5-2-2009

Travel time estimation in congested urban networks using point detectors data

Anas Mohammad Mahmoud

TRAVEL TIME ESTIMATION IN CONGESTED URBAN NETWORKS USING POINT DETECTORS DATA

By

Anas Mohammad Mahmoud

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master's of Science
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

May 2009

TRAVEL TIME ESTIMATION IN CONGESTED URBAN NETWORKS USING

POINT DETECTORS DATA

By

Anas Mohammad Mahmoud

Approved:

Eric A. Hansen
Associate Professor of Computer Science and Engineering
(Major Advisor)

Li Zhang
Associate Professor of Civil and Environmental Engineering
(Director of Thesis)

Julian E. Boggess
Associate Professor of Computer Science and Engineering
(Committee Member)

Edward B. Allen
Associate Professor of Computer Science and Engineering
(Graduate Coordinator)

Sarah A. Rajala
Dean of James Worth Bagley College of Engineering

Name: Anas Mohammad Mahmoud

Date of Degree: May 2, 2009

Institution: Mississippi State University

Major Field: Computer Science

Major Professor: Eric Hansen

Title of Study: TRAVEL TIME ESTIMATION IN CONGESTED URBAN NETWORKS USING POINT DETECTORS DATA.

Pages in Study: 111

Candidate for Degree of Master's of Science

A model for estimating travel time on short arterial links of congested urban networks, using currently available technology, is introduced in this thesis. The objective is to estimate travel time, with an acceptable level of accuracy for real-life traffic problems, such as congestion management and emergency evacuation. To achieve this research objective, various travel time estimation methods, including highway trajectories, multiple linear regression (MLR), artificial neural networks (ANN) and K – nearest neighbor (K-NN) were applied and tested on the same dataset. The results demonstrate that ANN and K-NN methods outperform linear methods by a significant margin, also, show particularly good performance in detecting congested intervals. To ensure the quality of the analysis results, set of procedures and algorithms based on traffic flow theory and test field information, were introduced to validate and clean the data used to build, train and test the different models.

DEDICATION

I would like to dedicate this research to my family; my dad, mum, and brothers and to all of my sincere friends and colleagues.

ACKNOWLEDGMENTS

The author wishes to express his warm thanks to Dr. Li Zhang from the Civil Engineering Department for his dedicated and faithful help from the beginning to the end. Special thanks to Dr. Eric Hansen from the Computer Science and Engineering Department for his continuous mentoring and valuable advices while pursuing this work. Sincere regards should be extended also to the committee member Dr. Julian Boggess for his cooperation and ultimate help during the organization phase of this work, and Dr. Edward Allen the graduate coordinator of the Computer Science and Engineering department for his continued support.

Finally, warm thanks to my family and my friends, I believe that without their spiritual support, this work would not have come into being.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

During the 20th century, solutions for traffic problems were limited to improving physical infrastructure by building more roads, expanding the existing ones and adding more bridges, tunnels and traffic signals. With a rapid increase in the urban population and motorization, expanding the infrastructure has become more impossible due to physical space constraints and environmental concerns. Realizing that, government agencies and transportation professionals around the world have shifted their focus to a new approach that uses traffic information to manage and guide traffic. A set of new information-based systems have been designed to support the new approach; these systems are known as Intelligent Transportation systems (ITS). ITS can be defined as transportation systems that apply well-established technologies in communications, control, electronics, and computer hardware and software to improve transportation system performance [1]. Advanced traveler Information Systems (ATIS), Advanced Public Transportation System (APTS) and Advanced Traffic Management Systems (ATMS) are examples of ITS applications. Among the different ITS applications, ATIS is aimed to provide the public with accurate traffic information that allows them to make their own informed decisions, and so, help alleviate the traffic problems.

Such information includes: work zone, road conditions, closures, incidents, weather conditions, and traffic parameters such as travel time, speed, flow and occupancy. Among the different traffic information that ITS applications provide, travel time is the most important and yet the most complex parameter to collect and disseminate.

On the road, travel time is the result of the complex dynamics of traffic flow. This complexity comes from the inherently non-linear behavior of humans (drivers, operators and pedestrians), the non-predictable environmental conditions, such as weather changes or natural disasters and the manmade factors, such as work zones and traffic accidents. This non-linearity makes travel time a property of each individual trip for each individual traveler [2].

A wide range of research is available on estimating travel time on freeways, but limited effort has been done on arterial links of urban networks. This is due to the fact that users are concerned more about planning for long trip rather than relatively short ones. But In certain environments, such as grid-like road networks and certain situations, such as emergency evacuation, the availability of such information is critical. In grid-like road networks, which consist of large number of short links, the overall travel time of the network is the result of the travel times of its short links, so it is important to study those links to be able to understand the network. In situations such as emergency evacuation, where each second counts, it is important to provide the network users with such information to guide them through the evacuation area. These are the motivations for this research on travel time on arterial links of congested urban networks.

1.2 Objective and Research Goals

The objective of this thesis is to introduce a model that uses the currently available technology to estimate travel time on short arterial links of congested urban networks. The goal is to estimate travel time, with acceptable level of accuracy that can be applied to real-life traffic problems, such as, congestion management and emergency evacuation.

To achieve the research goals, the thesis addresses the following key steps:

1- Problem settings: Identifying the main characteristics of the problem, its scope, input and output.

2- Data Collection: Identifying the data required to conduct the research, the test geographic location, time and data collection methods and tools.

3- Data Quality: Assuring the quality of the collected data by verifying set of data validity assumptions, based on the traffic flow theory and field information.

4- Data Cleaning and Pre-processing: Developing set of procedures and algorithms to clean and pre-process the data.

5- Estimating travel time: Evaluating different travel time estimation methods that are usually employed to estimate travel time on highways, by comparing their performance on arterial links of congested urban networks.

Figure 1 shows the estimation model in real life.

1.3 Organization of the Thesis

The rest of the Thesis is organized as follows. Chapter 2 provides background information and a literature review for travel time Measurement, Estimation and Prediction methods. Chapter 3 describes the data collection, verification, cleaning and preprocessing. Chapter 4 describes the data analysis of this study; it includes analyzing and comparing different travel time estimation methods. Finally, Chapter 5 concludes the thesis and describes potential future work.

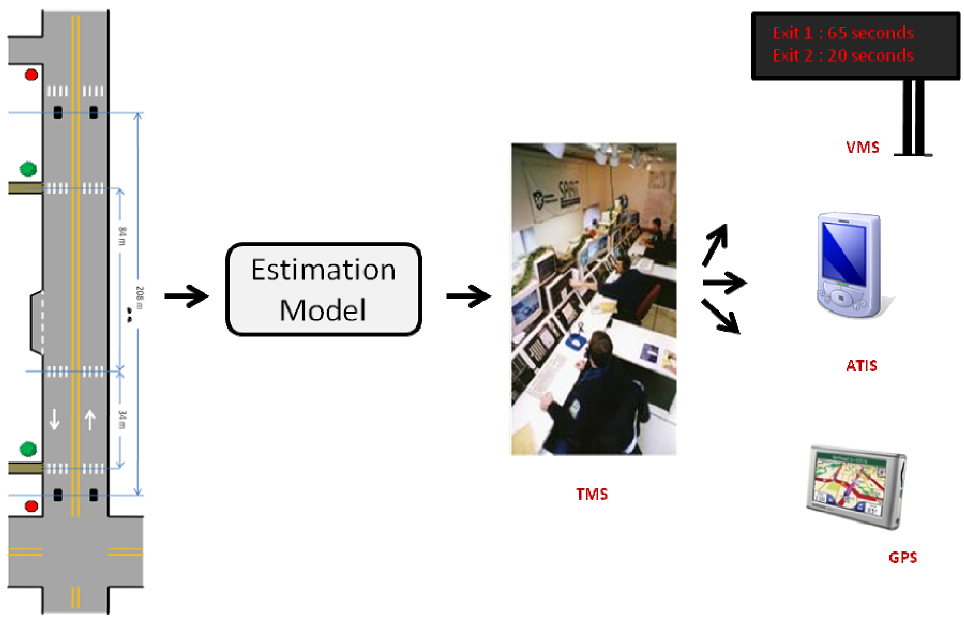

Figure 1 Travel Time Estimation Model in Real-life

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

2.1 What is Travel Time?

The definition of travel time varies depending on the context in which the measure is used. Usually, in a controlled laboratory environment, with ideal experimental conditions, travel time refers to the amount of time required for an object to move from one point to another, assuming that the object moves with no external factors that affect its movement in anyway. In the real world, the effect of the environment is inevitable, so, a more comprehensive definition is needed to reflect the effect of the external conditions on travel time.

In the literature, real-life travel time has been defined in different ways, depending on the application and the vocabulary of the traffic researcher. Smith et al [3], defines travel time as "the time it takes travelers to traverse a particular corridor". Schrader et al, [4] defines travel time as "the amount of time required to travel from one point to another on a given route". Van Lint [2] defines the individual travel time on a route *r* at departure time *t* as "the time it takes an individual traveler to traverse that particular route". According to the Traffic flow theory, travel time is the total time required for a vehicle to travel from one substantial point to another over a specified route under prevailing conditions such as work zone, weather and road conditions.

Even though the different definitions vary in their wordings, they all agree on four key elements: time, traveler, movement and route. The difference comes usually when travel time definition is expanded to describe the statistical nature (individual, average), the scope of the measure (Highway or Arterial) or any other Travel time property the researcher studies (reliability, consistency, etc).

From a statistical perspective, there are two main ways to describe travel time over a specified route: the individual and the average travel time. In the real-world, average travel time is a more practical measure, because it encompasses the individual drivers' behaviors and gives a more real description of the ground truth situation.

Multiple definitions can be found in the literature for average travel time. Bhaskar A. et al, [5] defines average travel time per vehicle as "The ratio between total Travel time and number of vehicles arrived" [5]. Singh & Abu-Lebdeh, [6] refers to average travel time as "The average value of the travel time incurred by individual vehicles if each vehicle can be tracked on the arterial link". Ruimin L. et al, [7] defines the mean travel time in a time period as "The average journey time of vehicles which start the specific journey during that time window" and Van Lint [2], describes the mean travel time on a route r for vehicles departing in period p as "The average time it takes these vehicles to traverse the specific route under the prevailing conditions on r during p".

Mathematically, if we assume that N vehicles move from point A to point B during time period p with a travel time TTi for each individual vehicle i, then the average Travel time during that time period is

$$TTp = \frac{1}{N}\sum_{i=1}^{N} \mathrm{TT_i} \tag{2.1}$$

In real-world road networks, where there are different traffic conditions, travel time value is composed of two main components: running time (TT_r), or the time in which the mode of transport is in motion, and delay time (TT_d), which is the time when the traffic is stopped due to traffic signals, congestions, traffic incidents or any other condition [8]. Equation 2 shows the different components of travel time.

$$TT = TT_r + TT_d \tag{2.2}$$

Based on Eq. 2.2, Eq. 2.1 which describes the average Travel time for N vehicles during period p, can be expanded to:

$$TTp = \frac{1}{N}\sum_{i=1}^{N}(TT_{ri} + TT_{di}) \tag{2.3}$$

Generally speaking, in highways and uninterrupted flow facilities (e.g., freeways or expressways) with stable traffic flow patterns, TT_r is the dominant term in Eq 2.2, but in arterial links, where the traffic has an interrupted nature, both TT_r and TT_d have proportional effect depending on the traffic flow. Figure 2 shows both the running and the delay time components of Travel time on an arterial link.

Figure 2 describes the movement of a vehicle on an arterial link. The movement is represented by the speed of the vehicle. The term TT_d is the time where the speed is zero, i.e., the vehicle is stopping, and TT_r is the time where the speed is larger than zero, i.e. the vehicle is moving.

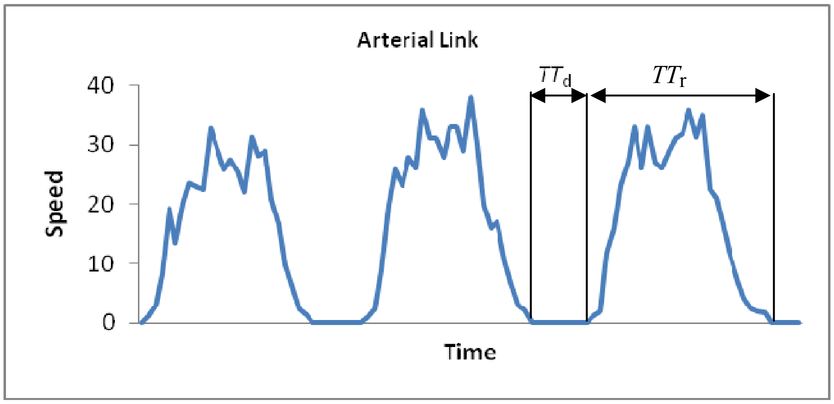

Figure 2 Travel Time Characteristics on The Road

2.2 Travel time Measurement, Estimation and Prediction

At this early stage of the thesis, it is important to distinguish between the different concepts of travel time measurement, estimation and prediction. In general, the techniques that are used to collect travel time data can be classified into two main categories: direct & indirect methods. In direct methods, Travel time data is collected directly from the field, under real traffic conditions, using tools and methods that were designed especially for that purpose. Such a process of collecting data directly is called Measurement. On the other hand, in indirect methods, travel time values are constructed indirectly from other traffic parameters which physically affect its value, such as flow, occupancy and speed.

Estimation and prediction can both be described as indirect methods. The difference between them is that the estimation process operates in a known state of traffic conditions and is used to generate experienced travel time, which refers to the time the vehicle has already taken to traverse the link of interest (the realized time).

On the other hand, the prediction process operates in an unknown state of traffic conditions and used for calculating travel time for future departure times.

Estimation can be divided further into online and offline methods depending on the “age” of the data used in the estimation process. Offline estimation refers to estimating travel time from historical traffic data that has already been collected and archived, while online estimation is the process of estimating travel time from the current real-time or near real-time traffic data. The relationship between travel time measurement, estimation and prediction is shown the in Figure 3.

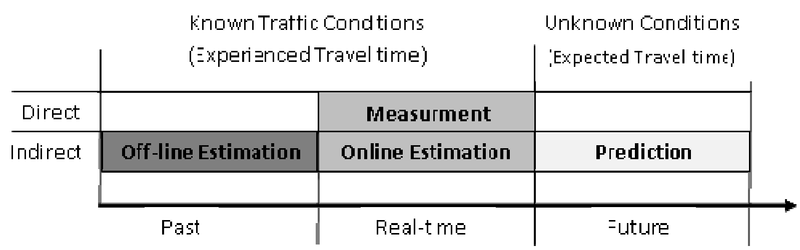

Figure 3 Travel Time Measurements, Estimation and Prediction Diagram

Travel time measurement, online estimation and prediction methods are the backbone of the ATIS systems. These systems aim to provide the users with pre-trip or en route travel information so that users can make smart route decisions in order to maximize their travel efficiency and safety.

Off-line travel time estimation is usually used in analytical studies to evaluate transportation system performance and to help in designing and planning future systems. It is also considered an essential step in building Travel time prediction method.

Multiple methods with different accuracies and reliabilities have been developed to measure, estimate and predict Travel time. This chapter provides a full description of the different Travel time data collection methods and the cons & pros of each method.

2.2.1 Direct Methods

Direct methods, can be divided into different categories depending on many factors such as operation, instrumentation and level of complexity. A very well-known classification is to divide them into road-based and vehicle-based techniques. In road-based techniques, data collection tools are deployed at different locations on the roadway, such as loop detectors, or on the side of the road, such as video cameras and radars. Vehicle-based methods are methods that collect data using tools installed-in or mounted-on a moving vehicle such as a global positioning system (GPS) or distance measuring instrument (DMI).

2.2.1.1 Road-based Techniques

Vehicle Re-identification is the main data collection methodology for road-based techniques. The main idea behind vehicle re-identification methods is to calculate Travel time between two points as the difference between the arrival times of a certain vehicle at each point. The main idea is to identify a vehicle at one point, record its arrival time at that point, re-identify the vehicle at another point, record its arrival time at that second point, and then, calculate the Travel time of that vehicle as the difference between the two timestamps. Sometimes the arrival time at the first point is called the departure time.

Mathematically, in vehicle re-identification methods, the definition of the Travel time of a vehicle that departed from point A at time (t_A) and arrived point B at time (t_b) (Figure 4) can be described using Eq. 2.4. The average travel time of N vehicles that departed from point A and arrived point B during time period p can be described using Eq. 5.5 .

$$TT_{AB} = t_B - t_A \tag{2.4}$$

$$TT_{(AB)Average} = \frac{1}{N}\sum_{i=1}^{N}(t_{Bi} - t_{Ai}) \tag{2.5}$$

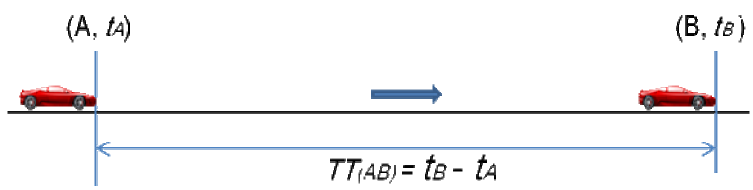

Figure 4 Measuring Travel Time on The Roadway

Multiple methods and techniques have been developed for vehicle matching (re-identification). Those methods vary greatly in their accuracy and complexity. Two well-known techniques for vehicle re-identification are license plate matching and signature matching. Description of each follows.

2.2.1.1.1 License Plate Matching

License plate matching (LPM), as the name implies, re-identifies vehicles by matching the license plates, taking advantage of the fact that each vehicle has a unique plate. In the field, there are multiple ways to implement this technique. The most naive is the manual way, in which the data is collected by human observers at the field, who write down the vehicles plates at two different points or record an audio tape and then process the data to do the matching and calculate travel time. This method requires employing, at least, one person at each check point, equipped with the required tools to do the manual data collection, since most of the time that the two points of interest are distributed over a large geographic area. To overcome this problem, a video camera is deployed at each check point to do the data collection. In this method, the vehicles plates are videotaped at the points of interest and then the videos are transcribed, either manually, by matching license plates by a human observer, or automatically, using software that does license plate characters recognition. Manual matching is considered accurate, but it is also a time- consuming process. The automatic approach saves time. But its accuracy depends to a large extent on the quality of the recognition software used [9].

LPM techniques have varying degrees of success; but overall they are considered a good choice for data collection. On the other hand, such methods are manpower-and-time consuming, which limits their ability to generate real time or near real-time results. In addition, they are affected by environmental conditions such as weather conditions and the time of day, which makes the plates sometimes hard to observe clearly. Also, such methods raise socio-political concerns of surveillance and the perceived loss of privacy as travelers may feel that they are under observation all the time.

2.2.1.1.2 Signature based Techniques

Signature –based techniques were developed to overcome the limitations of the plate matching methods by observing other characteristics of the vehicle that are less affected by environmental conditions and can be recognized much faster. Some of those characteristics are clearly observed, even in bad lighting situation, such as vehicle color, type and model. Some of them are less obvious, such as magnetic or electronic characteristics.

Signatures can come from a wide variety of detectors, including video cameras, which are a good choice for observing external characteristics and feature-based tracking [10]. Other signatures come from inductive loops [11], laser-rangefinder-based sensors [12], ultrasonic detectors [13] and automatic toll collection tag readers, which takes advantage of already installed toll tags on the vehicles to do the matching [14].

The main advantage of signature based techniques is their ability to work for long time periods without being affected by external conditions. Also, the fact that most of them communicate directly with traffic centers increases their ability to generate real-time data. The main disadvantage of this technology is the extremely high cost for installation and maintenance and the potential unreliability of the technology.

2.2.1.2 Vehicle-based Techniques

In vehicle based techniques, data collection tools are installed-in or mounted-on a moving vehicle which travels between the points of interest. Similar to road-based technologies, a wide range of techniques are used with varying degrees of complexity, ranging from test vehicles with manual data collection to advanced ITS probe vehicles equipped with more sophisticated technology.

The simplest technique of vehicle-based data collection is to use a test vehicle with manual data collection. A passenger sitting in the vehicle keeps track of the vehicle location and timestamp at each check point and report that information to the traffic center, either directly using a cell phone, or after the test is over. A more advanced technology in test vehicles uses a Distance Measuring Instrument (DMI) which determines Travel time using the speed and distance information provided by the transmission system of the test vehicle [8].

With the evolution of ITS, more sophisticated technology has been deployed on test vehicles. Data collection vehicles with ITS technology are known as ITS probe vehicles. Such vehicles use multiple techniques to collect Travel time data such as Automatic vehicle identification (AVI), in which, a vehicle with a special electronic identification tag is tracked on the traffic stream [15], or automatic vehicle location (AVL) in which vehicles are tracked using GPS technology, with two-way communication used to receive a signal from the satellite, and identify the moving vehicle position and send that information to the traffic center [8]. Cellular phones is another popular method which tracks the telephone calls to identify the probe vehicle location [16].

Advantages of using vehicle based techniques include their ability to collect data without interrupting the traffic flow or raising any privacy concerns, their ability to collect data for large areas, and the low cost per unit. On the other hand, the quality of data collected by vehicles-based techniques is directly related to the number of running vehicles at a time [17]. So to get more accurate information, more probe vehicles need to be employed, which means a large increase in both construction cost (purchase necessary equipment, install the equipment, and train personnel to operate the system and collect data) and operating cost (the gas prices and managing test vehicles at run time). As a result, such methods are mostly used for understanding the big picture of traffic patterns in the test area and not to collect large datasets.

2.2.2 Indirect Methods

While direct methods collect travel time as a function of time (Eq. 2.1) with no assumptions about the other traffic parameters, indirect methods derive Travel time as a function of other traffic parameters such as volume, speed and occupancy. Using indirect methods, the average Travel time in a time period p can be described as Eq. 2.6.

$$TT_p = F(X) \tag{2.6}$$

where X, is a vector of different traffic parameters observed during time interval p.

Traffic parameters are usually collected using point detectors such as Inductive loop detectors. These detectors provide, with varying degrees of accuracy, different basic traffic parameters, such as, traffic counts, speed, headway and occupancy within fixed time periods and at specific locations in the road network. Point detectors are the most widely installed traffic detectors in traffic network all over the world. For this reason, traffic systems that are based on point detector are considered cost-effective compared to other sensor based systems. They use already installed technology.

Indirect methods can be classified into three main categories: instantaneous, data driven and model based. Next is a brief review of each category.

2.2.2.1 Model-based Methods

Model based methods estimate and predict travel time based on traffic propagation models of traffic flow theory.

Traffic simulation, had had outstanding performance as a robust approach for quantifying traffic operations and modeling the nonlinear conditions of different traffic systems. Simulation has been employed by traffic researchers to generate Travel time. Different simulation models have been used to, estimate Travel time [18], generate data and evaluate other estimation models [19]. A wide selection of commercial simulation packages with different capabilities is available. CORSIM and VISSIM are examples of well-known traffic simulators [20].

In general, traffic simulation can be microscopic [21], macroscopic [22] or mesoscopic [23]. Microscopic models simulate the behavior of individual vehicles behavior to predict Travel time directly, based on assumptions that driver-behavior, such as, car following, gap-acceptance and risk-avoidance. Macroscopic traffic flow models predict the characteristics of a traffic stream based on its corresponding flow, average speed, density and stability properties, then derive Travel time indirectly from those parameters. Analogues of physical phenomena, such as gas and fluid dynamics, are used in describing the macroscopic behaviors of the traffic flow. Mesoscopic is similar to the macroscopic simulation but it uses a discrete flow representation of groups of vehicles.

While microscopic simulation models provide a detailed representation of the traffic process, macroscopic and mesoscopic models capture the general dynamics of large networks in much less detail. Recently, multiple hybrid models that combine different simulation models have been developed to simulate traffic systems and they have promising results [24] [25].

The main advantage of model- based methods is their ability to quantifying traffic conditions, which helps to provide deeper understanding of traffic behavior in the area of interest. Another advantage is their generic nature, since they can be applied to multiple test-beds and their flexibility as completely controlled experimental systems. On the other hand, tools in this category are well known for their computational complexity that consumes more hardware resources and requires expert personnel to develop and maintain. Also, the quality of the output depends heavily on the quality of the data that is used to build and calibrate the model.

2.2.2.2 Instantaneous Methods

Also known as “Trajectory methods”, these are very popular methods for online travel time estimation on highways. Such methods assume stationary traffic condition (speed, flow and density) at the link of interest, thereby, generalizing the point measurements on the section.

Several trajectory methods have been introduced in the literature to calculate Travel time. For a link with length L, Travel time can be calculated directly using formulas based on the distance, speed & time relation.

The most famous instantaneous method is the half distance method, which assumes that the entrance speed applies to one half of the link and the exit speed applies to the other half [26]. Figure 5 shows how this method works for a section of the road with length L and two point detectors A & B.

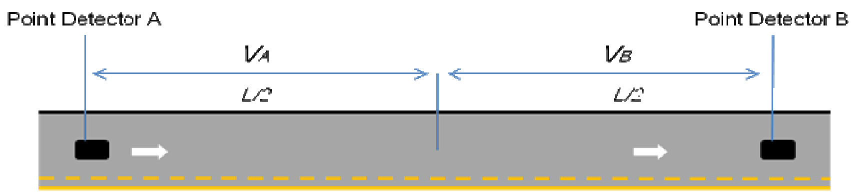

Figure 5 Duel Loop Detectors Structure (Speed Trap) for Trajectory Methods

$$TT = \frac{1}{2}\left\{\frac{L}{V_A} + \frac{L}{V_B}\right\} \tag{2.7}$$

Where:
TT: is the Travel time
L: is the link length
V_A: is the speed at point A (Upstream)
V_B: is the speed at point B (Downstream)

Instantaneous methods are well known for their simplicity and mathematical efficiency and they have shown good potential for estimating travel time on freeways; and this is because traffic conditions tend to behave in a stationary way on highways under normal circumstances. On the other hand, they are not very successful on arterial links, where the traffic has interrupted nature, which violates the main stationary traffic conditions assumption. Previous research showed that the trajectory methods performance dropped dramatically in cases of traffic delays resulting congestion or incidents [27]. More analysis of trajectory methods is provided in chapter four.

2.2.2.3 Data-Driven Methods

In this category, Travel time is derived from other traffic parameters using different mathematical and statistical relations that are all based on the usage of historical data to infer present information, relaying on the fact that traffic patterns often repeat themselves over time.

Figure 6 shows the basic operation of the data driven models and the central role the historical data plays in building such models.

There is a wide body of research that covers a large number of data driven models that have been developed or investigated to derive Travel time. Next is a brief review of the most well-known models.

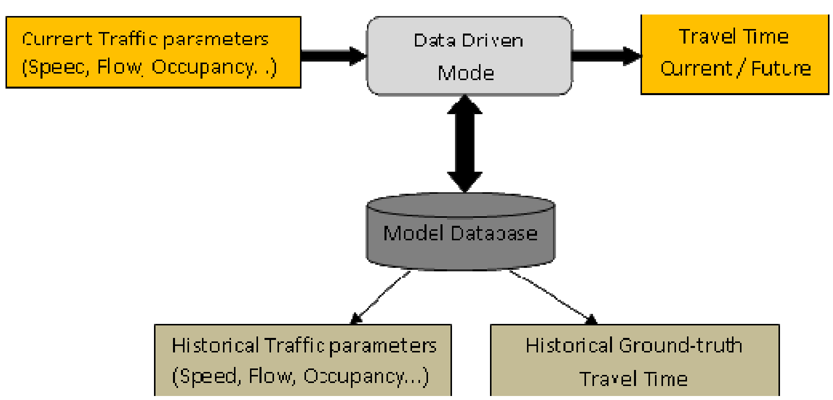

Figure 6 Data Driven Model Operation

2.2.2.3.1 Regression Methods

Regression analysis is a statistical technique that tries to explain the relation between a set of independent input variables and a dependent out variable. The ultimate goal is to derive the regression equation which describes the output variable as a function of the input variables [28].

Multiple Travel time prediction methods, based on the regression analysis have been introduced in the literature. Rice & Van Zwet, [29] predicted route Travel times through a linear regression of the sum of current instantaneous section level Travel times and historical travel times [29]. Nikovski D., et al, [30] conducted a comparison between different statistical methods, including linear regression, regression trees, and locally-weighted regression and found that linear regression is very competitive in terms of accuracy, computational time and memory resources, especially for large historical data sets.

Kwon et al, [31] used linear regression with stepwise variable selection method to estimate travel time. The results show that linear regression has a good potential for short-term Travel time forecasting [31].

2.2.2.3.2 Time Series Methods

In statistics, a time series is a sequence of measurements made consecutively in time. A time series forecasting model is a statistical method that is used to estimate or predict certain measure, by studying its behavior in the past to capture the essential features of the long-term behavior of the system.

Different time series techniques have been employed to predict travel time. Ishak & Al-Deek, [32] used a nonlinear time series traffic prediction model, focusing on the factors that have a significant impact on the forecasting accuracy of Travel times [32]. Rice & Van Zwet, [29] implemented a linear time series model in which the system coefficients vary as functions of the departure time. The model showed a promising results for small datasets, but was less accurate for larger datasets [29]. ARIMA, Autoregressive Integrated Moving Average, is a very well-known time series model that was first introduced on 1979 by Ahmed and cook [33] is used when the data is collected in a non-stationary way [34]. ARIMA(X) model, introduced by Willimas in 2001 [35], is a generalization of the ARIMA model. Although it increases the complexity of the ARIMA model, it relatively generates more accurate results [35].

Adaptive Kalman filtering is another famous Time Series model that is used in Travel time estimation studies. Kalman filter is efficient recursive filter that estimates the state of a linear dynamic system from a series of real-time noisy measurements [36].

The ability of this filter to generate accurate measures from noisy data made it a very appealing technique to study the inherently noisy traffic systems.

In the previous research, various versions of Kalman filter have been used to improve travel time estimates, by incorporating data from a small sample of probe vehicles or noisy data from point detectors [37][38][39]. It is also widely used in data cleaning and filling techniques [2][40].

2.2.2.3.3 Artificial Neural Networks

Artificial Neural networks are non-linear mathematical data modeling tools that emulate biological neural networks. An ANN can perform highly complex mappings on nonlinearly related data by inferring subtle relationships between input and output parameters. It can, in principle, generalize from a limited quantity of training data to overall trends in functional relationships.

ANN has always been an appealing technique to solve complex systems in the real world. Their nonparametric and nonlinear nature, in addition to the fact that no pre-assumption about the underlying model need to be made, increased their popularity in different research domains as a robust flexible way to solve complex nonlinear systems where formal analysis would be impossible.

In transportation engineering, ANNs have shown solid performance in modeling complex traffic systems dynamics [41], especially in solving the problem of estimating and predicting travel time. Fu and Rilett, [42] investigated the feasibility of using ANN models for estimating the O-D Travel time in a traffic network. The results showed that the ANN has great potential for modeling 0-D Travel times during non-recurring congestion. Mark et al, [43] used a freeway section to develop an artificial neural network (ANN) capable of predicting experienced travel time between two points on the transportation network. The experiments demonstrated that the ANNs were able to reasonably predict experienced travel time. Jeong and Rilett, [44] used a historical data based model, regression models, and artificial neural network (ANN) Models to predict bus arrival time. The study showed that ANN models outperformed the historical data based model and the regression models in terms of prediction accuracy. In addition to the above examples, different types of ANNs with various levels of complexity and accuracy have been employed. Feed forward [45], back propagation [46], recurrent [47], state space [48], dynamic, radial basis function [49], modular [50], and time delay [51] neural networks are examples of the different types that have been investigated.

It is common to combine ANNs with adaptive filtering techniques to improve their accuracy. For instance, Kalman filtering is a very well-known filer that has been intensively used with ANNs to estimate travel time. It is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements [52]. For more information about the Kalman file and neural networks refer to [53].

Among the data driven methods, neural networks in particular have demonstrated a promising performance. The main drawback of neural network models is the fact that they treat the system as a black box, So, in general, models that are obtained with neural networks are not understandable in terms of physical parameters, which limits its ability to describe the functional relationships that were deduced between the different system variables [54]. The mathematical approach for neural networks is described in detail in chapter 4.

2.3 Summary

This chapter provided a systematic literature review for travel time as traffic parameter. It began by defining travel time and deriving the basic key elements of the definition, which includes time, traveler, movement and route, and discussed the two main components of travel time; running time and delay time and its different scopes, whether arterial links or highway and statistical descriptions (average, individual). Next, the chapter described the data collection process. It started by distinguishing between Travel time measurement, estimation and prediction by defining each process, its scope of operation and its output.

Next, the section reviewed in detail the basic methods for Travel time measurement, estimation and prediction. The section provided a full literature review to explain each method's operation, output and merits and compared the different methods. In general, Travel time can either be measured directly from the field or derived indirectly from other traffic parameters. The direct methods can be classified into road-based and vehicle-based, while the indirect methods can be classified into instantaneous, model-based and data-driven-based methods. Figure 7 summarizes the different methods for travel time measurement, estimation and prediction.

The first three levels represent the main categories and sub categories. Lower levels are examples that can be expanded further to include new models or divide the current models into more models. This chart describes, to some level, the main methods for Travel time measurement, estimation and prediction, especially at its first three levels. However, different research perspectives might come up with a different classification.

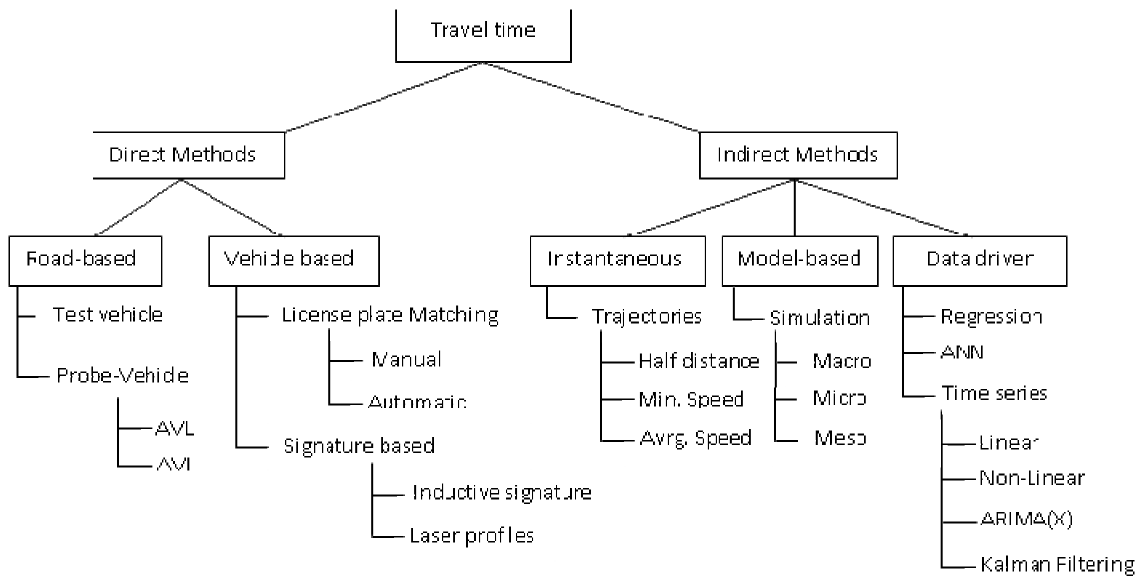

Figure 7 Travel Time Measurements, Estimation and Prediction Classification

CHAPTER III

DATA COLLECTION AND ANALYSIS

This chapter describes the data collection, verification, cleaning and preprocessing in this thesis. The first section describes the data collection process, including the required data, test geographic location, time, tools and techniques. The second section describes the data verification phase, which verifies the data quality using assumptions based the traffic flow theory and the test field information. The third section describes the data cleaning process, which is used to clean the detection errors generated by the internal malfunctioning of the point detectors. The last section describes the data pre-processing, which is the process of generating ground truth travel time information from the detectors' data. The goal of data collection, verification, cleaning and pre-processing is to collect a large, accurate dataset that can be used to build an accurate travel time estimation model.

3.1 Data Collection

Data in transportation research usually comes from different sources: simulation models, traffic centers and those collected by the researcher.

While simulation models are usually used to save the cost in (time, tools and man power), they always have the drawback of generating indirect real-life data. The data obtained from traffic centers has the advantage of being real-life data, but since the researcher has no control over the way the data was collected, a lot of data cleaning and pre-processing is needed. Collecting the data locally by the researcher has the advantage of generating real-life data, and the researcher has control over the test location, time and tools; on the other hand, it has the highest cost among the other ways (time, manpower and acquiring the tools). It is up to the researcher to decide what method to use based on his research scope, the budget and the availability of tools and manpower, keeping in mind the various tradeoffs. For this thesis, the data was collected locally by the researcher.

3.1.1 What to Collect

The main task in this thesis is to estimate travel time indirectly from other traffic parameters, using a data-driven model therefore the collected data should include two types of information, the input data, which includes basic traffic parameters such as flow, speed and occupancy, averaged and aggregated per time period, and the output dataset, which contains Ground-truth average travel time information. A description of the different traffic parameters follows.

3.1.1.1 Input dataset:

Volume: Traffic volumes are counts made for some specific time period. Usually in transportation, volumes are described as the number of vehicles that passed a certain point per hour. Traffic volume can be collected using point detectors that record the count of the vehicles passing through the detector detection space.

Headway: This parameter refers to the separation between two consecutive vehicles. Headway can be expressed in term of time or distance. Usually it is measured as the difference in time or distance between two predetermined points on adjacent vehicles (Figure 8).

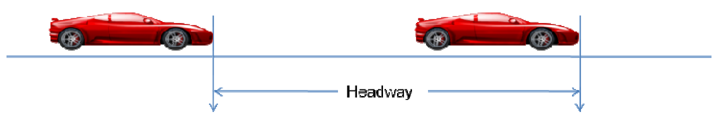

Figure 8 Headway Traffic Parameter

Speed: Speed is probably the most well-known traffic parameter for the public, and one of the most influential factors on travel time.

There are two ways to calculate average speed in a traffic stream:

- *Time mean speed (TMS)*: The average speed of all the vehicles passing a point over some specified time period. TMS can be collected using fixed point detectors.

$$TMS = \frac{1}{N}\sum_{i=1}^{N} V_i \qquad (3.1)$$

- *Space mean speed (SMS)*: The average speed of all vehicles moving over a given link over some specified time period. SMS can be collected from the field using probe vehicles or AVL techniques. Space Mean Speed can be represented as the harmonic mean of the point speed.

$$SMS = \frac{N}{\sum_{i=1}^{N} V_i} \qquad (3.2)$$

Different formulas for deriving the relation between SMS and TMS have been introduced in the literature. Usually the relation ends up being linear with location-specific constants that are derived statistically from the field data [2][55]. For this thesis, which uses data from point detectors, the TMS is used as the default speed measure.

Occupancy: is another traffic parameter that is used to quantify traffic behavior. It refers to the percentage of road way that is covered with vehicles. Point detectors measure occupancy as the percentage of time the detector was occupied.

Length: the average length of vehicles passing over a specified point on the roadway over a specified period of time. Usually, lengths are measured in feet and are used to classify vehicles. Figure 9 shows the standard classification in this thesis.

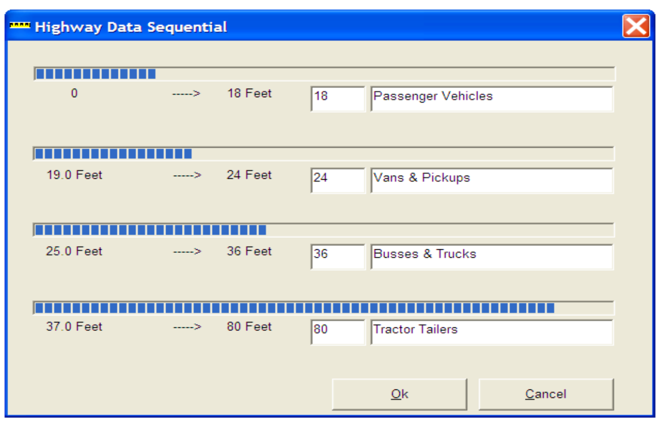

Figure 9 Vehicles Length Classification Used in The Study

3.1.1.2 Output Dataset

The output data set has one parameter only, which is the travel time parameter. While obtaining the input dataset was an easy task using the point detector, collecting travel time information was more sophisticated task. (See sections 3.3 and 3.4).

3.1.2 Geographic Location

This thesis is concerned with short continuous links (< mile) in congested urban networks, with high traffic demand, high pedestrian density, different traffic behaviors and multiple intersections. Several candidate links were observed for a week on the campus of Mississippi State University. The final selection was Hardy road (Figure 10). It is a continuous short link (.13 mile = 210 m) with 4 crosswalks and a library drop box. It ends at an intersection controlled by stop signs and experiences heavy traffic density during the day and heavy pedestrian density, which resembles the density of emergency evacuation.

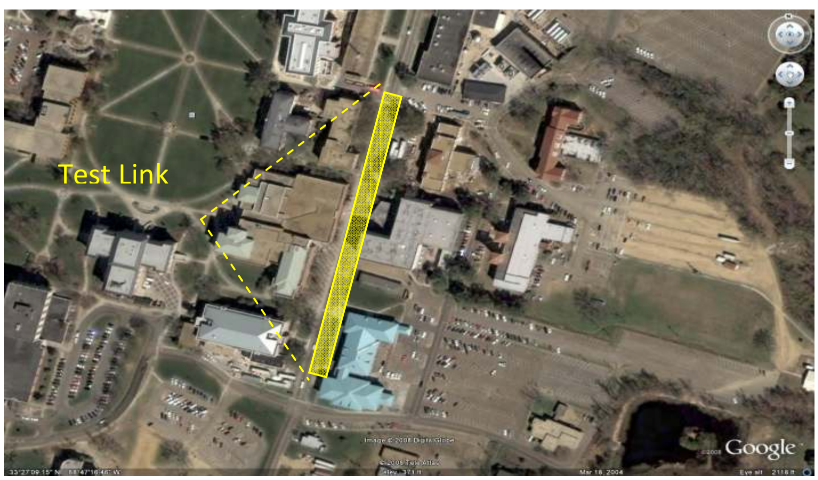

Figure 10 Geographic Location of the Test Link (Hardy Road)

Figure 11 shows the traffic demand on the selected test link represented by the vehicles count during a 24 hour period. The Figure shows that the traffic increases at 8:00 am, which is the start of the work day (Morning peak period). It hits another peak at 2:00

pm, the lunch break hour (Mid-day peak period). Finally, it hits the maximum at 6:00 pm - 7:00 pm, the time when students start to leave the campus (evening period).

3.1.3 Time

The data collection process took place between October 27, 2008 and November 14, 2008. The data was collected only during the working days of the week (Monday - Friday), 24 hours a day. Weekends were excluded from data collection because the traffic was insufficient. Table 1 shows the Gantt chart of the data collection process.

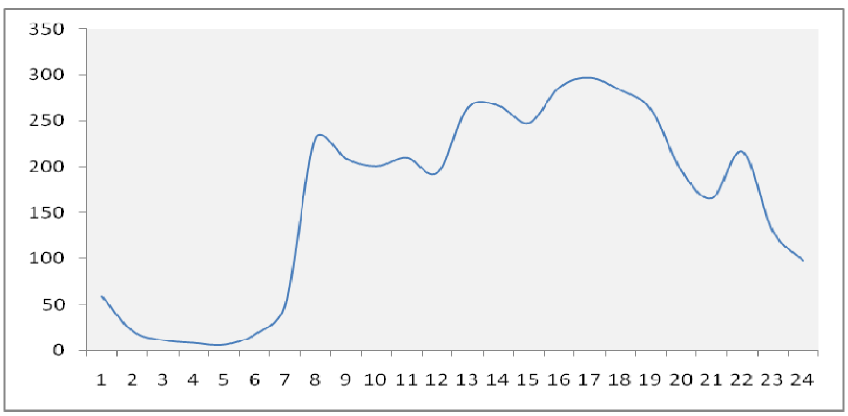

Figure 11 Traffic Volumes During The 24 Hours of The Day at Hardy Road

Table 1 Data Collection Gantt Chart

ID	Task Name	26 Oct 2008						2 Nov 2008							9 Nov 2008					
		27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Point Detectors																			
2	Video taping																			
3	Probe vehicle GPS																			

Point detectors were deployed for three weeks. The data from the first week was ignored because one of the detectors failed to collect data due to internal mal-functioning. A GPS probe vehicle took place all three weeks in separate periods (3 routes a day). Videotaping took place at the third week. Ten hours of video taping was conducted at different times on different days.

3.1.4 Tools and Techniques

Three main data collection techniques were used to prepare the dataset for this thesis. The required data includes traffic parameters (volume, occupancy, speed, average length, etc) and ground-truth travel time. The three techniques were video-based signature matching, a probe vehicle with GPS, and point detectors. The data collected by different methods was integrated to generate one accurate dataset. Next is a description of each method and its output.

3.1.4.1 Point Detectors

Point detectors were the main data source in. Four point detectors were deployed on the roadway, two in each lane, upstream and downstream (Figure 12). The goal was to construct a dual point detectors structure, or what's known a speed trap. Having this structure on a continuous link, gives a redundant-data advantage, which helps in data cleaning and verification [56].

NC-200 point detectors were used, which is a portable traffic analyzer designed to provide accurate count, speed, and classification data. The sensor is placed directly in the traffic lane to measure data, and can be installed and removed quickly and easily. The

NC-100/200 utilizes Vehicle Magnetic Imaging (VMI) technology to detect vehicle count, speed and classification. The data is easily exported to Highway Data Management (HDM) software, where it can be presented in the form of reports, charts and graphs.

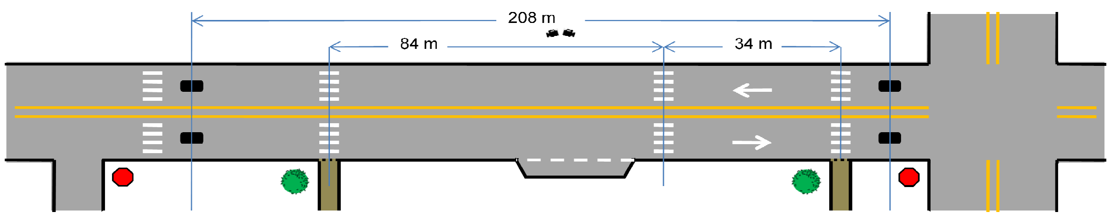

Figure 12 Point Detectors on The Test Link

NC200 detectors are less noticeable to traffic which results in more accurate information, because people tend to slow down when they notice a detector on the roadway. Figure 13 shows the detectors on the roadway.

Figure 13 Point Detectors at The Roadway

The point detectors are the main data source of both the input and the output sides of our dataset. On the input side, they provide us with the basic traffic parameters such as count, speed, headway and occupancy. On the output side, the detectors provide the timestamp of each vehicle passing over the detectors' detection space. This data will help to derive ground-truth travel time information later in this thesis. Figure 14 shows the sequential data collected by the NC200 detectors.

Sequential Detail Report

HI-Star ID: 5202	Begin: Nov/10/2008 12:00:00 AM	End: Nov/15/2008 12:00:00 AM
Street: Hardy road	Lane: To Lib 6 - 1 (C)	Hours: 120.00
State: MS	Oper: OOS	Period: Sequential
City: Mississippi State	Posted: 55	Raw Count: 19053
County: Oktobeha	AADT Factor: 1	AADT Count: 3811

Count	Date / Time	Advice	Speed	Length	Gap Time in Seconds	Headway in Feet	Tail Gating
1	2008/11/10 00:00:42.0C	Normal	32 MPH	15.00FT	0	0	No
2	2008/11/10 00:01:45.0C	Normal	28 MPH	16.00FT	0	0	Yes
3	2008/11/10 00:02:40.0C	Normal	26 MPH	17.00FT	0	0	Yes
4	2008/11/10 00:02:45.0C	Normal	25 MPH	14.00FT	0	0	Yes
5	2008/11/10 00:03:49.0C	Normal	26 MPH	14.00FT	0	0	Yes
6	2008/11/10 00:05:19.0C	Normal	28 MPH	11.00FT	0	0	Yes
7	2008/11/10 00:05:43.0C	Normal	45 MPH	16.00FT	0	0	Yes
8	2008/11/10 00:06:10.0C	Normal	30 MPH	18.00FT	0	0	Yes
9	2008/11/10 00:08:26.0C	Normal	54 MPH	22.00FT	0	0	Yes
10	2008/11/10 00:09:25.0C	Normal	32 MPH	11.00FT	0	0	Yes

Figure 14 Sequential Data Report Generated by HDS NC200 Software

Each row represents a vehicle detection, which contains vehicle sequence number, timestamp, the vehicle length, its type (based on its length), the speed, the headway information and tailgating.

3.1.4.2 Video Based Signature matching

This technique is used to collect ground-truth travel time data. Two camcorders, with high storage capacity (up to 60 GB) and high zooming ratio (up to 35X), were mounted at a midpoint on the link. The first camcorder was zoomed toward the upstream station, videotaping the front side of the vehicles passing the upstream station, and the second camcorder was zoomed at the downstream station, videotaping the backside of the vehicles passing the downstream station. The goal in mounting the two camcorders at one point was to gain more control over the videotaping process and to save the cost of running two separate stations at two different locations.

Five videotaping sessions, of two hours each, were conducted during the third week of the data collection process. Table 2 shows the sessions distribution over the week.

Table 2 Videotaping Sessions Gantt Chart

Date	Time									
	8	9	10	11	12	13	14	15	16	17
Monday, November 10, 2008										
Tuesday, November 11, 2008										
Wednesday, November 12, 2008										
Thursday, November 13, 2008										
Friday, November 14, 2008										

The data collected by video was used to collect ground truth travel time and volume information. The two camcorders were synchronized to start taping at the same time and the internal clock of each camera was also synchronized with the detectors clock, by adjusting them to the time of the computer machine which was used to program the point detectors.

The videos were then observed by a human observer for travel time and count analysis. As implied in the literature review, this is a time consuming process. Each hour of videotaping takes up to four hours of analysis. To speed up the analysis process, a C# tool was developed to run the upstream and the downstream videos and add vehicles into a special database to do the matching. This tool does not do any kind of image processing. The task of matching is left to the human observer, who monitors the two videos and saves vehicles' entries to the DB (

). Table 3 shows the DB Tables used to store the vehicles information.

Table 3 Videotaping Database Tables

Station 1		
No	Description	Time
22	White Toyota	1:09:12
23	Black SUV	1:09:25
24	Red Truck	1:09:45
25	Blue Mustang	1:10:01

Station 2		
No	Description	Time
26	Silver Honda	1:09:58
27	white Toyota	1:10:03
28	Blue Mustang	1:10:24
29	Red Truck	1:10:33

From the above Tables, data count is simply the number of vehicles that were recorded and travel time is the difference between the timestamps of each couple of matching vehicles.

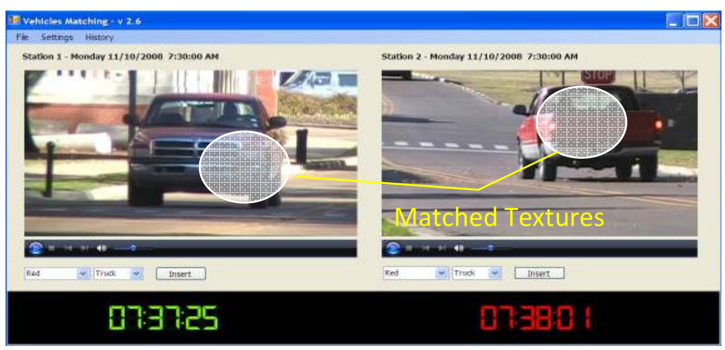

Figure 15 Signature Matching Tool

3.1.4.3 Probe Vehicle with GPS

A probe vehicle was another technique used for data collection. A test vehicle was equipped with a GPS device which was mounted on the roof of the vehicle and connected to a laptop provided with special software to manage the GPS data using a USB cable.

The cable was passed through the passenger side window and the laptop was operated by the passengers in the vehicle. The GPS device used was a GPS 18 Deluxe USB GPS Sensor with nRoute and City Select Navigation Software provided by GARMIN® (Figure 16).

In order to obtain “ground truth” data, the vehicle was driven on the test area with the GPS programmed to log the vehicle speed and position every second. With the known position of each station and the known position of the vehicle it was possible to determine the travel time between the two stations. The data we collected using the GPS will be used to assess some of the travel time estimation methods later in this thesis.

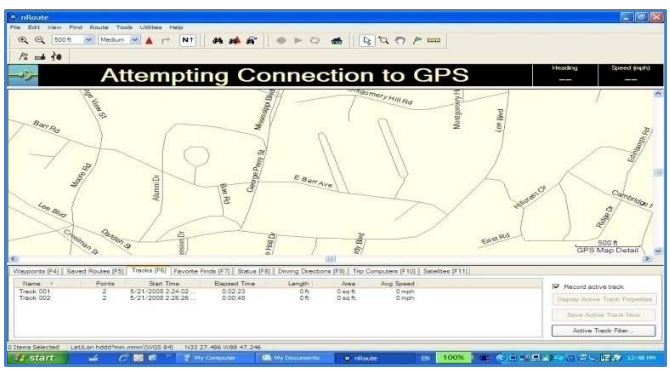

Figure 16 GPS Data

After the data was collected, a data quality analysis process was conducted to test the data validity, analyze the traffic behavior on the test link and clean the data. Only the data collected in the third week on the right lane of the road was considered. The data verification and cleaning process is described next.

3.2 Data Verification (Traffic Flow Characteristics)

Traffic data quality (TDQ) is a very important issue in the ITS community. The efficiency of ITS applications depends to a large extent on the quality of the data used. In this thesis, the quality of the collected dataset was assessed using a set of theories based on traffic flow theory, the test field information, and data integrity between the downstream and the upstream stations. The goal is to measure how closely the collected data matches the actual and expected conditions.

To speed up the analysis process, a C# tool was developed to read the NC 200 detectors database and arrange and aggregate the data (volume, time mean speed and average vehicles length) at different time intervals (1 minute – 1 hour). The tool has also options to perform analysis on sub datasets, and compare data from different detectors.

3.2.1 Traffic Flow Theory Assumptions

The main assumption under this category is based on the fact that different traffic parameters are expected to behave in correspondence to the daily schedule of the campus and in conformance with the traffic flow theory.

I. Volume Assumption:

The traffic volume is expected to increase dramatically during the day, especially at the peak hours, and decreases at night. The traffic is also assumed to follow a certain daily pattern during the week. To verify the assumption, different charts were generated to plot the relation between the traffic (represented by the hourly volume) and the time. Figure 17 shows the relation between the volume (hourly rate) and the time of the day,

Figure 18 shows the relation of the traffic volume (hourly rate) and the day of the week. The chart shows that the traffic (represented by the volume) daily pattern repeats itself each day of the week. Verifying the above two assumptions provided the first proof of the collected data validity.

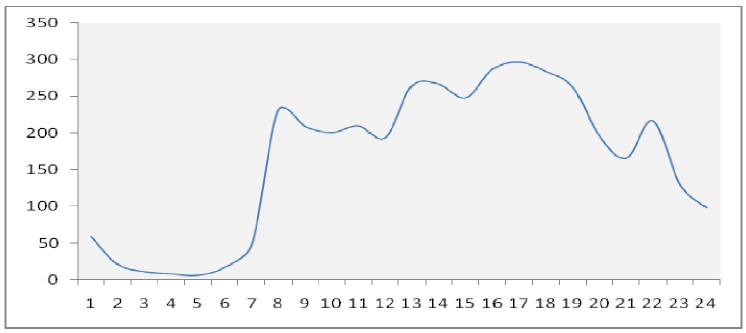

Figure 17 The Hourly Volume on The Test Link

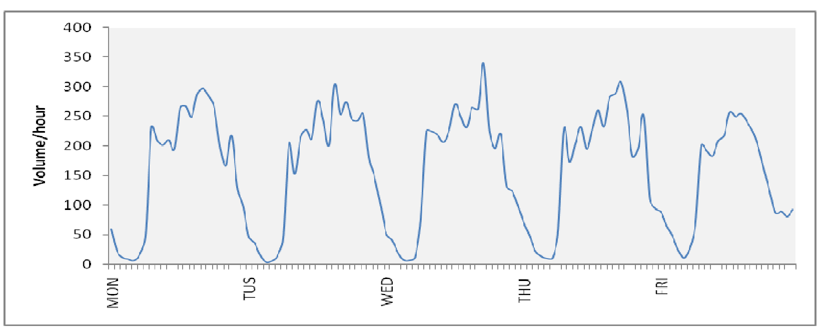

Figure 18 The Daily Pattern of The Traffic Volume During Week

II. Volume vs. Speed Assumption:

The second assumption is that the average speed detected by the point detectors is inversely related to the traffic volume. This assumption is based on the fact that when traffic volume increases in short links, the congestion possibility increases, and so, the time mean speed decreases. Mathematically, this assumption is based on the flow-speed-density relation (Eq. 3.3)

$$q = k \times v \tag{3.3}$$

where:
q: flow (vehicle/hour)
k: Density(vehicle/mile)
v: Speed (mile/hour)

Figure 19 shows that the average speed goes down whenever the volume increases and vice versa. The value of the speed was magnified in the above chart to emphasize the inverse relation.

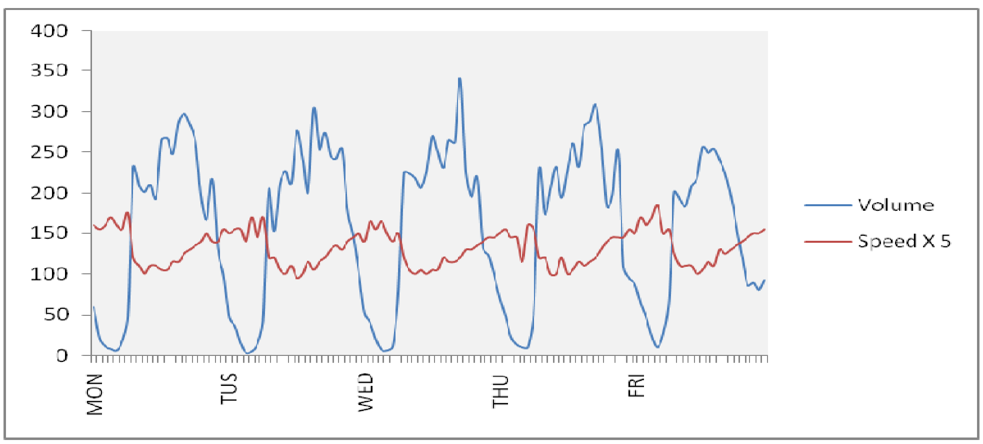

Figure 19 Volume vs. Speed Data During The Week

III. Volume vs. Occupancy Assumption:

In transportation engineering, occupancy refers to the percentage of roadway that is covered with vehicles. Point detectors measure occupancy as the percentage of time the detector was occupied, i.e., covered with a vehicle. So the occupancy is expected to be directly related to the volume. Figure 20 shows that the occupancy values reach in maximum at the peak hours. This validates the assumption.

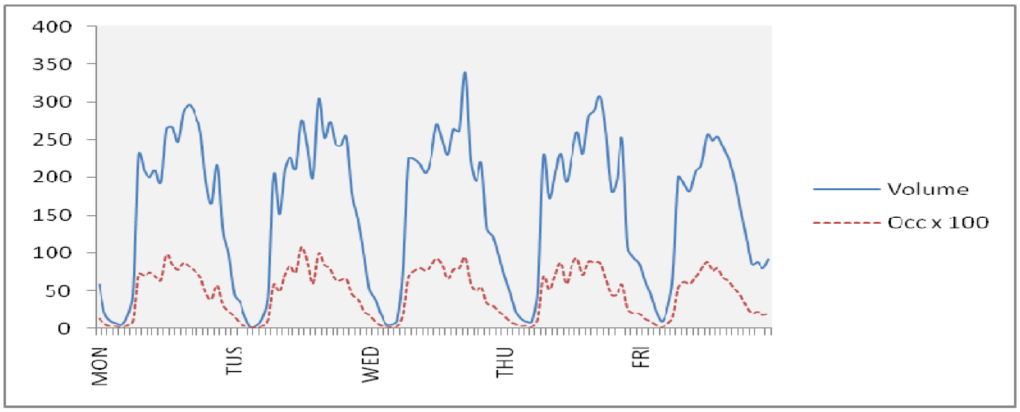

Figure 20 The Direct Relationship Between Volume and Occupancy

3.2.2 Data Integrity Assumptions

Assumptions under this category are based on the deployed dual loop detectors structure (Speed Trap). The different traffic parameters from the upstream and downstream detectors are compared.

I. Speed assumption:

The Time Mean Speed at the downstream is less than the Time Mean Speed at the upstream station. This assumption is based on the fact that the downstream end of the road is directly followed by a stop-sign-controlled intersection, and so, vehicles are expected to slow down at that end. To verify this assumption, the hourly Time Mean Speed collected by both detectors was compared. Figure 21 clearly shows that during the week, the downstream speed was always less than the upstream speed, which verifies our assumption and gives another proof about the collected data validity.

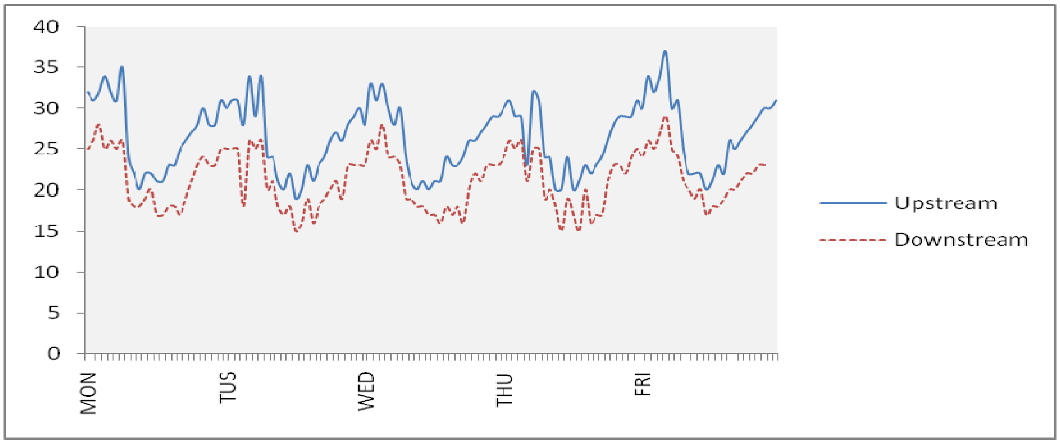

Figure 21 Downstream vs. Upstream Time Mean Speed

Figure 22 shows the speed distribution for upstream and downstream stations.

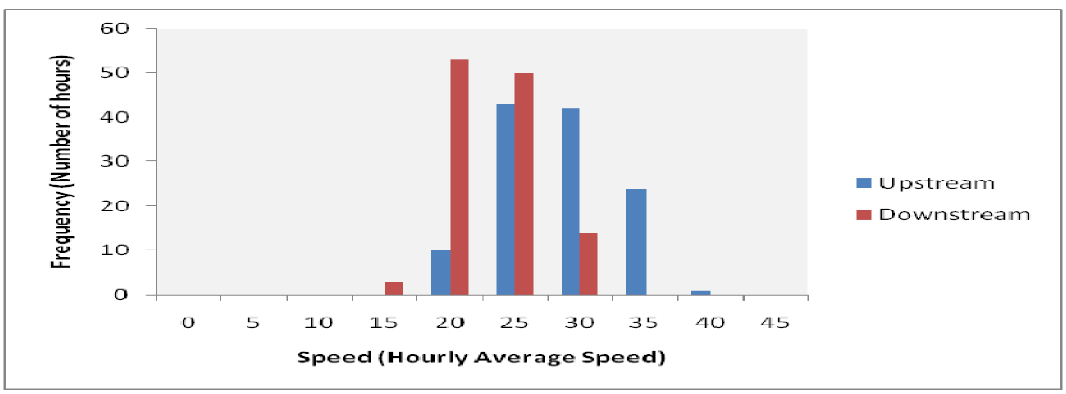

Figure 22 Speed Distribution

II. Volume assumption:

This assumption implies that the point detectors at both the upstream and the downstream stations should detect the exact same number of vehicles. This assumption is based on the fact that the test link between the two stations is continuous with no exits, so there is no way for any vehicle that has passed the upstream station to exit the test link except from the downstream station. Moreover, there is no other source of traffic between the two stations. To verify this assumption, the hourly volumes from both stations are compared in Figure 23.

The Figure shows that there is a difference between the number of vehicles detected at the upstream and the downstream stations, especially at the peak hours of the day. The chart shows that the downstream detector detected more vehicles than the upstream. Clearly, the assumption was not verified, and so, a data cleaning procedure was needed to deal with the difference.

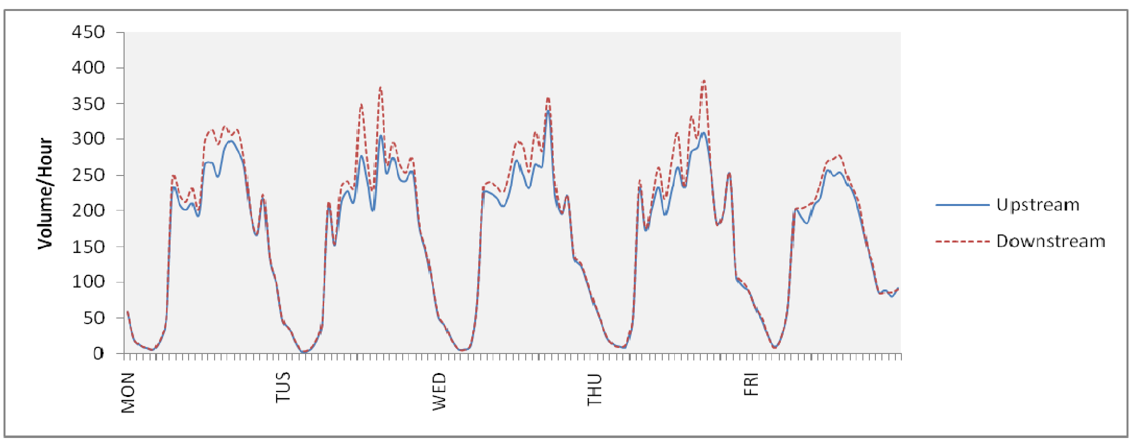

Figure 23 Downstream vs. Upstream Volume

3.3 Data Cleaning

An interesting observation about the chart in Figure 15 is that there has not been a case where the upstream station detected more vehicles than the downstream station. This observation suggests two sources of error. Either the upstream detector failed to detect some vehicles or the downstream detector "over detected" vehicles.

Another fact is that the set of detectors is 99.9% efficient in detecting vehicles (0.1% chance of missing a vehicle). This information is provided by the manufacturer. In addition, it was verified locally by testing the set of detectors previously using ground truth-data.

The above two facts imply that the error was in the downstream station, i.e., the downstream detector has over detected the volume.

To verify the assumption, the ground truth data collected using the camcorders was used. The ground-truth data from different days was compared with the data from both detectors. Table 4 shows the data comparison for 5 different hours.

Table 4 Vehicles Count, Upstream, Downstream and Actual

Time	Upstream Count	Downstream Count	Actual Count
Monday 08:00 - 09:00	209	223	209
Monday 13:00 - 14:00	267	313	266
Wednesday 09:00 - 10:00	218	234	218
Wednesday 12:00 - 13:00	270	295	268
Friday 09:00 - 10:00	183	207	183
MAE[1] (Vehicle)	.6	25.6	

The Table shows a big difference in the Mean Absolute Error of both detectors which verifies the assumption that the downstream detector was the problem.

[1] Mean Absolute Error

3.3.1 Detectors Error Description

As described above, the downstream station witnesses multiple congested periods during the day, especially at the peak hours. This is reflected in the diagram in Figure 24, which compares the occupancy at both upstream and downstream detectors.

The chart shows that the occupancy (the time the detector was occupied) jumps during the peak hours for both detectors. It also shows that for the same number of vehicles, the downstream detector has the highest occupancy rate (close to 100%), which reflects the congested state of that point of the link.

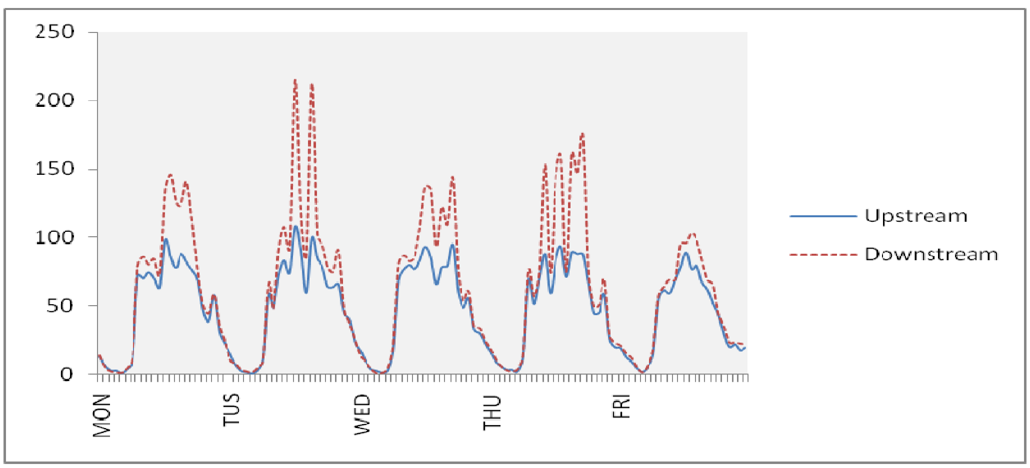

Figure 24 Downstream vs. Upstream Occupancy

A closer look was taken at the sequential data recorded by the downstream detector. Two main things were observed. At peak time, the density of vehicles with “Less than the minimum” and “Exceeded Maximum length” types increases, and the density of consecutive detections with the exact same timestamp also increases. On the field, no two vehicles can have the exact same timestamp; also, no vehicle can have a size less than 0 feet and a vehicle of length more than 80 feet is highly unlikely in the campus network. So, any presence of "not-normal" vehicles or vehicles with the same timestamp represents a suspicious case of detection error, caused by malfunction of the detectors or traffic analyzers during the congested period.

The set of traffic analyzers used to collect the data uses Vehicle Magnetic Imaging (VMI) to detect vehicles as they move through the earth's magnetic field. Every vehicle has parts that are constructed from iron. When a vehicle passes over the Traffic Analyzer, the iron parts interfere with the earth's magnetic field. This disturbance creates electrical signal changes in the Traffic Analyzer sensors. As a result, the Traffic Analyzer can determine vehicle presence, count each vehicle, measure vehicle speed, and record vehicle length.

At free flow velocities, the time each detector is occupied by a vehicle (the on-time) should be virtually identical for both upstream and downstream stations, which means any vehicle detected by the upstream station should also be detected by the downstream station. The ideal performance of the speed trap is represented in Figure 25.

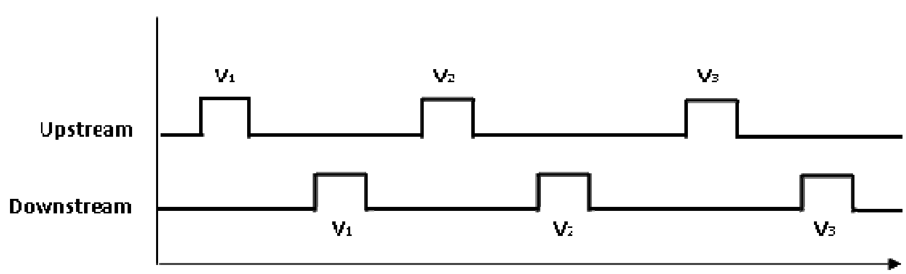

Figure 25 Normal Operation of The Speed Trap

The above diagram shows an ideal operation of the speed trap. At the upstream detector, three pulses (on-times) were detected, which mean three vehicles have passed over it (V1, V2, V3). The downstream detector also has three pulses which shows that all vehicles detected by the upstream station have also been detected by the downstream station (Table 5).

Table 5 Data of Normal Operation of The Speed Trap

No	Upstream		Downstream	
	Timestamp	Type	Timestamp	Type
1	11:26:01	Normal	11:26:41	Normal
2	11:26:31	Normal	11:26:55	Normal
3	11:27:46	Normal	11:28:14	Normal

At congestion times, the speed drops dramatically and some vehicles sometimes come to a complete stop over the detector, or do multiple consecutive runs and stop before completely getting out the of detection space of the detector. This kind of driving behavior causes the magnetic field over the detector to change multiple times (on/off) for the same vehicle, and so, generates the detection errors.

Two main types of error usually take place in this situation: multiple-detections and pulse break up. In both cases, the detector detects the same vehicle more than one time. Figure 26 describes this error. The Figure shows that only three vehicles passed the upstream detector with nearly an equal amount of time between the two consecutive vehicles. On the other hand, the downstream station detected four vehicles.

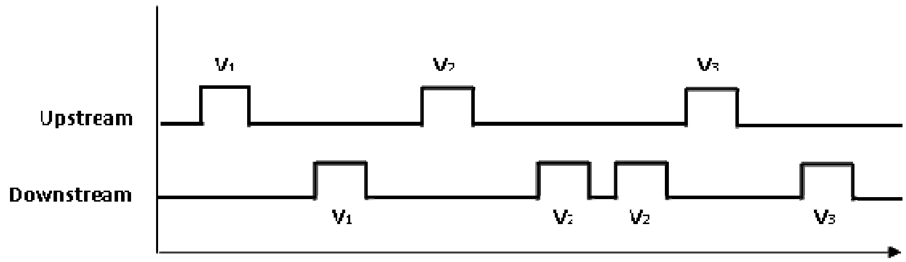

Figure 26 Multiple Detection Error

Figure 18 shows the multiple-detection error, in which vehicle two has altered the magnetic field over the detector twice, and so, has been detected twice. Usually, the time between the multiple detections of the same vehicle is less than one second. Table 6 shows the multiple-detection error at the downstream detector.

Table 6 Data Represents Multiple Detection Error in The Speed Trap

No	Upstream		Downstream	
	Timestamp	Type	Timestamp	Type
1	11:26:01	Normal	11:26:41	Normal
2	11:26:31	Normal	11:26:55	Normal
3	11:27:46	Normal	11:26:55	Normal
4			11:28:14	Normal

The pulse break up error happens when a single vehicle registers multiple detections because the sensor output flickers off and back on, breaking a single detection pulse into multiple small pulses. Figure 27 shows the pulse break up error.

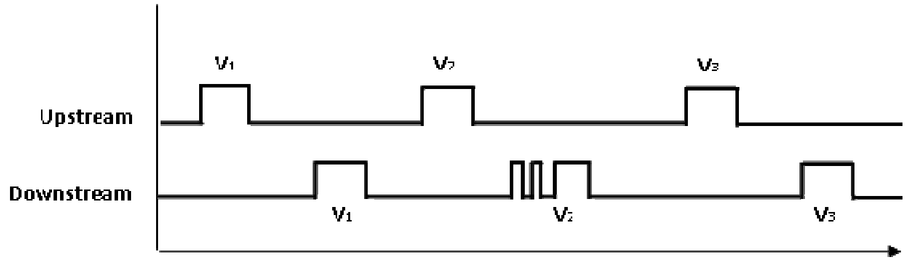

Figure 27 Pulse Breakup Error

In Figure 27, the pulse for the second vehicle has been broken up into three small pulses, which mean the detector has recorded two more detections for a single vehicle. This kind of error is more common than the multiple-detection error, and usually, in this case, the detector records a sequence of multiple "not-normal" detections. Table 7 shows the pulse breakup error.

Table 7 Data Represents Pulse Breakup Error

No	Upstream		Downstream	
	Timestamp	Type	Timestamp	Type
1	11:26:01	Normal	11:26:41	Normal
2	11:26:31	Normal	11:26:55	Less than Minimum Length
	11:27:46	Normal	11:26:56	Exceeded Maximum Length
3			11:26:57	Less than Minimum Length
4			11:28:14	Normal

3.3.2 Data Cleaning Algorithm

To clean the data, an algorithm has been developed to look for "suspicious" cases and clean them. The algorithm scans the sequential dataset looking for vehicles with the exact same timestamp and only keeps one of them. It does the same for vehicles with "Less than the minimum" or "Exceeded Maximum length" types and only keeps one of them as a representative of the one vehicle that was registered multiple times.

Example:

Table 8 represents a subset from the sequential data of the downstream detector at peak time.

Table 8 Sequential Detector Data at Peak Time

Id	Time	Type	Length
1898	2008/11/10 13:59:39.00	Less than Minimum Length	0.00
1899	2008/11/10 13:59:39.00	Less than Minimum Length	0.00
1900	2008/11/10 14:00:42.00	Exceeded Maximum Length	14.00
1901	2008/11/10 14:00:24.00	Normal	14.00
1903	2008/11/10 14:00:57.00	Less than Minimum Length	0.00
1904	2008/11/10 14:00:59.00	Normal	13.00

1- Entries 1898 and 1899 in Table 8 have the exact timestamp. So one of them is deleted (Table 9).

Table 9 Detector Sequential Data After Cleaning Entry 1898.

Id	Time	Type	Length
1899	2008/11/10 13:59:39.00	Less than Minimum Length	0.00
1900	2008/11/10 14:00:42.00	Exceeded Maximum Length	14.00
1901	2008/11/10 14:00:24.00	Normal	14.00
1903	2008/11/10 14:00:57.00	Less than Minimum Length	0.00
1904	2008/11/10 14:00:59.00	Normal	13.00

2- Entries 1899 and 1900 in Table 9 are examined. Their time is not the same, but they are both not normal (less than minimum length & exceeded maximum length). Therefore 1899 is deleted.

Table 10 Detector Sequential Data After Cleaning Entry 1899

Id	Time	Type	Length
1900	2008/11/10 14:00:42.00	Exceeded Maximum Length	14.00
1901	2008/11/10 14:00:56.00	Normal	14.00
1903	2008/11/10 14:00:57.00	Less than Minimum Length	0.00
1904	2008/11/10 14:00:59.00	Normal	13.00

3- Entries 1900 and 1901 in Table 10 are examined. Their time is not the same: 1901 is normal, 1900 is not normal and the time difference between them is more than 4 seconds. So they are both kept (Table 11).

Table 11 Detector Sequential Data After Step 3

Id	Time	Type	Length
1900	2008/11/10 14:00:42.00	Exceeded Maximum Length	14.00
1901	2008/11/10 14:00:56.00	Normal	14.00
1902	2008/11/10 14:00:57.00	Less than Minimum Length	0.00
1903	2008/11/10 15:00:22.00	Normal	13.00

4- Entries 1901 and 1902 in Table 11 are examined. One of them is not normal and the time difference between them is less than 4 seconds. So the not normal entry is deleted (Table 12)

Table 12 Detector Sequential Data After Step 4

Id	Time	Type	Length
1900	2008/11/10 14:00:42.00	Exceeded Maximum Length	14.00
1901	2008/11/10 14:00:56.00	Normal	14.00
1904	2008/11/10 15:00:22.00	Normal	13.00

5- 1901 and 1904 in Table 12 are examined. They are both normal and they have different timestamps, so both of them are kept (Table 13).

Table 13 Cleaned Data

Id	Time	Type	Length
1900	2008/11/10 14:00:42.00	Exceeded Maximum Length	14.00
1901	2008/11/10 14:00:56.00	Normal	14.00
1904	2008/11/10 15:00:22.00	Normal	13.00

The algorithm deleted three records that represent detection errors and kept three records that represent the three vehicles that actually passed over the detector at that time.

A C# tool was developed to run this algorithm on both downstream and upstream sequential data. The tool was fed the weekly data. The results were verified using the ground truth data, and also compared with the upstream detector. Table 12 shows a large decrease in the difference in the number of detected vehicles on both stations

Table 14   Data After and Before Cleaning

	Upstream	Downstream	Difference
Original	19055	20501	1446
Cleaned	18994	19182	188

Figure 28 shows that the two lines of both the upstream and downstream volumes are nearly identical after cleaning the data.

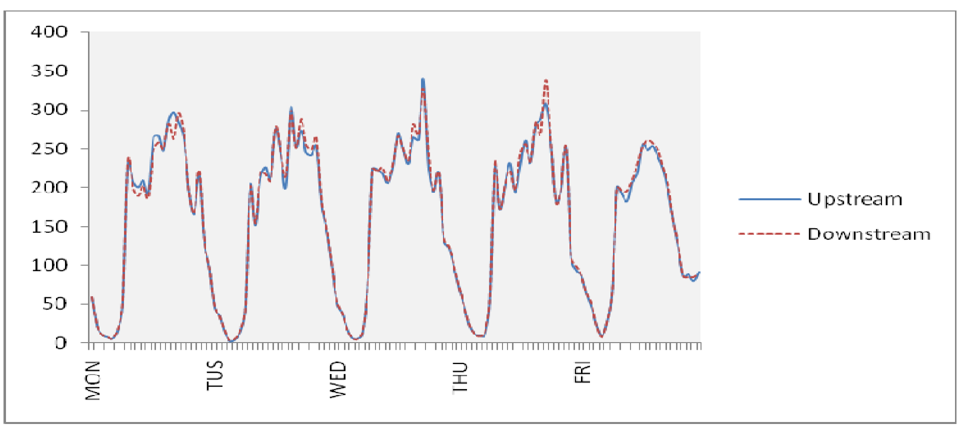

Figure 28   Hourly Volume Values After Data Cleaning

3.4 Data Pre-Processing

Speed traps represent an appealing structure for traffic engineers to derive travel time information from. Many procedures in the literature have been introduced to work with such structure and showed promising results. The methods for deriving travel time from of dual loop detectors structure can be classified into two main categories. The first category is for the trajectory methods [57] (section 1.2.2.2) and the second category is based on using the speed trap as a signature based system. By comparing certain attributes of the vehicle, such as the vehicle length [58] detected by the dual detectors structure, the vehicles can be identified and the travel time is simply calculated as the difference between the timestamps the vehicle has been registered by each detector. In this section, another way for deriving travel time from dual loop detectors data is introduced. This method is based on the fact that each vehicle passes the upstream station cannot exit the link except from the downstream end and there is no other source of traffic between the two detectors.

Using the sequential data of the dual loop detectors to calculate travel time is shown in Figure 29.

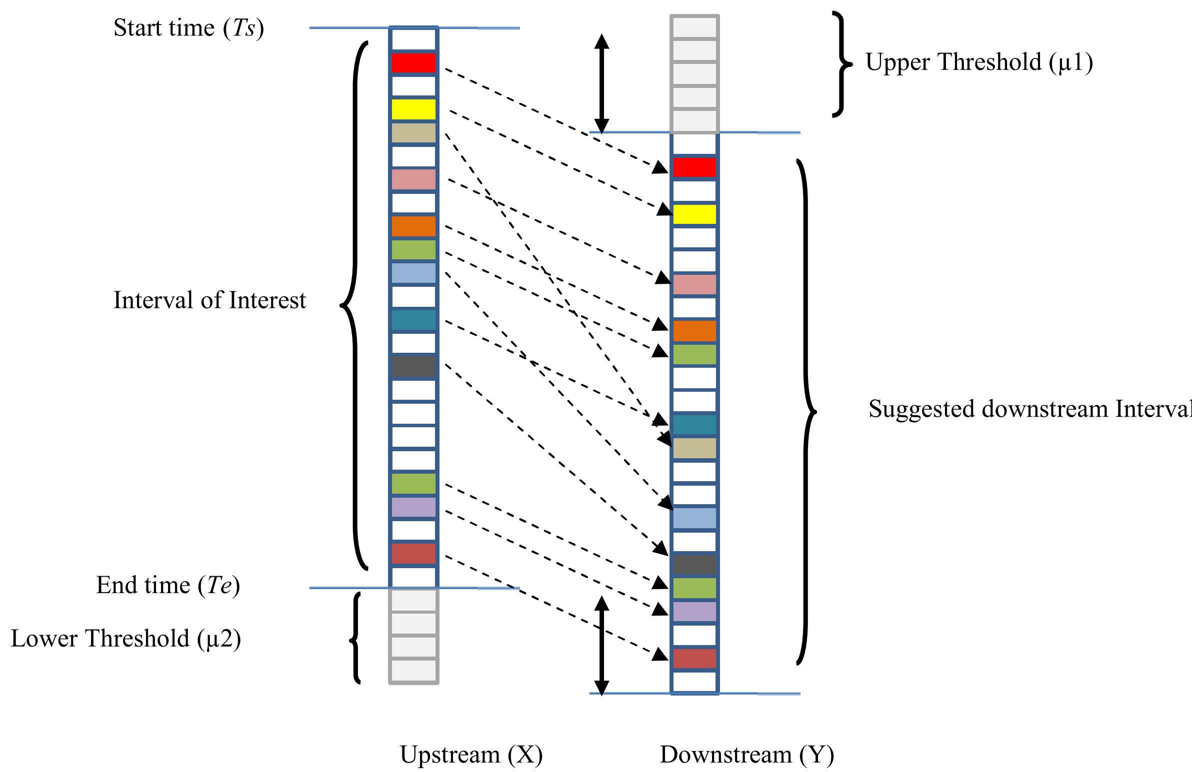

Figure 29 Deriving Travel Time from Speed Trap Algorithm

In Figure 21 the upstream column X (Eq. 3.1) represents the vehicles passed over the upstream detector during the time interval (T_s, T_e). Each vehicle can be represented as the pair (s_i, t_i) where s_i is the signature of the vehicle and t_i is the timestamp of the time the vehicle passed over the detector.

The downstream column Y (Eq. 3.2) represents the vehicles passed over the downstream detector during the time interval$(T_e + \mu 2, T_s + \mu 1)$. Each vehicle can be represented as a pair (s_j, t_j) where s_j is the signature of the vehicle and t_j is its timestamp or the time the vehicle passed over the detector.

$$X = \{(s_i, t_i) |\ 1 < i < N\} \tag{3.4}$$

$$Y = \{(s_j, t_j) \mid 1 < j < M\} \tag{3.5}$$

The colored boxes and the arrows represent the vehicles' relative order. It is not necessary that the vehicles keep their relative order between the upstream and the downstream stations.

Determining the best upstream aggregation interval was done based on the ground truth travel time data, rather than found in a statistically rigorous fashion way. Multiple factors affect the size of the time interval used to aggregate travel time information, such as the fluctuation of the traffic demand per time, the ground truth travel time information, and the molding approach (real-time, near-real-time). In general, the selected time interval should satisfy two main conditions: it should be as small as possible to generate near-real-time meaningful information and it should be larger than the maximum travel time observed on the link. It is important for the interval to be small, because the larger the interval the more details will be lost.

The field study showed that traffic demands changes each 5-8 minutes interval, and the maximum travel time observed on the link was 127 seconds. So a 5 minutes interval will be the best choice under these conditions. Figure 30 shows the ground truth information and the different aggregating intervals. The diagram shows that the 5 minutes interval best represents the data without losing much detail.

Determining the threshold values ($\mu 1, \mu 2$) of the adjacent downstream interval is more like calibrating the system. The downstream interval start-time is approximated based on the travel time measured using the videos data. For each interval, threshold time is derived based on traffic volume in that interval and the time of the day. The different thresholds were stored in a lookup Table and the required value was looked up based on the count of vehicles during that period and the time of the day the interval belongs to.

After determining the upstream and the downstream intervals, travel times are calculated as the difference between travel times in both intervals. Ideally, after cleaning the data and determining the right threshold values, both intervals should have the same number of vehicles and the travel time can then be calculated using Eq. 3.6.

$$TTp = \frac{1}{N}\left[\sum_{i=1}^{N} T_{di} - \sum_{i=1}^{N} T_{ui}\right] \tag{3.6}$$

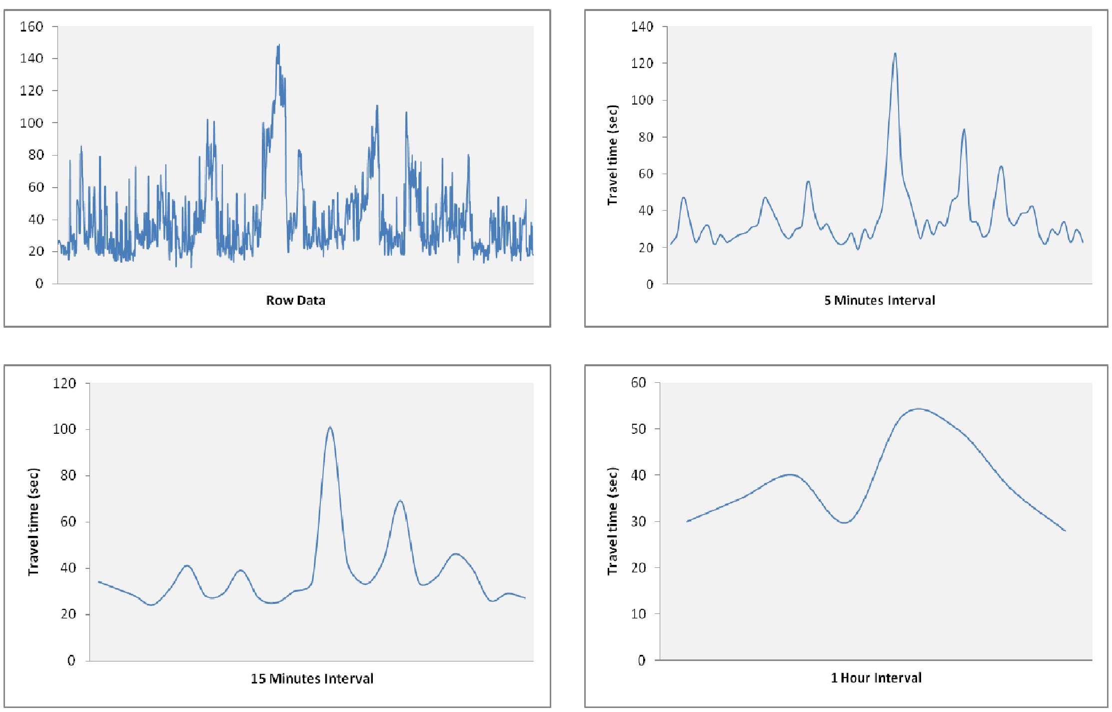

Figure 30 Different Aggregation Intervals

The pseudo-code for deriving travel time from the dual loop detectors structure on a continuous link works as follow:

Assuming that:

P_1: Lower Interval boundary
P_2: Upper Interval boundary
P: Interval length
T_u : Upstream interval dataset (timestamps of all the vehicles detected by the upstream station in the time interval P)
T_{uMin} : Minimum timestamp in the upstream interval
T_{uMax}: Maximum travel time in the upstream interval
N: Number of vehicles in the upstream interval
T_d: Downstream interval dataset
T_{dMin}: Minimum timestamp in the downstream interval
M: Number of vehicles in the upstream interval
Threshold: approximated travel time at the selected interval
T_{uAvrg}: Average of timestamps of the upstream interval
T_{dAvrg}: Average of timestamps of the downstream interval
TT_p: Average travel time at the selected time interval (p)

The pseudo code explains the algorithm

T_u = SELECT timestamp FROM upstream WHERE timestamp BETWEEN P_1 AND P_2
$N = count(T_u)$
$T_{dMin} = T_{uMin} + Threshold$
T_d= SELECT TOP(N) timestamp FROM downstream WHERE timestamp > T_{dMin}
$T_{uAvrg} = Avarage(T_u)$
$T_{dAvrg} = Avarage(T_d)$
$TT_p = T_{dAvrg} - T_{uAvrg}$
$P_1 = P_2$
$P_2 = P_2 + P$
Repeat from 1

Example: Calculate travel time at 5 minutes time intervals starting from 0:09:50.

Starting with average travel time at the time interval (P_1, P_2)= (0:09:50 – 0:09:55).
$P = 0{:}05{:}00$
$T_u = < 9{:}50{:}07, 9{:}50{:}07, \ldots, 9{:}54{:}55 >$

$T_{uMin} = 9:50:07$
$T_{uMax} = 9:54:55$
$N = 13$
$T_{dMin} = 9:50:27$
$T_u = < 9:50:29, 9:50:07, \ldots, 9:55:15 > ($
Table **15**)
$T_{uAvrg} = \frac{1}{13} \sum_{i=1}^{N} T_{ui} = 9:52:15$
$T_{dAvrg} = \frac{1}{13} \sum_{i=1}^{N} T_{di} = 9:52:46$
$TT_p = 9:25:46 - 9:52:15 = 00:00:31\ sec$
$P_1 = 0:09:55$
$P_2 = 0:10:00$
Repeat

Table 15 Upstream and Downstream Sequential Data (09:50 am – 09:55 am)

Interval	No.	Upstream Timestamp	Upstream Timestamp
Interval (0:09:50 - 0:09:55)	1	9:50:07	9:50:29
	2	9:50:07	9:50:31
	3	9:50:11	9:50:34
	4	9:51:15	9:51:44
	5	9:51:24	9:51:48
	6	9:52:36	9:52:54
	7	9:52:39	9:53:01
	8	9:52:41	9:53:03
	9	9:52:27	9:53:06
	10	9:53:00	9:53:43
	11	9:53:21	9:54:53
	12	9:54:27	9:54:56
	13	9:54:55	9:55:15

The results were validated against the ground truth data collected by the video tapes. The comparison showed that the above method was able to generate travel time with an absolute error[2] of 1.5 seconds. Figure 31 compares the Actual travel time data and the data generated by the algorithm.

The drawbacks of this algorithm come from the fact that it depends heavily on the threshold value, and so it needs to be calibrated carefully with the largest possible ground truth data.

In addition, the algorithm is very sensitive to the detectors mal-functioning errors such as the multiple-detection and the pulse-breakup errors that can cause the algorithm to return negative or very large values. So it is expected to work much better on the freeway. Finally, this algorithm can only be used on continuous links (sections from the highway) where each vehicle that passes the upstream station has no way to exit the test link except from the downstream station with no other sources of traffic.

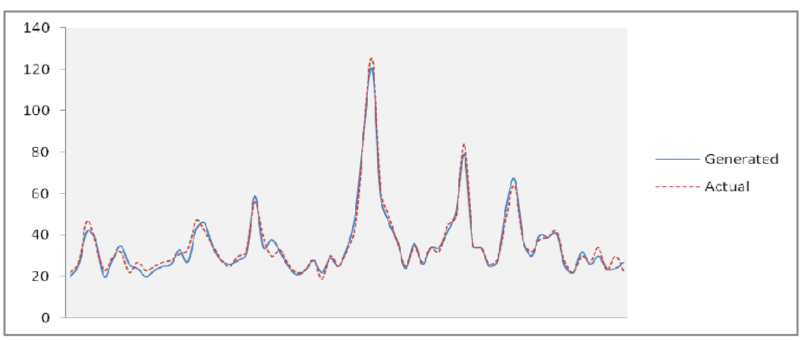

Figure 31 Actual vs. Calculated Travel Time

3.5 Summary

This chapter described the data collection and preprocessing. It started by defining the required data for this thesis, the geographic location, test time, data collection

[2] See Chapter 4

methods and tools, and output of each method. The next section describe the data verification process and the data validity assumptions that were used to assess the data quality, based on traffic flow theory and the integrity between point detectors data in the speed trap. In section three, the chapter provided a brief description of the data cleaning algorithm. Finally, the algorithm for deriving the data from the dual loop detector structure (speed trap) was described. The output of the process described in this section is a dataset that contains the information needed for this thesis.

CHAPTER IV

TRAVEL TIME ESTIMATION

This chapter describes the analysis part of this study. The first section describes the problem's input, output and the error measures which are used to assess and compare the different estimation methods. Section two presents the different methods for estimating travel time. Highway Trajectories, Multiple Linear Regression (MLR), Artificial Neural Networks (ANN) and K Nearest Neighbor (K-NN) are investigated in this section. The last part of the chapter concludes the results of the analysis and describes the potential future work in this domain.

4.1 Problem description

The data pre-processing phase, which includes data collection, verification, and cleaning, has created a dataset with set of ground-truth data samples; each sample includes set of observed traffic parameters (volume, speed, occupancy, length) in a certain time interval and the corresponding observed travel time parameter at that interval. For this thesis, the data was aggregated in 5 minutes intervals. Table 16 shows a sub-dataset from the problem final dataset.

Each entry in the input dataset can be represented as a vector (X_p) (Eq. 4.1)

$$X_p = (\langle x_1, x_2, x_3, \dots, x_n \rangle, \hat{x}) \tag{4.1}$$

Table 16 System Dataset

Interval	Count (v)	speed (mph)	Length (ft)	occupancy (%)	TT (sec)
11/10/2008 7:05	13	31	17	4.83	26
11/10/2008 7:10	16	28	14	9.27	30
11/10/2008 7:15	13	33	16	7.02	31
11/10/2008 7:20	17	27	17	10.83	21
11/10/2008 7:25	15	27	23	6.02	31

$$X_p = (\langle x_1, x_2, x_3, \dots, x_n \rangle, \hat{x}) \tag{4.1}$$

where: *(p)* is the time interval in which the sample parameters were observed, $\langle x_1, x_2, x_3, \dots, x_n \rangle$ is the traffic parameters vector aggregated at that time interval (input features) and $(\hat{x})$ is the corresponding average travel time at the same interval (output/response feature).

The objective of this study is to develop a model to estimate $(\hat{x})$ from the set of input parameters. The ultimate goal is to describe a relation (f) (Eq. 4.2) that can be generalized over other unseen samples, where the $(\hat{x})$ is unknown.

$$\hat{x} = f(\langle x_1, x_2, x_3, \dots, x_n \rangle) \tag{4.2}$$

4.1.1 Problem Characteristics

Understanding the problem is crucial for deciding on the techniques that will be used to solve it. In general, the characteristics of any problem can be extracted from analyzing its input and its expected output.

Based on the input dataset, the problem in this study can be described as a supervised learning problem. Supervised learning is a method for learning a function by example, which means the model learns a certain relation between set of inputs and their corresponding outputs by “looking at” some observed examples that relate those variables. Those examples combined are known as the *training dataset*. The goal is to generalize from the training data to unseen situations in acceptable way.

Another important property of the problem can be extracted from the type of output. In general, learning problems can be classified, based on the output of the learning model, into classification and regression problems [28]. In classification problems, the output of the problem is the class or the category the response variable belongs to, so, the system relation can be described as a discrete function. While in regression problems, the output of is the actual value of the response variable, so, the system relation can be described as a continuous function.

In this study, since a training dataset observed from the field is used to estimate travel time values, the problem can be described as a supervised learning, regression problem, and so the methods selected to solve this problem must be based on that.

4.1.2 Error Measurements

Different error measures are usually applied in supervised learning problems to assess the quality of the different learning methods. In general, the measures quantify the deviation of the estimated values from their actual values.

Error measures can be used to compare the performance of different methods or to assess the performance of a single method by comparing the different outputs of multiple runs of the same method, under different settings. For the scope of this research, three main error measures will be used: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean absolute percentage error (MAPE).

Assuming that, in a dataset of N samples, for each sample (*i*), (X_i) is the observed (Ground-Truth) value and (Y_i)is the estimated value then:

1 – Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (X_i - Y_i)^2} \quad (4.3)$$

The root mean square error is used by most neural networks models to assess the performance of the network and deciding when to stop training.

2- Absolute Error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |X_i - Y_i| \quad (4.4)$$

The MAE has the advantage of keeping the unit of the measure. In this study, where travel time is being studied, MAE is expressed in term of seconds.

3- Mean absolute percentage error (MAPE):

$$MAPE = 100\% \times \frac{1}{N} \sum_{i=1}^{N} \frac{|X_i - Y_i|}{X_i} \quad (4.5)$$

In the MAPE the absolute difference between the actual and the estimated value is divided by the actual value, and so, expresses the MAE in a generic percentage way.

To best evaluate the different methods, all methods will be tested using the same test-set which reflects a real-life pattern and contains previously unseen samples. The data preprocessing analysis in this study showed that Friday has a different traffic pattern from the rest of the week (Figure 32), so, the data from Monday to Thursday will be used as the training dataset for the different models, while the data from Friday will be left out for testing. To make sure the test-set contains only previously unseen data, test samples that already exist on training dataset was removed.

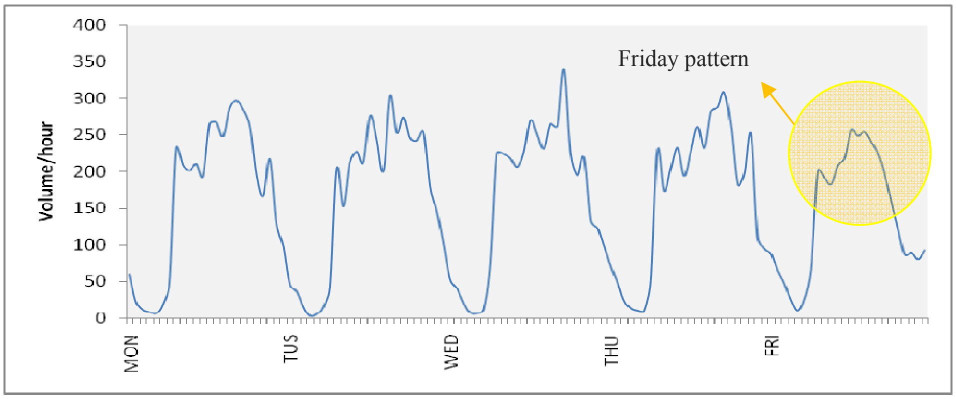

Figure 32 Traffic Patterns Through The Week (Monday - Friday)

The test dataset contains 100 5-minutes intervals which represent the travel time data for nearly 8 hours on Friday from 7:00 am to 3:00 pm. Figure 33 shows travel time data over that period. This curve will be the ground-truth data reference for evaluating the different estimation methods.

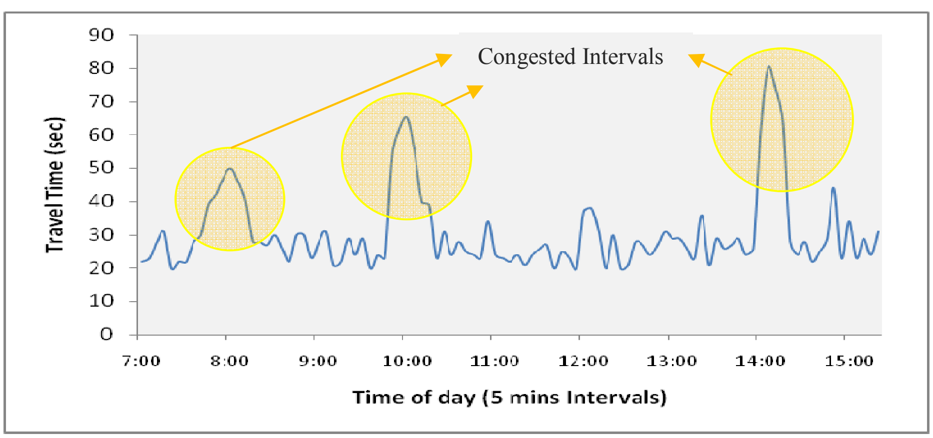

Figure 33 Travel Time Data on Friday (7:00 am – 3:00 pm)

The above chart also shows the congested intervals at 8:00 AM, 10:00 AM and 2:00 PM. At these intervals, travel times jump dramatically which simulates emergency evacuation situations. The ability of the different estimation methods to detect these intervals will be assessed.

4.1.3 Dimension Reduction: Correlation analysis

A basic problem in machine learning is to identify the set of features that best represents the system, or in other words, the features that actually affect the output of the system. This process is called dimension reduction.

The main advantages of a dimension reduction are reducing the system complexity, and so, removing noise, improving model performance and increasing speed of learning and accuracy of prediction. In fact, when the system is represented with fewer features, it becomes easier to understand the process that underlies the data, which leads to more accurate knowledge extraction from the model, especially in black-box models such as ANN.

There are two main methods for reducing dimensionality of the system [29], feature selection and feature extraction. In feature selection, the goal is to select the (*k)* of the (*d*) dimensions that gives the most information about the system, while other dimensions (*d*-*k*) are discarded. In features extraction, a new set of (*k*) dimensions that are combinations of the original (d) dimensions are introduced.

From a statistical perspective, the main hypothesis for a representative feature set, is that it contains features that are highly correlated with the response variable, yet uncorrelated with each other. Based on this hypothesis, a correlation analysis was conducted on the feature dataset of this study. The analysis was conducted using Matlab Statistics Toolbox ™. The results are shown in Table 17 which represents the correlation array of the different system parameters.

Table 17 Correlation Analysis Array

	volume	speed	Length	occupancy	travel time
volume	1	-0.4447	-0.195	0.952	0.6038
speed	-0.444	1	0.418	-0.3105	-0.3482
Length	-0.195	0.418	1	0.0941	-0.1469
occupancy	0.952	-0.3105	0.0941	1	0.5488
travel time	0.6038	-0.3482	-0.1469	0.5488	1

Table 17 shows the high correlation between occupancy and volume. This correlation is reflected in the direct relation between the two parameters in the dataset (Figure 20). From this analysis we assume that using one of them is enough to model their effect on the system output. The Table also shows the different effects of the volume, speed occupancy and length on the travel time. It is clear that volume has the highest positive effect, followed by occupancy, followed by speed (which has a negative effect on the travel time) and finally, the average vehicle length. The explanation for the little effect of the vehicle length is that, in environments such as the campus road network, where people usually drive small vehicles, the variance in the vehicles length during the day is relatively low in comparison to the variation of the volume and the speed. Table 18 shows the Standard deviation of the different parameters.

Table 18 Standard Deviation and Mean Value for Different Input Features

	Mean	Std
volume	18.77111	6.77
occupancy	4.75	3.2
speed	23.66667	4.30
Length	15.72667	1.8

4.2 Travel time Estimation

This section introduces the different methods used to solve the problem in this study. Each method's settings, design, input and output are described in detail. All methods will be tested using Friday's dataset. RMSE, MAE and MAPE are the standard error measures for inter & intra-methods evaluation.

4.2.1 Average vs. Actual

This method assumes that travel time at any time interval and under any traffic conditions is always the same as the mean travel time. This simplification gives a base reference to assess and compare other methods. Performance based on this simplification will be used as a threshold to decide whether the evaluated method is "worthwhile" or not. Also, the average itself will be used to identify the congested intervals. Any interval with travel time more than the average travel time is considered a congested interval.

Table **19** shows the error measures values if we always use average travel time. The test was applied on Friday's data.

Table 19   Error Measures for The Average

RMSE	MAE	MAPE%
11.90	8.0	24%

Figure 34 shows where the average line stands from the actual Ground-truth values. The results show that the average has nearly 8 seconds MAPE and 24% deviation from the actual values.

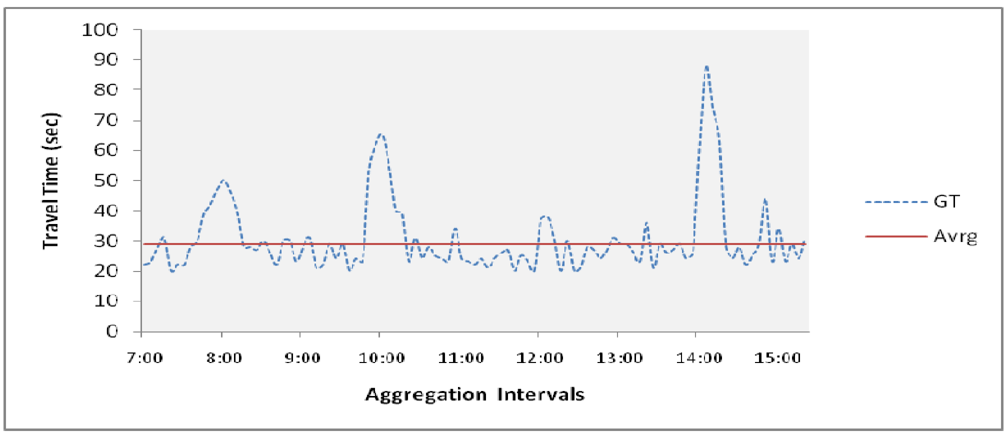

Figure 34   Average vs. Actual

4.2.2   Trajectory Methods

As implied in section 2.2.2.2, instantaneous or trajectory methods are very popular methods for estimating travel time on highways. In this section, several trajectories will be applied to solve the system. The objective is to assess highway trajectories' ability to estimate travel time on arterial links of congested urban networks.

Several trajectory methods have been introduced in the literature. The following are some popular methods.

Assuming that:
TT: is the travel time
L: is the length of the link
$V_{A,B}$: is the point speed at the upstream/downstream stations

Then:

1 – *Half Distance Method*: This is the most widely-used trajectory for estimating travel time on highways. It assumes that the point speed at the upstream station applies to half of the link and the point speed at the downstream station applies to the other half (Eq. 4.6).

$$TT = \frac{1}{2}\left\{\frac{L}{V_A} + \frac{L}{V_B}\right\} \tag{4.6}$$

2- *Average speed Method*: The Average speed method assumes that the average of the downstream and the upstream speeds applies to the link (Eq. 4.7).

$$TT = \frac{L}{(V_A + V_B)/2} \tag{4.7}$$

3- *Minimum speed*: In this method the minimum speed between the upstream and the downstream stations is applied to the link (point speed is generalized over the link).

$$TT = \frac{L}{MIN(V_A, V_B)} \tag{4.8}$$

The analysis was conducted on Friday's data and the results in term of the different error measures are shown in Table 20.

Table 20 Error Measures of The Trajectories

Method		Error	
	RMSE	MAE	MAPE%
Avrg. speed	13.92	8.86	24
Half-Distance	13.84	8.73	23
Min. speed	13.42	8.34	22

The results show that the minimum speed method slightly outperforms the other two methods. The results also show that trajectory methods could not beat the average estimate, which outperformed all of them with nearly 0.5 second absolute error.

This implies that the highway trajectory methods were not successful in estimating travel time on short arterial links of congested networks. The difference in performance for trajectory methods between highways and arterials can be explained as follows: on sections of highway, the traffic is rarely interrupted under ordinary situations and people tend to drive at nearly a constant speed, which is usually the speed limit of the highway, and so, the point speed at the upstream and downstream stations of the highway section can be generalized over the section. In other words, the relation between TMS and SMS are nearly linear [59][60].

On arterials, traffic has an interrupted nature due to the congestions which result from traffic signals, the high pedestrians' density and other factors that result in dramatic speed changes dramatically over the link, introducing high nonlinearity in the relation between the SMS and the TMS, and so, the point speed fails to represent the section speed. The performance of the Minimum speed trajectory is shown in Figure 35.

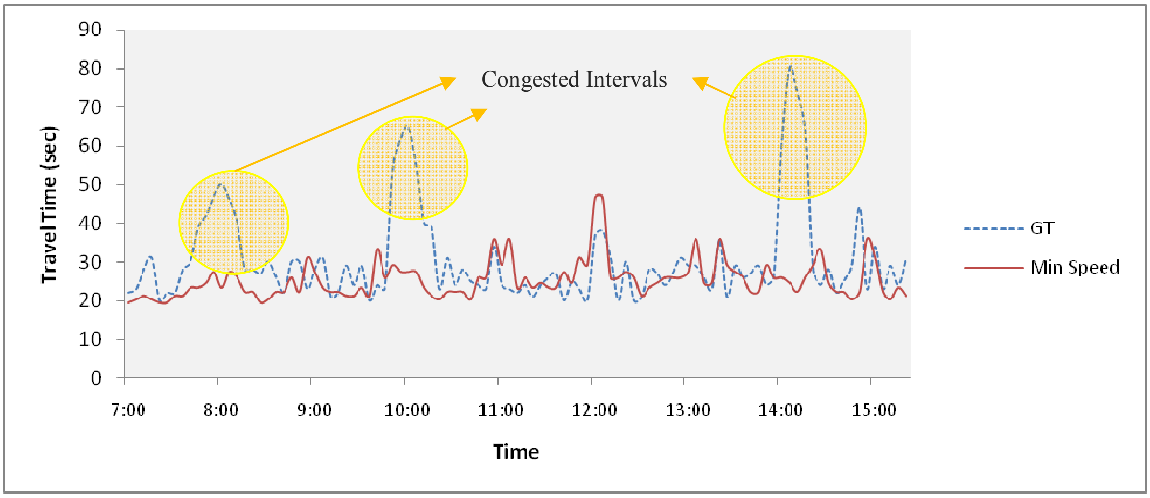

Figure 35 Minimum-Speed Trajectory Estimation vs. Ground-Truth Data

Figure 4 shows that trajectories were, somehow, able to estimate travel time under usual free-congestion conditions (travel time 20 – 30 second) but failed badly during time intervals with congestions (Table 21).

Table 21 Minimum-speed Trajectory Performance Under Congestions

RMSE	MAE	MAPE %
22.29	17.34	35

This analysis leads to the conclusion that the point speed alone is not enough to estimate travel time on arterial links. Other traffic parameters should be introduced to help capturing more of the system characteristics, and so, generate more accurate estimates. Next section investigates the linear effect of other traffic parameters on travel time.

4.2.3 Multiple Linear Regression

As mentioned in Section (2.2.2.3), regression analysis is a statistical technique that tries to explain the linear relation between a set of independent input variables and a dependent output/response variable. The ultimate goal is to derive the regression equation which describes the output variable as a linear function of the input variables. If we assume that $X_i = (\langle x_1, x_2, x_3, \dots, x_n \rangle, y_i)$ represents system input/output vector, where $\langle x_1, x_2, x_3, \dots, x_n \rangle$ is the input features and (y_i) is the output feature, then the linear regression relation can be described as Eq. 4.9.

$$y_i = f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_i x_i \tag{4.9}$$

Different regression methods, such as linear regression, multiple linear regression (MLR), regression trees, and locally-weighted regression have been studied intensively for the travel time estimation problems. In this thesis, where the relationship between multiple traffic parameters and travel time are being investigated, multiple linear regression analysis is used.

MLR is considered another supervised learning method for modeling the relationship between multiple input variables and an output (response) variable by fitting a linear equation based on the observed data. The linear relationship is usually evaluated by the least square estimator, which is aimed at minimizing the sum of squared deviations of the actual observed values of the response variable from those estimated by the model. Mathematically, the least square estimator (LSE) of a parameter is obtained by minimizing Eq. 4.9 with respect to the parameter:

$$\mathrm{LSE} = \sum[\mathrm{Yi} - \mathrm{y_i})]^2 \tag{4.10}$$

where (Y_i) is the actual value and $(\mathrm{y_i})$ is the estimated value using $f(x)$ which is some linear relation that relates the different input variables to the output (Eq. 4.10)

The goal from conducting this kind of analysis is to test the linearity of problem. If the system can be represented by a linear equation, then, there is no need to apply other complex methods such as fuzzy logic, ANN or genetic algorithms as these methods may create unnecessary overhead that affects the performance and the running time of the model.

To conduct the MLR analysis, the *regress* function provided in the Statistic Toolbox ™ of Matlab was used. The command *b = regress(y, X)* returns a p-by-1 vector b of coefficient estimates for a multi-linear regression of the responses in y on the predictors in X. X is an n-by-p matrix of p predictors at each of n observations. y is an n-by-1 vector of observed responses [61]. In this study, X contains the basic traffic parameters <volume, speed, length, occupancy> and y is the travel time Parameter <travel time>

The analysis was conducted twice, first with all the parameters, and then without the average length parameter. The goal was to validate the statistical analysis of dimensions reduction which suggested that the average length has a limited effect on the system. The analysis results are shown in Table 22.

Table 22 MLR Results

feature	Coefficients	
	f_1	f_2
volume	2.124	1.8343
speed	-0.2787	0.2657
Length	1.0992	0.0
occupancy	-0.0743	-0.0415

The analysis returned the coefficients that best fit the relation between the inputs and the output. The results can be represented as equations: f_1 and f_2 (Eq. 4.11 and Eq. 4.12)

$$f_1 = 2.124x_1 - .2787x_2 + 1.0992x_3 - .0743x_4 \tag{4.11}$$

$$f_2 = 1.8343x_1 + .2657x_2 - .0415x_4 \tag{4.12}$$

To evaluate the results, Equations f_1 and f_2 were applied to the test dataset (Friday data) used in evaluating the average and the trajectory methods. Table 23 shows the evaluation results represented by the different error measurements.

Table 23 Error Measures of The MRL Method

Relation		Error	
	RMSE	**MAE**	**MAPE**
f_1	8.95	6.34	0.21
f_2	8.82	6.27	0.20

The results show that MLR outperforms both the average and the trajectories methods by about 2 seconds which implies that the introduction of other traffic parameters beside speed captures more of the system features and so improves the estimates. The results also confirm the dimension reduction analysis the average length has a very limited effect on performance which is reflected in the different error measurements.

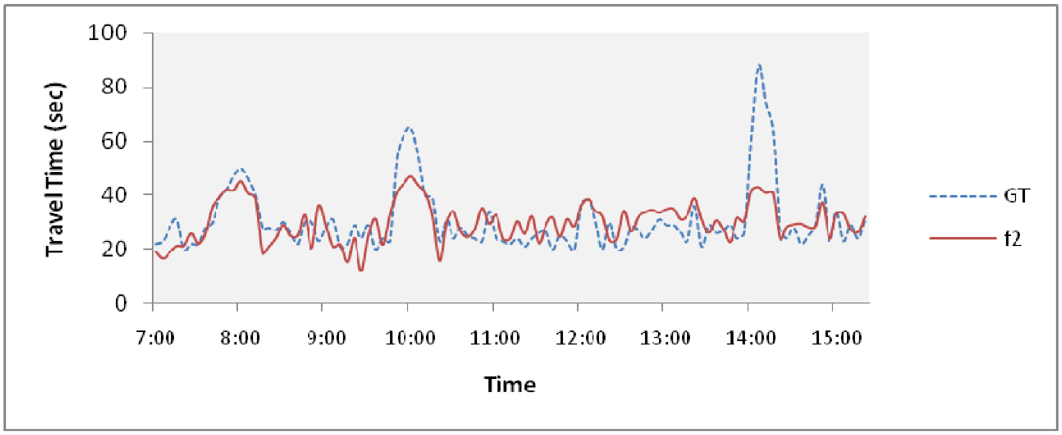

Figure 36   MLR Estimation vs. Ground Truth Data

Figure 36 shows that although the multiple linear analysis was more successful in estimating travel time, it still suffers at congested periods (Table 24), which means the system has a more complicated non-linear relationships that MLR cannot detect. To overcome this, a nonlinear technique (ANN) is used to capture the nonlinearity of the system.

Table 24   MLR Performance Under Congestions

RMSE	MAE	MAPE %
12.46	8.71	18

4.2.4 Artificial Neural Networks

Artificial neural networks are non-linear mathematical data modeling tools that emulate biological neural networks. An ANN can perform highly complex mappings of nonlinearly related data by detecting subtle relationships among different input and output parameters of a system. The main goal is to develop an understanding of a certain relation, based on a limited quantity of training data, which can be then generalized on other datasets.

Neural networks outperformed other data-driven methods for travel time estimation, because they are able to model spatial and temporal travel time information that results from the nonlinear nature of traffic systems.

The steps for employing a neural network to solve or model a certain problem are shown in Figure 37.

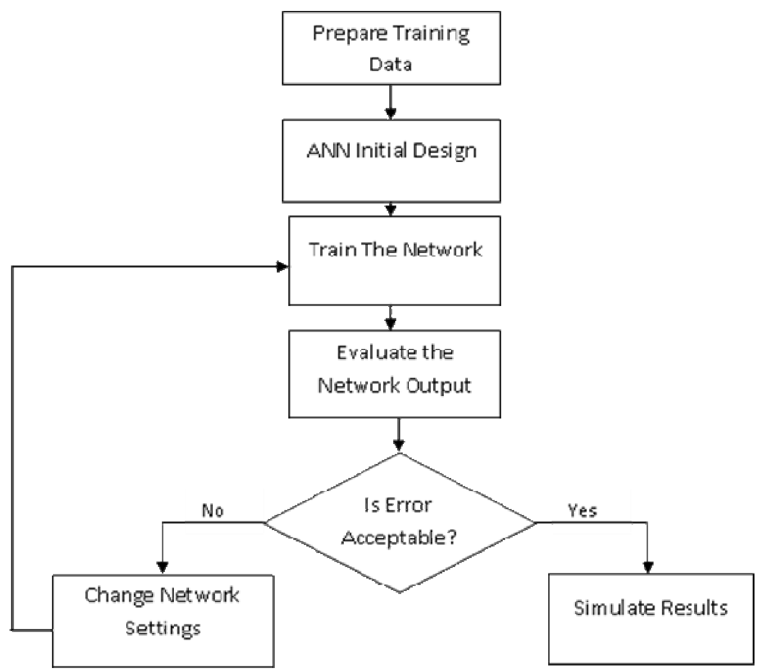

Figure 37 Steps for Solving a Problem Using Neural Network

Steps in the above Figure can be summarized as follows:

1- Prepare the training data:
This step includes collecting, verifying and cleaning the data. Data preparation is a domain-related process and should be based on a careful analysis and

understanding of the problem space. This step has already been conducted in this study (Chapter 3).

2- Design the network:
This step includes deciding on the initial network settings, the ANN Model to be used, the trainings algorithm, the number of hidden layers and the number of neurons in each layer. The decisions in this step depend on the complexity of problem, the problem settings (supervised or unsupervised, regression or classification) and previous experience in the domain.

3- Train the network:
Training the network can be done using any third party software that provides an ANN package, such as, WEKA or Matlab Neural Network Toolbox ^{TM}.

4- Evaluate the network:
Performance is assessed using the different error measures. If the results are not "good enough", several steps can be taken to improve them, such as, retraining using the same settings, changing the number of hidden neurons, changing the training algorithm or using larger training datasets.

5- Simulate the network response to new inputs:
After the ANN is trained and evaluated, it can be used to simulate new inputs.

Several neural networks models have been investigated in transportation research (see section 2.2.2.4). But in general, feed-forward neural network (FFNN), trained with the back-propagation (BP) learning algorithm, is the most commonly used. In this study, the feasibility of using FFNN-BP to estimate travel time on short links is assessed.

4.2.4.1 Multi-Layer Feed-forward Neural networks

Multi-Layer feed-forward neural networks are a very popular supervised learning technique. In this model, the data only flows in one direction, from input to intermediate/hidden to output layers of the network, with no cross-connections between units in the same layer or backward (recurrent) connections from layers downstream.

A multilayer FFNN has at least three layers, input, hidden and output. In a three-layer network, the inputs are fed to the network through the input layer, which basically fans-out the input values to the first hidden layer with no significant processing. Usually, the inputs are normalized to fit in the interval [-1, 1].

Each input from the input layer is fed to each neuron in the hidden layer, the inputs are multiplied by a certain value called a "weight" (w_{ik}) and usually biased by another value(θ_k). The weights and biased values are added together to produce a value (u_k) which is fed to the transfer function of the neuron(f), known as the Activation Function, which performs a certain transformation on the data before distributing it to the next layer, which is the output layer in a three-layer network. The processing of a Neuron in the neural network is described in Figure. Eq. 4.14 describes the process mathematically.

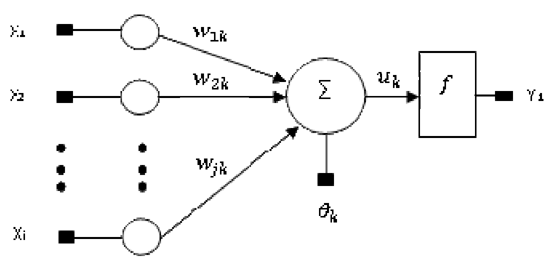

Figure 38 Neuron Architecture

$$Y_k = f\left(\sum_{j=0}^{n} w_{jk} \cdot z_{j(k-1)} + \theta_{ik}\right) \tag{4.14}$$

In general, there are three types of activation functions:

1- *Threshold*: returns 0 if the input is less than a specified threshold value (v) and 1 otherwise.

$$F(x) = \begin{cases} 0 & x < v \\ 1 & x \geq v \end{cases} \quad (4.15)$$

2- *Sigmoid*: Smooth and monotonically increasing function, can range from [0, 1] which is known as the logistic function (Eq. 4.16) or from [-1,1] which is known as the Hyperbolic tangent (Eq. 4.17)

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \quad (4.16)$$

$$tanh\left(\frac{x}{2}\right) = \frac{1-e^{-x}}{1+e^{-x}} \quad (4.17)$$

3- *Piecewise-Linear*: an approximation of a sigmoid function (Eq. 4.18).

$$F(x) = \begin{cases} 0 & x < -\frac{1}{2} \\ v & -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 1 & x > \frac{1}{2} \end{cases} \quad (4.18)$$

Figure 39 shows the different function shapes.

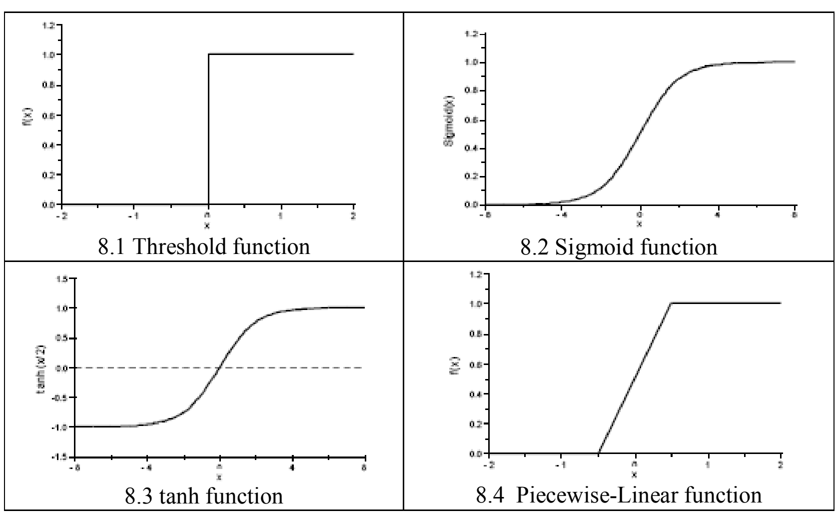

Figure 39 Different Activation Functions

The universal approximation theorem for neural networks says that a multi-layer neural network with one hidden layer and sigmoidal activation functions can approximate any continuous function [62].

When designing a neural network, some critical design decisions must be made to ensure the optimum performance of the network. Usually, in feed-forward neural networks, three main decisions must be addressed: the training algorithm to be used, number of hidden layers, and the number of neurons in each layer.

4.2.4.2 The Training Algorithm

The intelligence of the neural network lays in the values of the weights between neurons. Several methods have been introduced to adjust those weights to maximize the performance of ANNs. Such methods are called Learning algorithms. A Learning algorithm can be defined as a set of rules that controls the process of adjusting the network weights, thus allowing it to learn how to respond to specified training conditions with acceptable error rate. The operation of training algorithms is explained in Figure 40.

Although several algorithms are available to train multi-layer feed forward neural networks with sigmoidal activation functions, backpropagation, since first introduced in 1974 by Paul Werbos [63], has always been the most common algorithm due to its ability to approximate nearly any nonlinear function.

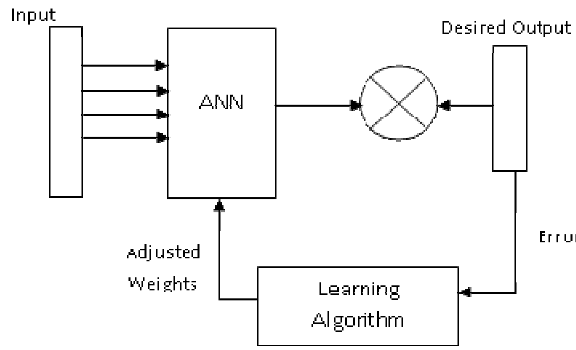

Figure 40 Learning Algorithm Rule in Neural Networks Architecture

The basic idea behind BP training is that the error of the output layer nodes is back-propagated through the networks to determine the errors of the nodes in the hidden layers. After applying the input values from the training dataset, the output of the network is compared with the actual/observed output values to determine the value of a predefined error function. The error is then fed back through the network layers. Using this information, the BP algorithm adjusts the weights of each connection to reduce the value of the error function by some small amount. Repeating this procedure for sufficiently large number of cycles causes the network to converge eventually to a state where the error in the output values is acceptable, which means that the network has learned the target pattern. This iterative process is called learning with backpropagation.

Mathematically, backpropagation works as a generalization of the delta rule for non-linear activation functions. The operation of the backpropagation algorithm can be derived as follow:

The error in the network can be described using the sum error square function as:

$$E(w_{ij}) = \frac{1}{2}\sum_{i=1}^{N}(t_i - x_i)^2 \tag{4.19}$$

where t_i is the desired output and x_i is the actual output of the network layer i under w_{ij} weight value. The goal of training process is to minimize the error function value. To find a local minimum of a function using gradient descent, steps proportional to the negative of the gradient are taken, and since the error in the network changes with respect to the weight, the best direction to move in to find a local minimum is the weight direction, which means, the weight should changed to be proportional to the negative of the derivative of the error function with respect to the weight (Eq. 4.20).

$$\Delta w = -\eta \frac{\partial E(w)}{\partial W} \tag{4.20}$$

In Eq. 4.20 (–η) is referred to as the learning rate. It usually takes value between [0, 1]. To find a local minimum of the error we take the partial derivative of Eq. 4.21 with respect to the weight.

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial\left[\frac{1}{2}(t_j - x_j)^2\right]}{\partial w_{ji}} \tag{4.21}$$

The above derivative (Eq. 4.22) can be decomposed using the chain rule to Eq. 4.22

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial\left[\frac{1}{2}(t_j - x_j)^2\right]}{\partial x_j}\frac{\partial x_j}{\partial w_{jis}} \tag{4.22}$$

The first derivative is easy to find (Eq. 4.23)

$$-(t_s - x_s) \tag{4.23}$$

The second derivative can also be decomposed using the chain rule based on the fact that the activation of unit j is a function of the input to the unit, (uj), which is in turn a function of the weights into the unit (Eq. 4.24).

$$\frac{\partial x_j}{\partial w_{ji}} = \left(\frac{\partial x_j}{\partial u_j}\right)\left(\frac{\partial u_j}{\partial w_{ji}}\right) \tag{4.24}$$

Assuming the activation function now is unknown, the first derivative on the right-hand side of Eq. 4.25 can be represented as:

$$\grave{f}(u_i) \tag{4.25}$$

The second derivative on the right-hand side of Eq. 4.24 can be derived as follows based on Eq. 4.26

$$\frac{\partial u_i}{\partial w_{ji}} = \frac{\partial(\sum x_k w_{jk} + \theta_{ik})}{\partial w_{ji}} = x_i \tag{4.26}$$

Putting all the parts together results in Eq. 4.27

$$\frac{\partial E}{\partial w_s} = -(t_s - x_s)\grave{f}(u_i)\, x_i \tag{4.27}$$

Rewriting the above equation

$$\Delta w_{ji} = \eta\left(t_j - x_j\right)\grave{f}(u_i)\, x_i \tag{4.28}$$

Eq. 4.28 is the equation for updating each single weight.

To derive the rule for the output layer the above equation is rewritten as Eq. 4.29

$$\Delta w_s = \eta y_i \delta_j \tag{4.29}$$

where:

$$\delta_j = \left(t_j - x_j\right)\grave{f}\left(u_j\right) \tag{4.30}$$

Now since the activation function is sigmoid function, which can be derived easily, the above equation can be rewritten as Eq. 4.31 which represents the *deltas for the output layer*

$$\delta_j = x_j(1 - x_j)(t_j - x_j) \tag{4.31}$$

Following a similar analysis for the delta rule it can be proved that that for *neuron q in hidden layer p*, delta is:

$$\delta_p(q) = x_p(q)[1 - x_p(q)] \sum w_{p+1}(q,i)\delta_{p+1}(i) \tag{4.32}$$

Each delta value for hidden layers requires that the delta value for the layer after it be calculated, which is clear in the term $\delta_{p+1}(i)$ of Eq. 2.32. This means the error from the output layer is slowly propagated backwards through the network.

After deciding on the network model and the training algorithm, two more important decisions about the internal network architecture must be made carefully: the number of hidden layers and the number of neurons in each hidden layer.

4.2.4.3 Number of hidden layers

In linear and generalized linear models where a linear relation between the input and the output parameters can be derived, no hidden layers are required. Actually, in such systems, using a linear regression model to approximate the system linear function is more convenient. On the other hand, in complex nonlinear systems, where no clear linear relation can be approximated to represent the system; hidden layers play a central role in expanding the space of hypotheses that the network can represent enhancing the training process.

There is no theory yet to determine the number of hidden layers that are needed to learn a given function based on the properties of the system, but it has been shown mathematically and by experience, that one hidden layer, with sufficient number of nodes, is nearly capable of approximating any continuous function from any non-linear relation [64]. Two layers are usually used to approximate discontinuous functions, but using two layers or more rarely enhances performance and creates unnecessary overhead that affects the prediction accuracy and the running time of the model.

4.2.4.4 Number of hidden neurons:

Determining the number of neurons in the hidden layers is a very important decision in deciding on the overall neural network architecture. The number of neurons has a tremendous influence on the performance of the network, and so a very careful decision must be made to ensure acceptable results.

Using too many neurons in the hidden layers can result in several problems such as network overfitting or information memorizing, in addition to increasing the time and complexity of the training process. Overfitting occurs when the network learns too many specific samples due to the large number of neurons in the hidden layers, and essentially ends up memorizing the training data instead of capturing the desired pattern resulting in poor generalization. Overfitting can also happen when the network has a very small sample set to learn from, or when the network is over trained.

Multiple techniques are available to detect and process overfitting problems in neural networks. In general, the key concept of the solution is to bridge the gap between the system and the network complexity. Some popular methods are using sufficiently large training datasets [65] and early stopping [66].

On the other hand, using too few neurons can lead to an underfitting problem which occurs when the network is not complex or "smart" enough to learn a certain relation because the number of neurons in the hidden layers is inadequate to detect all the relationships between the system parameters, and so, results in a very high error rate.

Since there is no solid theory to determine the optimum number of neurons, some sort of compromise must be reached experimentally between too many and too few neurons. A role of thumb is to start with a relatively small number of hidden neurons and keep increasing until the output is acceptable.

Figure 41 shows underfitting, optimal performance, and overfitting of a neural network.

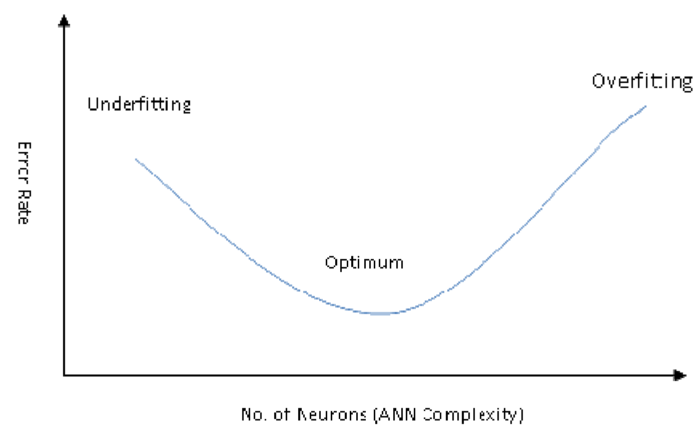

Figure 41 Overfitting, Underfitting and Optimum Performance of Neural Networks

4.2.4.5 Training the network

Matlab Neural Networks Toolbox ™ was used to build and run the neural network used in this thesis. The code for FFNN training in Matlab is shown in Figure 42. Before the actual training started, the inputs and the targets were normalized to fit within the range [-1, 1]. This is commonly done in ANNs to speed up the training process [67]. The initial data was divided into three subsets as follows: 75% for training, 15% for validation and 15% for testing. The training subset is used to train and build the system. The validation dataset is used to “tune” the model, in case there are some training parameters that the training dataset failed to detect. Finally, the testing dataset is used to judge the “quality” of the model by testing it using unseen data.

As mentioned earlier, the training process is repeated for several iterations, through which the error propagates back through the network allowing it to adjust the parameter weights and minimizing the error function.

Cross validation was used to prevent overfitting by stopping training at the right iteration. There are several techniques for implementing cross validation, test-set/ holdout method, K-fold cross validation method and leave-one-out method.

The test-set method is the simplest kind of cross validation. Validation samples are chosen randomly from the initial dataset to form the validation dataset and the remaining samples are retained as the training data. This technique is usually used when the initial dataset is large enough, and so, holding out a certain percentage for validation does not affect the training process performance.

```
function [net,ps,ts] = fitwithnet(p,t)

rand('seed',6.67426666E8)

% Normalize Inputs and Targets
[normInput,ps] = mapminmax(p);
[normTarget,ts] = mapminmax(t);

% Create Network
numInputs = size(p,1);
numHiddenNeurons = 7;  % Adjust as desired
numOutputs = size(t,1);
net = newff(minmax(normInput),[numHiddenNeurons,numOutputs]);

% Divide up Samples
testPercent = 0.20;  % Adjust as desired
validatePercent = 0.20;  % Adust as desired
[trainSamples,validateSamples,testSamples] =
dividevec(normInput,normTarget,testPercent,validatePercent);

% Train Network
[net,tr] =
train(net,trainSamples.P,trainSamples.T,[],[],validateSamples,testSampl
es);

 % Simulate Network
[normTrainOutput,Pf,Af,E,trainPerf] =
sim(net,trainSamples.P,[],[],trainSamples.T);
[normValidateOutput,Pf,Af,E,validatePerf] =
sim(net,validateSamples.P,[],[],validateSamples.T);
[normTestOutput,Pf,Af,E,testPerf] =
sim(net,testSamples.P,[],[],testSamples.T);

% Reverse Normalize Outputs
trainOutput = mapminmax('reverse',normTrainOutput,ts);
validateOutput = mapminmax('reverse',normValidateOutput,ts);
testOutput = mapminmax('reverse',normTestOutput,ts);
```

Figure 42 Neural Network design and training in Matlab

In K-fold cross validation, the initial data set is divided into k subsets. Each iteration, one subset is kept for validation and k-1 subsets are used for training. The validation process is repeated k times until each single k is used once in validation.

In leave-one-alone cross validation, only one sample from the initial dataset is left out for validation and the rest of the data is used for training. The training process is repeated until each sample in the initial data set is used once in validation. This method is actually a k-fold method with k equal number of samples. This technique is used when the initial dataset is small and maximal use of the initial data is required. In this study, where there is a relatively large dataset, the test-set validation was used.

Training stops when the final mean-square error is acceptable and the test-set error and the validation set error have similar characteristics. A plot of the training errors, validation errors, and test errors is shown in Figure 43. The best validation performance of the network in this study occurred at iteration 8, and the network at this iteration is returned.

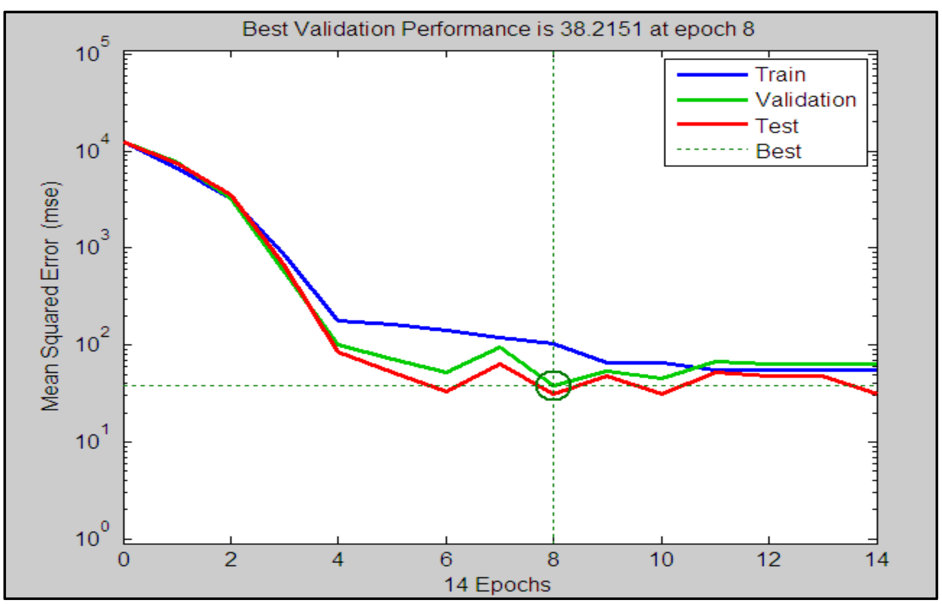

Figure 43 Training Curve

The training started with four input features, <volume, speed, length, occupancy>, one neuron in the hidden layer, and one output parameter, <travel time>. The numbers then increased until reaching the optimum results at 5 Neurons. Figure 44 shows the error measurers for different number of neurons in the hidden layer. The final architecture of the network is shown in Figure 45.

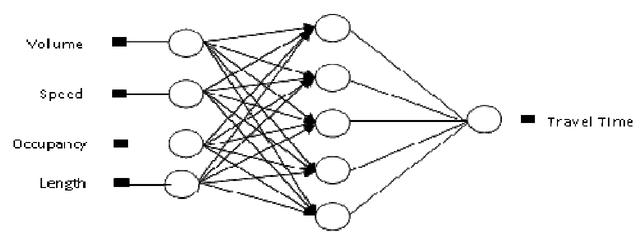

Figure 44 Travel Time Artificial Neural Network's Final Architecture

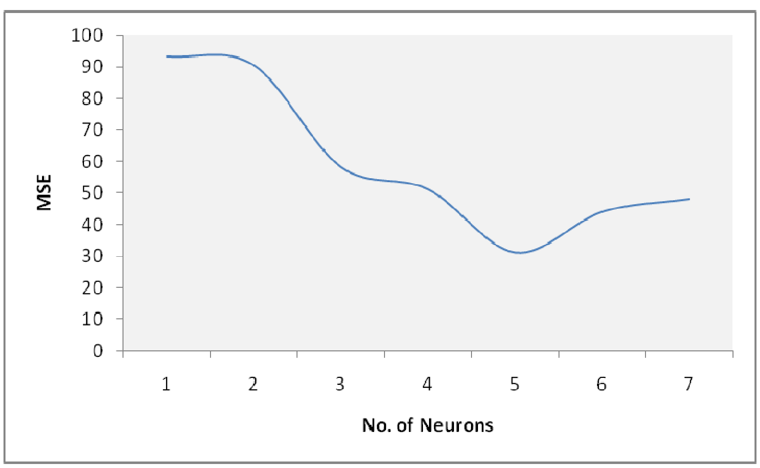

Figure 45 MSE vs. Number of Neurons in The Hidden Layer

After the network was built and trained, it was simulated on Friday's data. As mentioned above, the network was trained with normalized data so that the results fall into the range [-1, 1]. So, before simulating the network on a new dataset, the data should be normalized using the same scale that was used to normalize the training data. This can be done through the *mapminmax* function provided in the Matlab Neural Network Toolbox ™. After feeding the normalized data to the network using the *sim* function, the results should be un-normalized to scale the data back to its actual values. The un-normalization can also be done using the same *mapminmax* function but with different parameters.

Figure 46 shows the simulation code in Matlab.

```
[tn,ts] = mapminmax(t);
[pn,ps] = mapminmax(p);
pnew = mapminmax('apply',pt,ps); //scale the data to [-1,1]
an = sim(net,pnew);
a = mapminmax('reverse',an,ts); // scale the data back to original
```

Figure 46 ANN Simulation Code

The results of simulating the data using Friday's data are shown in Figure 47 and Table 10.

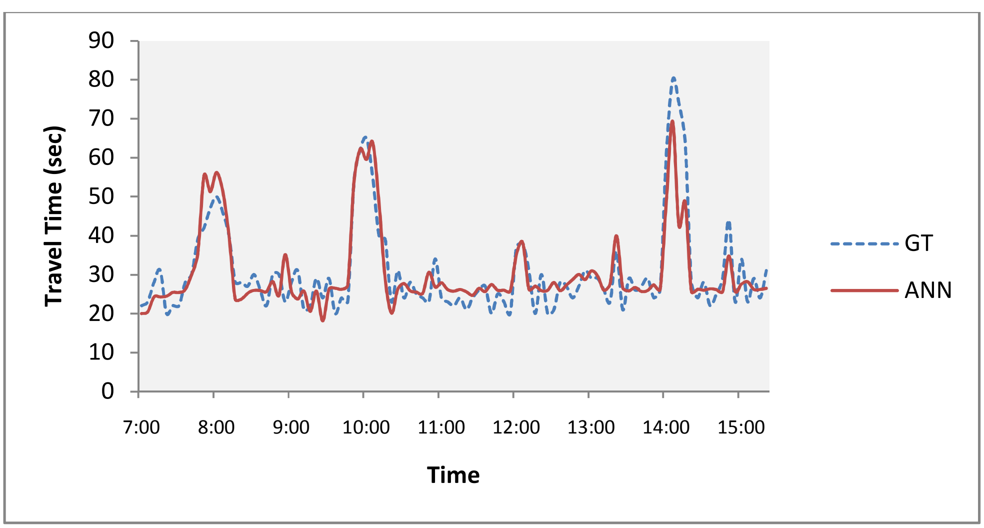

Figure 47 ANN Estimation vs. Actual data

Table 25 ANN Performance

RMSE	MAE	MAPE
6.43	4.3	14%

The results of the above analysis show that the neural network outperformed all the previous methods by a good margin, which supports the previous suggestion that the relation between traffic parameters and travel time on the arterial links is nonlinear. The results also show that the neural network was able to successfully detect the congested intervals with high accuracy (Table 26).

Table 26 ANN Performance Under Congestions

RMSE	MAE	MAPE %
8.67	6.39	14

To test the dimension reduction, the network was also trained without the average length parameter. The results showed that the performance dropped by nearly 0.75 seconds absolute error. Even though average length is not highly correlated with travel time, having it as an input feature to the network still has a considerable positive effect on the performance.

The generalization in this particular network could be improved by introducing other traffic-related features that might have influence on the travel time, such as pedestrians' density, time of the day, incidents rate, etc.

The ANN model can also be used for other links by including link-related features such as the length of the link, number of cross walks, number of intersections, and so on.

Eventually, after all features are taken into consideration, only one neural network trained with sample links data will be needed to estimate travel time on the different links of the network.

4.2.5 K Nearest Neighbor Method

K nearest neighbor (K-NN) is a supervised learning, statistical method, in which the value of an object is estimated based on the values of its neighbors in the training space. It assumes that objects close in distance are potentially similar [68].

K-NN is a non parametric, supervised learning, regression model, in which the estimation function is constructed directly at runtime from the training data without having predefined assumptions that relate the different input parameters to the output parameter. However, this kind of behavior limits its ability to response to unexpected conditions. It also means a slow response time because of the large amount of calculations needed to search the k-neighbors. In addition, K-NN can be fooled easily by irrelevant attributes, so a careful pre-processing analysis must be done to remove such attributes.

The closeness between two objects in the dataset is measured using the Euclidean distance equation (Eq. 4.33) which is the root of square differences between coordinates of the objects. Mathematically, if we assume that X and Y are two multi-attributes objects in the training dataset where $X = (\langle x_1, x_2, x_3, \dots, x_n \rangle, \hat{x})$ and $Y = (\langle y_1, y_2, y_3, \dots, y_n \rangle, \hat{y})$ then the Euclidean distance between them can be represented as:

$$d(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 +, \dots, +(x_n - y_n)^2} \quad (4.33)$$

Once the input vector is fed to the model, the K-NN algorithm starts measuring the distance from sample vectors in the dataset to the input vector, keeping only the K nearest vectors in the input vector neighbors' dataset. After the nearest neighbors are specified, the output of the input vector is estimated to be the average of the outputs of its neighbors (Eq. 4.34).

$$\hat{x}_i = \frac{1}{K} \sum_{j=i}^{K} \hat{y}_j \qquad (4.34)$$

The accuracy of the K-NN model is directly related to the size and the quality of the training dataset, which should be determined based on a careful observation of the problem space. In real-life traffic systems, where the traffic flow follows certain patterns, K-NN has shown a good potential for travel time estimation [69] [70].

The goal of applying K-NN analysis in his study is to assess the ability of non-parametric historical data models to predict travel time for arterial links. A special C# tool was developed to implement the K-NN algorithm. The value of K was determined based on MAE. Table 27 and Figure 48 shows the MAE and the MAPE for the different K values.

Table 27 Error Values for Different Ks

K	Error		
	RMSE	MAE	MAPE
1	10.87	6.03	0.20
2	7.88	5.42	0.18
3	7.58	5.21	0.17
4	6.95	4.85	0.16
5	6.68	4.56	0.15
6	6.62	4.51	0.14
7	6.43	4.36	0.14
8	6.51	4.37	0.14
9	6.16	4.16	13
10	6.66	4.40	0.14
11	6.88	4.6	0.15
12	7.22	4.70	0.15

The Table shows that this method gave the best results at k=9. Figure 18 shows the minimum value at K=9

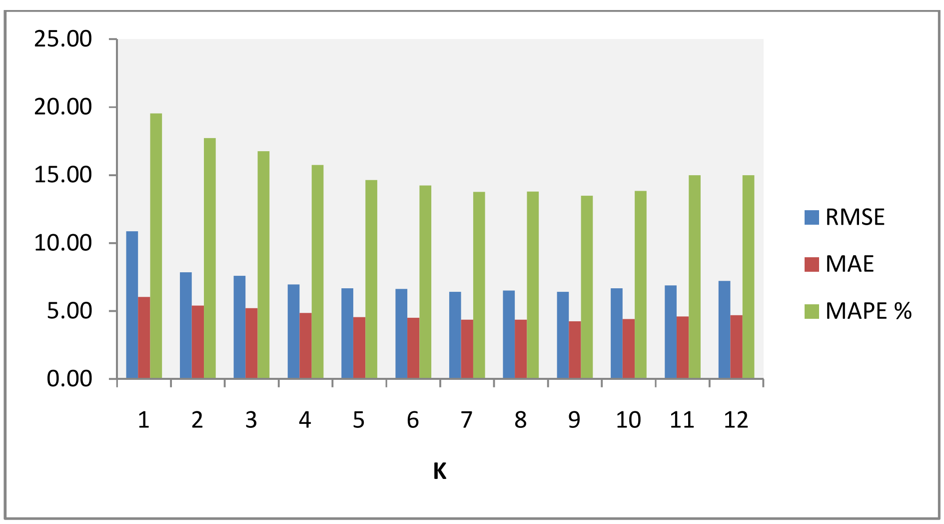

Figure 48 Error Measurements for Different K Values

At k =9, The K-NN method outperformed all previous methods. Figure 49 shows the 9-NN results versus the ground truth data.

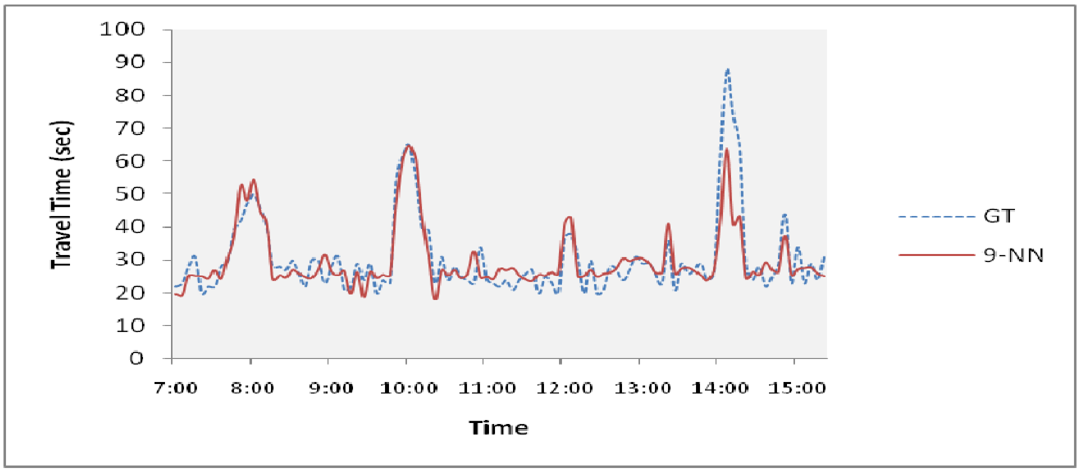

Figure 49 9-KK vs. Actual Data

The results of analysis show that the K-NN method achieves reasonable performance regardless of the scope of estimation (highways or arterials) for both free and congested intervals (Table 28). The results can be justified based on the fact the KNN is a non-parametric method. It depends only on similar historical situations to generate current estimates without any prior-assumptions.

Table 28 K-NN Performance Under Congestions

RMSE	MAE	MAPE %
9.40	6.50	14

The performance of K-NN can be improved by including more historical data in the training set keeping in mind the tradeoffs in term of increasing the search and response time. Performance can be also improved by using locally weighted K-NN in which features with more influence on the system are scaled to give more value to their closeness to the target.

The main drawback of K-NN is it works only on the observed link, which means each single link in the road network has to have its own historical database.

4.3 Conclusions

Different methods for travel time estimation were evaluated in this chapter. The chapter started by describing the problem characteristics and the error measurements (RMSE, MAE & MAPE) which were used to compare and assess the different methods. All estimation models were tested against Friday's data. The analysis started with measuring the different error rates that result from comparing the test data against the average travel time value. The goal in measuring the average value performance was to use it as a reference point for testing the practicality of other methods. The first travel time estimation methods investigated in this chapter were the highway trajectory methods. Three-known trajectories were used: half-distance, average speed and minimum speed. The results showed that the highway trajectories failed to beat the average threshold. This led to the conclusion that, even though such methods usually generate good results on highways, they do not have good performance on arterials. This failure can be explained by the fact that the assumption of stationary traffic conditions on highways does not hold for arterials which have interrupted traffic nature.

The second method evaluated was linear regression. The goal was to test the linearity of the problem. The *regress* function of the Matlab Statistics Toolbox ™ was used. The results showed that MLR outperformed the average threshold method by nearly a 2 seconds margin, but it failed to detect the congested periods that usually result from the non-linear behavior of the system. This analysis led to the conclusion that the problem has some linearity between its different parameters but modeling that linearity alone to estimate travel time is not enough to generate accurate results, especially at the intervals where the system is congested.

The next approach tried was neural networks. The goal in using neural networks was to model the nonlinear dynamics of the system that MLR could not detect. A multi layer feed forward neural network trained with backpropagation was employed. The Matlab Neural Networks Toolbox ™ was used to design, train and simulate the network. Results show that the FFNN outperformed the average threshold by nearly 4 seconds. Another interesting observation was that the neural network was much more successful in detecting the congested periods, confirming the assumption of the non-linearity of the system.

The final approach tried was the nonparametric K-nearest neighbor method. A locally developed C# tool was used to implement the K-NN algorithm. The tool starts with k=1 and keeps trying until reaching a minimum error rate. The best performance was encountered at k = 9. The results showed that K-NN is very competitive with ANNs in terms of accuracy, with a similar error rate. The performance of the different methods is shown in Table 29 and Figure 50.

From the above analysis, we conclude that among the different methods for estimating travel time on highways, non-linear models (ANN) and non parametric historical methods (K-NN) achieve the best performance on arterial links. ANN and KNN also showed a good performance in capturing congested intervals. Table 30 and Figure 51 show how the different methods performers under congestion. The congested analysis was performed on intervals that have travel time more than the average travel time (29 seconds).

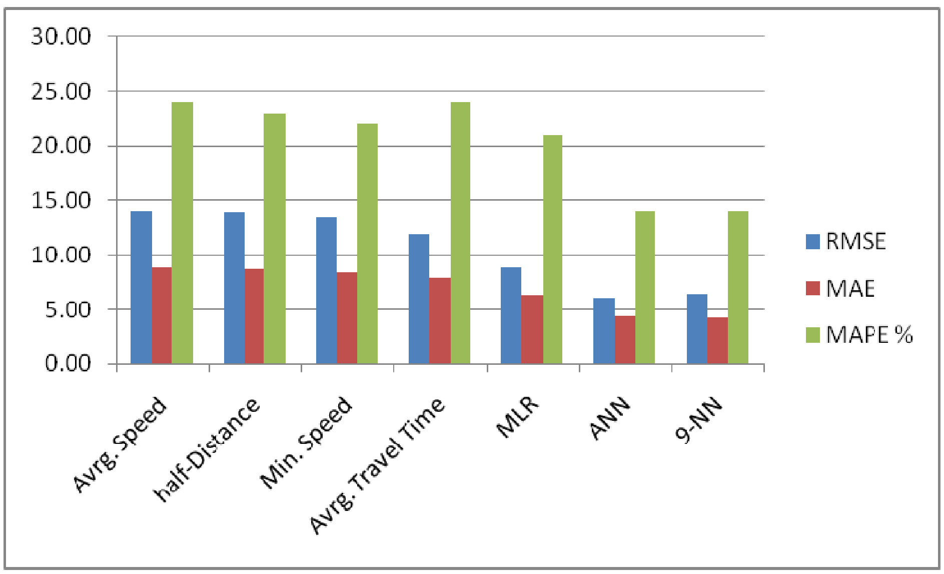

Figure 50 Performance of Different Methods

Table 29 Error Measures for Different Methods

Method	Error		
	RMSE	MAE	MAPE %
Avrg. speed	13.92	8.86	24
half-Distance	13.84	8.73	23
Min. speed	13.42	8.34	22
Avrg. travel time	11.90	7.90	24
MLR	8.83	6.27	21
ANN	6.03	4.3	14
9-NN	6.16	4.16	13

Table 30 Estimation Methods Performance Under Congested Intervals

Method	Error		
	RMSE	MAE	MAPE %
Trajectories	22.29	17.34	35
Average	19.64	13.94	26
MLR	12.46	8.71	18
9-KNN	9.40	6.50	14
ANN	8.67	6.39	14

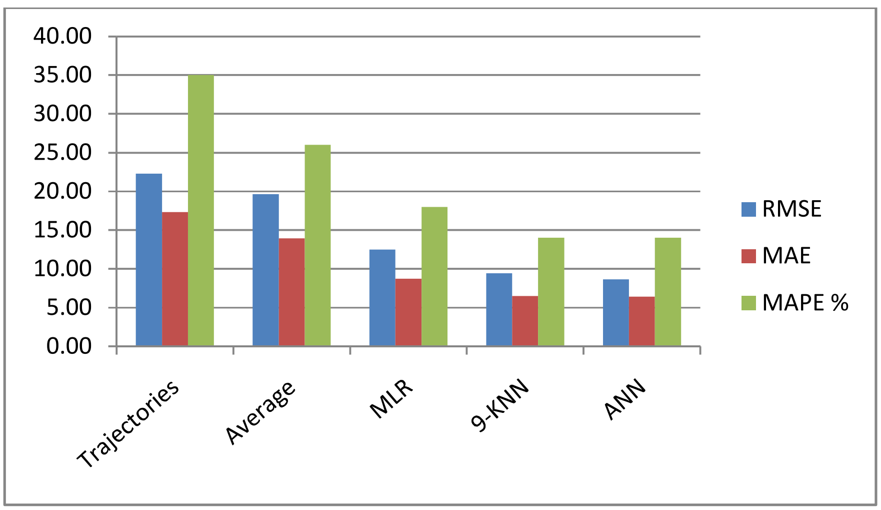

Figure 51 Estimation Methods Performance Under Congested Intervals

The results from ANN can be improved by including more features that have a greater effect on travel time. For K-NN, the results can be improved by increasing the size of the training set. From a practical perspective, the power of ANNs in generalizing from limited samples gives them a great advantage over the K-NN method. The ANN model can be expanded to work with different links. If the link-related features such as the length, number of crosswalks or traffic signals from sample links that represents different links-categories in the network are taken into consideration, it is a strong candidate for deployment in practical systems. K-NN works only for the observed links, which means each single link in the network has to have it is own historical database to approximate travel time.

4.4 Summary

This chapter described the data analysis phase of the thesis. The first section described the main problem characteristics and different error measures that were used to assess the different travel time estimation methods. Next, the chapter introduced four different methods for travel time estimates, including highway trajectory methods, multiple linear regression analysis, artificial neural networks and K nearest neighbor. The last section of the chapter presented an analysis and discussion of the results.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

A model for estimating travel time on short arterial links of congested urban networks, using the current available technology, was introduced in this thesis. The objective was to estimate Travel time, with acceptable level of accuracy that can be applied to real-life traffic problems, such as, congestion management and emergency evacuation.

To insure the quality of the analysis results, set of procedures and algorithms, based on the traffic flow theory and the field information, were introduced to validate and clean the data used to build, train and test the different models.

To achieve the research objective, various travel time estimation methods, including Highway Trajectories, Multiple Linear Regression (MLR), Artificial Neural networks (ANN) and K –Nearest Neighbor (K-NN) were modeled and tested using the same dataset. The results demonstrate that ANN and K-NN methods outperformed the linear methods by a good margin, also, showed a distinguished performance in detecting congested intervals.

From practical perspective, the power of ANNs in generalizing from limited samples gives them a great advantage over the K-NN method. The ANN model can be expanded to work with different links, if the link-related features such as the length, number of crosswalks or traffic signals from sample links that represents different links-categories in the network are taken into consideration, which makes it a strong candidate for deployment in practical systems. While K-NN works only for the observed links, which means each single link in the network has to have it is own historical database to approximate travel time.

5.2 Future Work

Future work will be focused on improving the generalization of the ANN. More traffic parameters that affect the travel time parameter will be introduced, such as, pedestrians' density, time of the day, incidents history, stop signs, work zone and so on.

The next step, after the ANN achieves the highest possible accuracy on the link, is to expand the model over other links on the road network. This can be done by carefully studying the road network, analyzing the traffic behavior on its different links and identifying the main categories of the links and the features of each category. Once all this information is available, it can be fed to the ANN to estimate travel time on the whole network.

Once the ANN is adjusted to work with the network, it can be integrated with other related systems. For example the emergency evacuation system, to help in minimizing the response time and maximizing the evacuation speed during emergency situations. The model can also be used in traffic management centers to help in optimizing the performance of the network, especially during congestions.

REFERENCES

[1] U.S. Department of Transportation (USDOT), Federal Highway Administration (FHWA), What We Have Learned from Intelligent Transportation Systems, (Washington, D.C. December 2000).

[2] Van Lint, J. W. C., Reliable Travel Time Prediction for Freeways, Ph.D Thesis, Submitted to Civil Engineering for Delft University of Technology, Netherland, 2004.

[3] Smith, B.L., Holt, R.B., and Park, B.B., “Travel time estimation for urban freeway performance measurement: understanding and improving upon the extrapolation method”, Transportation Research Board 83rd Annual Meeting, CD-ROM, Washington DC, 2004

[4] Schrader C. C., Kornhauser A. L., Friese L. M., “Using Historical Information in Forecasting Travel Times”, Transport Research Board 83rd Annual Conference, CD-ROM, Washington, D.C., Jan. 2004

[5] Bhaskar A., Chung E., Dumont A-G., Mouzon O. de., “Analytical Modelling and Sensitivity Analysis for Travel Time Estimation on Signalised Urban Network" Transportation Research Board 87th Annual Meeting, CD-ROM, Washington D.C., 2008.

[6] Singh, A.K. & Abu-Lebdeh, G., “State Space Neural Networks for Travel Time Predictions in Signalized Networks” Proceedings of the 86th Annual Transportation Research Board Meeting, CD-ROM, Washington D.C., 2007

[7] Ruimin L., Geoffrey R., Majid S., Modelling and Estimation of Travel Time Variability Transportation Research Board 87th Annual Meeting, CD-ROM, Washington D.C., 2008.

[8] Travel Time Data Collection Handbook, Report no. FHWA-PL-98-035, Texas Transportation Institute, College Station, Texas, 1998.

[9] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, Sei-Wan Chen. "Automatic license plate recognition", IEEE Transactions on Intelligent Transportation Systems, 2004, pp. 42- 53

[10] Coifman B, Beymer D, McLauchlan P, Malik J, "A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance", Transportation Research: Part C, 1998, pp. 271-288.

[11] C. Oh and S. G. Ritchie, "Real-Time inductive-signature-based level of service for signalized intersections", Transportation Research Record 1802, pp. 97-104, submitted to Washington, D.C., TRB, National Research Council, June 2002, Paper No. 02-3684

[12] K. Kreeger, R. McConnell, "Structural range image target matching for automated link Travel Time computation", Presented at the 3rd Annual World Congress on Intelligent Transportation Systems, Orlando, FL, Oct. 1996.

[13] Yokota T, Inoue T, Kobayashi Y, Takagi K, "Travel Time Measuring System Based on Platoon Matching: A Field Study", Proceedings of the ITS America Annual Meeting. Houston, Texas, April 1996, pp. 576-580

[14] Dion F, Rakha H, "Estimating Spatial Travel Times Using Automatic Vehicle Identification Data", Transportation Research Board Preprint CD-ROM, 2003 Washington, D.C.

[15] S. Levine, W. McCasland, "Monitoring Freeway Traffic Conditions with Automatic Vehicle Identification Systems", ITE Journal, Vol. 64, No. 3, March 1994, pp. 23-28.

[16] B. L. Smith, H. Zhang, M. Fontaine, M. Green, "Cell Phone Probes as an ATMS Tool", Research Report No. UVACTS-15-5-79, Center for Transportation Studies at the University of Virginia, Charlottesville, VA, 2003.

[17] Guiyan Jiang, Longhui Gang, Zhili Cai, "Impact of Probe Vehicles Sample Size on Link Travel Time Estimation", IEEE Intelligent Transportation Systems Conference, ITSC '06, Sept. 2006, pp. 505 - 509

[18] Liu Y., Lin P.-W., Lai X., Chang G-L, Marquess A., "Developments and Applications of Simulation-Based Online Travel Time Prediction System: Traveling to Ocean City, Maryland", Transportation Research Record 1959, Transportation Research Board, Washington, D.C, pp. 92-104, 2006b.

[19] Van Lint J. W. C., Van der Zijpp N. J., “Improving A Travel Time Estimation Algorithm by Using Dual Loop Detectors”, Transportation Research Record, Transportation Research Board, Washington D.C., pp. 41–48, 2003.

[20] Loren Bloomberg, Jim Dale, A Comparison of the VISSIM and CORSIM Traffic Simulation Models, Paper prepared for the Institute of Transportation Engineers Annual Meeting, 2000.

[21] Punzo, Vincenzo; Ciuffo, Biagio; Torrieri, Vincenzo, “How Parameters of Microscopic Traffic Flow Models Relate to Traffic Dynamics in Simulation: Implications for Model Calibration”, Transportation Research Board 88th Annual Meeting, CD-ROM, Washington D.C., 2009.

[22] Yu, Lei; Xu, Yaofang; Song, Guohua; Hao, Yanzhao; Guo, Shuxia; Shi, Qinyi, “Development and Application of Macroscopic Emissions Model for China”, Transportation Research Board 88th Annual Meeting, CD-ROM, Washington D.C., 2009.

[23] Gangi, Massimo Di; Velonà Pietro, “Use of Multimodal Mesoscopic Approach in Modeling Pedestrian Evacuation”, Transportation Research Board 88th Annual Meeting, CD-ROM, Washington D.C., 2009

[24] Menneni Sandeep; Sun Carlos; Vortisch Peter, Integrated Microscopic and Macroscopic Calibration for Psychophysical Car-Following Models, Transportation Research Board 88th Annual Meeting, CD-ROM, Washington D.C., 2009

[25] W. Burghout, H. Koutsopoulos, and I. Andreasson, "Hybrid mesoscopic-microscopic traffic simulation," Transportation Research Record: 1934, pp. 218-225, 2005.

[26] Coifman B., “Estimating Travel Times and Vehicle Trajectories on Freeways Using Dual Loop Detectors”, Transportation Research A, 36 No 4, 2002, pp. 351-364.

[27] Lindveld C. D. R., Thijis R., Bovy P. H., Van der Zijpp N. J., “Evaluation of Online Travel Time Estimators and Predictors”, Transportation Research Record, 1719, Transportation Research Board, Washington, D.C. paper No 206, pp. 45-53, 2000

[28] Ethem ALPAYDIN, “Introduction to Machine Learning”, The MIT press, October 2004.

[29] Rice J., Van Zwet E., “A simple and effective method for predicting travel times on freeways”, Proceedings of the IEEE Conference on Intelligent Transportation Systems, Oakland, CA, United States, 2001, pp. 227–232.

[30] Nikovski D., Nishiuma N., Goto Y., Kumazawa H., “Univariate short-term prediction of road Travel Times”, Intelligent Transportation Systems, Proceedings. IEEE, 13-15 Sept. 2005 Page(s):1074 - 1079

[31] Kwon, J., Coifman, B., and Bickel, P., (2000). "Day-to-Day Travel Time Trends and Travel Time Prediction from Loop Detector Data," Transportation Research Record no. 1717, Transportation Research Board, pp. 120-129.

[32] Ishak S., Al-Deek H., “Performance evaluation of short-term time-series traffic prediction model”, Journal of Transportation Engineering, 2002, 128(6), 490–498.

[33] Ahmed M. S., Cook, A. R., “Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques”, Transportation Research Record, 722, Transportation Research Board, Washington, DC, pp. 1-9, 1979.

[34] Billings, D. Jiann-Shiou Yang, “Application of the ARIMA Models to Urban Roadway Travel Time Prediction - A Case Study”, IEEE International Conference on Systems, Man and Cybernetics, SMC, 2006. Pp. 2529-2534.

[35] Williams B. M, “Multivariate traffic flow prediction: An evaluation of arimax modelling”, Transportation Research Board 80th Annual Meeting, CD-Rom, National Academies Press, Washington D.C. USA, 2001.

[36] http://en.wikipedia.org/wiki/Kalman_filter

[37] Shou-Ren Hu, Chi-Bang Chen, “Dynamic estimation of freeway origin-destination demand and travel time using extended Kalman filtering algorithm**”,** IEEE International Conference on Networking, Sensing and Control, 2004, pp. 1329 – 1334

[38] Van Lint, J.W.C., “Online Learning Solutions for Freeway Travel Time Prediction IEEE Transactions on Intelligent Transportation Systems”, 2008, pp. 38 – 47

[39] Hao Liu, van Lint, H., van Zuylen, H.; Ke Zhang, “Two distinct ways of using kalman filters to predict urban arterial travel time**”,** IEEE Intelligent Transportation Systems Conference (ITSC '06), 2006, pp.845 – 850

[40] Thijs, R., Lindveld, C. D. R., Van der Zijpp, N. J., Bovy, P. H. L. “Evaluation of Travel Time Estimation and Prediction Algorithms: Evaluation Results from the DACCORD Project”, Transportation and Planning Section, Faculty of Civil Engineering and Geosciences, Delft University of Technology.

[41] Dougherty, M. (1995), ‘A review of neural networks applied to transport’, Transportation Research C 3(4), 247–260.

[42] Fu L., Rilett L.R., “Dynamic O-D Travel Time estimation using an artificial neural network”, Vehicle Navigation and Information Systems Conference, 1995. Proceedings. In conjunction with the Pacific Rim TransTech Conference. 6th International VNIS. 'A Ride into the Future' 30 July-2 Aug. 1995 Page(s):236 - 242

[43] Mark, C.D.; Sadek, A.W.; Rizzo, D. Predicting experienced Travel Time with neural networks: a PARAMICS simulation study.; Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on 3-6 Oct. 2004 Page(s):906 - 911

[44] Jeong, R.; Rilett, R.; Bus arrival time prediction using artificial neural network model Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on 3-6 Oct. 2004 Page(s):988 - 993

[45] DONGJOO PARK, RILETT L. R. “Forecasting freeway link travel times with a multilayer feedforward neural network”, Computer–Aided Civil and Infrastructure Engineering. 1999 , pp. 357-367

[46] Zou Liang; Zhu Ling-xiang; “Link Travel Time Estimation Model Fusing Data from Mobile and Stationary Detector Based on BP Neural Network”, International Conference on Communications, Circuits and Systems Proceedings, 25-28 June 2006, pp. 2146 – 2149.

[47] Yang, F., Sun, H., Tao, Y. & Ran, B. (2004), Temporal difference learning with recurrent neural network in multi-step ahead freeway speed prediction, in ‘Transportation Research Board Annual Meeting, CD-Rom’, National Academies Press, Washington D.C.

[48] Van Lint, J. W. C., Hoogendoorn, S. P., and Van Zuylen, H. J., “Freeway Travel Time Prediction with State-Space Neural Networks - Modeling State-Space Dynamics with Recurrent Neural Networks”, Transportation Research Record, 1811, Transportation Research Board, Washington, D.C, pp. 30–39, 2002.

[49] Park, D., Messer, C. J., and Urbanik, T. II., "Short-term Traffic Volume Forecasting Using Radial Basis Function Neural Networks", Transportation Research Record, 1651,Transportation Research Board, Washington D.C., 1998, pp. 39-47.

[50] Park D., Rilett L. R., "Forecasting Multiple Period Freeway Link Travel Times Using Modular Neural Networks", Transportation Research Record, 1617, Transportation Research Board, Washington D.C., 1998, pp. 163-170.

[51] Abdulhai B., Porwal H., Recker W., "Short term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks", Proceedings of the 78th Annual Meeting of the Transportation Research Board, National Academies Press, CD-Rom, Washington D.C., 1999.

[52] Liu H., H. J. v. Zuylen, et al, "Urban Arterial Travel Time Prediction with State-Space Neural Networks and Kalman Filters", The 85th Annual Meeting of TRB, Washington D.C, 2006.

[53] Simon Haykin, "Kalman Filtering and Neural Networks", John Wiley & Sons, USA, 2001.

[54] Huisken, G. and E. C. V. Berkum. "A Comparative Analysis of Short-Range Travel Time Prediction Methods." 82nd Transportation Research Board Annual Meeting, CD-ROM Washington D.C., 2003.

[55] Shen Luou, Hadi Mohammed, Estimation of Segment Travel Time Based on Point Traffic Detector Measurements, Transportation Research Board 88th Annual Meeting, CD-ROM, Washington D.C., 2009.

[56] Coifman B., "Using Dual Loop Speed Traps to Identify Detector Errors", Transportation Research Record No. 1683, Transportation Research Board, 1999, pp.47-58.

[57] Coifman B., "Estimating Travel Times and Vehicle Trajectories on Freeways Using Dual Loop Detectors", Transportation Research A, 36 No 4, 2002, pp. 351-364.

[58] Coifman B., Cassidy M.J., "Automated Travel Time Measurement Using Vehicle Lengths From Loop Detector Speed Traps", Research Report UCB-ITS-PRR-2000-12, University of California, Berkeley, 2000.

[59] Wardrop J.G. Some Theoretical Aspects of Road Traffic Research, Proceedings of the Institute of Civil Engineers, Vol. 1-2, 1952, pp. 325-378.

[60] Garber, N., and Hoel L., Traffic and Highway Engineering, Third Edition, 2002

[61] Statistics Toolbox User Guide, Mathworks, 2007. P.p. 83-86.

[62] Balázs Csanád Csáji. Approximation with Artificial Neural Networks; Faculty of Sciences; Eötvös Loránd University, Hungary, 2001.

[63] Paul John Werbos, "The roots of backpropagation: from ordered derivatives to neural networks and political forecasting", National Science Foundation, 1994.

[64] Bishop, C.M., "Neural Networks for Pattern Recognition, Oxford", Oxford University Press, 2005.

[65] "What is overfitting and how can I avoid it?", http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-3.html

[66] Rich Caruana, Steve Lawrence, Lee Giles, Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping, Neural Information Processing Systems Conference, 2000.

[67] Howard Demuth, Mark Beale, Martin Hagan, "Neural Network Toolbox User's guide", Mathworks, 2005

[68] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[69] Mei Lam TAM, William H.K. LAM, Short-term Travel Time Prediction for Congested Urban Road Networks, Transportation Research Board 88th Annual Meeting, CD-ROM, Washington DC, 2009.

[70] D. Nikovski, N. Nishiuma, Y. Goto and H. Kumazawa, Univariate Short-Term Prediction of Road Travel Times, Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems Vienna, Austria, September 13-16, 2005. Pp. 1074 - 10