine learning techniques. They include Mean Square Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

As mentioned in the literature review, three machine learning techniques are utilised including multi-linear regressions, neural networks and support vector machines (SVM). Neural networks used in this thesis are feed-forward neural networks with evolution learning and resilient back-propagation training algorithm. The regression SVM (SVR) with linear and non-linear kernel has also been used and studied. The training process of individual machine learning technique is separately designed due to different stop criteria and different pre-defined hyper-parameters optimisations of the corresponding machine learning technique.

4.6.1 Multi-linear regression

Multi-linear regression (MLR) is a standard machine learning technique. The population regression line is described in Equation 3.5, and the regression line is defined in Equation 3.6. MLR does not require a hyper-parameter ($\Theta = \emptyset$) to model the relationship between variables and a response variable. The grid search for hyper-parameter optimisation is omitted while MLR is applied in NLIM.

The training process for multi-linear regression is a traditional process of regression line generation based on Equation 3.6 and Equation 3.7 in Section 3.2. The training function is defined as below:

4.6.2 Feed-forward evolution learning neural network

Feed-forward evolution learning neural network (FF-EL-ANN) is a supervised learning neural network where parameters, topology and rules of the feed-forward neural

Algorithm 4.6 Pseudo-code of learning algorithm for MLR

function LEARNING(T^{in}, T^{out}, θ) ▷ θ is not used in learning funtion of multi-linear regression
  $T^{in} = \{t_{i1}, t_{i2}, ..., t_{in}\}$ ▷ $i \in \{1, 2, ..., k\}$
  where i ∈ 1..k, k is number of rows in T^{in}, n is number of columns in T^{in}
  $T^{out} = \{o_1, o_2, ..., o_n\}$
  for i=1 to k **do**
    Compute β_i using $o_j = \sum \beta_i t_{ij}$, $j \in \{1, 2, ..., n\}$
  end for
  $\beta \leftarrow \{\beta_1, \beta_2, ..., \beta_k\}$
end function

network are generated using an evolutionary algorithm. The evolutionary algorithm is a genetic algorithm (GA). In this thesis, a fully connected feed-forward neural network (FF-ANN) is utilised. FF-ANNs need three parameters: the number of layers, the number of neurons per layer and an activation function. Every neuron and connection in the FF-ANN is specified directly and explicitly in the genotype of the GA. The population of genotypes is directly mapped to the population of neural network phenotypes. The fitness function is the performance of a neural network phenotype. The mutation probability and the crossover probability are fixed to 0.1 and 0.24 following a series of manual tuning and experiments. Population size is a pre-defined parameter of GA. The genetic algorithm is used to optimise FF-ANNs parameters. Therefore, hyper-parameter includes the number of layers, number of neurons per layer, activation function and population size of GA. Grid searching for hyper-parameter optimisation is applied to identify appropriate hyper-parameters which help to improve the performance of FF-EL-ANN.

The training function for FF-EL-ANN is described as in Algorithm 4.7.

Algorithm 4.7 Pseudo-code of learning algorithm for FF-EL-ANN, FF-RPROP-ANN and SVR

function LEARNING(T^{in}, T^{out}, θ)
  $S = \{(t_1^{in}, t_1^{out}), (t_2^{in}, t_2^{out}), ..., (t_m^{in}, t_m^{out})\}$ ▷ $t_i^{in} \in T^{in}$, $t_i^{out} \in T^{out}$ respectively
  $k \leftarrow 5$ ▷ Number of k-folds is fixed to 5
  partition S into $S_1, S_2, ..., S_k$
  $A \leftarrow$ FF-EL-ANN (or $A \leftarrow$ FF-RPROP-ANN, or $A \leftarrow$ SVR)
  for p=1 to k **do**
    $h_{i,\theta} = A(S \setminus S_i, \theta)$ ▷ limit under-fitting and over-fitting are applied
    $error_i \leftarrow L_{S_i}(h_i, \theta)$
  end for
  $best \leftarrow argmin_i[error_i]$
  $h_{best} = A(S \setminus S_{best}, \theta)$
  save h_{best}
end function

4.6.3 Feed-forward resilient back-propagation neural network

Feed-forward resilient back-propagation neural network (FF-RPROP-ANN) is a supervised learning neural network. The resilient back-propagation learning algorithm is a learning heuristic. The important parameter for RPROP is the learning rate. RPROP is also used to train a full connected FF-ANN. The hyper-parameter of FF-RPROP-ANN includes the number of layers, number of neurons per layer, activation function and learning rate. The grid search discovers the appropriate hyper-parameter of FF-RPROP-ANN. The training function for FF-RPROP-ANN is described in Algorithm 4.7.

4.6.4 Support vector machine regression

Support vector machine regression (SVR) can be set up with one of the kernels in Equation 3.23. In this thesis, the linear and polynomial kernels are chosen. SVR combined with the two kernels are two different machine learning techniques in NLIM. They represent a linear and a non-linear support vector machine. The linear kernel does not require any hyper-parameter hence the grid searching is omitted. In contrast, polynomial kernel requires three parameters. They are degree (d), slope (γ) and complexity (C). The grid search is conducted to find an appropriate hyper-parameter for the SVR. The training function for SVR is described in Algorithm 4.7.

4.7 Experiment data

The datasets introduced in this section will be used to study the relationship between parameters of neighbouring traffic links as well as demonstrate the performance of NLIM employing different machine learning techniques. Because traffic networks are complex and dynamic, the travel time from different traffic networks can vary significantly.

4.7.1 Artificial data

Artificial data is generated from a model rather than gathered from real-world events. Another name for artificial data is synthetic data. It is created algorithmically, and it is

used as a well defined test environment to confirm correct operation of techniques and systems.

Methodology of generating artificial data

To create an artificial travel time dataset, transport models are applied. A travel time function of a traffic link (Equation 4.13) was developed by US Bureau of Public Roads (BPR) in 1964. It described the relationship between traffic flows and travel time. The BPR formula uses flow to estimate travel time, observed flows and link capacity to represent different relationships of various types of traffic links, de Dios Ortúzar and Willumsen (2011), Londoño and Lozano (2012). The BPR function is used in the thesis as it has the advantages of relating to junction delay modelling. The BPR formula is utilised to model travel time in a traffic link layout. Figure 4.5 shows an example of the relationship between travel time and traffic flows produced by the BPR formula in a traffic link in a motorway. The BPR function is defined in Equation 4.13.

$$t = t_0[1 + \alpha(\frac{x + \gamma * x'}{k})^{\beta}] \tag{4.13}$$

where t is the travel time (seconds per mile), x is the demand traffic flow rate (vehicles per hour) on the link, and x' refers to the traffic flow rate (vehicles per hour) on the opposite direction of the link. x' is only used if the link has no distinct lanes for the opposite direction. t_0 is the free-flow travel time (seconds per mile) on the link, and parameter k describes the link capacity (vehicles per hour). α and β represent the traffic/delay parameters of a specific link type.

Suggested values of α and β for the formula parameters in Equation 4.13 vary. They depend on characteristics of the traffic network which is being modelled. The common values for α and β are 0.15 and 4.0 respectively Mathew (2018). Zhao and Kockelman (2011) have suggested higher values of α and β, 0.84 and 5.5 respectively. Observations of the Danish road system in the work of Nielsen and Jorgensen (2008) shows that the values for α are between 0.8 and 1.2 and values for β are between 1.5 and 4.0. They showed that larger roads yielded higher values for β.

Consider a traffic link layout with eight junctions and six neighbouring links as shown in Figure 4.1(b). The artificial data is produced based on the traffic link layout and the

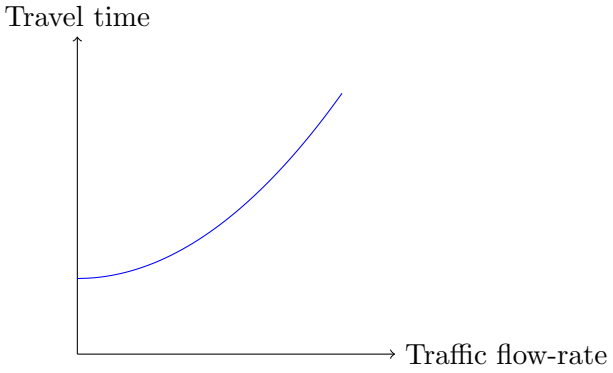


FIGURE 4.5: Traffic travel time and traffic flow relationship

TABLE 4.1: Constants for links in the traffic link layout shown in Figure 4.1(b)

	t_0	α	k	β
AD	100	0.15	1500	4.0
BD	500	0.15	1100	4.0
CD	200	0.15	1000	4.0
DE	250	0.15	2000	4.0
EF	270	0.15	1000	4.0
EG	300	0.15	1100	4.0
EH	120	0.15	1300	4.0

Equation 4.13. All links in the link layout belong to major link type. According to the BPR formula setting of Danish network in Nielsen and Jorgensen (2008), the α and β are set to 0.15 and the 4.0; x' is set to 0 due to objective of modelling the major links in this section, t_0, α, k and β parameters of each traffic link in the traffic link layout are assigned with different values as shown in Table 4.1.

Based on the Danish network, the traffic link layout contains observed links' traffic flow rate x by a time interval of 15 minutes ($numS = 60 * 24/15 = 96$). Traffic flow x of each traffic link is produced. The random re-sampling method used in this thesis relies on a method which has been introduced and applied in the works of Petrik et al. (2016) and Matas et al. (2012). The re-sampling techniques do not require any tuning of the parameter distributions configuration. x is individually produced by random draw with uniform probabilities.

The travel time of links is generated using Equation 4.13. Define t_{AD}, t_{BD}, ..., t_{EG} are travel time of link AD, BD, ..., EG which are respectively produced based on traffic flow rates x_{AD}, x_{BD}, ...,x_{EG}.

Algorithm 4.8 Pseudo-code of artificial data generation algorithm

```
1: function ArtificialDataG(p, numS, dateSet, L_O, L_N, DLSet)
2:     α ← 0.15
3:     β ← 4.0
4:     for each date ∈ dateSet do
5:         for slot = 0 to numS do
6:             link ← L_O
7:             Set values of t_0 and k of link
8:             vehCL ← randN(1, 9)
9:             x_link ← randR(0, k)
10:            t_link = t_0[1 + α(x_link/k)^β]
11:            vehCL_O ← vehCL)
12:            x_O ← x_link
13:            if p ≤ randR(0, 1) then
14:                record ← {link, slot, dayofweek(date), vehCl, t_link}
15:                Save(record)
16:            end if
17:            for each link ∈ L_N do
18:                Set values of t_0 and k of link
19:                if link ∈ DLSet then
20:                    vehCL ← vehCl_O
21:                    x_link ← x_O
22:                else
23:                    vehCL ← randN(1, 9)
24:                    x_link ← randR(0, k)
25:                end if
26:                t_link = t_0[1 + α(x_link/k)^β]
27:                if p ≤ randR(0, 1) then
28:                    record ← {link, slot, dayofweek(date), vehCl, t_link}
29:                    Save(record)
30:                end if
31:            end for
32:        end for
33:    end for
34: end function
```

The artificial dataset in this thesis is generated from the model which simulates the moving observers. The dataset is made to be sparse (for many periods of time, in a particular link, there should not be any data available, Jones et al. (2013)). To simulate the behaviour of the moving observers, a probability of having travel time data is set to 0.2 (p=20%) for any particularity time of day.

A link in a traffic link layout is an independent link if traffic condition on the link does not affect the target link and vice versa. Define $setDependentLink$ is a set of dependent links of the target link (L_O). Define $DLSet_F$ and $DLSet_R$ are a set of the front dependent links ($DLSet_F \subset L_{NF}$) and the rear dependent links ($DLSET_R \subset L_{NR}$) respectively ($DLSet_F \cup DLSet_R \equiv DLSet$) of a target link. The flow rate is used in this research to generate the artificial data that can represent the relationship between the dependent

links and the target link. Assume that the most of the vehicle getting out of $DLSet_R$ will pass through the target link and most of the vehicle getting out of the target link will pass through the $DLSet_F$.

TABLE 4.2: Statistics of the artificial data, the unit of measure for travel time is seconds per mile, light grey rows indicate the dependent links of link DE.

Link name	Number of samples	Minimum Travel Time	Mean Travel Time	Maximum Travel Time
AD	6977	100	102.89	114.96
BD	7084	500	551.69	758.64
CD	7058	200	230.86	351.47
DE	35040	250	252.37	261.83
EF	7084	270	310.48	474.48
EG	7009	300	331.90	455.18
EH	7031	120	126.50	151.82

The artificial data set for the traffic link layout that is shown in Figure 4.1(b) is generated based on the Algorithm 4.8, where *dateSet* is a set of dates, $randR(lowerbound, upperbound)$ is a uniform random function that can produce a real-value random number between lower bound and upper bound, $randN(lowerbound, upperbound)$ is a uniform random function that can generate an integer-value number between a lower bound and upper bound, *vheCl* is a numeric representation of a vehicle class and $dayofweek(date)$ is a function that can convert date to a numeric representation of the day of the week.

Parameters for the algorithm are set-up as discussed in Section 4.4: p is set to 0.2; *numSlot* is set to 96; *dateSet* is set to 365 days, $L_{NR} = \{AD, BD, CD\}$, $L_{NF} = \{EF, EG, EH\}$, $L_O = \{DE\}$ and $DLSet = \{BD, EG\}$. The artificial data sparsity is 65.78%. Other parameters are shown in Table 4.1.

Table 4.2 shows the statistics of the artificial travel time dataset including the number of samples, minimum, mean and maximum travel time of links corresponding to the setup parameters.

An example of travel time data matrix produced by the Algorithm 2 for a full traffic link model with $numS = 10$, $dateSet = \{03{-}07{-}2018\}$, $L_N = \{AD, BD, CD, EF, EG, EH\}$

and $setDependentLink = \emptyset$:

$$S = S_f = \begin{bmatrix} 1 & 0 & 0 & \_ & \_ & \_ & \dots & \_ & 300 & \_ \\ 1 & 1 & \_ & \_ & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 2 & 0 & 105 & 500 & \_ & \_ & 270 & \_ & \_ \\ 1 & 3 & 0 & \_ & \_ & \_ & \_ & \_ & \_ & 122 \\ 1 & 4 & 0 & 105 & 520 & 200 & 250 & 279 & 310 & 123 \\ 1 & 5 & 0 & \_ & \_ & 200 & \_ & \_ & \_ & \_ \\ 1 & 6 & 0 & \_ & \_ & \_ & 253 & \_ & \_ & \_ \\ 1 & 7 & \_ & \_ & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 8 & \_ & \_ & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 9 & \_ & \_ & \_ & \_ & \_ & \_ & \_ & \_ \end{bmatrix} \tag{4.14}$$

$$S_f^{in} = \begin{bmatrix} 1 & 0 & 0 & \_ & \_ & \_ & \_ & 300 & \_ \\ 1 & 1 & \_ & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 2 & 0 & 105 & 500 & \_ & 270 & \_ & \_ \\ 1 & 3 & 0 & \_ & \_ & \_ & \_ & \_ & 122 \\ 1 & 4 & 0 & 105 & 520 & 200 & 279 & 310 & 123 \\ 1 & 5 & 0 & \_ & \_ & 200 & \_ & \_ & \_ \\ 1 & 6 & 0 & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 7 & \_ & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 8 & \_ & \_ & \_ & \_ & \_ & \_ & \_ \\ 1 & 9 & \_ & \_ & \_ & \_ & \_ & \_ & \_ \end{bmatrix} \tag{4.15}$$

$$S_f^{out} = \begin{bmatrix} \_ \\ \_ \\ \_ \\ \_ \\ 250 \\ \_ \\ 253 \\ \_ \\ \_ \\ \_ \end{bmatrix} \tag{4.16}$$

There are nine vehicle classes which pass through the traffic link layout. Different vehicle classes produce somewhat different travel times Department of Transport (2016). A day is divided into a $numS$ time slot.

As can be seen in Matrix 4.14, 4.15 and 4.16, the artificial data is manufactured with hight data sparsity. The sparsity of matrix 4.14 calculated using Equation 4.6 is 60%. If the day of a week and the time of the day are not counted in the sparsity calculation as they are always available, the actual data sparsity for the data in Matrix 4.14 is calculated as below:

$$R = 100\% \frac{60}{100 - 20} = 100\% \frac{60}{80} = 75\% \tag{4.17}$$

For the target link DE (Matrix 4.14 and 4.16), there are two data samples over the ten intervals of time. Therefore, the sparsity of the link DE (R_{DE}) is 80%. In practice, travel time data collected by moving observes is having similar sparsity to the artificial data, Department of Transport (2012), Jones et al. (2013), Tang et al. (2018). The high sparsity of data is a challenge for any methodology that uses the data.

4.7.2 SUMO data

SUMO stands for Simulation of Urban Mobility. SUMO is a popular open-source for traffic simulation. It is partially developed by the Institute of Transportation Systems (German Aerospace Centre). It is a portable, microscopic road traffic simulator, and it is being designed to simulate large road networks, Krajzewicz et al. (2012). SUMO has been used in many studies such as Bedi et al. (2015), Bedogni et al. (2015), Behrisch and Weber (2014), Daniel Krajzewicz and Bieker (2012), Nguyen et al. (2016) to simulate traffic networks. SUMO can simulate vehicles in detail .i.e. departure and arrival time, lanes to use, velocity and positions. SUMO can often provide a large number of different tests. SUMO uses XML file format to store values of all these measurement. SUMO also offers an API to control simulations by using a socket connection, Daniel Krajzewicz and Bieker (2018).

Traffic network

An entire day traffic simulation scenario is utilised in this thesis. The scenario is for "TAPAS Cologne" which represents the traffic network within Cologne city, Germany. The scenario was initially built according to the demand from TAPAS which computes mobility demand for an area. The data for the scenario was from a closed-source road network. It was mapped onto a map that is produced by the OpenStreetMap project (Figure 4.6).

(a) The TAPAS Cologne Scenario. (b) Case study area.

FIGURE 4.6: The TAPAS Cologne traffic network in SUMO, Daniel Krajzewicz and Bieker (2018).

Data format

The output of a SUMO simulation is XML format files. Several types of traffic information of vehicle are included in the output files such as route travel time, traffic delay, traffic flow, traffic density etc. The travel time data can be reconstructed from route travel time based on the departure time, the arrival time of a route and the exit time of vehicle in a traffic link.

For example, a vehicle which vehicle's id is 65 in Figure 4.7, the travel time of link 48966314#0, 48966314#1 and 18123776 are 16.00 (16.00-0), 7.00 (23.00-16.00), 10.00 (33.00-23.00) respectively. The travel time of other links is precisely reconstructed using this method.

```
<?xml version="1.0" encoding="UTF-8"?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/routes_file.xsd">
    <vehicle id="176" type="1" depart="6.00" arrival="22.00">
        <route edges="-18619294" exitTimes="22.00"/>
    </vehicle>

    <vehicle id="65" type="1" depart="0.00" arrival="33.00">
        <route edges="48966314#0 48966314#1 18123776" exitTimes="16.00 23.00 33.00"/>
    </vehicle>

    <vehicle id="150" type="1" depart="6.00" arrival="35.00">
        <route edges="-18619253#2 -18619253#1 -18619253#0 18619253#0" exitTimes="21.00 27.00 32.00 35.00"/>
    </vehicle>

    <vehicle id="126" type="1" depart="16.00" arrival="37.00">
        <route edges="48966314#2" exitTimes="37.00"/>
    </vehicle>

    <vehicle id="15" type="0" depart="0.00" arrival="38.00">
        <route edges="-18619244 -48966314#0" exitTimes="29.00 38.00"/>
    </vehicle>

    <vehicle id="120" type="0" depart="0.00" arrival="44.00">
        <route edges="52116125#4 52116127 -48822786 -48898791" exitTimes="8.00 10.00 16.00 44.00"/>
    </vehicle>
```

FIGURE 4.7: The XML output of a SUMO simulation.

Reconstruct link layout from SUMO map

Neighbouring links of a target link a reconstructed from a SUMO route file. An example of a route file is illustrated in the Figure 4.8 as below:

```
<route id="26" edges="-10688726#1 -10688726#0 10688725#0 10688725#1 51193195 51189369 4726947#0 -48966314#3 -48966314#2 48966314#2" />
<route id="27" edges="48966314#0 48966314#1" />
<route id="28" edges="48823044 -48823044 -48823045 -52116129 29461984#0 29461984#1 52116117 52116125#4 52116127 -48822786 -48898791 48898791" />
<route id="30" edges="-48966314#2 18123776 4726957#0 4726957#1" />
<route id="32" edges="-18619253#2 -18619253#1 -18619252#0 -18619248#1 18619248#1" />
```

FIGURE 4.8: The route file format of SUMO.

For each link in the routes, it's front and rear links have to be determined. The set of a target link and its front and rear links is a link layout. For example in the route id 26 in Figure 4.8, if -10688726#0 is a target link, it has two neighbouring links: -10688726#1 and 10688725#0. The former is the rear link, and the later is a front link. Replicating the process in route id 26 for all other routes in the route file, a set of -10688726#0 s' neighbouring links is determined.

There are 30947 target links and 30947 traffic link layout accordingly in the case study area. The links in the case study area belong to different link categories in practice. However, the TAPAS Cologne traffic network in SUMO does not specify those categories. So in the dataset produced by SUMO simulator on this area, all the links are treated as only one type.

4.7.3 WebTRIS data

Highways England is a government-owned company. The company is in charge of operating, maintaining and improving motorways and A roads in England. WebTRIS is a web portal owned by Highways England which publishes traffic information of motorway and A roads in 15 minutes periods since 2015. The traffic information includes the average velocity, the average flow, traffic volume etc., Highways-England (2018).

Travel times and speeds are gathered using a combination of sources. They include Automatic Number Plate Recognition (ANPR) cameras, in-vehicle GNSS and inductive loops built into the road surface, Highways-England (2016).

Traffic network

The study area used in this work is located in the East Midlands in the UK. The traffic network contains a total of 74 motorway links and 41 A links including M6, M45, A5, A14 and A428 The sensors' locations are indicated by a blue and yellow circles in Figure 4.9.

WebTRIS provides both a direct download from the website as well as provides an API for retrieving a list of sites' id. The travel time data can be gathered from a site or multiple sites based on either batch download functionality or the sites' id using the API.

Data format

Figure 4.10 shows the dataset format which is retrieved using the API. The data format includes the site name, the report date, the interval of time ending, the statistics of the vehicle size, the average velocity and the total volume of the link where the site is located.

The Euclidean distance between two sites can be calculated based on the GNSS reference of the sites. The travel time of the link is calculated using Equation 4.18.

$$\bar{t} = \frac{3600 * \bar{v}}{d} \tag{4.18}$$

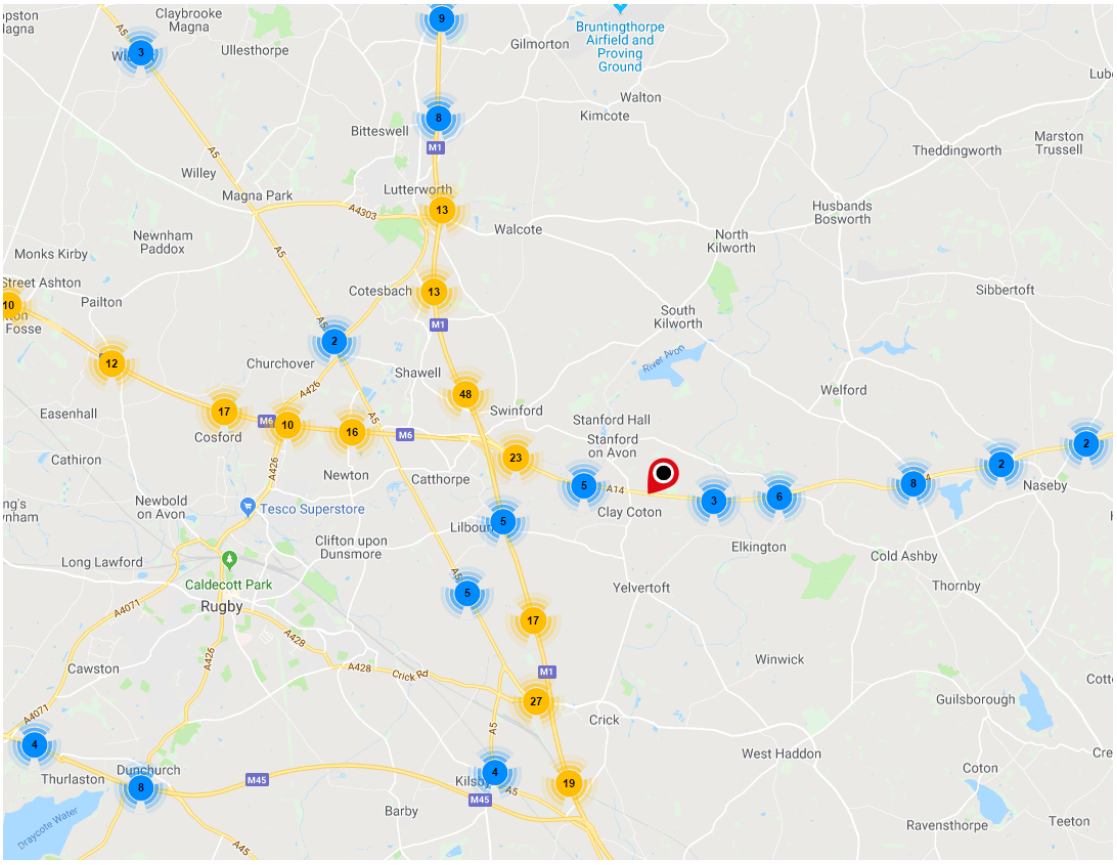


FIGURE 4.9: The experiment area in the East Midland, England from WebTRIS, Highways-England (2018).

Site Name	Report Date	Time Period Ending	Time Interval	0 - 520 cm	521 - 660 cm	661 - 1160 cm	1160+ cm	0 - 10 mph	11 - 15 mph	16 - 20 mph	21 - 25 mph	26 - 30 mph	31 - 35 mph	36 - 40 mph	41 - 45 mph	46 - 50 mph	51 - 55 mph	56 - 60 mph	61 - 70 mph	71 - 80 mph	80+ mph	Avg mph	Total Volume
M1/3339B	09/03/2018	00:14:00	0	34	4	14	58															59	110
M1/3339B	09/03/2018	00:29:00	1	33	5	15	50															58	103
M1/3339B	09/03/2018	00:44:00	2	37	7	16	50															59	110
M1/3339B	09/03/2018	00:59:00	3	42	5	10	42															61	99
M1/3339B	09/03/2018	01:14:00	4	23	6	13	42															57	84
M1/3339B	09/03/2018	01:29:00	5	23	9	13	41															58	86
M1/3339B	09/03/2018	01:44:00	6	28	7	11	59															59	105
M1/3339B	09/03/2018	01:59:00	7	22	5	9	50															58	86
M1/3339B	09/03/2018	02:14:00	8	21	2	18	43															58	84
M1/3339B	09/03/2018	02:29:00	9	24	5	10	41															58	80
M1/3339B	09/03/2018	02:44:00	10	18	8	7	47															56	80

FIGURE 4.10: WebTRIS Data Format.

where $\bar{t}$ (sec) is the average travel time of vehicles on the link, $\bar{v}$ (mph) is the average velocity of vehicles, d (miles) is the distance of the traffic link which is GNSS Euclidean distance between two sites, Cai et al. (2011).

$$d \approx R\sqrt{(lat_1 - lat_2)^2 + \cos^2(\frac{lat_1 + lat_2}{2})(lon_1 - lon_2)^2} \tag{4.19}$$

where R is Earth's radius ($R \approx 3958.756$ miles), lat and lon are latitude and longitude respectively.

4.7.4 Floating car data

Teletrac (formerly Trafficmaster) is a US software company with offices in the United Kingdom. They provide a cloud-based GNSS tracking software for fleet tracking. More than 250,000 vehicles in more than 87 countries have provided tracking information using their software teletracnavman (2018).

According to Wright (1973), a floating car was a concept used to obtain traffic flow and journey time. Since the 2000s, a Floating car is any car from which GPS positions are continually recorded via in-car equipment, smartphones, etc., de Fabritiis et al. (2008), Derrmann et al. (2016), Jones et al. (2013), Leodolter et al. (2015), Pan et al. (2011), Protschky, Feit and Linnhoff-Popien (2015), Protschky, Ruhhammer and Feit (2015), Rahmani et al. (2013, 2014), Wang et al. (2012). Floating Car Data (FCD) used in this research refers to travel time data which is gathered from GNSS tracking of floating cars by TrafficMaster.

Traffic network

The travel time data was collected from September 2009 to February 2012 in Leicestershire, UK. The dataset used in this thesis comprises travel times for 22053 traffic links. They include 67 motorway links, 22 trunk links, 911 primary link, 1457 A links, 843 B link and 13752 minor links (Table 4.3). In 13752 minor links, 5226 links have data sparsity less than or equal to 99%. They account for 38% of the total minor links. The traffic network is shown in Figure 4.11.

TABLE 4.3: Number of links are included in the experiment

Link type	Number of link
Motorway	67
Trunk	22
Primary	911
A	1457
B	843
Minor (data sparsity $\leq$ 99%)	5226
Minor (data saprsity $>$ 99%)	8526
	Total: 22053

(a) The Leicestershire traffic map. (b) Case study area.

FIGURE 4.11: The Leicestershire map vs case study area.

Data format

The raw data collected from FCD contain reconstructed link travel times at 15 minutes intervals. Thereby, a day starting from 00h00 to 23h59, was divided equally into 96 time slots. For example, time slot 34 covers the period from 08:30:00 – 08:44:59. The format of the FCD dataset is described in Table 4.4, 4.5 and 4.6.

TABLE 4.4: FCD data format

Variable name	Format	Resolution	Minimum	Maximum	Description
link_id	char(17)	n/a	n/a	n/a	Link id
link_ref	integer	n/a	n/a	n/a	Link reference
date_1	YYYY-MM-DD	1 day	2006-06-01	2020-01-01	Date
time_per	integer	15 minutes	0	95	Time period
data_source	integer	n/a	1	2	Data source
veh_cls	integer	n/a	1	9	Vehicle class
N	integer	n/a	1	99	Number of observations
av_jt	integer	10^{-2} second	1	8640000	Average journey time
sum_sq_jt	integer	10^{-2} sec*sec	0	99999999	Sum of squares of journey times
network	integer	n/a	1	10	INT verion identifie

The travel time data unit is 10^{-2} second. The average travel time in minutes per miles for links in the traffic network is approximately 2.46 (minutes/miles). Total links' length is roughly 14,000 kilometres or 8,700 miles. There are nine vehicle classes which are based on the payload and the size of the vehicle. The vehicle classes are shown in Table 4.5. The travel time data mainly is from TrafficMaster (95%), and a small proportion is from Norwich Union (5%). There are no different format between two travel time data sources.

The dataset is in a CSV file format. A CSV file consists of data for an individual link on a monthly basis. The size of total CSV files for Leicestershire traffic network is approximately 60Gb.

TABLE 4.5: Vehicle category descriptions

Class (veh_cls)	Descriptions
1	Cars
2	LGVs (up to 3500kg)
3	HGVs (up to 3500kg)
4	HGVs (over 7500kg)
5	Buses (including minibuses)
6	Taxis
7	Motorised caravans
8	Other vehicles
9	Unknown

TABLE 4.6: The CSV file format of the Leicestershire traffic map using in FCD dataset

Data field	Format	example	Description
TOID	string	4000000019182789A	ITN identifier and direction of the link
Wayness	integer	1	indicate one-way (1) or two-way link (2)
Name	string	ATTERTON LANE	name of the road that contains the link
Number	string	A444	number of the road that contains the link
DescriptiveGroup	string	Named Road	description of the link categories
DescriptiveTerm	string	Minor Road	link category
ChangeDate	date	2007-09-21	date of the map which the link is added
VersionDate	date	2007-09-21	the version date of the map
VersionNumber	integer	432863	version number of the map
StartX	integer	432863	x coordinate of the beginning of the link
StartY	inteer	297610	y coordinate of the beginning of the link
MidX	integer	433971	x coordinate of the middle of the link
MidY	integer	298258	y coordinate of the middle of the link
EndX	integer	435305	x coordinate of the ending of the link
EndY	integer	298329	y coordinate of the ending of the link
LinkLength	integer	2753	the length of the link in Yards

Table 4.6 shows the CSV format of traffic map in the FCD dataset. The LinkLength ranges from 1 to 3860 yards. The average length of links is 117.8703 yards.

(a) The actual traffic link network, OpenStreetMap contributors (2017).

(b) ITN representation.

FIGURE 4.12: Difference between actual traffic network and ITN traffic network.

Integrated Transport Network (ITN) is used in the FCD dataset. This network

provides a highly accurate representation of roads on the ground. $TOID$ is made up of a combination of link Id in ITN and a direction indicator of the link. Furthermore, 'A' indicates the direction of traffic flow as digitised in the original ITN and 'B' indicates the opposite flow direction.

The $Number$ and $DescriptiveTerm$ data field are used to determine the category of a link. $StartX$, $StartY$, $EndX$ and $EndY$ are used to determine neighbouring links. The Neighbouring links shares ($StartX$,$StartY$) or ($EndX$, $EndY$) of a target link. The traffic link layout is henceforward reproduced from the target link and the adjacent links.

Chapter 5

Experiment results

5.1 Introduction

The travel time from different traffic networks can vary significantly. The Neighbouring Inference Method (NLIM) may perform differently in the different traffic networks. The robustness of the proposed method is demonstrated using distinct travel time datasets, (NLIM) is implemented using two artificial datasets and using two historical datasets gathered from two traffic real-world networks.

The artificial datasets are generated based on the BPR function and the from SUMO simulation of Cologne traffic network. Two real-world datasets include travel times of several major links in the UK East Midland provided by WebTRIS and travel times of all link categories in Leicestershire supplied by TrafficMaster. The experiment results are used to confirm the travel time estimation ability of NLIM on sparse and irregular datasets.

This chapter demonstrates the ability of NLIM to estimate travel time as well as demonstrates the advantages of NLIM compared to historical travel average (HA) and moving average (MA) which are two standard statistics models. Although some other methodologies discussed in the literature review chapter provides techniques for travel time estimation for a sparse and irregular dataset, they are not applicable to this study as it required specific data structures and data constraints meanwhile the techniques introduced in this thesis do not contain such requirements' properties. The datasets in this thesis only provide information about sparse travel times in each link in sparse

time intervals. They do not have the trajectory of vehicles as well as information about cars and drivers that are required in the method in Tang et al. (2018).

Along with NLIM, the ability of SMS on increasing the number of labelled data to enhance the performance of NLIM on TrafficMater dataset is analysed and evaluated in this chapter. As mentioned in the previous chapter, the SMS is a methodology able to discover a list of traffic link models which has the similarities with a target traffic link model in order to deal with the high data sparsity and irregularity in datasets.

The chapter is set out as follows: Section 5.2 describes performances of NLIM on four distinct datasets. Section 5.3 gives results of the SMS methodology applied to the FCD dataset. The performance of NLIM is compared to those of HA and MA methods. Section 5.4 provides an overall summary of the results.

5.2 Neighbouring link inference method

In NLIM, machine learning techniques play a critical role in modelling the relationship between links. The learning effectiveness of machine learning techniques on datasets affects the performance of the NLIM models. Therefore, selecting the best machine learning technique for a dataset is an essential step in the proposed methodology.

NLIM is demonstrated using different machine learning techniques. The primary purpose of using various machine learning techniques is to study the effectiveness of the individual machine learning technique in terms of learning the relationship between a target link and its adjacency links on the dataset provided. Three performance metrics are used in this thesis to evaluate those machine learning techniques. They include RMSE, MAE and MAPE.

Time to train and test are two other essential indicators in this stage to determine which machine learning technique is suitable for NLIM when it is applied to a traffic network where millions of models will be trained and thousands of designated links' travel time will be estimated.

A dataset is divided into a training dataset and a test dataset. The training dataset consists of 60% of the total number of labelled data, and the test dataset includes 40% of the total number of labelled data. The training dataset is used to model traffic

link models in a traffic link layout while the test dataset is applied to evaluate the performance of the model which has been produced by machine learning techniques as unseen data. Machine learning models are always trained on training datasets using 5-folds cross-validation as discussed in the section 3.6.1.

As mentioned in the literature review, three machine learning techniques are utilised including multi-linear regression, neural network and support vector machine (SVM). Neural networks employed in this thesis are the feed-forward neural network with evolution learning and resilient back-propagation training algorithm. The regression SVMs (SVRs) with linear and non-linear kernels have also been used and studied. The training process of individual machine learning is separately designed due to the different stop training criteria and different pre-defined hyper-parameters optimisation of the corresponding machined learning technique and the different pre-defined hyper-parameter optimisation of the corresponding machined learning technique.

5.2.1 Experiment 1: Artificial dataset

In this section, NLIM is demonstrated using different machine learning techniques on an artificial datasets that are produced by the ArtificialDataG (Algorithm 4.8). Parameters for the algorithm are set-up as discussed in Section 4.7.1: p is set to 0.2; $numSlot$ is set to 96; $dateSet$ is set to 365 days, $L_{NR} = \{AD, BD, CD\}$, $L_{NF} = \{EF, EG, EH\}$, $L_O = \{DE\}$ and $DLSet = \{BD, EG\}$. The artificial data sparsity is 65.78%. The setup parameters are as shown in Table 4.1.

The traffic link layout gives 64 traffic link models in total. The artificial dataset of the traffic link layout which has been generated can be used to model 40 of 64 traffic models. 14 traffic models are not eligible to model using NLIM because they have an insufficient number of labelled data according to Equation 4.12.

The performance of NLIM models, as well as DR-M-GMM on the artificial dataset, is demonstrated in Table 5.1. Feed-forward evolution learning artificial neural network (FF-EL-ANN), feed-forward resilient back-propagation neural network (FF-RPROP-ANN), support vector machine regression with the polynomial kernel (SVR-NLK) and support vector machine regression with linear kernel (SVR-LK) are employed in the experiments. The performance metrics are RMSE, MAE and MAPE.

TABLE 5.1: The performance metrics of NLIM models on artificial dataset, different machine learning techniques applied, with and without DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

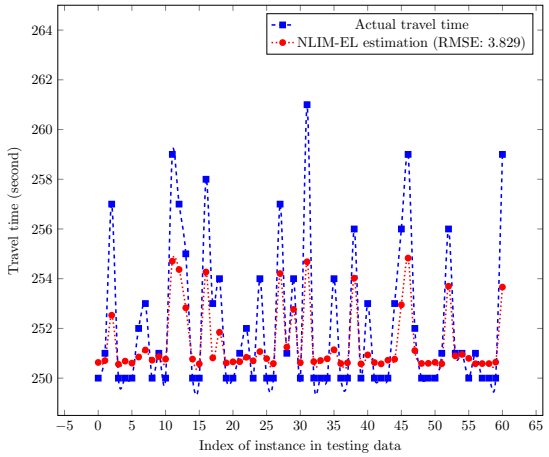
	DR-M-GMM					Original dataset				
Machine learning technique	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
RMSE (ms)										
MLR	0.751	0.938	1.633	3.237	3.766	0.556	1.003	1.626	3.512	3.341
FF-EL-ANN	0.635	0.872	1.53	3.163	3.766	0.516	0.913	1.446	3.139	3.734
FF-RPROP-ANN	0.356	1.044	1.857	3.195	4.04	0.243	0.564	1.4	3.177	3.914
SVR-LK	6.622	9.225	17.415	28.172	55.546	7.393	9.249	18.117	36.434	84.751
MAE (ms)										
MLR	0.459	0.832	1.52	3.625	3.291	0.492	0.852	1.972	3.04	3.479
FF-EL-ANN	0.432	0.722	1.29	2.505	2.821	0.339	0.725	1.239	2.504	2.79
FF-RPROP-ANN	0.197	0.693	1.348	2.422	2.937	0.163	0.321	1.015	2.512	2.859
SVR-LK	5.199	8.669	13.353	20.17	35.679	6.284	8.7	13.741	23.223	52.103
MAPE (%)										
MLR	0.21	1.124	3.035	5.611	7.142	0.938	1.006	2.205	4.783	7.019
FF-EL-ANN	0.17	0.284	0.505	0.986	1.104	0.133	0.286	0.492	0.986	1.091
FF-RPROP-ANN	0.077	0.271	0.526	0.949	1.16	0.064	0.126	0.4	0.989	1.12
SVR-LK	2.048	3.451	5.244	7.902	13.934	2.49	3.464	5.409	9.101	20.313

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

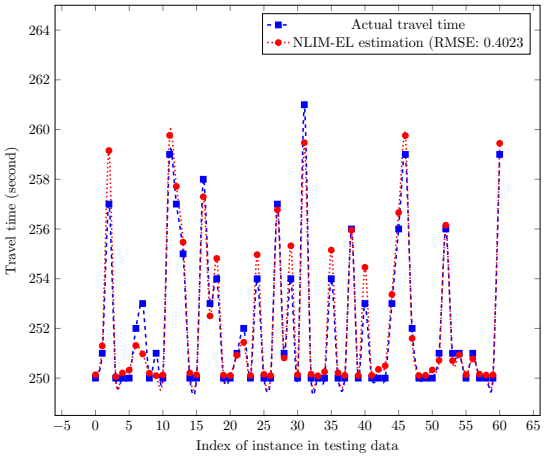
Table 5.1 presents information using a five-number summary of NLIM models' performance in different machine learning techniques using different performance metrics. The five-number summary includes Lower-whisker, Lower-quartile, Median, Upper-quartile and Upper-whisker. The median is the centre value of NLIM models' performance metric, and it gives a brief picture of other values. The five-number summary shows whether the distribution of the performance metric is skewed and whether there are unusual observations in the travel time dataset. The five-number summary is used because it can describe a large number of NLIM models which are included in the experiments. The five number summaries can be compared to another performance metrics as well as it is easy to illustrate using either boxplot or box and whiskers graph.

Based on the experiment results in Table 5.1, SVR-NLK is not able to model 41 traffic link models. It can also be seen that SVR has a poor performance with a linear kernel. The SVR-NLK produces at least 15 times higher errors compared to those using FF-EL-ANN and FF-RPROP-ANN. Especially, SVR-NLK is not suitable for the dataset due to long training times. The experiments reconfirm its behaviour mentioned in the literature. It can be seen that NLIM employed different machine learning techniques has different performances. As can be seen in Table 5.1, DR-M-GMM can enhance the performance of NLIM.

Define NLIM-EL and NLIM-RPROP are NLIM that are employing FF-EL-ANN and FF-RPROP-ANN respectively. Define NLIM-EL-OD and NLIM-RPROP-OD are NLIM-EL and NLIM-RPROP that uses DR-M-GMM. Figure 5.1 showed actual travel times and estimated travel time of the target link DE ($L_D = DE$) using NLIM-EL

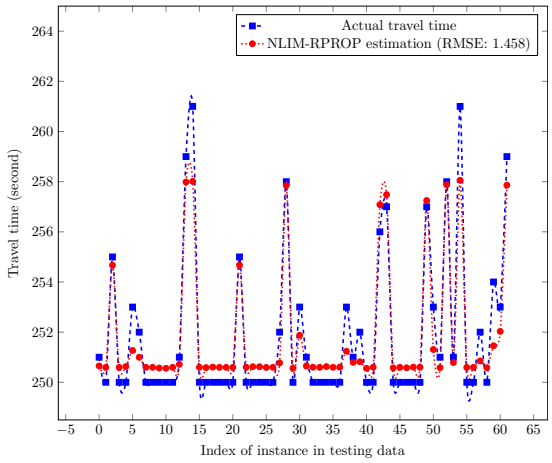


(a) NLIM-EL (original tranning dataset)

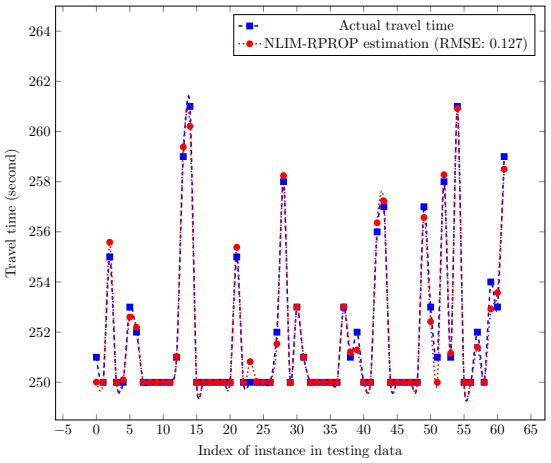


(b) NLIM-EL-OD (outliers excluded in training dataset)

FIGURE 5.1: DE_AD_BD_CD modelled by NLIM on artificial unseen dataset



(a) NLIM-RPROP (original training dataset)



(b) NLIM-RPROP-OD (outliers excluded in training dataset)

FIGURE 5.2: DE_AD_BD_EG modelled by NLIM on artificial unseen dataset

(with and without outliers detection/removal) models on the artificial test data of the traffic link model $L_M^{DE} = \{AD, BD, CD\}$. Figure 5.2 shows actual travel times and estimated travel time of the target link DE ($L_D = DE$) using FF-RPROP-ANN (with and without outliers detection/removal) models on the synthetic test data of the traffic link model $L_M^{DE} = \{AD, BD, EG\}$. $L_M^{DE} = \{AD, BD, CD\}$ and $L_M^{DE} = \{AD, BD, EG\}$ are selected because they give the best performance while using FF-EL-ANN and FF-RPROP-ANN, respectively.

Table 5.1 does not show many differences between the performances of NLIM on two datasets, which are with and without outliers, however in a particular traffic link

TABLE 5.2: Ascending order of RMSE of NLIM-RPORP-OD is to show the ability of NLIM to learn the temporal and spatial relationship of links in traffic link layout. The red links (BD, EG) were assigned as dependent links of the target link DE when the artificial dataset was generated. It means that most of the vehicle getting out of the BD link will pass through the target link and most of the vehicle getting out of the target link will pass through the EG link.

Id	Model name	No. training instances	No. outliers detected	No. testing instances	RMSE*
1	DE,AD,**BD**,**EG**	222	8	62	0.127
2	DE,AD,**BD**,EH	172	36	54	0.131
3	DE,**EG**	4841	759	1409	0.135
4	DE,**BD**,CD,EH	211	10	59	0.15
5	DE,CD,**EG**,EH	219	10	61	0.228
6	DE,**BD**,CD	920	210	279	0.403
7	DE,AD,**BD**	939	145	273	0.407
8	DE,AD,**BD**,CD	225	10	61	0.516
9	DE,**BD**,CD,**EG**	219	0	59	0.818
10	DE,AD,EF,**EG**	215	0	59	1.057
11	DE,CD,**EG**	1142	0	283	1.1
12	DE,CD,EF,**EG**	207	9	59	1.254
13	DE,AD,**EG**	1129	0	279	1.349
14	DE,AD,**EG**,EH	221	20	66	1.354
15	DE,AD,CD,**EG**	214	15	61	1.553
16	DE,**BD**,EH	1047	106	288	1.929
17	DE,**BD**,EF	1021	92	276	2.048
18	DE,EF,**EG**	1035	102	281	2.161
19	DE,**BD**,EF,**EG**	190	15	54	2.35
20	DE,**BD**,EF,EH	195	29	60	3.082
21	DE,AD,**BD**,EF	200	15	59	3.449
22	DE,**BD**	4848	816	1420	4.711
23	DE,**BD**,**EG**,EH	206	22	61	5.148
24	DE,**BD**,**EG**	1158	0	288	6.168
25	DE,**BD**,CD,EF	202	0	52	7.181
26	DE,**EG**,EH	1118	0	279	7.879
27	DE,AD,CD,EH	186	34	59	9.304
28	DE,EF	4739	925	1420	10.021
29	DE,AD	5573	0	1404	10.143
30	DE,CD,EF	1046	79	279	10.205
31	DE,EF,EH	1030	108	282	10.217
32	DE,CD,EF,EH	235	0	65	10.556
33	DE,CD,EH	1104	64	289	10.592
34	DE,EH	5102	516	1413	11.185
35	DE,CD	5643	0	1415	11.273
36	DE,AD,EF	1133	0	280	11.552
37	DE,AD,CD	1028	120	286	11.718
38	DE,AD,CD,EF	193	35	61	13.424
39	DE,AD,EH	1098	0	275	14.211
40	DE,AD,EF,EH	208	17	61	16.324

* RMSE of NLIM-RPROP-OD. Travel time data unit is in a second

model such as DE_AD_BD_CD and DE_AD_BD_EG (Figure 5.1 and 5.2) show that DR-M-GMM based on Gaussian mixture model can detect travel time outliers. Ten and eight travel time outliers are identified in DE_AD_BD_EG and DE_AD_BD_EG respectively. The RMSE of those two models are 3.829, 0.4023 and 1.458, 0.127 for dataset includes and excludes outliers, respectively. Overall, the NLIM works better on the dataset which excludes outliers than the original dataset.

The list of traffic models' performance is shown in Table 5.2. The table further illustrates the number of original samples available in each traffic model and the number of travel time outliers which are detected and removed by DR-M-GMM. The RMSE of FF-EL-ANN orders the list of traffic models affected by dependent links to the target link DE. The models without a dependent traffic link always show higher error i.e models have Id greater or equal to 27.

The models that include either BD or EG or both links have outstanding performances despite ANN-EL-OD, or ANN-RPROP-OD technique is used to model them. Although, some model have the number training labelled greater than those of the top twenty models (i.e traffic link model $L_M^{DE} = \{EH\}$ has 5012 training instances, $L_M^{DE} = \{EF\}$ has 4739 training instances and $L_M^{DE} = \{AD\}$ has 5573 training instances), their performance is less accurate compared to those of top twenty models in Table 5.2. It can be seen that the traffic link models (Id is in between 27 and 40) in Table 5.2 do not contain any dependent link. It proves that NLIM-EL-OD and NLIM-RPROP-OD can identify the dependent links and they can model the relationship between links of a traffic link model using the artificial dataset.

Time to train and test the different machine learning techniques has also been considered in this work. The training time is vital because of the NLIM works on an extensive traffic network with thousands of traffic links. The estimated time is also an important indicator in near real-time travel time estimation.

TABLE 5.3: Training and testing time of NLIM on artificial dataset. (1) Lower-whisker, (2) Lower-quartile,(3) Median, (4) Upper-quartile, (5) Upper-whisker

	(1)	(2)	(3)	(4)	(5)
Training time [milliseconds]					
MLR	3.49E-05	0.0001246	0.00025975	0.0004875	0.0269879
FF-EL-ANN	0.50731	0.78159875	2.0522891	5.7364601	13.4753285
FF-RPROP-ANN	0.7841345	1.3596814	3.33414355	9.1476801	21.3501359
SVR-LK	0.10041	0.10107125	0.111166	0.20039905	3.5493465
Test time [milliseconds]					
MLR	5.4E-06	1.81E-05	3.14E-05	5.255E-05	0.0012631
FF-EL-ANN	0.0903518	0.09305	0.0960817	0.0991307	0.1011442
FF-RPROP-ANN	0.0903919	0.09274685	0.0952136	0.0973707	0.1052399
SVR-LK	0.0906738	0.09575615	0.0990418	0.1000641	0.1922713

Table 5.3 shows that NLIM with FF-RROP-ANN has the highest training times. However, its testing time is not significantly different compared to other machine learning techniques. Furthermore, other machine learning techniques such as MLR, SVM-LK, FF-EL-ANN can reduce the training time when NLIM works on a large number of traffic link models, but NLIM with FF-RPORP-ANN can produce better travel time for near real-time estimation. Consequently, the four machine learning techniques mentioned in Table 5.3 are considered in the future experiments on other datasets.

Summary of results

This section has demonstrated the ability of the application of multi-variable Gaussian mixture model and the proposed NLIM methodology on the artificial dataset generated by BPR function. Five different machine learning techniques have been used to demonstrate the ability of NLIM in modelling the traffic links.

NLIM shows the effectiveness of modelling the relationship between temporal and spatial relationships in traffic links of a traffic link model. The performances of the NLIM using different machine learning techniques vary. The SVM techniques are likely inappropriate for modelling the traffic link model due to the inadequate performance.

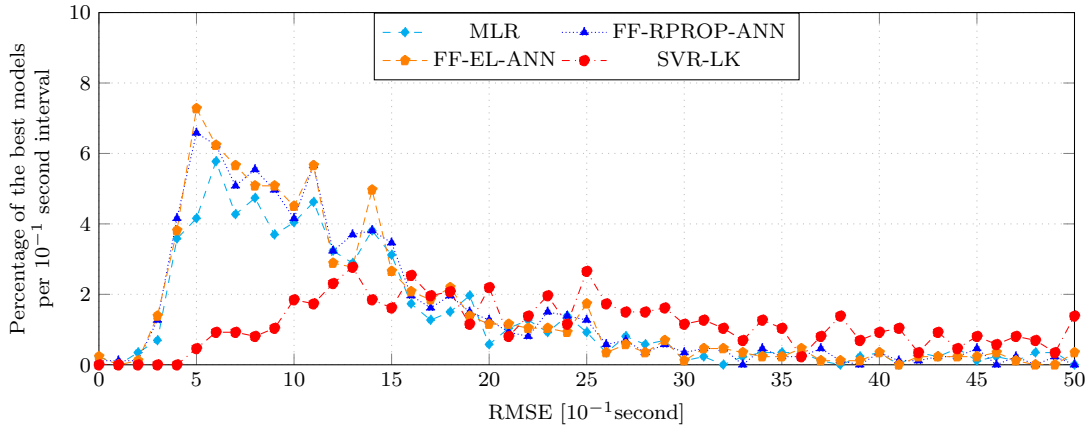
The result shows that FF-EL-ANN and FF-RPROP-ANN can identify the target links BD and EG on the traffic link models and they can produce travel time estimation with low errors. The application of multivariable Gaussian mixture model on detection and removal of outliers shows promising results. The DR-M-GMM can detect the travel time outliers in a combination of travel times.

The artificial data generated by BPR function represents flow, delay and travel time relationship in traffic. The dataset may not adequately show the dynamic and uncertain of the real traffic data. NLIM still needs to be carefully assessed on other datasets to confirm the performance of the proposed methods and to evaluate NLIM strengths and weaknesses.

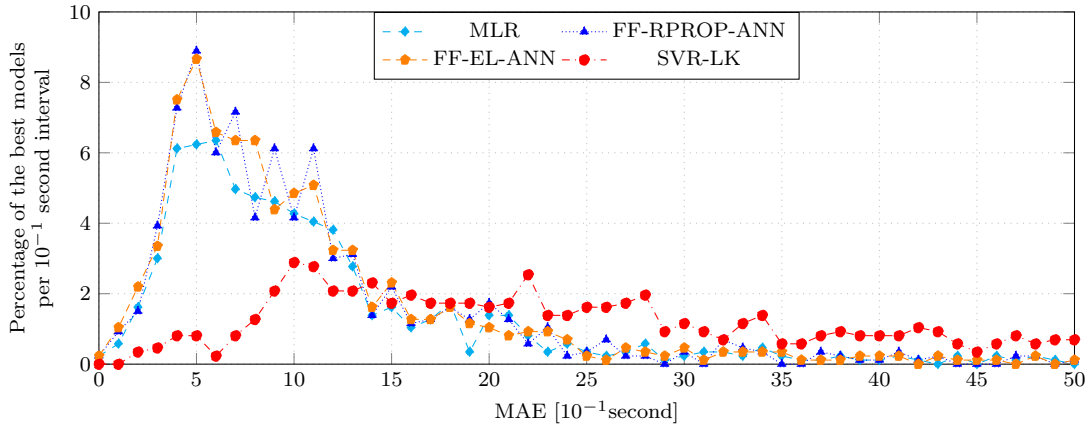
5.2.2 Experiment 2: SUMO dataset

The second synthetic dataset used in this thesis is the SUMO dataset. The SUMO simulator can provide travel time data for links in an extensive traffic network. The dataset can be generated on demand and in a great detail. The synthetic data from SUMO also gives the flexibility that the real data cannot provide while still keeping true to actual network behaviours such as travel time in a different time intervals, and the density of travel time samples.

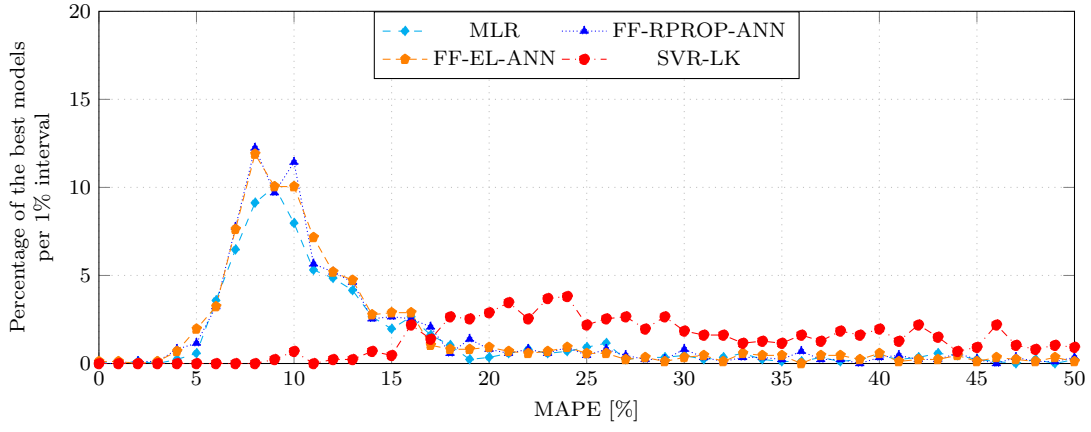
The dataset gathered from SUMO contents 30947 links in total. And 3840 links have data sparsity more than or equal to 99%. 1826 link models have a sufficient number of training and testing data according to Equation 4.12. The links in this experiment do



(a) RMSE



(b) MAE



(c) MAPE

FIGURE 5.3: Histogram (vertical scale is in percentage of the total models) of the best models vs different performance criteria achieved by Neighbouring Link Inference Method (NLIM) using multi-linear regression (MLR), feed forward resilient back-propagation neural network (FF-RPROP-ANN), feed forward evolutionary learning neural network (FF-EL-ANN) and support vector machine regression (linear kernel) (SVR-LK) on SUMO unseen data. Outliers are identified and removed from the unseen test data by applying Algorithm 4.3.

not classify into different link types due to a lack of the information of link types in the simulation scenario. Four machine learning techniques are employed in the experiment. They are MLR, FF-EL-ANN, FF-RPROP-ANN and SVR-LR. Three different error metrics are used to evaluate the performance of NLIM models. These metrics are RMSE, MAE and MAPE.

NLIM-MLR, NLIM-SVR-LK are defined as NLIM that employs MLR and SVR-LK, respectively. And MLIM-MLR-OD, NLIM-SVR-LK-OD are NLIM-MLR and NLIM-SVR-LK that uses DR-M-GMM, respectively. Figure 5.3 demonstrates the performance of NLIM with different machine learning techniques by different performance metrics. The four machine learning techniques including MLR, FF-EL-ANN, FF-RPROP-ANN and SVR-LK are utilised to model traffic links. Performance metrics of the models are evaluated on unseen data. Figure 5.3(a) is for RMSE, Figure 5.3(b) is for MAE, and Figure 5.3(c) is for MAPE. The best model represents the performance of NLM models in a link layout. The best model is one within models which is generated from the traffic link layout, and it has the smallest performance metric.

TABLE 5.4: The performance metrics of NLIM models on SUMO dataset, different machine learning techniques applied, with and without DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

Machine learning technique	DR-M-GMM					Original dataset				
	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
RMSE [seconds]										
MLR	0.00	0.95	1.79	13.78	6842603	0.00	0.91	1.62	7.75	784360
FF-RPROP-ANN	2.8E-3	0.79	1.30	2.68	388.54	2.3E-3	0.77	1.31	2.70	386.54
FF-EL-ANN	0.00	0.77	1.27	2.59	255.95	0.00	0.77	1.27	2.58	385.29
SVR-LR	0.00	3.98	14.75	94.80	2842603	0.00	4.00	14.88	96.2	282603
MAE [seconds]										
MLR	0.00	0.76	1.45	11.78	703	0.00	0.74	1.32	6.15	1861.00
FF-RPROP-ANN	2.7E-3	0.63	1.04	2.17	214.71	2.3E-3	0.63	1.05	2.19	241.79
FF-EL-ANN	0.00	0.62	1.04	2.09	177.42	0.00	0.62	1.04	2.09	226.85
SVR-LR	0.00	1.75	3.53	9.00	16.00	0.00	1.75	3.55	9.06	1686.00
MAPE [%]										
MLR	0.00	9.41	14.33	62.53	4852312	0.00	9.19	13.53	43.16	4284263
FF-RPROP-ANN	0.06	8.43	11.25	18.74	3042601	0.05	8.48	11.39	18.68	3142621
FF-EL-ANN	3E-4	8.30	11.06	18.26	3746032	0.00	8.30	11.04	18.39	3984613
SVR-LR	0.00	22.87	35.41	58.34	4342635	0.00	23.01	35.52	59.00	4842601

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

According to the results in Figure 5.3, the three performance metrics which are given by four different machine learning techniques within NLIM are notable small. Both the RMSE and the MAE of the vast majority of models are less than or equal to 4.5 seconds. Many of NLIM models have MAPE less than or equal to 20%. The performances of FF-EL-ANN shows similar to the performance of FF-RPROP on unseen data. The performance of MLR is slight less accurate than those of FF-EL-ANN and FF-RPROP-ANN. However, the performance of SVR-LK notably

shows less reliable compared to the rest. Especially, MAPE of NLIM-SVR-OD shows significant high compared to the other machine learning techniques.

Table 5.4 presents information from a five-number summary (lower-whisker, lower-quartile, median, upper-quartile and upper-whisker) of the NLIM performances. Different metrics are used to illustrate the performances of NLIM. NLIM models perform on SUMO unseen data give the metrics. It clearly shows that the SUMO dataset which is preprocessed by DR-M-GMM can be used to model NLIM better than the original dataset. The performances of 75% of the best models of NLIM-RPROP-OD and NLIM-EL-OD are less than or equal to 1.0 seconds for using either RMSE or MAE metric. The performances of NLIM-MLR-OD models are less accurate than others. However, the performances of NLIM-MLR-OD models are still satisfactory. RMSE and MAE of 75% of the best ANN-MLR-OD models are less than or equal to 1.62 seconds. MAPE of 75% of the best NLIM-EL-OD and NLIM-RPROP-OD models are less than or equal 20%. Those for NLIM-MLR-OD and NLIM SVR-LK are less than or equal 64%.

TABLE 5.5: The statistics of the number outliers over 3840 links detected by Algorithm 4.3 in SUMO dataset.

Minimum	Maximum	Mean	StdEv
0	52	0.83	3.16

There are slight differences between the performance of NLIM on the original dataset and NLIM on datasets excluding outliers. Table 5.5 shows that the number of outliers which were detected by DR-M-GMM is a small amount. The mean of the number of outliers is 0.83, and the standard deviation is 3.16. Hence, the performance of NLIM trained on data with and without outliers are slightly different. DR-M-GMM do not show conclusively on detecting outliers in travel time SUMO dataset compared to those from BPR function because the dataset may not contain many outliers.

Summary of results

In this section, the application of multi-variable Gaussian mixture model and the proposed NLIM methodology has been studied on the SUMO dataset. Five different machine learning techniques have been used within NLIM to model the traffic link models.

NLIM shows the effectiveness of modelling the between temporal and spatial relationship of travel times in traffic links. The performances of the NLIM using different machine learning techniques vary. The SVM techniques are likely inappropriate for modelling the traffic link model due to the sparse MAPE performance metric. The RMSE and MAE performance metrics of SVM-LK are still acceptable compared to other machine learning techniques. The performances of NLIM employed SVR-LK using RMSE, MAE and MAPE metric do not show accurately on travel time estimation for short links due to the increase of MAPE metric. The results indicate that the travel time dataset from SUMO simulation has few outliers that can be detected by DR-M-GMM.

5.2.3 Experiment 3: WebTRIS dataset

A similar analysis is undertaking using the real travel time dataset from WebTRIS. The dataset is a one-year travel time of 74 motorway links and 41 A links in East Midlands in the UK. The traffic links include M6, M45, A5, A14 and A28. The experiment area and the detail of the dataset are described in Section 4.7.3. The performance metrics are created using the unseen data. Three performance metrics including RMSE, MAE and MAPE are utilised. The best model represents the performance of NLM models in a link layout. The best model is one within models which is generated from the traffic link layout, and it has the smallest performance metric. The performance of the best model in traffic layouts is used in the analysis.

Initially the training and testing time of the four machine learning techniques are tested using the dataset. Figure 5.4 shows results of the analysis. The experiment shows that SVR-LK is not suitable to train traffic models on a big travel time dataset due to very long training time. Precisely, the training time of NLIM-SVR-LK dramatically increases at 3000 training sample from a few seconds to 30 seconds. At 8000 training samples (over the maximum number of labelled data: 45651 samples), the training time is 9000 seconds despite grid search was applied to select hyper-parameters for SVR-LK before training. Due to the long training time of SVR-LK models on a large number of labelled data, SVR-LK are not employed in NLIM on the real travel time dataset.

Figure 5.4 also shows the training time of the other machine learning techniques. They work well on the real travel time dataset. They are all less than or equal to 17 seconds for around 8000 training data samples.

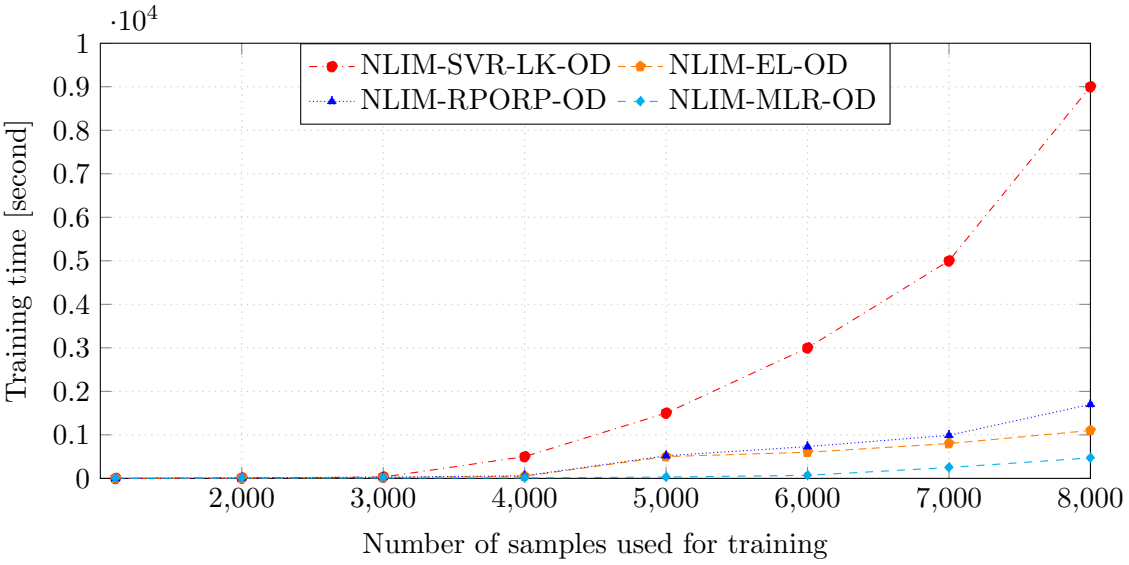
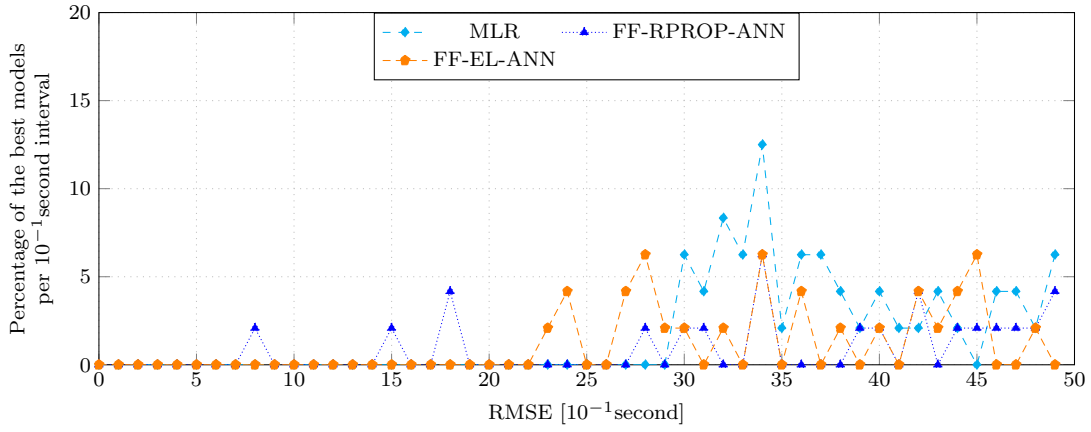


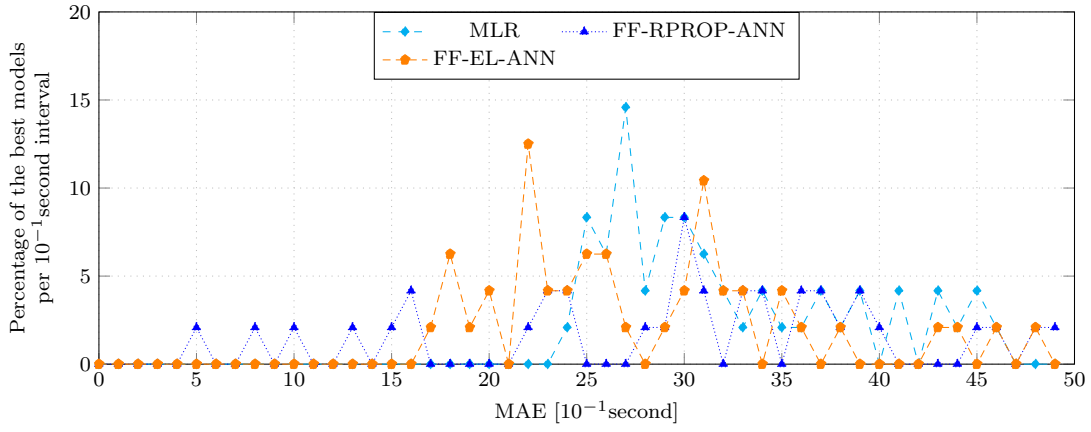
FIGURE 5.4: NLIM training time vs the training sample size on WebTRIS dataset. The maximum and minimum number of labelled data of models are 45651 and 1108 respectively

Figure 5.5 demonstrates the performance of NLIM in different machine learning techniques by different performance metrics (RMSE, MAE and MAPE). MLR, FF-EL-ANN and FF-RPROP-ANN are employed to model traffic links. Performance metrics of those models are calculated on the WebTRIS unseen data. Figure 5.5(a) is for RMSE, Figure 5.5(b) is for MAE, and Figure 5.5(c) is for MAPE. The performances metrics of NLIM models show higher values than those on two synthetic datasets (artificial dataset and SUMO dataset) however, it can be seen that all performance metrics of three machine learning techniques employed in NLIM are remarkably low values on the unseen data. They still show conclusively on learning the temporal and spatial relationships between travel times in links comparing those on SUMO and BPR datasets. Both RMSE and MAE of NLIM models are less than 9.0 seconds for more than 75% of the total NLIM models. The vast majority of the NLIM models have MAPE less than or equal to 15% in the experiments for all MLR, FF-EL-ANN and FF-RPROP-ANN.

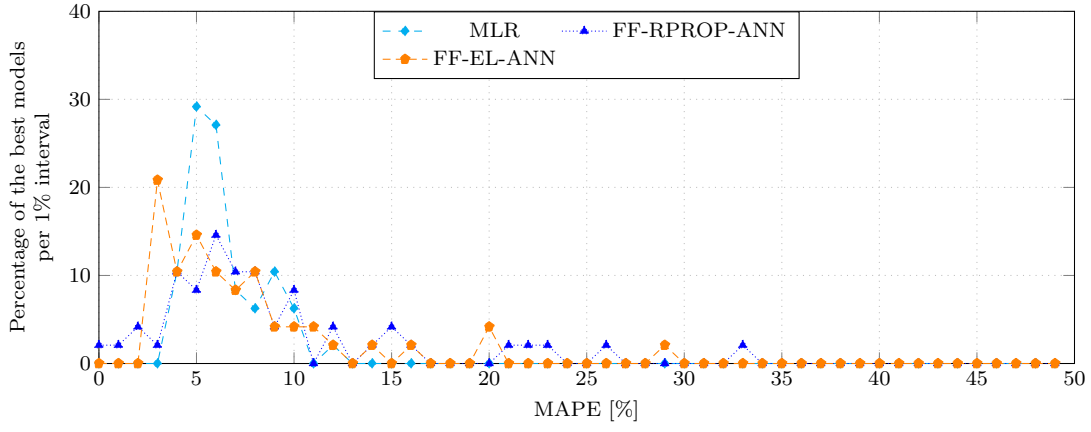
The difference between the five-number summary of the NLIM model using dataset that was preprocessed by DR-G-MM (Algorithm 4.3) and those using original dataset repeatedly show promising performance of the outlier detection and removal algorithm. NLIM models using preprocessing dataset perform better than NLIM models using original dataset.



(a) RMSE



(b) MAE



(c) MAPE

FIGURE 5.5: Histogram (vertical scale is in percentage of the total models) the best models vs different performance criteria achieved by Neighbouring Link Inference Method (NLIM) using multi-linear regression (MLR), feed forward resilient back-propagation neural network (FF-RPROP-ANN), feed forward evolutionary learning neural network (FF-EL-ANN) and zupport vector machine regression (linear kernel) (SVR-LK) on WebTRIS unseen dataset. Outliers are identified and removed from the unseen test data by applying Algorithm 4.3.

TABLE 5.6: The performance metrics of NLIM models on WebTRIS dataset (different machine learning techniques applied) with and without DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

	DR-M-GMM					Original dataset				
Machine learning technique	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
RMSE [seconds]										
MLR	3.08	3.41	3.93	4.77	6.03	3.08	3.43	4.04	4.85	7.50
FF-RPROP-ANN	0.97	3.68	6.26	8.12	35.29	0.82	4.64	6.37	8.75	14.81
FF-EL-ANN	2.13	3.48	4.38	5.57	11.07	2.30	3.63	5.21	6.87	13.99
MAE [seconds]										
MLR	2.49	2.81	3.23	3.86	5.54	2.49	2.77	3.12	3.84	5.45
FF-RPROP-ANN	0.57	2.44	3.83	5.86	35.02	0.56	3.06	3.96	5.93	11.57
FF-EL-ANN	1.32	2.33	2.75	3.67	7.76	1.74	2.42	3.12	3.89	6.34
MAPE [%)]										
MLR	4.79	5.60	6.73	8.93	12.74	4.79	5.51	6.56	8.11	14.87
FF-RPROP-ANN	1.06	4.86	8.15	14.064	55.39	0.91	6.33	8.68	14.26	37.27
FF-EL-ANN	2.18	4.28	5.89	8.54	26.21	3.03	4.71	7.34	10.61	31.71

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

TABLE 5.7: The statistics of the number outliers detected by DR-M-GMM on WebTRIS dataset on 158 traffic models (minimum, average and maximum training samples are 1250, 19061 and 47625)

Minimum	Maximum	Mean	StdEv
0	2190	375.92	482.62

MLR, FF-EL-ANN and FF-RPROP models using WebTRIS dataset show different performances on unseen data. Although NLIM-MLR models yield more accurate on several estimations for the target links than FF-EL-ANN and FF-RPROP-ANN (Figure 5.5), overall, FF-EL-ANN and FF-RPROP-ANN provide more substantial of NLIM models that have RMSE, MAE, MAPE less than or equal to 3.08 second, 2.49 second and 2% than MLR. The difference between the performances of the three machine learning techniques is not significant based on the overlap of the five-number values on Table 5.6. Note that traffic links on this experimental are all motorway and A traffic links. Between target link and adjacent links mostly do not have traffic lights in between the traffic link layout, so the temporal and spatial relationships between travel times of links are less complex.

Table 5.6 presents NLIM performances using the five-number summary by different performance metrics. It clearly shows that NLIM performs better on the WebTRIS dataset which excludes outliers than on the dataset which includes outliers. 75% of the best model of NLIM-MLR-OD, the best model of NLIM-RPROP-OD and the best model of NLIM-EL-OD are less than or equal to 9.0 seconds for both RMSE and MAE. NLIM-MLR-OD can produce a higher number of models that have RMSE around of less than 2.5 seconds, but overall, NLIM-EL-OD and NLIM-RPROP-OD can model a

higher number of traffic link layouts.

Table 5.7 indicates the statistics of the number of outliers which are removed using DR-M-GMM. The average of 375.92 outliers have been detected and removed per traffic link model dataset. 1.97% of the data have been identified as outliers and hence were removed from the training set. The standard deviation of the number of outliers is quite high. It indicates that the number of outliers removed in individual traffic considerably varies.

Summary of results

Three machine learning techniques (MLR, FF-EL-ANN, FF-RPROP-ANN) have been employed in NLIM to learn the temporal and spatial relationships between travel time in links using WebTRIS dataset. The NLIM-MLR-OD, NLIM-EL-OD, NLIM-RPROP-OD show accurate travel time estimation on the unseen dataset.

NLIM-MLR-OD models work well on WebTRIS dataset. That might be because of the motorway links, and vital A links are included in the experiment only. Overall, NLIM-MLR-OD, NLIM-RPROP-OD and NLIM-EL-OD performed well in terms of the adopted performance indicators.

5.2.4 Experiment 4: FCD dataset

The relationship between links in a traffic link model has been examined using artificial travel time dataset which gathered by using BPR formula in Algorithm 4.8, SUMO dataset and WebTRIS dataset. The results have shown that the proposed method can detect and model the temporal and spatial relationship between the travel time of the target link and its neighbouring links.

For the FCD dataset, the travel time data was collected from September 2009 to February 2012 in Leicestershire, UK. The dataset used in this thesis comprises travel times for 22053 traffic links. They include 67 motorway links, 22 trunk links, 911 primary link, 1457 A links, 843 B link and 13752 minor links (Table 4.3). In 13752 minor links, 5226 links have data sparsity less than or equal to 99%. They account for 38% of the total minor links.

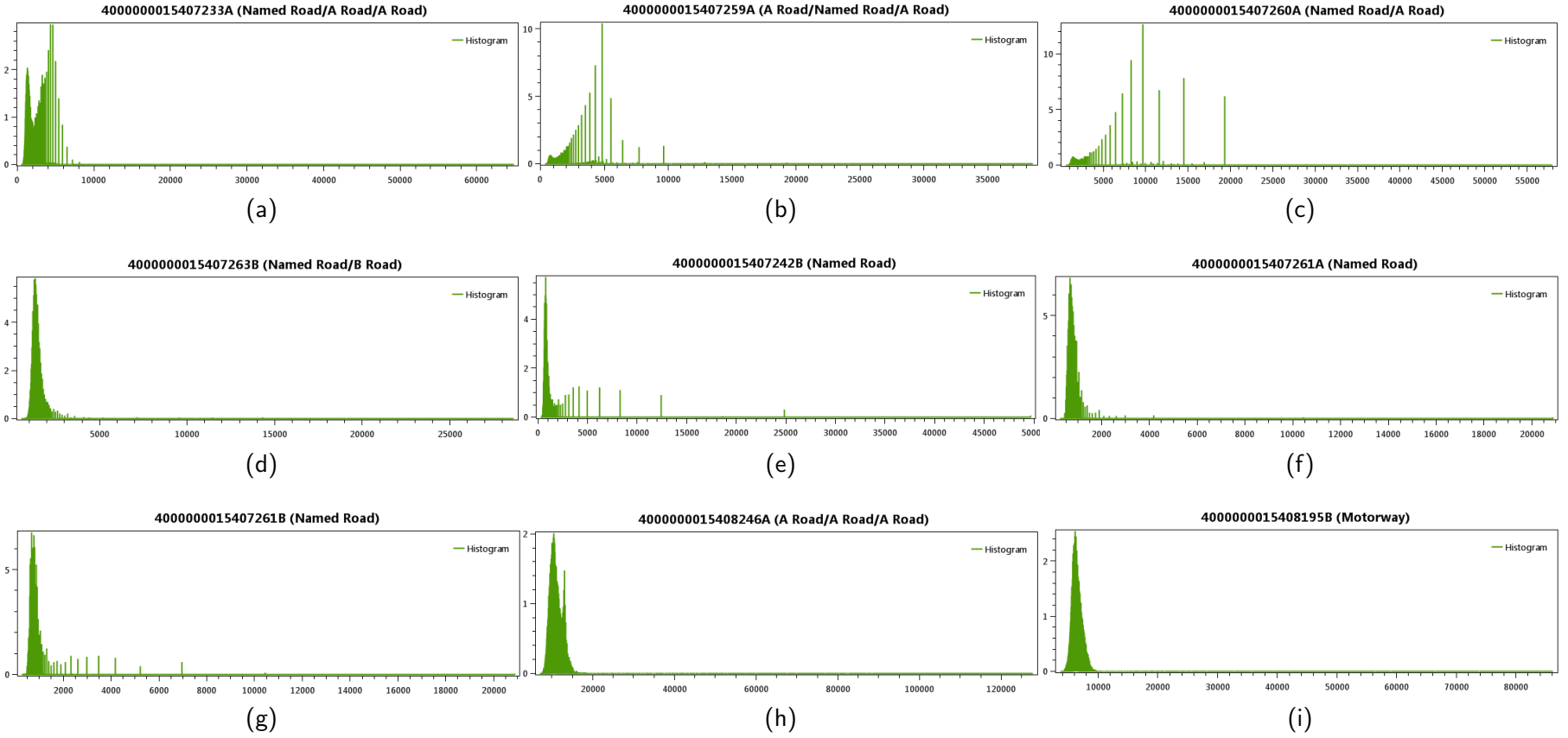


FIGURE 5.6: Histogram of travel time on traffic links (travel time is in a 10^{-2}second). The x-axis is travel time. The y-axis is the number of travel time samples per 10^{-2}seconds. Each histogram is for travel time of an example traffic link in the traffic network.

The raw data collected from FCD contain reconstructed link travel times at 15 minute intervals. Thereby, a day starting from 00h00 to 23h59, was divided equally into 96-time slots. For example, time slot 34 covers the period from 08:30:00 – 08:44:59. The travel time data unit is a 10^{-2}second. The average travel time in minutes per miles for links in the traffic network is approximately 2.46 (minutes/miles). Total links' length is roughly 14,000 kilometres or 8,700 miles. There are nine vehicle classes which are based on the payload and the size of the vehicle.

Initially, some frequency histogram of travel times against there values are analysed and data sparsity of travel time in links are calculated in order to give an insight into the complexity, irregularity and sparsity of the dataset. Figure 5.6 shows the frequency histogram of travel times of several traffic links, Figure 5.7 graphically illustrates the data sparsity in link over the traffic network and Figure 5.8 plots data sparsity in links.

Figure 5.6 graphically shows that the histogram of traffic links is different and the range of travel times in traffic links varies. The travel times range from less than three seconds to over 600 seconds. The figure also graphically describes the distribution of travel times on traffic links have long right tails and different scales. It means that there are some very high travel time values on all studied traffic links. Hence, the preprocessing data for the travel times in the dataset is most likely required. It includes outliers detection/removal and data normalisation.

According to the travel time data that is acquired from Leicestershire from 2009 to 2012, the data sparsity of links visually shows very high values in Figure 5.7 where data sparsity is presented by colour on map for every links. Figure 5.8(a) indicates that 69.42% of total links have data sparsity is less than or equal to 99%. Figure 5.8(b) shows that the number of travel time samples is different by the time of the day. There are more travel time data for the time interval between 7am and 7pm. NLIM is proposed for large scale traffic networks, hence it should perform well for most of the links in the traffic networks.

The FCD dataset which is a real historical travel time dataset is used to assess the performances of NLIM. The NLIM employed MLR, FF-EL-ANN and FF-RPROP-ANN are trained on the dataset. The performance metrics of NLIM models are given by using unseen data. After that, the performances of NLIM models will be compared to the achievements of NLIM models on three previous datasets. As discussed in the previous

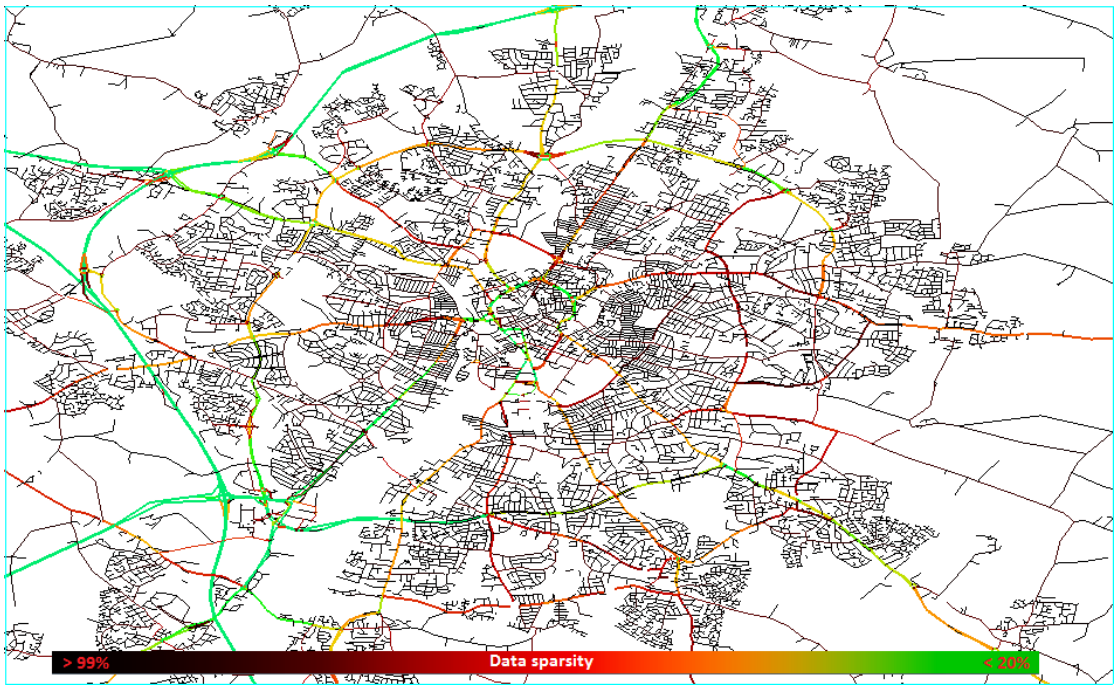


FIGURE 5.7: Experiment data sparsity visual in the Experiment 4 map

section, due to the long training time on a large number of labelled data and less accurate of SVR-LK models, so it is eliminated in this experiment.

To demonstrate the advantages of NLIM models in different machine learning techniques, their performance is compared with those of Historical Average (HA) and Moving Average (MA) methods Tang et al. (2018). HA and MA are typical standard methods that allow estimating the current travel time by using historical travel time data. HA uses the corresponding average of the historical travel time of a time slot on a target link to estimate the current time slot travel time on the link. Meanwhile, MA uses moving average of three-time slots right before the current time slot to estimate the current time slot travel time Tang et al. (2018).

Input features for training and testing NLIM models in a link layout using the proposed method are sparse historical travel time of neighbouring links, time of day(time slot), vehicle class and day of a week. Day of a week is inferred from the date. Output feature is travel time corresponding to the input features of a target link.

13527 links produce 13527 link layouts. There are 338177 traffic link models in total. The models are carefully trained and verified to make sure relationships between temporal and spatial of travel times in links correctly learnt. The performances of models are evaluated by RMSE, MAE and MAPE performance metrics on unseen data. Outliers of

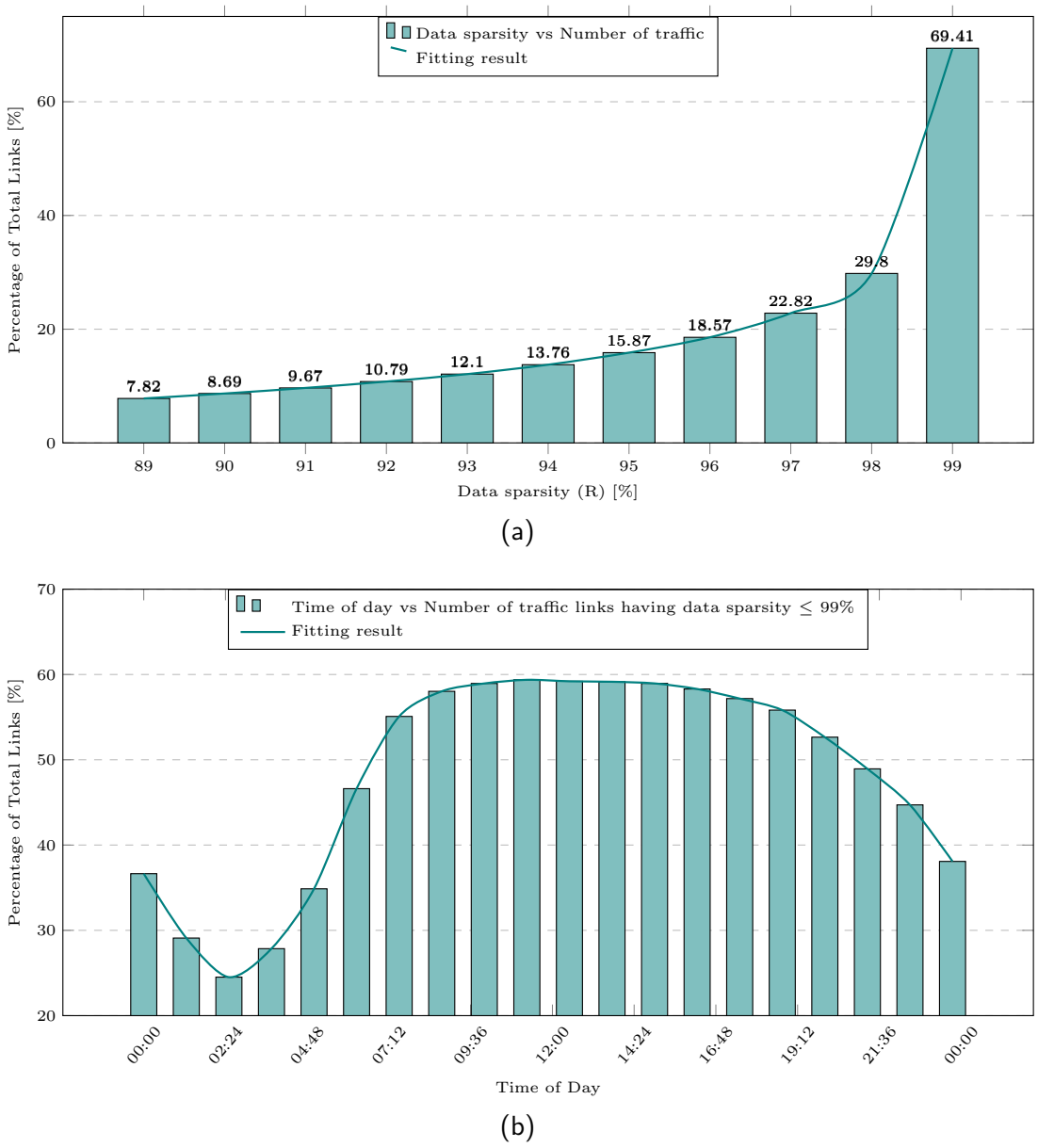


FIGURE 5.8: Experiment 4 data sparsity in links using acquired data (2006-2012) in Leicestershire by time of day.(a) Percentage of total links vs data sparsity.(b) Time of day vs Traffic links having data sparsity $\leq 99\%$.

data set on 13527 traffic links are detected and removed using the proposed DR-M-GMM (Algorithm 4.3). The parameter k and γ are set up as mentioned in Section 4.4 ($k = 5$, $\gamma = 0.1$).

Table 5.8 presents statistics of NLIM, MA and HA performances using the five-number summary in different performance metrics. The performances of FF-EL-ANN and FF-RPROP-ANN are significantly better to those achieved using MLR, HA and MA. The performances of MLR are more accurate than those of HA and MA. It clearly shows that NLIM performs better on the FCD dataset which excludes outliers than on

TABLE 5.8: The performance metrics of NLIM models (different machine learning technique applied), MA models and HA models on unseen dataset with and without DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

Machine learning technique	DR-M-GMM					Original dataset				
	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
RMSE [seconds]										
MA	0.23	1.64	3.39	9.12	473.51	0.24	1.65	3.40	9.16	473.51
HA	0.12	2.43	5.78	12.76	625.03	0.15	2.46	5.81	12.82	625.03
MLR	0.12	1.62	4.18	12.63	153.45	0.12	1.71	4.38	127.41	153.45
FF-EL-ANN	0.04	1.41	3.13	7.85	548.28	0.04	1.43	3.24	7.99	548.42
FF-RPROP-ANN	0.04	1.48	3.25	8.10	548.15	0.03	1.52	3.33	8.08	548.76
MAE [seconds]										
MA	0.13	0.91	1.71	3.52	310.19	0.14	0.93	1.72	3.56	310.19
HA	0.08	1.22	2.65	5.60	454.30	0.09	1.25	2.67	5.63	454.30
MLR	0.15	0.84	1.96	5.10	830.26	0.15	0.88	1.87	4.20	830.26
FF-EL-ANN	0.02	0.76	1.63	3.54	380.59	0.02	0.82	1.77	3.92	244.85
FF-RPROP-ANN	0.02	0.80	1.72	3.74	424.95	0.02	0.86	1.84	4.01	225.00
MAPE [%]										
MA	12.67	25.02	30.01	38.86	403.31	14.61	27.78	32.41	41.16	403.31
HA	12.41	22.89	30.83	45.73	401.26	14.44	25.19	31.15	46.31	401.26
MLR	8.03	18.24	24.69	40.31	7894.34	8.08	19.6216	25.90	38.87	7894.34
FF-EL-ANN	3.07	12.72	17.15	25.78	910.30	3.21	14.17	19.99	31.52	743.53
FF-RPROP-ANN	1.26	13.42	18.08	27.14	3177.59	1.09	14.48	20.48	32.02	1175.61

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

the unfiltered dataset. 75% of the best model of NLIM-MLR-OD, the best model of NLIM-RPROP-OD and the best model of NLIM-EL-OD are less than or equal to 8.1 and 3.7 seconds for RMSE and MAE respectively.

In comparison, according to the performances metrics of NLIM in Table 5.1, 5.4, 5.6 and 5.8, NLIM models trained on the FCD dataset of varying link categories (motorway, trunk, primary, A, B and minor link) give slightly less accurate compared to those trained on the artificial, SUMO and WebTRIS dataset. On average, RMSE, MAE and MAPE of NLIM of 13527 links on FCD dataset are lower than NLIM of 115 links on WebTRIS. They are approximately 2.0 seconds, 1.8 seconds and 10% for RMSE, MAE and MAPE respectively.

The FCD dataset contains six link categories. The number of links is analysed on experiment dataset is approximately 117 times the number of links on WebTRIS dataset. The number of travel time models in FCD dataset is approximately 980 times the number of models on WebTRIS dataset. The traffic network in this experiment covers rural, urban as well as motorway traffic network. As can be seen in the previous section, the performance of NLIM on WebTRIS data which includes most of the motorway links and a several vital A links are very promising results. High data sparsity might impact NLIM performance and high variability of urban travel time which is profoundly affected by traffic light cycles, queuing delays, the pedestrians and cyclists etc. Furthermore, the historical data collected on urban traffic network are contaminated with noise.

TABLE 5.9: The statistics of the number outliers detected by DR-M-GMM over 338177 traffic link models on FCD dataset

Minimum	Maximum	Mean	StdEv
0	13448	224.56	658.73

Table 5.9 shows a statistics of the number of outliers of the NLIM models in FCD dataset that is detected and removed by using Algorithm 4.3 (DR-M-GMM). The minimum, maximum, mean and standard divination of the number outliers on FCD dataset are approximately those on WebTRIS dataset. The mean and standard deviation of the outliers are 224.56 and 658.73 while those in WebTRIS are 375.92 and 482.62 respectively.

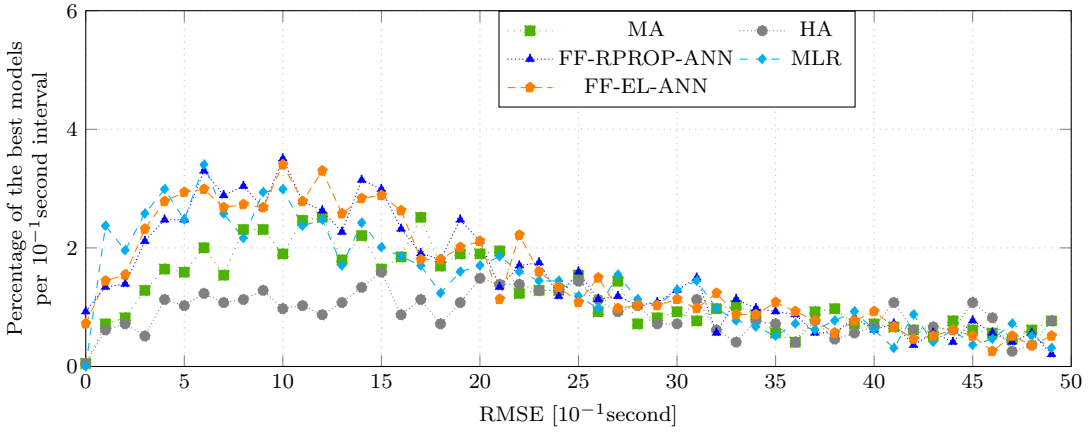
TABLE 5.10: Data sparsity (%) on Motorway, Trunk, Primary, A, B, Minor link types observed on the historical dataset.

		Urban				
	Motorway	Trunk	Primary	A	B	Minor
Lower whisker	0.0	20.0	36.9	0.0	39.9	68.6
Lower quartile 25%	10.5	40.2	70.7	76.8	83.7	92.9
Median	19.5	74.6	77.6	81.3	87.1	95.5
Upper quartile 75%	54.9	81.5	85.0	85.9	90.5	97.7
Upper whisker	93.1	98.7	100	100	100	100

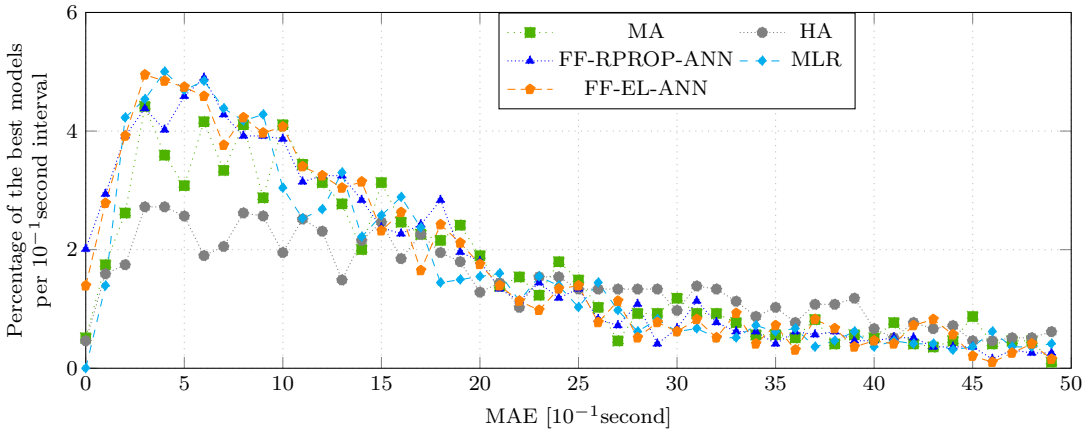
As it can be seen in Table 5.10, the data used in this experiment have very high data sparsity. Majority of links in the traffic network have data sparsity greater than 70%. Especially on minor links, for which the data sparsity is higher than 90%. They consequently make the urban traffic links more challenging to model compared to the motorway links.

Focussing closer on the performances of NLIM, MA and HA, Figure 5.9 illustrates the performances of MA, HA, and NLIM using different performance metrics. Each figure is for a specific performance metric of a particular MA, HA and NLIM model. It contains three sub-figures. They present histograms of best NLIM, HA and MA models using RMSE metric(Figure 5.9(a)), MAE metric (Figure 5.9(b)) and MAPE metric (Figure 5.9(c)). The histograms of the best models show the percentage of the best models per error units (RMSE and MAE are 10^{-1}, MAPE is 1% respectively).

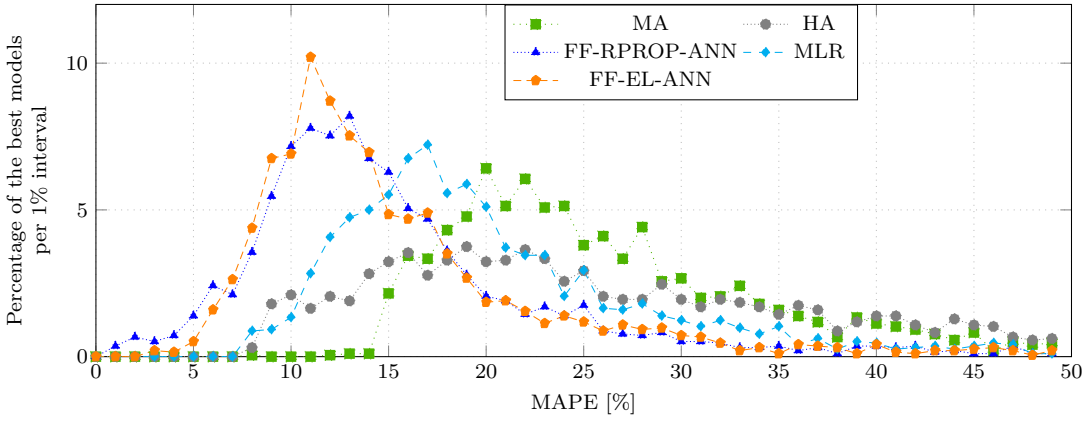
In Table 5.8 and Figure 5.9, it can be seen that the performance of NLIM-EL-OD models is roughly similar to NLIM-RPROP-OD models on unseen dataset. There is no difference between their performance metrics. However, NLIM-RPROP-OD and NLIM-EL-OD work much better than NLIM-MLR-OD on all performance metrics provided. These results are different with the artificial datasets where NLIM-EL-OD,



(a) RMSE of the best models



(b) MAE of the best models



(c) MAPE of the best models

FIGURE 5.9: Histogram (vertical scale is in percentage of the total models) of the best models vs their performance metric achieved by MA, HA and Neighbouring Link Inference Method (NLIM) using multi-linear regression (MLR), feed forward resilient back-propagation neural network (FF-RPROP-ANN), feed forward evolutionary learning neural network (FF-EL-ANN) on unseen dataset. Outliers are identified and removed from the unseen test data by applying Algorithm 4.3. Sub-figures are in different scales.

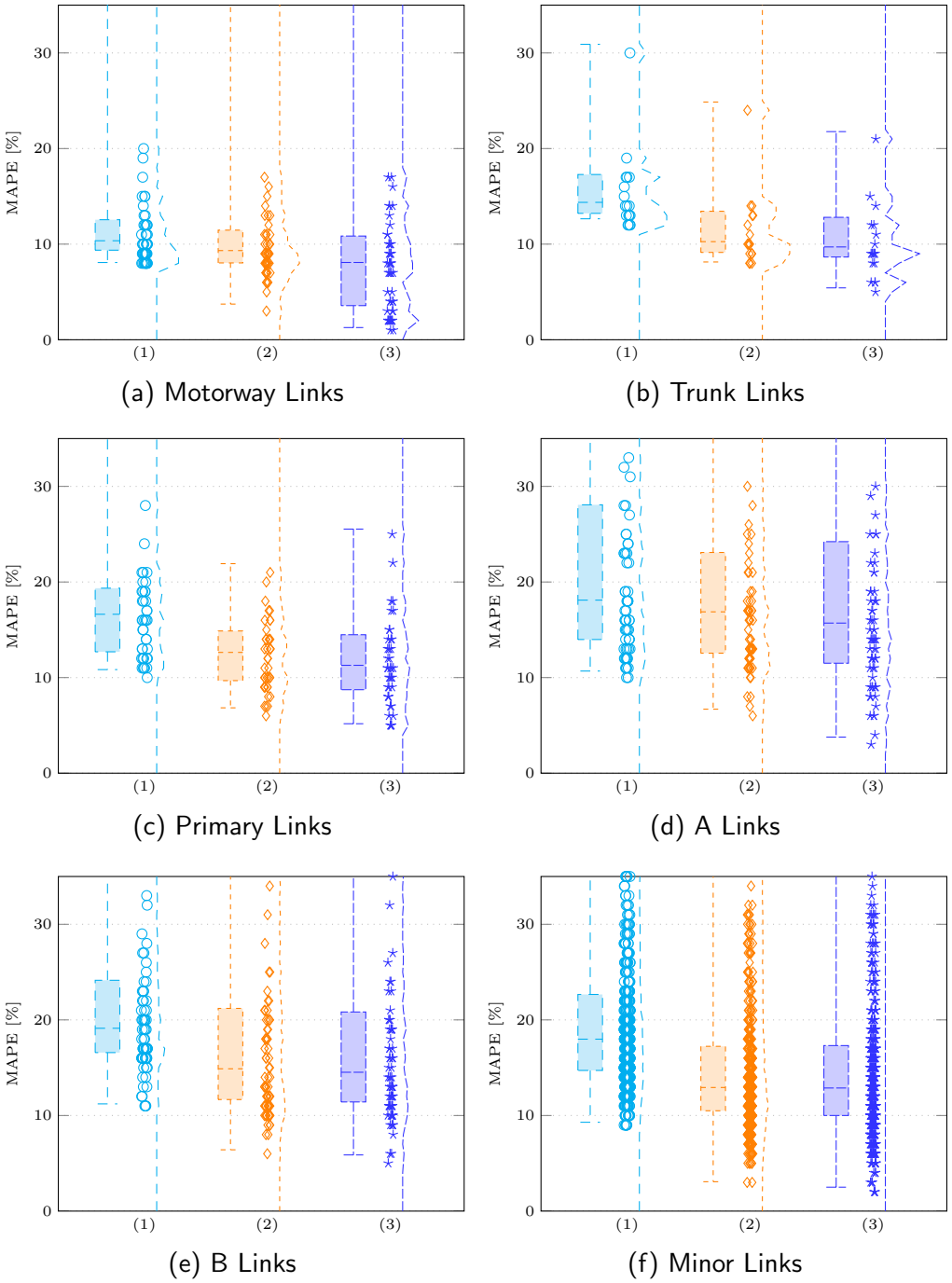


FIGURE 5.10: Density of the best NLIM models, different machine learning techniques applied, of Motorway (a), Trunk (b), Primary (c), A (d), B (e) and Minor (f) links and their MAPEs [%] achieved on unseen data. Sub-figures are in the same scale. (1) is NLIM-MLR-OD, (2) is NLIM-EL-OD and (3) is NLIM-RPROP-OD. The density of best models in each method is presented by boxplot (lower whisker, lower quartile, median, upper quartile, upper whisker), visualisation of the actual individual model and the histogram of the models respectively. Some high MAPE data points are out of the figure, hence corresponding upper-whiskers can not be shown.

NLIM-RPROP-OD and NLIM-MLR-OD do not show significant differences. Still, the results on the dataset are similar with those on WebTRIS dataset. The results for

FCD dataset are similar to the results for WebTRIS dataset because WebTRIS traffic network is likely a sub-problem of FCD traffic network where the traffic network contains only motorway links and vital A links.

Results in Figure 5.9 reconfirms that the performance of NLIM-EL-OD and NLIM-RPROP-OD models always dominate those of MA and HA. However, the performance of NLIM-MLR-OD models does not show significantly different what from those of MA and HA. That because of MLR is a simple machine learning based statistics. Therefore, it might share some characteristics of the other two statistics-based methods.

The MAPE performance metric of NLIM models is specified for individual traffic link categories. Figure 5.10 and Table 5.11 demonstrate the performance of NLIM in different machine learning techniques which are FF-EN-ANN, FF-RPROP-ANN and MLR respectively.

The RMSE and MAE performance metrics are not considered in the analysis of the performance of NLIM in different link categories. As mentioned in the theoretical framework chapter, both MAE and RMSE expose average model prediction error in units of the variable of interest. RMSE and MAE tend to be increased as the test instance size increases. In this experiment, the number of unseen labelled data for testing in each traffic link model often varies and the number of traffic links in each traffic link category and the travel time value in each traffic link is usually different. Hence, the RMSE and MAE performance metrics cannot be used to compare NLIM models' performance between target links. They are also not utilised to evaluate the performances of NLIM between link types.

The results in Figure 5.10 and Table 5.11 show that NLIM employed FF-EL-ANN and FF-RPROP-ANN can estimate more accurately travel time on motorway links and trunk links than on the rest in the link category. In more details, 75% of NLIM-RPROP-OD best models and 75% of NLIM-EL-OD best models of motorway link category have MAPE less than 11.5%. 100% of the NLIM-RPROP-OD models and 100% of the NLIM-EL-OD models in motorway link category have MAPE less than or equal to 50.51%. 75% of the NLIM-RPROP-OD best models and 75% of the NLIM-EL-OD best models in trunk link category have MAPE less than 13.5%. And 100% of the NLIM-RPROP-OD models and 100% of the NLIM-EL-OD models in

TABLE 5.11: MAPE (%) of NLIM models on unseen dataset (different machine learning techniques applied) with DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

Machine learning technique	(1)	(2)	(3)	(4)	(5)
Motorway links					
MLR	8.07	9.36	10.35	12.55	56.85
FF-EL-ANN	3.72	8.03	9.32	11.46	50.51
FF-RPROP-ANN	1.27	3.56	8.08	10.84	50.22
Trunk links					
MLR	12.65	13.21	14.37	17.28	30.90
FF-EL-ANN	8.13	9.13	10.26	13.43	24.86
FF-RPROP-ANN	5.42	8.65	9.70	12.80	21.76
Primary links					
MLR	10.83	12.69	16.62	19.35	37.55
FF-EL-ANN	6.83	9.68	12.63	14.87	21.91
FF-RPROP-ANN	5.17	8.73	11.28	14.48	25.53
A links					
MLR	10.68	13.98	18.10	28.07	233.41
FF-EL-ANN	6.69	12.55	16.87	23.08	237.88
FF-RPROP-ANN	3.77	11.50	15.70	24.22	164.75
B links					
MLR	11.21	16.57	19.12	24.14	298.28
FF-EL-ANN	6.41	11.66	14.88	21.20	317.67
FF-RPROP-ANN	5.87	11.41	14.52	20.82	149.32
Minor links					
MLR	9.29	14.71	17.98	22.66	200.98
FF-EL-ANN	3.07	10.49	12.93	17.23	153.28
FF-RPROP-ANN	2.50	10.00	12.87	17.31	442.24

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

trunk link category have MAPE less than or equal to 25%. The analysis is based on the upper-quartiles which represent 75% of the models and the upper-whiskers which explain the whole set (100%) of the models.

Summary of results

22053 traffic links have been included in the experiment area. The links fall into five traffic link categories. They are motorway, trunk, primary, A, B and minor link. The experiment area is a sub traffic network of the whole of Leicestershire traffic network. 13527 links which have data sparsity less than or equal to 99% have been modelled by NLIM-MLR, NLIM-EL and NLIM-RPROP. They are approximately 61.34% of total links in the experiment area. The performances of NLIM models have been evaluated.

The data of the links in the experiment area is sparse and irregular. The data sparsity in a traffic link is dependent on the link type. It increases statistically in the less busy

links. The lowest data sparsity is on motorway link, and the highest data sparsity is on the minor link.

The NLIM with different machine learning techniques have been trained and tested on the FCD dataset. The results show that NLIM-MLR-OD performs less accurate than NLIM-EL-OD and NLIM-RPROP-OD. The NLIM works better on motorway links and trunk links than the remaining. Performances of NLIM-EL and NLIM-RPROP models always dominate those of MA and HA models. Performances of NLIM-MLR models are slightly better compared to the achievements of HA and MA models.

In general, the performances of NLIM-MLR-OD, NLIM-EL-OD and NLIM-RROP-OD models are adequate and promising to use on travel time estimation on a large traffic network.

5.3 Similar model searching on FCD dataset

The original framework of Similar Models Searching (SMS) is precisely described in Section 4.5. The main idea of SMS is to discover a list of traffic link models which have the similarities with a target traffic link model. The training dataset of similarity models can be used as training dataset of a target traffic link model. The original training dataset of the target traffic models is reinforced with training labelled data from the similar models. SMS can be employed once initial NLIM models for a traffic networks are available. SMS methodology also requires diverse in NLIM models.

As mentioned in Section 5.2.4, the number of data samples in each model is not identical. The most of the motorway, trunk and primary traffic links have a large amount of travel time data that can be used for training and testing. However, the A, B and minor traffic links have a lower amount of labelled data. Hence, the performance of the NLIM models in those traffic links can be less accurate than NLIM models in motorway, trunk and primary traffic links. The similar NLIM models searching (SMS) is applied to solve the problem of low amount of labelled data for traffic links and to improve the performance of NLIM of the target links.

After using NLIM to train 338177 link models that are produced by 13527 link layouts in the subset of the Leicestershire traffic network, a collection of NLIM models is obtained.

TABLE 5.12: Statistics of the number of training samples which is increased by using SMS on experiment 4 dataset

Link type	Lower-whisker	Lower-quartile*	Median	Upper-quartile*	Upper-whisker
Motorway	0.0	0.0	5012.0	32722.0	136749.0
Trunk	0.0	0.0	4866.0	25521.5	125831.0
Primary	0.0	0.0	6091.5	28548.0	170304.0
A	0.0	811.0	14203.0	40278.0	192344.0
B	0.0	72.25	12192.0	33380.75	137737.0
Minor	0.0	0.0	11349.0	34866.0	207320.0

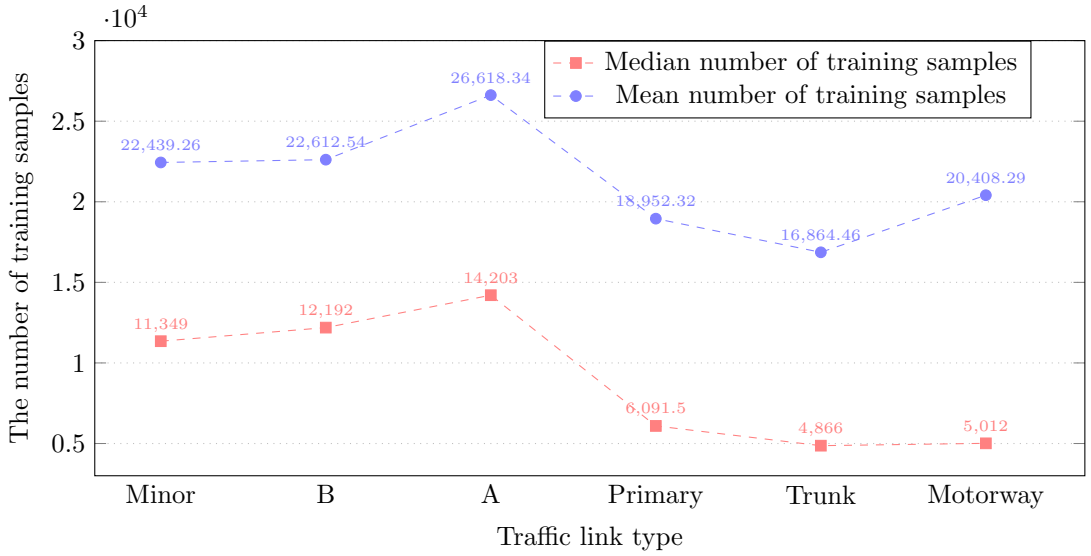
* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

Each target link has a combination of NLIM models. The smallest model size consists of the target link and one of the neighbouring traffic links. The largest model size is the full neighbouring link model that comprise the target link and all adjacent links. The diversity of model size and relationship between traffic links gives a possibility of having many potential similar models in the collection of NLIM models.

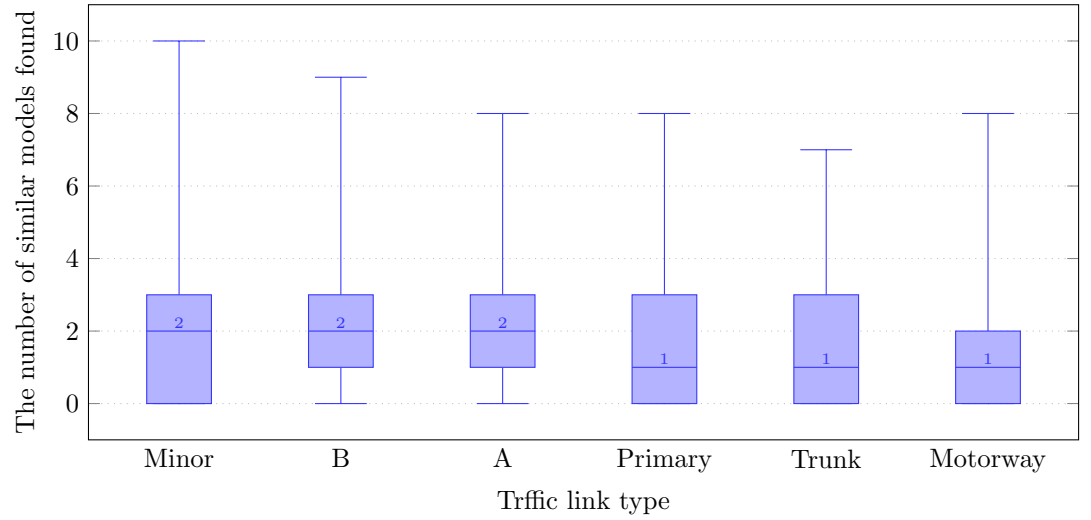
The SMS looks for similar models among 338177 NLIM travel time models. After similar models are found, the SMS does a further step to check if training data of the potential similar models can be adapted to enhance the performance of the selected NLIM model. The proposed method is described in Algorithm 4.5. By using SMS, the NLIM model does not only utilise data of its link model but also of other similar link models in the traffic network. This effectively strengthen the temporal and spatial relationship between travel times in links of the study link model.

This section presents the results obtained from applying the proposed methods in this work on the FCD dataset. The resulting of SMS performances are then compared to the NLIM performances on the same dataset. It is important to emphasise that the results presented in this section were obtained from unseen data, i.e. not the data used for training or validation dataset (in k-fold cross-validation).

Table 5.12 and Figure 5.11 show the statistic of the number of training data samples increasing (compared to the original training dataset) after SMS applied to the 338177 NLIM travel time models. The mean describes the average number of the samples amplified in the selected models by target link type. It can be seen that 25% of total NLIM models for the motorway, trunk, primary and minor links do not increase the number of training data when SMS is applied (lower-quartiles are zero in Table 5.12). 25% of the other link categories slightly raise the amount of training data when SMS is used. They are 811 and 72 for A and B links respectively.



(a) Traffic link types vs mean and median of the number of training samples which is increased by using SMS.



(b) Traffic link types vs five-number statistics of the number of similar models found by using SMS

FIGURE 5.11: Traffic link types vs the number of training samples which is increased and the number of similar NLIM models found by using SMS (Algorithm 4.5).

It also can be seen that SMS works more effectively on the minor, B and A links than on the primary, trunk and motorway links. The mean number of training samples increasing is 22439, 22612 and 26618 for minor, B and A links respectively while those on primary, trunk and motorway links are 18952, 16864 and 20408.

Figure 5.11 shows the means are higher than the medians. Hence, the distribution of the number of training samples increasing is skewed. It means that most training sample increasing is lower than the average. It also means by extension that some significant

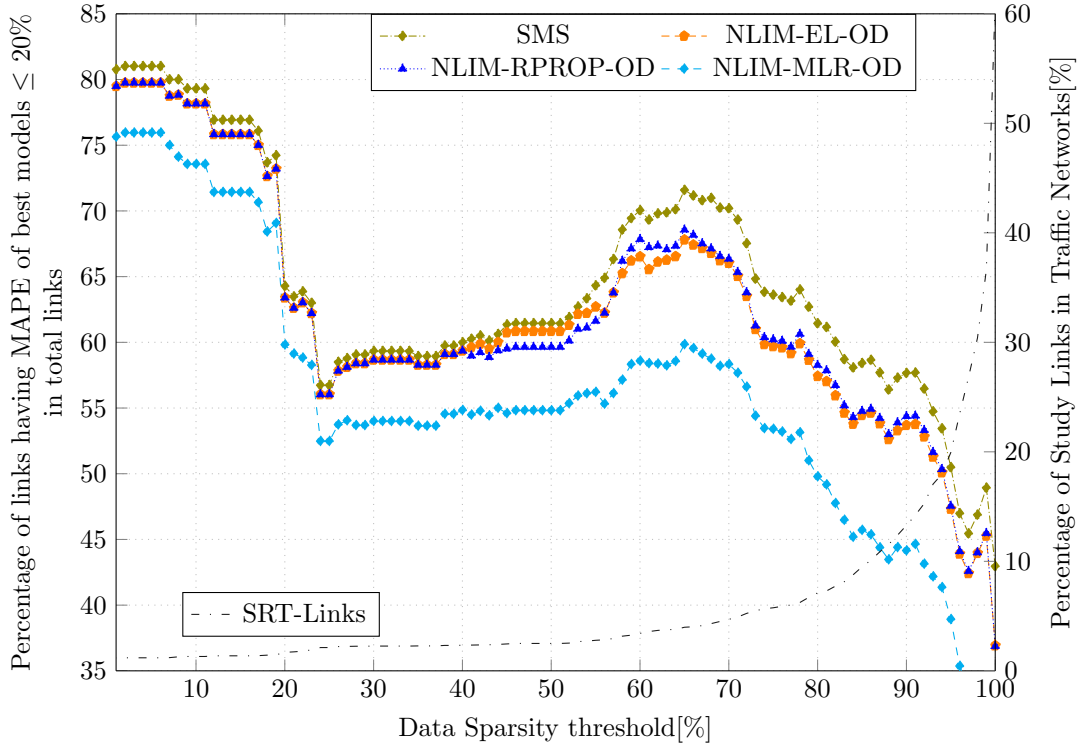


FIGURE 5.12: Percentage of links that have MAPE of the best model less than or equal to 20% vs sparsity threshold achieved by Neighbouring link inference method with similar model searching (SMS), NLIM employed FF-EL-ANN (NLIM-EL-OD), NLIM employed FF-RPROP-ANN (NLIM-RPRO-OD), NLIM employed MLR (NLIM-MLR-OD) on the unseen data. Outliers are identified and removed from the unseen test data by applying Algorithm 4.3.

amount of training samples increasing are big enough to move the mean despite there being more number of samples increasing are small.

Data sparsity of link categories is shown in Table 5.10 in the previous section. The statistics in Table 5.10 indicate that data are more sparse on the urban traffic links than on the motorway links. The lower quartile, the median and the upper quartile of data sparsity on motorway links are 54.9%, 19.5% and 10.5% respectively. Their values are significantly higher on the urban links. The lower quartile, the median and the upper quartile of data sparsity of data sparsity in urban links are often higher than 20% compared to those of motorway link.

A data sparsity threshold (SRT) is set before an experiment is conducted. Any link in the traffic network that has data sparsity less than or equal SRT value will be involved in the investigation. Hence, the number of traffic links, traffic link layouts, possible models in the experiment are dependent on the SRT value. From now on, traffic links involved in the investigation at specific SRT value is named SRT-Links.

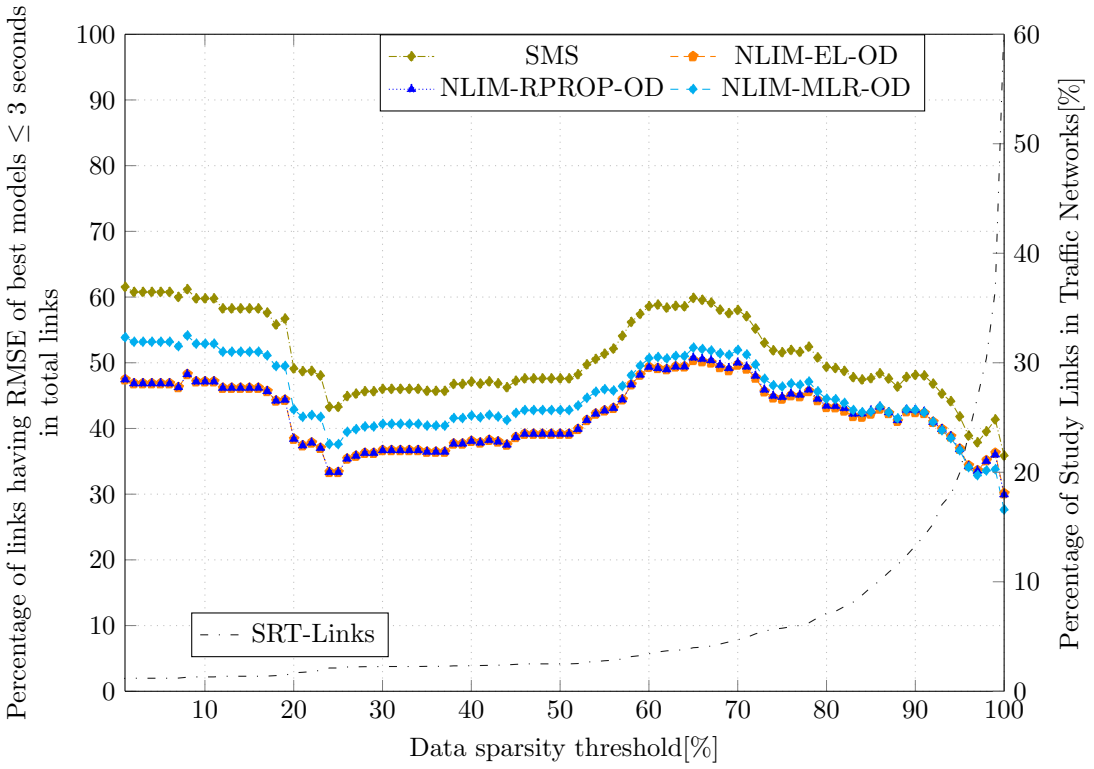


FIGURE 5.13: Percentage of links that have RMSE of the best model less than or equal to 3 seconds vs sparsity threshold achieved by Neighbouring link inference method with similar model searching (SMS), NLIM employed FF-EL-ANN (NLIM-EL-OD), NLIM employed FF-RPROP-ANN (NLIM-RPRO-OD), NLIM employed MLR (NLIM-MLR-OD) on the unseen data. Outliers are identified and removed from the unseen test data by applying Algorithm 4.3.

According to the reuslts in previous section (Figure 5.9(a), 5.9(b) and 5.9(c)), if RMSE, MAE and MAPE are 3.0, 3.0 and 20% respectively, the NLIM can cover around greater than 75% the total number of target links in the traffic network. Hence, those numbers are choose to evaluate the number of target links that will be cover by NLIM and SMS per data sparsity threshold.

In Table 5.10, it can be seen that the data sparsity of links in the experiment traffic network varies depending on the link types. It is in a range from 0% to 100%. The effect of data sparsity threshold of the experiment on the performances of SMS, NLIM-EL-OD, NLIM-RPROP-OD and NLIM-MLR-OD have been investigated. The results in Figure 5.12, 5.13 and 5.14 clearly show that the data sparsity threshold has an impact on the number links involved in the experiment.

For the MAPE performance metric, it can be seen in the Figure 5.12 when the data sparsity threshold was set to a shallow value (i.e. SRT = 0%-50%), the number of

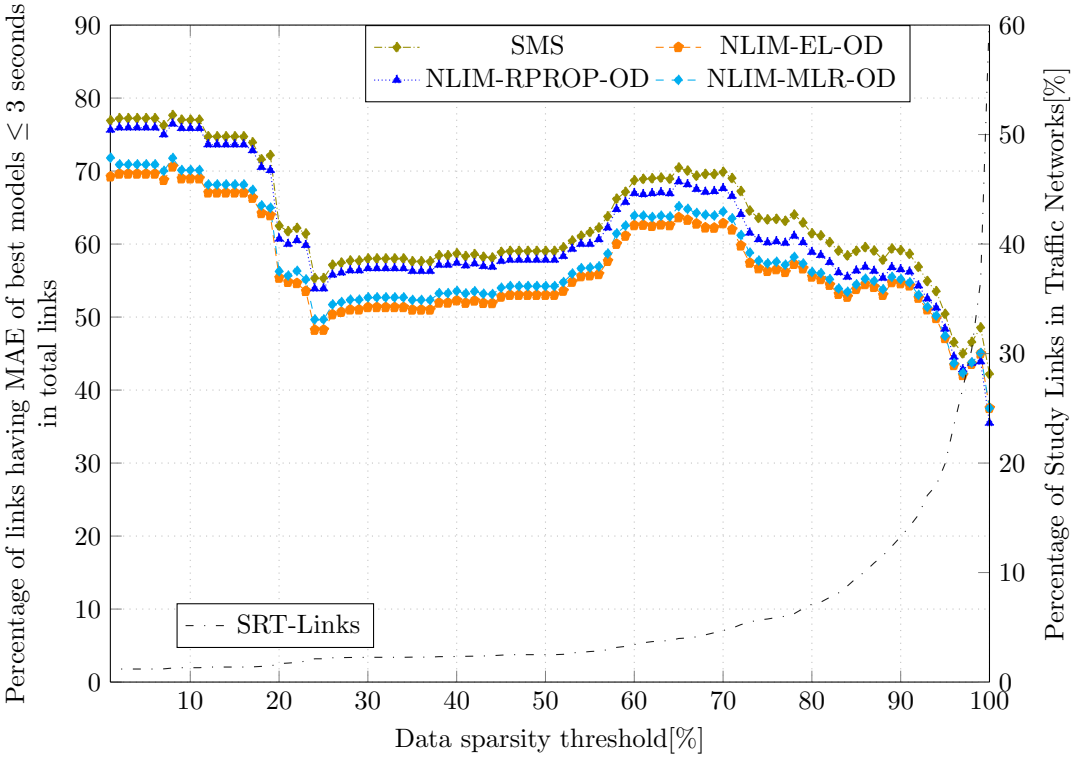


FIGURE 5.14: Percentage of links that have MAE of the best model less than or equal to 3 seconds vs sparsity threshold achieved by Neighbouring link inference method with similar model searching (SMS), NLIM employed FF-EL-ANN (NLIM-EL-OD), NLIM employed FF-RPROP-ANN (NLIM-RPRO-OD), NLIM employed MLR (NLIM-MLR-OD) on the unseen data. Outliers are identified and removed from the unseen test data by applying Algorithm 4.3.

TABLE 5.13: Statistics of the performance metrics of NLIM and SMS models on FCD dataset (different machine learning techniques applied) with DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

Machine learning technique	(1)	(2)	(3)	(4)	(5)
RMSE [seconds]					
MLR	0.12	1.62	4.18	12.63	153.45
FF-EL-ANN	0.04	1.41	3.13	7.85	548.28
FF-RPROP-ANN	0.04	1.48	3.25	8.10	548.15
SMS	0.02	1.03	2.37	6.06	275.38
MAE [seconds]					
MLR	0.15	0.84	1.96	5.10	830.26
FF-EL-ANN	0.02	0.76	1.63	3.54	380.59
FF-RPROP-ANN	0.02	0.80	1.72	3.74	424.95
SMS	0.01	0.53	1.17	2.63	130.96
MAPE [%]					
MLR	8.03	18.24	24.69	40.31	7894.34
FF-EL-ANN	3.07	12.72	17.15	25.78	910.30
FF-RPROP-ANN	1.26	13.42	18.08	27.14	3177.59
SMS	0.804	9.52	13.5942	19.56	428.90

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

SRT-Links is less than 5%. The number of SRT-Links is dramatically increased from over 10% to over 60% when SRT value rises from 80% to 99%. However, the number of the best traffic link models that have MAPE less than or equal to 20% in STR-Links is a drop down from approximate 70% to under 20% including SMS. But the number of SMS models which have MAPE less than or equal to 20% is always higher from 5% to 10% than those of NLIM-EL-OD, NLIM-RPROP-OD and NLIM-MLR-OD.

For the RMSE and MAE performance metrics, the same trends are observed in Figure 5.13 and 5.14. The number of the best traffic link models that have RMSE less than or equal to 3 seconds is also a notable decrease from approximate 60% to under 35%, and the number of the best traffic link models that have MAE less than or equal to 3 seconds is a noticeable decline from approximate 70% down to under 30% when SRT value rises from approximate 70% to 99%. Still, the number of SMS models which have RMSE less than or equal to 3 seconds and have MAE less than or equal to 3 seconds are always significant higher than those of NLIM-EL-OD, NLIM-RPROP-OD and NLIM-MLR-OD.

The performances of the SMS was evaluated regarding a very high data sparsity (SRT=99%) to show the ability of SMS in modelling the links. At data sparsity threshold value 99%, the number of SRT-Links is 13527, and the number of traffic link models is 338177. According to the statistics in Table 5.13, more than 75% of the best SMS models have MAPE less than or equal to 19.56%.

It also can be seen in the Figure 5.12, 5.13 and 5.14 that, NLIM and SMS have the best performance at SRT=70% and the number of target links accordingly having accurate travel time estimation is approximate 10800 (50%*98%*22053) traffic links which are approximate 80% of sufficient traffic links involved in the experiment.

Focussing closer to the results, the performance of the SMS methods are evaluated for each specific link category which is defined on Table 2.1. A selected traffic link layout can be modelled by multiple NLIM models. Therefore it also can be modelled by multi SMS models. The performances of SMS models and NLIM models on the traffic link layout are compared based on the performance of the best SMS and the best NLIM using MAPE performance metric.

Figure 5.15 and Table 5.14 present the relationship between density of the best SMS and the best NLIM models in motorway, trunk, primary, A, B and minor link category,

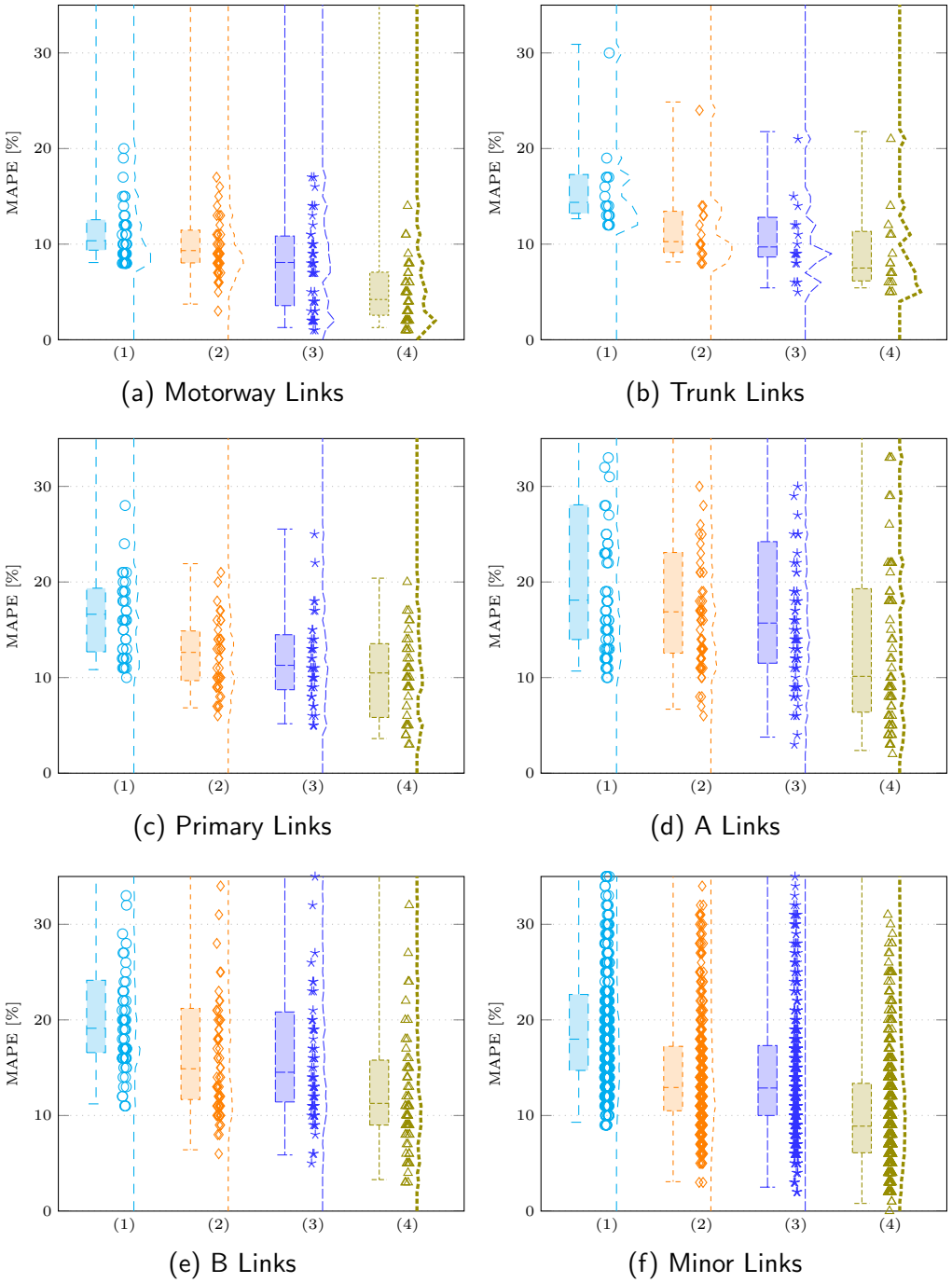


FIGURE 5.15: Density of the best NLIM models, different machine learning techniques applied, of Motorway (a), Trunk (b), Primary (c), A (d), B (e) and Minor (f) links and their MAPEs [%] achieved on unseen data. Sub-figures are in the same scale. (1) is for MLR, (2) is for FF-EL-ANN, (3) is for FF-RPROP-ANN and (4) is for SMS. The density of best models in each method is presented by boxplot (lower whisker, lower quartile, median, upper quartile, upper whisker), visualisation of the actual individual model and the histogram of the models respectively. Some high MAPE data points are out of the figure, hence corresponding upper-whiskers can not be shown.

and their MAPE achieved on the unseen data, respectively. The NLIM employs MRL, FF-EL-ANN and FF-RPROP-ANN. The SMS uses the temporal and spatial relationship

TABLE 5.14: Statistics of the MAPE (%) of NLIM models on unseen dataset (different machine learning techniques applied) with DR-M-GMM: (1) Lower-whisker, (2) Lower-quartile*,(3) Median, (4) Upper-quartile*, (5) Upper-whisker

Machine learning technique	(1)	(2)	(3)	(4)	(5)
Motorway links					
MLR	8.07	9.36	10.35	12.55	56.85
FF-EL-ANN	3.72	8.03	9.32	11.46	50.51
FF-RPROP-ANN	1.27	3.56	8.08	10.84	50.22
SMS	1.27	2.55	4.21	7.06	51.77
Trunk links					
MLR	12.65	13.21	14.37	17.28	30.90
FF-EL-ANN	8.13	9.13	10.26	13.43	24.86
FF-RPROP-ANN	5.42	8.65	9.70	12.80	21.76
SMS	5.42	6.15	7.49	11.33	21.76
Primary links					
MLR	10.83	12.69	16.62	19.35	37.55
FF-EL-ANN	6.83	9.68	12.63	14.87	21.91
FF-RPROP-ANN	5.17	8.73	11.28	14.48	25.53
SMS	3.62	5.83	10.48	13.54	20.39
A links					
MLR	10.68	13.98	18.10	28.07	233.41
FF-EL-ANN	6.69	12.55	16.87	23.08	237.88
FF-RPROP-ANN	3.77	11.50	15.70	24.22	164.75
SMS	2.37	6.38	10.13	19.29	112.59
B links					
MLR	11.21	16.57	19.12	24.14	298.28
FF-EL-ANN	6.41	11.66	14.88	21.20	317.67
FF-RPROP-ANN	5.87	11.41	14.52	20.82	149.32
SMS	3.29	9.00	11.27	15.78	220.50
Minor links					
MLR	9.29	14.71	17.98	22.66	200.98
FF-EL-ANN	3.07	10.49	12.93	17.23	153.28
FF-RPROP-ANN	2.50	10.00	12.87	17.31	442.24
SMS	0.80	6.100	8.88	13.36	149.83

* Lower-quartile and Upper-quartile express 25% and 75% of total models respectively.

which is modelled by NLIM-RPROP-OD for the searching similar model process.

It can be seen that SMS always outperforms NLIM on all link types, especially on minor traffic link types such as B and minor links. The historical travel time data collected on urban traffic network are contaminated with noise, and as it can be seen in Figure 5.12, 5.13 and 5.14, the travel data used in this research have a very high data sparsity for the urban traffic links.

The majority of links in the urban traffic network have data sparse rates greater than 70%. Especially on minor links, for which the data sparsity is greater than 90%. It consequently makes the urban traffic links more challenging to model compared to the

motorway links. However, SMS can reinforce NLIM working more effectively on less busy links such as B, and Minor links.

MAPE of the best SMS of A, B and Minor links are reduced (Figure 5.15 and Table 5.14). From Table 5.14, one can observe that the number of minor links having MAPE less than 12% is increased from approximately 50% to above 75%. And the number of B links haveing MAPE less than or equal to 15% is also raised from 50% to 75%.

It appears that reinforcement training data from similar NLIM models support more information for a target NLIM model to learn precisely the spatial and temporal relationship between travel times in links in a traffic link especially for a dataset with variability, irregularity and sparsity which are often characteristics of urban travel time.

Summary of results

Improving the performance of NLIM in minor links which have datasets with high data sparsity and irregularity links has been considered in this section. The main idea is to adapt travel time data of similar NLIM models to improve a selected NLIM model. The similar model searching (SMS) has been evaluated on FCD dataset. NLIM was firstly used for traffic links to create a collection of NLIM models. Then, the similar model searching method was applied. Results show that SMS is capable of improving the performance of NLIM on learning the temporal and spatial relationship between the travel time of a target link and travel time of its neighbouring link despite the high data sparsity and irregularity of the dataset.

The number of training samples is increased where SMS has been applied. SMS can increase the amount of training samples on the minor, B and A links but less so on the primary, trunk and motorway links. The number of similar models of each selected traffic link model varies. It ranges from 0 to 10 similar models. The average for the amount of the similar models found by SMS is 2 and 3 for each traffic link category.

The performance of SMS always dominates the performance of NLIM on all traffic link categories. Especially, SMS works more effectively on minor links. 75% of SMS models can produce travel time data which have MAPE error less than 20%. 50% of SMS

models can estimate near real-time travel time that has MAPE less than 13.5%, and 25% of SMS models can calculate near real-time travel that has MAPE less than 9.52%.

It can be concluded that reinforcement training data from similar NLIM models provide more information for SMS to learn the temporal and spatial relationship between the travel time of links supporting the high variability of urban traffic travel time and high data sparsity. It also can be concluded that SMS outperforms the NLIM-MLR-OD, NLIM-EL-OD and NLIM-RPROP-OD.

5.4 Chapter summary

NLIM with different machine learning techniques performs well on four different datasets. NLIM shows its ability to learn the temporal and spatial relationship between travel times on links not only on the artificial datasets but also on the complex historical travel time datasets. NLIM also shows the ability to deal with sparse and irregular data.

The obtained results also show that different traffic link models have a varying degree of relation, and NLIM works effectively on all traffic link categories in the real datasets. However, NLIM performs better on motorway links and other major links than the remaining. NLIM with any machine learning techniques always dominates MA and HA method. The results also show that the training time of SVR is a significant increase when the number of training instances is increasing. It reconfirms that SVR is not suitable for NLIM in large travel time datasets.

The number of labelled data is increase where SMS is applied. NLIM working with SMS can approximate travel time data more accurate compared to those on NLIM alone. The number of labelled data is increased more on the minor, B and A links than those on the primary, trunk and motorway links. The number of similar models of each selected traffic link model found by SMS varies. It ranges from 0 to 10 similar models.

Chapter 6

Conclusions, Recommendations and Future work

This chapter gives a general overview of the major contributions of the research and summarises the main points of various chapters. It also offers practical implications resulting from the findings and sparks a high level of general interest and motivates further research.

6.1 Conclusion

Three hypotheses were presented in Chapter 1:

Hypothesis 1: *Relationships between temporal and spatial properties of travel times in neighbouring traffic links can be learnt to enhance the estimate of travel time of a target link.*

A traffic network was decomposed into traffic links layout consisting of a targeted link and adjacent links. The link layouts were continuously decomposed into traffic link models. The target link is a link where traffic-related information needs to be determined. The neighbouring links are links that might contain information that can be used for the traffic parameters estimation. A traffic link model consists of a target link and at least one of the adjacent links of a traffic link layout. A travel time

estimation technique, namely the neighbouring link inference method (NLIM) was proposed method to model traffic links in a traffic link layout.

Many experiments on four different datasets were carried out to test Hypothesis 1. Each dataset was divided into a training dataset and a test dataset. The performance metrics including RMSE, MAE and MAPE were evaluated on unseen data. Three machine learning techniques were utilised within NLIM including multi-linear regression (MLR), artificial neural network (ANN) and support vector machine (SVM). Machine learning models were always trained on training datasets using 5-folds cross-validation. The training process of individual machine learning technique was separately designed due to the different stop criteria and different hyper-parameters optimisation of the corresponding machine learning technique. The best model represents the performance of NLM models in a link layout. The best model is one within models which is generated from the traffic link layout, and it has the smallest performance (a measure of error) metric. The performance of the best model in traffic layouts was used in the analysis.

• Artificial dataset:

NLIM shows the effectiveness of modelling the relationship between temporal and spatial relationships in traffic links of a traffic link model. They can produce travel time estimation with low errors. The results from testing NLIM on the artificial data set gave many important insights. However, the artificial dataset was not sufficient to proof NLIMs effectiveness. Hence NLIM was also applied on other, measurement based datasets.

• Sumo dataset:

The second synthetic dataset used in this thesis was the SUMO dataset. The links in this experiment do not classify into different link types due to a lack of the information of link types in the simulation scenario. Three performance metrics which are given by different machine learning techniques within NLIM are notably small. The performance of NLIM models is similar to the achievements of NLIM on the Artificial dataset.

• WebTRIS dataset:

A similar analysis was undertaken using the real travel time dataset from WebTRIS. The performances metrics of NLIM models showed higher values (measures of error) than those on two synthetic datasets (artificial dataset and SUMO dataset). However,

the performance metrics of NLIM were low values on the unseen data. They showed conclusively that learning the temporal and spatial relationships between travel times in links is possible. The vast majority of the NLIM models estimated effectively travel time for target links. One reason for the large number of models was that only motorway and A links were included in the experiment.

• FCD dataset:

The FCD dataset consists of motorway, trunk, primary, A, B and minor link categories (B and minor categories are minor link type). The histogram differed between traffic links and the range of travel times in the traffic links varied. The distribution of travel times on traffic links had long right tails and different scales. There were some very high travel time values on all studied traffic links. Traffic links data in FCD dataset showed to be sparse and irregular. The number of travel time samples differed strongly by the time of day. The data sparsity in a traffic link was dependent on the link type. It is larger in less busy links. The lowest data sparsity was on motorway link, and the highest data sparsity was on minor links. NLIM worked better on motorway links and trunk links than on other links. Performances of NLIM models always dominated those of two typical standard methods used in this field: Moving Average (MA) and Historical Average (HA), since MA and HA only used the temporal property of historical travel times on a target link to estimate its current travel time.

NLIM effectively learned the temporal and spatial relationships between travel times in neighbouring links over the four different datasets hence confirming hypothesis 1.

Hypothesis 2: *Relationships between temporal and spatial properties of travel times in a traffic link model can be similar with those in other traffic link models in the same traffic network.*

The introduced method Similar Model Searching (SMS) discovered a list of traffic link models which have the similarity with a selected traffic link model. The selected model is similar to a target link model if they satisfy two conditions:

• The number of neighbouring links in the selected model is equal to the number of neighbouring links in the target link model.

• The relationship between the neighbouring links and a target link in the selected model is similar to the relationship between the neighbouring links and the target link in the target link model.

The condition 2 needs to use the selected NLIM model to examine the target NLIM model. The link length of a link in the traffic link model, the shape of the traffic link layout and the shape of the traffic link model were not directly considered as conditions in similar model searching because they are already included in the link relationships.

After using NLIM to train link models, a collection of NLIM models was obtained. Each target link had a combination of NLIM models. The smallest model size consists of the target link and one of the neighbouring traffic links. The largest model size is the full neighbouring link model that comprise the target link and all adjacent links. The diversity of model size and relationship between traffic links gave many potential similar models in the collection of NLIM models.

After SMS was applied, a number of similar models was found in the traffic network for both major and minor link type. The number of similar models of each selected traffic link model varied. It ranged from 0 to 10 similar models. The average for the amount of the similar models found by SMS is 2 and 3 for each traffic link category. The existing of similar models confirms hypothesis 2.

Hypothesis 3: *Use of labelled data from similar traffic models for a selected traffic model can improve the performance of the traffic model regarding travel time estimation.*

The utilisation of the proposed SMS methodology significantly increased the number of training samples subsequently used for modelling the selected traffic link models. The distribution of the number of training samples increasing is skewed. The number of training samples increasing was lower than the average. Several significant increasing in amounts of training samples were remarkable. The results showed that SMS worked more effectively on the minor, B and A links than on the primary, trunk and motorway links. The number of SMS models was always higher those of NLIM which have the same performance metrics. SMS outperformed NLIM on all link categories, especially on minor traffic link types such as B and minor links. It appeared that reinforcement training data from similar NLIM models support more information for a target NLIM model. Hence, NLIM was able to learn precisely the spatial and temporal relationship

between travel times in neighbouring links in a traffic link model, especially for a dataset with variability, irregularity and sparsity which are often characteristics of urban travel time. This improved the NLIM models' performance, which confirms hypothesis 3.

6.1.1 Findings

In the following sections, the findings of a number of experiments will be discussed.

Temporal and spatial relationships between travel times on traffic links can be used as source information to estimate travel time for both urban and motorway traffic networks

The application of NLIM to four datasets clearly demonstrated the above finding. NLIM employs different machine learning techniques on the datasets to investigate the correlations between temporal and spatial relationships between travel times on neighbouring traffic links. The details of the methodology were described in Chapter 4. The results in Chapter 5 demonstrated that there are temporal and spatial relationships of travel times between adjacency links in a traffic link layout. The obtained results have a similar performance for the four different datasets in terms of adopted metrics. The relationships are different in different traffic link models.

The first experiment was carried out with the artificial dataset which was generated based on the BPR function. The details of how to generate the dataset were described in Chapter 4. Traffic link BD and EG are setup as dependent links of traffic link DE. The setup for the scenario is that BD is the rear traffic link, and EG is the front traffic link of the traffic link DE, traffic flow from BD feeds in the traffic link DE and traffic flow from DE feeds in the links EG. The results in Chapter 5 show that traffic link models which contain either BD or EG or both of them can be effectively used to estimate travel time for target traffic link DE. In other words, the temporal and spatial relationships are efficiently learnt by all used machine learning techniques. Although other traffic models in the experiment traffic link layout have the higher number of labelled data for training the models, the results showed those models have significantly less performance in travel time estimation for the target traffic link compared to those containing the dependent traffic link BD and EG.

In the second experiment, NLIM was applied to another artificial dataset obtained from SUMO simulation of TAPAS Cologne scenario. The dataset was presented in Chapter 4. The results in Chapter 5 demonstrated that NLIM is effective in modelling the temporal and spatial relationship in the traffic link models. The results also showed that different traffic link models have a varying degree of relation. The finding on the dataset also confirms the characteristics of the support vector machine technique that it spends more time to generalise the pattern of a large training data instances as mentioned in Chapter 4. Models generated with SVR also exhibit less accurate estimates than other used machine learning techniques.

In the third experiment, the same methodology was applied to a travel time dataset of motorways and A links located in East Midlands, UK. The dataset was described in Chapter 4. The results presented in Chapter 5 demonstrated that the proposed method is able to work with real-world data and achieve more than satisfactory performance. It was also found that the training time of SVR significantly increased when the number of training instances increased. Thereby, SVR is not recommended for NLIM in large dataset scenarios.

22053 traffic links in a sub-area of Leicestershire traffic network were used in the last experiment. The links fall into five traffic link categories: motorway link, trunk link primary link, A link, B link and minor link. The case study area was a part of the traffic network of the whole of Leicestershire with travel times collected from floating cars. The dataset has 13527 links which have data sparsity less than or equal to 99%. These links were involved in the experiment. They represent approximately 61.34% of total traffic links in the experiment area. The data sparsity of traffic link is dependent on the link type. It increased statistically in the less significant links. The lowest data sparsity was found on motorway links, and the highest data sparsity was found on the minor links. The details of the dataset were shown in Chapter 4. The results presented in Chapter 5 indicate that NLIM still works effectively on all traffic link categories in the real dataset. However, the NLIM works better on motorway links and trunk links than the minor links. The NLIM with any machine learning techniques always dominated MA (Moving Average) and HA (Historical Average) methods.

In overall, the performance of NLIM is more than adequate and NLIM can be used for travel time estimation in an extensive traffic network. In other words, the travel

time of a targeted link can be estimated by using relation which is temporal and spatial relationships between travel times of traffic links in a traffic link model learnt by NLIM.

Travel time in different models can be in different scales but the temporal and spatial relationship in traffic link models can be similar

In an extensive traffic network such as the traffic network in the experiment 4 in Chapter 5, the lower bound and upper bound of the travel times in links can be significantly different as they can be seen in Figure 5.6. In the original scale of travel times in traffic link models, it was difficult to recognise the similarity of temporal and spatial relationship between two different traffic models, however the use of the proposed SMS on the models which travel times on links were normalised into scale of [0,1] was able to find several similar traffic link models.

The original framework of the Neighbouring Link Inference Method with Similar Models Searching (SMS) is described in details in Section 4.5. The main idea of Similar Model Searching (SMS) was to discover a list of traffic link models which have the similarity with a target traffic link model. Hereafter, the training dataset of similarity models can take part in the training dataset of selective traffic link model. SMS is not able to work as a stand-alone methodology. It must be used after NLIM. SMS methodology also requires a diversity of NLIM models. Therefore, FCD dataset was an excellent case study to demonstrate the ability of SMS.

The results illustrated and discussed in Chapter 5 showed that the number of training samples significantly increased where SMS was applied. SMS can create more training sample on the minor, B and A links than on the primary, trunk and motorway links. The results also showed that the number of similar models of each selected traffic link model varies. It ranges from 0 to 10 similar models. The average amount of the similar models found was 2 and 3 for major and minor link type, respectively.

The performance of SMS always dominated results obtained by NLIM on all traffic link categories. Especially, SMS worked more effectively on less major links. Most of the SMS models can produce travel time data in near-real-time. 50% of the SMS models can estimate near real-time travel time that has MAPE less than 13.5%, and a quarter of SMS models can calculate near real-time travel time that has MAPE less than 9.52%.

Gaussian mixture model can be used to detect outliers of travel time data

The DR-M-GMM that was described in Chapter 4 was applied to four different travel time datasets in Chapter 5, and the results demonstrated that the number of outliers detected is varied over the different datasets. The application of multivariable Gaussian mixture model on detection and removal of outliers demonstrated its practicability. It is believed this algorithm can be applied not only for traffic-related datasets but also in other applications.

The DR-M-GMM was designed not only for travel time outliers detection on a single link but also for a traffic link model. The DR-M-GMM can detect the travel time outliers in a combination of travel times in traffic link model. The NLIM applied over the four datasets indicated that travel time estimation was more accurate when it incorporated the DR-M-GMM.

6.1.2 Contributions

Major contributions

The major contributions of the thesis are summarised below:

1. A novel methodology to estimate travel times in complex and dynamic transportation networks was presented. The methodology, namely Neighbouring Link Inference Method (NLIM), employs machine learning techniques to learn temporal and spatial dependencies between traffic links resulting in a model of a transportation network. The developed model can be used to estimate travel times for traffic links. One of the advantages of this method is its capability to perform well on datasets with high sparsity and irregularity. The datasets or data feeds often have entries only for major links or entries collected at highly irregular intervals. Having embedded knowledge about the temporal and spatial dependencies between travel times of a target link and its adjacent links the model can overcome sparsity in input data and provide accurate estimations. Details were given in Chapter 4.

2. A novel methodology, namely similar model searching (SMS) was introduced. The proposed method can enhance the learning performance of machine learning technique of temporal and spatial dependencies of travel times on traffic links' datasets with high sparsity and irregularity. SMS greatly improves the estimation capabilities of the final models. The main idea of SMS is to discover a list of traffic link models which are similar to the target traffic link model. After that, the labelled data of similarity models together with the target model training dataset is utilised as the new labelled dataset for training the target model. Details were given in Chapter 4.

3. A novel application of outliers detection and removal using multivariate Gaussian mixture models was presented. An outlier is an observation point that is distant from other observations. The outliers influence statistical characteristics, and they may lead to erroneous conclusions. To remove outliers in a matrix, the m-GMM is used to cluster the rows of a matrix into k row distributions where each element in a row is a variable of the multivariate. Structure and size of the rows distributions (clusters of rows) are indicators to detect travel time outliers. Details were given in Chapter 4.

Part of this research was published in Vu et al. (2016, 2017). Details are presented in Appendix A.

Subsidiary contributions

The subsidiary contributions of the thesis are summarised as follows:

1. A comprehensive literature review which provided the context and motivation for this research. There were six main topics that were disused and analysed. The investigation was stressed on modelling travel time from sparse data with low sampling rates using machine learning techniques in extensive urban traffic networks. A comprehensive evaluation of the strengths and weaknesses of the existing travel time estimation methodologies was given. The related literature has also been reviewed to identify the gaps in previous research and to set a background of the study. Details were given in Chapter 2 and Chapter 3.

2. An insight into sparse and noisy traffic data. Many experiments and data analyses were conducted to give an insight into sparse and noisy data. It provided critical information in order to select suitable techniques for travel time models and to select an appropriate type of intelligent transport system application to which the proposed methodologies intend to be integrated. Details were given in Chapter 4 and Chapter 5.

3. The application and evaluation of the developed methods on different datasets was presented. It uses temporal and spatial dependencies of traffic links and their travel times to approximate travel time data which are currently not available. For this study, the methods were implemented and subsequently evaluated in four distinct case studies. Chapters 4 and 5 and Appendix B gave a partial insight to some of the implementation issues and recommendation for future applications to another case studies.

6.2 Recommendations and Future work

In the following sections, a number of recommendations and possible future works are described.

Apply NLIM and SMS into an ITS application: The first application of NLIM and SMS in Chapter 5 to four different datasets have proven that it is adequate to estimate travel times in a near real-time for target links using the temporal and spatial relationship between historical travel time in traffic links. There is scope to investigate the complexity of modelling a very large traffic network such as Leicestershire traffic network. An outcome of the investigation can be used to develop an "elasticity" model for entire traffic network which can represent traffic conditions of links by using floating car data in some traffic links.

Extend the SMS method: The proposed SMS looks for NLIM similar models based on the temporal and spatial relationship between travel times of links that have been learnt by NLIM. As mentioned in Chapter 4 and Chapter 5, SMS is not the stand-alone method, and it needs to be applied after NLIM. There is a demand to make SMS independent from NLIM. In other words, there is avenue to be investigate that may give

more insight about traffic model similarity, and after that, the findings can be applied into SMS to search for similar traffic link models.

Extend the knowledge of temporal and spatial relationships between travel time on traffic links using other techniques: Other techniques such as transfer learning and semi-supervised learning, which may benefit to modelling relationship of links from high data sparsity, are also merit for further investigation.

Apply deep learning to travel time estimation: As mentioned in Chapter 2, the recent developments in technology, particular in the industrial 4.0 revolution gives the age of big-data transportation which provides researchers with a fantastic opportunity to expand the knowledge of the travel time estimation domain. Multiple-layer architectures or deep architectures in Deep learning algorithms can be used to extract inherent features in big-data from the highest level to the lowest level Lv et al. (2015). They can be used to discover huge amounts of structure in the big-data. As travel time estimation process is naturally complicated, deep learning techniques can represent traffic parameters without prior knowledge, which has a satisfying performance for travel time estimation.

Appendix A

Published Papers

Following is a list of the papers presented during the period of this research:

1. Luong Vu, Benjamin Passow, Daniel Paluszczyszyn, Lipika Deka and Eric Goodyer (2017). Neighbouring Link Travel Time Inference Method Using Artificial Neural Network. 2017 IEEE Symposium Series on Computational Intelligence (SSCI).

2. Luong Vu, Benjamin Passow and Eric Goodyer (2016). Urban Road Traffic Link Travel Time Estimation Based on Sparse Data. In Computer system engineering: Theory and Application: International Student Workshop 2016. Miedzygorze, Poland.

Parts of the research conducted will be shortly submitted to a journal for review:

1. Luong Vu, Benjamin Passow, Daniel Paluszczyszyn, Lipika Deka and Eric Goodyer. Estimation of Travel Times for Minor Roads in Urban Areas Using Sparse Data.

2. Luong Vu, Benjamin Passow, Daniel Paluszczyszyn, Lipika Deka and Eric Goodyer. Estimation of Travel Time using Temporal and Spatial Relationships in Sparse Data.

Appendix B

Details code map for TravelTimeEstimator solution

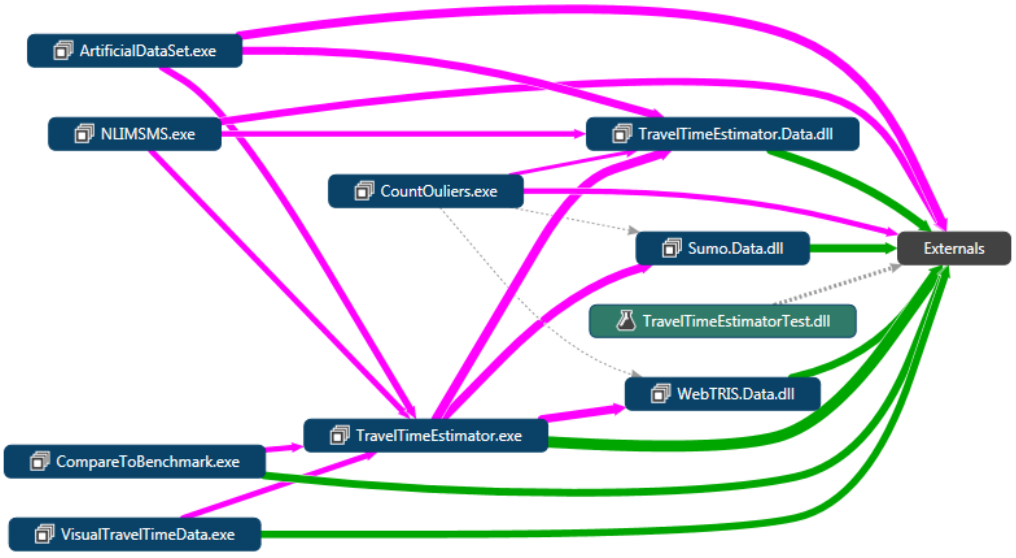


FIGURE B.1: Code Map for TravelTimeEstimator

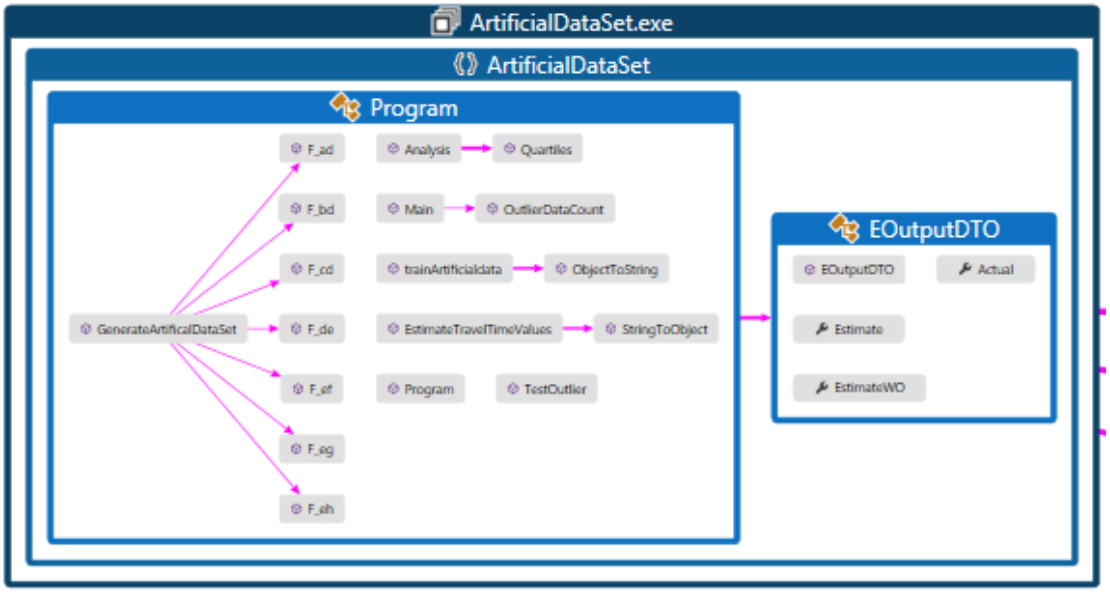


FIGURE B.2: ArtificialDataSet code diagram

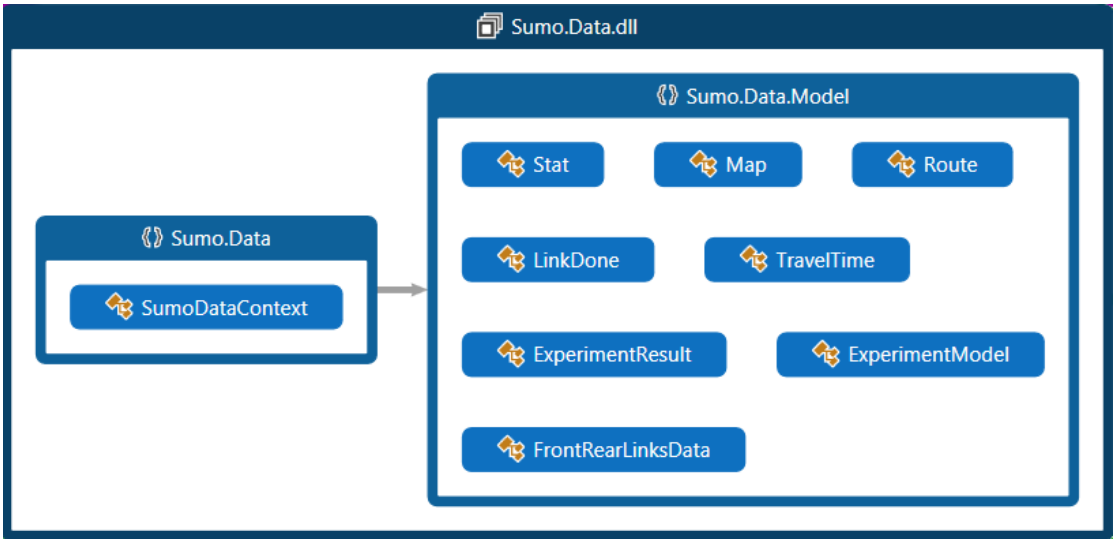


FIGURE B.3: Sumo.Data code diagram

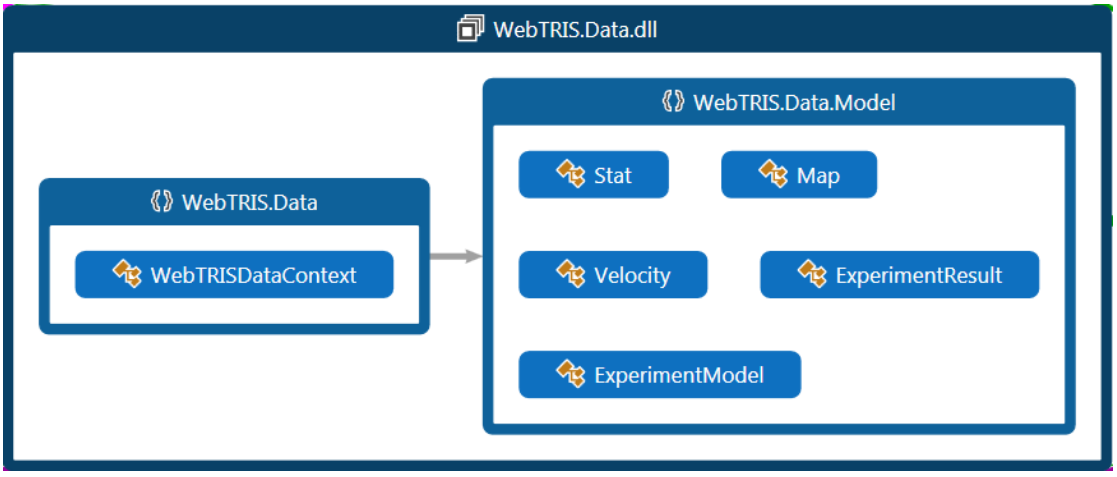


FIGURE B.4: WebTRIS.Data code diagram

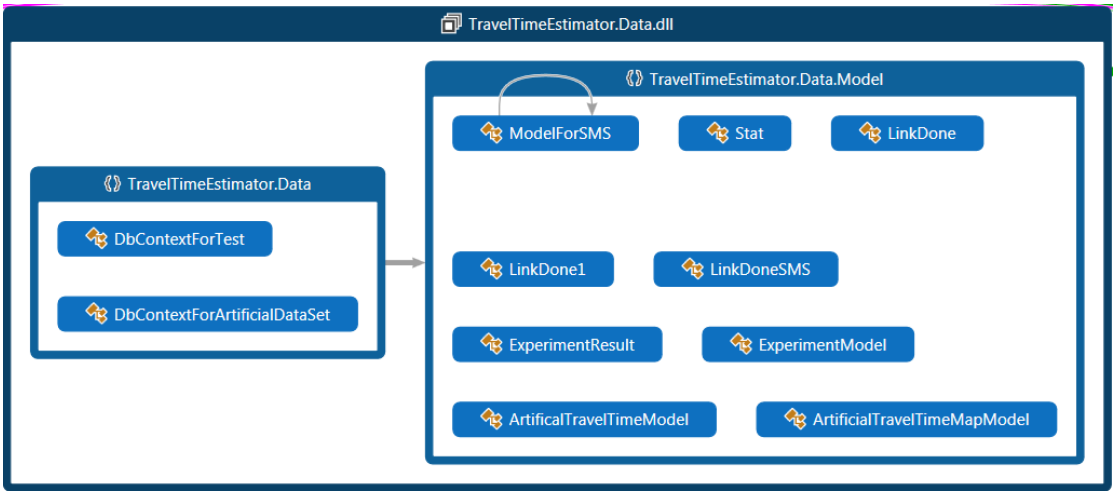


FIGURE B.5: TravelTimeEstimatorData code diagram

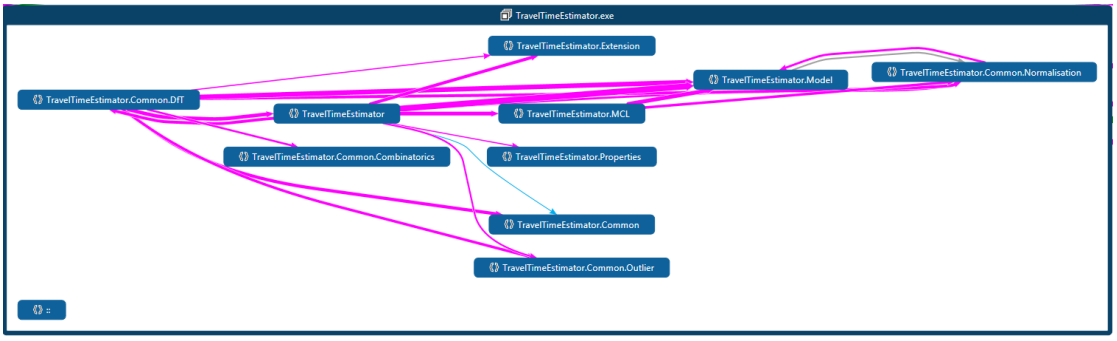


FIGURE B.6: TravelTimeEstimator code diagram

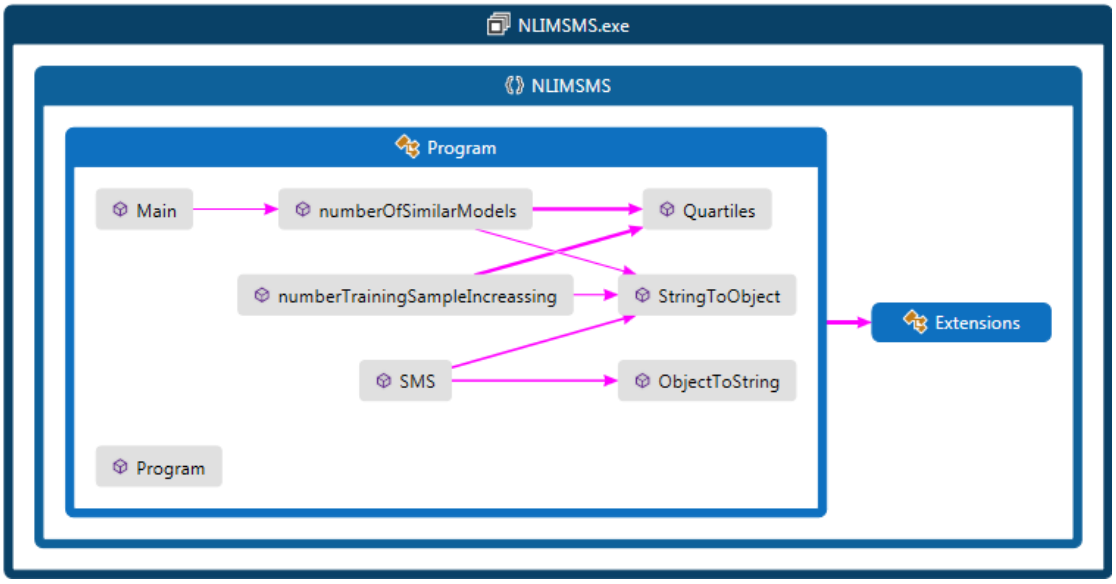


FIGURE B.7: NLIMSMS code diagram

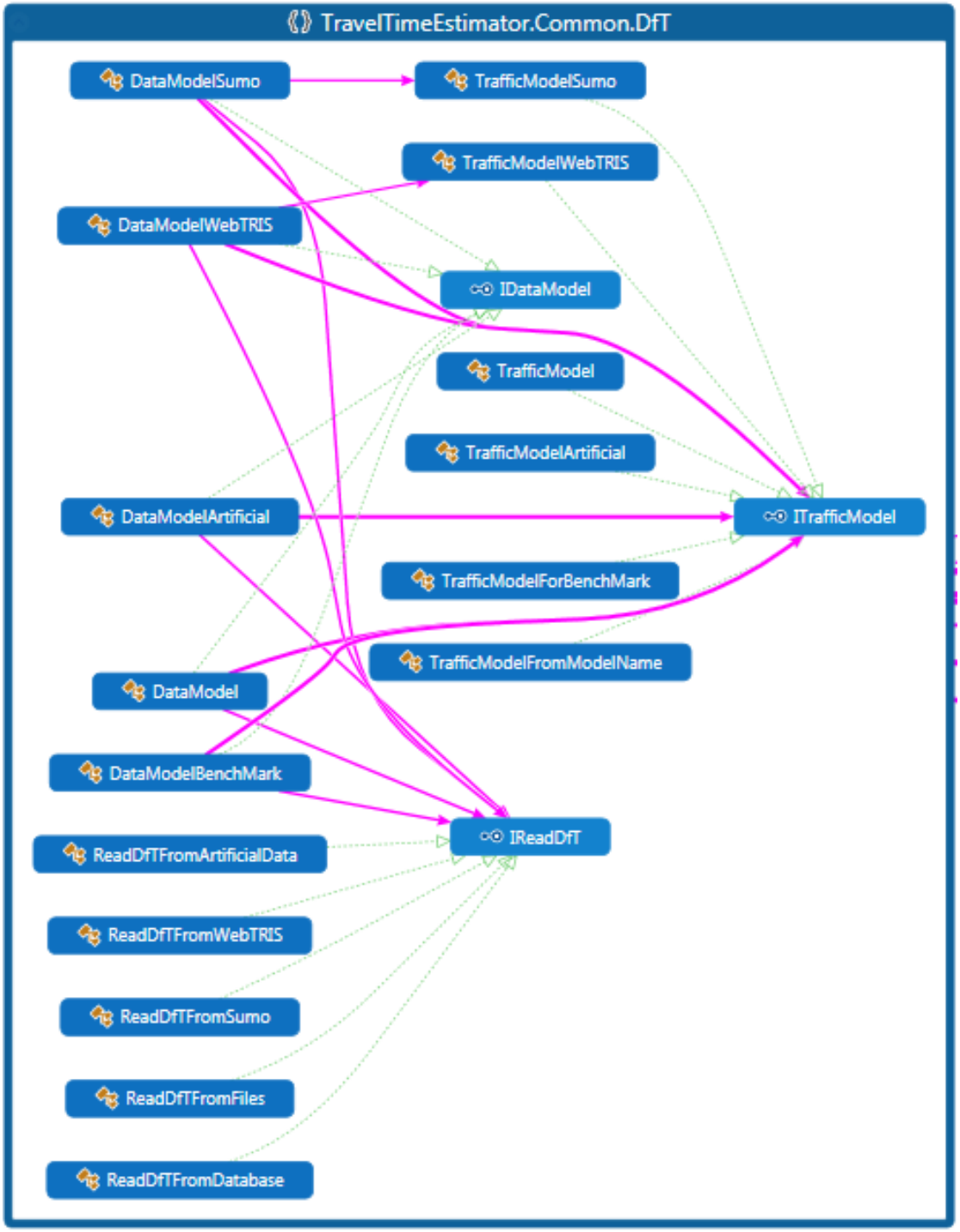


FIGURE B.8: TravelTimeEstimator.Common.DfT code diagram

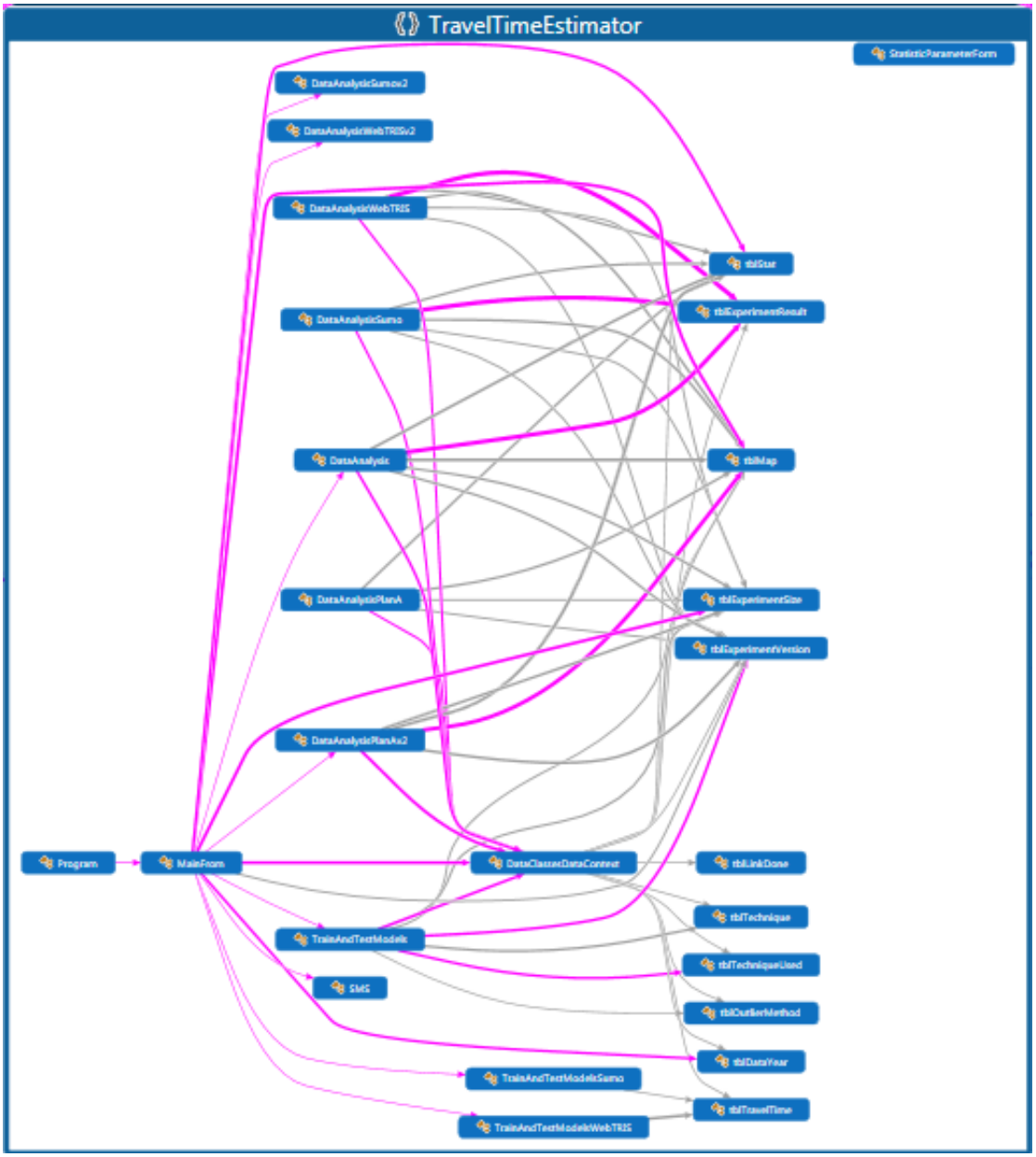


FIGURE B.9: TravelTimeEstimatorSub code diagram

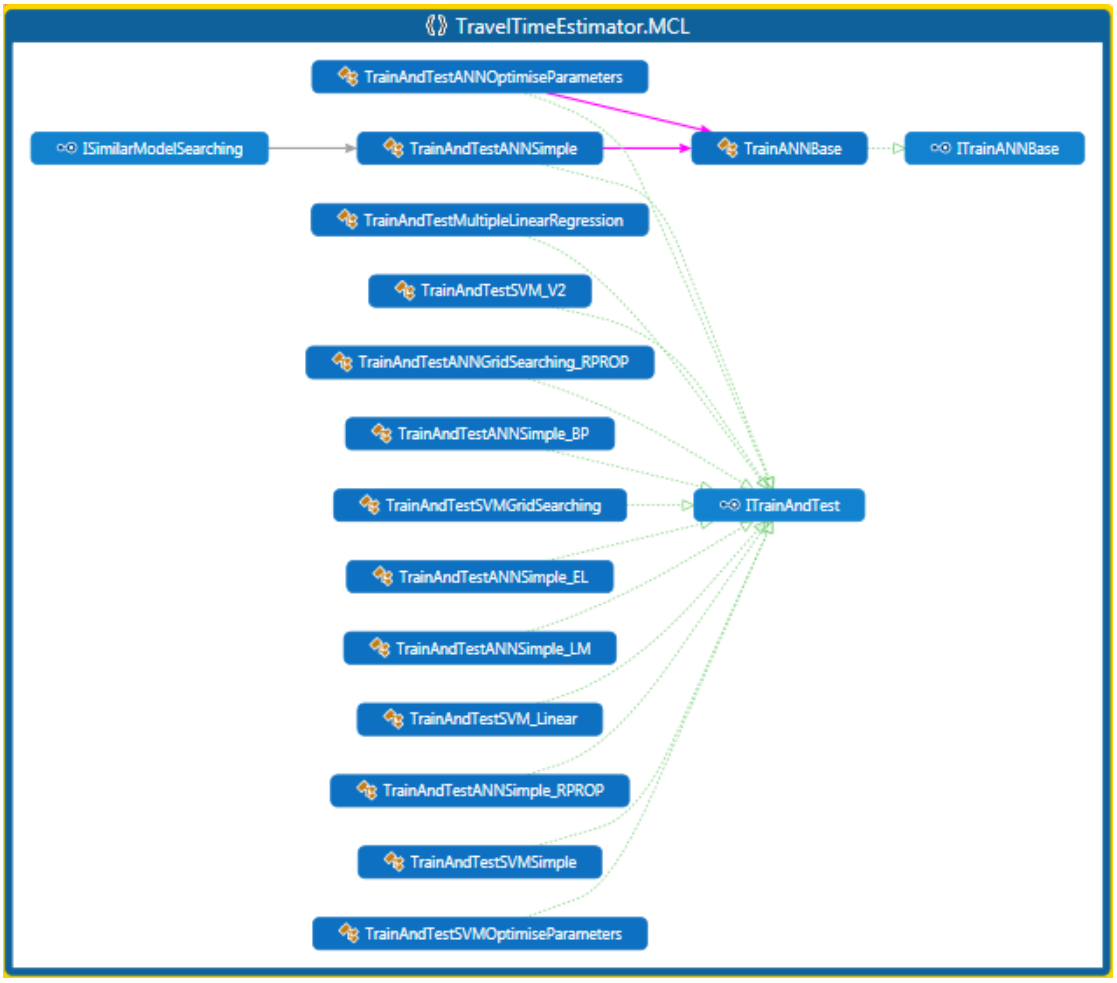


FIGURE B.10: TravelTimeEstimator.MCL code diagram

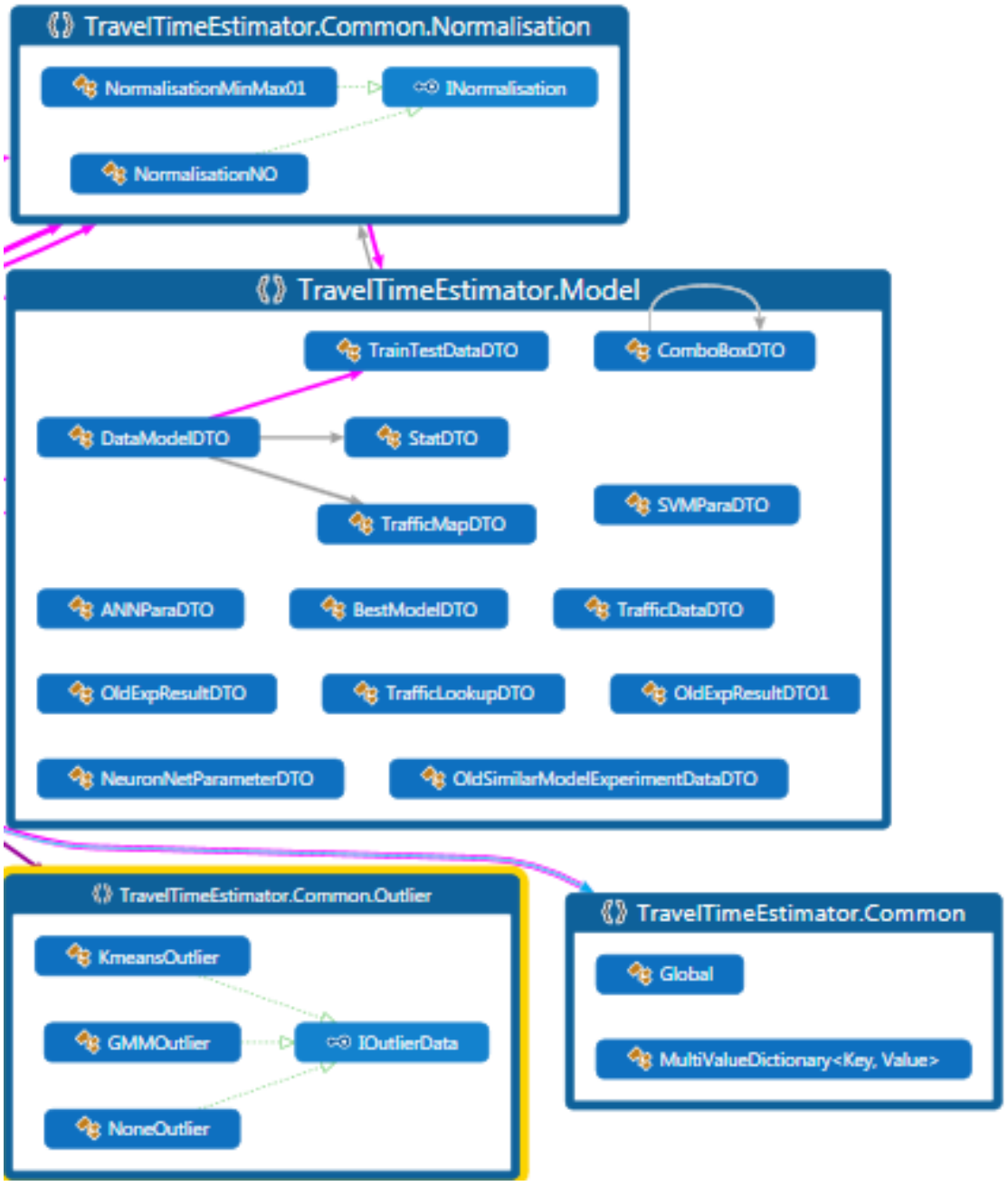


FIGURE B.11: TravelTimeEstimator: Common, Model and Common.Outlier code diagram

Bibliography

Abu-Lebdeh, G. and Singh, A. K. (2011), 'Modeling arterial travel time with limited traffic variables using conditional independence graphs & state-space neural networks', *Procedia - Social and Behavioral Sciences* **16**(0), 207 – 217. 6th International Symposium on Highway Capacity and Quality of Service.
URL: *http://www.sciencedirect.com/science/article/pii/S187704281100989X*

Ahn, G.-H., Ki, Y.-K. and Kim, E.-J. (2014), 'Real-time estimation of travel speed using urban traffic information system and filtering algorithm', *Intelligent Transport Systems, IET* **8**(2), 145–154.

Arlot, S. and Celisse, A. (2010), 'A survey of cross-validation procedures for model selection'.

Bedi, P., Jindal, V., Garg, R. and Dhankani, H. (2015), A preemptive approach to reduce average queue length in vanets, *in* '2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)', pp. 2089–2095.

Bedogni, L., Gramaglia, M., Vesco, A., Fiore, M., Härri, J. and Ferrero, F. (2015), 'The bologna ringway dataset: Improving road network conversion in sumo and validating urban mobility via navigation services', *IEEE Transactions on Vehicular Technology* **64**(12), 5464–5476.

Behrisch, M. and Weber, M. (2014), *Modeling Mobility with Open Data*.

Bergstra, J. and Bengio, Y. (2012), 'Random search for hyper-parameter optimization'.

Bilbao, I. and Bilbao, J. (2017), Overfitting problem and the over-training in the era of data: Particularly for artificial neural networks, *in* '2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)', pp. 173–177.

Bischof, H., Leonardis, A. and Selb, A. (1999), 'Mdl principle for robust vector quantisation', *Pattern Analysis & Applications* **2**(1), 59–72.
URL: *https://doi.org/10.1007/s100440050015*

Burges, C. J., Schölkopf, B. and Smola, A. J. (1999), *Advances in kernel methods. chapter Support Vector Machines for Dynamic Reconstruction of a Chaotic System*, The MIT Press.

Cai, G., Chen, B. M. and Lee, T. H. (2011), *Unmanned Rotorcraft Systems*.

Capes, D. and Hewitt, R. (2005), 'Integration improves traffic management in york, uk', *Proceedings of the Institution of Civil Engineers - Municipal Engineer* **158**(4), 275–280.
URL: *https://doi.org/10.1680/muen.2005.158.4.275*

Cheng, Y., Lee, R. K., Lim, E. and Zhu, F. (2013), Delayflow centrality for identifying critical nodes in transportation networks, *in* '2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)', pp. 1462–1463.

Cherkassky, V. and Mulier, F. M. (2007), *Learning from Data: Concepts, Theory, and Methods, 2nd Edition*, Wiley-IEEE Press.

Chitraranjan, C. D., Denton, A. M. and Perera, A. S. (2016), A complete observation model for tracking vehicles from mobile phone signal strengths and its potential in travel-time estimation, *in* '2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)', pp. 1–7.

Chitraranjan, C. D., Perera, A. S. and Denton, A. M. (2015), Tracking vehicle trajectories by local dynamic time warping of mobile phone signal strengths and its potential in travel-time estimation, *in* '2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)', pp. 445–450.

Claesen, M. and Moor, B. D. (2015), Hyperparameter search in machine learning, *in* 'MIC 2015: The XI Metaheuristics International Conference'.

Cookson, G. and Pishue, B. (2017), 'Inrix global traffic scorecard'.

Daniel Krajzewicz, Jakob Erdmann, M. B. and Bieker, L. (2012), 'Recent development and applications of sumo – simulation of urban mobility'.

Daniel Krajzewicz, Jakob Erdmann, M. B. and Bieker, L. (2018), 'Sumo user documentation'.
URL: [1]

[1] `http://sumo.dlr.de/wiki/SUMO_User_Documentation`

Díaz, J. J. V., González, A. B. R. and Wilby, M. R. (2016), 'Bluetooth traffic monitoring systems for travel time estimation on freeways', *IEEE Transactions on Intelligent Transportation Systems* **17**(1), 123–132.

de Dios Ortuzar, J. and G.Willumsen, L. (2011), *MODELLING TRANSPORT (Fourth Edition)*.

de Dios Ortúzar, J. and Willumsen, L. G. (2011), *MODELLING TRANSPORT 4th Edition*, John Wiley & Sons, Ltd.

de Fabritiis, C., Ragona, R. and Valenti, G. (2008), Traffic estimation and prediction based on real time floating car data, *in* 'Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on', pp. 197–203.

Deng, L., He, Z. and Zhong, R. (2013), The bus travel time prediction based on bayesian networks, *in* 'Information Technology and Applications (ITA), 2013 International Conference on', pp. 282–285.

Department of Transport, U. (2012), 'Guidance on road classification and the primary route network'.

Department of Transport, U. (2016), 'Congestion (average speed during the weekday morning peak) on local 'a' roads – methodology'.

Derbel, A. and Boujelbene, Y. (2015), Bayesian network for traffic management application: Estimated the travel time, *in* 'Web Applications and Networking (WSWAN), 2015 2nd World Symposium on', pp. 1–6.

Derrmann, T., Frank, R., Faye, S., Castignani, G. and Engel, T. (2016), Towards privacy-neutral travel time estimation from mobile phone signalling data, *in* '2016 IEEE International Smart Cities Conference (ISC2)', pp. 1–6.

Dong, J. and Mahmassani, H. (2012), 'Stochastic modeling of traffic flow breakdown phenomenon: Application to predicting travel time reliability', *Intelligent Transportation Systems, IEEE Transactions on* **13**(4), 1803–1809.

Ernst, J. M., Krogmeier, J. V. and Bullock, D. M. (2014), 'Estimating required probe vehicle re-identification requirements for characterizing link travel times', *IEEE Intelligent Transportation Systems Magazine* **6**(1), 50–58.

Fei, X., Lu, C.-C. and Liu, K. (2011), 'A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction', *Transportation Research Part C: Emerging Technologies* **19**(6), 1306 – 1318.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X11000325*

Fleischmann, B., Gnutzmann, S. and Sandvoβ, E. (2004), 'Dynamic vehicle routing based on online traffic information', *Transportation Science* **38**(4), 420–433.
URL: *http://dx.doi.org/10.1287/trsc.1030.0074*

Fushiki, T. (2011), 'Estimation of prediction error by using k-fold cross-validation', *Statistics and Computing* **21**(2), 137–146.
URL: *https://doi.org/10.1007/s11222-009-9153-8*

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*.

Gori, M. and Tesi, A. (1992), 'On the problem of local minima in backpropagation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(1), 76–86.

Green, S. B. (1991), 'How many subjects does it take to do a regression analysis', *Multivariate Behavioral Research* **26**(3), 499–510. PMID: 26776715.
URL: *https://doi.org/10.1207/s15327906mbr2603*

Guo, Q., Heng, C. K., Theng, Y. L., Ong, Y. S. and Tan, P. S. (2015), Offline time-sensitive travel time estimation in an urban road network, *in* '2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)', pp. 848–852.

Hadachi, A., Lecomte, C., Mousset, S. and Bensrhair, A. (2011), An application of the sequential monte carlo to increase the accuracy of travel time estimation in urban areas, *in* 'Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on', pp. 157–162.

Hadachi, A., Mousset, S. and Bensrhair, A. (2012), Practical testing application of travel time estimation using applied monte carlo method and adaptive estimation from probes, *in* 'Intelligent Vehicles Symposium (IV), 2012 IEEE', pp. 1078–1083.

Hadachi, A., Mousset, S. and Bensrhair, A. (2013), 'Approach to estimate travel time using sparsely sampled gps data in urban networks', *Electronics Letters* **49**(15), 957–958.

Hage, R. M., Bétaille, D., Peyret, F., Meizel, D. and Smal, J. C. (2012), Unscented kalman filter for urban link travel time estimation with mid-link sinks and sources, *in* '2012 15th International IEEE Conference on Intelligent Transportation Systems', pp. 1632–1637.

Hamerly, G. and Elkan, C. (2003), Learning the k in k-means, *in* 'Proceedings of the 16th International Conference on Neural Information Processing Systems', NIPS'03,

MIT Press, Cambridge, MA, USA, pp. 281–288.
URL: *http://dl.acm.org/citation.cfm?id=2981345.2981381*

Haykin, S. (2008), *Neural Networks and Learning Machines (Third Edition)*.

Highways-England (2016), 'Highways england network journey time and traffic flow data'.

Highways-England (2018), 'Webtris'.
URL: *http://webtris.highwaysengland.co.uk/*

Hinsbergen, C. P. I. J. V., Hegyi, A., Lint, J. W. C. V. and Zuylen, H. J. V. (2011), 'Bayesian neural networks for the prediction of stochastic travel times in urban networks', *IET Intelligent Transport Systems* **5**(4), 259–265.

Holland, J. H. (1992), *Adaptation in Natural and Artificial Systems*.

Hsu, C.-W., Chang, C.-C. and Lin, C.-J. (2016), A practical guide to support vector classification, Technical report, Department of Computer Science National Taiwan University, Taipei 106, Taiwan.

Huang, L. and Barth, M. (2008), A novel loglinear model for freeway travel time prediction, *in* 'Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on', pp. 210–215.

Huemer, A., Góngora, M. and Elizondo, D. (2010), A robust reinforcement based self constructing neural network, *in* 'The 2010 International Joint Conference on Neural Networks (IJCNN)', pp. 1–7.

Jang, J. (2016), 'Outlier filtering algorithm for travel time estimation using dedicated short-range communications probes on rural highways', *IET Intelligent Transport Systems* **10**(6), 453–460.

Jenelius, E. and Koutsopoulos, H. N. (2013), 'Travel time estimation for urban road networks using low frequency probe vehicle data', *Transportation Research Part B: Methodological* **53**, 64 – 81.
URL: *http://www.sciencedirect.com/science/article/pii/S0191261513000489*

Jobson, J. D. (1991), *Multiple Linear Regression*.

Jones, M., Geng, Y., Nikovski, D. and Hirata, T. (2013), Predicting link travel times from floating car data, *in* 'Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on', pp. 1756–1763.

Kan, C. D., Chen, W. L., Lin, C. H., Wang, J. N., Lu, P. J., Chan, M. Y. and Wu, J. T. (2018), 'Handmade trileaflet valve design and validation for pulmonary valved conduit reconstruction using taguchi method and cascade correlation machine learning model', *IEEE Access* **6**, 7088–7099.

Kassambara, A. (2017), *Practical Guide to Cluster Analysis in R*, STHDA.

Kim, G. (2017), Travel time estimation in vehicle routing problem, *in* '2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)', pp. 1004–1008.

Kim, J.-H. (2009), 'Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap', *Computational Statistics & Data Analysis* **53**(11), 3735 – 3745.
URL: *http://www.sciencedirect.com/science/article/pii/S0167947309001601*

Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L. (2012), 'Recent development and applications of SUMO - Simulation of Urban MObility', *International Journal On Advances in Systems and Measurements* **5**(3&4), 128–138.

Krishna Menon, A. (2018), 'Large-scale support vector machines: Algorithms and theory'.

Krishnamoorthy, R. K. (2008), Travel time estimation and forecasting on urban roads, PhD thesis.

Lawrence, S. and Giles, C. L. (2000), Overfitting and neural networks: conjugate gradient and backpropagation, *in* 'Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium', Vol. 1, pp. 114–119 vol.1.

Lee, K., Prokhorchuk, A., Dauwels, J. and Jaillet, P. (2017), Estimation of travel time from taxi gps data, *in* '2017 IEEE Symposium Series on Computational Intelligence (SSCI)', pp. 1–6.

Leodolter, M., Koller, H. and Straub, M. (2015), Estimating travel times from static map attributes, *in* 'Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on', pp. 121–126.

Li, C.-S. and Chen, M.-C. (2013), 'Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks', *Neural Computing and Applications* **23**(6), 1611–1629.
URL: *http://dx.doi.org/10.1007/s00521-012-1114-z*

Li, L., Chen, X., Li, Z. and Zhang, L. (2013), 'Freeway travel-time estimation based on temporal-spatial queueing model', *Intelligent Transportation Systems, IEEE Transactions on* **14**(3), 1536–1541.

Lin, G., Xin, L., Feng, H. and Ying, L. (2014), A new outlier detection algorithm and its application in intelligent transportation system, *in* 'Information Technology and Artificial Intelligence Conference (ITAIC), 2014 IEEE 7th Joint International', pp. 442–445.

Liu, Y., Starzyk, J. A. and Zhu, Z. (2008), 'Optimized approximation algorithm in neural networks without overfitting', *IEEE Transactions on Neural Networks* **19**(6), 983–995.

Londoño, G. and Lozano, A. (2012), Suitable cost functions for signalized arterials and freeways, in the user equilibrium assignment problem.

Lu, L., Wang, J., He, Z. and Chan, C. Y. (2018), 'Real-time estimation of freeway travel time with recurrent congestion based on sparse detector data', *IET Intelligent Transport Systems* **12**(1), 2–11.

Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F. (2015), 'Traffic flow prediction with big data: A deep learning approach', *IEEE Transactions on Intelligent Transportation Systems* **16**(2), 865–873.

Ma, X. and Koutsopoulos, H. N. (2008), A new online travel time estimation approach using distorted automatic vehicle identification data, *in* '2008 11th International IEEE Conference on Intelligent Transportation Systems', pp. 204–209.

Ma, X., Yu, H., Wang, Y. and Wang, Y. (2015), 'Large-scale transportation network congestion evolution prediction using deep learning theory', **10**, e0119044.

Maiti, S., Pal, A., Pal, A., Chattopadhyay, T. and Mukherjee, A. (2014), Historical data based real time prediction of vehicle arrival time, *in* 'Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on', pp. 1837–1842.

Matas, A., Raymond, J.-L. and Ruiz, A. (2012), 'Traffic forecasts under uncertainty and capacity constraints', *Transportation* **39**(1), 1–17.
URL: *https://doi.org/10.1007/s11116-011-9325-1*

Mathew, T. V. (2018), 'Transportation network design'.
URL: [2]

McCulloch, W. S. and Pitts, W. (1943), 'A logical caluclaus of the ideas immanent in nervous activity'.

[2] `https://www.civil.iitb.ac.in/tvm/1100_LnTse/206_lnTse/plain/plain.html`

Meiying, J., Hainana, L. and Led, N. (2015), The evaluation studies of regional transportation accessibility based on intelligent transportation system: Take the example in yunnan province of china, *in* '2015 International Conference on Intelligent Transportation, Big Data and Smart City', pp. 862–865.

Meng, Z., Wang, C., Peng, L., Teng, A. and Qiu, T. Z. (2017), Link travel time and delay estimation using transit avl data, *in* '2017 4th International Conference on Transportation Information and Safety (ICTIS)', pp. 67–72.

Molinaro, A. M., Simon, R. and Pfeiffer, R. M. (2005), 'Prediction error estimation: a comparison of resampling methods'.

Narayanan, A., Mitrovic, N., Asif, M. T., Dauwels, J. and Jaillet, P. (2015), Travel time estimation using speed predictions, *in* '2015 IEEE 18th International Conference on Intelligent Transportation Systems', pp. 2256–2261.

Nguyen, P. T. M., Passow, B. N. and Yang, Y. (2016), Improving anytime behavior for traffic signal control optimization based on nsga-ii and local search, *in* '2016 International Joint Conference on Neural Networks (IJCNN)', pp. 4611–4618.

Nielsen, O. A. and Jorgensen, R. M. (2008), 'Estimation of speed-flow and flow-density relations on the motorway network in the greater copenhagen region', *IET Intelligent Transport Systems* **2**(2), 120–131.

OpenStreetMap contributors (2017), 'Planet dump retrieved from https://planet.osm.org ', `https://www.openstreetmap.org`.

Oracle (2018), 'Oracle9i olap services developer's guide to the olap dml release 1 (9.0.1)'. **URL:** [3]

Ortega-Zamorano, F., Jerez, J. M., Muñoz, D. U., Luque-Baena, R. M. and Franco, L. (2016), 'Efficient implementation of the backpropagation algorithm in fpgas and microcontrollers', *IEEE Transactions on Neural Networks and Learning Systems* **27**(9), 1840–1850.

Pan, S., Jiang, B., Zou, N. and Jia, L. (2011), Average travel speed estimation using multi-type floating car data, *in* 'Information and Automation (ICIA), 2011 IEEE International Conference on', pp. 192–197.

Passow, B., Elizondo, D., Chiclana, F., Witheridge, S. and Goodyer, E. (2013), Adapting traffic simulation for traffic management: A neural network approach, *in* 'Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on', pp. 1402–1407.

[3] `https://docs.oracle.com/cd/A91202_01/901_doc/olap.901/a86720/esdatao6.htm`

Pelleg, D. and Moore, A. (2000), X-means: Extending k-means with efficient estimation of the number of clusters, *in* 'International Conference on Machine Learning, 2000 (ICML2000)'.

Petrik, O., Moura, F. and de Abreu e Silva, J. (2016), 'Measuring uncertainty in discrete choice travel demand forecasting models', *Transportation Planning and Technology* **39**(2), 218–237.
URL: *https://doi.org/10.1080/03081060.2015.1127542*

Petrovska, N. and Stevanovic, A. (2015), Traffic congestion analysis visualisation tool, *in* 'Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on', pp. 1489–1494.

Pirc, J., Turk, G. and Žura, M. (2016), 'Highway travel time estimation using multiple data sources', *IET Intelligent Transport Systems* **10**(10), 649–657.

Pirc, J., Turk, G. and Zura, M. (2015), 'Using the robust statistics for travel time estimation on highways', *Intelligent Transport Systems, IET* **9**(4), 442–452.

Prasad, N., Singh, R. and Lal, S. P. (2013), Comparison of back propagation and resilient propagation algorithm for spam classification, *in* '2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation', pp. 29–34.

Protschky, V., Feit, S. and Linnhoff-Popien, C. (2015), On the potential of floating car data for traffic light signal reconstruction, *in* '2015 IEEE 81st Vehicular Technology Conference (VTC Spring)', pp. 1–5.

Protschky, V., Ruhhammer, C. and Feit, S. (2015), Learning traffic light parameters with floating car data, *in* '2015 IEEE 18th International Conference on Intelligent Transportation Systems', pp. 2438–2443.

Rahimi, A. and Recht, B. (2008), Random features for large-scale kernel machines, *in* J. C. Platt, D. Koller, Y. Singer and S. T. Roweis, eds, 'Advances in Neural Information Processing Systems 20', Curran Associates, Inc., pp. 1177–1184.
URL: *http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf*

Rahmani, M., Jenelius, E. and Koutsopoulos, H. (2013), Route travel time estimation using low-frequency floating car data, *in* 'Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on', pp. 2292–2297.

Rahmani, M., Jenelius, E. and Koutsopoulos, H. (2014), Floating car and camera data fusion for non-parametric route travel time estimation, *in* 'Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on', pp. 1286–1291.

Refaeilzadeh, P., Tang, L. and Liu, H. (2016), *Cross-Validation*, Springer New York, New York, NY, pp. 1–7.
URL: [4]

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

Rodrigue, J.-P., Comtois, C. and Slack, B. (2013), *The Geography of Transport Systems (3rd Edition)*, Routledge.

Rodzin, S., Rodzina, O. and Rodzina, L. (2016), Neuroevolution: Problems, algorithms, and experiments, *in* '2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)', pp. 1–4.

Rosenblatt, F. (1958), 'The perceptron: a probabilistic model for information storage and organisation in the brain'.

Scholkopf, B. and Smola, A. J. (2001), *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press Cambridge, MA, USA.

Shawe-Taylor, J. and Cristianini, N. (2004), *Kernel Methods for Pattern Analysis*, cambridge university press.

Shawn M. Turner, William L. Eisele, R. J. B. and Holdener, D. J. (1998), *Travel Time Data Collection Handbook*, Texas Transportation Institute.

Steeb, W.-H. (2008), *The Nonlinear Workbook (Fourth edition)*.

Su, H., Hu, Y. and Xu, J. (2010), Travel time estimating algorithms based on fuzzy radial basis function neural networks, *in* 'Control and Decision Conference (CCDC), 2010 Chinese', pp. 2653–2657.

Suzuki, K. (2011), *ARTIFICIAL NEURAL NETWORKS: INDUSTRIAL AND CONTROL ENGINEERING APPLICATIONS*.

Tabachnick, B. G. and Fidel, L. S. (2007), *Using Multivariate Statistics*.

Tang, K., Chen, S. and Liu, Z. (2018), 'Citywide spatial-temporal travel time estimation using big and sparse trajectories', *IEEE Transactions on Intelligent Transportation Systems* pp. 1–12.

teletracnavman (2018), 'teletracnavman'.
URL: *https://www.teletracnavman.co.uk/company/press*

[4] `https://doi.org/10.1007/978-1-4899-7993-3_565-2`

Tettamanzi, A. and Tomassini, M. (2011), *Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems*.

Tomaras, D., Boutsis, I. and Kalogeraki, V. (2015), Travel time estimation in real-time using buses as speed probes, *in* '2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)', pp. 63–68.

Tu, H., van Lint, H. and Zuylen, H. V. (2006), Travel time variability versus freeway characteristics, *in* '2006 IEEE Intelligent Transportation Systems Conference', pp. 383–388.

van Hinsbergen, C., Hegyi, A., van Lint, J. and van Zuylen, H. (2011), 'Bayesian neural networks for the prediction of stochastic travel times in urban networks', *Intelligent Transport Systems, IET* **5**(4), 259–265.

Vapnik, V. N. (2000), *The nature of statistical learning theory*, Information Science and Statistics.

Vidović, K., Mandžuka, S. and Brčić, D. (2017), Estimation of urban mobility using public mobile network, *in* '2017 International Symposium ELMAR', pp. 21–24.

Vlahogianni, E. I., Karlaftis, M. G. and Golias, J. C. (2014), 'Short-term traffic forecasting: Where we are and where we're going', *Transportation Research Part C: Emerging Technologies* **43, Part 1**(0), 3 – 19. Special Issue on Short-term Traffic Flow Forecasting.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X14000096*

Vu, L. H., Passow, B. N. and Goodyer, E. (2016), Urban road traffic link travel time estimation based on sparse data, *in* 'Computer Systems Engineering 2016'.

Vu, L. H., Passow, B. N., Paluszczyszyn, D., Deka, L. and Goodyer, E. (2017), Neighbouring link travel time inference method using artificial neural network, *in* '2017 IEEE Symposium Series on Computational Intelligence (SSCI)', pp. 1–8.

Wan, N. and Vahidi, A. (2014), Probabilistic estimation of travel times in arterial streets using sparse transit bus data, *in* '17th International IEEE Conference on Intelligent Transportation Systems (ITSC)', pp. 1292–1297.

Wang, J., Indra-Payoong, N., Sumalee, A. and Panwai, S. (2014), 'Vehicle reidentification with self-adaptive time windows for real-time travel time estimation', *IEEE Transactions on Intelligent Transportation Systems* **15**(2), 540–552.

Wang, J., Wong, K. and Chen, Y. (2012), Short-term travel time estimation and prediction for long freeway corridor using nn and regression, *in* 'Intelligent

Transportation Systems (ITSC), 2012 15th International IEEE Conference on', pp. 582–587.

Waury, R., Hu, J., Yang, B. and Jensen, C. S. (2017), Assessing the accuracy benefits of on-the-fly trajectory selection in fine-grained travel-time estimation, *in* '2017 18th IEEE International Conference on Mobile Data Management (MDM)', pp. 240–245.

Waury, R., Jensen, C. S. and Torp, K. (2018), Adaptive travel-time estimation: A case for custom predicate selection, *in* '2018 19th IEEE International Conference on Mobile Data Management (MDM)', pp. 96–105.

Wei, W., Xiucheng, G., Jing, J. and Bin, R. (2010), Gps probe based freeway real-time travel speed estimation using kalman filter, *in* 'Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on', Vol. 1, pp. 797–800.

Williams, C. K. I. and Seeger, M. (2001), Using the nyström method to speed up kernel machines, *in* T. K. Leen, T. G. Dietterich and V. Tresp, eds, 'Advances in Neural Information Processing Systems 13', MIT Press, pp. 682–688.
URL: *http://papers.nips.cc/paper/1866-using-the-nystrom-method-to-speed-up-kernel-machines.pdf*

Wosyka, J. and Pribyl, P. (2012), Real-time travel time estimation on highways using loop detector data and license plate recognition, *in* 'ELEKTRO, 2012', pp. 391–394.

Wright, C. (1973), 'A theoretical analysis of the moving observer method'.

Wu, B., Li, K., Yang, M. and Lee, C. (2017), 'A reverberation-time-aware approach to speech dereverberation based on deep neural networks', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **25**(1), 102–111.

Yang, Q., Wu, G., Boriboonsomsin, K. and Barth, M. (2013), Arterial roadway travel time distribution estimation and vehicle movement classification using a modified gaussian mixture model, *in* '16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)', pp. 681–685.

Yeon, J. and Ko, B. (2007), Comparison of travel time estimation using analysis and queuing theory to field data along freeways, *in* 'Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on', pp. 530–538.

Yi, S., Li, H. and Wang, X. (2015), Pedestrian travel time estimation in crowded scenes, *in* '2015 IEEE International Conference on Computer Vision (ICCV)', pp. 3137–3145.

Yildirimoglu, M. and Geroliminis, N. (2013), 'Experienced travel time prediction for congested freeways', *Transportation Research Part B: Methodological* **53**, 45 – 63.
URL: *http://www.sciencedirect.com/science/article/pii/S0191261513000465*

Youn, E. and Jeong, M. K. (2009), 'Class dependent feature scaling method using naive bayes classifier for text datamining', *Pattern Recognition Letters* **30**(5), 477 – 485.
URL: *http://www.sciencedirect.com/science/article/pii/S0167865508003553*

Zhan, X., Hasan, S., Ukkusuri, S. V. and Kamga, C. (2013), 'Urban link travel time estimation using large-scale taxi data with partial information', *Transportation Research Part C: Emerging Technologies* **33**, 37 – 49.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X13000740*

Zhang, L. and Mao, X. (2015), 'Vehicle density estimation of freeway traffic with unknown boundary demand 2013-supply: an interacting multiple model approach', *Control Theory Applications, IET* **9**(13), 1989–1995.

Zhang, Y., Chen, S. and Wan, Y. (2009), An intelligent algorithm based on grid searching and cross validation and its application in population analysis, *in* '2009 International Conference on Computational Intelligence and Natural Computing', Vol. 2, pp. 96–99.

Zhao, X. and Spall, J. C. (2016), Estimating travel time in urban traffic by modeling transportation network systems with binary subsystems, *in* '2016 American Control Conference (ACC)', pp. 803–808.

Zhao, Y. and Kockelman, K. M. (2011), The propagation of uncertainty through travel demand models: An exploratory analysis.

Zheng, P. and McDonald, M. (2009), 'Estimation of travel time using fuzzy clustering method', *Intelligent Transport Systems, IET* **3**(1), 77–86.

Zou, Y., Zhu, X., Zhang, Y. and Zeng, X. (2014), 'A space–time diurnal method for short-term freeway travel time prediction', *Transportation Research Part C: Emerging Technologies* **43, Part 1**, 33 – 49. Special Issue on Short-term Traffic Flow Forecasting.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X13002222*